**January, 1991**

LIDS-TH 2036

**Research Supported By:**

NSF MIP-9015281
AFOSR 88-0032C
ARO DAAL03-86-K-0171

# A Stochastic Modeling Approach to Multiscale Signal Processing*

Chou, K.C.

# A STOCHASTIC MODELING APPROACH TO MULTISCALE SIGNAL PROCESSING

by

Kenneth Chien-ko Chou

S.B., Massachusetts Institute of Technology (1985)
S.M., Massachusetts Institute of Technology (1987)
E.E., Massachusetts Institute of Technology (1987)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 1991

Signature of Author ______________________________
Department of Electrical Engineering and Computer Science
May 13, 1991

Certified by ______________________________
Alan S. Willsky
Thesis Supervisor

Accepted by ______________________________
Arthur Smith
Chairman, Department Committee on Graduate Students

# A Stochastic Modeling Approach to Multiscale Signal Processing

by

Kenneth Chien-ko Chou

Submitted to the Department of Electrical Engineering
and Computer Science on May 13, 1991 in partial
fulfillment of the requirements for the degree of
Doctor of Philosophy

## Abstract

In recent years there has been much interest in multiscale signal analysis, a large part of which is due to the recent flurry of research in the study of the wavelet transform. Though multiscale analysis seems like a natural enough paradigm in which to solve various signal processing problems, there has been no satisfactory statistical theory to provide a means of formulating optimal estimation and identification problems and assessing the performance of solutions. This thesis provides a statistical framework for multiscale signal processing based on stochastic models motivated by the wavelet transform. We first consider models defined on lattices which are naturally motivated by the synthesis equation of the wavelet transform. These are state-space models in which the points at each level of the lattice represent the state of the process at that level or *scale*. Our models can be used to describe processes which possess self-similar characteristics and in fact they model rather well $1/f$-type processes as well as, perhaps surprisingly, other well-studied processes. We formulate the smoothing problem for our class of models and develop fast algorithms for computing the optimal smoother based on the wavelet transform. In addition we show how to deal optimally with the issue of applying the wavelet transform to model signals defined on finite-length intervals. We then consider state-space models which are defined on the homogeneous tree, a special case of the lattice, and which are not so directly tied to the structure of the wavelet transform. We formulate the smoothing problem for this class of models and, by taking advantage of the structure of the tree, we develop an optimal smoother which is a generalization of the Rauch-Tung-Striebel algorithm to the tree. The filtering step of the smoother is a natural extension of Kalman filtering which includes a new set of discrete Riccati equations. The analysis of these equations leads us to develop elements of a system theory which allow us to provide bounds for the filter error covariance as well as to provide asymptotic results. We then give examples of using our overall framework to smooth $1/f$-type processes as well as standard, stationary 1st-order Gauss-Markov processes. Our results using single-scale data show the relative richness of our model class in describing a wide variety of phenomena. In particular our examples of smoothing Gauss-Markov processes show that our smoother performs rather well in comparison to standard smoothers. Moreover, as are results using multiscale data show, our framework can be used to optimally fuse data at multiple scales and characterize the performance of this fusion with *no additional algorithmic complexity.* Furthermore, by using our smoothing algorithm on trees we can handle the case where our data may be distributed *arbitrarily* at various scales. Also, our approach can be easily applied to 2D, where the potential computational savings is even more dramatic. Finally, we apply our framework to a 1D version of the optical flow problem in computer vision as a means of illustrating the potential of a formalism.

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

## ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Alan Willsky, for his boundless source of ideas, enthusiasm and insight throughout the course of this work. He has been a valuable mentor and has provided me with unfailing support throughout my graduate work. I would especially like to thank him for giving me the opportunity to live in France three years ago (and, while we were there, for instilling in me a deep appreciation for fine wine!). I especially thank my hosts at IRISA, Albert Benveniste and Michel Basseville, for making me feel welcome during my stay there and for partaking in such a fruitful exchange of ideas during the early stages of this work. I would like to thank the members of my thesis committee for their input and guidance. Also, I thank Ramine Nikoukhah for his assistance last summer, the result of which is the section on the triangularization of the Hamiltonian dynamics in Chapter 3.

I wish to thank my officemates, past and present, for their friendship and their constant willingness to discuss a wide of range of ideas. They are Cuneyt Ozveren, Mike Chin, Mark Luettgen, and Peyman Milanfar. My friendships with Stuart Golden and Darrin Taylor have also been most valuable and the supportiveness of Clem Karl these past couple of months is much appreciated. My experience at LIDS has been extremely pleasant due to the openness of the students and the supportiveness of the staff, i.e. Kathleen O'Sullivan and Sheila Hegarty. I especially thank Sanjoy Mitter for being so supportive throughout my stay at LIDS.

My involvement in jazz these past few years has been really fun for me and it has made life far more than just an "ordinary gig". My thanks go to members, past and present, of Eastern Standard Time, for their comradship and musical inspiration. Special thanks to Chris Adler for being such a great person. I acknowledge the support of my close friends and family, without whom life would be considerably less rich. To Greg Williams, for being there when I needed him, to Tom Richardson for being my soul mate, and to Bernard LeGoff for just being himself! Keith Nabors deserves credit for putting up with my shenanigans throughout the years. Thanx go to Jennifer Schultz for making these past past few months special(XOXO).

Finally, as with all of my educational endeavors thus far, this thesis wouldn't have been possible without the constant support of my parents. I dedicate this thesis to them.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction and Background

## 1.1 Introduction

In recent years there has been considerable interest and activity in the signal and image processing community in developing multi-resolution processing algorithms. Among the reasons for this are the apparent or claimed computational advantages of such methods and the fact that representing signals or images at multiple scales is an evocative notion – it seems like a "natural" thing to do. One of the more recent areas of investigation in multiscale analysis has been the emerging theory of multiscale representations of signals and wavelet transforms [9, 20, 21, 22, 23, 28, 35, 36, 40, 54]. This theory has sparked an impressive flurry of activity in a wide variety of technical areas, at least in part because it offers a common unifying language and perspective and perhaps the promise of a framework in which a rational methodology can be developed for multiscale signal processing, complete with a theoretical structure that pinpoints when multiresolution methods might be useful and why.

It is important to realize, however, that the wavelet transform by itself is not the only element needed to develop a methodology for signal analysis. To understand this one need only look to another orthonormal transform, namely the Fourier transform which decomposes signals into their frequency components rather than its components at different resolutions. The reason that such a transform is useful is that its use simplifies the description of physically meaningful classes of signals and important classes of transformations of those signals. In particular stationary stochastic processes are *whitened* by the Fourier transform so that individual frequency components of such a process are statistically uncorrelated. Not only does this greatly simplify their analysis, but, it also allows us to deduce that frequency-domain operations such as Wiener or matched filtering–or their time domain realizations as linear shift-invariant systems–aren't just convenient things to do. They are in fact the *statistically optimal* things to do. In analogy, what is needed to complement wavelet transforms for the construction of a rational framework for multi-resolution signal analysis is the identification of a rich class of signals and phenomena whose description is simplified by wavelet transforms. Having this, we then have the basis for developing a methodology for *scale domain* filtering and signal processing, for deducing that such operations are indeed the right ones to use, and for developing a new and potentially powerful set of insights and perspectives on signal and image analysis that are complementary to those that are the heritage of Fourier.

In this thesis we develop a theory for multiresolution stochastic processes and models aimed at achieving the objectives of describing a rich class of phenomena and of providing the foundation for a theory of optimal multiresolution statistical signal processing. In developing this theoretical framework we have tried to keep in mind the three distinct ways in which multi-resolution features can enter into a signal or image analysis problem. First, the phenomenon under investigation may possess features and physically significant effects at multiple scales. For example, fractal models have often been suggested for the description of natural scenes, topography, ocean wave height, textures, etc. [4, 37, 38, 44]. Also, anomalous broadband transient events or spatially-localized features can naturally be thought of as the superposition of finer resolution features on a more coarsely varying background. As we will see, the modeling framework we describe is rich enough to capture such phenomena. For example, we will see that 1/f -like stochastic processes as in [55, 56] are captured in our framework as are, surprisingly, useful models of many other processes. Secondly, whether the underlying phenomenon has multi-resolution features or not, it may be the case that the data that has been collected is at several different resolutions. For example the resolutions of remote sensing devices operating in different bands– such as IR, microwave, and various band radars– may differ. Furthermore, even if only one sensor type is involved, measurement geometry may lead to resolution differences (for example, if zoomed and un-zoomed data are to be fused or if data is collected at different sensor-to-scene distances). As we will see, the framework we describe provides a natural way in which to design algorithms for such multisensor fusion problems in that treating the case of multiple sensors is no more difficult than treating the case of a single sensor. Furthermore, our approach leads to algorithms which are extremely efficient and highly parallelizable.

Finally, whether the phenomenon or data have multi-resolution features or not, the signal analysis *algorithm* may have such features motivated by the two principal manifestations of the at least superficially daunting complexity of many image processing problems. The first and more well-known of these is the use of multiresolution algorithms to combat the computational demands of such problems by solving coarse (and therefore computationally simpler) versions and using these to guide (and hopefully speed up) their higher resolution counterparts. Multigrid relaxation algorithms[10, 11, 39, 42] for solving partial differential equations are of this type as are a variety of computer vision algorithms. As we will see, the stochastic models we describe lead to several extremely efficient computational structures for signal processing.

The second and equally important issue of complexity stems from the fact that a multi-resolution formalism allows one to exercise very direct control over "greed" in signal and image reconstruction. In particular, many imaging problems are, in principle, ill-posed in that they require reconstructing more degrees of freedom then one has elements of data. In such cases one must "regularize" the problem in some manner, thereby guaranteeing accuracy of the reconstruction at the cost of some resolution. Since the usual intuition is precisely that one should have higher confidence in the reconstruction of lower resolution features, we are led directly to the idea of reconstruction at multiple scales, allowing the resolution-accuracy tradeoff to be confronted directly (e.g. some of these ideas are explored in [15] where a multiresolution approach is used to solve an inverse conductivity problem). As we will see the algorithms arising in our framework allow such multiscale reconstruction and provide the analytical tools both for assessing resolution versus accuracy and for correctly accounting for fine scale fluctuations as a source of "noise" in coarser scale reconstructions.

In this thesis we analyze classes of multiscale stochastic processes which are largely motivated by the structure of the wavelet transform. These processes are defined on lattices where each level of the lattice has the interpretation of a particular scale representation of the process. In our models scale plays the role of a time-like variable. For example our processes are Markov in scale rather than in time. The fact that scale is time-like for our models allows us to draw from the theories of dynamic systems and recursive estimation in analyzing the properties of our models and in developing efficient, highly parallelizable algorithms for performing optimal estimation. For our models defined on general lattices we develop a smoothing algorithm, an algorithm which computes estimates of a multiscale process based on multiscale data, which uses the wavelet transform to transform the overall smoothing problem into a set of independently computable, small 1D standard smoothing problems. Note that although we focus on 1D signals in this thesis, the fact that scale is a time-like variable is true as well in the case of 2D, where similar types of models lead to efficient recursive and iterative algorithms; the computational savings in this case are even more dramatic than in the case of 1D.

We also analyze a class of state-space models defined on the homogeneous tree, which is a special case of the lattice with a great deal of additional structure. This structure in fact allows us to extend the notion of Kalman filtering to trees where the filter recursion is performed in scale rather than in time. The analysis of our filter leads us to develop elements of a system theory on trees and to derive results on asymptotic properties of the filter. Our smoothing algorithm is a generalization of the Rauch-Tung-Striebel smoothing algorithm[45] to trees.

Finally, we give numerical results which give an indication of the richness of our models in describing processes and which provide an indication of the potential of our framework in handling multiscale data. In particular we give results which show that our models do rather well in smoothing $1/f$-type processes as well as standard, stationary 1st-order Gauss-Markov processes. The fact that our smoothers compare favorably to standard smoothers on single scale data must be viewed in the context of the fact that our smoothers handle the case of optimally fusing multiscale data with no additional complexity. To illustrate this we give examples of how naturally our general framework can be used to incorporate multiscale data of varying SNR's as well as multiscale data of varying degrees of coverage. We also provide an example in which we use our framework to compute optical flow. This illustrates the potential of our framework in solving a complex problem in computer vision in an efficient, multiscale fashion.

In the remainder of this chapter we motivate the development of our multiscale models by first giving a brief overview of the wavelet transform and showing how the transform leads naturally to the study of processes indexed on lattices. We then give some background on $1/f$-type processes and in particular we cite recent results on using wavelets to represent these processes which lead naturally to multiscale models. We follow this with an introduction to a class of state-space models defined on homogeneous trees, a special case of lattices, for which we provide a great deal of analysis. And finally, we conclude this chapter with an outline of this thesis.

## 1.2 The Wavelet Transform

While there are several ways in which to introduce and motivate our modeling framework, one that provides a fair amount of insight begins with the wavelet transforms. However, the key for modeling is not to view the transform as a method for analyzing signals but rather as a mechanism for *synthesizing* or *generating* such signals beginning with coarse representations and adding fine detail one scale at a time. Specifically, let us briefly recall the structure of multiscale representations associated with orthonormal wavelet transforms [21, 35].

The multiscale representation of a continuous signal $f(x)$ consists of a sequence of approximations of that signal at finer and finer scales where the approximations of $f(x)$ at the $m$th scale consists of a weighted sum of shifted and compressed (or dilated) versions of a basic <u>scaling function</u> $\phi(x)$:

$$f_m(x) = \sum_{n=-\infty}^{+\infty} f(m,n)\phi(2^m x - n) \tag{1.2.1}$$

As $m \rightarrow \infty$ the approximation consists of a sum of compressed, weighted, and shifted versions of the function $\phi(x)$ whose choice is far from arbitrary. In particular in order for the $(m+1)$st approximation to be a refinement of the $m$th, we require $\phi(x)$ to be representable at the next scale:

$$\phi(x) = \sum_n \sqrt{2} h(n) \phi(2x - n) \tag{1.2.2}$$

As shown in [21], $h(n)$ must satisfy several conditions for eq.(1.2.1) to be an orthonormal series and for several other properties of the representation to hold. In particular $h(n)$ must be the impulse response of a quadrature mirror filter (QMF) [21, 47], where the condition for $h(n)$ to be a QMF is as follows.

$$\sum_k h(k)h(k-2n) = \delta_n \tag{1.2.3}$$

which has the following frequency domain interpretation where $H(w)$ denotes the discrete-time Fourier transform of $h(n)$.

$$|H(w)|^2 + |H(w+\pi)|^2 = 1 \tag{1.2.4}$$

The simplest example of such a $\phi, h$ pair is the Haar approximation with

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & otherwise \end{cases} \tag{1.2.5}$$

and

$$h(n) = \begin{cases} \frac{\sqrt{2}}{2} & n = 0, 1 \\ 0 & otherwise \end{cases} \tag{1.2.6}$$

By considering the incremental detail added in obtaining the $(m+1)$st scale approximation from the $m$th, we arrive at the wavelet transform. Such a transform is based on a single function $\psi(x)$ that has the property that the full set of its scaled translates $\left\{2^{m/2}\psi(2^m x - n)\right\}$ form a complete orthonormal basis for $L^2$. In [21] it is shown that $\phi$ and $\psi$ are related via an equation of the form

$$\psi(x) = \sum_n \sqrt{2} g(n) \phi(2x - n) \tag{1.2.7}$$

where $g(n)$ and $h(n)$ form a *conjugate mirror filter* pair [47], and that

$$f_{m+1}(x) = f_m(x) + \sum_n d(m,n) 2^{m/2} \psi(2^m x - n) \tag{1.2.8}$$

Note that for $g(n)$ and $h(n)$ to form a conjugate mirror filter pair they must obey the following algebraic relationships.

$$\begin{aligned} \sum_k g(k)h(k-2n) &= 0 && (1.2.9) \\ \sum_k g(k)g(k-2n) &= \delta_n && (1.2.10) \\ \sum_k h(n)h(n-2k) + \sum_k g(n)g(n-2k) &= \delta_n && (1.2.11) \end{aligned}$$

One particular choice of $g(n)$ is the following.

$$g(n) = (-1)^n h(1-n) \tag{1.2.12}$$

Thus, $f_m(x)$ is simply the partial orthonormal expansion of $f(x)$, up to scale $m$, with respect to the basis defined by $\psi$. For example if $\phi$ and $h$ are as in eq.(1.2.5), eq.(1.2.6), then

$$\psi(x) = \begin{cases} 1 & 0 \leq x \ < 1/2 \\ -1 & 1/2 \leq x \ < 1 \\ 0 & otherwise \end{cases} \tag{1.2.13}$$

$$g(n) = \begin{cases} \frac{\sqrt{2}}{2} & n = 0 \\ -\frac{\sqrt{2}}{2} & n = 1 \\ 0 & otherwise \end{cases} \tag{1.2.14}$$

and $\left\{2^{m/2}\psi(2^m x - n)\right\}$ is the *Haar basis.*

Thus, we can view a function $f(x)$ as having coarse scale approximations $f_m(x)$ that live in subspaces of $L^2$ which we denote as $V_m$. Furthermore, these subspaces are nested, i.e. $...V_m \subset V_{m+1}...$, and in fact $\bigcup_m V_m = L^2$. The scaling functions $\phi(x)$ span these subspaces in that $V_m = \overline{span\{\phi(2^m x - n) \mid \forall n\}}$, while the wavelet functions $\psi(x)$ span the orthogonal complement between successive subspaces; i.e. $V_{m+1} \ominus V_m = \overline{span\{\psi(2^m x - n) \mid \forall n\}}$. The structure of these spaces is entirely governed by the choice of the filter pair $h(n)$, $g(n)$, which satisfy the algebraic properties in eq.'s(1.2.9-1.2.11).

One of the appealing features of the wavelet transforms for the analysis of signals is that they can be computed recursively in scale, from fine to coarse. Specifically, if we have the coefficients $\{f(m+1,\cdot)\}$ of the $(m+1)$st-scale representation we can "peel off" the wavelet coefficients at this scale and at the same time carry the recursion one complete step by calculating the coefficients $\{f(m,\cdot)\}$ at the next somewhat coarser scale:

$$f(m,n) = \sum_k h(2n-k)f(m+1,k) \tag{1.2.15}$$

$$d(m,n) = \sum_k g(2n-k)f(m+1,k) \tag{1.2.16}$$

Note that the operation in eq.'s(1.2.15,1.2.16)) is simply a linear convolution followed by downsampling by a factor of two. These equations are often referred to as the wavelet *analysis* equations. For convenience we define the operations in eq.'s(1.2.15,1.2.16)) formally in terms of "coarsening" and "differencing" operators, respectively. These operators map $l^2$ sequences, i.e. infinite square-summable sequences, into $l^2$ sequences and are defined as follows.

$$(H_m f(m+1,\cdot))_n \triangleq \sum_k h(2n-k)f(m+1,k) \tag{1.2.17}$$

$$(G_m f(m+1,\cdot))_n \triangleq \sum_k g(2n-k)f(m+1,k) \tag{1.2.18}$$

where the operators are indexed with the subscript $m$ to denote that they map sequences at scale $m+1$ to sequences at scale $m$[1]. From eq.'s(1.2.3,1.2.10,1.2.9 we have the following fundamental algebraic properties of the operators $H_m$ and $G_m$.

$$H_m H_m^* = I \tag{1.2.19}$$

$$G_m G_m^* = I \tag{1.2.20}$$

$$H_m G_m^* = 0 \tag{1.2.21}$$

[1] Note that for the case of infinite sequences the operators as defined here are precisely equivalent for each scale; i.e. they are not a function of $m$. However, we adhere to this notation for the reasons that a) we may allow for the QMF filter to *differ* at each scale and b) for the case of finite-length sequences the operators *are* in fact different at every scale due to the fact that the the number of points differ from scale to scale.

where "$*$" denotes the adjoint of the operator.

Reversing this process we obtain the synthesis form of the wavelet transform in which we build up finer and finer representations via a coarse-to-fine scale recursion:

$$f(m+1,n) = \sum_k h(2k-n)f(m,k) + \sum_k g(2k-n)d(m,k) \tag{1.2.22}$$

Eq.(1.2.22) is referred to as the wavelet *synthesis* equation. Expressed in terms of the operators $H_m$ and $G_m$ we have

$$f(m+1,n) = (H_m^* f(m,\cdot))_n + (G_m^* f(m,\cdot))_n \tag{1.2.23}$$

or

$$H_m^* H_m + G_m^* G_m = I \tag{1.2.24}$$

which is an expression of eq.(1.2.11) in operator form.

Thus, we see that the synthesis form of the wavelet transform defines a *dynamical* relationship between the coefficients $f(m,n)$ at one scale and those at the next. Indeed this relationship defines an infinite lattice on the points $(m,n)$, where $(m+1,k)$ is connected to $(m,n)$ if $f(m,n)$ influences $f(m+1,k)$. This structure is illustrated in Figure 1.2.1 for the case where $h(n)$ is a 4-tap filter, where each level of the lattice represents an approximation of our signal at some scale $m$. Note that the dynamics in eq.(1.2.22) are now with respect to *scale* rather than time. It is this fact which allows us to apply both ideas and intuition from the study of dynamic systems evolving in time to the case of multiscale representations of signals. Note also that the fact that scale plays the role of a time-like variable is true whether our representation is of a 1D signal or a 2D signal. This is an important fact which has implications for the development of models and algorithms in 2D in which we use this notion of recursion in scale.

In developing a class of conveniently parametrized models which describes a rich class of phenomena, we need additional structure to supplement our dynamical multiscale representations. If we view these multiscale representations more abstractly, much as in the notion of a state model, as capturing the features of signals up to a particular scale that are relevant for the "prediction" of finer-scale approximations, we

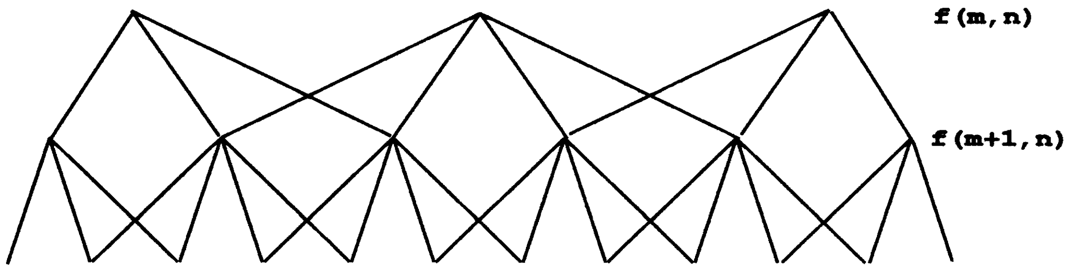

Figure 1.2.1: Infinite Lattice Representing Domain of Scaling Coefficients

can define rich classes of stochastic processes and models that contain the multiscale wavelet representations of eq.'s(1.2.15-1.2.22) as special (and in a sense degenerate) cases. Carrying this a bit farther, let us return to the point made in the introduction that for wavelet transforms to be useful it should be the case that their application simplifies the description or properties of signals. For example, this clearly would be the case for a stochastic process that is whitened by eq.(1.2.15), eq.(1.2.16), i.e. for which the wavelet coefficients $\{d(m,\cdot)\}$ at a particular scale are white and uncorrelated with the lower resolution version $\{f(m,\cdot)\}$ of the signal. In the next section we review results which indicate that one such class of processes, the class of processes known as *fractional Brownian motion*[38], is indeed simplified by wavelet analysis and that processes generated by driving the wavelet synthesis equation with uncorrelated coefficients can exhibit fractal behavior.

With wavelet coefficients which are uncorrelated, eq.(1.2.22) represents a first-order recursion in scale that is driven by white noise. However, as we know from time series analysis, white-noise-driven first-order systems yield a comparatively small class of processes which can be broadened considerably if we allow higher-order dynamics. In this thesis we introduce and investigate classes of stochastic state-space models on lattices for which the coefficients at each node of the lattice can now be finite-dimensional *vectors* rather than just scalars. As further motivation for such a framework, note that in sensor fusion problems one wishes to consider collectively an entire set of signals or images from a suite of sensors. In this case one is immediately confronted with the need to use higher-order models in which the actual observed signals may represent samples from such a model at *several* scales, corresponding to the differing resolutions of individual sensors.

In summary, the wavelet transform provides a unifying framework for the representation of signals at multiple scales. The structure of the transform can be interpreted in the context of infinite lattices, of which Figure 1.2.1 is an example, where each level of the lattice can be used to represent both the approximation of the signal at that scale (the scaling coefficients) and the difference between that approximation and the approximation at the next finest scale (the wavelet coefficients). From the point of view of *modeling*, the wavelet transform synthesis equation provides the impetus for defining dynamic state models in scale. This is the basis for developing efficient algorithms which are recursive in scale rather than in time. We emphasize that these ideas are easily extendible to the case of 2D where it remains natural to consider dynamic models and recursive algorithms in scale and where the potential for computational savings is even greater.

## 1.3 1/f-Type Processes

The study of $1/f$ processes originated in the modeling of physical phenomena which were observed to have long-term temporal dependencies. A model for these processes was proposed by Mandelbrot and Van Ness[38]. The class of processes described by this model is referred to as the class of fractional Brownian motion processes and it includes as a special case the ordinary Brownian motion process. It is a Gaussian zero-mean *nonstationary* process indexed by a single scalar parameter $0 < H < 1$ ($H = 1/2$ corresponds to ordinary Brownian motion). If we denote the process as $B_H(t)$, its covariance has the following form.

$$r_{B_H}(t,s) \triangleq E[B_H(t)B_H(s)]$$

$$= \frac{\sigma^2}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H}) \tag{1.3.25}$$

It was noted that these the increments of these processes are *stationary* and they obey the property of statistical self-similarity[38]; i.e. the statistics of the process are invariant to time scaling. This property of scale invariance already hints at the possible use of the wavelet transform as a useful analysis tool for these processes.

Since the fractional Brownian motion process is non-stationary, one would expect that the use of non-stationary transforms might simplify the analysis of such processes and perhaps lend further insight into their structure. Flandrin[25] uses a variety of non-stationary transforms in order to analyze fractional Brownian motion, including the Wigner-Ville transform and the continuous-scale wavelet transform ( the wavelet transform using continuous scale and translation parameters rather than powers of two). He makes precise the notion of spectrum for this non-stationary process by defining an *averaged* spectrum with respect to each of these transforms.

In the case of the Wigner-Ville transform , which is defined as

$$W(t,w) \triangleq \int_{-\infty}^{-\infty} r_{B_H}(t+\frac{\tau}{2}, t-\frac{\tau}{2})e^{-iw\tau}d\tau \tag{1.3.26}$$

the averaged spectrum is taken to be

$$S_W(w;T) \triangleq \frac{1}{T}\int_0^T W(t,w)dt \tag{1.3.27}$$

The result of taking the limit of this spectrum as the period goes to infinity is

$$\lim_{T\to\infty} S_W(w;T) = \frac{1}{|w|^{2H+1}} \tag{1.3.28}$$

Furthermore, for any finite $T$ and any given frequency $w_0$ it is shown that for $T \gg \frac{1}{w_0}$,

$$S_W(w_0;T) \approx \frac{1}{|w_0|^{2H+1}} \tag{1.3.29}$$

A similar result is shown using the continuous-scale wavelet transform, which is defined as

$$C(t,a) \triangleq \frac{1}{\sqrt{a}}\int_{-\infty}^{-\infty} B_H(s)g(\frac{s-t}{a})ds \tag{1.3.30}$$

where $a$ is the continuous scale parameter and $g(t)$ is the wavelet function. To define a notion of power spectrum in this case we define

$$r_{B_H}(t,s;a) \triangleq E[C(t,a)C(s,a)] \tag{1.3.31}$$

which turns out to be a function of $\frac{t-s}{a}$; in other words when analyzed relative to a given scale, $B_H(t)$ is stationary, which is intuitively satisfying given the fact that the increments of $B_H(t)$ are stationary. So by fixing $a$, the Fourier transform of $r_{B_H}(t,s;a)$, denoted as $S_C(w;a)$, is well defined. The averaged spectrum is taken to be the average of $S_C(w;a)$ taken over all scales and integrated using the normalizing measure for the continuous-scale wavelet transform.

$$S'_W(w) \triangleq \int_0^\infty S_C(w;a)\frac{da}{a^2} \tag{1.3.32}$$

$$= \frac{1}{|w|^{2H+1}} \tag{1.3.33}$$

These results show that in a very precise sense the spectrum of fractional Brownian motion is indeed $1/f$-like. Furthermore, the use of wavelet transforms seems to simplify a great deal of the analysis of these processes. In particular the wavelet coefficients of these non-stationary processes are stationary. It turns out that an even greater deal of simplification occurs by using wavelets to analyze these processes as borne out in the results of Tewfik[34]. These results consist of the analysis of the covariance of fractional Brownian motion, eq.(1.3.25), using the orthonormal wavelet transform. The results indicate that the wavelet coefficients of fractional Brownian motion are stationary and furthermore, the correlation between coefficients in both scale and translation decays hyperbolically fast. The fast decay of the coefficients is due to the vanishing moments property of $\psi(t)$ where an increase in the number of vanishing moments of $\psi(t)$ necessarily entails an increase in the order of the QMF filter $h(n)$. Thus, these results indicate that wavelets do very well in decorrelating or *whitening* the fractional Brownian motion process. This has strong implications for the modeling of $1/f$-type processes as shown in the results of Wornell[55].

The results of Wornell also concerns $1/f$-type processes and in particular he considers processes whose measured spectra are of the following form.

$$S(w) = \frac{\sigma^2}{|w|^\gamma} \tag{1.3.34}$$

where $0 < \gamma < 2$. Note that $\gamma$ ranges slightly differently from the fractional power associated with the spectrum of the fractional Brownian motion process. These spectra correspond to a particular range of physical phenomena as studied in the work of Keshner[33].

Motivated by Flandrin's[25] earlier use of wavelets to analyze $1/f$ processes, Wornell shows how the wavelet transform may be used to produce models which approximate $1/f$-type spectral characteristics. In particular he constructs models in which the wavelet coefficients $d(m,n)$ are completely uncorrelated with respect to both $m$ and $n$, and their variances obey the following power law in scale:

$$E[d^2(m,n)] = \sigma^2 2^{-\gamma m} \tag{1.3.35}$$

where $\sigma^2$ is a positive constant. He shows that the process resulting from driving the wavelet synthesis equations with coefficients obeying eq.(1.3.35) has an "averaged" spectrum of the following form, where the notion of averaging corresponds to assuming the model has uniformly distributed random phase:

$$\frac{k_1}{|w|^\gamma} \leq S(w) \leq \frac{k_2}{|w|^\gamma} \tag{1.3.36}$$

for some $0 < k_1 \leq k_2 < \infty$ which depend on the choice of wavelet function. Thus, the wavelet basis represents a Karhunen-Loeve-like expansion of $1/f$ processes. From the perspective of modeling this result shows that by assuming the wavelet coefficients are white and their variances are a function only of scale, the resulting process exhibits $1/f$-type spectra. Note that as we will show this corresponds exactly to a special case of one of our models. The fact that the variances of the coefficients are equivalent modulo a factor that varies geometrically with scale coincides intuitively with the fact that the increments of the fractional Brownian motion process are self-similar.

In summary, wavelets go a long way in simplifying the analysis of $1/f$-type processes. For example, the fact that while these processes are non-stationary their *increments* are stationary is characterized by the fact that the wavelet coefficients are stationary. Furthermore, the wavelet transform provides a Karhunen-Loeve-like decomposition in that the wavelet coefficients of these processes are nearly uncorrelated. Finally, by *setting* the the wavelet coefficients to be uncorrelated with variances decaying geometrically in scale, we can actually *synthesize* a process with an average spectrum corresponding to a $1/f$-type process. Thus, the wavelet transform can be used rather naturally to both analyze and model $1/f$-type processes.

## 1.4 Multiscale Representations and Trees

In this section we introduce the class of state-space models on homogeneous trees which is the subject of Chapter 3. Homogeneous trees are a special case of the lattice with a great deal of additional structure of which we take advantage in developing several highly efficient smoothing algorithms, including some that do *not* involve the wavelet transform. This additional structure also allows us to analyze much more deeply the structure of both our models and our algorithm.

Recall that the scaling coefficients, $f(m,n)$, of a function reside on an infinite lattice where the structure of the lattice is determined by the choice of wavelet representation. As a specific case the Haar representation naturally defines a dyadic tree structure on the points $(m,n)$ in which each point has two equally-weighted (i.e. $h(0) = h(1)$) descendants corresponding to the two subdivisions of the support interval of $\phi(2^m x - n)$, namely those of $\phi(2^{(m+1)}x - 2n)$ and $\phi(2^{(m+1)}x - 2n - 1)$. The structure of the tree is illustrated in Fig.1.4.2 where for notational convenience we denote each node of the tree by a single abstract index $t$, i.e. $t = (m,n)$, where $T$ denotes the set of all nodes and $m(t)$ denotes the scale or $m$-component of $t$. The development in the previous sections provide the motivation for the study of stochastic processes $x(m,n)$ defined on homogeneous trees (i.e. trees where the number of branches emanating from each node is constant) where $x(m,n)$ can in some sense be viewed as a stochastic model for the scaling coefficients of the Haar representation of a signal.

Let us make several comments about this case. First, as illustrated in Figure 1.4.2, with this and any of the other lattices associated with the wavelet transform, the scale index $m$ is time-like. For example it defines a natural direction of recursion for our representation: from coarse-to-fine in the synthesis of a signal and from fine to coarse in the analysis (e.g. in the Haar case $x(m,n)$ is directly obtainable from $x(m+1,2n)$,

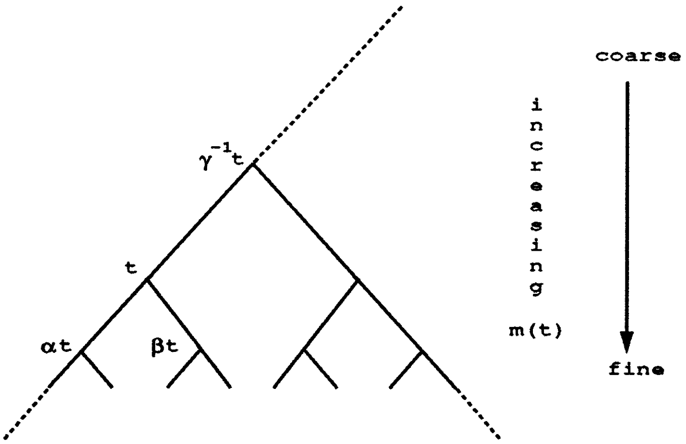

Figure 1.4.2: Dyadic Tree Representation

$x(m+1,2n+1)$). In the case of our tree, with increasing $m$ - i.e. the direction of synthesis - denoting the forward direction, we then can define a unique backward shift $\gamma^{-1}$ and two forward shifts $\alpha$ and $\beta$ (see Figure 1.4.2). In particular, $\alpha t = (m+1, 2n)$, $\beta t = (m+1, 2n+1)$, and $\gamma^{-1}t = (m-1, [\frac{n}{2}])$ where $[y]$ =integer part of $y$.

As in the synthesis description of multiscale representations, one of the classes of stochastic models we consider in this thesis are naturally described as evolving from coarse-to-fine scales. Specifically, we consider the following class of state-space models on trees:

$$x(t) = A(t)x(\gamma^{-1}t) + B(t)w(t) \tag{1.4.37}$$

where $\{w(t), t \in T\}$ is a set of independent, zero-mean Gaussian random variables. Note that this model is defined locally along the nodes of the tree and that the recursion is defined downwards in *scale.* We also note that while we develop our formalism for the case of dyadic trees there is a trivial extension of everything we do to the 2D case in which trees with four downward branches or *quadtrees* are used to model 2D processes. The use of a simple quadtree model of this type has been explored in the context of image coding and reconstruction[19, 52].

As we will see this formalism, which generalizes finite-dimensional state models to homogeneous trees, can be used to capture fractal-like behavior. Moreover, these models provide surprisingly accurate descriptions of a broad variety of stochastic processes and also lead to extremely efficient and highly parallelizable algorithms for optimal estimation and for the fusion of multiresolution measurements using multiscale, scale-recursive generalizations of Kalman filtering and smoothing.

## 1.5 Outline

The following is an outline of the remainder of this thesis.

- Chapter 2:
  We consider processes motivated directly by the structure of the wavelet transform which are defined on lattices. We construct state-space models for these processes in which *scale* plays the role of a time-like variable and then analyze the eigenstructure of these processes. We formulate the multiscale smoothing problem for our lattice processes and develop an algorithm which is highly efficient for smoothing single scale data and which incorporates multiscale data with *no added complexity.* This algorithm uses the wavelet structure of the process to decouple the computation into a set of small, independent standard smoothers. We present two versions of the algorithm, one which goes from coarse to fine then fine to coarse and another which goes from fine to coarse then coarse to fine. We then discuss the problem of applying the wavelet transform to finite-length data. The typical approach is to base the transform on cyclic convolutions rather than on linear convolutions and to perform the scale by scale recursion up to some specified coarse scale. We present a more general perspective on the problem of adapting the wavelet transform to finite-length data which includes as a *special case* the approach using cyclic convolutions as well as other approaches which provide modifications of the wavelet transform to provide Karhunen-Loeve expansions of windowed multiscale processes.

- Chapter 3:
  We formulate and analyze the multiscale smoothing problem for our class of

state-space models on homogeneous trees. These models can be thought of as a special case of one of our lattice models in which the lattice is associated with the Haar wavelet. We derive a two-sweep smoother for these models which is highly efficient and extremely parallelizable. We exploit the structure of the tree to which we generalize notions from Kalman filtering theory and linear system theory. Our Kalman filter propagates upwards along the tree, accumulating measurements from descendants nodes. This leads us to the analysis of a new set of Riccati equations. We then define notions of reachability, observability, and reconstructibility which allow us to give results on bounding the error covariance of our generalized Kalman filter. We also define a natural notion of stability for our processes which, along with our bounds on the error covariance, allows us to give results on the stability of our filter as well as asymptotic properties. Our overall smoothing algorithm for our tree processes is a generalization of the Rauch-Tung-Striebel algorithm to trees which consists of the filtering step as a first sweep up the tree followed by a sweep down the tree using a recursion that is driven by the previous filtered estimates. Furthermore, by viewing the smoothing problem from the Hamiltonian perspective we show that we can derive a two-sweep algorithm by factoring the Hamiltonian dynamic equations. In fact a property of these dynamics is the fact that they *cannot* be diagonalized; i.e. there is no two-filter smoothing algorithm on trees. We give a derivation of a Maximum-Likelihood version of the Rauch-Tung-Striebel algorithm based on *triangularizing* these dynamics. This algorithm has the added advantage of being able to handle sparse, *non-uniformly* sampled data. This is an important point of consideration with regard to sensor fusion problems. As in the case of our lattice models our algorithms apply equally well, with no difference in complexity, to both the case of single scale and the case of multiscale data. Finally, we present an iterative algorithm for solving the smoothing problem which bears some resemblance to multigrid algorithms. Like our Rauch-Tung-Striebel algorithm, this algorithm does not rely on the wavelet structure of the process and can be applied to general measurement scenarios. We develop this algorithm in the context of our tree models and show how it may be extended to the case of lattice models.

- Chapter 4:
  To demonstrate the applicability of our multiscale estimation framework and to give an indication of how well our models represent processes, we give numerical examples of the results of using our framework on specific problems. In particular we give examples of smoothing well-studied processes using our models. For the case of 1st-order Gauss-Markov processes we show that the relative difference in performance of standard smoothers using the "true" model versus that of our smoothers based on our multiscale models is arguably insignificant. We show that the same is true for the smoothing of 1/f-type processes. Furthermore, we demonstrate the performance of our smoothers for multiscale data as well as for sparse, non-uniformly sampled data. In the case of non-uniformly sampled data we give an example in which coarse data of full coverage is used to interpolate sparse fine scale data. Finally, we apply our framework to the optical flow problem. We formulate the problem in the context of a smoothing problem, defined precisely within in our framework, which can be solved extremely efficiently using our multiscale algorithms.

# Chapter 2

# Multiscale Processes on Lattices

## 2.1 Introduction

In this chapter we describe and analyze multiscale stochastic processes defined on lattices. We construct a class of state-space models which is directly motivated by the synthesis equation of the wavelet transform. We characterize the eigenstructure of these processes, which is essentially related to the wavelet transform. We then consider the multiscale fusion problem in which noisy measurements of our process at different scales must be fused optimally to estimate the process. In this context we develop an optimal fusion algorithm based on using the wavelet transform to diagonalize the smoothing problem, i.e. transform the problem into a set of independent subproblems, resulting in an efficient and parallelizable procedure. In fact by working in the wavelet-transform domain we reduce the multiscale smoothing problem to a set of independently computable 1D smoothing problems where the length of the smoothing interval is at most $log(N)$, where $N$ is the number points at the finest scale. Finally, we discuss the significant problem of adapting the wavelet transform to treat finite-length data. We characterize the problem in such a way as to suggest a variety of possible transforms which are finite-length and orthonormal. One of these examples is the case of using cyclic convolutions at each scale for both the coarsening as well as the differencing operations. This procedure amounts to assuming a periodic representation of the signal at each scale. Though the assumption of periodicity is commonly used to adapt the wavelet transform to finite sequences, this can potentially introduce unwanted distortions at the edges of the interval. Other possibilities include transforms which are *not* based on using cyclic convolutions. We develop in detail examples of these other possibilities and give an interpretation of these transforms as a type of windowing of the signal at multiple scales.

## 2.2 Eigenstructure of Lattice Models

In this section we define our class of lattice models which is directly motivated by the synthesis equation, eq.(1.2.22), of the wavelet transform and analyze its eigenstructure. As we will show, included in our class of models is precisely the model used by Wornell[55, 56] to describe $1/f$ processes. The Karhunen-Loeve expansion for the covariance function of our processes at a given scale is achieved exactly by the wavelet transform. We will use this fact to derive an efficient smoothing algorithm based on diagonalizing the problem using the wavelet transform. We develop our ideas for the case of the infinite lattice, corresponding to the wavelet transform as defined in Chapter 1, and later discuss the issue of adapting the wavelet transform to the case of finite-length data.

Consider an infinite lattice corresponding to a wavelet whose scaling filter, $h(n)$, is an FIR filter of length $P$. Recall that each level of the lattice can be viewed as the domain of an $l^2$ sequence representing either successively finer approximations of a signal, the scaling coefficients $f(m,n)$ where $f$ gets finer with increasing $m$, or the corresponding wavelet coefficients, $d(m,n)$. We define models whose state, $x(m) \in l^2$, can in some sense be viewed as the stochastic analog of the scaling coefficients at scale $m$. Here, $x(m)$ is defined to be an infinite-dimensional random vector whose elements are defined at a particular scale of the lattice. Let $H_m$ and $G_m$ denote the coarsening and differencing operators defined in eq.'s(1.2.17,1.2.18), where the subscript $m$ denotes that the operators map vectors at scale $m+1$ to vectors at scale $m$. Let $G_m^*$ and $H_m^*$ denote their respective adjoint operators, i.e. operators that map vectors at scale $m$ to vectors at scale $m+1$.

If we consider the synthesis equation of the wavelet transform, eq.(1.2.22), driven by uncorrelated wavelet coefficients $d(m,n)$, where the variances are constant along scale but varying from scale to scale, we get the following stochastic-dynamic state model where we define the scale index $m$ from an initial coarse scale, $L$, to the finest scale, $M$, and where we assume that the coarsest scaling coefficients $f(L,n)$ are uncorrelated. Thus, with $x(m)$ corresponding to $f(m,\cdot)$ and $w(m)$ to $d(m,\cdot)$ we have for $m = L, L+1, ...M-1$

$$\begin{aligned} E[x(L)x(L)^T] &= \tilde{\Lambda}_L \qquad (2.2.1) \\ &= \tilde{\lambda}_L I \\ x(m+1) &= H_m^* x(m) + G_m^* w(m) \qquad (2.2.2) \\ E[w(i)w(i)^T] &= \Lambda_i \qquad (2.2.3) \\ &= \lambda_i I \ , \ i = L, L+1, ...M-1 \end{aligned}$$

Note that if we let $\lambda_i = \sigma^2 2^{-\gamma i}$, i.e. the relationship in eq.(1.3.35), this model is precisely the one considered by Wornell for modeling a $1/f$-type process with spectral parameter $\gamma$. It is easy to show that the covariance of the process at each scale $m$, i.e. $E[x(m)x^T(m)]$, is as follows.

$$\begin{aligned} R_{xx}(m) &\triangleq E[x(m)x(m)^T] \qquad (2.2.4) \\ &= (\prod_{i=L}^{m-1} H_i^*)\tilde{\Lambda}_L(\prod_{m-1}^{i=L} H_i) \\ &+ \sum_{k=L}^{m-1}(\prod_{i=k+1}^{m-1} H_i^*)G_k^*\Lambda_k G_k(\prod_{i=m-1}^{k+1} H_i) \qquad (2.2.5) \end{aligned}$$

where we have, for simplicity of notation, adapted standard matrix-vector notation for $R_{xx}(m)$ and the infinite vector $x(m)$. We now proceed to analyze the eigenstructure of $R_{xx}(m)$. To do this we need the following properties of the operators $G_m$ and $H_m$, which are the fundamental algebraic relations between the coarsening and differencing operators as stated in eq.'s(1.2.19-1.2.21,1.2.24).

$$H_i H_i^* = I \ , \ G_i G_i^* = I \qquad (2.2.6)$$

$$H_i^* H_i + G_i^* G_i = I \qquad (2.2.7)$$

$$H_i G_i^* = 0 \qquad (2.2.8)$$

We begin by defining unit vectors in $l^2$ as follows.

$$\delta_i^j \triangleq [...0,...,0,\underbrace{1}_{i\text{th}},0,...0...]^T \tag{2.2.9}$$

where the superscript $j$ is used formally to denote that the vector corresponds to the $j$th scale of the lattice. Note that the superscript in this case is unnecessary since $\delta_i^j$ is not a function of scale. We adhere to this notation, since for the case of finite-length data $\delta_i^j$ *will* be a function of scale. The following lemma characterizes the eigenvectors for the correlation matrix $R_{xx}(m)$.

**Lemma 2.2.1** *The vectors $\overline{v}_i^L(m)$, $v_n^l(m)$ for $l = L, ...m-1$ and for $i, n \in \mathcal{Z}$ are eigenvectors of the correlation matrix at scale $m$, $R_{xx}(m)$, where*

$$\overline{v}_i^L(m) \triangleq (\prod_{j=L}^{m-1} H_j^*)\delta_j^L \tag{2.2.10}$$

*and*

$$v_n^l(m) \triangleq (\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \tag{2.2.11}$$

*The following holds:*

$$R_{xx}(m)\overline{v}_i^L(m) = \tilde{\lambda}_L\overline{v}_i^L(m) \tag{2.2.12}$$

$$R_{xx}(m)v_n^l(m) = \lambda_l v_n^l(m) \tag{2.2.13}$$

*for $l = L, ...m-1$, $i, n \in \mathcal{Z}$ where $\tilde{\lambda}_L, \lambda_l$ are scalars.*

## Proof

We show the existence of scalars $\tilde{\lambda}^L$ and $\lambda^l$ for $l = L, ...m-1$ such that $R_{xx}(m)\overline{v}_i^L(m) = \tilde{\lambda}^L\overline{v}_i^L(m)$ and $R_{xx}(m)v_n^l(m) = \lambda^l v_n^l(m)$ for $n = 0, 1, ...2^l - 1$ $i = 0, 1, ...2^L - 1$. Let

$$\Gamma_k \triangleq (\prod_{i=k+1}^{m-1} H_i^*)G_k^*\Lambda_k G_k(\prod_{i=m-1}^{k+1} H_i) \tag{2.2.14}$$

$$\Psi_L \triangleq (\prod_{i=L}^{m-1} H_i^*)\tilde{\Lambda}_L(\prod_{m-1}^{i=L} H_i) \tag{2.2.15}$$

so that

$$R_{xx}(m)v_n^l(m) = \Psi_L v_n^l(m) + (\sum_{k=L}^{m-1} \Gamma_k)v_n^l(m) \tag{2.2.16}$$

We now examine separately the structures of $\Psi_L v_n^l(m)$ and $\Gamma_k v_n^l(m)$. We begin with $\Psi_L v_n^l(m)$ by noting that for $l = L, ...m-1$

$$\begin{aligned} \Psi_L v_n^l(m) &= (\prod_{i=L}^{m-1} H_i^*)\tilde{\Lambda}_L(\prod_{m-1}^{i=L} H_i)(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \\ &= 0 \end{aligned} \tag{2.2.17}$$

which follows from eq.(2.2.6) applied $m-1-l$ times followed by the application of eq.(2.2.8). Next, we examine the structure of $\Gamma_k v_n^l(m)$ which can be written out explicitly as

$$\Gamma_k v_n^l(m) = \lambda_k(\prod_{i=k+1}^{m-1} H_i^*)G_k^*G_k(\prod_{i=m-1}^{k+1} H_i)(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \tag{2.2.18}$$

We consider the following three cases. For $k < l$: Using eq.(2.2.6) we can reduce eq.(2.2.18) to the following.

$$\Gamma_k v_n^l(m) = \lambda_k(\prod_{i=k+1}^{m-1} H_i^*)G_k^*G_k(\prod_{i=l}^{k+1} H_i)G_l^*\delta_n^l \tag{2.2.19}$$

Applying the orthogonality condition, eq.(2.2.8), to eq.(2.2.19), we arrive at the following.

$$\Gamma_k v_n^l(m) = 0 \tag{2.2.20}$$

For $k > l$: Using eq.(2.2.6) we can reduce eq.(2.2.18) to the following.

$$\begin{aligned} \Gamma_k v_n^l(m) &= \lambda_k(\prod_{i=k+1}^{m-1} H_i^*)G_k^*G_k(\prod_{i=l+1}^{k} H_i^*)G_l^*\delta_n^l \\ &= 0 \end{aligned} \tag{2.2.21}$$

where the last equality follows from eq.(2.2.8).

And finally, for $k = l$: Using eq.(2.2.6) we can reduce eq.(2.2.18) to the following.

$$\begin{aligned} \Gamma_l v_n^l(m) &= \lambda_l(\prod_{i=l+1}^{m-1} H_i^*)G_l^*G_lG_l^*\delta_n^l \\ &= \lambda_l(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \\ &= \lambda_l v_n^l(m) \end{aligned} \tag{2.2.22}$$

where the second equality results from the application of eq.(2.2.6) to $G_l G_l^*$.

Thus, from eq.(2.2.16) and eq.'s(2.2.20,2.2.21, 2.2.22) we get

$$R_{xx}(m) v_n^l(m) = \lambda_l v_n^l(m) \tag{2.2.23}$$

Finally, we consider the structures of $\Psi_L \overline{v}_n^L(m)$ and $\Gamma_k \overline{v}_n^L(m)$. Using eq.(2.2.8) we can write

$$\begin{aligned} \Gamma_k \overline{v}_n^L(m) &= \lambda_k (\prod_{i=k+1}^{m-1} H_i^*) G_k^* G_k (\prod_{i=L}^{k} H_i^*) \delta_n^L \\ &= 0 \end{aligned} \tag{2.2.24}$$

Using eq.(2.2.6) we can write

$$\begin{aligned} \Psi_L \overline{v}_n^L(m) &= \tilde{\lambda}_L (\prod_{i=L}^{m-1} H_i^*) \delta_n^L \\ &= \tilde{\lambda}_L \overline{v}_n^L(m) \end{aligned} \tag{2.2.25}$$

From eq.'s(2.2.24,2.2.25) we conclude that

$$R_{xx}(m) \overline{v}_n^L(m) = \tilde{\lambda}_L \overline{v}_n^L(m) \tag{2.2.26}$$

We have shown that with $\tilde{\lambda}^L = \tilde{\lambda}_L$ and $\lambda^l = \lambda_l$ for $l = L, ...m - 1$, $R_{xx}(m)\overline{v}_n^L(m) = \tilde{\lambda}^L \overline{v}_n^L(m)$ and $R_{xx}(m) v_n^l(m) = \lambda^l v_n^l(m)$, thus proving the lemma. □

Note that computing the representation of a sequence in the basis, $\left\{\overline{v}_i^L(m), v_n^l(m) \mid \quad i, n \in \mathcal{Z} \quad l = L, ...m - 1\right\}$, is done efficiently using the wavelet transform of the sequence, i.e. recursively computing in scale the wavelet coefficients via the analysis equations, eq.'s(1.2.15,1.2.16). We also have the following lemma which shows the eigenvectors to be mutually orthogonal.

**Lemma 2.2.2** *The vectors $\overline{v}_i^L(m)$, $v_n^l(m)$ for $l = L, ...m - 1$, $n = 0, 1...2^l - 1$, $i = 0, 1, ...2^L - 1$ are mutually orthogonal; i.e. a) $[v_i^j(m)]^T [v_k^l(m)] = \delta_{i-k, j-l}$, b) $[v_i^j(m)]^T [\overline{v}_k^L(m)] = 0$, and c) $[\overline{v}_i^L(m)]^T [\overline{v}_j^L(m)] = \delta_{i-j}$.*

**Proof**

We begin with part a). For $j < l$:

$$\begin{aligned}
[v_i^j(m)]^T[v_k^l(m)] &= (\delta_i^j)^T G_j(\prod_{s=m-1}^{j+1} H_s)(\prod_{s=l+1}^{m-1} H_s^*)G_l^*\delta_k^l && (2.2.27)\\
&= (\delta_i^j)^T G_j(\prod_{s=l}^{j+1} H_s)G_l^*\delta_k^l && (2.2.28)\\
&= 0 && (2.2.29)
\end{aligned}$$

where the second equality follows from eq.(2.2.6), while the third equality follows from eq.(2.2.8).

And similarly for $j > l$:

$$\begin{aligned}
[v_i^j(m)]^T[v_k^l(m)] &= (\delta_i^j)^T G_j(\prod_{s=m-1}^{j+1} H_s)(\prod_{s=l+1}^{m-1} H_s^*)G_l^*\delta_k^l && (2.2.30)\\
&= (\delta_i^j)^T G_j(\prod_{s=l+1}^{j} H_s^*)G_l^*\delta_k^l && (2.2.31)\\
&= 0 && (2.2.32)
\end{aligned}$$

where again the second equality follows from eq.(2.2.6), while the third equality follows from eq.(2.2.8).

Finally, for $j = l$:

$$\begin{aligned}
[v_i^l(m)]^T[v_k^l(m)] &= (\delta_i^l)^T G_l(\prod_{s=m-1}^{l+1} H_s)(\prod_{s=l+1}^{m-1} H_s^*)G_l^*\delta_k^l && (2.2.33)\\
&= (\delta_i^l)^T G_l G_l^*\delta_k^l && (2.2.34)\\
&= (\delta_i^l)^T\delta_k^l && (2.2.35)\\
&= \delta_{i-k,j-l} && (2.2.36)
\end{aligned}$$

We now proceed with part b).

$$\begin{aligned}
[v_i^j(m)]^T[\overline{v}_k^L(m)] &= (\delta_i^j)^T G_j(\prod_{s=m-1}^{j+1} H_s)(\prod_{s=L}^{m-1} H_s^*)\delta_k^L && (2.2.37)\\
&= (\delta_i^j)^T G_j(\prod_{s=L+1}^{j} H_s^*)G_L^*\delta_k^L && (2.2.38)\\
&= 0 && (2.2.39)
\end{aligned}$$

where the second equality follows from eq.(2.2.6), while the third equality follows from eq.(2.2.8).

Finally, part c).

$$\begin{aligned}
[\overline{v}_i^L(m)]^T[\overline{v}_j^L(m)] &= (\delta_i^L)^T(\prod_{s=m-1}^{L} H_s)(\prod_{s=L}^{m-1} H_s^*)\delta_j^L && (2.2.40)\\
&= (\delta_i^L)^T\delta_j^L && (2.2.41)\\
&= \delta_{i-j} && (2.2.42)
\end{aligned}$$

where the second equality again follows from eq.(2.2.6). □

We have the following corollaries to Lemmas 2.2.1,2.2.2 which will be used in the next section to diagonalize the smoothing problem.

**Corollary 2.2.1** *Given the collection of eigenvectors, $\{\overline{v}_i^L(m), v_n^l(m) \mid l = L, ...m - 1\ i, n \in \mathcal{Z}\}$, and the fine-to-coarse scaling operator $H_{m-1}$, the following holds.*

$$\begin{aligned}
H_{m-1}\overline{v}_i^L(m) &= \overline{v}_i^L(m-1) \quad i \in \mathcal{Z} && (2.2.43)\\
H_{m-1}v_n^l(m) &= v_n^l(m-1) \quad l = L, ...m-2 \ \ n \in \mathcal{Z} && (2.2.44)\\
H_{m-1}v_n^l(m) &= 0 \quad l = m-1 \ \ n \in \mathcal{Z} && (2.2.45)
\end{aligned}$$

**Proof**

Consider the case where $l = m - 1$. Note that from the orthogonality condition, eq.(2.2.8),

$$\begin{aligned}
H_{m-1}v_n^{m-1}(m) &= H_{m-1}G_{m-1}^*\delta_n^{m-1}\\
&= 0 && (2.2.46)
\end{aligned}$$

For $l = L, ..., m - 2$

$$\begin{aligned}
H_{m-1}v_n^l(m) &= H_{m-1}(\prod_{i=l+1}^{m-1} H_i^*)G_{m-1}^*\delta_n^{m-1}\\
&= (\prod_{i=l+1}^{m-2} H_i^*)G_l^*\delta_n^l\\
&= v_n^l(m-1) && (2.2.47)
\end{aligned}$$

And finally,

$$\begin{aligned} H_{m-1}\overline{v}_n^L(m) &= H_{m-1}(\prod_{i=L}^{m-1} H_i^*)\delta_n^L \\ &= (\prod_{i=L}^{m-2} H_i^*)\delta_n^L \\ &= \overline{v}_n^L(m-1) \end{aligned} \qquad (2.2.48)$$

□

**Corollary 2.2.2** *Given the collection of eigenvectors,* $\{\overline{v}_i^L(m), v_n^l(m) \mid l = L, ...m - 1\ i, n \in \mathcal{Z}\}$*, and the fine-to-coarse wavelet operator* $G_{m-1}$*, the following holds.*

$$G_{m-1}\overline{v}_i^L(m) = 0 \quad i \in \mathcal{Z} \qquad (2.2.49)$$

$$G_{m-1}v_n^l(m) = 0 \quad l = L, ...m-2 \ \ n \in \mathcal{Z} \qquad (2.2.50)$$

$$G_{m-1}v_n^l(m) = \delta_n^l \quad l = m-1 \ \ n \in \mathcal{Z} \qquad (2.2.51)$$

**Proof**

Consider the case where $l = m - 1$. From eq.(2.2.6)

$$\begin{aligned} G_{m-1}v_n^{m-1}(m) &= G_{m-1}G_{m-1}^*\delta_n^{m-1} \\ &= \delta_n^{m-1} \end{aligned} \qquad (2.2.52)$$

From the orthogonality condition, eq.(2.2.8), for $l = L, ..., m - 2$

$$\begin{aligned} G_{m-1}v_n^l(m) &= G_{m-1}(\prod_{i=l+1}^{m-1} H_i^*)G_{m-1}^*\delta_n^{m-1} \\ &= 0 \end{aligned} \qquad (2.2.53)$$

and also

$$\begin{aligned} G_{m-1}\overline{v}_n^L(m) &= G_{m-1}(\prod_{i=L}^{m-1} H_i^*)\delta_n^L \\ &= 0 \end{aligned} \qquad (2.2.54)$$

□

**Corollary 2.2.3** *Given the collection of eigenvectors, $\{\overline{v}_i^L(m), v_n^l(m) \mid l = L, ...m - 1 \; i, n \in \mathcal{Z}\}$, and the coarse-to-fine operator, $H_m^*$, the following holds.*

$$\begin{aligned} H_m^* \overline{v}_i^L(m) &= \overline{v}_i^L(m+1) \quad i \in \mathcal{Z} & (2.2.55) \\ H_m^* v_n^l(m) &= v_n^l(m+1) \quad l = L, ...m-1 \;\; n \in \mathcal{Z} & (2.2.56) \end{aligned}$$

**Proof**

To show eq.(2.2.55) we have

$$\begin{aligned} H_m^* \overline{v}_i^L(m) &= H_m^* (\prod_{j=L}^{m-1} H_j^*) \delta_j^L \\ &= \overline{v}_i^L(m+1) \quad i \in \mathcal{Z} & (2.2.57) \end{aligned}$$

To show eq.(2.2.56) we have

$$\begin{aligned} H_m^* v_n^l(m) &= H_m^* (\prod_{i=l+1}^{m-1} H_i^*) G_l^* \delta_n^l \\ &= v_n^l(m+1) \quad l = L, ...m-1 \;\; n \in \mathcal{Z} & (2.2.58) \end{aligned}$$

□

We now consider an important variation of the models we have considered so far. We consider lattice models in which the elements of the state vectors at each scale are no longer simply scalars but rather finite-dimensional *vectors*. In other words, if we let $d$ denote the dimension of these vectors and if $x_i(m)$ denotes the $i$th entry of $x(m)$, then $x_i(m) \in \mathcal{R}^d$. By abuse of notation we now let $H_m, G_m$ be vector versions of the coarsening and differencing operators; i.e.

$$\begin{aligned} (H_m f(m+1, \cdot))_n &\triangleq \sum_k h(2n-k) f(m+1, k) & (2.2.59) \\ (G_m f(m+1, \cdot))_n &\triangleq \sum_k g(2n-k) f(m+1, k) & (2.2.60) \\ f(m+1, k) &\in \mathcal{R}^d & (2.2.61) \end{aligned}$$

We specify our model as follows for $m = L, L+1, ...M-1$,

$$\begin{aligned} E[x(L)x(L)^T] &= \mathcal{P}_x(L) & (2.2.62) \\ x(m+1) &= H_m^* \mathcal{A}(m+1) x(m) + \mathcal{B}(m+1) w(m+1) & (2.2.63) \\ E[w(i)w(j)^T] &= \mathcal{Q}(i) \delta_{i-j} \;\; , \;\; i = L+1, ...M & (2.2.64) \end{aligned}$$

where

$$\begin{aligned}
\mathcal{A}(m) &\triangleq diag(..., A(m), ...A(m), ...) && (2.2.65)\\
\mathcal{B}(m) &\triangleq diag(..., B(m), ...B(m), ...) && (2.2.66)\\
\mathcal{Q}(m) &\triangleq diag(..., Q(m), ...Q(m), ...) && (2.2.67)\\
\mathcal{P}_x(L) &\triangleq diag(..., P_x(L), ...P_x(L), ...) && (2.2.68)
\end{aligned}$$

and where $A(m), B(m), Q(m)$, and $P_x(L)$ are *finite-dimensional* matrices representing the system matrix, the process noise matrix, the process noise covariance matrix, and the initial state covariance matrix, respectively. As we will show, the diagonal operators $\mathcal{A}(m)$, $\mathcal{B}(m)$, $\mathcal{Q}(m)$, and $\mathcal{P}_x(L)$ do not affect the block eigenspace structure of $x(m)$, but do affect the eigenvalues. Note that by augmenting our model class to include finite-dimensional state vectors defined on lattices, we allow for the possibility of higher-order models. This allows us to consider a considerably richer class of processes which is parametrized by the matrices corresponding to our state model. Note also that this model bears resemblance to Laplacian Pyramid schemes[12] where the added detail in going from one scale to the next is *not* constrained by the differencing operator $G_m$. Finally, this is the type of model we develop on trees in the next chapter which corresponds to the case where $H_m^*$ corresponds to the Haar representation.

The covariance of the process at each scale $m$, i.e. $E[x(m)x^T(m)]$, is as follows where we have used the fact that the operators $\mathcal{A}(m)$, $\mathcal{B}(m)$, $\mathcal{Q}(m)$, $\mathcal{P}_x(L)$ and their adjoints commute with the operators $H_m, H_m^*$.

$$\begin{aligned}
R_{xx}(m) &\triangleq E[x(m)x(m)^T] && (2.2.69)\\
&= (\overline{\Phi}(m-1, L)\mathcal{P}_x(L)\overline{\Phi}^*(m-1, L))(\prod_{i=L}^{m-1} H_i^*)(\prod_{m-1}^{i=L} H_i)\\
&+ \sum_{k=L+1}^{m-1} (\overline{\Phi}(m-1, k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1, k))(\prod_{i=k}^{m-1} H_i^*)(\prod_{i=m-1}^{k} H_i)\\
&+ \mathcal{B}(m)\mathcal{Q}(m)\mathcal{B}^*(m)
\end{aligned}$$

where for $i \geq j$

$$\overline{\Phi}(i, j) = \begin{cases} I & i = j \\ \mathcal{A}(i)\overline{\Phi}(i-1, j) & i > j \end{cases} \qquad (2.2.70)$$

It turns out that the block eigenstructure of this process is precisely the same as that of the process defined earlier in eq.'s(2.2.1-2.2.3), modulo the fact that we now must deal with $d \times d$ blocks as the basic elements of our decomposition. In fact what we will refer to as *block-eigenvectors* will have the same structure as in the case of our scalar model except that $H_m, G_m$ are now the appropriate vector versions and the resulting block-eigenvectors are infinite sequences of $d \times d$ *matrices* rather than an infinite sequence of scalars. The "eigenvalues", however, will differ as they will in fact now be $d \times d$ *eigenmatrices* which will be functions of the matrices $A(m), B(m), Q(m)$, and $P_x(L)$. We first redefine our unit vectors as follows.

$$\delta_i^j \triangleq [\ldots, 0_d, \ldots, 0_d, \underbrace{I_d}_{i\text{th}}, 0_d, \ldots 0_d, \ldots]^T \tag{2.2.71}$$

where the superscript $j$ is again used to denote that the vector (in $(l^2)^d$) corresponds to the $j$th scale of the lattice and where $I_d$ is the $d \times d$ identity matrix (and $0_d$ the $d \times d$ zero matrix). We have the following lemma concerning the block-eigenstructure of $R_{xx}(m)$, where the proof is patterned after that of Lemma 2.2.1.

**Lemma 2.2.3** *The block vectors $\overline{v}_i^L(m)$, $v_n^l(m)$ for $l = L, \ldots m-1$ and for $i, n \in \mathcal{Z}$ are block-eigenvectors of the correlation matrix at scale $m$, $R_{xx}(m)$, where*

$$\overline{v}_i^L(m) \triangleq (\prod_{j=L}^{m-1} H_j^*)\delta_j^L \tag{2.2.72}$$

*and*

$$v_n^l(m) \triangleq (\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \tag{2.2.73}$$

*The following holds:*

$$\begin{aligned} R_{xx}(m)\overline{v}_i^L(m) &= diag(\ldots, \tilde{\lambda}_L, \ldots \tilde{\lambda}_L, \ldots)\overline{v}_i^L(m) && (2.2.74) \\ R_{xx}(m)v_n^l(m) &= diag(\ldots, \lambda_l, \ldots \lambda_l, \ldots)v_n^l(m) && (2.2.75) \end{aligned}$$

*for $l = L, \ldots m-1$, $i, n \in \mathcal{Z}$ where $\tilde{\lambda}_L, \lambda_l$ are $d \times d$ matrices of the form*

$$\tilde{\lambda}_L = \sum_{k=L+1}^{m} (\Phi(k,L)B(k)Q(k)B^T(k)\Phi^T(k,L)) + \Phi(m-1,L)P_x(L)\Phi^T(m-1,L) \tag{2.2.76}$$

$$\lambda_l = \sum_{k=l+1}^{m} (\Phi(k,l)B(k)Q(k)B^T(k)\Phi^T(k,l)) \tag{2.2.77}$$

*where*

$$\Phi(i,j) = \begin{cases} I & i = j \\ A(i)\Phi(i-1,j) & i > j \end{cases} \tag{2.2.78}$$

**Proof**

We show the existence of matrices $\tilde{\lambda}^L$ and $\lambda^l$ for $l = L, ...m-1$ such that $R_{xx}(m)\bar{v}_i^L(m) = diag(..., \tilde{\lambda}_L, ...\tilde{\lambda}_L, ...)\bar{v}_i^L(m)$ and $R_{xx}(m)v_n^l(m) = diag(..., \lambda^l, ...\lambda^l, ...)v_n^l(m)$ for $i, n \in \mathcal{Z}$. Let

$$\Psi_L \triangleq (\overline{\Phi}(m-1,L)\mathcal{P}_x(L)\overline{\Phi}^*(m-1,L))(\prod_{i=L}^{m-1} H_i^*)(\prod_{m-1}^{i=L} H_i) \tag{2.2.79}$$

$$\Gamma_k \triangleq (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))(\prod_{i=k}^{m-1} H_i^*)(\prod_{i=m-1}^{k} H_i) \tag{2.2.80}$$

so that

$$R_{xx}(m)v_n^l(m) = \Psi_L v_n^l(m) + (\sum_{k=L+1}^{m-1} \Gamma_k)v_n^l(m) + (\mathcal{B}(m)\mathcal{Q}(m)\mathcal{B}^*(m))v_n^l(m) \tag{2.2.81}$$

Immediately, we have that

$$(\mathcal{B}(m)\mathcal{Q}(m)\mathcal{B}^*(m))v_n^l(m) = diag(..., B(m)Q(m)B^T(m), ....B(m)Q(m)B^T(m), ...)v_n^l(m) \tag{2.2.82}$$

We now examine separately the structures of $\Psi_L v_n^l(m)$ and $\Gamma_k v_n^l(m)$. We begin with $\Psi_L v_n^l(m)$ by noting that for $l = L, ...m-1$

$$\begin{aligned} \Psi_L v_n^l(m) &= (\overline{\Phi}(m-1,L)\mathcal{P}_x(L)\overline{\Phi}^*(m-1,L)) \\ &\times (\prod_{i=L}^{m-1} H_i^*)(\prod_{m-1}^{i=L} H_i)(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \\ &= 0 \end{aligned} \tag{2.2.83}$$

which follows from eq.(2.2.6) applied $m-1-l$ times followed by the application of eq.(2.2.8). Next, we examine the structure of $\Gamma_k v_n^l(m)$ which can be written out explicitly as

$$\begin{aligned} \Gamma_k v_n^l(m) &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k)) \\ &\times (\prod_{i=k}^{m-1} H_i^*)(\prod_{i=m-1}^{k} H_i)(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \end{aligned} \tag{2.2.84}$$

We consider the following three cases. For $k < l + 1$: Using eq.(2.2.6) we can reduce eq.(2.2.84) to the following.

$$\begin{aligned} \Gamma_k v_n^l(m) &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k)) \\ &\times (\prod_{i=k}^{m-1} H_i^*)(\prod_{i=l}^{k} H_i)G_l^*\delta_n^l \end{aligned} \tag{2.2.85}$$

By applying the orthogonality condition, eq.(2.2.8), we arrive at the following.

$$\Gamma_k v_n^l(m) = 0 \tag{2.2.86}$$

For $k > l + 1$: Using eq.(2.2.6) we can reduce eq.(2.2.84) to the following.

$$\begin{aligned} \Gamma_k v_n^l(m) &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \\ &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))v_n^l(m) \\ &= diag(\ldots,M_1,\ldots M_1,\ldots)v_n^l(m) & (2.2.87) \\ M_1 &= \Phi(m-1,k)B(k)Q(k)B^T(k)\Phi^T(m-1,k) & (2.2.88) \end{aligned}$$

And finally, for $k = l + 1$: Using eq.(2.2.6) we can reduce eq.(2.2.84) to the following.

$$\begin{aligned} \Gamma_k v_n^l(m) &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_n^l \\ &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))v_n^l(m) \\ &= diag(\ldots,M_2,\ldots M_2,\ldots)v_n^l(m) & (2.2.89) \\ M_2 &= \Phi(m-1,k)B(k)Q(k)B^T(k)\Phi^T(m-1,k) & (2.2.90) \end{aligned}$$

Thus, from eq.(2.2.81), eq.(2.2.82), and eq.'s(2.2.86,2.2.87, 2.2.89) we get

$$\begin{aligned} R_{xx}(m)v_n^l(m) &= (\sum_{k=l+1}^{m} (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))) \\ &\times v_n^l(m) \\ &= diag(\ldots,M_3,\ldots M_3,\ldots)v_n^l(m) & (2.2.91) \\ M_3 &= \sum_{k=l+1}^{m} (\Phi(m-1,k)B(k)Q(k)B^T(k)\Phi^T(m-1,k)) & (2.2.92) \end{aligned}$$

Finally, we consider the structures of $\Psi_L \overline{v}_n^L(m)$ and $\Gamma_k \overline{v}_n^L(m)$. Using eq.(2.2.8) we can write

$$\begin{aligned}
\Gamma_k \overline{v}_n^L(m) &= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))(\prod_{i=k}^{m-1} H_i^*)(\prod_{i=L}^{k-1} H_i^*)\delta_n^L \\
&= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))(\prod_{i=L}^{m-1} H_i^*)\delta_n^L \\
&= (\overline{\Phi}(m-1,k)\mathcal{B}(k)\mathcal{Q}(k)\mathcal{B}^*(k)\overline{\Phi}^*(m-1,k))\overline{v}_n^L(m) \\
&= diag(..., M_4(k), ...M_4(k), ...)\overline{v}_n^L(m) \qquad (2.2.93) \\
M_4(k) &= \Phi(m-1,k)B(k)Q(k)B^T(k)\Phi^T(m-1,k) \qquad (2.2.94)
\end{aligned}$$

Using eq.(2.2.6) we can write

$$\begin{aligned}
\Psi_L \overline{v}_n^L(m) &= (\overline{\Phi}(m-1,L)\mathcal{P}_x(L)\overline{\Phi}^*(m-1,L))(\prod_{i=L}^{m-1} H_i^*)\delta_n^L \\
&= (\overline{\Phi}(m-1,L)\mathcal{P}_x(L)\overline{\Phi}^*(m-1,L))\overline{v}_n^L(m) \\
&= diag(..., M_5, ...M_5, ...)\overline{v}_n^L(m) \qquad (2.2.95) \\
M_5 &= \Phi(m-1,L)P_x(L)\Phi^T(m-1,L) \qquad (2.2.96)
\end{aligned}$$

From eq.'s(2.2.81,2.2.93,2.2.95) we conclude that

$$\begin{aligned}
R_{xx}(m)\overline{v}_n^L(m) &= diag(..., (\sum_{k=L+1}^{m} M_4(k)) + M_5, ...(\sum_{k=L+1}^{m} M_4(k)) + M_5, ...) \\
&\times \overline{v}_n^L(m) \qquad (2.2.97)
\end{aligned}$$

In summary we have shown that with

$$\tilde{\lambda}_L = \sum_{k=L+1}^{m} (\Phi(k,L)B(k)Q(k)B^T(k)\Phi^T(k,L)) + \Phi(m-1,L)P_x(L)\Phi^T(m-1,L) \qquad (2.2.98)$$

$$\lambda_l = \sum_{k=l+1}^{m} (\Phi(k,l)B(k)Q(k)B^T(k)\Phi^T(k,l)) \qquad (2.2.99)$$

for $l = L, ...m-1$

$$\begin{aligned}
R_{xx}(m)\overline{v}_i^L(m) &= diag(..., \tilde{\lambda}_L, ...\tilde{\lambda}_L, ...)\overline{v}_i^L(m) \qquad (2.2.100) \\
R_{xx}(m)v_n^l(m) &= diag(..., \lambda_l, ...\lambda_l, ...)v_n^l(m) \qquad (2.2.101)
\end{aligned}$$

for $l = L, ...m-1$, $i, n \in \mathcal{Z}$, thus proving the lemma. □

## 2.3 Efficient Optimal Smoothing

In this section we consider the problem of optimally estimating one of our processes given sensors of varying SNR's and differing resolutions. We formulate this sensor fusion problem as an optimal smoothing problem in which the optimally smoothed estimate is formed by combining noisy measurements of our lattice process at various scales. In other words each sensor is modeled as a noisy observation of our process at some scale of the lattice. We treat the smoothing problem in a way which is sufficiently general to account for situations in which we have measurements at either one or multiple scales. For example, for the problem of smoothing one of our processes given measurements at a single scale we can think of having measurements exclusively at the finest scale of the lattice. In the case of the fusion problem in which we have measurements at more than one scale we consider having measurements at multiple scales of the lattice. In the following development we assume the model in eq.(2.2.63).

Consider the following multiscale measurements for $m = L, L+1, ...M$.

$$y(m) = \mathcal{C}(m)x(m) + v(m) \tag{2.3.102}$$

where

$$\begin{aligned} \mathcal{C}(m) &\triangleq diag(..., C(m), ...C(m), ...) && (2.3.103)\\ \mathcal{R}(m) &\triangleq diag(..., R(m), ...R(m), ...) && (2.3.104)\\ E[v(i)v(j)^T] &= \mathcal{R}(i)\delta_{i-j} && (2.3.105) \end{aligned}$$

and where $C(m)$ is a $b \times d$ matrix and $R(m)$ is a $b \times b$ matrix. We define the **smoothed** estimate, denoted as $x^s(m)$, to be the expected value of $x(m)$ conditioned on $y(i)$ for $i = L, L+1, ...M$; i.e.

$$x^s(m) = E[x(m)|y(L), ...y(M)] \tag{2.3.106}$$

We define the **coarse-to-fine filtered** estimate, to be the expected value of $x(m)$ conditioned on $y(i)$ for $i = L, L+1, ...m$; i.e

$$\hat{x}(m|m) = E[x(m)|y(L), ...y(m)] \tag{2.3.107}$$

We define the **coarse-to-fine one-step predicted** estimate to be the expected value of $x(m)$ conditioned on $y(i)$ for $i = L, L+1, ...m-1$; i.e

$$\hat{x}(m|m-1) = E[x(m)|y(L), ...y(m-1)] \tag{2.3.108}$$

From standard Kalman filtering theory, we can derive a recursive filter with its associated Riccati equations, where the recursion in the case of our lattice models is in the scale index $m$. We choose to solve the smoothing problem via the Rauch-Tung-Striebel(RTS) algorithm[45]. This gives us a correction sweep that runs recursively from fine to coarse scales with the initial condition of the recursion being the final point of the Kalman filter. The following equations describe the "down" sweep, i.e. the filtering step from coarse to fine scales.

For $m = L, ..., M$:

$$\hat{x}(m|m-1) = H_{m-1}^{*}\mathcal{A}(m)\hat{x}(m-1|m-1) \tag{2.3.109}$$

$$\hat{x}(m|m) = \hat{x}(m|m-1) + \mathcal{K}(m)[y(m) - \mathcal{C}(m)\hat{x}(m|m-1)] \tag{2.3.110}$$

$$\mathcal{K}(m) = \mathcal{P}(m|m-1)\mathcal{C}^{*}(m)\mathcal{S}(m) \tag{2.3.111}$$

$$\mathcal{S}(m) = (\mathcal{C}(m)\mathcal{P}(m|m-1)\mathcal{C}^{*}(m) + \mathcal{R}(m))^{-1} \tag{2.3.112}$$

$$\begin{aligned} \mathcal{P}(m|m-1) &= H_{m-1}^{*}\mathcal{A}(m)\mathcal{P}(m-1|m-1)\mathcal{A}^{*}(m)H_{m-1} \\ &+ \mathcal{B}(m)\mathcal{Q}(m)\mathcal{B}^{*}(m) \end{aligned} \tag{2.3.113}$$

$$\mathcal{P}^{-1}(m|m) = \mathcal{P}^{-1}(m|m-1) + \mathcal{C}^{*}(m)\mathcal{R}^{-1}(m)\mathcal{C}(m) \tag{2.3.114}$$

with initial conditions

$$\hat{x}(L|L-1) = 0 \tag{2.3.115}$$

$$\mathcal{P}(L|L-1) = \mathcal{P}_x(L) \tag{2.3.116}$$

We also have the following equations for the correction sweep of the Rauch-Tung-Striebel algorithm, i.e. the "up" sweep from fine to coarse scales.

For $m = M-1, M-2, ...L+1, L$:

$$x^{s}(m) = \hat{x}(m|m) + \mathcal{P}(m|m)\mathcal{A}^{*}(m+1)H_m\mathcal{P}^{-1}(m+1|m)[x^{s}(m+1) - \hat{x}(m+1|m)] \tag{2.3.117}$$

$$\mathcal{P}^{s}(m) = \mathcal{P}(m|m) + E(m)\left[\mathcal{P}^{s}(m+1) - \mathcal{P}(m+1|m)\right]E^{*}(m) \tag{2.3.118}$$

$$E(m) = \mathcal{P}(m|m)\mathcal{A}^{*}(m+1)H_m\mathcal{P}^{-1}(m+1|m) \tag{2.3.119}$$

with initial conditions

$$\begin{aligned} x^s(M) &= \hat{x}(M|M) && (2.3.120)\\ \mathcal{P}^s(M) &= \mathcal{P}(M|M) && (2.3.121) \end{aligned}$$

Note that we could equally have chosen to start the RTS algorithm going from fine to coarse scales followed by a correction sweep from coarse to fine, i.e. an up-down rather than the down-up algorithm just described. We treat this case in the next section.

We now proceed to show how the smoothing problem can be decomposed into a set of standard 1D smoothing problems, where we use the down-up version of the RTS algorithm. By transforming our state vectors and data, i.e. by representing them in the wavelet basis, we end up with a set of independent 1D RTS smoothing problems which can be computed in parallel. Let us define the following transformed quantities.

$$\begin{aligned} \hat{z}_{j,k}(m|m-1) &\triangleq (v_k^j(m))^T \hat{x}(m|m-1) && (2.3.122)\\ \overline{P}_{j,k}(m|m-1) &\triangleq (v_k^j(m))^T \mathcal{P}(m|m-1) v_k^j(m) && (2.3.123)\\ \hat{z}_{j,k}(m|m) &\triangleq (v_k^j(m))^T \hat{x}(m|m) && (2.3.124)\\ \overline{P}_{j,k}(m|m) &\triangleq (v_k^j(m))^T \mathcal{P}(m|m) v_k^j(m) && (2.3.125)\\ \hat{u}_{L,k}(m|m-1) &\triangleq (\overline{v}_k^L(m))^T \hat{x}(m|m-1) && (2.3.126)\\ \tilde{P}_{L,k}(m|m-1) &\triangleq (\overline{v}_k^L(m))^T \mathcal{P}(m|m-1) \overline{v}_k^L(m) && (2.3.127)\\ \hat{u}_{L,k}(m|m) &\triangleq (\overline{v}_k^L(m))^T \hat{x}(m|m) && (2.3.128)\\ \tilde{P}_{L,k}(m|m) &\triangleq (\overline{v}_k^L(m))^T \mathcal{P}(m|m) \overline{v}_k^L(m) && (2.3.129)\\ z_{j,k}^s(m) &\triangleq (v_k^j(m))^T z^s(m) && (2.3.130)\\ P_{j,k}^s(m) &\triangleq (v_k^j(m))^T P^s(m) v_k^j(m) && (2.3.131)\\ \tilde{z}_{L,k}^s(m) &\triangleq (v_k^L(m))^T z^s(m) && (2.3.132)\\ \tilde{P}_{L,k}^s(m) &\triangleq (v_k^L(m))^T P^s(m) v_k^L(m) && (2.3.133) \end{aligned}$$

These quantities represent the transformed versions of the predicted, filtered, and smoothed estimates in the Rauch-Tung-Striebel algorithm, along with their respective error covariances, in the transform domain. The eigenvectors used to transform these quantities involve, as previously defined, $d \times 1$ vector versions of the operators $H_i$ and $G_i$. We also need to represent the transformed data, where the data at each scale, $y(m)$, is an infinite-dimensional vector whose entries are finite-dimensional vectors of dimension $b \times 1$. We represent these vectors using eigenvectors which now involve $b \times 1$ vector versions of the operators $H_i$ and $G_i$. In particular, we define the following operators

$$\begin{aligned}(\overline{H}_m \overline{f}(m+1,\cdot))_n &\triangleq \sum_k h(2n-k)\overline{f}(m+1,k) && (2.3.134)\\ (\overline{G}_m \overline{f}(m+1,\cdot))_n &\triangleq \sum_k g(2n-k)\overline{f}(m+1,k) && (2.3.135)\\ \overline{f}(m+1,k) &\in \mathcal{R}^b && (2.3.136)\end{aligned}$$

and the following unit vectors

$$\overline{\delta}_i^j \triangleq [\ldots,0_b,\ldots,0_b,\underbrace{I_b}_{i\text{th}},0_b,\ldots 0_b,\ldots]^T \qquad (2.3.137)$$

We denote the corresponding block eigenvectors as $\mathcal{V}$ to distinguish them from their $d \times 1$ versions. In particular, we have

$$\begin{aligned}\overline{\mathcal{V}}_i^L(m) &\triangleq (\prod_{j=L}^{m-1} \overline{H}_j^*)\overline{\delta}_j^L && (2.3.138)\\ \mathcal{V}_n^l(m) &\triangleq (\prod_{i=l+1}^{m-1} \overline{H}_i^*)\overline{G}_l^*\overline{\delta}_n^l && (2.3.139)\end{aligned}$$

Finally, we define our transformed data as follows.

$$\begin{aligned}\overline{y}_{j,k}(m) &\triangleq (\mathcal{V}_k^j(m))^T y(m) && (2.3.140)\\ \tilde{y}_{L,k}(m) &\triangleq (\overline{\mathcal{V}}_k^L(m))^T y(m) && (2.3.141)\end{aligned}$$

Note that for each scale $m$, where $m = L+1,\ldots M$, our transform indices range as follows.

$$\begin{aligned}j &= L,\ldots m-1\\ k &\in \mathcal{Z} && (2.3.142)\end{aligned}$$

That is, for each $m$ other than at the coarsest scale, $L$, we transform our quantities so that they involve eigenvectors whose coarsest scale is $L$. As we will show, quantities associated with $m = L$ are *already* block-diagonalized, due to the fact that our model assumes a block-diagonalized structure at this coarsest scale. For example, the fact that the measurement at the coarsest scale, $y(L)$, has a block-diagonal covariance structure follows from the fact that $x(L)$ has a block-diagonal covariance structure and the fact that our measurements are uncorrelated along the nodes of the lattice.

The following lemma is essential to the results in this section.

**Lemma 2.3.1** *Let $\mathcal{F}(m)$ and $\mathcal{G}(m)$ be a block diagonal operators of the following form*

$$\begin{aligned} \mathcal{F}(m) &= diag(..., F(m), ...F(m), ...) & (2.3.143) \\ \mathcal{E}(m) &= diag(..., E(m), ...E(m), ...) & (2.3.144) \end{aligned}$$

*where $F(m)$ is a $d \times d$ matrix and $E(m)$ is a $b \times d$ matrix. We have the following relationships.*

$$\begin{aligned} (v_k^j(m))^T \mathcal{F}(m) &= F(m)(v_k^j(m))^T & (2.3.145) \\ (v_k^i(m))^T \mathcal{F}(m) v_l^j(m) &= F(m)\delta_{i-j,k-l} & (2.3.146) \\ (\overline{v}_k^L(m))^T \mathcal{F}(m) &= F(m)(\overline{v}_k^L(m))^T & (2.3.147) \\ (\overline{v}_k^L(m))^T \mathcal{F}(m) \overline{v}_l^L(m) &= F(m)\delta_{k-l} & (2.3.148) \\ \mathcal{E}(m) v_k^j(m) &= \mathcal{V}_k^j(m) E(m) & (2.3.149) \\ \mathcal{E}(m) \overline{v}_k^j(m) &= \overline{\mathcal{V}}_k^j(m) E(m) & (2.3.150) \\ \mathcal{E}^*(m) \mathcal{V}_k^j(m) &= v_k^j(m) E^T(m) & (2.3.151) \\ \mathcal{E}^*(m) \overline{\mathcal{V}}_k^j(m) &= \overline{v}_k^j(m) E^T(m) & (2.3.152) \end{aligned}$$

**Proof**

We begin with eq.(2.3.145).

$$\begin{aligned} (v_k^j(m))^T \mathcal{F}(m) &= (\delta_j^k)^T G_k (\prod_{i=k+1}^{m-1} H_i) \mathcal{F}(m) \\ &= (\delta_j^k)^T \mathcal{F}(m) G_k (\prod_{i=k+1}^{m-1} H_i) \end{aligned}$$

$$
\begin{aligned}
&= F(m)(\delta_j^k)^T G_k(\prod_{i=k+1}^{m-1} H_i) \\
&= F(m)(v_k^j(m))^T
\end{aligned}
$$

where the second equality follows from the fact that $\mathcal{F}(m)$ commutes with the operators $H_i$ and $G_i$.

We now consider eq.(2.3.146).

$$
\begin{aligned}
(v_k^i(m))^T \mathcal{F}(m) v_l^j(m) &= (\delta_i^k)^T G_k(\prod_{i=k+1}^{m-1} H_i)\mathcal{F}(m)(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_j^l \\
&= (\delta_i^k)^T \mathcal{F}(m) G_k(\prod_{i=k+1}^{m-1} H_i)(\prod_{i=l+1}^{m-1} H_i^*)G_l^*\delta_j^l \\
&= \begin{cases} F(m)(\delta_i^k)^T \delta_j^l & k = l \\ 0 & otherwise \end{cases} \\
&= \begin{cases} F(m)\delta_{i-j,k-l} & k = l \\ 0 & otherwise \end{cases}
\end{aligned}
$$

where the second equality follows from the fact that $\mathcal{F}(m)$ commutes with the operators $H_i$ and $G_i$ and where the third equality uses the properties in eq.'s(2.2.6,2.2.8). Similar arguments can be used to derive eq.'s(2.3.147,2.3.148).

We now consider eq.(2.3.149).

$$
\begin{aligned}
\mathcal{E}(m) v_k^j(m) &= \mathcal{E}(m)(\prod_{i=k+1}^{m-1} H_i^*)G_k^*\delta_j^k \\
&= (\prod_{i=k+1}^{m-1} \overline{H_i^*})\overline{G_k^*}\mathcal{E}(m)\delta_j^k \\
&= (\prod_{i=k+1}^{m-1} \overline{H_i^*})\overline{G_k^*}\overline{\delta}_j^k E(m) \\
&= \mathcal{V}_k^j(m) E(m)
\end{aligned}
$$

where the second equality follows from the fact that $\mathcal{E}(m)$ commutes with the operators $H_i^*$ and $G_i^*$ in the following way.

$$\mathcal{E}(m) H_i^* = \overline{H_i^*}\mathcal{E}(m) \tag{2.3.153}$$

$$\mathcal{E}(m) G_i^* = \overline{G_i^*}\mathcal{E}(m) \tag{2.3.154}$$

A similar argument can be used to derive eq.(2.3.150).

Finally, consider eq.(2.3.151).

$$\begin{aligned}
\mathcal{E}^*(m)\mathcal{V}_k^j(m) &= \mathcal{E}^*(m)(\prod_{i=k+1}^{m-1} \overline{H}_i^*)\overline{G}_k^*\overline{\delta}_j^k \\
&= (\prod_{i=k+1}^{m-1} H_i^*)G_k^*\mathcal{E}^*(m)\overline{\delta}_j^k \\
&= (\prod_{i=k+1}^{m-1} H_i^*)G_k^*\delta_j^k E^T(m) \\
&= v_k^j(m)E^T(m)
\end{aligned}$$

where the second equality follows from the fact that $\mathcal{E}^*(m)$ commutes with the operators $\overline{H}_i^*$ and $\overline{G}_i^*$.

$$\mathcal{E}^*(m)\overline{H}_i^* = H_i^*\mathcal{E}^*(m) \tag{2.3.155}$$

$$\mathcal{E}^*(m)\overline{G}_i^* = G_i^*\mathcal{E}^*(m) \tag{2.3.156}$$

A similar argument can be used to derive eq.(2.3.152). □

We now proceed to derive the equations for filtering and smoothing in the transform domain. As we will see, these results form the basis of a smoothing algorithm in the transform domain which consists of a collection of *independent* smoothing algorithms each of which smooths, in scale, a $d \times 1$ state vector rather than an infinite-dimensional vector. We begin with the one-step predicted estimate $\hat{x}(m|m-1)$.

**Definition 2.3.1** *The* **transformed one-step predicted** *estimates* $\hat{u}_{L,k}(m|m-1)$ *and* $\hat{z}_{j,k}(m|m-1)$ *are defined as follows.*

$$\hat{u}_{L,k}(m|m-1) \triangleq (\overline{v}_k^L(m))^T\hat{x}(m|m-1) \tag{2.3.157}$$

$$\hat{z}_{j,k}(m|m-1) \triangleq (v_k^j(m))^T\hat{x}(m|m-1) \tag{2.3.158}$$

**Lemma 2.3.2** *The transformed one-step predicted estimates* $\hat{u}_{L,k}(m|m-1)$ *and* $\hat{z}_{j,k}(m|m-1)$ *evolve according to the following equations.*

*For* $k \in \mathcal{Z}$ *we have*

$$\begin{aligned}
\hat{u}_{L,k}(m|m-1) &= A(m)\hat{u}_{L,k}(m-1|m-1) \qquad (2.3.159) \\
m &= L+1, L+2, \ldots M
\end{aligned}$$

*For $j = L, L+1, \ldots M-2 \quad k \in \mathcal{Z}$ we have*

$$
\begin{aligned}
\hat{z}_{j,k}(m|m-1) &= A(m)\hat{z}_{j,k}(m-1|m-1) \\
m &= j+2, j+3, \ldots M
\end{aligned}
\tag{2.3.160}
$$

*with the initial conditions $j = L, L+1, \ldots M-1 \quad k \in \mathcal{Z}$*

$$
\hat{z}_{j,k}(j+1|j) = 0 \tag{2.3.161}
$$

**Proof**

Multiplying both sides of eq.(2.3.109) by $(v_k^j(m))^T$, we get for $m = L+1, \ldots M$

$$
\begin{aligned}
(v_k^j(m))^T\hat{x}(m|m-1) &= \hat{z}_{j,k}(m|m-1) \\
&= (v_k^j(m))^T H_{m-1}^* \mathcal{A}(m)\hat{x}(m-1|m-1) \\
&= \begin{cases} (v_k^j(m-1))^T \mathcal{A}(m)\hat{x}(m-1|m-1) & j = L, \ldots m-2 \quad k \in \mathcal{Z} \\ 0 & j = m-1 \quad k \in \mathcal{Z} \end{cases} \\
&= \begin{cases} A(m)\hat{z}_{j,k}(m-1|m-1) & j = L, \ldots m-2 \quad k \in \mathcal{Z} \\ 0 & j = m-1 \quad k \in \mathcal{Z} \end{cases}
\end{aligned}
\tag{2.3.162}
$$

where the third equality follows from Corollary 2.2.1 and where the last equality follows from Lemma 2.3.1. A similar argument applies in deriving eq.(2.3.159). □

We now consider the error covariances $\mathcal{P}(m|m)$ and $\mathcal{P}(m|m-1)$ and show how each of these quantities propagates in the transform domain.

**Definition 2.3.2** *The* **diagonalized one-step predicted** *error covariances, $\overline{P}_{j,k}(m|m-1), \tilde{P}_{L,k}(m|m-1)$ are defined as follows.*

$$
\begin{aligned}
\overline{P}_{j,k}(m|m-1) &\triangleq (v_k^j(m))^T \mathcal{P}(m|m-1) v_k^j(m) && (2.3.163) \\
\tilde{P}_{L,k}(m|m-1) &\triangleq (\overline{v}_k^L(m))^T \mathcal{P}(m|m-1) \overline{v}_k^L(m) && (2.3.164)
\end{aligned}
$$

**Definition 2.3.3** *The* **diagonalized filtered** *error covariances, $\overline{P}_{j,k}(m|m), \tilde{P}_{L,k}(m|m)$ are defined as follows.*

$$
\begin{aligned}
\overline{P}_{j,k}(m|m) &\triangleq (v_k^j(m))^T \mathcal{P}(m|m) v_k^j(m) && (2.3.165) \\
\tilde{P}_{L,k}(m|m) &\triangleq (\overline{v}_k^L(m))^T \mathcal{P}(m|m) \overline{v}_k^L(m) && (2.3.166)
\end{aligned}
$$

**Lemma 2.3.3** *The diagonalized one-step predicted error covariances, $\overline{P}_{j,k}(m|m-1)$ and $\tilde{P}_{L,k}(m|m-1)$, evolve according to the following equations.*

*For $j = L, L+1, ...M-2 \quad k \in \mathcal{Z}$ we have*

$$\begin{aligned} \overline{P}_{j,k}(m|m-1) &= A(m)\overline{P}_{j,k}(m-1|m-1)A^T(m) + B(m)Q(m)B^T(m) \\ m &= j+2, j+3, ...M \end{aligned} \tag{2.3.167}$$

*with the initial conditions $j = L, L+1, ...M-1 \quad k \in \mathcal{Z}$*

$$\overline{P}_{j,k}(j+1|j) = B(j+1)Q(j+1)B^T(j+1) \tag{2.3.168}$$

*For $k \in \mathcal{Z}$ we have*

$$\begin{aligned} \tilde{P}_{L,k}(m|m-1) &= A(m)\tilde{P}_{L,k}(m-1|m-1)A^T(m) + B(m)Q(m)B^T(m) \\ m &= L+1, L+2, ...M \end{aligned} \tag{2.3.169}$$

*The diagonalized filtered error covariances, $\overline{P}_{j,k}(m|m)$ and $\tilde{P}_{L,k}(m|m)$, evolve according to the following equations.*

*For $j = L, L+1, ...M-1 \quad k \in \mathcal{Z}$ we have*

$$\overline{P}^{-1}_{j,k}(m|m) = \overline{P}^{-1}_{j,k}(m|m-1) + C^T(m)R^{-1}(m)C(m) \tag{2.3.170}$$

$$m = j+1, j+2, ...M$$

$$\overline{P}^{-1}_{L,k}(L|L) \triangleq P_x^{-1}(L) + C^T(L)R^{-1}(L)C(L) \tag{2.3.171}$$

*For $k \in \mathcal{Z}$ we have*

$$\tilde{P}^{-1}_{L,k}(m|m) = \tilde{P}^{-1}_{L,k}(m|m-1) + C^T(m)R^{-1}(m)C(m) \tag{2.3.172}$$

$$m = L+1, L+2, ...M$$

$$\tilde{P}^{-1}_{L,k}(L|L) \triangleq P_x^{-1}(L) + C^T(L)R^{-1}(L)C(L) \tag{2.3.173}$$

*Furthermore, the operators $\mathcal{P}(m|m)$ and $\mathcal{P}(m|m-1)$ are block-diagonalized by our eigenvectors $v^j_k(m)$ and $\overline{v}^L_i(m)$. Namely,*

$$(v^i_k(m))^T\mathcal{P}(m|m)v^j_l(m) = \overline{P}_{j,k}(m|m)\delta_{i-j,k-l} \tag{2.3.174}$$

$$\begin{aligned}
(\overline{v}_k^L(m))^T\mathcal{P}(m|m)\overline{v}_l^L(m) &= \tilde{P}_{L,k}(m|m)\delta_{k-l} && (2.3.175)\\
(v_k^i(m))^T\mathcal{P}(m|m-1)v_l^j(m) &= \overline{P}_{j,k}(m|m-1)\delta_{i-j,k-l} && (2.3.176)\\
(\overline{v}_k^L(m))^T\mathcal{P}(m|m-1)\overline{v}_l^L(m) &= \tilde{P}_{L,k}(m|m-1)\delta_{k-l} && (2.3.177)
\end{aligned}$$

**Proof**

We show our results by induction on $m$. From our initial condition, eq.(2.3.116), and our update equation, eq.(2.3.114), we have

$$\mathcal{P}(L|L) = (\mathcal{P}_x^{-1}(L) + \mathcal{C}^*(L)\mathcal{R}^{-1}(L)\mathcal{C}(L))^{-1} \qquad (2.3.178)$$

The fact that $\mathcal{P}_x(L), \mathcal{R}(L)$ and $\mathcal{C}(L)$ are block-diagonal implies that $\mathcal{P}(L|L)$ is block-diagonal. In particular,

$$\mathcal{P}^{-1}(L|L) = diag(..., \overline{P}_{L,k}^{-1}(L|L), ...\overline{P}_{L,k}^{-1}(L|L), ...) \qquad (2.3.179)$$

where

$$\begin{aligned}
\overline{P}_{L,k}^{-1}(L|L) &= \tilde{P}_{L,k}^{-1}(L|L)\\
&= P_x^{-1}(L) + C^T(L)R^{-1}(L)C(L) && (2.3.180)
\end{aligned}$$

Applying our prediction equation, eq.(2.3.113), to $\mathcal{P}(L|L)$ we get

$$\begin{aligned}
\mathcal{P}(L+1|L) &= H_L^*\mathcal{A}(L+1)\mathcal{P}(L|L)\mathcal{A}^*(L+1)H_L\\
&+ \mathcal{B}(L+1)\mathcal{Q}(L+1)\mathcal{B}^*(L+1) && (2.3.181)
\end{aligned}$$

By multiplying on the left by $(v_k^L(L+1))^T$ and on the right by $v_l^L(L+1)$ and by applying Lemma 2.3.1 and the fact that the operator $\mathcal{P}(L|L)$ is block diagonal, we get that

$$\begin{aligned}
(v_k^L(L+1))^T\mathcal{P}(L+1|L)v_l^L(L+1) &= B(L+1)Q(L+1)B^T(L+1)\delta_{k-l}\\
&= \overline{P}_{L,k}(L+1|L)\delta_{k-l} && (2.3.182)
\end{aligned}$$

Similarly, by multiplying on the left by $(\overline{v}_k^L(L+1))^T$ and on the right by $\overline{v}_l^L(L+1)$, we get that

$$\begin{aligned}
(\overline{v}_k^L(L+1))^T\mathcal{P}(L+1|L)\overline{v}_l^L(L+1) &= (A(L+1)\tilde{P}_{L,k}(L|L)A^T(L+1)\\
&+ B(L+1)Q(L+1)B^T(L+1))\delta_{k-l}\\
&= \tilde{P}_{L,k}(L+1|L)\delta_{k-l} && (2.3.183)
\end{aligned}$$

Thus, we have verified eq.(2.3.169) for $m = L+1$ and eq.(2.3.168) for $j = L$. We have also shown that $\mathcal{P}(m|m)$ and $\mathcal{P}(m|m-1)$ are block-diagonalized by our eigenvectors $v_k^j(m)$ and $\overline{v}_i^L(m)$ for $m = L+1$. By updating $\mathcal{P}(L+1|L)$ using eq.(2.3.114) then multiplying on the left by $(v_k^L(L+1))^T$ and on the right by $v_l^L(L+1)$, we get

$$\begin{aligned}(v_k^L(L+1))^T\mathcal{P}^{-1}(L+1|L+1)v_l^L(L+1) &= (\overline{P}_{L,k}^{-1}(L+1|L) \\ &+ C^T(L+1)R^{-1}(L+1)C(L+1))\delta_{k-l} \\ &= \overline{P}_{L,k}^{-1}(L+1|L+1)\delta_{k-l} \qquad (2.3.184)\end{aligned}$$

where the first equality follows from Lemma 2.3.1 and the fact that $\mathcal{P}(L+1|L)$, $\mathcal{C}(L+1)$, and $\mathcal{R}(L+1)$ are block diagonal. This verifies eq.(2.3.170) for $j = L$ and $m = L+1$. A similar argument verifies eq.(2.3.172) for $j = L$ and $m = L+1$.

We now assume that $\mathcal{P}(m-1|m-1)$ and $\mathcal{P}(m-1|m-2)$ are block-diagonalized by $v_k^j(m-1)$ and $\overline{v}_i^L(m-1)$. We first show that $\mathcal{P}(m|m-1)$ is block-diagonalized by $v_k^j(m)$ and $\overline{v}_i^L(m)$ and indeed satisfies eq.'s(2.3.167,2.3.168). Multiplying eq.(2.3.113) on the left by $(v_k^i(m))^T$ and on the right by $v_l^j(m)$ we get

$$\begin{aligned}(v_k^i(m))^T\mathcal{P}(m|m-1)v_l^j(m) &= (v_k^i(m))^T H_{m-1}^* \mathcal{A}(m)\mathcal{P}(m-1|m-1)\mathcal{A}^*(m)H_{m-1}v_l^j(m) \\ &+ (v_k^i(m))^T\mathcal{B}(m)\mathcal{Q}(m)\mathcal{B}^*(m)v_l^j(m) \\ &= \begin{cases} (A(m)\overline{P}_{j,k}(m-1|m-1)A^T(m) & \\ +B(m)Q(m)B^T(m))\delta_{i-j,k-l} & j = L, ...m-2 \;\; k \in \mathcal{Z} \\ (B(m)Q(m)B^T(m))\delta_{i-j,k-l} & j = m-1 \;\; k \in \mathcal{Z} \end{cases} \\ &= \overline{P}_{j,k}(m|m-1)\delta_{i-j,k-l} \qquad (2.3.185)\end{aligned}$$

where the second equality follows from our induction assumption, Corollary 2.2.1, Lemma 2.3.1, and the fact that $(v_k^j(m))^T v_k^j(m) = I_d$. A similar argument can be made to verify eq.(2.3.169).

The fact that $\mathcal{P}(m|m)$ is block-diagonalized by $v_k^j(m)$ and $\overline{v}_i^L(m)$ can be shown as follows. Taking the inverse of each side of eq.(2.3.114) then multiplying on the left by $(v_k^i(m))^T$ and on the right by $v_l^j(m)$, we get

$$\begin{aligned}(v_k^i(m))^T\mathcal{P}(m|m)v_l^j(m) &= (\overline{P}_{j,k}^{-1}(m|m-1) + C^T(m)R^{-1}(m)C(m))\delta_{i-j,k-l} \\ &= \overline{P}_{j,k}(m|m)\delta_{i-j,k-l} \qquad (2.3.186)\end{aligned}$$

where the first equality follows from eq.(2.3.185) and Lemma 2.3.1. A similar argument can be made to verify eq.(2.3.172). □

We now consider the updated estimate $\hat{x}(m|m)$. In considering the scale evolution of this quantity in the transform domain, we must bear in mind the fact that the data is transformed using eigenvectors, $\mathcal{V}_n^l(m)$, which are defined with respect to the operators $H_i$ and $G_i$ operating on $b \times 1$ vectors rather than on $d \times 1$ vectors. The following results show that in fact this does not present a problem with respect to preserving the independence of the filtering steps with respect to the different transform components.

**Definition 2.3.4** *The diagonalized or* **transformed** *data, $\overline{y}_{j,k}(m)$ and $\tilde{y}_{L,k}(m)$, are defined as follows.*

$$\overline{y}_{j,k}(m) \triangleq (\mathcal{V}_k^j(m))^T y(m) \tag{2.3.187}$$

$$\tilde{y}_{L,k}(m) \triangleq (\overline{\mathcal{V}}_k^L(m))^T y(m) \tag{2.3.188}$$

**Definition 2.3.5** *The* **diagonalized filtered** *estimates, $\hat{z}_{j,k}(m|m)$ and $\hat{u}_{L,k}(m|m)$, are defined as follows.*

$$\hat{z}_{j,k}(m|m) \triangleq (\mathcal{V}_k^j(m))^T \hat{x}(m|m) \tag{2.3.189}$$

$$\hat{u}_{L,k}(m|m) \triangleq (\overline{\mathcal{V}}_k^L(m))^T \hat{x}(m|m) \tag{2.3.190}$$

**Lemma 2.3.4** *The diagonalized filtered estimates, $\hat{z}_{j,k}(m|m)$ and $\hat{u}_{L,k}(m|m)$, evolve according to the following equations.*

*For $j = L, L+1, \ldots M-1$ and $k \in \mathcal{Z}$*

$$\begin{aligned} \hat{z}_{j,k}(m|m) &= \hat{z}_{j,k}(m|m-1) + \overline{K}_{j,k}(m)(\overline{y}_{j,k}(m) - C(m)\hat{z}_{j,k}(m|m-1)) \\ m &= j+1, \ldots M \end{aligned} \tag{2.3.191}$$

*For $k \in \mathcal{Z}$*

$$\begin{aligned} \hat{u}_{L,k}(m|m) &= \hat{u}_{L,k}(m|m-1) + \tilde{K}_{L,k}(m)(\overline{y}_{L,k}(m) - C(m)\hat{u}_{L,k}(m|m-1)) \\ m &= L, L+1, \ldots M \end{aligned} \tag{2.3.192}$$

*where*

$$\begin{aligned}\overline{K}_{j,k}(m) &\triangleq (v_k^j(m))^T\mathcal{K}(m)\mathcal{V}_k^j(m) &(2.3.193)\\ \tilde{K}_{L,k}(m) &\triangleq (\overline{v}_k^j(m))^T\mathcal{K}(m)\overline{\mathcal{V}}_k^j(m) &(2.3.194)\end{aligned}$$

*and furthermore,*

$$\begin{aligned}(v_k^i(m))^T\mathcal{K}(m)\mathcal{V}_l^j(m) &= \overline{K}_{j,k}(m)\delta_{i-j,k-l} &(2.3.195)\\ (\overline{v}_k^i(m))^T\mathcal{K}(m)\overline{\mathcal{V}}_l^j(m) &= \tilde{K}_{j,k}(m)\delta_{i-j,k-l} &(2.3.196)\end{aligned}$$

**Proof**

We begin by premultiplying eq.(2.3.110) by $(v_k^j(m))^T$. For $m = L, ...M$

$$\begin{aligned}(v_k^j(m))^T\hat{x}(m|m) &= \hat{z}_{j,k}(m|m) &(2.3.197)\\ &= \hat{z}_{j,k}(m|m) + (v_k^j(m))^T\mathcal{K}(m)[y(m) - \mathcal{C}(m)\hat{x}(m|m-1)]\end{aligned}$$

We now use the following fact, which follows from the fact that $\left\{v_n^l(m)\overline{v}_n^L(m)\right\}$ is a complete orthonormal basis for $l^2$.

$$\sum_{l=L}^{M}\sum_n v_n^l(m)(v_n^l(m))^T + \sum_n \overline{v}_n^L(m)(\overline{v}_n^L(m))^T = I_2 \qquad (2.3.198)$$

where $I_2$ is taken to be the identity operator for $l^2$; i.e. $I_2x = x$ for $x \in l^2$. Note that

$$\begin{aligned}I_2\mathcal{P}(m|m-1)I_2 &= (\sum_{l=L}^{M}\sum_n v_n^l(m)(v_n^l(m))^T &(2.3.199)\\ &+ \sum_n \overline{v}_n^L(m)(\overline{v}_n^L(m))^T)\\ &\times \mathcal{P}(m|m-1)(\sum_{l=L}^{M}\sum_n v_n^l(m)(v_n^l(m))^T + \sum_n \overline{v}_n^L(m)(\overline{v}_n^L(m))^T)\\ &= (\sum_{l=L}^{M}\sum_n v_n^l(m)\overline{P}_{l,n}(m|m-1)(v_n^l(m))^T\\ &+ \sum_n \overline{v}_n^L(m)\tilde{P}_{L,n}(m|m-1)(\overline{v}_n^L(m))^T)\end{aligned}$$

where the second equality follows from the fact that $\mathcal{P}(m|m-1)$ is diagonalized by $\left\{v_n^l(m), \overline{v}_n^L(m)\right\}$. The result

$$\mathcal{C}(m)\mathcal{P}(m|m-1)\mathcal{C}^*(m) = \mathcal{C}(m)(\sum_{l=L}^{M}\sum_n v_n^l(m)\overline{P}_{l,n}(m|m-1)(v_n^l(m))^T$$

$$\begin{aligned}
&+ \sum_{n} \overline{v}_n^L(m)\tilde{P}_{L,n}(m|m-1)(\overline{v}_n^L(m))^T)\mathcal{C}^*(m) \\
&= \sum_{l=L}^{M}\sum_{n} \mathcal{V}_n^l(m)C(m)\overline{P}_{l,n}(m|m-1)C^T(m)(\mathcal{V}_n^l(m))^T \\
&+ \sum_{n} \overline{\mathcal{V}}_n^L(m)C(m)\tilde{P}_{L,n}(m|m-1)C^T(m)(\overline{\mathcal{V}}_n^L(m))^T
\end{aligned} \tag{2.3.200}$$

follows from Lemma 2.3.1. Building on this we get

$$\begin{aligned}
(\mathcal{C}(m)\mathcal{P}(m|m-1)\mathcal{C}^*(m)+\mathcal{R}(m))^{-1} &= \sum_{l=L}^{M}\sum_{n} \mathcal{V}_n^l(m)(C(m)\overline{P}_{l,n}(m|m-1)C^T(m) \\
&+ R^{-1}(m))^{-1}(\mathcal{V}_n^l(m))^T \\
&+ \sum_{n} \overline{\mathcal{V}}_n^L(m)(C(m)\tilde{P}_{L,n}(m|m-1)C^T(m) \\
&+ R^{-1}(m))^{-1}(\overline{\mathcal{V}}_n^L(m))^T
\end{aligned} \tag{2.3.201}$$

It then follows, using Lemma 2.3.1, that

$$\begin{aligned}
\mathcal{K}(m) &= \mathcal{P}(m|m-1)\mathcal{C}^*(m)\Bigg[\sum_{l=L}^{M}\sum_{n} \mathcal{V}_n^l(m)(C(m)\overline{P}_{l,n}(m|m-1)C^T(m) \\
&+ R^{-1}(m))^{-1}(\mathcal{V}_n^l(m))^T + \sum_{n} \overline{\mathcal{V}}_n^L(m) \\
&\times (C(m)\tilde{P}_{L,n}(m|m-1)C^T(m) + R^{-1}(m))^{-1}(\overline{\mathcal{V}}_n^L(m))^T\Bigg] \\
&= \sum_{l=L}^{M}\sum_{n} \mathcal{P}(m|m-1)v_n^l(m)C^T(m)(C(m)\overline{P}_{l,n}(m|m-1)C^T(m) \\
&+ R^{-1}(m))^{-1}(\mathcal{V}_n^l(m))^T + \sum_{n} \mathcal{P}(m|m-1)\overline{v}_n^L(m)C^T(m) \\
&\times (C(m)\tilde{P}_{L,n}(m|m-1)C^T(m) + R^{-1}(m))^{-1}(\overline{\mathcal{V}}_n^L(m))^T
\end{aligned} \tag{2.3.202}$$

Now, we consider premultiplying $\mathcal{K}(m)$ by $(v_k^j(m))^T$.

$$\begin{aligned}
(v_k^j(m))^T\mathcal{K}(m) &= \overline{P}_{j,k}(m|m-1)C^T(m)(C(m)\overline{P}_{j,k}(m|m-1)C^T(m) \\
&+ R^{-1}(m))^{-1}(\mathcal{V}_k^j(m))^T
\end{aligned} \tag{2.3.203}$$

where we have used the fact that

$$(v_k^i(m))^T\mathcal{P}(m|m-1)v_l^j(m) = \overline{P}_{j,k}(m|m-1)\delta_{i-j,k-l} \tag{2.3.204}$$

Finally,

$$\begin{aligned} v_k^j(m))^T\mathcal{K}(m)[y(m) - \mathcal{C}(m)\hat{x}(m|m-1)] &= \overline{P}_{j,k}(m|m-1)C^T(m)(C(m)\overline{P}_{j,k}(m|m-1)C^T(m) \\ &+ R^{-1}(m))^{-1}(\mathcal{V}_k^j(m))^T[y(m) - \mathcal{C}(m)\hat{x}(m|m-1)] \\ &= \overline{P}_{j,k}(m|m-1)C^T(m)(C(m)\overline{P}_{j,k}(m|m-1)C^T(m) \\ &+ R^{-1}(m))^{-1}[\overline{y}_{j,k}(m) - C(m)\hat{z}_{j,k}(m|m-1)] \end{aligned} \tag{2.3.205}$$

From eq.'s(2.3.193,2.3.1972.3.205) it follows that

$$\hat{z}_{j,k}(m|m) = \hat{z}_{j,k}(m|m-1) + \overline{K}_{j,k}(m)(\overline{y}_{j,k}(m) - C(m)\hat{z}_{j,k}(m|m-1)) \tag{2.3.206}$$

thus, proving eq.(2.3.191). Eq.(2.3.192) can be derived in a similar manner. □

We now complete the diagonalization of our smoothing algorithm by deriving the recursions for our fine-to-coarse sweep in the transform domain. In particular we show how the recursion for both the smoothed estimate and its error covariance in the transform domain decouple.

**Definition 2.3.6** *The* **diagonalized smoothed** *estimates, $z^s_{j,k}(m)$ and $\tilde{z}^s_{L,k}(m)$, are defined as follows.*

$$z^s_{j,k}(m) \triangleq (v_k^j(m))^T z^s(m) \tag{2.3.207}$$

$$\tilde{z}^s_{L,k}(m) \triangleq (v_k^L(m))^T z^s(m) \tag{2.3.208}$$

**Lemma 2.3.5** *The diagonalized smoothed estimates, $z^s_{j,k}(m)$ and $\tilde{z}^s_{L,k}(m)$, evolve according to the following equations.*

*For $j = L, L+1, ...M-1$ and $k \in \mathcal{Z}$*

$$\begin{aligned} z^s_{j,k}(m) &= \hat{z}_{j,k}(m|m) + \overline{P}_{j,k}(m|m)A^T(m+1)\overline{P}^{-1}_{j,k}(m+1|m)[z^s_{j,k}(m+1) - \hat{z}_{j,k}(m+1|m)] \\ m &= M-1, M-2, ...L+1, L \end{aligned} \tag{2.3.209}$$

*with initial condition*

$$z^s_{j,k}(M) = \hat{z}_{j,k}(M|M) \tag{2.3.210}$$

*For $k \in \mathcal{Z}$*

$$\begin{aligned} \tilde{z}^s_{L,k}(m) &= \hat{u}_{j,k}(m|m) + \tilde{P}_{j,k}(m|m)A^T(m+1)\tilde{P}^{-1}_{j,k}(m+1|m)[\tilde{z}^s_{j,k}(m+1) - \hat{u}_{j,k}(m+1|m)] \\ m &= M-1, M-2, ...L+1, L \end{aligned} \qquad (2.3.211)$$

*with initial condition*

$$\tilde{z}^s_{j,k}(M) = \hat{u}_{j,k}(M|M) \qquad (2.3.212)$$

**Proof**

By multiplying eq.(2.3.117) on the left by $(v^j_k(m))^T$ we get

$$\begin{aligned} z^s_{j,k}(m) &= (v^j_k(m))^T x^s(m) \\ &= \hat{z}_{j,k}(m|m) \\ &+ (v^j_k(m))^T \mathcal{P}(m|m) I_2 \mathcal{A}^T(m) H_m \mathcal{P}^{-1}(m+1|m) I'_2 \\ &\times [x^s(m+1) - \hat{x}(m+1|m)] \qquad (2.3.213) \\ I_2 &= \sum_{l=L}^{M} \sum_n v^l_n(m)(v^l_n(m))^T + \sum_n \overline{v}^L_n(m)(\overline{v}^L_n(m))^T \qquad (2.3.214) \\ I'_2 &= \sum_{l=L}^{M} \sum_n v^l_n(m+1)(v^l_n(m+1))^T + \sum_n \overline{v}^L_n(m+1)(\overline{v}^L_n(m+1))^T \\ & \qquad (2.3.215) \end{aligned}$$

Note that from

$$(v^i_k(m))^T \mathcal{P}(m|m) v^j_l(m) = \overline{P}_{j,k}(m|m)\delta_{i-j,k-l} \qquad (2.3.216)$$

we get that

$$(v^j_k(m))^T \mathcal{P}(m|m) I_2 = \overline{P}_{j,k}(m|m) v^j_k(m))^T \qquad (2.3.217)$$

From Corollary 2.2.3 and Lemma 2.3.1 we get that

$$\begin{aligned} (v^j_k(m))^T \mathcal{P}(m|m) I_2 \mathcal{A}^T(m) H_m \mathcal{P}^{-1}(m+1|m) I'_2 &= \overline{P}_{j,k}(m|m) A^T(m+1) v^j_k(m+1))^T \\ &\times \mathcal{P}^{-1}(m+1|m) I'_2 \end{aligned}$$
$$(2.3.218)$$

From

$$(v^i_k(m+1))^T \mathcal{P}^{-1}(m+1|m) v^j_l(m+1) = \overline{P}^{-1}_{j,k}(m+1|m)\delta_{i-j,k-l} \qquad (2.3.219)$$

we get that

$$\begin{aligned}(v_k^j(m))^T\mathcal{P}(m|m)I_2\mathcal{A}^T(m)H_m\mathcal{P}^{-1}(m+1|m)I_2' &= \overline{P}_{j,k}(m|m)A^T(m+1) \\ &\times \overline{P}_{j,k}^{-1}(m+1|m)(v_k^j(m+1))^T\end{aligned} \tag{2.3.220}$$

Substituting eq.(2.3.220) into eq.(2.3.213) we get

$$z_{j,k}^s(m) = \hat{z}_{j,k}(m|m) + \overline{P}_{j,k}(m|m)A^T(m+1)\overline{P}_{j,k}^{-1}(m+1|m)[z_{j,k}^s(m+1) - \hat{z}_{j,k}(m+1|m)] \tag{2.3.221}$$

thus, proving eq.(2.3.209). Eq.(2.3.211) is proved in a similar manner. □

**Definition 2.3.7** *The* **diagonalized smoothed error** *covariances, $P_{j,k}^s(m)$ and $\tilde{P}_{j,k}^s(m)$, are defined as follows.*

$$P_{j,k}^s(m) \triangleq (v_k^j(m))^T\mathcal{P}^s(m)v_k^j(m) \tag{2.3.222}$$

$$\tilde{P}_{L,k}^s(m) \triangleq (\overline{v}_k^L(m))^T\mathcal{P}^s(m)\overline{v}_k^L(m) \tag{2.3.223}$$

**Lemma 2.3.6** *The diagonalized smoothed error covariances, $P_{j,k}^s(m)$ and $\tilde{P}_{j,k}^s(m)$, evolve according to the following equations.*

*For $j = L, L+1, ...M-1$ and $k \in \mathcal{Z}$*

$$P_{j,k}^s(m) = \overline{P}_{j,k}(m|m) + \overline{E}_{j,k}(m)\left[P_{j,k}^s(m+1) - \overline{P}_{j,k}(m+1|m)\right]\overline{E}_{j,k}^T(m) \tag{2.3.224}$$

$$\overline{E}_{j,k}(m) = \overline{P}_{j,k}(m|m)A^T(m+1)\overline{P}_{j,k}^{-1}(m+1|m) \tag{2.3.225}$$

$$m = M-1, M-2, ...L+1, L$$

*with initial condition*

$$P_{j,k}^s(M) = \overline{P}_{j,k}(M|M) \tag{2.3.226}$$

*For $k \in \mathcal{Z}$*

$$\tilde{P}_{L,k}^s(m) = \tilde{P}_{j,k}(m|m) + \overline{E}'_{j,k}(m)\left[\tilde{P}_{j,k}^s(m+1) - \tilde{P}_{j,k}(m+1|m)\right](\overline{E}'_{j,k})^T(m) \tag{2.3.227}$$

$$\overline{E}'_{j,k}(m) = \tilde{P}_{j,k}(m|m)A^T(m+1)\tilde{P}_{j,k}^{-1}(m+1|m) \tag{2.3.228}$$

$$m = M-1, M-2, ...L+1, L$$

*with initial condition*

$$\tilde{P}^s_{j,k}(M) = \tilde{P}_{j,k}(M|M) \tag{2.3.229}$$

*Furthermore, we have that*

$$\begin{aligned}
(\overline{v}^L_k(m))^T \mathcal{P}^s(m)\overline{v}^L_l(m) &= \tilde{P}^s_{L,k}(m)\delta_{k-l} &\quad (2.3.230)\\
(v^i_k(m))^T \mathcal{P}^s(m)\overline{v}^j_l(m) &= P^s_{j,k}(m)\delta_{i-j,k-l} &\quad (2.3.231)
\end{aligned}$$

**Proof**

We show eq.(2.3.224) by induction on $m$. Assume that

$$(v^i_k(m+1))^T \mathcal{P}^s(m+1) v^j_l(m+1) = P^s_{j,k}(m+1)\delta_{i-j,k-l} \tag{2.3.232}$$

Multiplying eq.(2.3.118) on the left by $(v^i_k(m))^T$ and on the right by $v^j_l(m)$, we get

$$\begin{aligned}
(v^i_k(m))^T \mathcal{P}^s(m) v^j_l(m) &= \overline{P}_{j,k}(m|m)\delta_{i-j,k-l}\\
&+ (v^i_k(m))^T E(m) I'_2 \left[\mathcal{P}^s(m+1) - \mathcal{P}(m+1|m)\right] I'_2 E^*(m) v^j_l(m)
\end{aligned} \tag{2.3.233}$$

where $I'_2$ is defined in eq.(2.3.215). From Corollary 2.2.3, Lemma 2.3.1, and the fact that

$$\begin{aligned}
(v^i_k(m))^T \mathcal{P}(m|m) v^j_l(m) &= \overline{P}_{i,k}(m|m)\delta_{i-j,k-l} &\quad (2.3.234)\\
(v^i_k(m))^T \mathcal{P}^{-1}(m+1|m) v^j_l(m) &= \overline{P}^{-1}_{i,k}(m+1|m)\delta_{i-j,k-l} &\quad (2.3.235)
\end{aligned}$$

we get

$$\begin{aligned}
(v^i_k(m))^T E(m) I'_2 &= (v^i_k(m))^T \mathcal{P}(m|m) I_2 \mathcal{A}^*(m+1) H_m \mathcal{P}^{-1}(m+1|m) I'_2\\
&= \overline{P}_{i,k}(m|m) A^T(m+1) \overline{P}^{-1}_{i,k}(m+1|m)(v^i_k(m+1))^T
\end{aligned} \tag{2.3.236}$$

Substituting eq.(2.3.236) into eq.(2.3.233) and using the fact that

$$(v^i_k(m+1))^T \mathcal{P}(m+1|m) v^j_l(m+1) = \overline{P}_{j,k}(m+1|m)\delta_{i-j,k-l} \tag{2.3.237}$$

we get

$$\begin{aligned}(v_k^i(m))^T \mathcal{P}^s(m) v_l^j(m) &= \overline{P}_{j,k}(m|m)\delta_{i-j,k-l} \\ &+ \overline{P}_{j,k}(m|m)A^T(m+1)\overline{P}_{j,k}^{-1}(m+1|m) \\ &\times \left[P_{j,k}^s(m+1) - \overline{P}_{j,k}(m+1|m)\right]\delta_{i-j,k-l} \\ &\times \overline{P}_{j,k}^{-1}(m+1|m)A(m+1)\overline{P}_{j,k}(m|m) \\ &= P_{j,k}^s(m)\delta_{i-j,k-l} \qquad (2.3.238)\end{aligned}$$

By substituting eq.(2.3.236) into eq.(2.3.233) and by using the fact that

$$\begin{aligned}(v_k^i(M))^T \mathcal{P}^s(M) v_l^j(m+1) &= (v_k^i(M))^T \mathcal{P}(M|M) v_l^j(m+1) \\ &= \overline{P}_{j,k}(M|M)\delta_{i-j,k-l} \qquad (2.3.239)\end{aligned}$$

we show eq.(2.3.238) is true for $m = M+1$. Thus, we have verified eq.'s(2.3.224,2.3.231). Eq.'s(2.3.227,2.3.230) are derived in a similar manner. □

The previous results can be grouped together to form a smoothing algorithm consisting of a collection of 1D Rauch-Tung-Striebel smoothing algorithms, each of which can be performed in parallel. In particular we have the following algorithm.

**Algorithm 2.3.1** *Consider the smoothing problem for a lattice defined over a finite number of scales, labeled from coarse to fine as $m = L, L+1, \ldots M$. The following set of equations describes the solution to the smoothing problem, transformed onto the space spanned by the eigenvectors of $R_{xx}(m)$, in terms of independent standard Rauch-Tung Striebel smoothing algorithms.*

*DOWN SWEEP:*

*For $j = L, L+1, \ldots M-2$ and $k \in \mathcal{Z}$:*

$$\hat{z}_{j,k}(m|m-1) = A(m)\hat{z}_{j,k}(m-1|m-1) \qquad (2.3.240)$$

$$\overline{P}_{j,k}(m|m-1) = A(m)\overline{P}_{j,k}(m-1|m-1)A^T(m) + B(m)Q(m)B^T(m) \qquad (2.3.241)$$

$$m = j+2, j+3, \ldots M$$

*with the initial conditions for $j = L, L+1, \ldots M-1$ and $k \in \mathcal{Z}$*

$$\hat{z}_{j,k}(j+1|j) = 0 \qquad (2.3.242)$$

$$\overline{P}_{j,k}(j+1|j) = B(j+1)Q(j+1)B^T(j+1) \qquad (2.3.243)$$

*For $j = L, L+1, ...M-1$ and $k \in \mathcal{Z}$:*

$$\begin{aligned}
\hat{z}_{j,k}(m|m) &= \hat{z}_{j,k}(m|m-1) + \overline{K}_{j,k}(m)(\overline{y}_{j,k}(m) - C(m)\hat{z}_{j,k}(m|m-1)) && (2.3.244)\\
\overline{P}_{j,k}^{-1}(m|m) &= \overline{P}_{j,k}^{-1}(m|m-1) + C^T(m)R^{-1}(m)C(m) && (2.3.245)\\
m &= j+1, j+2, ...M\\
\overline{K}_{j,k}(m) &\triangleq (v_k^j(m))^T\mathcal{K}(m)\mathcal{V}_k^j(m) && (2.3.246)
\end{aligned}$$

*For $k \in \mathcal{Z}$ we have*

$$\begin{aligned}
\hat{u}_{L,k}(m|m-1) &= A(m)\hat{u}_{L,k}(m-1|m-1) && (2.3.247)\\
\tilde{P}_{L,k}(m|m-1) &= A(m)\tilde{P}_{L,k}(m-1|m-1)A^T(m) + B(m)Q(m)B^T(m) && (2.3.248)\\
m &= L+1, L+2, ...M
\end{aligned}$$

*with the initial conditions*

$$\begin{aligned}
\hat{u}_{L,k}(L|L-1) &= 0 && (2.3.249)\\
\tilde{P}_{L,k}(L|L-1) &= P_x(L) && (2.3.250)
\end{aligned}$$

*For $k \in \mathcal{Z}$ we have*

$$\begin{aligned}
\hat{u}_{L,k}(m|m) &= \hat{u}_{L,k}(m|m-1) + \tilde{K}_{L,k}(m)(\overline{y}_{L,k}(m) - C(m)\hat{u}_{L,k}(m|m-1)) && (2.3.251)\\
\tilde{P}_{L,k}^{-1}(m|m) &= \tilde{P}_{L,k}^{-1}(m|m-1) + C^T(m)R^{-1}(m)C(m) && (2.3.252)\\
m &= L, L+1, ...M\\
\tilde{K}_{L,k}(m) &\triangleq (\overline{v}_k^j(m))^T\mathcal{K}(m)\overline{\mathcal{V}}_k^j(m) && (2.3.253)
\end{aligned}$$

*UP SWEEP:*
*For $j = L, L+1, ...M-1$ and $k \in \mathcal{Z}$*

$$z_{j,k}^s(m) = \hat{z}_{j,k}(m|m) + \overline{P}_{j,k}(m|m)A^T(m+1)\overline{P}_{j,k}^{-1}(m+1|m)[z_{j,k}^s(m+1) - \hat{z}_{j,k}(m+1|m)] \quad (2.3.254)$$

$$\begin{aligned} P^s_{j,k}(m) &= \overline{P}_{j,k}(m|m) + \overline{E}_{j,k}(m)\left[P^s_{j,k}(m+1) - \overline{P}_{j,k}(m+1|m)\right]\overline{E}^T_{j,k}(m) && (2.3.255)\\ \overline{E}_{j,k}(m) &= \overline{P}_{j,k}(m|m)A^T(m+1)\overline{P}^{-1}_{j,k}(m+1|m) && (2.3.256)\\ m &= M-1, M-2, ...j+2, j+1 && (2.3.257) \end{aligned}$$

*with initial condition*

$$\begin{aligned} z^s_{j,k}(M) &= \hat{z}_{j,k}(M|M) && (2.3.258)\\ P^s_{j,k}(M) &= \overline{P}_{j,k}(M|M) && (2.3.259) \end{aligned}$$

*For $k \in \mathcal{Z}$*

$$\begin{aligned} \tilde{z}^s_{L,k}(m) &= \hat{u}_{j,k}(m|m) + \tilde{P}_{j,k}(m|m)A^T(m+1)\tilde{P}^{-1}_{j,k}(m+1|m)[\tilde{z}^s_{j,k}(m+1) - \hat{u}_{j,k}(m+1|m)] \\ & && (2.3.260)\\ \tilde{P}^s_{L,k}(m) &= \tilde{P}_{j,k}(m|m) + \overline{E}'_{j,k}(m)\left[\tilde{P}^s_{j,k}(m+1) - \tilde{P}_{j,k}(m+1|m)\right](\overline{E}'_{j,k})^T(m) && (2.3.261)\\ \overline{E}'_{j,k}(m) &= \tilde{P}_{j,k}(m|m)A^T(m+1)\tilde{P}^{-1}_{j,k}(m+1|m) && (2.3.262)\\ m &= M-1, M-2, ...L+1, L && (2.3.263) \end{aligned}$$

*with initial condition*

$$\begin{aligned} \tilde{z}^s_{L,k}(M) &= \hat{u}_{L,k}(M|M) && (2.3.264)\\ \tilde{P}^s_{L,k}(M) &= \tilde{P}_{L,k}(M|M) && (2.3.265) \end{aligned}$$

The structure of this algorithm is illustrated in Figure 2.3.1. Note that our algorithm is highly efficient in that we've transformed the problem of smoothing infinite dimensional vectors to one of smoothing in parallel a set of *finite* dimensional vectors. Also, the smoothing procedure takes place in *scale* rather than in time, and for finite data of length $N$ this interval is at most of order $logN$. In particular, the number of scale components at each scale increases with finer scales. Since only data at scales at or finer than the scale of a particular component provide information about that component, this leads to smoothers of different length in scale, with their length indexed by scale component. Let us analyze the complexity of our overall algorithm for smoothing our lattice processes. We first transform our data using the wavelet transform which is fast: $O(lN)$ where $N$ is the number of points at the finest scale

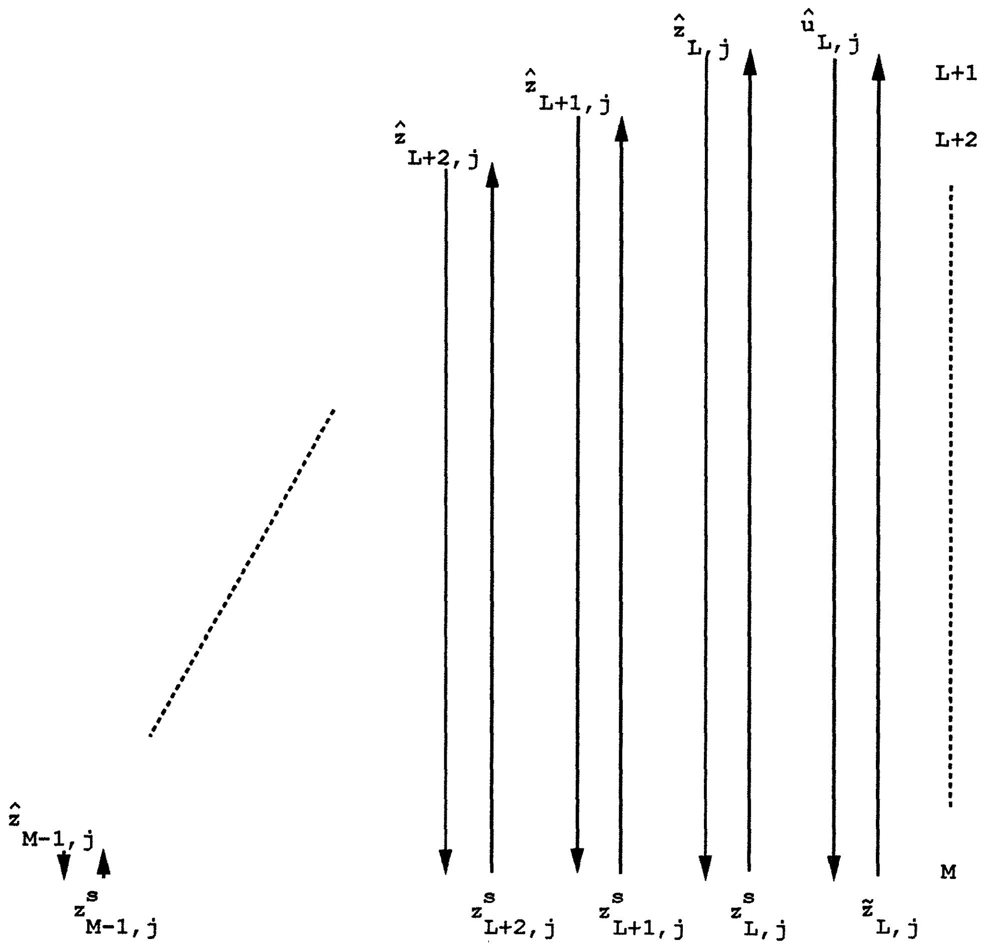

Figure 2.3.1: Parallel 1D Smoothing - Down-up

and $l$ is the length of the QMF filter. We then perform in parallel our 1D smoothers. Note that as illustrated in Figure 2.3.1 the different scale components are smoothed using RTS smoothers over different length intervals, where the typically the length of the interval is at most $logN$. Even if these smoothers are computed serially the total computation is $O(lN)$. After performing the parallel 1D smoothers on these transformed variables an additional inverse transformation is required, which is performed again using the inverse wavelet transform (the synthesis equations). Thus, the overall procedure is of complexity $O(lN)$. We show in detail later how the wavelet transform is adapted to the case of finite length sequences. We also point out that our algorithm generalizes Wornell's smoothing algorithm[56] to allow for multiscale measurements and to allow for higher-order models.

As a final note concerning lattice models and our lattice-based smoother, our transform method of parallelizing the smoothing problem requires the matrix $C(m)$ in eq.(2.3.102) to have constants along the diagonal for all $m$, i.e. that the same measurements are made at all points at any particular scale. The case of missing data at a given scale is an example in which this structure is violated. This is relevant to situations in which one might want to use coarse data to interpolate sparsely distributed fine data. This problem is easily handled by our framework using models based on homogeneous trees, which is the topic of Chapter 3. We can also use an iterative algorithm to solve this problem in the context of our lattice models; this is also discussed in Chapter 3.

## 2.4 Up-down Version of the Algorithm

In this section we describe a version of our transform approach to smoothing lattice processes which performs the Rauch-Tung-Striebel algorithm starting with a fine to coarse or "up" sweep followed by a coarse to fine or "down" sweep. In principle there is no reason why the filtering step must necessarily be defined from coarse to fine as in the previous section. However, since our state model is defined in the direction from coarse to fine, having the filtering step be defined in the same direction simplified the derivation of the transform approach in the last section. Given that we've already diagonalized the smoothing problem in this context, in order to derive an up-down version of the algorithm we can do this entirely in the transform domain using the smoothing equations derived in the previous section.

In particular note that the filtering step for each of the components in Algorithm 2.3.1 can be interpreted as a Kalman filter for the following state model and observation equation.

$$\begin{aligned} z_{j,k}(m+1) &= A(m)z_{j,k}(m) + B(m)w_{j,k}(m) && (2.4.1)\\ E[w_{j,k}(m)w_{j,k}^T(m)] &= Q(m) && (2.4.2)\\ \overline{y}_{j,k}(m) &= C(m)z_{j,k}(m) + r_{j,k}(m) && (2.4.3)\\ E[r_{j,k}(m)r_{j,k}^T(m)] &= R(m) && (2.4.4)\end{aligned}$$

If we assume $A(m)$ is invertible for all $m$, we can use results on backward Markovian models[53] to write down the corresponding backwards model for eq.(2.4.1).

$$\begin{aligned} z_{j,k}(m-1) &= D_{j,k}(m)z_{j,k}(m) - A^{-1}(m)B(m)\tilde{w}_{j,k}(m) && (2.4.5)\\ D_{j,k}(m) &= A^{-1}(m)\{I - B(m)B^T(m)(P_{j,k}^x(m))^{-1}\} && (2.4.6)\\ E[\tilde{w}_{j,k}(m)\tilde{w}_{j,k}^T(m)] &= I - B^T(m)(P_{j,k}^x(m))^{-1}B(m) && (2.4.7)\\ &\triangleq \tilde{Q}(m) \end{aligned}$$

where

$$P_{j,k}^x(m) \triangleq (v_k^j(m))^T R_{xx}(m) v_k^j(m) \qquad (2.4.8)$$

Similarly, we have

$$u_{L,k}(m-1) = \overline{D}_{L,k}(m)u_{L,k}(m) - A^{-1}(m)B(m)\overline{w}_{L,k}(m) \qquad (2.4.9)$$

$$\begin{aligned}
\overline{D}_{L,k}(m) &= A^{-1}(m)\{I - B(m)B^T(m)(\overline{P}^x_{L,k}(m))^{-1}\} && (2.4.10)\\
E[\overline{w}_{L,k}(m)\overline{w}^T_{L,k}(m)] &= I - B^T(m)(\overline{P}^x_{L,k}(m))^{-1}B(m) && (2.4.11)\\
&\triangleq \overline{Q}(m)
\end{aligned}$$

where

$$\overline{P}^x_{L,k}(m) \triangleq (\overline{v}^L_k(m))^T R_{xx}(m)\overline{v}^L_k(m) \qquad (2.4.12)$$

Now, if we use our newly defined backward models for our transformed variables and apply the standard Rauch-Tung-Striebel algorithm to each of these, we can transform the algorithm in the previous section into the following up-down algorithm.

**Algorithm 2.4.1** *Consider the smoothing problem for a lattice defined over a finite number of scales, labeled from coarse to fine as $m = L, L+1, \ldots M$. The following set of equations describes the solution to the smoothing problem, transformed onto the space spanned by the eigenvectors of $R_{xx}(m)$, in terms of independent standard Rauch-Tung Striebel smoothing algorithms.*

*UP SWEEP:*

*For $j = L, L+1, \ldots M-2$ and $k \in \mathcal{Z}$:*

$$\begin{aligned}
\hat{z}_{j,k}(m|m+1) &= D(m+1)\hat{z}_{j,k}(m+1|m+1) && (2.4.13)\\
\overline{P}_{j,k}(m|m+1) &= D(m+1)\overline{P}_{j,k}(m+1|m+1)D^T(m+1)\\
&+ B(m+1)\tilde{Q}(m+1)B^T(m+1) && (2.4.14)\\
m &= M-1, M-2, \ldots j+2, j+1
\end{aligned}$$

*with the initial conditions for $j = L, L+1, \ldots M-1$ and $k \in \mathcal{Z}$*

$$\begin{aligned}
\hat{z}_{j,k}(M|M+1) &= 0 && (2.4.15)\\
\overline{P}_{j,k}(M|M+1) &= P^x_{j,k}(M) && (2.4.16)
\end{aligned}$$

*For $j = L, L+1, \ldots M-1$ and $k \in \mathcal{Z}$:*

$$\begin{aligned}
\hat{z}_{j,k}(m|m) &= \hat{z}_{j,k}(m|m+1) + K_{j,k}(m)(\overline{y}_{j,k}(m) - C(m)\hat{z}_{j,k}(m|m+1))\\
& && (2.4.17)\\
\overline{P}^{-1}_{j,k}(m|m) &= \overline{P}^{-1}_{j,k}(m|m+1) + C^T(m)R^{-1}(m)C(m) && (2.4.18)
\end{aligned}$$

$$
\begin{aligned}
m &= M, M-1, \ldots j+2, j+1 \\
K_{j,k}(m) &\triangleq \overline{P}_{j,k}(m|m+1)C^T(m)V^{-1}(m) && (2.4.19) \\
V(m) &= C(m)\overline{P}_{j,k}(m|m+1)C^T(m) + R(m) && (2.4.20)
\end{aligned}
$$

*For $k \in \mathcal{Z}$ we have*

$$
\begin{aligned}
\hat{u}_{L,k}(m|m+1) &= \overline{D}(m+1)\hat{u}_{L,k}(m+1|m+1) && (2.4.21) \\
\tilde{P}_{L,k}(m|m+1) &= \overline{D}(m+1)\tilde{P}_{L,k}(m+1|m+1)\overline{D}^T(m+1) \\
&+ B(m+1)Q(m+1)B^T(m+1) && (2.4.22) \\
m &= M-1, M-2, \ldots L+2, L+1
\end{aligned}
$$

*with the initial conditions*

$$
\begin{aligned}
\hat{u}_{L,k}(M|M+1) &= 0 && (2.4.23) \\
\tilde{P}_{L,k}(M|M+1) &= \overline{P}^x_{j,k}(M) && (2.4.24)
\end{aligned}
$$

*For $k \in \mathcal{Z}$ we have*

$$
\begin{aligned}
\hat{u}_{L,k}(m|m) &= \hat{u}_{L,k}(m|m+1) + \tilde{K}_{L,k}(m)(\overline{y}_{L,k}(m) - C(m)\hat{u}_{L,k}(m|m+1)) \\
& && (2.4.25) \\
\tilde{P}^{-1}_{L,k}(m|m) &= \tilde{P}^{-1}_{L,k}(m|m+1) + C^T(m)R^{-1}(m)C(m) && (2.4.26) \\
m &= M, M-1, \ldots L+2, L+1 \\
\tilde{K}_{L,k}(m) &\triangleq \tilde{P}_{j,k}(m|m+1)C^T(m)\tilde{V}^{-1}(m) && (2.4.27) \\
\tilde{V}(m) &= C(m)\tilde{P}_{j,k}(m|m+1)C^T(m) + R(m) && (2.4.28)
\end{aligned}
$$

*DOWN SWEEP:*

*For $j = L, L+1, \ldots M-1$ and $k \in \mathcal{Z}$*

$$
\begin{aligned}
z^s_{j,k}(m+1) &= \hat{z}_{j,k}(m+1|m+1) \\
&+ \overline{P}_{j,k}(m+1|m+1)D^T(m)\overline{P}^{-1}_{j,k}(m|m+1)[z^s_{j,k}(m) - \hat{z}_{j,k}(m|m+1)] \\
& && (2.4.29) \\
P^s_{j,k}(m+1) &= \overline{P}_{j,k}(m+1|m+1) + \overline{E}_{j,k}(m+1)\left[P^s_{j,k}(m) - \overline{P}_{j,k}(m|m+1)\right]\overline{E}^T_{j,k}(m+1)
\end{aligned}
$$

$$(2.4.30)$$

$$\overline{E}_{j,k}(m+1) = \overline{P}_{j,k}(m+1|m+1)D^T(m)\overline{P}_{j,k}^{-1}(m|m+1) \quad (2.4.31)$$

$$m = j+1, j+2, \ldots M-2, M-1 \quad (2.4.32)$$

*with initial condition*

$$z_{j,k}^s(j+1) = \hat{z}_{j,k}(j+1|j+1) \quad (2.4.33)$$

$$P_{j,k}^s(j+1) = \overline{P}_{j,k}(j+1|j+1) \quad (2.4.34)$$

*For k ∈ $\mathcal{Z}$*

$$\begin{aligned} \tilde{z}_{L,k}^s(m+1) &= \hat{u}_{j,k}(m+1|m+1) \\ &+ \tilde{P}_{j,k}(m+1|m+1)\overline{D}^T(m)\tilde{P}_{j,k}^{-1}(m|m+1)[\tilde{z}_{j,k}^s(m) - \hat{u}_{j,k}(m|m+1)] \end{aligned} \quad (2.4.35)$$

$$\tilde{P}_{L,k}^s(m+1) = \tilde{P}_{j,k}(m+1|m+1) + \overline{E}'_{j,k}(m+1)\left[\tilde{P}_{j,k}^s(m) - \tilde{P}_{j,k}(m|m+1)\right](\overline{E}'_{j,k})^T(m+1) \quad (2.4.36)$$

$$\overline{E}'_{j,k}(m+1) = \tilde{P}_{j,k}(m+1|m+1)\overline{D}^T(m)\tilde{P}_{j,k}^{-1}(m|m+1) \quad (2.4.37)$$

$$m = L+1, L+2, \ldots M-2, M-1 \quad (2.4.38)$$

*with initial condition*

$$\tilde{z}_{L,k}^s(L+1) = \hat{u}_{L,k}(L+1|L+1) \quad (2.4.39)$$

$$\tilde{P}_{L,k}^s(L+1) = \tilde{P}_{L,k}(L+1|L+1) \quad (2.4.40)$$

Note that as in the case of the down-up version of the algorithm, each transform component is smoothed over an interval whose length is indexed by scale. In this version the filtering step of each transform component begins at the finest scale and filters upward to the scale of the component, followed by a correction sweep back down to the finest scale.

## 2.5 Finite Length Wavelet Transforms

In this section we discuss the problem of adapting the wavelet transform, thus far defined only for infinite sequences, to the case of finite-length sequences, i.e. producing a transform that maps finite-length sequences into finite-length sequences. We characterize the problem in such a way as to show a variety of possibilities for viewing the finite-length wavelet transform as an orthonormal matrix transformation. Note that both the analysis and synthesis equations, eq.'s(1.2.15,1.2.16,1.2.22), for computing the wavelet and scaling coefficients are defined as operations on infinite length sequences. Adapting these equations to the case of finite length sequences *while preserving both the orthogonality and the invertibility* of the transformation proves to be non-trivial for the following reason. Take a 10-point sequence (i.e. a sequence whose non-zero support is 10 points), $x(n)$, and consider performing its wavelet transform using a QMF filter, $h(n)$, of length four. To compute the scaling coefficients at the next coarsest scale we apply eq.(1.2.15) to $x(n)$, resulting in a scaling coefficient sequence, $c(n)$, which is of length 6 (the linear convolution results in a 13-point sequence, while the downsampling by a factor of two reduces this to a 6-point sequence). Similarly, by applying eq.(1.2.16) to $x(n)$ we get a wavelet coefficient sequence, $d(n)$, which is also of length 6. Thus, the overall transformation from the nonzero portion of $\{x(n)\}$ to the nonzero portions of $\{c(n), d(n)\}$ in this case is a map from $\mathcal{R}^{10}$ to $\mathcal{R}^{12}$, which makes it impossible for it even to be invertible, let alone representable by an orthonormal matrix. This example is illustrated Figure 2.5.2, where $x(n)$ is defined as indicated on the first level of a truncated lattice and $\{c(n), d(n)\}$ are mapped into the second level where the lattice branches are illustrated for the case where the operators $H_i, G_i$ correspond to a QMF filter, $h(n)$, of length four and only branches connecting to points in the nonzero portion of $x(n)$ are shown.

Thus, we can already see the fundamental problem in trying to develop an orthonormal matrix transformation based on the wavelet transform. At each scale we must have a well-defined orthonormal transformation from our approximation at that scale into its scaling coefficients and its wavelet coefficients at the next coarsest scale. To see how this can be done it is sufficient to focus on our previous example involv-

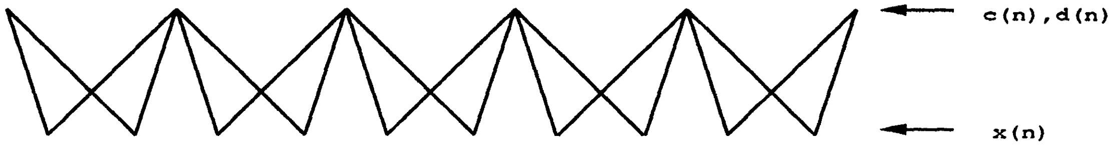

Figure 2.5.2: Transformation of a 10-pt. Sequence $x(n)$ into its 6-pt. Scaling Coefficients $c(n)$ and its 6-pt. Wavelet Coefficients $d(n)$

ing the map from $x(n)$ into $\{c(n), d(n)\}$. We can write the transformation in our example explicitly as follows. We denote our 4-tap QMF filter, $h$, as a row vector $[\ h_0 \quad h_1 \quad h_2 \quad h_3\ ]$. Similarly, our filter, $g$, is denoted as $[\ g_0 \quad g_1 \quad g_2 \quad g_3\ ]$ where from eq.(1.2.12) we know that

$$[\ g_0 \quad g_1 \quad g_2 \quad g_3\ ] = [\ h_3 \quad -h_2 \quad h_1 \quad -h_0\ ] \tag{2.5.41}$$

If we think of the non-zero portion of our sequence $x(n)$ as a vector, $x$, in $\mathcal{R}^{10}$ and the non-zero portions of $c(n), d(n)$ as vectors, $c$ and $d$, in $\mathcal{R}^6$, our maps $x(n) \mapsto c(n)$ and $x(n) \mapsto d(n)$ can be thought of as the following $6 \times 10$ matrices.

$$H \triangleq \begin{bmatrix} h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 \end{bmatrix} \tag{2.5.42}$$

$$G \triangleq \begin{bmatrix} g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{bmatrix} \qquad (2.5.43)$$

where

$$c = Hx \qquad (2.5.44)$$

$$d = Gx \qquad (2.5.45)$$

Note that $c$ and $d$ are *precisely* the non-zero portions of the sequences one obtains by applying the operators $H_i, G_i$ to $x(n)$. Thus, we can in fact reconstruct $x(n)$ from $c, d$ using our synthesis equation, eq.(1.2.24). In matrix notation

$$x = H^T c + G^T d \qquad (2.5.46)$$

If we denote our overall map $x \mapsto c, d$ as the $12 \times 10$ matrix

$$U \triangleq \begin{bmatrix} H \\ G \end{bmatrix} \qquad (2.5.47)$$

then eq.(2.5.46) says that $U^T U = I$. Note, however, that it is *not* the case that $UU^T = I$, since $U$ is not even square. Another way of seeing this is observing, using our QMF properties eq.'s(1.2.3,1.2.9,1.2.10), that

$$HH^T \neq I \qquad (2.5.48)$$

$$GG^T \neq I \qquad (2.5.49)$$

$$GH^T \neq 0 \qquad (2.5.50)$$

The failure of these conditions to hold is due primarily to the first and last rows of $H$ and $G$. In Figure 2.5.2 these correspond to the averaging performed at the edges of both ends of the lattice. Note that the rows of $H$ are mutually orthogonal and the rows of $G$ are mutually orthogonal. The reason for eq.'s(2.5.48,2.5.49) is simply the fact that the edge rows of $H$ and $G$ are not normalized so that their inner products equal one. The reason for eq.(2.5.50) is the fact that the edge rows of $G$ are not orthogonal to the edge rows of $H$.

If we want our overall transformation, $U$, to be orthonormal, we must somehow eliminate two of its rows. Note that if we eliminate the first and last rows of the matrix $H$ we get

$$\tilde{H} \triangleq \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \end{bmatrix} \qquad (2.5.51)$$

The matrix $[\tilde{H}^T G^T]^T$ is now square and it is sufficient to check the following conditions for its orthonormality.

$$\tilde{H}H^T = I \qquad (2.5.52)$$
$$GG^T = I \qquad (2.5.53)$$
$$G\tilde{H}^T = 0 \qquad (2.5.54)$$

Eq.'s(2.5.52,2.5.54) follow from our QMF properties eq.'s(1.2.3,1.2.9). As noted before, however, eq.(2.5.53) does not quite hold due to the fact that the the first and last rows of $G$ are not properly normalized. Examining $G$ in detail and using the QMF property in eq.(1.2.10) we see that

$$GG^T = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & b \end{bmatrix} \qquad (2.5.55)$$

where

$$a = g_2^2 + g_3^2 \qquad (2.5.56)$$
$$b = g_0^2 + g_1^2 \qquad (2.5.57)$$

Thus, we can satisfy eq.(2.5.53) simply by normalizing the first and last rows of $G$ by $a$ and $b$, respectively. So, if we let

$$\tilde{G} \triangleq \begin{bmatrix} \tilde{g}_2 & \tilde{g}_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{g}_0 & \tilde{g}_1 \end{bmatrix} \tag{2.5.58}$$

where the $\tilde{g}_i$'s are properly normalized, then the following defines an orthonormal matrix in $\mathcal{R}^{10\times 10}$.

$$\tilde{U} \triangleq \begin{bmatrix} \tilde{H} \\ \tilde{G} \end{bmatrix} \tag{2.5.59}$$

Our transformation $\tilde{U}$ maps $x$ into scaling coefficients $c$ of length 4 while mapping $x$ into wavelet coefficients $d$ of length 6. This has the following interpretation. While $\tilde{U}$ maps the nonzero portion of $x(n)$ into the nonzero portion of its wavelet coefficients, $d(n)$, at the next coarsest scale, normalizing the coefficients at the edges, it maps the nonzero portion of $x(n)$ into the nonzero portion of its scaling coefficients, $c(n)$, while *zeroing* the two scaling coefficients at the edges. Note that if we perform our transformation recursively in scale, at scale each scale we end up with scaling coefficients which are zeroed at the edges, leaving us with fewer and fewer scaling coefficients as we go to coarser scales. If we take our example one step coarser in scale, i.e. we apply the same idea used to create $\tilde{U}$ on the scaling coefficients $c$, we end up mapping $c$ into one scaling coefficient and three wavelet coefficients at the next coarsest scale. The overall two scale decomposition results in scaling coefficients defined on the lattice in Figure 2.5.3. The resulting wavelet coefficients reside on the lattice in Figure 2.5.4, where the dotted lines represent averaging at the edges due to the normalized $\tilde{g}_i$'s. Thus, we have defined a way of adapting the wavelet transform to the case of finite-length sequences which has the interpretation of producing a set of scaling coefficients which are zeroed at the edges at each scale and producing a set of wavelet coefficients which are include additional edge points at each scale.

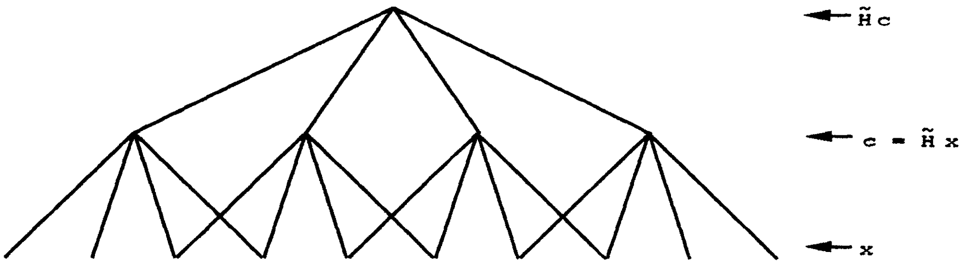

Figure 2.5.3: Lattice Representing Domain of Scaling Coefficients for 2-scale Decomposition Based on Zeroing Edge Scaling Coefficients

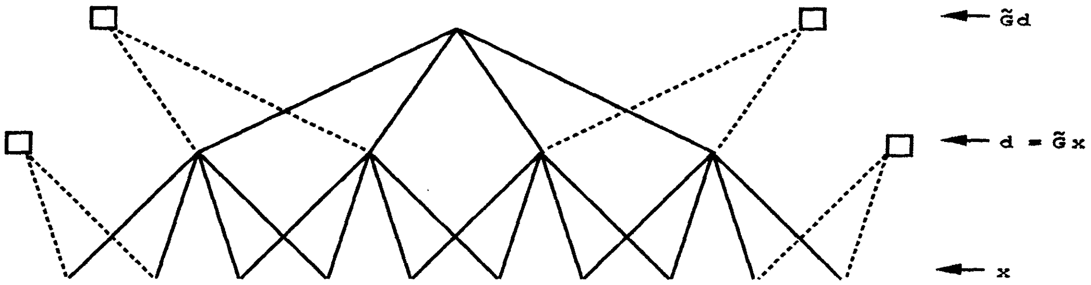

Figure 2.5.4: Lattice Representing Domain of the Wavelet Coefficients for 2-scale Decomposition Based on Zeroing Edge Scaling Coefficients

Note that the coefficients $d(n)$ and $c(n)$ play a symmetric role in our procedure for making our transformation $U$ orthonormal. In particular we could equally well have zeroed the edges of of our wavelet coefficients $d(n)$ rather than our scaling coefficients $c(n)$. This possibility leads us to defining

$$\hat{H} \triangleq \begin{bmatrix} \tilde{h}_2 & \tilde{h}_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{h}_0 & \tilde{h}_1 \end{bmatrix} \qquad (2.5.60)$$

$$\hat{G} \triangleq \begin{bmatrix} g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{bmatrix} \qquad (2.5.61)$$

where the $\tilde{h}_i$'s are properly normalized. In this case the matrix $[\hat{H}^T \hat{G}^T]^T$ is orthonormal. As in our previous case with $\tilde{U}$ this procedure can be performed at each scale. The resulting lattice structure, however, is different. Figure 2.5.5 illustrates domain of the scaling coefficients for a two scale decomposition while Figure 2.5.6 illustrates domain of the wavelet coefficients.

Finally, we describe a way of making our transformation $U$ orthonormal by using *cyclic* convolutions. The idea is to take a finite-length sequence, say of length $l$ where $l$ is a multiple of two, and to perform our coarsening and differencing operations using $l$-point cyclic convolutions. Thus, in our example convolving the 10-point $x(n)$ with $h(n)$ using a 10-point circular convolution followed by downsampling by a factor of two, yields a 5-point sequence $c(n)$ while convolving $x(n)$ with $g(n)$ using a 10-point circular convolution followed by downsampling by a factor of two, yields a 5-point sequence $d(n)$. The matrices representing these operations are defined as follows,

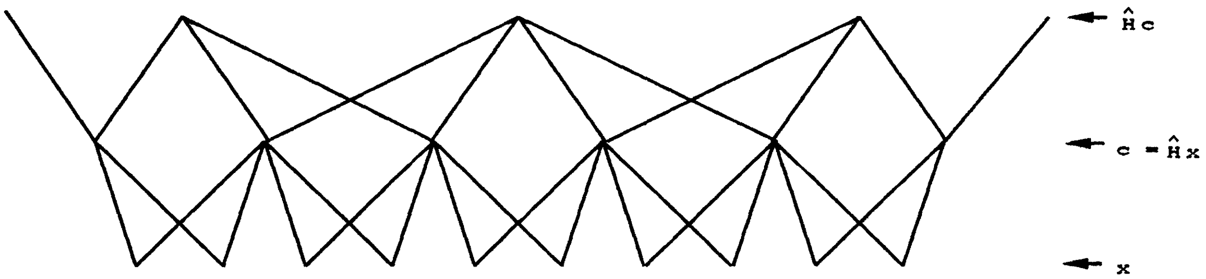

Figure 2.5.5: Lattice Representing Domain of the Scaling Coefficients for 2-scale Decomposition Based on Zeroing Edge Wavelet Coefficients

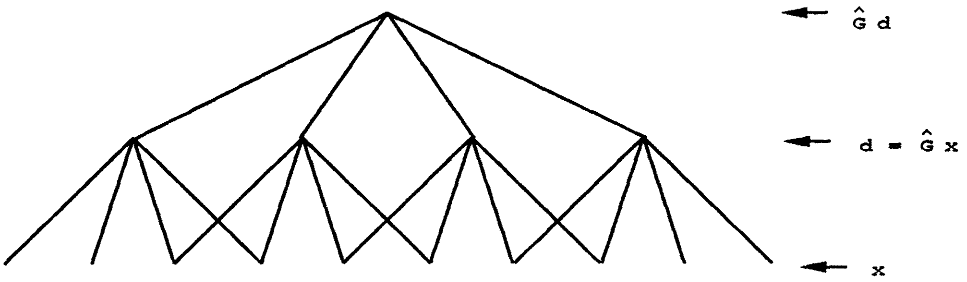

Figure 2.5.6: Lattice Representing Domain of the Wavelet Coefficients for 2-scale Decomposition Based on Zeroing Edge Wavelet Coefficients

where the top row of each matrix is a cyclic permutation of the other rows.

$$H_c \triangleq \begin{bmatrix} h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \end{bmatrix} \qquad (2.5.62)$$

$$G_c \triangleq \begin{bmatrix} g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{bmatrix} \qquad (2.5.63)$$

From our QMF properties we can show that the matrix $[H_c^T G_c^T]^T$ is indeed orthonormal.

We have described several ways of adapting the wavelet transform so that it performs an orthonormal matrix transformation on a finite-dimensional vector. Though we have focused on a simple 3-scale example, our approach can obviously be used for QMF filters of arbitrary order and for transforms decomposing signals into an arbitrary number of scales. For example the approach using cyclic convolutions works for sequences which are powers two, where each step of the transform amounts to taking cyclic convolutions of length equal to the number of points at that particular scale followed by downsampling by a factor of two. Note in principal one could iterate this procedure to a coarse scale consisting of a single point. Due to the fact, however, that for scales coarser than the length of the QMF filter the degree of wrap-around due to the cyclic convolution is so great, one typically iterates to the coarsest at which the number of points is greater than the length of the filter.

We now proceed to show how the results in the previous section apply to the case of the finite-length wavelet transform. To do this we describe in detail the general form of the approach based on zeroing out the scaling coefficients at the ends of each level of the lattice, i.e. the approach yielding $\tilde{U}$ in our earlier example. In this case, for QMF filters of length $P$ the scaling coefficients reside on a lattice where the number of points at the $i$th scale obeys the following recursion.

$$f(i+1) = 2f(i) + P - 2 \tag{2.5.64}$$

where at the coarsest scale, $L$, $f(L) = 1$. Figures 2.5.3 and 2.5.4 illustrate the domains of the scaling coefficients and the wavelet coefficients respectively for the coarsest 3 scales where the filter length is 4.

To see how our approach generalizes we consider the matrices $\tilde{H}_i$ and $\tilde{G}_i$, mapping points at the $i+1$st scale to the $i$th scale, where the filter length $P = 6$. They are given explicitly as follows where the solid horizontal lines demarcate the edge vectors associated with $\tilde{G}_i$.

$$\tilde{H}_i = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 & & \cdots & & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & & \cdots & 0 \\ \vdots & & & & & & & & & & & \vdots \\ 0 & \cdots & 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \end{bmatrix} \tag{2.5.65}$$

$$\tilde{G}_i = \left|\begin{array}{cccccccccccc} \hline \tilde{g}_4 & g_5 & 0 & 0 & 0 & 0 & 0 & 0 & & \cdots & & 0 \\ \tilde{g}_2 & \tilde{g}_3 & \tilde{g}_4 & \tilde{g}_5 & 0 & 0 & 0 & 0 & & \cdots & & 0 \\ \hline g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & 0 & 0 & & \cdots & & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & 0 & & \cdots & 0 \\ \vdots & & & & & & & & & & & \vdots \\ 0 & \cdots & 0 & 0 & g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & g_4 & g_5 \\ \hline 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{g}_0 & \tilde{g}_1 \\ \hline \end{array}\right| \tag{2.5.66}$$

Note that our matrices $\tilde{H}_i$ and $\tilde{G}_i$ satisfy all the algebraic relations satisfied by our operators $H_i$ and $G_i$ for the case of the infinite lattice. Thus, the results in the previous section apply with the only changes being purely dimensional ones. Also, block extensions of $\tilde{H}_i$ and $\tilde{G}_i$ can be defined exactly as in the case for $H_i$ and $G_i$. In particular let us define a state model in which we denote the vector of states at the

$m$th scale of our finite lattice to be $x(m)$, where each entry, $x_i(m)$, is a $d$ dimensional state vector. Our matrices $\tilde{H}_i$ and $\tilde{G}_i$ are now block matrices where each entry which was previously a scalar is now a $d \times d$ matrix equal to that scalar times $I_d$. Note that in the case of finite lattices the subscript $i$ is indeed necessary now as the dimensions of these matrices vary with scale.

We specify our model on the finite lattice as follows for $m = L, L+1, ...M-1$,

$$E[x(L)x(L)^T] = \mathcal{P}_x(L) \tag{2.5.67}$$

$$x(m+1) = \tilde{H}_m^T \mathcal{A}(m+1)x(m) + \mathcal{B}(m+1)w(m+1) \tag{2.5.68}$$

$$E[w(i)w(j)^T] = \mathcal{Q}(i)\delta_{i-j} \ , \ i = L+1, ...M \tag{2.5.69}$$

where

$$\mathcal{A}(m) \triangleq diag(..., A(m), ...A(m), ...) \tag{2.5.70}$$

$$\mathcal{B}(m) \triangleq diag(..., B(m), ...B(m), ...) \tag{2.5.71}$$

$$\mathcal{Q}(m) \triangleq diag(..., Q(m), ...Q(m), ...) \tag{2.5.72}$$

$$\mathcal{P}_x(L) \triangleq diag(..., P_x(L), ...P_x(L), ...) \tag{2.5.73}$$

and where $A(m), B(m), Q(m)$, and $P_x(L)$ are $d \times d$ matrices representing the system matrix, the process noise matrix, the process noise covariance matrix, and the initial state covariance matrix, respectively. The block-eigenstructure of our finite-lattice process is precisely as we derived in the previous section, where the infinite dimensional block operators are now replaced by our block versions of $\tilde{H}_i$ and $\tilde{G}_i$. In particular

$$\delta_i^j \triangleq [0_d, ..., 0_d, \underbrace{I_d}_{i\text{th}}, 0_d, ...0_d]^T \tag{2.5.74}$$

where the superscript $j$ is again used to denote that the vector (now in $\mathcal{R}^{f(j)\times d}$) corresponds to the $j$th scale of the lattice and where $I_d$ is the $d \times d$ identity matrix (and $0_d$ the $d \times d$ zero matrix). The block vectors $\overline{v}_i^L(m)$, $v_n^l(m)$ for $l = L, ...m-1$ and for $i = 0, 1, 2...f(L)-1$ and $n = 0, 1, 2...f(l)-1$ are block-eigenvectors of the correlation matrix of the process at scale $m$, $R_{xx}(m)$, where

$$\overline{v}_i^L(m) \triangleq (\prod_{j=L}^{m-1} \tilde{H}_j^T)\delta_j^L \tag{2.5.75}$$

and

$$v_n^l(m) \triangleq (\prod_{i=l+1}^{m-1} \tilde{H}_i^T)\tilde{G}_l^T \delta_n^l \tag{2.5.76}$$

As we did for the infinite case we can now transform the smoothing problem using a wavelet basis composed of the block vectors $\overline{v}_i^L(m)$ and $v_n^l(m)$. Our transformed variables are formed as in eq.'s(2.3.122-2.3.133), except that now we have a finite number of variables to estimate. In particular for each scale index, $j$, the translation index $k$ ranges from 0 to $f(j) - 1$. Our wavelet transform smoothing algorithm now applies, in either down-up or up-down form, as described in Sections 2.3 and 2.4.

# Chapter 3

# Multiscale Representations and Stochastic Processes on Trees

## 3.1 Introduction

In this chapter we develop a framework for performing estimation using models defined on trees as described in Section 1.4. Note that the dyadic tree is a special case of the lattice and that what we developed in the previous chapter applies to a subclass of the models considered in this chapter. There is, however, a great deal of additional structure in processes defined on trees which allows us to analyze the estimation problem much more deeply than is possible for general lattices. In particular we present a smoothing algorithm which is a generalization of the Rauch-Tung-Striebel algorithm for smoothing temporal processes. Our algorithm includes a generalized notion of Kalman filtering propagating up the tree based on filtering subtrees of data. This filtering procedure includes a *merge* step at each scale which has no counterpart in standard Kalman filtering theory. This leads us to the analysis of a new set of discrete Riccati equations indexed in scale. We then decompose the filter into two parts one of which corresponds to the Maximum-Likelihood filter, the other corresponding to propagating the prior information. We derive the Maximum-Likelihood filter in several ways, one of which corresponds to triangularizing the Hamiltonian dynamics. From the structure of the Hamiltonian it will become clear that we can *only* triangularize and not diagonalize the dynamics. We also present elements of a system theory for processes on trees in analyzing our filter, including notions of reachability, reconstructibility, and observability. We then provide a notion of stability for systems propagating upward on trees. Using our system theory, we give results on bounding the error covariance of our filter as well as results on asymptotic properties of the filter. Finally, we note that all of the results in this chapter apply equally well (with simple modifications) in 2D, using quadtrees instead of dyadic trees.

## 3.2 Dynamic Stochastic Models on Trees

Recall the following class of state-space models on trees:

$$x(t) = A(t)x(\gamma^{-1}t) + B(t)w(t) \tag{3.2.1}$$

where $\{w(t), t \in T\}$ is a set of independent, zero-mean Gaussian random variables. If we are dealing with a tree with unique root node, 0, we require $w(t)$ to be independent of $x(0)$, the zero-mean initial condition. The covariance of $w(t)$ is $I$ and that of $x(0)$ is $P_x(0)$. If we wish the model eq.(3.2.1) to define a process over the entire infinite tree, we simply require that $w(t)$ is independent of the "past" of $x$, i.e. $\{x(\tau) | m(\tau) < m(t)\}$. If $A(t)$ is invertible for all $t$, this is equivalent to requiring $w(t)$ to be independent of *some* $x(\tau)$ with $\tau \neq t$, $m(\tau) < m(t)$.

Let us make several comments about this model. Note first that the model *does* evolve along the tree, as both $x(\alpha t)$ and $x(\beta t)$ evolve from $x(t)$. Secondly, we note that, as with all of our lattice processes, this process has a Markovian property: given $x$ at scale $m$, $x$ at scale $m+1$ is independent of $x$ at scales less than or equal to $m-1$. Indeed for this to hold all we need is for $w$ to be independent from scale to scale and not necessarily at each individual node. Also, the analysis we perform from Section 3.4 onward focuses on the case in which $A(t)$ and $B(t)$ are functions of $m(t)$; i.e. we focus in these sections on a translation-invariant model where $A(t) = A(m(t))$ and $B(t) = B(m(t))$. As we will see this leads to significant computational simplification and also, when this dependence is chosen appropriately, these models lead to processes possessing self-similar properties from scale to scale. Note that in this case the approach in the previous chapter applies.

Note that the second-order statistics of $x(t)$ are easily computed. In particular the covariance $P_x(t) = E[x(t)x^T(t)]$ evolves according to a Lyapunov equation on the tree:

$$P_x(t) = A(t)P_x(\gamma^{-1}t)A^T(t) + B(t)B^T(t) \tag{3.2.2}$$

Let $K_{xx}(t,s) = E[x(t)x^T(s)]$. Let $s \wedge t$ denote the least upper bound of $s$ and $t$, i.e. the first node that is a predecessor of both $t$ and $s$. Then

$$K_{xx}(t,s) = \Phi(t, s \wedge t)P_x(s \wedge t)\Phi^T(s, s \wedge t) \tag{3.2.3}$$

where for $m(t_1) \geq m(t_2)$

$$\Phi(t_1,t_2) = \begin{cases} I & t_1 = t_2 \\ A(t_1)\Phi(\gamma^{-1}t_1,t_2) & m(t_1) > m(t_2) \end{cases} \tag{3.2.4}$$

As we will see in a moment, the multiscale estimation algorithm we will analyze involves a fine-to-coarse recursion requiring a corresponding version of eq.(3.2.1). Assuming that $A(t)$ is invertible for all $t$ we can directly apply the results of [53]:

$$x(\gamma^{-1}t) = F(t)x(t) - A^{-1}(t)B(t)\tilde{w}(t) \tag{3.2.5}$$

with

$$\begin{aligned} F(t) &= A^{-1}(t)[I - B(t)B^T(t)P_x^{-1}(t)] \\ &= P_x(\gamma^{-1}t)A^T(t)P_x^{-1}(t) \end{aligned} \tag{3.2.6}$$

and where

$$\tilde{w}(t) = w(t) - E[w(t)|x(t)] \tag{3.2.7}$$

$$\begin{aligned} E[\tilde{w}(t)\tilde{w}^T(t)] &= I - B^T(t)P_x^{-1}(t)B(t) \\ &\triangleq \tilde{Q}(t) \end{aligned} \tag{3.2.8}$$

Note that $\tilde{w}(t)$ is a white noise process along all upward paths on the tree – i.e. $\tilde{w}(s)$ and $\tilde{w}(t)$ are uncorrelated if $t = \gamma^{-r}s$ or $s = \gamma^{-r}t$ for some $r$; otherwise $\tilde{w}(s)$ and $\tilde{w}(t)$ are **not** uncorrelated.

We now formulate the problem of smoothing our processes based on multiscale measurements. We consider the estimation of the stochastic process described by eq.(3.2.1) based on the measurements

$$y(t) = C(t)x(t) + v(t) \tag{3.2.9}$$

where $\{v(t), t \in T\}$ is a set of independent zero-mean Gaussian random variables independent of $x(0)$ and $\{w(t), t \in T\}$. The covariance of $v(t)$ is $R(t)$. The model eq.(3.2.9) allows us to consider multiple resolution measurements of our process. This model provides a natural framework for addressing the sensor-fusion problem, where one is interested in optimally fusing measurements at various resolutions. This includes the case for example where one might be interested in fusing coarse data with *sparse* fine data. If $M$ denotes the finest scale, sparse data at this scale can be represented by having $C(t) = 0$ for some of the points at which $m(t) = M$. The single resolution problem, i.e. when $C(t) = 0$ unless $m(t) = M$, is also of interest as it corresponds to the problem of restoring a noise corrupted version of a stochastic process possessing a multiscale description.

Let us now examine what we will refer to as the scale-varying model, i.e. the model whose parameters vary in scale only, for which the analysis following Section 3.3 applies. In this case the covariance evolves according to the following equation.

$$P_x(t) = A(m(t))P_x(\gamma^{-1}t)A^T(m(t)) + B(m(t))B^T(m(t)) \tag{3.2.10}$$

Note in particular that if $P_x(\tau)$ depends only on $m(\tau)$ for $m(\tau) \leq m(t) - 1$, then $P_x(t)$ depends only on $m(t)$. We will assume that this is the case and therefore will write $P_x(t) = P_x(m(t))$. Note that this is always true if we are considering the subtree with single root node 0. Also if $A(m)$ is invertible for all $m$, and if $P_x(t) = P_x(m(t))$ at *some* scale(i.e. at all $t$ for which $m(t)$ equals $m$ for some $m$), then $P_x(t) = P_x(m(t))$ for *all* $t$. Our covariance, $K_{xx}(t,s)$, now has the following form.

$$K_{xx}(t,s) = \Phi(m(t), m(s \wedge t))P_x(m(s \wedge t))\Phi^T(m(s), m(s \wedge t)) \tag{3.2.11}$$

where for $m_1 \geq m_2$

$$\Phi(m_1, m_2) = \begin{cases} I & m_1 = m_2 \\ A(m_1)\Phi(m_1 - 1, m_2) & m_1 > m_2 \end{cases} \tag{3.2.12}$$

Also, let $d(s,t)$ denote the distance from $s$ to $t$, i.e. the number of branches on the shortest path from $s$ to $t$. Then $d(s,t) = d(s, s \wedge t) + d(t, s \wedge t) = d(t,s)$ and if $A(m(t)) = A$, then

$$K_{xx}(t,s) = A^{d(t,s\wedge t)} P_x(m(s \wedge t))(A^T)^{d(s,s\wedge t)} \tag{3.2.13}$$

Furthermore, if $A$ is stable and if $B(m(t)) = B$, let $P_x$ be the solution to the algebraic Lyapunov equation

$$P_x = AP_xA^T + BB^T \tag{3.2.14}$$

In this case if $P_x(0) = P_x$(if we have a root node), or if we assume that $P_x(\tau) = P_x$ for $m(\tau)$ sufficiently negative[1], then $P_x(t) = P_x$ for all $t$, and we have the stationary model

$$\begin{aligned} K_{xx}(t,s) &= A^{d(t,s\wedge t)} P_x (A^T)^{d(s,s\wedge t)} \\ &= K_{xx}(d(t, s \wedge t), d(s, s \wedge t)) \end{aligned} \tag{3.2.15}$$

As a final note, we point out that there is one class of scalar stochastic processes on trees that has been the subject of substantial analysis. In [3] these are referred to as stationary processes but we prefer to use that terminology for the larger class of processes for which $K_{xx}(t,s)$ depends only on $d(t, s \wedge t)$ and $d(s, s \wedge t)$. The class of processes considered in [3] is characterized by the condition that $K_{xx}(t,s)$ depends only on $d(s,t)$ and we refer to these as *isotropic processes.* Note that eq.(3.2.15) represents an isotropic covariance if $AP_x = P_xA^T$, which shows the connection to the class of reversible stochastic processes[1]. For example in the scalar case

$$K_{xx}(t,s) = \left\{ \frac{B^2}{1 - A^2} \right\} A^{d(s,t)} \tag{3.2.16}$$

Some of our other research has examined the modeling of isotropic processes on trees; this is the subject of [5, 6].

## 3.3 Two-Sweep, Rauch-Tung-Striebel Algorithm

In this section we derive the equations for our two-sweep algorithm for smoothing measurement data on trees. The algorithm is a generalization of the well-known

[1] Once again if $A$ is invertible, if $P_x(t) = P_x$ at *any* single node, $P_x(t) = P_x$ at *all* nodes.

Rauch-Tung-Striebel(RTS) smoothing algorithm for causal state models. Recall that the standard RTS algorithm involves a forward Kalman filtering sweep followed by a backward sweep to compute the smoothed estimates. The generalization to our models on trees has the same structure, with several important differences. First for the standard RTS algorithm the procedure is completely symmetric with respect to time – i.e. we can start with a reverse-time Kalman filtering sweep followed by a forward smoothing sweep. For processes on trees, the Kalman filtering sweep **must** proceed from fine-to-coarse(i.e. in the reverse direction from that in which the model eq.(3.2.1) is defined) followed by a coarse-to-fine smoothing sweep[2]. Furthermore the Kalman filtering sweep, using the backward model eq.'s(3.2.5-3.2.8) is somewhat more complex for processes on trees. In particular one full step of the Kalman filter recursion involves a measurement update, **two** parallel backward predictions(corresponding to backward prediction along both of the paths descending from a node), and the **fusion** of these predicted estimates. This last step has no counterpart for state models evolving in *time* and is one of the major reasons for the differences between the analysis of temporal Riccati equations and that presented in this paper. As a final remark we note that our algorithm involves a pyramidal set of steps and a considerable level of parallelism.

To begin, let us recall the structure of the standard Rauch-Tung-Striebel algorithm for a standard state model in time whose state we denote as $z(t)$. The first step of the process consists of a Kalman filter for computing $\hat{z}(t|t)$ for all $t$; at any time $t$ the prediction step yields $\hat{z}(t+1|t)$ while updating with the new measurement $y(t+1)$ yields $\hat{z}(t+1|t+1)$. The second step propagates backward in time combining the smoothed estimate $\hat{z}_s(t+1)$ with the filtered estimate at the previous point in time $\hat{z}(t|t)$ (or equivalently $\hat{z}(t+1|t)$) to compute $\hat{z}_s(t)$. In the case of estimation on trees, we have a very similar structure; indeed the backward sweep and measurement update are identical in form to the RTS algorithm. The prediction step is, however, somewhat more complex, and while it can be written as a single step, we prefer to

[2] The reason for this is not very complex. To allow the measurement on the tree at one point to contribute to the estimate at another point on the same level of the tree, one must use a recursion that first moves up and then down the tree. Reversing the order of these steps does not allow one to realize such contributions.

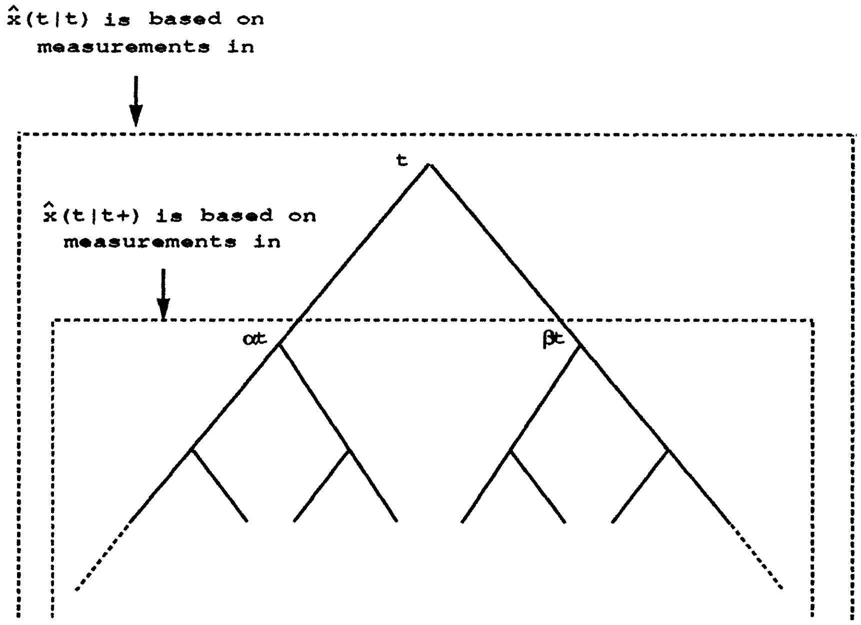
$\hat{x}(t|t)$ is based on measurements in
$\hat{x}(t|t+)$ is based on measurements in
t
αt
βt

Figure 3.3.1: Representation of Measurement Update and Merged Estimates

think of it as two parallel prediction steps, each as in RTS, followed by a *merge* step that has no counterpart for state models evolving in *time*. One other difference is that the forward sweep of our algorithm is from fine-to-coarse and thus involves the backward version eq.(3.2.5) of our original model eq.(3.2.1).

To begin let us define some notation:

$$\begin{aligned} Y_t &= \{y(s)|s = t \text{ or } s \text{ is a descendant of } t\} \\ &= \{y(s)|s \in (\alpha,\beta)^*t \ , \ m(s) \leq M\} && (3.3.17) \\ Y_t^+ &= \{y(s)|s \in (\alpha,\beta)^*t \ , \ t < m(s) \leq M\} && (3.3.18) \\ \hat{x}(\cdot|t) &= E[x(\cdot)|Y_t] && (3.3.19) \\ \hat{x}(\cdot|t+) &= E[x(\cdot)|Y_t^+] && (3.3.20) \end{aligned}$$

The interpretation of these estimates is provided in Figure 3.3.1.

We begin by considering the measurement update step. Specifically, suppose that we have computed $\hat{x}(t|t+)$ and the corresponding error covariance, $P(t|t+)$. Then, standard estimation results yield

$$\hat{x}(t|t) = \hat{x}(t|t+) + K(t)[y(t) - C(t)\hat{x}(t|t+)] \qquad (3.3.21)$$

$$K(t) = P(t|t+)C^T(t)V^{-1}(t) \tag{3.3.22}$$

$$V(t) = C(t)P(t|t+)C^T(t) + R(t) \tag{3.3.23}$$

and the resulting error covariance is given by

$$P(t|t) = [I - K(t)C(t)]P(t|t+) \tag{3.3.24}$$

Note that the computations begin on the finest level(m(t)=M) with $\hat{x}(t|t+) = 0$, $P(t|t+) = P_x(t)$ for $m(t) = M$.

Suppose now that we have computed $\hat{x}(\alpha t|\alpha t)$ and $\hat{x}(\beta t|\beta t)$. Note that $Y_{\alpha t}$ and $Y_{\beta t}$ are disjoint and these estimates can be calculated in parallel. We then compute $\hat{x}(t|\alpha t)$ and $\hat{x}(t|\beta t)$ which are given by

$$\hat{x}(t|\alpha t) = F(\alpha t)\hat{x}(\alpha t|\alpha t) \tag{3.3.25}$$

$$\hat{x}(t|\beta t) = F(\beta t)\hat{x}(\beta t|\beta t) \tag{3.3.26}$$

with corresponding error covariances given by

$$P(t|\alpha t) = F(\alpha t)P(\alpha t|\alpha t)F^T(\alpha t) + Q(\alpha t) \tag{3.3.27}$$

$$Q(\alpha t) = A^{-1}(\alpha t)B(\alpha t)\tilde{Q}(\alpha t)B^T(\alpha t)A^{-T}(\alpha t) \tag{3.3.28}$$

$$P(t|\beta t) = F(\beta t)P(\beta t|\beta t)F^T(\beta t) + Q(\beta t) \tag{3.3.29}$$

$$Q(\beta t) = A^{-1}(\beta t)B(\beta t)\tilde{Q}(\beta t)B^T(\beta t)A^{-T}(\beta t) \tag{3.3.30}$$

Eq.(3.3.25) and eq.(3.3.26) follow from projecting both sides of our backward model eq.(3.2.5) onto $Y_{\alpha t}$ and $Y_{\beta t}$, respectively. By noting that the dynamics of the one-step prediction error are identical to the dynamics of our backward model eq.(3.2.5), we arrive at eq.'s(3.3.27,3.3.29) by squaring both sides of the equation and taking expectations.

These estimates must then be merged to form $\hat{x}(t|t+)$. The derivation of this computation can be given as follows. By definition

$$\hat{x}(t|t+) = E[x(t)|Y_{\alpha t}, Y_{\beta t}] \tag{3.3.31}$$

But from our model, eq.(3.2.1), we can decompose $Y_{\alpha t}$ and $Y_{\beta t}$ in the following way.

$$Y_{\alpha t} = M_{\alpha t}x(t) + \xi_1 \tag{3.3.32}$$

$$Y_{\beta t} = M_{\beta t}x(t) + \xi_2 \tag{3.3.33}$$

where the matrices $M_{\alpha t}$ and $M_{\beta t}$ contain products of $A(s)$, $m(s) > m(t)$, and the vectors $\xi_1$ and $\xi_2$ are functions of the driving noises $w(s)$ and the measurement noises $v(s)$ for $s$ in the subtree strictly below $\alpha t$ and $s$ in the subtree strictly $\beta t$, respectively, the latter fact implying $\xi_1 \perp \xi_2$. We also let

$$\begin{aligned} R_{\alpha t} &= E[\xi_1 \xi_1^T] && (3.3.34) \\ R_{\beta t} &= E[\xi_2 \xi_2^T] && (3.3.35) \end{aligned}$$

We then write eq.(3.3.32) and eq.(3.3.33) as a single equation in the following way.

$$\mathcal{Y} = \mathcal{H}x(t) + \Xi \qquad (3.3.36)$$

where

$$\mathcal{H} = \begin{bmatrix} M_{\alpha t} \\ M_{\beta t} \end{bmatrix}, \Xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \mathcal{R} = E[\Xi \Xi^T] \qquad (3.3.37)$$

and $x(t) \perp \Xi$. We can write the optimal estimate of $x(t)$ given $\mathcal{Y}$ in the following way.

$$\begin{aligned} \hat{x}(t|t+) &= [P_x^{-1}(t) + \mathcal{H}^T \mathcal{R}^{-1} \mathcal{H}]^{-1} \mathcal{H}^T \mathcal{R}^{-1} \mathcal{Y} \\ &= [P_x^{-1}(t) + M_{\alpha t}^T R_{\alpha t}^{-1} M_{\alpha t} + M_{\beta t}^T R_{\beta t}^{-1} M_{\beta t}]^{-1} [M_{\alpha t}^T R_{\alpha t}^{-1} Y_{\alpha t} + M_{\beta t}^T R_{\beta t}^{-1} Y_{\beta t}] \\ & && (3.3.38) \end{aligned}$$

But since

$$\begin{aligned} P(t|\alpha t) &= [P_x^{-1}(t) + M_{\alpha t}^T R_{\alpha t}^{-1} M_{\alpha t}]^{-1} && (3.3.39) \\ P(t|\beta t) &= [P_x^{-1}(t) + M_{\beta t}^T R_{\beta t}^{-1} M_{\beta t}]^{-1} && (3.3.40) \end{aligned}$$

we can rewrite eq.(3.3.38) as

$$\begin{aligned} \hat{x}(t|t+) &= P(t|t+)[P^{-1}(t|\alpha t)\hat{x}(t|\alpha t) + P^{-1}(t|\beta t)\hat{x}(t|\beta t)] \\ & && (3.3.41) \\ P(t|t+) &= [P^{-1}(t|\alpha t) + P^{-1}(t|\beta t) - P_x^{-1}(t)]^{-1} && (3.3.42) \end{aligned}$$

We now derive the formulas for the second part of the RTS algorithm involving the propagation downward along the tree combining the smoothed estimate $\hat{x}_s(\gamma^{-1}t)$

with the filtered estimate $\hat{x}(t|t)$ to produce $\hat{x}_s(t)$. Our derivation relies essentially on the following orthogonal decomposition of $Y_0$(the measurements at every node on the tree).

For each $t$, $Y_t$, as defined in eq.(3.3.17) is the set of measurements in the subtree beneath $t$(and including the measurement at $t$). Let $Y_{\bar{t}}$ denote all the remaining measurements, and viewing this as one large vector, define

$$\nu_{\bar{t}|t} = Y_{\bar{t}} - E[Y_{\bar{t}}|Y_t] \tag{3.3.43}$$

so that $\nu_{\bar{t}|t} \perp Y_t$ and the linear span of the set of *all* measurements, $Y_0$, is given by

$$\text{span } Y_0 = \text{ span } \{Y_t, Y_{\bar{t}}\} = \text{ span } \left\{Y_t, \nu_{\bar{t}|t}\right\} \tag{3.3.44}$$

Then

$$\begin{aligned} \hat{x}_s(t) &= E[x(t)|Y_t, \nu_{\bar{t}|t}] \\ &= \hat{x}(t|t) + E[x(t)|\nu_{\bar{t}|t}] \end{aligned} \tag{3.3.45}$$

If we write $x(t)$ as

$$x(t) = \tilde{x}(t|t) + \hat{x}(t|t) \tag{3.3.46}$$

and note that

$$\hat{x}(t|t) \perp \nu_{\bar{t}|t} \tag{3.3.47}$$

then we can write the following.

$$\hat{x}_s(t) = \hat{x}(t|t) + E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] \tag{3.3.48}$$

Using the same argument on $\hat{x}_s(\gamma^{-1}t)$ allows us to write

$$\hat{x}_s(\gamma^{-1}t) = \hat{x}(\gamma^{-1}t|t) + E[\tilde{x}(\gamma^{-1}t|t)|\nu_{\bar{t}|t}] \tag{3.3.49}$$

Suppose the following equality were to hold.

$$E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] = L(E[\tilde{x}(\gamma^{-1}t|t)|\nu_{\bar{t}|t}]) \tag{3.3.50}$$

where $L$ is a matrix. Then eq.(3.3.48) and eq.(3.3.49) could be combined to yield the following formula.

$$\hat{x}_s(t) = \hat{x}(t|t) + L\left[\hat{x}_s(\gamma^{-1}t) - \hat{x}(\gamma^{-1}t|t)\right] \tag{3.3.51}$$

We now proceed to show that eq.(3.3.50) indeed holds and compute explicitly the matrix $L$. We begin with the following iterated expectation.

$$E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] = E[E[\tilde{x}(t|t)|\tilde{x}(\gamma^{-1}t|t), \nu_{\bar{t}|t}]|\nu_{\bar{t}|t}] \tag{3.3.52}$$

We now examine the inner expectation, $E[\tilde{x}(t|t)|\tilde{x}(\gamma^{-1}t|t), \nu_{\bar{t}|t}]$, in detail. In particular the linear span of $\left\{\tilde{x}(\gamma^{-1}t|t), \nu_{\bar{t}|t}\right\}$ has the following structure.

$$\begin{aligned} \text{span } \left\{\tilde{x}(\gamma^{-1}t|t), \nu_{\bar{t}|t}\right\} &= \text{span } \left\{\tilde{x}(\gamma^{-1}t|t), \tilde{w}_{s|x_s}, w_{s'}, v_{s''}\right\} && (3.3.53) \\ \tilde{x}(\gamma^{-1}t|t) &\perp \tilde{w}_{s|x_s}, w_{s'}, v_{s''} && (3.3.54) \end{aligned}$$

where

$$s, s', s'' \not\in \text{ subtree under } t \tag{3.3.55}$$

To show this we note the following decomposition of $Y_{\bar{t}}$.

$$Y_{\bar{t}} = L_1 x(\gamma^{-1}t) + f(\tilde{w}_{s|x_s}, w_{s'}, v_{s''}) \tag{3.3.56}$$

where $f$ is a linear function of its arguments. Substituting eq.(3.3.56) into eq.(3.3.43) yields

$$\nu_{\bar{t}|t} = L_1 \tilde{x}(\gamma^{-1}t|t) + f(\tilde{w}_{s|x_s}, w_{s'}, v_{s''}) \tag{3.3.57}$$

where we have used the fact that $f(\tilde{w}_{s|x_s}, w_{s'}, v_{s''}) \perp Y_t$. The fact that $\tilde{x}(\gamma^{-1}t|t) \perp f(\tilde{w}_{s|x_s}, w_{s'}, v_{s''})$ verifies eq.(3.3.53). Using eq.(3.3.53) we have that

$$E[\tilde{x}(t|t)|\tilde{x}(\gamma^{-1}t|t), \nu_{\bar{t}|t}] = E[\tilde{x}(t|t)|\tilde{x}(\gamma^{-1}t|t)] \tag{3.3.58}$$

where we have also used the fact that $f(\tilde{w}_{s|x_s}, w_{s'}, v_{s''}) \perp \tilde{x}(t|t)$. Substituting eq.(3.3.58) into eq.(3.3.52) we get

$$E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] = E[E[\tilde{x}(t|t)|\tilde{x}(\gamma^{-1}t|t)]|\nu_{\bar{t}|t}] \tag{3.3.59}$$

But by using our backward equations, eq.(3.2.5), eq.(3.3.25)(in the latter case with $\alpha t \mapsto t$ and $t \mapsto \gamma^{-1}t$) we find that

$$E[\tilde{x}(t|t)|\tilde{x}(\gamma^{-1}t|t)] = P(t|t)F^T(t)P^{-1}(\gamma^{-1}t|t)\tilde{x}(\gamma^{-1}t|t) \tag{3.3.60}$$

This leads to our desired result.

$$E[\tilde{x}(t|t)|\nu_{\bar{t}|t}] \;=\; P(t|t)F^T(t)P^{-1}(\gamma^{-1}t|t)E[\tilde{x}(\gamma^{-1}t|t)|\nu_{\bar{t}|t}] \tag{3.3.61}$$

Finally, eq.(3.3.50), eq.(3.3.51), and eq.(3.3.61) yield the following smoothing formula.

$$\hat{x}_s(t) = \hat{x}(t|t) + P(t|t)F^T(t)P^{-1}(\gamma^{-1}t|t)\left[\hat{x}_s(\gamma^{-1}t) - \hat{x}(\gamma^{-1}t|t)\right] \tag{3.3.62}$$

We can easily derive a recursion for the smoothing error. Let

$$\tilde{x}_s(t) \;\triangleq\; x(t) - \hat{x}_s(t) \tag{3.3.63}$$

$$J(t) \;\triangleq\; P(t|t)F^T(t)P^{-1}(\gamma^{-1}t|t) \tag{3.3.64}$$

Subtracting $x(t)$ from both sides of eq.(3.3.62) and changing signs on both sides, we get

$$\tilde{x}_s(t) = \tilde{x}(t|t) + J(t)\left[\tilde{x}_s(\gamma^{-1}t) - \tilde{x}(\gamma^{-1}t|t)\right] \tag{3.3.65}$$

By rearranging terms we get

$$\tilde{x}_s(t) - J(t)\hat{x}_s(\gamma^{-1}t) = \tilde{x}(t|t) - J(t)\hat{x}(\gamma^{-1}t|t) \tag{3.3.66}$$

By multiplying both sides of eq.(3.3.66) on the right by its transpose and taking expectations, we get

$$P_s(t) + J(t)E[\hat{x}_s(\gamma^{-1}t)\hat{x}_s^T(\gamma^{-1}t)]J^T(t) \;=\; P(t|t) + J(t)E[\hat{x}(\gamma^{-1}t|t)\hat{x}^T(\gamma^{-1}t|t)]J^T(t) \tag{3.3.67}$$

$$P_s(t) \;=\; E[[\tilde{x}_s(t)\tilde{x}_s^T(t)] \tag{3.3.68}$$

where we have relied on the fact that

$$E[\tilde{x}_s(t)\hat{x}_s^T(\gamma^{-1}t)] \;=\; 0 \tag{3.3.69}$$

$$E[\tilde{x}(t|t)\hat{x}^T(\gamma^{-1}t|t)] \;=\; 0 \tag{3.3.70}$$

And finally, since

$$E[\hat{x}_s(\gamma^{-1}t)\hat{x}_s^T(\gamma^{-1}t)] \;=\; P_x(\gamma^{-1}t) - P_s(\gamma^{-1}t) \tag{3.3.71}$$

$$E[\hat{x}(\gamma^{-1}t|t)\hat{x}^T(\gamma^{-1}t|t)] \;=\; P_x(\gamma^{-1}t) - P(\gamma^{-1}t|t) \tag{3.3.72}$$

it follows that

$$P_s(t) = P(t|t) + J(t)[P_s(\gamma^{-1}t) - P(\gamma^{-1}t|t)]J^T(t) \tag{3.3.73}$$

We now summarize the overall two-sweep algorithm:

**Upward Sweep**

Measurement Update:

$$\hat{x}(t|t) = \hat{x}(t|t+) + K(t)[y(t) - C(t)\hat{x}(t|t+)] \tag{3.3.74}$$

$$K(t) = P(t|t+)C^T(t)V^{-1}(t) \tag{3.3.75}$$

$$V(t) = C(t)P(t|t+)C^T(t) + R(t) \tag{3.3.76}$$

$$P(t|t) = [I - K(t)C(t)]P(t|t+) \tag{3.3.77}$$

One-step Prediction:

$$\hat{x}(\gamma^{-1}t|t) = F(t)\hat{x}(t|t) \tag{3.3.78}$$

$$P(\gamma^{-1}t|t) = F(t)P(t|t)F^T(t) + Q(t) \tag{3.3.79}$$

$$Q(t) = A^{-1}(t)B(t)\tilde{Q}(t)B^T(t)A^{-T}(t) \tag{3.3.80}$$

Merge Step:

$$\hat{x}(t|t+) = P(t|t+)[P^{-1}(t|\alpha t)\hat{x}(t|\alpha t) + P^{-1}(t|\beta t)\hat{x}(t|\beta t)] \tag{3.3.81}$$

$$P(t|t+) = [P^{-1}(t|\alpha t) + P^{-1}(t|\beta t) - P_x^{-1}(t)]^{-1} \tag{3.3.82}$$

**Downward Sweep**

$$\hat{x}_s(t) = \hat{x}(t|t) + J(t)\left[\hat{x}_s(\gamma^{-1}t) - \hat{x}(\gamma^{-1}t|t)\right] \tag{3.3.83}$$

$$P_s(t) = P(t|t) + J(t)[P_s(\gamma^{-1}t) - P(\gamma^{-1}t|t)]J^T(t) \tag{3.3.84}$$

$$J(t) = P(t|t)F^T(t)P^{-1}(\gamma^{-1}t|t) \tag{3.3.85}$$

Finally, in looking ahead to the results in the following sections we summarize the overall two-sweep algorithm for the scale-varying model.

**Upward Sweep**

Measurement Update:

$$\hat{x}(t|t) = \hat{x}(t|t+) + K(m(t))[y(t) - C(m(t))\hat{x}(t|t+)] \tag{3.3.86}$$

$$K(m(t)) = P(m(t)|m(t)+)C^T(m(t))V^{-1}(m(t)) \tag{3.3.87}$$

$$V(m(t)) = C(m(t))P(m(t)|m(t)+)C^T(m(t)) + R(m(t)) \tag{3.3.88}$$

$$P(m(t)|m(t)) = [I - K(m(t))C(m(t))]P(m(t)|m(t)+) \tag{3.3.89}$$

One-step Prediction:

$$\hat{x}(\gamma^{-1}t|t) = F(m(t))\hat{x}(t|t) \tag{3.3.90}$$

$$P(m(t)-1|m(t)) = F(m(t))P(m(t)|m(t))F^T(m(t)) + Q(m(t)) \tag{3.3.91}$$

$$Q(m(t)) = A^{-1}(m(t))B(m(t))\tilde{Q}(m(t))B^T(m(t))A^{-T}(m(t)) \tag{3.3.92}$$

Merge Step:

$$\hat{x}(t|t+) = P(m(t)|m(t)+)P^{-1}(m(t)|m(t)+1)[\hat{x}(t|\alpha t) + \hat{x}(t|\beta t)] \tag{3.3.93}$$

$$P(m(t)|m(t)+) = [2P^{-1}(m(t)|m(t)+1) - P_x^{-1}(t)]^{-1} \tag{3.3.94}$$

**Downward Sweep**

$$\hat{x}_s(t) = \hat{x}(t|t) + J(m(t))\left[\hat{x}_s(\gamma^{-1}t) - \hat{x}(\gamma^{-1}t|t)\right] \tag{3.3.95}$$

$$P_s(t) = P(t|t) + J(m(t))[P_s(\gamma^{-1}t) - P(\gamma^{-1}t|t)]J^T(m(t)) \tag{3.3.96}$$

$$J(m(t)) = P(m(t)|m(t))F^T(m(t))P^{-1}(m(t)-1|m(t)) \tag{3.3.97}$$

## 3.4 Maximum Likelihood Estimator

In this section we examine the difficulties in analyzing our filtering equations. These difficulties point to the need to decompose the filter into two parts; one representing what we call the **ML filter**, and the other representing our estimate of the mean of the process. Note that while everything in this section can be developed for the general case, we develop our ideas in the scale-varying case, i.e. the case where the model parameters vary only with scale, for which the results in the following sections apply.

We begin by rewriting the set of Riccati equations for our filtering problem as follows.

$$\begin{aligned} P(m|m+1) &= F(m+1)P(m+1|m+1)F^T(m+1) \\ &+ G(m+1)\tilde{Q}(m+1)G^T(m+1) \qquad (3.4.1) \\ P^{-1}(m|m) &= P^{-1}(m|m^+) + C^T(m)R^{-1}(m)C(m) \qquad (3.4.2) \\ P^{-1}(m|m^+) &= 2P^{-1}(m|m+1) - P_x^{-1}(m) \qquad (3.4.3) \end{aligned}$$

where

$$G(m(t)) \triangleq -A^{-1}(m(t))B(m(t)) \qquad (3.4.4)$$

Note that we can combine eq.(3.4.2,3.4.3) into the following single equation.

$$P^{-1}(m|m) = 2P^{-1}(m|m+1) - P_x^{-1}(m) + C^T(m)R^{-1}(m)C(m) \qquad (3.4.5)$$

The Riccati equations for our optimal filter, eq.'s(3.4.1-3.4.3), differ from standard Riccati equations in two respects: 1) the explicit presence of the prior state covariance $P_x(m(t))$ and 2) the presence of a scaling factor of 2 in eq.(3.4.3). The scaling factor is intrinsic to our Riccati equations and is due to the fact that we are fusing **pairs** of parallel information paths in going from level to level. The presence of $P_x(m(t))$ in the Riccati equations accounts for the double counting of prior information in performing this merge.

The presence of this prior variance term points to a significant complication in analyzing our filter. Specifically, in standard Kalman filtering analysis the Riccati equation for the error covariance can be viewed simply as the covariance of the error equations, which can be analyzed directly without explicitly examining the state dynamics since the error evolves as a state process itself. This is apparently **not** the case here because of the explicit presence of $P_x(m)$ in eq.(3.4.5). Indeed as we show later in this section, if one examines the backward model eq.'s(3.2.5-3.2.8) and the Kalman filter eq.'s(3.3.21,3.3.25,3.3.26,3.3.41) one finds that the upward dynamics for the error $x(t) - \hat{x}(t|t)$ are **not** decoupled from $x(t)$ **unless** $P_x^{-1}(m(t)) = 0$. This motivates the following decomposition of the estimator into a dynamic part based on an estimator in which $P_x^{-1} = 0$(the ML estimator) followed by a gain adjustment to account for prior information.

To be precise, let $P_{ML}(m|m+1)$ and $P_{ML}(m|m)$ denote the covariance of the estimation errors produced by our upward Kalman filter assuming that $P_x^{-1}(m) = 0$ for all $m$. These satisfy the following Riccati equation, which doesn't depend explicitly on $P_x(m)$.

$$P_{ML}(m|m+1) = A^{-1}(m+1)P_{ML}(m+1|m+1)A^{-T}(m+1) + G(m+1)G^T(m+1) \tag{3.4.6}$$

$$P_{ML}^{-1}(m|m) = 2P_{ML}^{-1}(m|m+1) + C^T(m)R^{-1}(m)C(m) \tag{3.4.7}$$

where our initial condition at level $M$ is $P_{ML}^{-1}(M|M) = 0$. As we will show the filtering equations for the ML estimator correspond exactly with the equations for the optimal(Bayesian) filter with $P_{ML}(m|m)$ and $P_{ML}(m|m+1)$ being substituted for $P(m|m)$ and $P(m|m+1)$ and by setting $P_x^{-1}(m) = 0$. In particular they can be written as follows.

**ML Filtering Equations**

Measurement Update:

$$\hat{x}_{ML}(t|t) = \hat{x}_{ML}(t|t+) + K_{ML}(m(t))[y(t) - C(m(t))\hat{x}_{ML}(t|t+)] \tag{3.4.8}$$

$$K_{ML}(m(t)) = P_{ML}(m(t)|m(t)+)C^T(m(t))V^{-1}(m(t)) \tag{3.4.9}$$

$$V(m(t)) = C(m(t))P_{ML}(m(t)|m(t)+)C^T(m(t)) + R(m(t)) \tag{3.4.10}$$

$$P_{ML}(m(t)|m(t)) = [I - K_{ML}(m(t))C(m(t))]P_{ML}(m(t)|m(t)+) \tag{3.4.11}$$

One-step Prediction:

$$\hat{x}_{ML}(\gamma^{-1}t|t) = A^{-1}(m(t))\hat{x}_{ML}(t|t) \tag{3.4.12}$$

$$\begin{aligned} P_{ML}(m(t)-1|m(t)) &= A^{-1}(m(t))P_{ML}(m(t)|m(t))A^{-T}(m(t)) \\ &+ G(m(t))G^T(m(t)) \end{aligned} \tag{3.4.13}$$

Merge Step:

$$\hat{x}_{ML}(t|t+) = P_{ML}(m(t)|m(t)+)P_{ML}^{-1}(m(t)|m(t)+1)[\hat{x}_{ML}(t|\alpha t) + \hat{x}_{ML}(t|\beta t)]$$

$$\tag{3.4.14}$$

$$P_{ML}^{-1}(m(t)|m(t)+) \;=\; 2P_{ML}^{-1}(m(t)|m(t)+1) \tag{3.4.15}$$

Before elaborating further on the ML estimator, we describe its relationship to the optimal estimator. The two are related in the following way.

$$\hat{x}(t|t) \;=\; P(m(t)|m(t))P_{ML}^{-1}(m(t)|m(t))\hat{x}_{ML}(t|t) \tag{3.4.16}$$

$$P^{-1}(m(t)|m(t)) \;=\; P_{ML}^{-1}(m(t)|m(t)) + P_x^{-1}(m(t)) \tag{3.4.17}$$

To derive these relationships we start by writing

$$Y_t = \mathcal{H}_t x(t) + \theta(t) \tag{3.4.18}$$

where

$$E[\theta(t)\theta^T(t)] \;=\; \mathcal{R}_t \tag{3.4.19}$$

$$E[\theta(t)] \;=\; 0 \tag{3.4.20}$$

Recall that $Y_t$ is the set $\{y(s)|s = t \text{ or } s \text{ is a descendant of } t\}$. Eq.(3.4.18) follows directly from our downward model for the process $x(t)$ on the tree. From eq.(3.4.18) we can write the following maximum likelihood estimate for $x(t)$, i.e. the estimate that maximizes the distribution of $Y_t$ conditioned on $x(t)$ where $x(t)$ is considered to be an unknown, **deterministic** parameter.

$$\hat{x}_{ML}(t|t) \;=\; (\mathcal{H}_t^T\mathcal{R}_t^{-1}\mathcal{H}_t)^{-1}\mathcal{H}_t^T\mathcal{R}_t^{-1}Y_t \tag{3.4.21}$$

$$P_{ML}^{-1}(m(t)|m(t)) \;=\; \mathcal{H}_t^T\mathcal{R}_t^{-1}\mathcal{H}_t \tag{3.4.22}$$

To derive an expression for the Bayesian estimate of $x(t)$ we can augment our previous measurements with an extra "measurement" of $x(t)$ in the following way.

$$\begin{bmatrix} Y_t \\ \overline{x}(t) \end{bmatrix} = \begin{bmatrix} \mathcal{H}_t \\ I \end{bmatrix} x(t) + \begin{bmatrix} \theta(t) \\ \eta(t) \end{bmatrix} \tag{3.4.23}$$

where

$$E[\theta(t)\eta^T(t)] \;=\; 0 \tag{3.4.24}$$

$$E[\eta(t)\eta^T(t)] \;=\; P_x(m(t)) \tag{3.4.25}$$

$$E[\eta(t)] \;=\; 0 \tag{3.4.26}$$

and $\overline{x}(t)$ is the mean of $x(t)$. The maximum likelihood solution to this problem gives the Bayesian estimate of $x(t)$, i.e. $E[x(t)|Y_t]$. The solution is given as follows.

$$\hat{x}(t|t) = P(m(t)|m(t))(P_{ML}^{-1}(m(t)|m(t))\hat{x}_{ML}(t|t) + P_x^{-1}(t)\overline{x}(t)) \tag{3.4.27}$$

$$P^{-1}(m(t)|m(t)) = P_{ML}^{-1}(m(t)|m(t)) + P_x^{-1}(m(t)) \tag{3.4.28}$$

Since we consider $x(t)$ to be a zero-mean process, eq.(3.4.27) and eq.(3.4.16) are equivalent.

We now proceed to show that $\hat{x}_{ML}(t|t)$ can in fact be computed using what we have defined as our ML filtering equations. In order to derive a recursion for $\hat{x}_{ML}(t|t)$ we solve the following ML problem for $x(t)$, whose solution can be shown to be equivalent to the solution of the ML problem defined by eq.'s(3.4.18-3.4.20).

$$\begin{bmatrix} y(t) \\ \hat{x}_{ML}(\alpha t|\alpha t) \\ \hat{x}_{ML}(\beta t|\beta t) \end{bmatrix} = \begin{bmatrix} C(m(t)) \\ A(m(t)) \\ A(m(t)) \end{bmatrix} x(t) + \begin{bmatrix} v(t) \\ B(m(t))w(\alpha t) + \tilde{x}_{ML}(\alpha t|\alpha t) \\ B(m(t))w(\beta t) + \tilde{x}_{ML}(\beta t|\beta t) \end{bmatrix} \tag{3.4.29}$$

where

$$\tilde{x}_{ML}(\alpha t|\alpha t) \triangleq \hat{x}_{ML}(\alpha t|\alpha t) - x(\alpha t) \tag{3.4.30}$$

$$\tilde{x}_{ML}(\beta t|\beta t) \triangleq \hat{x}_{ML}(\beta t|\beta t) - x(\beta t) \tag{3.4.31}$$

That the ML estimate of $x(t)$ given measurements defined in eq.(3.4.29) is equivalent to the ML estimate of $x(t)$ given measurements defined in eq.(3.4.18) is a consequence of the following lemma(for a proof of this result see [41]).

**Lemma 3.4.1** *Let $x$ and $z$ be unknown vectors and consider the following measurements*

$$y_1 = H_x x + v \tag{3.4.32}$$

$$y_2 = J_x x + J_z z + w \tag{3.4.33}$$

*where $v$ and $w$ are independent, zero-mean Gaussian vectors. If $x$ is estimable based on eq.(3.4.32), then we denote this ML estimate as $\hat{x}_1$ and its estimation error by $U_1$.*

*Consider the following measurement*

$$\hat{x}_1 = x + u \tag{3.4.34}$$

*where u is zero-mean Gaussian with variance $U_1$. If z is estimable based on eq.'s(3.4.32,3.4.33), its ML estimate and estimation error are equivalent to the ML estimate and estimation error of z based on eq.'s(3.4.34,3.4.33).*

The particular form of eq.(3.4.29) follows directly from our downward model. We can use the properties of this model to show the following.

$$\begin{aligned}
\Pi &\triangleq \begin{bmatrix} v(t) \\ B(m(t))w(\alpha t) + \tilde{x}_{ML}(\alpha t|\alpha t) \\ B(m(t))w(\beta t) + \tilde{x}_{ML}(\beta t|\beta t) \end{bmatrix} && (3.4.35) \\
E[\Pi\,\Pi^T] &= \begin{bmatrix} R(m(t)) & 0 & 0 \\ 0 & \Pi_1 & 0 \\ 0 & 0 & \Pi_2 \end{bmatrix} && (3.4.36) \\
\Pi_1 &= P_{ML}(\alpha t|\alpha t) + B(m(t))B^T(m(t)) && (3.4.37) \\
\Pi_2 &= P_{ML}(\beta t|\beta t) + B(m(t))B^T(m(t)) && (3.4.38) \\
& && (3.4.39)
\end{aligned}$$

Note that as we are considering the case where our model parameters vary only with scale, $\Pi_1$ in fact equals $\Pi_2$ and thus, we will now denote the updated error covariance as $P(m(t)|m(t))$. If we let

$$\begin{aligned}
\tilde{H} &\triangleq \begin{bmatrix} C(m(t)) \\ A(m(t)) \\ A(m(t)) \end{bmatrix} && (3.4.40) \\
\tilde{Y} &\triangleq \begin{bmatrix} y(t) \\ \hat{x}_{ML}(\alpha t|\alpha t) \\ \hat{x}_{ML}(\beta t|\beta t) \end{bmatrix} && (3.4.41) \\
\tilde{R} &\triangleq E[\Pi\,\Pi^T] && (3.4.42)
\end{aligned}$$

then our ML solution for $x(t)$ is given as follows.

$$\begin{aligned}
\hat{x}_{ML}(t|t) &= (\tilde{H}^T \tilde{R}^{-1} \tilde{H})^{-1} \tilde{H}^T \tilde{R}^{-1} \tilde{Y} \\
&= P_{ML}(m(t)|m(t)) \left[C^T(m(t)) R^{-1}(m(t)) y(t) + A^T(m(t))(P_{ML}(m(\alpha t)|m(\alpha t))\right. \\
&+ \left. B(m(t))B^T(m(t)))^{-1} \left(\tilde{x}_{ML}(\alpha t|\alpha t) + \hat{x}_{ML}(\beta t|\beta t)\right)\right] \qquad (3.4.43) \\
P_{ML}^{-1}(m(t)|m(t)) &= C^T(m(t)) R^{-1} C(m(t)) \\
&+ 2A^T(m(t))(P_{ML}(m(\alpha t)|m(\alpha t)) + B(m(t))B^T(m(t)))^{-1} A(m(t)) \\
& \qquad (3.4.44)
\end{aligned}$$

We can derive the ML predicted estimate as a function of the ML updated estimate in a similar fashion. In particular $\hat{x}_{ML}(\gamma^{-1}t|t)$, the ML estimate of $x(\gamma^{-1}t)$ based on $Y_t$, can be viewed as the ML estimate of $x(\gamma^{-1}t)$ based on the following "measurement".

$$\hat{x}_{ML}(t|t) = A(m(t)) x_{ML}(\gamma^{-1}t) + B(m(t)) w(t) + \tilde{x}_{ML}(t|t) \tag{3.4.45}$$

The ML estimator has the following solution.

$$\begin{aligned}
P_{ML}(m(\gamma^{-1}t)|m(t)) &= A^{-1}(m(t)) P_{ML}(m(t)|m(t)) A^{-T}(m(t)) \\
&+ A^{-1}(m(t)) B(m(t)) B^T(m(t)) A^{-T}(m(t)) \qquad (3.4.46) \\
\hat{x}_{ML}(\gamma^{-1}t|t) &= A^{-1}(m(t)) x_{ML}(t|t) \qquad (3.4.47)
\end{aligned}$$

We see that eq.'s(3.4.44,3.4.46) are precisely what we defined earlier as our ML Riccati equation in eq.'s(3.4.6,3.4.7). As an alternative way of seeing that our ML equations indeed provide recursions for an estimator whose error covariance $P_{ML}(m(t))$ satisfies eq.(3.4.28) we can start with the recursions for $P_{ML}(m(t))$, $P(m(t))$, and $P_x(m(t))$, then show that eq.(3.4.28) is satisfied for all $m$. This is done in Appendix 3A.

There are several reasons for viewing the optimal estimator as being decomposed into an ML part and a part due to the prior model. One is that the ML Riccati equations are simpler because they do not include the explicit presence of the prior information $P_x^{-1}(m(t))$. This simplicity is significant in that the ML Riccati equations are readily amenable to stability analysis. The important reason mentioned previously for focusing our analysis on the ML filter, and perhaps a deeper one, is that the error dynamics for the optimal filter cannot be written as a noise driven process with closed-loop dynamics whereas the error dynamics for the ML filter **can**. Let us flesh out this last point in more detail.

Let us begin by examining the dynamics of our filter in the upward sweep of the RTS algorithm, eq.'s(3.3.21-3.3.24, 3.3.25-3.3.28,3.3.41,3.3.42). We can rewrite the dynamics of the filter in update form, eq.(3.3.21), as follows.

$$\begin{aligned} \hat{x}(t|t) &= L(m(t))F(m(t)+1)(\hat{x}(\alpha t|\alpha t)+\hat{x}(\beta t|\beta t)) \\ &+ K(m(t))y(t) \qquad (3.4.48) \\ L(m(t)) &= P(m(t)|m(t))P^{-1}(m(t)|m(t)+1) \qquad (3.4.49) \end{aligned}$$

We can also write the dynamics for our process in a similarly symmetric form.

$$x(t)=\frac{1}{2}F(m(t)+1)[x(\alpha t)+x(\beta t)]+\frac{1}{2}G(m(t)+1)[\tilde{w}(\alpha t)+\tilde{w}(\beta t)] \qquad (3.4.50)$$

We can easily rewrite eq.(3.4.48) as

$$\begin{aligned} \hat{x}(t|t) &= (I-K(m(t))C(m(t)))L'(m(t))F(m(t)+1)(\hat{x}(\alpha t|\alpha t) \\ &+ \hat{x}(\beta t|\beta t))+K(m(t))y(t) \qquad (3.4.51) \\ L'(m(t)) &= P(m(t)|m(t)+)P^{-1}(m(t)|m(t)+1) \qquad (3.4.52) \end{aligned}$$

By doing straightforward manipulations on eq.(3.4.51) and eq.(3.4.50) we can get

$$\begin{aligned} \tilde{x}(t|t) &= (I-K(m(t))C(m(t)))x(t)-K(m(t))v(t) \\ &- (I-K(m(t))C(m(t)))L'(m(t))F(m(t)+1)(\hat{x}(\alpha t|\alpha t)+\hat{x}(\beta t|\beta t)) \\ & \qquad (3.4.53) \\ \tilde{x}(t|t) &= x(t)-\hat{x}(t|t) \qquad (3.4.54) \end{aligned}$$

The difficulty in proceeding any further with eq.(3.4.53) lies in the presence of the term $L'(m(t))$. In standard filtering $L'(m(t)) = I$; said another way there is no difference between $P(m(t)|m(t)+)$ and $P(m(t)|m(t)+1)$. Let us now write down the equations for the ML filter and its corresponding error.

$$\begin{aligned} \hat{x}_{ML}(t|t) &= \frac{1}{2}(I-K_{ML}(m(t))C(m(t)))A^{-1}(m(t)+1)(\hat{x}_{ML}(\alpha t|\alpha t)+\hat{x}_{ML}(\beta t|\beta t)) \\ &+ K_{ML}(m(t))y(t) \qquad (3.4.55) \end{aligned}$$

$$\begin{aligned}\tilde{x}_{ML}(t|t) &= (I - K_{ML}(m(t))C(m(t)))x(t) - K_{ML}(m(t))v(t) \\ &- \frac{1}{2}(I - K_{ML}(m(t))C(m(t)))A^{-1}(m(t)+1)(\hat{x}_{ML}(\alpha t|\alpha t) + \hat{x}_{ML}(\beta t|\beta t))\end{aligned} \tag{3.4.56}$$

Note that we can write the dynamics for our process using our downward model as follows.

$$x(t) = \frac{1}{2}A^{-1}(m(t)+1)(x(\alpha t) + x(\beta t)) - \frac{1}{2}A^{-1}(m(t)+1)B(m(t)+1)(w(\alpha t) + w(\beta t)) \tag{3.4.57}$$

By substituting eq.(3.4.57) into eq.(3.4.56) we get

$$\begin{aligned}\tilde{x}_{ML}(t|t) &= \frac{1}{2}(I - K_{ML}(m(t))C(m(t)))A^{-1}(m(t)+1)(\tilde{x}_{ML}(\alpha t|\alpha t) + \tilde{x}_{ML}(\beta t|\beta t)) \\ &- \frac{1}{2}(I - K_{ML}(m(t))C(m(t)))G(m(t)+1)(w(\alpha t) + w(\beta t)) - K_{ML}(m(t))v(t)\end{aligned} \tag{3.4.58}$$

Note that eq.(3.4.58) has the same algebraic structure as the the equations for the error dynamics of the standard Kalman filter except for the scaling factor of $\frac{1}{2}$ and the fact that there are two terms in the immediate past being merged. Both the scaling factor and the merging of pairs of points is crucial to the study of the stability of the filter. As we will see in Section 3.8 the appropriate scaling factor is necessary for controlling in some sense the potential growth that might occur in merging points.

Also, for future reference, let us rewrite eq.(3.4.58) using the following equality:

$$\frac{1}{2}(I - K_{ML}(m(t))C(m(t))) = P_{ML}(m(t)|m(t))P_{ML}^{-1}(m(t)|m(t)+1) \tag{3.4.59}$$

We can rewrite eq.(3.4.58) as

$$\begin{aligned}\tilde{x}_{ML}(t|t) &= P_{ML}(m(t)|m(t))P_{ML}^{-1}(m(t)|m(t)+1)A^{-1}(m(t)+1)(\tilde{x}_{ML}(\alpha t|\alpha t) + \tilde{x}_{ML}(\beta t|\beta t)) \\ &- P_{ML}(m(t)|m(t))P_{ML}^{-1}(m(t)|m(t)+1)G(m(t)+1)(w(\alpha t) + w(\beta t)) \\ &- K_{ML}(m(t))v(t)\end{aligned} \tag{3.4.60}$$

## 3.5 Up-down Smoothing Algorithm on the Tree Using Hamiltonian Triangularization

In this section we discuss an alternative derivation of the ML up-down smoothing algorithm. The derivation is based on triangularizing the Hamiltonian matrix for the smoothing problem. This approach not lends additional insight into the smoothing process. In particular as we will see in our derivation the structure of the Hamiltonian dynamics is *non-square*; thus, this leads to triagularizing these dynamics and not diagonalizing them. This approach also suggests an alternative way of computing the Bayesian smoother. In particular the incorporation of the prior variance, i.e. the variance of the top node of the tree, is done at the end of the upward sweep rather than being carried around in the dynamics of the upward sweep as is done in our previously derived smoother. Note that while we derive our equations for scale-invariant system parameters, the extensions to the scale-varying case as well as the general case are trivial.

Consider the following scale-invariant model defined with downward dynamics and its associated measurement equation for an $M$-level tree.

$$\begin{aligned} x(t) &= Ax(\gamma^{-1}t) + Bw(t) && (3.5.1) \\ y(t) &= Cx(t) + v(t) && (3.5.2) \end{aligned}$$

where $w$ and $v$ are white-noise processes with variances $Q$ and $R$ respectively. The Hamiltonian function for the smoothing problem is as follows.

$$\begin{aligned} H(x,\lambda) &= \sum_t \frac{1}{2}(y(t) - Cx(t))^T R^{-1}(y(t) - Cx(t)) + \sum_{t\neq t_0} \frac{1}{2} w^T(t)Q^{-1}w(t) && (3.5.3) \\ &+ \frac{1}{2}(x(t_0) - \overline{x}(t_0))^T P_0^{-1}(x(t_0) - \overline{x}(t_0)) + \sum_{t\neq t_0} \lambda_t^T(x(t) - Ax(\gamma^{-1}t) - Bw(t)) \end{aligned}$$

The optimal estimates of the state $x$, the noise $w$, and the Lagrange multiplier $\lambda$ are obtained by solving the following equations for the minimum of $H$.

$$\begin{aligned} 0 &= C^T R^{-1} C\hat{x}(t) - C^T R^{-1} y(t) - A^T(\hat{\lambda}(\alpha t) + \hat{\lambda}(\beta t)) + \hat{\lambda}(t) && (3.5.4) \\ 0 &= Q^{-1}\hat{w}(t) - B^T\hat{\lambda}(t) && (3.5.5) \\ 0 &= \hat{x}(t) - A\hat{x}(\gamma^{-1}t) - B\hat{w}(t) && (3.5.6) \end{aligned}$$

for all $t \neq t_0$ while for $t = t_0$ we have

$$\begin{aligned} 0 &= C^T R^{-1} C \hat{x}(t_0) - C^T R^{-1} y(t_0) \\ &- A^T(\lambda(\alpha t_0) + \lambda(\beta t_0)) + P_0^{-1}\hat{x}(t_0) - P_0^{-1}\bar{x}(t_0) \end{aligned} \tag{3.5.7}$$

By substituting eq.(3.5.5) into eq.(3.5.6) and reindexing the equations, we get the following set of equations solely in terms of $x$ and $\lambda$.

$$0 = C^T R^{-1} C \hat{x}(t) - C^T R^{-1} y(t) - A^T(\hat{\lambda}(\alpha t) + \hat{\lambda}(\beta t)) + \hat{\lambda}(t) \tag{3.5.8}$$

$$0 = \hat{x}(\alpha t) - A\hat{x}(t) - BQB^T \hat{\lambda}(\alpha t) \tag{3.5.9}$$

$$0 = \hat{x}(\beta t) - A\hat{x}(t) - BQB^T \hat{\lambda}(\beta t) \tag{3.5.10}$$

for all $t$ such that $t \neq t_0$ and $m(t) \neq M$ while for $t = t_0$ we have

$$\begin{aligned} 0 &= C^T R^{-1} C \hat{x}(t_0) - C^T R^{-1} y(t_0) \\ &- A^T(\lambda(\alpha t_0) + \lambda(\beta t_0)) + P_0^{-1}\hat{x}(t_0) - P_0^{-1}\bar{x}(t_0) \end{aligned} \tag{3.5.11}$$

$$0 = \hat{x}(\alpha t_0) - A\hat{x}(t_0) - BQB^T \hat{\lambda}(\alpha t_0) \tag{3.5.12}$$

$$0 = \hat{x}(\beta t_0) - A\hat{x}(t_0) - BQB^T \hat{\lambda}(\beta t_0) \tag{3.5.13}$$

and for $m(t) = M$ we have

$$0 = C^T R^{-1} C \hat{x}(t) - C^T R^{-1} y(t) + \hat{\lambda}(t) \tag{3.5.14}$$

By rewriting eq.'s(3.5.8-3.5.10) in matrix form we get the following Hamiltonian dynamics in terms of the points $t$, $\alpha t$, and $\beta t$ for $t \neq t_0$.

$$\mathcal{A} \begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix}_t + \Theta_\alpha \begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix}_{\alpha t} + \Theta_\beta \begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix}_{\beta t} = \begin{bmatrix} 0 \\ 0 \\ C^T R^{-1} y(t) \end{bmatrix} \tag{3.5.15}$$

where

$$\mathcal{A} = \begin{bmatrix} -A & 0 \\ -A & 0 \\ C^T R^{-1} C & I \end{bmatrix} \tag{3.5.16}$$

$$\Theta_\alpha = \begin{bmatrix} I & -BQB^T \\ 0 & 0 \\ 0 & -A^T \end{bmatrix} \tag{3.5.17}$$

$$\Theta_\beta = \begin{bmatrix} 0 & 0 \\ I & -BQB^T \\ 0 & -A^T \end{bmatrix} \tag{3.5.18}$$

We now consider the problem of deriving an up-down algorithm for solving for the smoothed estimate $\hat{x}$. Our approach to the problem is to triangularize the dynamics defined by eq.(3.5.15), thus yielding an algorithmic structure in which an upward recursion is followed by a downward recursion to produce $\hat{x}$. Consider the following time-varying transformation.

$$\begin{bmatrix} x^u \\ \hat{x} \end{bmatrix}_t = T_t \begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix}_t \tag{3.5.19}$$

where

$$T_t = \begin{bmatrix} \Gamma_t & I \\ I & 0 \end{bmatrix} \tag{3.5.20}$$

With respect to the transformed variables $x^u$ and $\hat{x}$ we now wish to transform the dynamics of eq.(3.5.15) into a form in which there is an upward recursion for $x^u$ decoupled from $\hat{x}$. Note that we are free to multiply eq.(3.5.15) on the left by an invertible matrix, $S_t$, without losing information. By doing so, we wish to transform the dynamics for $x^u$ and $\hat{x}$, eq.(3.5.15), into the following structure.

$$S_t \mathcal{A} T_t^{-1} \begin{bmatrix} x^u \\ \hat{x} \end{bmatrix}_t + S_t \Theta_\alpha T_{t+}^{-1} \begin{bmatrix} x^u \\ \hat{x} \end{bmatrix}_{\alpha t} + S_t \Theta_\beta T_{t+}^{-1} \begin{bmatrix} x^u \\ \hat{x} \end{bmatrix}_{\beta t} = \begin{bmatrix} C^T R^{-1} y(t) \\ 0 \\ 0 \end{bmatrix} \tag{3.5.21}$$

where

$$S_t = \begin{bmatrix} -P_t^{-1} A^{-1} & -P_t^{-1} A^{-1} & I \\ 0 & I & 0 \\ I & 0 & 0 \end{bmatrix} \tag{3.5.22}$$

$$S_t A T_t^{-1} = \begin{bmatrix} I & 0 \\ L_1 & L_2 \\ L_3 & L_4 \end{bmatrix} \tag{3.5.23}$$

$$S_t \Theta_\alpha T_{t+}^{-1} = \begin{bmatrix} F_{t+} & 0 \\ 0 & 0 \\ N & AP_tA^T\Gamma_{t+} \end{bmatrix} \tag{3.5.24}$$

$$S_t \Theta_\beta T_{t+}^{-1} = \begin{bmatrix} F_{t+} & 0 \\ N & AP_tA^T\Gamma_{t+} \\ 0 & 0 \end{bmatrix} \tag{3.5.25}$$

We now solve for the matrices $L_1 - L_4$, $F_{t+}$, and $N$ given eq.'s(3.5.22-3.5.25). It is straightforward to show that eq.'s(3.5.24,3.5.25) are actually equivalent. Thus, eq.'s(3.5.23,3.5.24) represent 12 constraint equations(9 of which are non-trivial) for our 6 unknowns. These 9 equations are explicitly as follows.

$$\Gamma_t = 2P_t^{-1} + C^TR^{-1}C \tag{3.5.26}$$

$$L_1 = 0 \tag{3.5.27}$$

$$L_3 = 0 \tag{3.5.28}$$

$$-A = L_1\Gamma_{t+} + L_2 \tag{3.5.29}$$

$$-A = L_3\Gamma_{t+} + L_4 \tag{3.5.30}$$

$$F_{t+}\Gamma_{t+} = -P_t^{-1}A^{-1} \tag{3.5.31}$$

$$F_{t+} = P_t^{-1}A^{-1}BQB^T - A^T \tag{3.5.32}$$

$$I = N\Gamma_{t+} + AP_tA^T\Gamma_{t+} \tag{3.5.33}$$

$$M = -BQB^T \tag{3.5.34}$$

It follows easily that

$$L_2 = -A \tag{3.5.35}$$

$$L_4 = -A \tag{3.5.36}$$

Thus, we can write the up-down algorithm as an upward recursion for $x^u$ followed by a downward correction sweep involving $\hat{x}$ and $x^u$.

For all $t$ such that $t \neq t_0$ and $m(t) \neq M$ the upward recursion is as follows.

$$x^u(t) = \mathcal{F}_{t+}(x^u(\alpha t) + x^u(\beta t)) + C^T R^{-1} y(t) \quad (3.5.37)$$

$$\mathcal{F}_{t+} = -F_{t+} = P_t^{-1} A^{-1} \Gamma_{t+}^{-1} \quad (3.5.38)$$

For $t = t_0$ we have

$$x^u(t_0) = \mathcal{F}_{t_0+}(x^u(\alpha t_0) + x^u(\beta t_0)) + C^T R^{-1} y(t_0) + P_0^{-1}\overline{x}(t_0) \quad (3.5.39)$$

where

$$\mathcal{F}_{t_0+} = P_{t_0}^{-1} A^{-1} \Gamma_{t_0+}^{-1} \quad (3.5.40)$$

$$\Gamma_{t_0+}^{-1} = 2P_{t_0}^{-1} + C^T R^{-1} C + P_0^{-1} \quad (3.5.41)$$

Note that we've incorporated our a priori variance $P_0$ at the end of the upward sweep.

The initial condition for the upward recursion, eq.(3.5.37), is determined by considering our constraints for all $t$ such that $m(t) = M$, namely eq.(3.5.14). We must now define $x^u$ along the bottom level to provide our desired initial conditions. Consider the following definition for all $s$ such that $m(s) = M$.

$$x^u(s) = \Gamma_M \hat{x}(s) + \hat{\lambda}(s) \quad (3.5.42)$$

$$\Gamma_M = C^T R^{-1} C \quad (3.5.43)$$

By substituting eq.(3.5.42) into eq.(3.5.14) we get our desired initial condition for $x^u$.

$$x^u(s) = C^T R^{-1} y(s) \quad (3.5.44)$$

Note that the system matrix for our upward recursion, $\mathcal{F}_{t+}$, can be written solely as a function of $P_t^{-1}$. But we can also write a recursion for $P_t^{-1}$. From eq.'s(3.5.26,3.5.31,3.5.32) we get the following.

$$P_t^{-1} = (A^{-1}(2P_{t+}^{-1} + C^T R^{-1} C)^{-1} A^{-T} + A^{-1} B Q B^T A^{-T})^{-1} \quad (3.5.45)$$

where for all $s$ such that $m(s) = M$

$$P_s^{-1} = 0 \quad (3.5.46)$$

Note that if we let

$$P_t^{-1} = P^{-1}(m(t)+1|m(t)) \tag{3.5.47}$$

then eq.(3.5.45) is precisely the Riccati equation for the Maximum-Likelihood estimator in information form.

We now write down the downward recursion for the smoothed estimate $\hat{x}(t)$. For all $t$ such that $t \neq t_0$ the downward recurion is as follows.

$$\hat{x}(t) = \mathcal{F}_t^T \hat{x}(\gamma^{-1}t) + \mathcal{F}_t^T A^{-1} B Q B^T x^u(t) \tag{3.5.48}$$

while for $t = t_0$ we have

$$\hat{x}(t_0) = \Gamma_{t_0+}^{-1} x^u(t_0) \tag{3.5.49}$$

We now summarize the overall two-sweep algorithm for the case of general parameters $A(t)$, $B(t)$, $Q(t)$, and $R(t)$:

**Upward Sweep**

For all $t$ such that $t \neq t_0$ and $m(t) \neq M$:

$$\begin{aligned}
\hat{x}(\gamma^{-1}t|t) &= (S(\alpha t)\hat{x}(t|\alpha t) + S(\beta t)\hat{x}(t|\beta t)) + C^T(t)R^{-1}(t)y(t) && (3.5.50)\\
S(t) &= P(\gamma^{-1}t|t)A^{-1}(t)P(t|t) && (3.5.51)\\
P(t|t) &= P^{-1}(\alpha t|t) + P^{-1}(\beta t|t) + C^T(t)R^{-1}(t)C(t) && (3.5.52)\\
P(\gamma^{-1}t|t) &= A^{-1}(t)P(t|t)A^{-T}(t) + A^{-1}(t)B(t)Q(t)B^T(t)A^{-T}(t) && (3.5.53)
\end{aligned}$$

Initial Conditions:

$$\begin{aligned}
\hat{x}(\gamma^{-1}t|t) &= 0 && (3.5.54)\\
P^{-1}(\gamma^{-1}t|t) &= 0 && (3.5.55)
\end{aligned}$$

for all $t$ such that $m(\gamma^{-1}t) = M$.

**Incorporation of Prior at Top Node**

$$\begin{aligned}
\hat{x}_s(t_0|t_0) &= P^{-1}(t_0|t_0)\hat{x}(t_0|t_o) && (3.5.56)\\
P^{-1}(t_0|t_0) &= P^{-1}(t_0|\alpha t_0) + P^{-1}(t_0|\beta t_0) + C^T(t_0)R^{-1}(t_0)C(t_0) + P_x^{-1}(t_0) && (3.5.57)
\end{aligned}$$

**Downward Sweep**

$$\begin{aligned}
\hat{x}_s(t) &= S^T(t)\hat{x}_s(\gamma^{-1}t) + S^T(t)A^{-1}(t)B(t)Q(t)B^T(t)\hat{x}(\gamma^{-1}t|t) && (3.5.58)\\
P_s(t) &= P(t|t) + J(t)[P_s(\gamma^{-1}t) - P(\gamma^{-1}t|t)]J^T(t) && (3.5.59)\\
J(t) &= P(t|t)A^{-T}(t)P^{-1}(\gamma^{-1}t|t) && (3.5.60)
\end{aligned}$$

## 3.6 Reachability, Observability, and Reconstructibility

In this section we develop certain system theoretic constructs which are useful in analyzing both the stability and the steady-state characteristics of our filter. In particular we define notions of reachability, observability, and reconstructibility on dyadic trees in terms of system dynamics going up the tree. At the end of this section we describe a dual theory for systems with downward dynamics. Note that for the remainder of this chapter we focus on the scale-varying case, i.e. the model parameters depend on scale only.

### 3.6.1 Upward Reachability

We begin with the notion of reachability for a system defined going up a tree. Analogous to the standard time-series case, reachability involves the notion of being able to reach arbitrary states at some point $t$ on the tree given arbitrary inputs in the past where in the case of processes evolving up a tree the past refers to points in the subtree under $t$. Recall that we can rewrite the dynamics for our backward process up the tree, eq.(3.2.5), in the following form.

$$x(t) = \frac{1}{2}F(m(t)+1)[x(\alpha t) + x(\beta t)] + \frac{1}{2}G(m(t)+1)[\tilde{w}(\alpha t) + \tilde{w}(\beta t)] \qquad (3.6.1)$$

Also, recall that in our backward model $\tilde{w}(t)$ is a white noise process along upward paths on the tree. For the analysis of reachability, however, we simply view $\tilde{w}(t)$ as the input to the system eq.(3.6.1).

We define the following vectors,

$$X_{M,t_0} \triangleq [x^T(\alpha^M t_0), x^T(\beta\alpha^{M-1}t_0), ...x^T(\beta^M t_0)]^T \tag{3.6.2}$$

$$\tilde{W}_{M,t_0} \triangleq [\ \tilde{w}^T(\alpha t_0) \quad \tilde{w}^T(\beta t_0) \quad ... \quad \tilde{w}^T(\alpha^M t_0) ... \tilde{w}^T(\beta^M t_0)\ ]^T \tag{3.6.3}$$

which have the following interpretation. Consider an arbitrary point on the tree, $t_0$. The vector $X_{M,t_0}$ denotes the vector of $2^M$ points at the $M$th level down in the subtree under $t_0$; i.e. $X_{M,t_0}$ includes all of the nodes at this level that influence the value of $x(t_0)$. The vector $\tilde{W}_{M,t_0}$ comprises the full set of inputs that influences $x(t_0)$ starting from initial condition $X_{M,t_0}$, i.e. the $\tilde{w}(t)$ in the entire subtree down to $M$ levels from $t_0$. We define **upward reachability** to be the following.

**Definition 3.6.1** *The system is* **upward reachable** *from $X_{M,t_0}$ to $x(t_0)$ if given any $\overline{X}_{M,t_0}$ and any desired $\overline{x}(t_0)$, it is possible to specify $\tilde{W}_{M,t_0}$ so if $X_{M,t_0} = \overline{X}_{M,t_0}$, then $x(t_0) = \overline{x}(t_0)$.*

In studying conditions for reachability since we are given $X_{M,t_0}$, we can set it equal to zero without loss of generality. Note that if $X_{M,t_0} = 0$, then we have

$$x(t_0) = \mathcal{G}\tilde{W}_{M,t_0} \tag{3.6.4}$$

where

$$\mathcal{G} \triangleq [\ \Psi(0) \quad \Psi(0) \quad \Psi(1) \quad \Psi(1) \quad \Psi(1) \quad \Psi(1) \quad ... \tag{3.6.5}$$

$$\underbrace{\Psi(M-2)...\Psi(M-2)}_{2^{M-1}\text{ times}} \quad \underbrace{\Psi(M-1)...\Psi(M-1)}_{2^{M}\text{ times}}\ ]$$

$$\Psi(i) \triangleq (\frac{1}{2})^{i+1}\phi(m(t_0), m(t_0)+i)G(m(t_0)+i+1) \tag{3.6.6}$$

$$\phi(m_1, m_2) \triangleq \begin{cases} I & m_1 = m_2 \\ F(m_1+1)\phi(m_1+1, m_2) & m_1 < m_2 \end{cases} \tag{3.6.7}$$

$$\phi(m-1, m) \triangleq F(m) \tag{3.6.8}$$

Let us also define the following quantity.

**Definition 3.6.2 Upward-reachability Grammian**

$$\begin{aligned}\mathcal{R}(t_0, M) &\triangleq \mathcal{G}\mathcal{G}^T \\ &= \sum_{i=0}^{M-1} 2^{-i-1}\phi(m(t_0), m(t_0)+i)G(m(t_0)+i+1) \\ &\times\ G^T(m(t_0)+i+1)\phi^T(m(t_0), m(t_0)+i) \qquad (3.6.9)\end{aligned}$$

From eq.(3.6.4) we see that the ability to reach all possible values of $x(t_0)$ given arbitrary inputs, $\tilde{W}_{M,t_0}$, depends on the rank of the matrix $\mathcal{G}$. This, along with the fact that the rank of $\mathcal{G}$ equals the rank of $\mathcal{G}\mathcal{G}^T$, leads to the following, where $x(t)$ is an $n$-dimensional vector:

**Proposition 3.6.1** *The system is* **upward reachable** *from $X_{M,t_0}$ to $x(t_0)$ iff $\mathcal{G}$ has rank $n$ iff $\mathcal{R}(t_0, M)$ has rank $n$.*

Note that $\mathcal{R}(t_0, M)$ bears a strong similarity to the standard reachability grammian for the following system.

$$x(m) = \frac{1}{2}F(m+1)x(m+1) + \frac{1}{2}G(m+1)u(m+1) \qquad (3.6.10)$$

where the reachability grammian in this case is

$$\begin{aligned}\mathcal{R}^*(m, m+M) &\triangleq \sum_{i=0}^{M-1} 2^{-2i-2}\phi(m, m+i)G(m+i+1) \\ &\times\ G^T(m+i+1)\phi^T(m, m+i) \\ &= \mathcal{G}^*(\mathcal{G}^*)^T \\ \mathcal{G}^* &\triangleq [\ \Psi(0)\quad \Psi(1)\quad ...\quad \Psi(M-2)\quad \Psi(M-1)\ ] \qquad (3.6.11)\end{aligned}$$

In fact it is evident from the definitions in eq.'s(3.6.5,3.6.11) that the rank of $\mathcal{G}$ is equivalent to the rank of $\mathcal{G}^*$. This leads to the following corollary.

**Corollary 3.6.1** *The system is* **upward reachable** *from $X_{M,t_0}$ to $x(t_0)$ iff for any $\alpha, \beta \neq 0$ $\mathcal{R}^*_{\alpha,\beta}(m(t_0), m(t_0)+M)$ has rank $n$, where $\mathcal{R}^*_{\alpha,\beta}(m(t_0), m(t_0)+M)$ is the reachability grammian for the system*

$$x(m) = \alpha F(m+1)x(m+1) + \beta G(m+1)u(m+1) \qquad (3.6.12)$$

Note that if $F$ and $G$ are constant in eq.(3.6.1), then reachability is equivalent to the usual condition, i.e. $rank[G|FG|...|F^{M-1}G] = n$.

### 3.6.2 Upward Observability and Reconstructibility

We develop the notion of observability and the notion of reconstructibility on trees. Defined on trees, observability corresponds to the notion of being able to uniquely determine the points at the bottom of a subtree, i.e. the "initial conditions", given knowledge of the inputs and observations in the subtree. It is also useful to develop the weaker notion corresponding to being able to uniquely determine the single point at the top of a subtree given knowledge of the inputs and observations in the subtree. This notion is analogous to reconstructibility for standard systems; thus, we adopt the same term for the notion on trees.

Let us define

$$Y_{M,t_0} \triangleq [\ y^T(t_0)|\ \ y^T(\alpha t_0),\ \ y^T(\beta t_0)|\ \ \ldots\ \ |y^T(\alpha^M t_0), \ldots y^T(\beta^M t_0)\ ]^T \tag{3.6.13}$$

where

$$y(t) = C(m(t))x(t) \tag{3.6.14}$$

**Definition 3.6.3** *The system is* **upward observable** *from $X_{M,t_0}$ to $x(t_0)$ if given knowledge of $\tilde{W}_{M,t_0}$ and $Y_{M,t_0}$, we can uniquely determine $X_{M,t_0}$.*

Note that if $\tilde{W}_{M,t_0} = 0$ then

$$Y_{M,t_0} = \mathcal{H}_M X_{M,t_0} \tag{3.6.15}$$

where $\mathcal{H}_M$ is most easily visualized if we partition it compatibly with the levels of the observations in $Y_{M,t_0}$:

$$\mathcal{H}_M \triangleq \left[ \overbrace{\begin{array}{cccccccccccc}
\Theta(0) & \Theta(0) & \dots & & & & & & & & \dots & \Theta(0) \\
\hdashline
\Theta(1) & \dots & & & \dots & \Theta(1) & 0 & \dots & & & \dots & 0 \\
0 & \dots & & & \dots & 0 & \Theta(1) & \dots & & & \dots & \Theta(1) \\
\hdashline
\Theta(2) & \dots & \Theta(2) & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\
0 & \dots & 0 & \Theta(2) & \dots & \Theta(2) & 0 & \dots & 0 & 0 & \dots & 0 \\
0 & \dots & 0 & 0 & \dots & 0 & \Theta(2) & \dots & \Theta(2) & 0 & \dots & 0 \\
0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & \Theta(2) & \dots & \Theta(2) \\
\hdashline
 & & & & \vdots & & & \vdots & & & & \\
 & & & & \vdots & & & \vdots & & & & \\
\hdashline
\Theta(M) & 0 & \dots & & & & & & & & \dots & 0 \\
0 & \Theta(M) & \dots & & & & & & & & \dots & 0 \\
\vdots & & & & & & & & & & & \vdots \\
0 & 0 & \dots & & & & & & & & \dots & \Theta(M)
\end{array}}^{2^M \text{ blocks}} \right] \tag{3.6.16}$$

Here

$$\Theta(i) \triangleq (\frac{1}{2})^{M-i} C(m(t_0)+i)\phi(m(t_0)+i, m(t_0)+M) \tag{3.6.17}$$

As a simple example to help clarify the structure of the matrix $\mathcal{H}_M$ consider the matrix $\mathcal{H}_2$ for the scale-invariant case, i.e. where $F(m) = F$, $C(m) = C$.

$$\mathcal{H}_2 = \begin{bmatrix} \frac{1}{4}CF^2 & \frac{1}{4}CF^2 & \frac{1}{4}CF^2 & \frac{1}{4}CF^2 \\ \frac{1}{2}CF & \frac{1}{2}CF & 0 & 0 \\ 0 & 0 & \frac{1}{2}CF & \frac{1}{2}CF \\ C & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \end{bmatrix} \tag{3.6.18}$$

That is, at level $i$, there are $2^i$ measurements each of which provides information about the **sum** of a block of $2^{M-i}$ components of $X_{M,t_0}$. Note that this makes clear that upward observability is indeed a very strong condition. Specifically, since successively larger blocks of $X_{M,t_0}$ are summed as we move up the tree, subsequent measurements provide **no** information about the differences among the values that have been summed. For example consider $M = 1$. In this case $y(t)$ contains information about the sum $x(\alpha t) + x(\beta t)$, and thus information about $x(\alpha t) - x(\beta t)$ must come from $y(\alpha t)$ and $y(\beta t)$. This places severe constraints on the system matrices. In particular a necessary condition for observability is that $y$ have dimension larger than $\frac{n}{2}$(otherwise $\mathcal{H}_M$ has fewer rows than columns).

We also define the following.

**Definition 3.6.4 Upward-observability Grammian**

$$\mathcal{M}_M \triangleq \mathcal{H}_M^T \mathcal{H}_M \tag{3.6.19}$$

*where*

$$\mathcal{M}_k = U(k,0) \tag{3.6.20}$$

$$U(k,k) \triangleq \sum_{i=0}^{k} (\frac{1}{2})^{2(k-i)} \phi^T(m(t_0)+i, m(t_0)+k) \mathcal{C}(m(t_0)+i) \phi(m(t_0)+i, m(t_0)+k) \tag{3.6.21}$$

$$\mathcal{C}(k) \triangleq C^T(k)C(k) \tag{3.6.22}$$

$$U(k,l) \triangleq \begin{bmatrix} U(k,l+1) & S(k,l) \\ S(k,l) & U(k,l+1) \end{bmatrix} \tag{3.6.23}$$

*and $S(k,l)$ is a block matrix with $2^{k-l-1} \times 2^{k-l-1}$ blocks each of which equals*

$$T(k,l) = \sum_{i=0}^{l}(\frac{1}{2})^{2(k-i)}\phi^T(m(t_0)+i,m(t_0)+k)C^T(m(t_0)+i)C(m(t_0)+i)\phi(m(t_0)+i,m(t_0)+k) \tag{3.6.24}$$

Once again we consider the scale-invariant case, this time in order to make explicit the structure of the matrix $\mathcal{M}_M$. The following is $\mathcal{M}_2$ for the scale-invariant case.

$$\mathcal{M}_2 = \begin{bmatrix} M_1 & M_2 & M_3 & M_3 \\ M_2 & M_1 & M_3 & M_3 \\ M_3 & M_3 & M_1 & M_2 \\ M_3 & M_3 & M_2 & M_1 \end{bmatrix} \tag{3.6.25}$$

where

$$M_1 = \frac{1}{16}F^{2T}C^TCF^2 + \frac{1}{4}FC^TCF + C^TC \tag{3.6.26}$$

$$M_2 = \frac{1}{16}F^{2T}C^TCF^2 + \frac{1}{4}FC^TCF \tag{3.6.27}$$

$$M_3 = \frac{1}{16}F^{2T}C^TCF^2 \tag{3.6.28}$$

From eq.(3.6.15) we see that being able to uniquely determine $X_{M,t_0}$ from $Y_{M,t_0}$ is equivalent to requiring the nullspace of the matrix $\mathcal{H}_M$ to be $\{0\}$. This leads to the following.

**Proposition 3.6.2** *The system is* **upward observable** *from $X_{M,t_0}$ to $x(t_0)$ iff $\mathcal{N}(\mathcal{H}_M) = \{0\}$ iff $\mathcal{M}_M$ is invertible.*

A much weaker notion than that of observability is the notion of reconstructibility. Reconstructibility requires only the ability to determine the single point at the top of a subtree given knowledge of the inputs and observations in the subtree.

**Definition 3.6.5** *The system is* **upward reconstructible** *from $X_{M,t_0}$ to $x(t_0)$ if given knowledge of $\tilde{W}_{M,t_0}$ and $Y_{M,t_0}$, we can uniquely determine $x(t_0)$.*

We also define the following.

**Definition 3.6.6 Upward-reconstructibility Grammian**

$$\begin{aligned}\mathcal{O}(t_0, M) &= I_M \mathcal{H}_M^T \mathcal{H}_M I_M^T \\ &= \sum_{i=0}^{M} 2^i \phi^T(m(t_0)+i, m(t_0)+M) C^T(m(t_0)+i) \\ &\times \; C(m(t_0)+i)\phi(m(t_0)+i, m(t_0)+M) \end{aligned} \quad (3.6.29)$$

*where*

$$I_M = \underbrace{[I|I|...|I]}_{2^M \text{ times}} \quad (3.6.30)$$

*and each $I$ is an $n \times n$ identity matrix.*

Note that if $\tilde{W}_{M,t_0} = 0$, then

$$x(t_0) = \Phi(t_0) X_{M,t_0} \quad (3.6.31)$$

where

$$\Phi(t_0) \triangleq (\frac{1}{2})^M \phi(m(t_0), m(t_0)+M) I_M \quad (3.6.32)$$

Since the condition of reconstructibility only requires being able to uniquely determine the single point $x(t_0)$ from the measurements in the subtree, we guarantee this condition by requiring that any vector in the nullspace, $\mathcal{N}(\mathcal{H}_M)$, is also in the nullspace, $\mathcal{N}(\Phi(t_0))$. We thus have the following theorem, the proof of which can be found in the Appendix 3B.

**Theorem 3.6.1** *The system is* **upward reconstructible** *iff $\mathcal{N}(\mathcal{H}) \subseteq \mathcal{N}(\Phi(t_0))$. If $F(m)$ is invertible for all $m$, this is equivalent to the invertibility of $\mathcal{O}(t_0, M)$.*

Note that $\mathcal{O}(t_0, M)$ bears a strong similarity to the standard **observability** grammian for the following system.

$$x(m) = \alpha F(m+1)x(m+1) + G(m+1)u(m+1) \quad (3.6.33)$$

$$y(m) = \beta C(m) x(m) \quad (3.6.34)$$

where the observability grammian in this case is

$$\begin{aligned}\mathcal{O}_{\alpha,\beta}(m(t_0), m(t_0)+M) &\triangleq \sum_{i=0}^{M} \alpha^{2(M-i)} \beta^2 \phi^T(m(t_0)+i, m(t_0)+M) C^T(m(t_0)+i) \\ &\times \; C(m(t_0)+i)\phi(m(t_0)+i, m(t_0)+M) \end{aligned} \quad (3.6.35)$$

This results in the following corollary.

**Corollary 3.6.2** *Assuming that $F(m)$ is invertible for all $m$, the system is* **upward reconstructible** *from $X_{M,t_0}$ to $x(t_0)$ iff $\mathcal{O}_{\alpha,\beta}(m(t_0), m(t_0)+M)$ has rank $n$.*

Let us comment on an important difference between the concepts of reconstructibility and observability for our upward propagating systems and those for standard temporal systems. For standard systems observability implies reconstructibility and the two concepts are equivalent if the state transition matrix is invertible. In our case, observability certainly implies reconstructibility, but the former remains a much stronger condition even if $\phi$ is invertible. In this case reconstructibility is equivalent to being able to determine the **average values** of the components of the initial state. We can define this notion formally as follows.

**Definition 3.6.7** *The system is* **upward coarsely observable** *from $X_{M,t_0}$ to $x(t_0)$ if given knowledge of $\tilde{W}_{M,t_0}$ and $Y_{M,t_0}$, we can uniquely determine $ave(X_{M,t_0})$ where*

$$ave(X_{M,t_0}) \triangleq (\frac{1}{2})^M I_M X_{M,t_0} \tag{3.6.36}$$

Note that if $\phi$ is invertible from eq.'s(3.6.31 and 3.6.32) we get

$$\begin{aligned} ave(X_{M,t_0}) &= (\frac{1}{2})^M I_M X_{M,t_0} \\ &= \phi^{-1}(m(t_0), m(t_0)+M)x(t_0) \end{aligned} \tag{3.6.37}$$

This leads to the following corollary.

**Corollary 3.6.3** *Assuming that $F(m)$ is invertible for all $m$, the system is* **upward reconstructible** *from $X_{M,t_0}$ to $x(t_0)$ iff the system is* **upward coarsely observable** *from $X_{M,t_0}$ to $x(t_0)$.*

Note that in contrast to our notion of observability our reachability concept going up the tree is actually rather weak since we have **many** control inputs in the subtree to achieve a **single** final state $x(t_0)$. As one might expect there is a dual theory for systems defined moving down the tree, but the tree asymmetry leads to some important differences. In particular, weak and strong concepts are interchanged.

For example, observability is concerned with determining the **single** initial state given observations in the subtree under $t_0$, while reconstructibility corresponds to determining the **entire** vector $X_{M,t_0}$. In this case if $\phi$ is invertible observability is equivalent to determining the **average value** of $X_{M,t_0}$. Similarly, reachability is concerned with reaching arbitrary values for the entire vector $X_{M,t_0}$, an extremely strong condition. A natural and much weaker condition is achieving an arbitrary average value for $X_{M,t_0}$. A complete picture of this system theory is be given in the next section.

### 3.6.3 Downward Systems

In this section we describe a dual theory for systems defined moving down the tree. Consider the following system described by downward dynamics.

$$x(t) = A(m(t))x(\gamma^{-1}t) + B(m(t))w(t) \tag{3.6.38}$$

with measurements

$$y(t) = C(m(t))x(t) \tag{3.6.39}$$

We define the following vector,

$$W_{M,t_0} \triangleq [\ w^T(t_0) \ \ |w^T(\alpha t_0) \ \ w^T(\beta t_0)| \ \ \ldots \ \ |w^T(\alpha^M t_0) \ldots w^T(\beta^M t_0)\ ]^T \tag{3.6.40}$$

where $W_{M,t_0}$ represents the the full set of inputs that influences $X_{M,t_0}$ (as previously defined) starting from the initial condition $x(\gamma^{-1}t_0)$, i.e. the $w(t)$ in the entire subtree down to $M$ levels from $t_0$, **including** the point $w(t_0)$. The reason for defining the initial condition as being $x(\gamma^{-1}t_0)$ rather than $x(t_0)$ is to insure a duality between the downward and the upward concepts. The necessity for the asymmetry in upward and downward definitions will become more clear as we begin to establish this duality.

Let us begin with the notion of observability for downward systems. This is the notion of determining uniquely the initial state, which in this case is $x(t_0)$, from the observations in the subtree down to $M$ levels from $t_0$, but **not including** the point $y(t_0)$. Again this asymmetry in upward and downward definitions is necessary for establishing a duality between upward and downward concepts. We define the following vector of observations,

$$\tilde{Y}_{M,t_0} \triangleq [\ y^T(\alpha t_0),\ \ y^T(\beta t_0)|\ \ \ldots\ \ |y^T(\alpha^M t_0), \ldots y^T(\beta^M t_0)\ ]^T \tag{3.6.41}$$

and also the following vector of inputs

$$W'_{M,t_0} \triangleq [\ w^T(\alpha t_0)\ \ w^T(\beta t_0)|\ \ \ldots\ \ |w^T(\alpha^M t_0) \ldots w^T(\beta^M t_0)\ ]^T \tag{3.6.42}$$

where $W'_{M,t_0}$ is simply $W_{M,t_0}$ without the input $w(t_0)$.

**Definition 3.6.8** *The system is* **downward observable** *from $x(t_0)$ to $X_{M,t_0}$ if given knowledge of $W'_{M,t_0}$ and $\tilde{Y}_{M,t_0}$, we can uniquely determine $x(t_0)$.*

Note that if $W'_{M,t_0} = 0$ then

$$\tilde{Y}_{M,t_0} = \tilde{\mathcal{H}}_M x(t_0) \tag{3.6.43}$$

where

$$\begin{aligned} \tilde{\mathcal{H}}_M^T &\triangleq [\ \tilde{\Psi}(0)\ \ \tilde{\Psi}(0)\ \ \tilde{\Psi}(1)\ \ \tilde{\Psi}(1)\ \ \tilde{\Psi}(1)\ \ \tilde{\Psi}(1)\ \ \ldots \\ &\qquad \underbrace{\tilde{\Psi}(M-2)\ldots\tilde{\Psi}(M-2)}_{2^{M-1}\text{ times}}\ \ \underbrace{\tilde{\Psi}(M-1)\ldots\tilde{\Psi}(M-1)}_{2^{M}\text{ times}}\ ] \end{aligned} \tag{3.6.44}$$

$$\tilde{\Psi}^T(i) \triangleq C(m(t_0)+i+1)\phi(m(t_0+i), m(t_0)) \tag{3.6.45}$$

$$\phi(m_1, m_2) \triangleq \begin{cases} I & m_1 = m_2 \\ A(m_1-1)\phi(m_1-1, m_2) & m_1 > m_2 \end{cases} \tag{3.6.46}$$

$$\phi(m+1, m) \triangleq A(m) \tag{3.6.47}$$

We also define the following.

**Definition 3.6.9 Downward-observability Grammian**

$$\tilde{\mathcal{M}}_M \triangleq \tilde{\mathcal{H}}_M^T \tilde{\mathcal{H}}_M \tag{3.6.48}$$

Since being able to uniquely determine $x(t_0)$ from $\tilde{Y}_{M,t_0}$ is equivalent to requiring the nullspace of $\tilde{\mathcal{H}}_M$ to be $\{0\}$, we get

**Proposition 3.6.3** *The system is* **downward observable** *from $x(t_0)$ to $X_{M,t_0}$ iff $\mathcal{N}(\tilde{\mathcal{H}}_M) = \{0\}$ iff $\tilde{\mathcal{M}}_M$ is invertible.*

Similarly to the case of observability and reconstructibility for upward systems we would expect the invertibility of $\phi$ to suggest an equivalence between these concepts for downward systems as well. But in this case weak and strong concepts are reversed; i.e. observability is equivalent to **coarse reconstructibility** for invertible $\phi$.

**Definition 3.6.10** *The system is* **downward coarsely reconstructible** *from $x(t_0)$ to $X_{M,t_0}$ if given knowledge of $W_{M,t_0}$ and $\tilde{Y}_{M,t_0}$, we can uniquely determine $ave(X_{M,t_0})$.*

If $\phi$ is invertible, then from eq.(3.6.37) we get

**Corollary 3.6.4** *Assuming that $F(m)$ is invertible for all $m$, the system is* **downward observable** *from $x(t_0)$ to $X_{M,t_0}$ iff the system is* **downward coarsely reconstructible** *from $x(t_0)$ to $X_{M,t_0}$.*

We now consider reachability for downward systems. In this case for the sake of establishing duality between downward and upward concepts we consider our initial condition to be $x(\gamma^{-1}t_0)$ rather than $x(t_0)$

**Definition 3.6.11** *The system is* **downward reachable** *from $x(\gamma^{-1}t_0)$ to $X_{M,t_0}$ if given any $\overline{x}(\gamma^{-1}t_0)$ and any desired $\overline{X}_{M,t_0}$, it is possible to specify $W_{M,t_0}$ so if $x(\gamma^{-1}t_0) = \overline{x}(\gamma^{-1}t_0)$, then $X_{M,t_0} = \overline{X}_{M,t_0}$.*

Note that if $x(\gamma^{-1}t_0) = 0$, then we have

$$X_{M,t_0} = \tilde{\mathcal{G}} W_{M,t_0} \tag{3.6.49}$$

where

$$\tilde{\mathcal{G}}_M^T \triangleq \left[\begin{array}{cccccccccccc}
\multicolumn{12}{c}{\overbrace{\hspace{12cm}}^{2^M \text{ blocks}}} \\
\tilde{\Theta}(0) & \tilde{\Theta}(0) & \cdots & & & & & & & & \cdots & \tilde{\Theta}(0) \\
\hline
\tilde{\Theta}(1) & \cdots & & & \cdots & \tilde{\Theta}(1) & 0 & \cdots & & & \cdots & 0 \\
0 & \cdots & & & \cdots & 0 & \tilde{\Theta}(1) & \cdots & & & \cdots & \tilde{\Theta}(1) \\
\hline
\tilde{\Theta}(2) & \cdots & \tilde{\Theta}(2) & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & \cdots & 0 & \tilde{\Theta}(2) & \cdots & \tilde{\Theta}(2) & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & \tilde{\Theta}(2) & \cdots & \tilde{\Theta}(2) & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \tilde{\Theta}(2) & \cdots & \tilde{\Theta}(2) \\
\hline
 & & & & \vdots & & & \vdots & & & & \\
 & & & & \vdots & & & \vdots & & & & \\
\hline
\tilde{\Theta}(M) & 0 & \cdots & & & & & & & & \cdots & 0 \\
0 & \tilde{\Theta}(M) & \cdots & & & & & & & & \cdots & 0 \\
\vdots & & & & & & & & & & & \vdots \\
0 & 0 & \cdots & & & & & & & & \cdots & \tilde{\Theta}(M)
\end{array}\right] \qquad (3.6.50)$$

Here

$$\tilde{\Theta}^T(i) \triangleq \phi(m(t_0)+M, m(t_0)+i)B(m(t_0)+i) \qquad (3.6.51)$$

Let us also define the following quantity.

**Definition 3.6.12 Downward-reachability Grammian**

$$\tilde{\mathcal{R}}(t_0, M) = \tilde{\mathcal{G}}\tilde{\mathcal{G}}^T \qquad (3.6.52)$$

Since reaching all possible states $X_{M,t_0}$ depends on having the rank of $\tilde{\mathcal{G}}$ be equal to the dimension of $X_{M,t_0}$, we get the following.

**Proposition 3.6.4** *The system is* **downward reachable** *from $x(t_0)$ to $X_{M,t_0}$ iff $\tilde{\mathcal{G}}$ has rank $m$ iff $\tilde{\mathcal{R}}(t_0,M)$ has rank $m$, where $m$ is the dimension of $X_{M,t_0}$.*

Finally, we define the following weaker concept of reachability which will prove to be the dual of reconstructibility for upward systems.

**Definition 3.6.13** *The system is* **downward coarsely reachable** *from $x(\gamma^{-1}t_0)$ to $X_{M,t_0}$ if given any $\overline{x}(\gamma^{-1}t_0)$ and any desired $ave(\overline{X}_{M,t_0})$, it is possible to specify $W_{M,t_0}$ so if $x(\gamma^{-1}t_0) = \overline{x}(\gamma^{-1}t_0)$, then $ave(X_{M,t_0}) = ave(\overline{X}_{M,t_0})$.*

Note that

$$ave(X_{M,t_0}) = (\frac{1}{2})^M I_M \tilde{\mathcal{G}} W_{M,t_0} \tag{3.6.53}$$

We also define the following.

**Definition 3.6.14 Downward-coarsely-reachability Grammian**

$$ave(\tilde{\mathcal{R}}(t_0,M)) = I_M \tilde{\mathcal{G}} \tilde{\mathcal{G}}^T I_M^T \tag{3.6.54}$$

**Proposition 3.6.5** *The system is* **downward coarsely reachable** *from $x(t_0)$ to $X_{M,t_0}$ iff $I_M\tilde{\mathcal{G}}$ has rank $n$ iff $ave(\tilde{\mathcal{R}}(t_0,M))$ has rank $n$, where $n$ is the dimension of the state.*

Having defined appropriately our downward system theoretic concepts, we can now state the following duality between upward and downward systems. Given the following downward model,

$$\begin{aligned} x(t_0) &= F^T(m(t_0))x(\gamma^{-1}t_0) + C^T(m(t_0))w(t_0) && (3.6.55) \\ x(t) &= \frac{1}{2}F^T(m(t))x(\gamma^{-1}t) + C^T(m(t))w(t) \quad m(t) > m(t_0) && (3.6.56) \\ y(t) &= \frac{1}{2}G^t(m(t))x(t) \quad m(t) > m(t_0) && (3.6.57) \end{aligned}$$

we have the following duality between the upward model(eq.'s(3.6.1,3.6.14)) and the downward model(eq.'s(3.6.55-3.6.57)).

| Upward | Downward |
|---|---|
| Reachability | Observability<br>(Coarse Reconstructibility) |
| Observability | Reachability |
| Reconstructibility<br>(Coarse Observability) | Coarse Reachability |

## 3.7 Bounds on the Error Covariance of the Filter

In the following sections we will analyze the stability of our upward Kalman filter via Lyapunov methods. As we will see our analysis of the ML filter will require bounds on $P_{ML}(m|m)$, and it will also be necessary to have bounds on $P(m|m)$ in order to infer stability of the optimal filter. Thus, in this section we begin by deriving strict upper and lower bounds for the optimal filter error covariance $P(m|m)$. We then use analogous arguments to derive upper and lower bounds for the ML filter error covariance $P_{ML}(m|m)$. Existence of these bounds depends on conditions that can be expressed in terms of the notions of upward reachability and upward reconstructibility developed in the previous section.

Recall our system whose dynamics are described by eq.(3.6.1) and whose measurements are described by eq.(3.6.14). We define the stochastic reachability grammian for this system as follows.

**Definition 3.7.1 Stochastic Reachability Grammian**

$$\begin{aligned}\overline{\mathcal{R}}(t_0,M) &\triangleq \sum_{i=0}^{M-1} 2^{-i-1}\phi(m(t_0),m(t_0)+i)G(m(t_0)+i+1)\\ &\times \tilde{Q}(m(t_0)+i+1)G^T(m(t_0)+i+1)\phi^T(m(t_0),m(t_0)+i) \quad (3.7.1)\end{aligned}$$

We define the stochastic reconstructibility grammian for this system as follows.

**Definition 3.7.2 Stochastic Reconstructibility Grammian**

$$\begin{aligned}\overline{\mathcal{O}}(t_0,M) \triangleq & \sum_{i=0}^{M} 2^i \phi^T(m(t_0)+i,m(t_0)+M)C^T(m(t_0)+i) \\ \times & R^{-1}(m(t_0)+i)C(m(t_0)+i)\phi(m(t_0)+i,m(t_0)+M) \quad (3.7.2)\end{aligned}$$

Among the assumptions that we make under which we prove our bounds is that the matrices $F(m)$, $F^{-1}(m)$, $G(m)$, $\tilde{Q}(m)$, $C(m)$, $R(m)$, and $R^{-1}(m)$ are bounded functions of $m$. In terms of our reachability and reconstructibility grammians these assumptions mean that for any $M_0 > 0$ we can find $\alpha, \beta > 0$ so that

$$\overline{\mathcal{R}}(t,M_0) \leq \alpha I \text{ for all } t \quad (3.7.3)$$

$$\overline{\mathcal{O}}(t,M_0) \leq \beta I \text{ for all } t \quad (3.7.4)$$

We define the notion of uniform reachability as follows.

**Definition 3.7.3** *An upward system is* **uniformly reachable** *if there exists* $\gamma, M_0 > 0$ *so that*

$$\overline{\mathcal{R}}(t,M_0) \geq \gamma I \text{ for all } t \quad (3.7.5)$$

This property insures that the process noise contributes a steady stream of uncertainty into the state. Intuitively, we would expect in this case that the error covariance $P(m|m)$ would never become equal to zero. In fact we prove that under uniform reachabilty $P(m|m)$ is lower bounded by a positive definite matrix.

We also need the notion of uniform reconstructibility, which is formulated as follows.

**Definition 3.7.4** *An upward system is* **uniformly reconstructible** *if there exists* $\delta, M_0 > 0$ *so that*

$$\overline{\mathcal{O}}(t,M_0) \geq \delta I \text{ for all } t \quad (3.7.6)$$

*where $M$ is the bottom level of a tree.*

This property insures a steady flow of information about the state of the system. Intuitively, we would expect that under this condition the uncertainty in our estimate remains bounded. In fact we prove that under the condition of uniform reconstructibility the error covariance, $P(m|m)$, is upper bounded.

Without loss of generality we can take $M_0$ to be the same in eq.'s(3.7.3-3.7.6) for any system which is uniformly reachable and reconstructible.

### 3.7.1 Upper Bound

We begin by deriving an upper bound for the optimal filter error covariance, $P(m|m)$. The general idea in deriving this bound is to make a careful comparison between the Riccati equations for our optimal filter and the Riccati equations for the standard Kalman filter. First consider the following lemma.

**Lemma 3.7.1** *Given the Riccati equation*

$$\begin{aligned} P(m|m+1) &= F(m+1)P(m+1|m+1)F^T(m+1) \\ &+ G(m+1)\tilde{Q}(m+1)G^T(m+1) \qquad (3.7.7) \\ P^{-1}(m|m) &= P^{-1}(m|m+1) + C^T(m)R^{-1}(m)C(m) \\ &+ P^{-1}(m|m+1) - P_x^{-1}(m) \qquad (3.7.8) \end{aligned}$$

*and the Riccati equation*

$$\begin{aligned} \overline{P}(m|m+1) &= F(m+1)\overline{P}(m+1|m+1)F^T(m+1) \\ &+ G(m+1)\tilde{Q}(m+1)G^T(m+1) \qquad (3.7.9) \\ \overline{P}^{-1}(m|m) &= \overline{P}^{-1}(m|m+1) + C^T(m)R^{-1}(m)C(m) \qquad (3.7.10) \end{aligned}$$

*we have that*

$$\overline{P}^{-1}(m|m) \leq P^{-1}(m|m) \qquad (3.7.11)$$

**Proof**

We first note that eq.(3.7.8) can be rewritten as

$$P^{-1}(m|m) = P^{-1}(m|m+1) + C^T(m)R^{-1}(m)C(m) + D^T(m)D(m) \qquad (3.7.12)$$

where $D^T(m)D(m)$ is positive semi-definite. This follows from the fact that $P(m|m+1) \leq P_x(m)$ or $P^{-1}(m|m+1) - P_x^{-1}(m) \geq 0$. The Riccati equation, eq.'s(3.7.9,3.7.10), characterizes the error covariance for the optimal filter corresponding to the following filtering problem.

$$\begin{aligned} x(m) &= F(m+1)x(m+1)+G(m+1)w(m+1) && (3.7.13)\\ E[w(m)w^T(m)] &= \tilde{Q}(m) && (3.7.14)\\ y(m) &= C(m)x(m)+v(m) && (3.7.15)\\ E[v(m)v^T(m)] &= R(m) && (3.7.16) \end{aligned}$$

Similarly, the Riccati equation, eq.'s(3.7.7,3.7.12), characterizes the error covariance for the optimal filter corresponding to the filtering problem involving the same state equation, eq.(3.7.13,3.7.14), but with the following measurement equation.

$$\begin{aligned} \tilde{y}(m) &= \begin{bmatrix} C(m) \\ D(m) \end{bmatrix} x(m)+u(m) && (3.7.17)\\ E[u(m)u^T(m)] &= \begin{bmatrix} R(m) & 0 \\ 0 & I \end{bmatrix} && (3.7.18) \end{aligned}$$

Since the filter corresponding to eq.(3.7.7,3.7.12) uses additional measurements compared to the filter corresponding to eq.(3.7.9,3.7.10), its error covariance can be no worse than the error covariance of the filter using fewer measurements; i.e. $P(m|m) \leq \overline{P}(m|m)$ or $\overline{P}^{-1}(m|m) \leq P^{-1}(m|m)$.

□

We now state and prove the following theorem concerning an upper bound for $P(m|m)$.

**Theorem 3.7.1** *Given uniform upper boundedness of the stochastic reconstructibility grammian, i.e. eq.(3.7.4), and given uniform reconstructibility of the system there exists $\kappa > 0$ such that for all $m$ at least $M_0$ levels from the initial level $P(m|m) \leq \kappa I$.*

**Proof**

Consider the following set of standard Riccati equations.

$$\begin{aligned} \overline{P}(m|m+1) &= F(m+1)\overline{P}(m+1|m+1)F^T(m+1) \\ &+ G(m+1)\tilde{Q}(m+1)G^T(m+1) && (3.7.19)\\ \overline{P}^{-1}(m|m) &= \overline{P}^{-1}(m|m+1)+C^T(m)R^{-1}(m)C(m) && (3.7.20) \end{aligned}$$

From standard Kalman filtering results we know that given $(F(m), R^{-\frac{1}{2}}(m)C(m))$ is a uniformly observable pair that is bounded above, there exists a $\kappa > 0$ such that $\overline{P}(m|m) \leq \kappa I$ or $\overline{P}^{-1}(m|m) \geq \kappa^{-1}I$. But by Corollary 3.6.2, $(F(m), R^{-\frac{1}{2}}(m)C(m))$ being a uniformly observable pair is equivalent to the original system being uniformly reconstructible. Also, the grammian $(F(m), R^{-\frac{1}{2}}(m)C(m))$ being bounded above is equivalent to our assumption of uniform upper boundedness of the stochastic reconstructibility grammian. Thus, under uniform reconstructibility and the uniform upper boundedness of the stochastic reconstructibility grammian of the original we deduce that $\overline{P}^{-1}(m|m) \geq \kappa^{-1}I$. But from Lemma 3.7.1 we know that $\overline{P}^{-1}(m|m) \leq P^{-1}(m|m)$. Thus, $P^{-1}(m|m) \geq \kappa^{-1}I$ or $P(m|m) \leq \kappa I$.

□

We can easily apply the previous ideas to derive an upper bound for $P_{ML}(m|m)$. Note that Lemma 3.7.1 would still apply if eq.(3.7.8) did not have the $P_x^{-1}(m)$ term and the matrices $F$ and $\tilde{Q}$ were replaced with the matrices $A^{-1}$ and $I$, respectively; i.e. the lemma would apply to the case of the ML Riccati equations. Then by using the same argument used to prove Theorem 3.7.1 we can show the following theorem.

**Theorem 3.7.2** *Given uniform upper boundedness of the stochastic reconstructibility grammian, i.e. eq.(3.7.4) where the state transition matrix $\phi$ corresponds to the system associated with dynamics $A^{-1}(m)$, and given uniform reconstructibility of the system there exists $\kappa' > 0$ such that for all $m$ at least $M_0$ levels from the initial level $P_{ML}(m|m) \leq \kappa' I$.*

### 3.7.2 Lower Bound

We now derive a lower bound for $P(m|m)$. As in deriving the upper bound, we appeal heavily to standard system theory.

**Lemma 3.7.2** *Let*

$$\overline{S}(m|m) \triangleq \frac{1}{2}(P^{-1}(m|m) - C^T(m)R^{-1}(m)C(m) + P_x^{-1}(m)) \quad (3.7.21)$$

$$\overline{S}(m|m+1) \triangleq F^{-T}(m+1)P^{-1}(m+1|m+1)F^{-1}(m+1) \quad (3.7.22)$$

*Given the Riccati equation*

$$\begin{aligned} S^*(m|m+1) &= 2F^{-T}(m+1)S^*(m+1|m+1)F^{-1}(m+1) \\ &+ F^{-T}(m+1)C^T(m)R^{-1}(m)C(m)F^{-1}(m+1) \qquad (3.7.23) \\ S^{*^{-1}}(m|m) &= S^{*^{-1}}(m|m+1) + G(m+1)\tilde{Q}(m+1)G^T(m+1) \qquad (3.7.24) \end{aligned}$$

*where* $\overline{S}(0|0) = S^*(0|0)$. *Then for all* $m$ $S^*(m|m) \geq \overline{S}(m|m)$.

**Proof**

By substituting eq.(3.7.12) into eq.(3.7.21) and collecting terms we get

$$\overline{S}(m|m) = P^{-1}(m|m+1) \qquad (3.7.25)$$

By substituting eq.(3.4.1) into eq.(3.7.25) we arrive at

$$\begin{aligned} \overline{S}(m|m) &= [F(m+1)P(m+1|m+1)F^T(m+1) \\ &+ G(m+1)\tilde{Q}(m+1)G^T(m+1)]^{-1} \\ &= [\overline{S}^{-1}(m|m+1) + G(m+1)\tilde{Q}(m+1)G^T(m+1)]^{-1} \qquad (3.7.26) \end{aligned}$$

where the the last equality results from the substitution of eq.(3.7.22). Also, by substituting eq.(3.7.21) into eq.(3.7.22) and collecting terms we get

$$\begin{aligned} \overline{S}(m|m+1) &= 2F^{-T}(m+1)\overline{S}(m+1|m+1)F^{-1}(m+1) \\ &+ F^{-T}(m+1)C^T(m)R^{-1}(m)C(m)F^{-1}(m+1) \\ &- F^{-T}(m+1)P_x^{-1}(m)F^{-1}(m+1) \qquad (3.7.27) \end{aligned}$$

Now we prove by induction that for all $m$ $S^*(m|m) \geq \overline{S}(m|m)$. Obviously, $S^*(0|0) \geq \overline{S}(0|0)$. As an induction hypothesis we assume $S^*(i+1|i+1) \geq \overline{S}(i+1|i+1)$. From eq.(3.7.27), eq.(3.7.23), and the fact that $F^{-T}(m+1)P_x^{-1}(m)F^{-1}(m+1) \geq 0$ we get that

$$S^{*^{-1}}(i|i+1) \leq \overline{S}^{-1}(i|i) \qquad (3.7.28)$$

Substituting eq.(3.7.24) and eq.(3.7.26) into eq.(3.7.28) and canceling terms we arrive at $S^{*^{-1}}(i|i) \leq \overline{S}^{-1}(i|i)$, i.e. $S^*(i|i) \geq \overline{S}(i|i)$.

□

**Theorem 3.7.3** *Given uniform upper boundedness of the stochastic reachability grammian, i.e. eq.(3.7.3), and given uniform reachability of the system there exists $L > 0$ such that for all $m$ at least $M_0$ levels from the initial level $P(m|m) \geq LI$.*

**Proof**

Consider the following set of standard Riccati equations.

$$\begin{aligned} S^*(m|m+1) &= 2F^{-T}(m+1)S^*(m+1|m+1)F^{-1}(m+1) \\ &+ F^{-T}(m+1)C^T(m)R^{-1}(m)C(m)F^{-1}(m+1) \qquad (3.7.29) \\ S^{*^{-1}}(m|m) &= S^{*^{-1}}(m|m+1) + G(m+1)\tilde{Q}(m+1)G^T(m+1) \qquad (3.7.30) \end{aligned}$$

From standard Kalman filtering results we know that if $(F^{-T}(m), G(m)\tilde{Q}^{\frac{1}{2}}(m))$ is a uniformly reachable pair that is bounded above, then there exists $N > 0$ such that $S^*(m|m) \leq NI$. However, from Corollary 3.6.1 and the invertibility of $F(m)$ the uniform reachability of the pair $(F^{-T}(m), G(m)\tilde{Q}^{\frac{1}{2}}(m))$ is equivalent to the original system being uniformly reachable. Also, the grammian $(F^{-T}(m), G(m)\tilde{Q}^{\frac{1}{2}}(m))$ being bounded above is equivalent to our assumption of uniform upper boundedness of the stochastic reachability grammian. Thus, under uniform reconstructibility and the uniform upper boundedness of the stochastic reconstructibility grammian of the original we deduce that $S^*(m|m) \leq NI$. But from Lemma 3.7.2 we know that $S^*(m|m) \geq \overline{S}(m|m)$. Thus, $\overline{S}(m|m) \leq NI$. But from eq.(3.7.21) we get

$$\frac{1}{2}(P^{-1}(m|m) - C^T(m)R^{-1}(m)C(m) + P_x^{-1}(m)) \leq NI \qquad (3.7.31)$$

It follows straightforwardly that

$$P^{-1}(m|m) \leq L^{-1}I \qquad (3.7.32)$$

where

$$L^{-1}I \geq 2NI + C^T(m)R^{-1}(m)C(m) \qquad (3.7.33)$$

Thus,

$$P(m|m) \geq LI \qquad (3.7.34)$$

□

Using analogous arguments we can derive a lower bound for $P_{ML}(m|m)$. Note that with the following definitions for $\overline{S}$ and the equations (3.7.23,3.7.24) where the matrices $F$ and $\tilde{Q}$ are replaced with the matrices $A^{-1}$ and $I$, respectively, Lemma 3.7.2 still applies.

$$\overline{S}(m|m) \triangleq \frac{1}{2}(P_{ML}^{-1}(m|m) - C^T(m)R^{-1}(m)C(m)) \tag{3.7.35}$$

$$\overline{S}(m|m+1) \triangleq A^T(m+1)P_{ML}^{-1}(m+1|m+1)A(m+1) \tag{3.7.36}$$

Using the same argument as in the proof of Theorem 3.7.3 with our current definitions for $\overline{S}$ we get that

$$\frac{1}{2}(P_{ML}^{-1}(m|m) - C^T(m)R^{-1}(m)C(m)) \leq NI \tag{3.7.37}$$

for $N > 0$. Equivalently,

$$P_{ML}^{-1}(m|m) \leq (L')^{-1}I \tag{3.7.38}$$

for

$$(L')^{-1}I \geq 2NI + C^T(m)R^{-1}(m)C(m) \tag{3.7.39}$$

Thus, we have the following theorem.

**Theorem 3.7.4** *Given uniform upper boundedness of the stochastic reachability grammian, i.e. eq.(3.7.3), where the state transition matrix $\phi$ corresponds to the system associated with dynamics $A^{-1}(m)$ and the matrix $\tilde{Q}$ is replaced with $I$, and given uniform reachability of the system there exists $L' > 0$ such that for all $m$ $P_{ML}(m|m) \geq L'I$.*

## 3.8 Upward Stability on Trees

In this section we formalize the notion of stability for dynamic systems evolving up the tree. The dynamics on which we are interested in focusing the major portion of our analysis are the ML error dynamics of eq.(3.4.60). Thus the general class of systems we wish to study here has the form

$$z(t) = \mathcal{F}(m(t)+1)[z(\alpha t) + z(\beta t)] + \mathcal{G}(m(t))u(t) \tag{3.8.1}$$

What we wish to do is to study the asymptotic stability of this system as the dynamics propagate up the tree. Since we are interested in internal stability, we will consider the autonomous system with $u \equiv 0$.

Intuitively what we would like stability to mean is that $z(t) \rightarrow 0$ as we propagate farther and farther away from the initial level of the tree. Note, however, that as we move up the tree(or equivalently as the initial level moves farther down), $z(t)$ is influenced by a geometrically increasing number of nodes at the initial level. For example, $z(t)$ depends on $\{z(\alpha t), z(\beta t)\}$ or, alternatively on $\{z(\alpha^2 t), z(\beta\alpha t), z(\alpha\beta t), z(\beta^2 t)\}$ or, alternatively on $\{z(\alpha^3 t), z(\beta\alpha^2 t), z(\alpha\beta\alpha t), z(\beta^2\alpha t), z(\alpha^2\beta t), z(\beta\alpha\beta t), z(\alpha\beta^2 t), z(\beta^3 t)\}$, etc. Thus in order to study asymptotic stability it is **necessary** to consider an infinite dyadic tree, with an infinite set of initial conditions corresponding to all nodes at the initial level. Note also, that we might expect that there would be a number of meanings we could give to "$z(t) \rightarrow 0$" – e.g. do we consider individual nodes at a level or the infinite sequence of values at all points at a level?

To formalize the notion of stability let us change the sense of our index of recursion so that $m$ increases as we move up the tree. Specifically, we arbitrarily choose a level of the tree to be our "initial" level, i.e. level 0, and we index the points on this initial level as $z_i(0)$ for $i \in \mathcal{Z}$. Points at the $m$th level up from level 0 are denoted $z_i(m)$ for $i \in \mathcal{Z}$. The dynamical equations we then wish to consider are of the form

$$z_i(m) = \mathcal{A}(m-1)(z_{2i}(m-1) + z_{2i+1}(m-1)) \tag{3.8.2}$$

Let $Z(m)$ denote the infinite sequence at level $m$, i.e. the set $\{z_i(m)\ ,\ i \in \mathcal{Z}\}$. The $p$-norm on such a sequence is defined as

$$\|Z(m)\|_p \triangleq (\sum_i \|z_i(m)\|_p^p)^{\frac{1}{p}} \tag{3.8.3}$$

where $\|z_i(m)\|_p$ is the standard $p$-norm for the finite dimensional vector $z_i(m)$.

We define the following notion of exponential stability for a system.

**Definition 3.8.1** *A system is $l_p$-exponentially stable if given any initial sequence $Z(0)$ such that $\|Z(0)\|_p < \infty$,*

$$\|Z(m)\|_p \leq C\alpha^m \|Z(0)\|_p \tag{3.8.4}$$

*where $0 \leq \alpha < 1$ and $C$ is a positive constant.*

From eq.(3.8.2) we can easily write down the following.

$$z_i(m) = \Phi(m,0) \sum_{j \in O_{m,i}} z_j(0) \tag{3.8.5}$$

where the cardinality of the set $O_{m,i}$ is $2^m$ and for $m_1 \geq m_2$

$$\Phi(m_1,m_2) = \begin{cases} I & m_1, m_2 \\ \mathcal{A}(m_1-1)\Phi(m_1-1,m_2) & m_1 > m_2 \end{cases} \tag{3.8.6}$$

As in the case of standard dynamic systems it is the state transition matrix, $\Phi(m,0)$, which plays a crucial role in studying stability on trees. However, unlike the standard case, as one can see from eq.(3.8.5), the nature of the initial condition that influences $z_i(m)$ depends crucially on $m$; in particular the number of points at level 0 to be summed up and scaled to give $z_i(m)$ is $2^m$. These observations lead to the following:

**Theorem 3.8.1** *The system defined in eq.(3.8.2) is $l_p$-exponentially stable if and only if*

$$2^{\frac{m(p-1)}{p}} \|\Phi(m,0)\|_p \leq K'\gamma^m \quad \textit{for all } m \tag{3.8.7}$$

*where $0 \leq \gamma < 1$ and $K'$ is a positive constant.*

**Proof**

Let us first show necessity. Specifically, suppose that for any $K > 0$, $0 \leq \gamma < 1$, and $M \geq 0$ we can find a vector $z$ and an $m \geq M$ so that

$$\|\Phi(m,0)z\|_p > K\gamma^m 2^{-\frac{m}{q}} \|z\|_p \tag{3.8.8}$$

where

$$\frac{1}{p} + \frac{1}{q} = 1 \tag{3.8.9}$$

Let $z$ and $m$ be such a vector and integer for some choice of $K$, $\gamma$, and $M$, and define an initial sequence as follows. Let $\rho_0, \rho_1, \rho_2, \ldots$ be a sequence with

$$\sum_{i=0}^{\infty} \rho_i^p = 1 \tag{3.8.10}$$

Then let

$$z_i(0) = \begin{cases} \rho_0 z & 0 \le i < 2^m \\ \rho_1 z & 2^m \le i < 2 \cdot 2^m \\ \vdots & \\ \rho_i z & j2^m \le i < (j+1)2^m \\ \vdots & \end{cases} \tag{3.8.11}$$

Note that

$$\begin{aligned} \|Z(0)\|_p^p &= \sum_{i=0}^{\infty} \|z_i(0)\|_p^p \\ &= 2^m \|z\|_p^p \end{aligned} \tag{3.8.12}$$

Also, note that

$$\begin{aligned} z_i(m) &= \Phi(m,0) \sum_{j=i2^m}^{(i+1)2^m-1} z_j(0) \\ &= 2^m \rho_i \Phi(m,0) z \end{aligned} \tag{3.8.13}$$

Thus,

$$\begin{aligned} \|Z(m)\|_p^p &= 2^{mp} \|\Phi(m,0)z\|_p^p \\ &> 2^{mp} K^p \gamma^{mp} 2^{\frac{-mp}{q}} \|z\|_p^p \\ &= 2^{mp} K^p \gamma^{mp} 2^{\frac{-mp}{q}} 2^{-m} \|Z(0)\|_p^p \\ &= K^p \gamma^{mp} \|Z(0)\|_p^p \end{aligned} \tag{3.8.14}$$

where the first equality comes from eq.(3.8.10), the inequality from eq.(3.8.8), the next equality from eq.(3.8.12), and the last equality from eq.(3.8.9). Hence for any $K$, $0 \le \gamma < 1$ and $M \ge 0$ we can find an initial $l_p$-sequence $Z(0)$ and an $m \ge M$ so that

$$\|Z(m)\|_p > K\gamma^m \|Z(0)\|_p \tag{3.8.15}$$

so that the system cannot be $l_p$-exponentially stable.

To prove sufficiency we use the following.

**Lemma 3.8.1** *A system is $l_p$-exponentially stable if for every $i$*

$$\|z_i(m)\|_p \leq K\beta^m(\sum_{j\in O_{m,i}} \|z_j(0)\|_p^p)^{\frac{1}{p}} \tag{3.8.16}$$

*where $0 \leq \beta < 1$ and $K$ is a positive constant.*

**Proof**

By raising both sides of eq.(3.8.16) to the $p$th power we get

$$\|z_i(m)\|_p^p \leq K^p(\beta^p)^m \sum_{j\in O_{m,i}} \|z_j(0)\|_p^p \tag{3.8.17}$$

Since eq.(3.8.17) holds for every $i$ we can write

$$\sum_i \|z_i(m)\|_p^p \leq K^p(\beta^p)^m \sum_i \|z_i(0)\|_p^p \tag{3.8.18}$$

The lemma follows from raising both sides of eq.(3.8.18) to the power of $\frac{1}{p}$.

□

**Lemma 3.8.2** *Consider the sequence of vectors $x_i$ for $i \in \mathcal{Z}$. Then, for any $m$ and any $j$*

$$\|\sum_{i\in O_{m,j}} x_i\|_p \leq 2^{\frac{m}{q}}(\sum_{i\in O_{m,j}} \|x_i\|_p^p)^{\frac{1}{p}} \tag{3.8.19}$$

*where $O_{m,j} = \{j, j+1, ...j+2^m-1\}$ and $q$ satisfies eq.(3.8.9).*

**Proof**

We first show the following.

$$\|a+b\|_p \leq 2^{\frac{1}{q}}(\|a\|_p^p + \|b\|_p^p)^{\frac{1}{p}} \tag{3.8.20}$$

Since $\|\cdot\|_p^p$ is a convex function, we can write

$$\|(\frac{1}{2})a + (1-\frac{1}{2})b\|_p^p \leq (\frac{1}{2})\|a\|_p^p + (1-\frac{1}{2})\|b\|_p^p \tag{3.8.21}$$

from which eq.(3.8.20) follows immediately. We now show the result by induction on $m$. Suppose for all $j$

$$\|\sum_{i\in O_{m,j}} x_i\|_p \leq 2^{\frac{m}{q}}(\sum_{i\in O_{m,j}} \|x_i\|_p^p)^{\frac{1}{p}} \tag{3.8.22}$$

Consider the summing $x_i$ over the two sets $O_{m,j_1}$ and $O_{m,j_2}$ where $j_2 = j_1 + 2^m$. From eq.(3.8.20) we get

$$\|(\sum_{i\in O_{m,j_1}} x_i + \sum_{i\in O_{m,j_2}} x_i)\|_p \leq 2^{\frac{1}{q}}(\|(\sum_{i\in O_{m,j_1}} x_i\|_p^p + \|(\sum_{i\in O_{m,j_2}} x_i\|_p^p)^{\frac{1}{p}} \tag{3.8.23}$$

Then by substituting into eq.(3.8.22) eq.(3.8.23) we get

$$\|\sum_{i\in O_{m,j_1}\cup O_{m,j_2}} x_i\|_p \leq 2^{\frac{(m+1)}{q}}(\|(\sum_{i\in O_{m,j_1}} x_i\|_p^p + \|(\sum_{i\in O_{m,j_2}} x_i\|_p^p)^{\frac{1}{p}} \tag{3.8.24}$$

□

We can now show sufficiency thereby completing the proof of the theorem. By applying the $p$-norm to eq.(3.8.5) and using the Cauchy-Schwarz inequality we get

$$\|z_i(m)\|_p \leq \|\Phi(m,0)\|_p\|\sum_{j\in O_{m,i}} z_j(0)\|_p \tag{3.8.25}$$

Using Lemma 3.8.2, we get

$$\|z_i(m)\|_p \leq \|\Phi(m,0)\|_p 2^{\frac{m}{q}}(\sum_{j\in O_{m,i}} \|z_j(0)\|_p^p)^{\frac{1}{p}} \tag{3.8.26}$$

By substituting eq.(3.8.7) into eq.(3.8.26) we get

$$\|z_i(m)\|_p \leq K'\gamma^m(\sum_{j\in O_{m,i}} \|z_j(0)\|_p^p)^{\frac{1}{p}} \tag{3.8.27}$$

which by Lemma 3.8.1 shows the system to be $l_p$-exponentially stable.

□

Note that referring to eq.'s(3.8.2,3.8.5,3.8.6) we see that the $l_p$-exponential stability of eq.(3.8.2) is equivalent to the usual exponential stability of the system

$$\xi(m) = 2^{\frac{p-1}{p}}A(m-1)\xi(m-1) \tag{3.8.28}$$

For example for $p = 2$, we are interested in the exponential stability of

$$\xi(m) = \sqrt{2}A(m-1)\xi(m-1) \tag{3.8.29}$$

If $A$ is constant this is equivalent to requiring $A$ to have eigenvalues with magnitudes $< \frac{\sqrt{2}}{2}$.

Note also that it is straightforward to show that if one considers the system with inputs and outputs

$$\begin{aligned} z_i(m) &= \mathcal{A}(m-1)(z_{2i}(m-1) + z_{2i+1}(m-1)) \\ &+ \mathcal{B}(m-1)(u_{2i}(m-1) + u_{2i+1}(m-1)) && (3.8.30) \\ y_i(m) &= \mathcal{C}(m)z_i(m) && (3.8.31) \end{aligned}$$

then if $\mathcal{B}(m)$ and $\mathcal{C}(m)$ are bounded, the asymptotic stability of the undriven dynamics imply bounded-input/bounded-output stability.

## 3.9 Filter Stability

In this section we show that the error dynamics of the maximum likelihood filter are stable and also that the same is true of the overall filter.

**Theorem 3.9.1** *Suppose that the ML system is uniformly reachable and uniformly reconstructible. Then, the error dynamics of the maximum likelihood filter are $l_2$-exponentially stable.*

**Proof**

The following proof follows closely the standard proof for stability of discrete-time Kalman filters given in [24]. Based on the comments at the end of the preceding section and on the ML error dynamics of eq.(3.4.60), we see that we wish to show that the following causal system is stable in the standard sense.

$$z(m) = P_{ML}(m|m)P_{ML}^{-1}(m|m-1)\sqrt{2}A^{-1}(m-1)z(m-1) \qquad (3.9.1)$$

Theorem's 3.7.2 and 3.7.4, i.e. the upper and lower bounds on $P_{ML}(m|m)$, allow us to define the following Lyapunov function.

$$V(z,m) \triangleq z^T(m)P_{ML}^{-1}(m|m)z(m) \qquad (3.9.2)$$

Let us also define the following quantity.

$$\begin{aligned} \tilde{z}(m) &\triangleq \sqrt{2}A^{-1}(m-1)z(m-1) && (3.9.3) \\ &= P_{ML}(m|m-1)P_{ML}^{-1}(m|m)z(m) && (3.9.4) \end{aligned}$$

Substituting eq.(3.4.7) into eq.(3.9.2) followed by algebraic manipulations, one gets

$$\begin{aligned} V(z,m) &= z^T(m)(2P_{ML}^{-1}(m|m-1)+C^T(m)R^{-1}(m)C(m))z(m) && (3.9.5) \\ &= 2z^T(m)(P_{ML}^{-1}(m|m)-2P_{ML}^{-1}(m|m-1))z(m)-z^T(m)C^T(m)R^{-1}(m)C(m))z(m) \\ &+ z^T(m)(2P_{ML}^{-1}(m|m-1))z(m) \\ &+ \frac{\tilde{z}^T(m)}{\sqrt{2}}P_{ML}^{-1}(m|m-1)\frac{\tilde{z}(m)}{\sqrt{2}}-\frac{\tilde{z}^T(m)}{\sqrt{2}}P_{ML}^{-1}(m|m-1)\frac{\tilde{z}(m)}{\sqrt{2}} && (3.9.6) \\ &= -(\sqrt{2}z(m)-\frac{\tilde{z}(m)}{\sqrt{2}})^TP_{ML}^{-1}(m|m-1)(\sqrt{2}z(m)-\frac{\tilde{z}(m)}{\sqrt{2}}) \\ &- z^T(m)C^T(m)R^{-1}(m)C(m)z(m)+\frac{\tilde{z}^T(m)}{\sqrt{2}}P_{ML}^{-1}(m|m-1)\frac{\tilde{z}(m)}{\sqrt{2}} && (3.9.7) \end{aligned}$$

But note that by using the matrix inversion lemma we get

$$\begin{aligned} \frac{\tilde{z}^T(m)}{\sqrt{2}}P_{ML}^{-1}(m|m-1)\frac{\tilde{z}(m)}{\sqrt{2}} &= V(z,m-1)-\Delta && (3.9.8) \\ \Delta &\geq 0 && (3.9.9) \end{aligned}$$

It follows that

$$\begin{aligned} V(z,m)-V(z,m-1) &\leq -(\sqrt{2}z(m)-\frac{\tilde{z}(m)}{\sqrt{2}})^TP_{ML}^{-1}(m|m-1)(\sqrt{2}z(m)-\frac{\tilde{z}(m)}{\sqrt{2}}) \\ &- z^T(m)C^T(m)R^{-1}(m)C(m)z(m) && (3.9.10) \end{aligned}$$

Stability follows from eq.(3.9.10) under the condition of uniform observability of the pair $(A^{-1}(m), R^{-\frac{1}{2}}(m)C(m))$ which by Corollary 3.6.2 is equivalent to uniform reconstructibility of the system.

□

Let us now examine the full estimation error after incorporating prior statistics. It is straightforward to see that

$$\tilde{x}(t|t) = P(m(t)|m(t))(P_{ML}^{-1}(m(t)|m(t))\tilde{x}_{ML}(t|t)+P_x^{-1}(m(t))x(t)) \qquad (3.9.11)$$

Thus we can view $\tilde{x}(t|t)$ as a linear combination of the states of two upward-evolving systems, eq.(3.4.60) for $\tilde{x}_{ML}(t|t)$ and one for $P_x^{-1}(m(t))x(t)$. Note first that since $P(m|m) \leq P_{ML}(m|m)$

$$||P(m(t)|m(t))P_{ML}^{-1}(m(t)|m(t))\tilde{x}_{ML}(t|t)|| \leq ||\tilde{x}_{ML}(t|t)|| \qquad (3.9.12)$$

and we already have the stability of the $\tilde{x}_{ML}(t|t)$ dynamics from Theorem 3.9.1. Turning to the second term in eq.(3.9.11), note first that thanks to Theorem 3.7.1, $P(m(t)|m(t))$ is bounded. Note also that the covariance of $P_x^{-1}(m(t))x(t)$ is simply $P_x^{-1}(m(t))$. By uniform reachability $P_x^{-1}(m(t))$ is bounded above. Thus, while $P_x(m(t))$ might diverge, the contribution to the error of the second term in eq.(3.9.11) is bounded.

Also, our previous analysis allows us to conclude that the full, driven $\tilde{x}_{ML}(t|t)$ dynamics are bounded-input, bounded-output stable from inputs $\tilde{w}$ and $v$ to output $\tilde{x}_{ML}(t|t)$. If we use eq.(3.4.50), together with eq.(3.2.10) and eq.'s(3.2.6-3.2.8) we can write down the following upward dynamics for $\xi(t) = P_x^{-1}(m(t))x(t)$:

$$\begin{aligned} \xi(t) &= \frac{1}{2}A^T(m(t)+1)(\xi(\alpha t)+\xi(\beta t)) \\ &+ \frac{1}{2}N(m(t)+1)(\tilde{w}(\alpha t)+\tilde{w}(\beta t)) \end{aligned} \tag{3.9.13}$$

where

$$N(m(t)+1) = P_x^{-1}(m(t))A^{-1}(m(t)+1)B(m(t)+1) \tag{3.9.14}$$

Note that in general there is no reason to constrain the autonomous dynamics of eq.(3.9.13) to be stable. However, if they are not, then reachability implies that $P_x(m) \to \infty$ so that $N(m) \to 0$ and the covariance of $\tilde{w} \to I$. The bounded-input, bounded-output stability of this system can be easily checked.

## 3.10 Steady-state Filter

In this section we study properties of our filter under steady-state conditions; i.e. we analyze the asymptotic properties of the filter. We state and prove several results. First we show that the error covariance of the ML estimator converges to a steady-state limit and that furthermore, the steady-state filter is $l_2$-exponentially stable.

**Theorem 3.10.1** *Consider the following system defined on a tree.*

$$\begin{aligned} x(t) &= Ax(\gamma^{-1}) + Bw(t) && (3.10.1) \\ y(t) &= Cx(t) + v(t) && (3.10.2) \end{aligned}$$

$$\begin{aligned} E[w(t)w^T(t)] &= I && (3.10.3) \\ E[v(t)v^T(t)] &= R && (3.10.4) \end{aligned}$$

*where $v(t)$ and $w(t)$ are white noise processes on the tree and $A$ is invertible. Suppose that $(A^{-1}, G)$ is a reachable pair and $(A^{-1}, R^{-\frac{1}{2}}C)$ is an observable pair, where $G = A^{-1}B$. The error covariance for the ML estimator, $P_{ML}(m|m)$, converges as $m \rightarrow \infty$ to $\overline{P}_\infty$, which is the unique positive definite solution to*

$$\begin{aligned} \overline{P}_\infty &= \frac{1}{2}A^{-1}\overline{P}_\infty A^{-T} + \frac{1}{2}GG^T \\ &- K_\infty(\frac{1}{2}CA^{-1}\overline{P}_\infty A^{-T}C^T + \frac{1}{2}CGG^TC^T + R)K_\infty^T \qquad (3.10.5) \end{aligned}$$

*where*

$$K_\infty = \overline{P}_\infty C^T R^{-1} \qquad (3.10.6)$$

Moreover, the autonomous dynamics of the steady-state ML filter, i.e.

$$e(t) = \frac{1}{2}(I - K_\infty C)A^{-1}(e(\alpha t) + e(\beta t)) \qquad (3.10.7)$$

are $l_2$-exponentially stable.

**Proof**

Recall the Riccati equations for the ML estimator where the scale variable $m$ increases in the direction upward along the tree.

$$\begin{aligned} P_{ML}(m|m+1) &= A^{-1}P_{ML}(m+1|m+1)A^{-T} + GG^T \\ & && (3.10.8) \\ P_{ML}^{-1}(m|m) &= 2P_{ML}^{-1}(m|m+1) + C^TR^{-1}C && (3.10.9) \end{aligned}$$

Convergence of $P_{ML}(m|m)$

In order to show the existence of a limit of $P_{ML}(m|m)$ as $m \rightarrow \infty$ we show that both a) $P_{ML}(m|m)$ is monotone-nonincreasing in $m$ and b) $P_{ML}(m|m)$ is bounded below.

a) We adopt the following notation.

$$\begin{aligned} P(m) &\triangleq P_{ML}(m|m) \; m \geq 0 && (3.10.10) \\ P(m; m') &\triangleq P(m - m') \; m \geq m' && (3.10.11) \end{aligned}$$

By the scale-invariance of our system showing

$$m_1 < m_2 \to P(m;m_1) \leq P(m;m_2) \tag{3.10.12}$$

is equivalent to demonstrating that $P(m)$ is monotone-nonincreasing.

We note that eq.'s(3.10.8,3.10.9) preserve positive definite orderings; i.e. if $P_1(m_2) \leq P_2(m_2)$ then $P_1(m;m_2) \leq P_2(m;m_2)$ for $m \geq m_2$. We now take

$$\begin{aligned} P_1(m_2) &= P(m_2;m_1) && (3.10.13) \\ P_2(m_2) &= \infty \text{ (initial condition for the ML estimator)} && (3.10.14) \end{aligned}$$

Then,

$$\begin{aligned} P_1(m;m_2) &= P(m;m_1) && (3.10.15) \\ P_2(m;m_2) &= P(m;m_2) && (3.10.16) \end{aligned}$$

for $m \geq m_2$. So by the property of positive definite ordering of the Riccati equations we know that

$$P_1(m;m_2) \leq P_2(m;m_2) \tag{3.10.17}$$

and thus,

$$P(m;m_1) \leq P(m;m_2) \tag{3.10.18}$$

b) The fact that $P_{ML}(m|m)$ is bounded below follows from Theorem 3.7.4 under our assumptions of reachability and observability.

Having established the convergence of $P_{ML}(m|m)$, let us denote the limit as follows.

$$\lim_{m \to \infty} P_{ML}(m|m) \triangleq \overline{P}_\infty \tag{3.10.19}$$

Note that by Theorem 3.7.4 $\overline{P}_\infty$ must be positive definite. We can also establish that $P_{ML}(m|m)$ must converge to the solution of the steady state Riccati eq.(3.10.5). Since $P_{ML}(m|m)$ both satisfies the Riccati eq.'s(3.10.8,3.10.9) and converges to a limit, this limit must satisfy the fixed point equation for eq.'s(3.10.8,3.10.9). This fixed point equation is precisely the steady state Riccati eq.(3.10.5).

Exponential Stability of $\frac{1}{2}(I - K_\infty C)A^{-1}$

In order for $\frac{1}{2}(I - K_\infty C)A^{-1}$ to be $l_2$-exponentially stable, it must have eigenvalues that are strictly less than $\frac{\sqrt{2}}{2}$. This fact follows from Theorem 3.8.1.

From Theorem 3.9.1 we know that the following system is exponentially stable with respect to $\|\cdot\|_2^*$.

$$z(t) = P_{ML}(m(t)|m(t))P_{ML}^{-1}(m(t)|m(t)-1)(z(\alpha t) + z(\beta t)) \tag{3.10.20}$$

which can be rewritten as

$$z(t) = \frac{1}{2}(I - K(m(t))C)A^{-1}(z(\alpha t) + z(\beta t)) \tag{3.10.21}$$

where

$$K(m(t)) = P_{ML}(m(t)|m(t))C^T R^{-1} \tag{3.10.22}$$

But, since $\lim_{m\to\infty} P_{ML}(m|m) = \overline{P}_\infty$, the system in eq.(3.10.21) in steady-state becomes

$$z(t) = \frac{1}{2}(I - K_\infty C)A^{-1}(z(\alpha t) + z(\beta t)) \tag{3.10.23}$$

Uniqueness of $\overline{P}_\infty$

Consider $P_1$ and $P_2$, both of which satisfy the steady state Riccati eq.(3.10.5). Thus,

$$\begin{aligned} P_1 &= \frac{1}{2}A^{-1}P_1A^{-T} + \frac{1}{2}GG^T \\ &- K_1(\frac{1}{2}CA^{-1}P_1A^{-T}C^T + \frac{1}{2}CGG^TC^T + R)K_1^T \end{aligned} \tag{3.10.24}$$

$$\begin{aligned} P_2 &= \frac{1}{2}A^{-1}P_2A^{-T} + \frac{1}{2}GG^T \\ &- K_2(\frac{1}{2}CA^{-1}P_2A^{-T}C^T + \frac{1}{2}CGG^TC^T + R)K_2^T \end{aligned} \tag{3.10.25}$$

Subtracting eq.(3.10.25) from eq.(3.10.24) we get

$$\begin{aligned} P_1 - P_2 &= \frac{\sqrt{2}}{2}(I - K_1C)A^{-1}(P_1 - P_2)(\frac{\sqrt{2}}{2}(I - K_1C)A^{-1})^T \\ &+ \Delta \end{aligned} \tag{3.10.26}$$

where $\Delta$ is a symmetric matrix. Note that we have established the fact that $\frac{\sqrt{2}}{2}(I - K_1C)A^{-1}$ has eigenvalues within the unit circle. From standard system theory this tells us that we can write $P_1 - P_2$ as a sum of positive semidefinite terms. This implies that $P_1 - P_2$ is positive semidefinite or $P_1 \geq P_2$. By subtracting eq.(3.10.24) from eq.(3.10.25) and using the same argument we can establish that $P_2 \geq P_1$.

□

Note that the preceding analysis assumed constant matrices $A, B, C, Q$, and $R$ invertible. We also assume the process is in steady-state; i.e. $P_x^{-1}$ is constant. As we are interested in asymptotic behavior there is no loss of generality in assuming this and there are two distinct cases. Specifically, if $A$ is stable, then the covariance $P_x(m(t))$ at all finite nodes(starting from an infinitely remote coarse level) is the positive definite(because of reachability) solution $P_x$ of eq.(3.2.14), and in this case, we have that

$$P(m|m) \rightarrow (\overline{P}_\infty^{-1} + P_x^{-1})^{-1} \tag{3.10.27}$$

On the other hand, if $A$ is unstable, $P_x^{-1}(m(t)) \rightarrow 0$ and

$$P(m|m) \rightarrow \overline{P}_\infty \tag{3.10.28}$$

Note that the existence of two distinct limiting forms for $P(m|m)$, depending on the stability of the original model is another significant deviation from standard causal theory.

## 3.11 Model Extensions and Iterative Algorithms

In this section we first describe an extension of the models considered in this chapter which allows finite correlations among the process noises at each scale of the tree. These processes are considered in [8] for which a system theory is developed, including results on the realization of these processes. We show how this class of processes can be treated in the same way as in our previous models. We then turn to the idea of developing iterative algorithms for smoothing our multiscale processes. We develop the idea in detail for the case of tree processes, then show how this may be extended to the case of general lattice processes.

We begin by discussing an extension of our state model on trees which lets the process noise in our model be *correlated* in finite subsets of the nodes of the tree. In

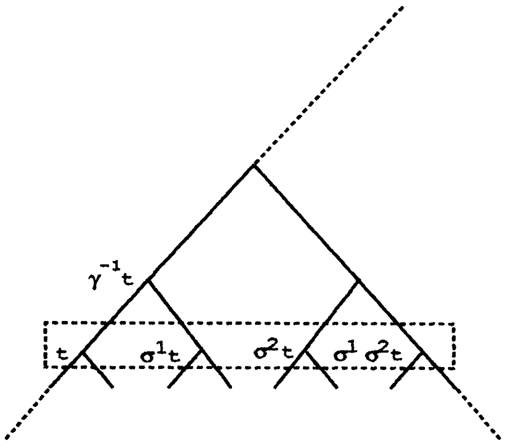

Figure 3.11.2: The Set of Points $M_2(\sigma)$ for a Particular Point $t$ on the Tree

particular we consider models where the state at each node on the tree $t$ is equal to a function of the state at $\gamma^{-1}t$ plus a linear combination of noises in the vicinity of $t$. To describe these models precisely, we first define the following set of operators, $\sigma^n$, comprising our operators $\alpha, \beta$ and $\gamma$. They are defined recursively as

$$\sigma^n t = \begin{cases} \beta\sigma^{n-1}\gamma^{-1}t & t = \beta\gamma^{-1}t \\ \alpha\sigma^{n-1}\gamma^{-1}t & t = \alpha\gamma^{-1}t \end{cases} \tag{3.11.29}$$

where we have as initial condition

$$\sigma^1 t = \begin{cases} \beta\gamma^{-1}t & t = \alpha\gamma^{-1}t \\ \alpha\gamma^{-1}t & t = \beta\gamma^{-1}t \end{cases} \tag{3.11.30}$$

Note that if we label the points along the level of the tree in binary notation, the action of $\sigma^n$ on the point 000...0 is equivalent to flipping the $n$th least significant bit. Note also that for any $n$ $\sigma^n\sigma^n = 0$ where 0 denotes the identity operator. We also define the set $M_k(\sigma)$ to be the set of all distinct monomials in $\sigma^i$ where $\sigma^i \in \{0, \sigma^1, \sigma^2, ...\sigma^k\}$. Figure 3.11.2 illustrates the result of applying each of the operators in $M_2(\sigma)$ on the point $t$.

Consider the following state model.

$$x(t) = A(t)x(\gamma^{-1}t) + \sum_{s \in M_k(\sigma)} B(st)w(st) \tag{3.11.31}$$

$$E[w(t_1)w^T(t_2)] = \delta_{t_1-t_2} \tag{3.11.32}$$

Note that the "process noise" for this model, i.e. the summation in eq.(3.11.31), is now correlated with neighboring points of $t$, in particular the points $st$ where $s \in M_k(\sigma)$.

In order to apply our recursive smoothing method to processes described by this class of models, we must somehow transform the model to have uncorrelated process noise. A simple way of doing this is to perform state augmentation, where we define our state to be the set of points of the form $st$ where $s \in M_k(\sigma)$. We illustrate this idea for the following special case of eq.(3.11.31).

$$x(t) = A(t)x(\gamma^{-1}t) + B(t)w(t) + B(\sigma t)w(\sigma t) \tag{3.11.33}$$

where for notational convenience we let $\sigma = \sigma^1$.

Let us label all distinct pairs of points of the form $\{t, \sigma t\}$ by the new index $\tau$. That is, $\tau$ now indexes the pairs of points of the tree as illustrated in Figure 3.11.3, and can now in fact be interpreted as the index for a new dyadic tree whose nodes represent pairs of points on the old tree. Let us define the following quantities with respect to $\tau$.

$$z(\tau) = \begin{bmatrix} x(t) \\ x(\sigma t) \end{bmatrix} \tag{3.11.34}$$

$$\overline{w}(\tau) = \begin{bmatrix} w(t) \\ w(\sigma t) \end{bmatrix} \tag{3.11.35}$$

$$\overline{A}(\tau) = \begin{bmatrix} A(t) & 0 \\ A(\sigma t) & 0 \end{bmatrix} \tag{3.11.36}$$

$$\overline{B}(\tau) = \begin{bmatrix} B(t) & B(\sigma t) \\ B(\sigma t) & B(t) \end{bmatrix} \tag{3.11.37}$$

We now have the following state model for $z$.

$$z(\tau) = \overline{A}(\tau)z(\gamma^{-1}\tau) + \overline{B}(\tau)\overline{w}(\tau) \tag{3.11.38}$$

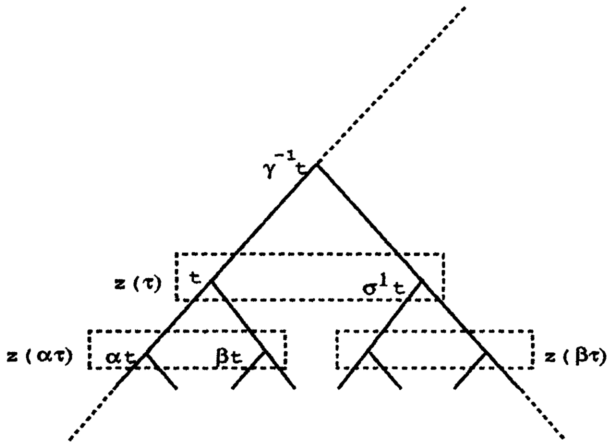

Figure 3.11.3: State Augmentation for Noises Correlated in Neighborhoods $\{t, \sigma t\}$

where

$$E[\overline{w}(\tau_1)\overline{w}^T(\tau_2)] = \overline{Q}(\tau_1)\delta_{\tau_1-\tau_2} \tag{3.11.39}$$

Note that our new model is precisely in the class of models we've considered in this chapter where $\tau$ is in fact the index of the nodes of a dyadic tree. Figure 3.11.3 illustrates how $z(\tau)$ is defined in terms of the tree on which $x(t)$ is defined. Note that the matrix $\overline{A}(t)$ is now singular and that we must use the following form for the dynamic matrix of our backward model.

$$P_z(\gamma^{-1}t)\overline{A}^T(t)P_z^{-1}(t) \tag{3.11.40}$$

Taking this last consideration into account, we can now apply our Rauch-Tung-Striebel algorithm on trees to smooth processes modeled by eq.(3.11.38).

Finally, we turn to the idea of developing an iterative algorithm for smoothing our processes. We consider the case of our tree model as in eq.(3.2.1). By using the local Markov structure of eq.(3.2.1), we develop an iterative algorithm for the computation of the optimal estimate of the process given multiscale measurements. As in the multigrid solution of partial differential equations, this approach may have significant computational advantages even if only the finest level estimates are actually desired and if only fine level measurements are available. We develop in detail the general approach to developing an iterative algorithm for the case of our tree models, then suggest what may be done to extend the approach to apply to general lattice models.

Let $Y$ denote the full set of measurements at all scales. Then, thanks to Markovian structure of our model in eq.(3.2.1) we have the following: For $m(t) = M$, the finest scale

$$\begin{aligned} E[x(t)|Y] &= E\left\{E[x(t)|x(\gamma^{-1}t), Y]|Y\right\} \\ &= E\left\{E[x(t)|x(\gamma^{-1}t), y(t)]|Y\right\} \end{aligned} \tag{3.11.41}$$

For $m(t) < M$

$$\begin{aligned} E[x(t)|Y] &= E\left\{E[x(t)|x(\gamma^{-1}t), x(\alpha t), x(\beta t), Y]|Y\right\} \\ &= E\left\{E[x(t)|x(\gamma^{-1}t), x(\alpha t), x(\beta t), y(t)]|Y\right\} \end{aligned} \tag{3.11.42}$$

The key now is to compute the inner expectations in eq.(3.11.41) and eq.(3.11.42), and to do this we need to view $x(\gamma^{-1}t)$, $x(\alpha t)$, and $x(\beta t)$ as *measurements* of $x(t)$. For the latter two, this comes directly from eq.(3.2.1). For $x(\gamma^{-1}t)$, however, we need the reverse-time version of eq.(3.2.1), i.e. eq.(3.2.5).

Let us now focus on the computation of the inner expectation of eq.(3.11.42). We can write the following equations for $x(\gamma^{-1}t)$, $x(\alpha t)$ $x(\beta t)$, $y(t)$.

$$y(t) = C(m(t))x(t) + v(t) \tag{3.11.43}$$

$$x(\gamma^{-1}t) = F(m(t))x(t) - A^{-1}(m(t))B(m(t))\tilde{w}(t) \tag{3.11.44}$$

$$x(\alpha t) = A(m(\alpha t))x(t) + B(m(\alpha t))w(\alpha t) \tag{3.11.45}$$

$$x(\beta t) = A(m(\beta t))x(t) + B(m(\beta t))w(\beta t) \tag{3.11.46}$$

which can be rewritten as

$$\mathcal{Y} = Hx(t) + \xi \tag{3.11.47}$$

where

$$H \triangleq \begin{bmatrix} C(m(t)) \\ F(m(t)) \\ A(m(\alpha t)) \\ A(m(\beta t)) \end{bmatrix}, \xi \triangleq \begin{bmatrix} v(t) \\ A^{-1}(m(t))B(m(t))\tilde{w}(t) \\ B(m(\alpha t))w(\alpha t) \\ B(m(\beta t))w(\beta t) \end{bmatrix} \tag{3.11.48}$$

and[3]

$$x(t) \perp \xi \tag{3.11.49}$$

Note the covariance of $\xi$ has the following structure.

$$\begin{aligned} E[\xi\xi^T] &= \begin{bmatrix} R(m(t)) & 0 & 0 & 0 \\ 0 & R_1(m(t)) & 0 & 0 \\ 0 & 0 & R_2(m(\alpha t)) & 0 \\ 0 & 0 & 0 & R_2(m(\alpha t)) \end{bmatrix} \qquad (3.11.50) \\ &\triangleq \mathcal{R} \end{aligned}$$

where

$$\begin{aligned} R_1(m(t)) &\triangleq A^{-1}(m(t))B(m(t))\tilde{Q}(t)B^T(m(t))A^{-T}(m(t)) && (3.11.51) \\ R_2(m(\alpha t)) &\triangleq B(m(\alpha t))B^T(m(\alpha t)) && (3.11.52) \\ &= B(m(\beta t))B^T(m(\beta t)) && (3.11.53) \end{aligned}$$

The inner expectation in eq.(3.11.42) can now be computed as follows.

$$\begin{aligned} E[x(t)|\mathcal{Y}] &= (P_x^{-1}(t) + H^T\mathcal{R}^{-1}H)^{-1}H^T\mathcal{R}^{-1}\mathcal{Y} \\ &= \mathcal{P}^{-1}\left\{K_1 y(t) + K_2 x(\gamma^{-1}t) + K_3 x(\alpha t) + K_4 x(\beta t)\right\} \qquad (3.11.54) \end{aligned}$$

where

$$\begin{aligned} K_1 &= C^T(m(t))R^{-1}(m(t)) && (3.11.55) \\ K_2 &= F^T(m(t))R_1^{-1}(m(t)) && (3.11.56) \\ K_3 &= A^T(m(\alpha t))R_2^{-1}(m(\alpha t)) && (3.11.57) \\ K_4 &= A^T(m(\beta t))R_2^{-1}(m(\alpha t)) && (3.11.58) \\ \mathcal{P} &= P_x^{-1}(t) + K_1C(m(t)) + K_2F(m(t)) + K_3A(m(\alpha t)) + K_4A(m(\beta t)) \\ & && (3.11.59) \end{aligned}$$

We can use a similar procedure for computing $E[x(t)|x(\gamma^{-1}t), y(t)]$ so that we can now carry out the outer expectations in eq.(3.11.41) and eq.(3.11.42) to yield the

[3] We denote $\perp$ to denote orthogonal in the sense that $a \perp b$ if $E[ab^T] = 0$.

following formulas for $\hat{x}(t) \triangleq E[x(t)|Y]$.
For $m(t) = M$

$$\hat{x}(t) = (\mathcal{P}')^{-1}\left\{C^T(m(t))R^{-1}(m(t))y(t) + F^T(m(t))R_1^{-1}(m(t))\hat{x}(\gamma^{-1}t)\right\} \quad (3.11.60)$$

For $m(t) < M$

$$\hat{x}(t) = \mathcal{P}^{-1}\left\{K_1 y(t) + K_2\hat{x}(\gamma^{-1}t) + K_3\hat{x}(\alpha t) + K_4\hat{x}(\beta t)\right\} \quad (3.11.61)$$

where

$$\mathcal{P}' = P_x^{-1}(t) + C^T(m(t))R^{-1}(m(t))C(m(t)) + F^T(m(t))R_1^{-1}(m(t))F(m(t)) \quad (3.11.62)$$

Thus, eq.(3.11.60) and eq.(3.11.61) are an implicit set of equations for $\{\hat{x}(t)|t \in T\}$. Note that the computation involved at each point on the tree involves only its three nearest neighbors and the measurement at that point. This suggests the use of a Gauss-Seidel relaxation algorithm for solving this set of equations. Note that the computations of all the points along a particular scale are independent of each other, allowing these computations to be performed in parallel. We could then arrange the computations of the relaxation algorithm so that we do all the computations at a particular scale in parallel, i.e. a Jacobi sweep at this scale, and the sweeps can be performed consecutively moving up and down the tree. The possibilities for parallelization are plentiful.

The following is one possible algorithm for computing smoothed estimates iteratively using our tree models. Let $X_k$ denote the vector of points along the $k$th level of the tree; $\hat{X}_k$ denotes the smoothed estimate of $X_k$.

**Algorithm 3.11.1** *Iterative Algorithm:*

```
1. Initialize $\hat{X}_0, ..., \hat{X}_M$ to 0.
2. Do Until Desired Convergence is Attained:
    (a) Compute in parallel eq.(3.11.60) for each entry of $\hat{X}_M$
    (b) For $k = M - 1$ to 0
```

```
        Compute in parallel eq.(3.11.61) for each entry of $\hat{X}_k$
    (c) For $k = 1$ to $M - 1$
        Compute in parallel eq.(3.11.61) for each entry of $\hat{X}_k$
```

Essentially, Algorithm 3.11.1 starts at the finest scale, moves sequentially up the tree to the coarsest scale, moves sequentially back down to the finest scale, then cycles through this procedure until convergence is attained. This bears resemblance to what is referred to in multigrid terminology[11] as a *V-cycle*. The issue of convergence is normally studied via the analysis of the global matrix formed from the set of implicit equations, eq.(3.11.60)-(3.11.61). However, our problem has a particular structure that allows us to give the following relatively simple argument for the convergence of a Gauss-seidel relaxation algorithm under any ordering of the local computations. We can think of the computation of $E[x(t)|Y]$ for all $t \in T$ as performing the minimization of a convex quadratic cost function with respect to $\{x(t) : t \in T\}$ and each Gauss-Seidel step is the local minimization with respect to a particular $x(t)$ with the remaining $x$'s held constant. Since each local minimization results in a reduction of the overall cost function and this function is convex, then the limit of the sequence of local minimizations results in the global minimization of the cost function.

Note that the development of our iterative algorithm relies essentially on the Markovianity of the model. In particular, it relies on the fact that the probability density of a point at a particular node conditioned on the remaining nodes is equal to the density of that point conditioned on a local neighborhood surrounding that point. In principle this idea can be applied to the case of our general lattice models in Chapter 2. For example the neighborhood on which we would be conditioning for the case of a 4-tap QMF filter is illustrated in Figure 3.11.4. Note that, unlike our transform approach which relies on the process' having a wavelet eigenstructure, our iterative approach can be applied to more general classes of lattice processes in which the model parameters need not be a function only of scale. For example, in the observation equation the matrix $C$ can vary arbitrarily on the lattice, allowing for the case of sparse data. Also, the class of models in which there is finite correlation in the process noise, a special case of which was described in the previous section, can

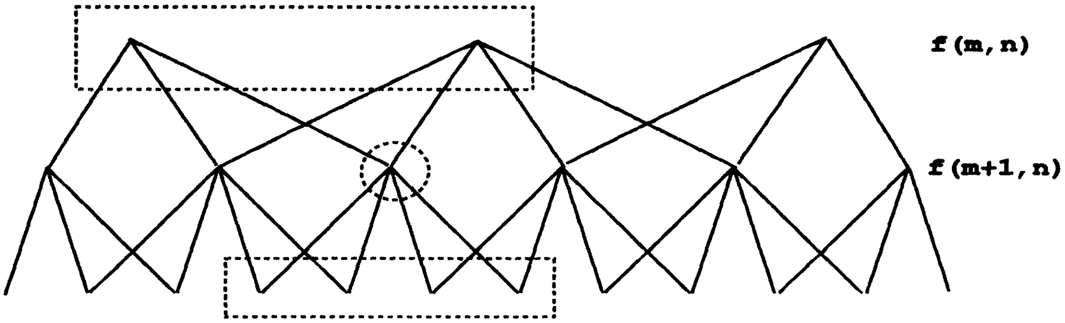

Figure 3.11.4: Neighborhood Involved in Local Computation of an Iterative Algorithm for the Case of a 4-tap QMF Lattice

be handled using an iterative approach.

## 3.12 Appendix 3A: ML Result

**Lemma 3.12.1** *Given the following recursions for all $m > M$*

$$P_{ML}^{-1}(m) = 2(A^{-1}P_{ML}(m+1)A^{-T} + A^{-1}BB^TA^{-T})^{-1} + C^TR^{-1}C \tag{3.12.63}$$

$$P_x^{-1}(m) = (A^{-1}P_x(m+1)A^{-T} - A^{-1}BB^TA^{-T})^{-1} \tag{3.12.64}$$

$$\begin{aligned} P^{-1}(m) &= 2(A^{-1}(I - BB^TP_x^{-1}(m+1))P(m+1)(I - P_x^{-1}(m+1)BB^T)A^{-T} \\ &+ A^{-1}B(I - B^TP_x^{-1}(m+1)B)B^TA^{-T})^{-1} + C^TR^{-1}C - P_x^{-1}(m) \end{aligned} \tag{3.12.65}$$

*where*

$$P_{ML}^{-1}(M) = 0 \tag{3.12.66}$$

$$P^{-1}(M) = P_x^{-1}(M) \tag{3.12.67}$$

*then the following is true for all $m > M$*

$$P^{-1}(m) = P_{ML}^{-1}(m) + P_x^{-1}(m) \tag{3.12.68}$$

**Proof**

For ease of notation we make the following definitions.

$$P \triangleq P(m+1) \tag{3.12.69}$$

$$P_{ML} \triangleq P_{ML}(m+1) \tag{3.12.70}$$

$$P_x \triangleq P_x(m+1) \tag{3.12.71}$$

We prove the lemma by induction, where we assume that

$$P^{-1} = P_{ML}^{-1} + P_x^{-1} \tag{3.12.72}$$

We now proceed to show the following.

$$P^{-1}(m) = P_{ML}^{-1}(m) + P_x^{-1}(m) \tag{3.12.73}$$

By substituting the right hand sides of eq.'s(3.12.63-3.12.65) into eq.(3.12.73) and performing cancellations we arrive at the following equivalent form for eq.(3.12.73).

$$\begin{aligned}(A^{-1}P_{ML}A^{-T} + A^{-1}BB^TA^{-T})^{-1} &+ (A^{-1}P_xA^{-T} - A^{-1}BB^TA^{-T})^{-1} \\ &= (A^{-1}(I - BB^TP_x^{-1})P(I - P_x^{-1}BB^T)A^{-T} \\ &+ A^{-1}B(I - B^TP_x^{-1}B)B^TA^{-T})^{-1} \end{aligned} \tag{3.12.74}$$

By left multiplying by $A^{-T}$ and right multiplying by $A^{-1}$ we arrive at

$$\begin{aligned}(P_{ML} + BB^T)^{-1} &+ (P_x - BB^T)^{-1} \\ &= ((I - BB^TP_x^{-1})P(I - P_x^{-1}BB^T) \\ &+ B(I - B^TP_x^{-1}B)B^T)^{-1} \end{aligned} \tag{3.12.75}$$

But by using the matrix inversion lemma,

$$(E + F)^{-1} = E^{-1} - E^{-1}(E^{-1} + F^{-1})^{-1}E^{-1} \tag{3.12.76}$$

we get that

$$\begin{aligned}((P_{ML} + BB^T)^{-1} + (P_x - BB^T)^{-1})^{-1} &= P_x - BB^T \\ &- (P_x - BB^T)(P_{ML} + P_x)^{-1}(P_x - BB^T) \end{aligned} \tag{3.12.77}$$

By substituting eq.(3.12.77) into eq.(3.12.75) then left and right multiplying by $(P_{ML} + BB^T)^{-1}$ we get the following equivalent form for eq.(3.12.75).

$$\begin{aligned}(P_x - BB^T)^{-1} - (P_{ML} + P_x)^{-1} &= P_x^{-1}PP_x^{-1} \\ &+ (P_x - BB^T)^{-1}B(I - B^TP_x^{-1}B)B^T(P_x - BB^T)^{-1} \end{aligned} \tag{3.12.78}$$

By using the matrix inversion lemma, eq.(3.12.76), we get that

$$\begin{aligned}(P_{ML} + P_x)^{-1} &= P_x^{-1} - P_x^{-1}(P_{ML}^{-1} + P_x^{-1})^{-1}P_x^{-1} \\ &= P_x^{-1} - P_x^{-1}PP_x^{-1} \end{aligned} \tag{3.12.79}$$

where the last equality follows from our induction assumption, eq.(3.12.72). By substituting eq.(3.12.79) into eq.(3.12.78) and performing cancellations we arrive at the following equivalent form for eq.(3.12.78).

$$(P_x - BB^T)^{-1} - P_x^{-1} = (P_x - BB^T)^{-1}B(I - B^T P_x^{-1} B)B^T (P_x - BB^T)^{-1} \quad (3.12.80)$$

By left multiplying eq.(3.12.80) by $(P_x - BB^T)$ we get

$$BB^T P_x^{-1} = B(I - B^T P_x^{-1} B)B^T (P_x - BB^T)^{-1} \quad (3.12.81)$$

By right multiplying eq.(3.12.81) by $(P_x - BB^T)$ we get

$$\begin{aligned} B(I - B^T P_x^{-1} B)B^T &= BB^T P_x^{-1}(P_x - BB^T) \\ &= BB^T (I - P_x^{-1} BB^T) \\ &= B(B^T - B^T P_x^{-1} BB^T) \\ &= B(I - B^T P_x^{-1} B)B^T \end{aligned} \quad (3.12.82)$$

□

## 3.13 Appendix 3B: Proof of Theorem 3.5.1

We define the following quantities.

$$\begin{aligned} Y_{M,t_0} &= \mathcal{H}_M X_{M,t_0} && (3.13.83) \\ x_{t_0} &= \Phi(t_0) X_{M,t_0} && (3.13.84) \\ \Phi(t_0) &= G I_{2^M} && (3.13.85) \end{aligned}$$

where $G$ is invertible(and thus $\Phi(t_0)$ is onto). We use $\mathcal{N}(\cdot)$ and $\mathcal{R}(\cdot)$ to denote nullspace and rangespace, respectively. A system is upward-reconstructible if given $Y_{M,t_0}$, $x_{t_0}$ is uniquely determined, i.e. $\mathcal{N}(\mathcal{H}_M) \subset \mathcal{N}(\Phi(t_0))$. We first prove the following lemma.

**Lemma 3.13.1** *For all $M$*

$$\mathcal{H}_M^T \mathcal{H}_M \Phi^T(t_0) = \Lambda \Phi^T(t_0) \qquad (3.13.86)$$

*where*

$$\Lambda = diag( \underbrace{\overline{\Lambda} ... \overline{\Lambda}}_{2^M \; times} ) \qquad (3.13.87)$$

*and $\overline{\Lambda}$ is some matrix.*

**Proof**

The structure of $\mathcal{H}_M^T \mathcal{H}_M$, which we denoted as $\mathcal{M}_M$, is described in a recursive fashion in eq.'s(3.6.20-3.6.24). We compute

$$\begin{aligned} \mathcal{M}_M \Phi^T(t_0) &= U(M,0)\Phi^T(t_0) \\ &= \begin{bmatrix} U(M,1)G^T I_{2^{M-1}}^T + 2^{M-1} T(M,0) G^T I_{2^{M-1}} \\ 2^{M-1} T(M,0) G^T I_{2^{M-1}} + U(M,1) G^T I_{2^{M-1}}^T \end{bmatrix} && (3.13.88) \end{aligned}$$

By repeating this procedure $M-1$ more times we get

$$U(M,0)\Phi^T(t_0) = \Lambda \Phi^T(t_0) \qquad (3.13.89)$$

where

$$\Lambda = \text{diag}( \underbrace{\overline{\Lambda} ... \overline{\Lambda}}_{2^M \; \text{times}} ) \qquad (3.13.90)$$

and

$$\overline{\Lambda} = \sum_{i=0}^{M-1} 2^{M-1-i} T(M,i) + U(M,M) \qquad (3.13.91)$$

□

We prove the following theorem.

**Theorem 3.13.1** *$\mathcal{N}(\mathcal{H}_M) \subset \mathcal{N}(\Phi(t_0))$ iff $\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)$ is invertible.*

**Proof**

a)

$$\mathcal{N}(\mathcal{H}_M) \subset \mathcal{N}(\Phi(t_0)) \longrightarrow \Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0) \text{ is invertible}$$

Assume $\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)$ is not invertible. Then for some $y \neq 0$, $y^T\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)y = 0$. This implies $\mathcal{H}_M\Phi^T(t_0)y = 0$. But the fact that $\Phi(t_0)$ is onto implies $\Phi^T(t_0)y \neq 0$. Furthermore, $\Phi^T(t_0)y \neq 0$ implies $\Phi(t_0)\Phi^T(t_0)y \neq 0$ since if it were true that $\Phi(t_0)\Phi^T(t_0)y = 0$, then $y^T\Phi(t_0)\Phi^T(t_0)y = 0$, which implies $\Phi^T(t_0)y = 0$. Thus, there exists a $z \neq 0$, namely $\Phi^T(t_0)y$, such that $\mathcal{H}_M z = 0$ and $\Phi(t_0) \neq 0$; i.e. it is not true that $\mathcal{N}(\mathcal{H}_M) \subset \mathcal{N}(\Phi(t_0))$.

b)

$$\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0) \text{ is invertible } \longrightarrow \mathcal{N}(\mathcal{H}_M) \subset \mathcal{N}(\Phi(t_0))$$

Assume that $\mathcal{N}(\mathcal{H}_M) \subset \mathcal{N}(\Phi(t_0))$ is false; i.e. there exists an $x$ such that $\mathcal{H}_M x = 0$ and $\Phi(t_0)x \neq 0$. Since $x \in \mathcal{R}(\Phi^T(t_0)) \oplus \mathcal{N}(\Phi(t_0))$, we can write $x = x_{\mathcal{R}(\Phi^T(t_0))} + x_{\mathcal{N}(\Phi(t_0))}$ where $x_{\mathcal{R}(\Phi^T(t_0))}$ is non-zero and $x_{\mathcal{N}(\Phi(t_0))}$ may or may not be non-zero. Since $\mathcal{H}_M x = 0$, $\mathcal{H}_M x_{\mathcal{R}(\Phi^T(t_0))} + \mathcal{H}_M x_{\mathcal{N}(\Phi(t_0))} = 0$, which means that $\mathcal{H}_M\Phi^T(t_0)y + \mathcal{H}_M x_{\mathcal{N}(\Phi(t_0))} = 0$ for some $y \neq 0$. Left multiplying by $\Phi(t_0)\mathcal{H}_M^T$, we get

$$\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)y + \Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M x_{\mathcal{N}(\Phi(t_0))} = 0 \qquad (3.13.92)$$

But from Lemma 3.13.1 and our definition for $\Phi(t_0)$, we get

$$\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M = \Phi(t_0)\Lambda^T = G\overline{\Lambda}^T[\ \underbrace{I...I}_{2^M \text{ times}}\ ] \qquad (3.13.93)$$

By substituting (3.13.93) into (3.13.92), we get

$$\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)y + G\overline{\Lambda}^T[\underbrace{I...I}_{2^M \text{ times}}]x_{\mathcal{N}(\Phi(t_0))} = 0 \tag{3.13.94}$$

for $y \neq 0$. But $x_{\mathcal{N}(\Phi(t_0))} \in \mathcal{N}(\Phi(t_0))$ implies that $\Phi(t_0)x_{\mathcal{N}(\Phi(t_0))} = 0$ or, using the definition of $\Phi(t_0)$, $G[\underbrace{I...I}_{2^M \text{ times}}]x_{\mathcal{N}(\Phi(t_0))} = 0$. But since $G$ is invertible, then $[\underbrace{I...I}_{2^M \text{ times}}]x_{\mathcal{N}(\Phi(t_0))} = 0$. Thus, eq.(3.13.94) collapses to $\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)y = 0$ for some $y \neq 0$, implying that $y^T\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)y = 0$ for some $y \neq 0$; i.e. $\Phi(t_0)\mathcal{H}_M^T\mathcal{H}_M\Phi^T(t_0)$ is not invertible.

□

# Chapter 4

# Applications and Numerical Examples

## 4.1 Introduction

In this chapter we explore the applicability of our multiscale estimation framework by performing a variety of numerical experiments. Having developed both theoretical results on and fast algorithms for the estimation of multiscale processes, we now provide numerical examples which demonstrate the utility of our framework for solving estimation problems involving both single scale as well as multiscale data.

First, we demonstrate the richness of our models in approximating well-known processes by comparing the performance of our smoother, using model parameters chosen so as to well-approximate the process, with the performance of standard smoothers. We give numerical examples in which we use our models to approximate both 1st order Gauss-Markov processes as well as 1/f-type processes and in each case compare the performance of the smoothers. In addition we use the Bhattacharyya distance measure as a way of evaluating how well our models approximate these processes. We also use this to show the effect of wavelet filter order on approximating processes. Our examples indicate that our multiscale models do rather well in smoothing different classes of processes. Besides being important from a modeling perspective, this is also an important consideration from a computational perspective given the fact that our multiscale algorithms based on these models are both efficient and highly parallelizable.

We also study examples of using multiscale measurements in doing estimation.

We see that the use of coarse-scale data can aid in estimating features which are not discernible using fine-scale data of poor quality. We then use our tree models to handle the case in which our data is sparsely distributed. In this case we show how coarse scale data of full-coverage can be used to interpolate features which are not directly measured with the sparse data. Note that our approach is trivially extendible to 2D, and thus, has considerable potential for solving multiscale problems in 2D, including the case for example of fusing spatially distributed data of various resolution and coverage.

## 4.2 Processes and Multiscale Models

In this section we describe the two types of processes we consider throughout the examples in this chapter (excluding the section on optical flow). They are the class of stationary Gauss-Markov processes and the class of $1/f$ processes. We also describe the specific multiscale models we use to approximate these processes and give examples of sample paths generated using these models to give a qualitative idea of the type of signals they represent.

We begin by describing the type of Gauss-Markov process which we use in our examples. Consider the following stationary 1st-order Gauss-Markov process.

$$\dot{x}(t) = -\beta x(t) + w(t) \tag{4.2.1}$$

$$E[x^2(t)] = 1 \tag{4.2.2}$$

This process has the following correlation function and associated power spectral density function.

$$\phi_{xx}(\tau) = e^{-\beta|\tau|} \tag{4.2.3}$$

$$S_{xx}(\omega) = \frac{2\beta}{\omega^2 + \beta^2} \tag{4.2.4}$$

In the numerical examples that follow we use a discretized version of eq.(4.2.1). In particular we use a sampled version of eq.(4.2.1) in which the sampling interval is small enough to minimize any aliasing effects. We choose $\beta = 1$ and take the sampling rate to be twice $\omega_0$ where $S_{xx}(\omega_0) = .002$, $S_{xx}(\omega)$ being the power spectral density function of $x(t)$. This yields a sampling interval of $\Delta = \pi/\omega_0$ where $\omega_0 = 30$. Our discretized model is as follows.

$$x(t+1) = \alpha x(t) + w(t) \tag{4.2.5}$$

$$E[x^2(t)] = 1 \tag{4.2.6}$$

$$\alpha = e^{-\beta\Delta} \simeq .9006 \tag{4.2.7}$$

We consider the following measurements of $x(t)$.

$$y(t) = x(t) + v(t) \tag{4.2.8}$$

$$E[v^2(t)] = R \tag{4.2.9}$$

$$Y = \{y(t)|t = 0, \ldots N-1\} \tag{4.2.10}$$

In the examples that follow we take the interval length $N = 128$.

While we would expect our models to yield good approximations for 1/f processes given recent results on this[55, 25], it comes somewhat as a surprise that we are able to estimate so well noisy Gauss-Markov processes using our models. Our ability to use the wavelet transform as a basis to *model* Gauss-Markov processes, however, is supported by recent results by Beylkin et al[9] in using the wavelet transform to *analyze* classes of kernels that share properties with the covariance functions of Gauss-Markov processes. Their main result is the transformation of kernels of the Calderon-Zygmund class into matrices with extremely sparse sets of non-negligible elements, providing a way of doing large-scale matrix-vector multiplications extremely fast. A feature of this work is the property that the number of vanishing moments determines the decay of the off-diagonal elements of the transformed kernels from their diagonals. The higher the number of vanishing moments the faster the decay, where an increase in the number of vanishing moments necessitates an increase in the order of the QMF filter, $h(n)$.

In the work of Golden[27] the covariance of Gauss-Markov processes is represented in terms of various wavelet bases. The examples in Figures 4.2.1,4.2.2,4.2.3 illustrate the effects of transforming an example of one of these kernels using the wavelet transform. Figure 4.2.1 is a gray-scale image of the covariance matrix of a stationary first

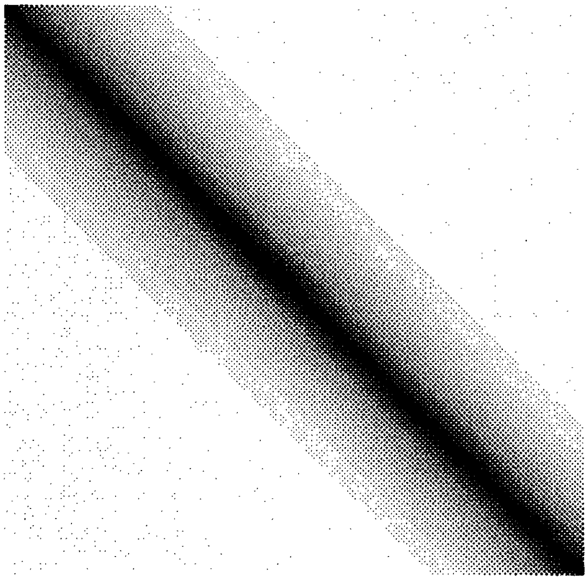

Figure 4.2.1: Covariance Matrix of a Stationary Gauss-Markov Process

order Gauss-Markov process defined on a finite interval, corresponding to the model in eq.(4.2.5). Note that these examples plot the correlation coefficients of the process, i.e. the covariance between two points normalized by the product of the standard deviation at each point. The diagonal of the matrix is thus is unity, and the off-diagonal terms decay exponentially away from the diagonal. In [27] this covariance matrix is transformed using various wavelet bases, i.e. the matrix undergoes a similarity transformation with respect to the basis representing the wavelet transform based on a variety of QMF filters, $h(n)$. This transformation corresponds essentially to the separable form of the 2D wavelet transform[36]. Figures 4.2.2,4.2.3 are the images of the covariance matrix in Figure 4.2.1 transformed using QMF filters of length 2 and 8, respectively. Note that aside from the finger-like patterns in these images, the off-diagonal elements are essentially zeroed. The finger patterns correspond to correlations between wavelet coefficients at different scales which share the same location in the interval. Note that even these correlations are weak.

The low level of inter-scale correlation in the wavelet representation of the Gauss-Markov process as illustrated in Figures 4.2.2 and 4.2.3 motivates the approximation

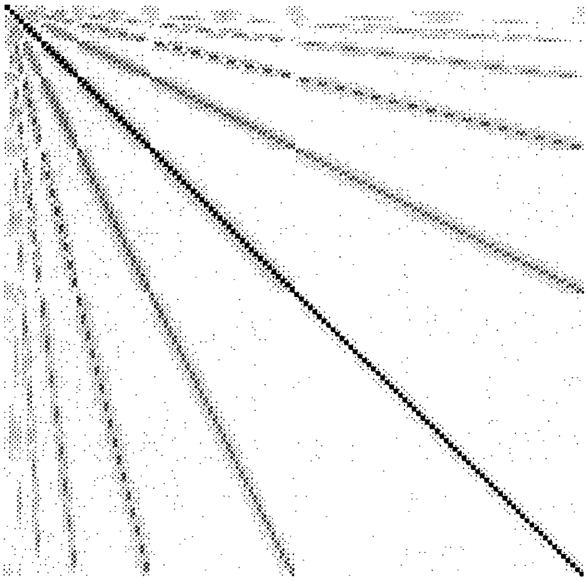

Figure 4.2.2: Representation of the Stationary Gauss-Markov Process in a Wavelet Basis using a 2-Tap QMF filter

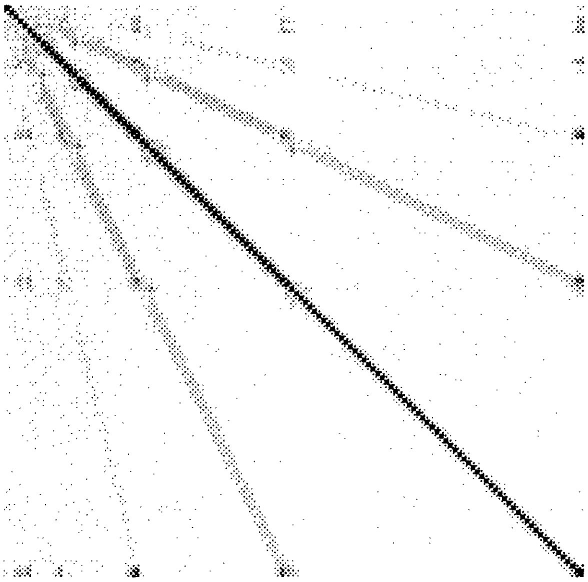

Figure 4.2.3: Representation of the Stationary Gauss-Markov Process in a Wavelet Basis using an 8-Tap QMF filter

of the wavelet coefficients of this process as uncorrelated. This results in a lattice model precisely as defined in eq.'s(2.2.1-2.2.3). We use this model as well as several other state models defined on trees as an approximation to the Gauss-Markov process in order to do fixed interval smoothing.

In particular, one class of models which we consider as approximations for both Gauss-Markov and $1/f$ processes is obtained precisely in the manner just described. That is, we construct models as in eq.'s(2.2.1-2.2.3) where the wavelet coefficients are assumed to be mutually uncorrelated. The results of Beylkin and Golden indicate that this model does very well in approximating Gauss-Markov processes. In this case the variances of the wavelet coefficients, $w(m)$ in eq.'s(2.2.1-2.2.3), are determined by doing a similarity transform on the covariance matrix of the process under investigation using a wavelet transform based on the Daubechies FIR filters[21]. In particular if $P_x$ denotes the true covariance matrix of the process, $V$ the diagonal matrix of wavelet coefficient variances, and $W$ is the wavelet transform matrix, then

$$\Lambda = WP_xW^T \tag{4.2.11}$$

$$V = WP_{approx}W^T \tag{4.2.12}$$

Thus, this approximate model corresponds to assuming that $\Lambda$ is diagonal (i.e. to neglecting its off-diagonal elements). For reference we give the following numerical values for the various QMF filters $h(n)$ we will use throughout this chapter, including the 2-tap Haar filter as well as the 4-tap, 6-tap, and 8-tap Daubechies filters.

$$2\ tap = \begin{bmatrix} 0.70710678118655 & 0.70710678118655 \end{bmatrix} \tag{4.2.13}$$

$$\begin{aligned} 4\ tap = \Big[\ & 0.48296291314500 \quad 0.83651630373800 \quad 0.22414386804200 \\ & -0.12940952255100 \ \Big] \end{aligned} \tag{4.2.14}$$

$$\begin{aligned} 6\ tap = \Big[\ & 0.33267055295000 \quad 0.80689150931100 \quad 0.45987750211800 \\ & -0.13501102001000 \quad -0.08544127388200 \quad 0.03522629188200 \ \Big] \end{aligned} \tag{4.2.15}$$

$$\begin{aligned} 8\ tap = \Big[\ & 0.23037781330900 \quad 0.71484657055300 \quad 0.63088076793000 \\ & -0.02798376941700 \quad -0.18703481171900 \quad 0.03084138183600 \\ & 0.03288301166700 \quad -0.01059740178500 \ \Big] \end{aligned} \tag{4.2.16}$$

Note that in adapting the wavelet transform to the finite interval we have chosen the method of using cyclic convolutions at each scale as described in Chapter 2. In this case the number of points at each scale is half the number of points at the next finest scale.

For the case of $1/f$-type processes the result of Wornell on approximating these processes using wavelets suggests that our lattice models can be easily used to model a $1/f$ process with parameter $\gamma$. In particular by simply taking the model of eq.'s(2.2.1-2.2.3) and setting the variances of $w(m)$ so that they obey the power law $2^{-\gamma m}$, we get *precisely* the model of Wornell. We will give examples in the following sections of smoothing noisy $1/f$ processes using this modeling scheme both for single scale data as well as multiple scale data.

Finally, we describe the particular tree models we use in some of our examples as approximate models for Gauss-Markov and $1/f$ processes. The simplest model we consider is the following scalar tree model which is assumed to be in steady state.

$$\begin{aligned} x_t &= a x_{\gamma^{-1}t} + w_t && (4.2.17) \\ E[w_t^2] &= 1 && (4.2.18) \\ E[x_t^2] &= p && (4.2.19) \end{aligned}$$

We refer to this model as the *2-parameter* model since it is parametrized by the parameters $a$ and $p$. In order to broaden the class of processes we might want to consider, we also consider a tree model which includes a parameter that controls the behavior of the process noise $w_t$. In particular we consider a model which has the property that its fine-scale variations have smaller variance than its coarse-scale variations. We know for example that $1/f$ processes can be synthesized using the wavelet transform as in [55], where the variances of the wavelet coefficients decay geometrically with finer scales. Thus, we consider the following scalar tree model.

$$\begin{aligned} x_t &= a x_{\gamma^{-1}t} + 2^{-\frac{\delta m(t)}{2}} w_t && (4.2.20) \\ E[w_t^2] &= 1 && (4.2.21) \\ E[x_{t_0}^2] &= p_0 && (4.2.22) \end{aligned}$$

We refer to this model as the *3-parameter* model since it parametrized by the parameters $a$, $p_0$, and $\delta$ where $\delta$ controls the amount of scaling in the variance of the process noise from scale to scale.

We use both the 2-parameter as well as the 3-parameter tree model to approximate both Gauss-Markov as well as $1/f$ processes. Our method for choosing the parameters of these models will be described in the next section.

Before we go on to present examples of smoothing processes using our multiscale models, we give examples of sample paths of our processes in order to give a qualitative idea of the type of processes our models represent. We first consider examples of our lattice processes and in particular we take our model to be the one in eq.'s(2.2.1-2.2.3), where the variances of $w(m)$ are set so that they obey the power law $2^{-\gamma m}$; as noted before in this case we get precisely the model of Wornell for processes whose measured spectrum is $1/|w|^{\gamma}$. Figure 4.2.4 shows the plots of two different sample paths, one corresponding to a process with spectrum $1/|w|^{0.1}$, the other with spectrum $1/|w|^{2}$. Both are generated using a 4-tap Daubechies filter. Note the presence of more high frequency energy in the process with $\gamma = .1$ as compared with the process with $\gamma = 2$. This illustrates in the time-domain the contrast between two processes of different rates of spectral decay. In Figure 4.2.5 we compare processes with spectrum $1/|w|$ generated using the 4-tap Daubechies filter and the 8-tap Daubechies filter. Note that from the sample paths it is difficult to discern any significant contrasting features between the two processes. Furthermore, it is even more difficult to assess the impact that the QMF filter order has on the approximation of processes. This issue will become clearer in following sections in the context of smoothing processes using lattice model approximations.

Finally, we show some sample paths of the tree processes which we consider. Figure 4.2.6 illustrates sample paths of our 2-parameter model for the case of $a = .9$ and the case of $a = .5$, where in each case $p = 5.26$. As we would expect, the points in the process corresponding to $a = .5$ appear much more uncorrelated compared with points in the process with $a = .9$. Recall from Chapter 3 that our 2-parameter model actually describes a process in which the correlation between any two points $t, s$ is proportional to $a^{d(t,s)}$ where $d(t, s)$ denotes the number of branches along the shortest path between $t$ and $s$ (i.e. an *isotropic* process). Figure 4.2.7 illustrates the

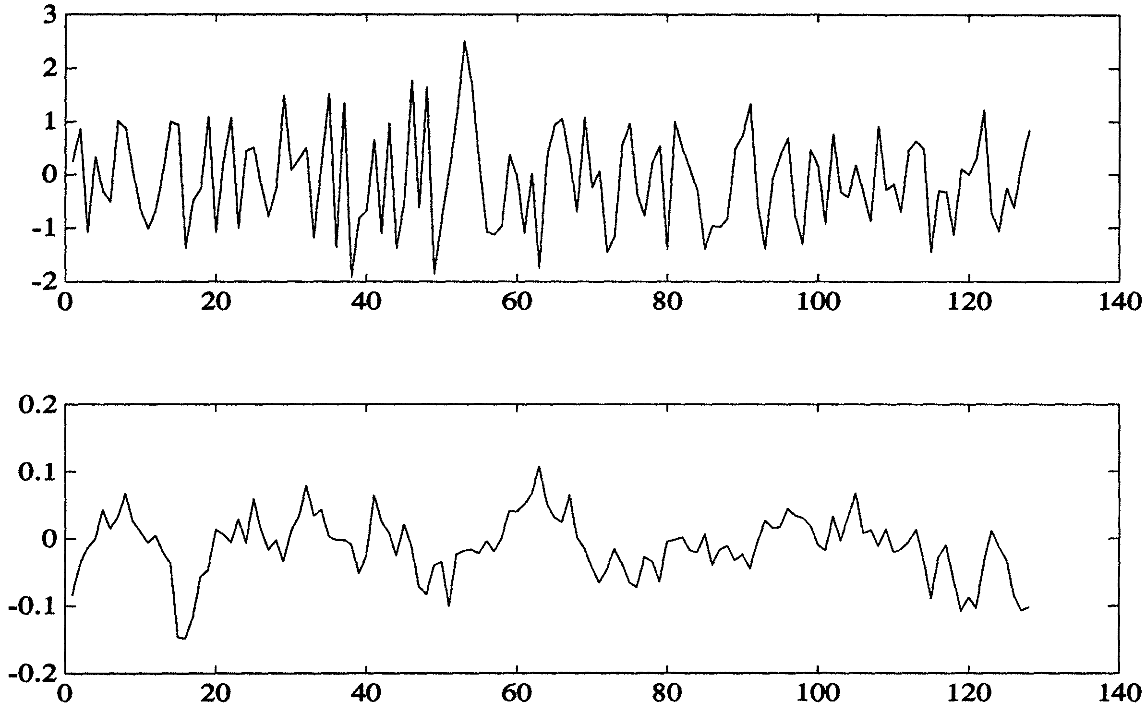

Figure 4.2.4: (a) Sample Path of a Lattice Process with Spectrum $1/|w|^{0.1}$ (b) Sample Path of a Lattice Process with Spectrum $1/|w|^2$

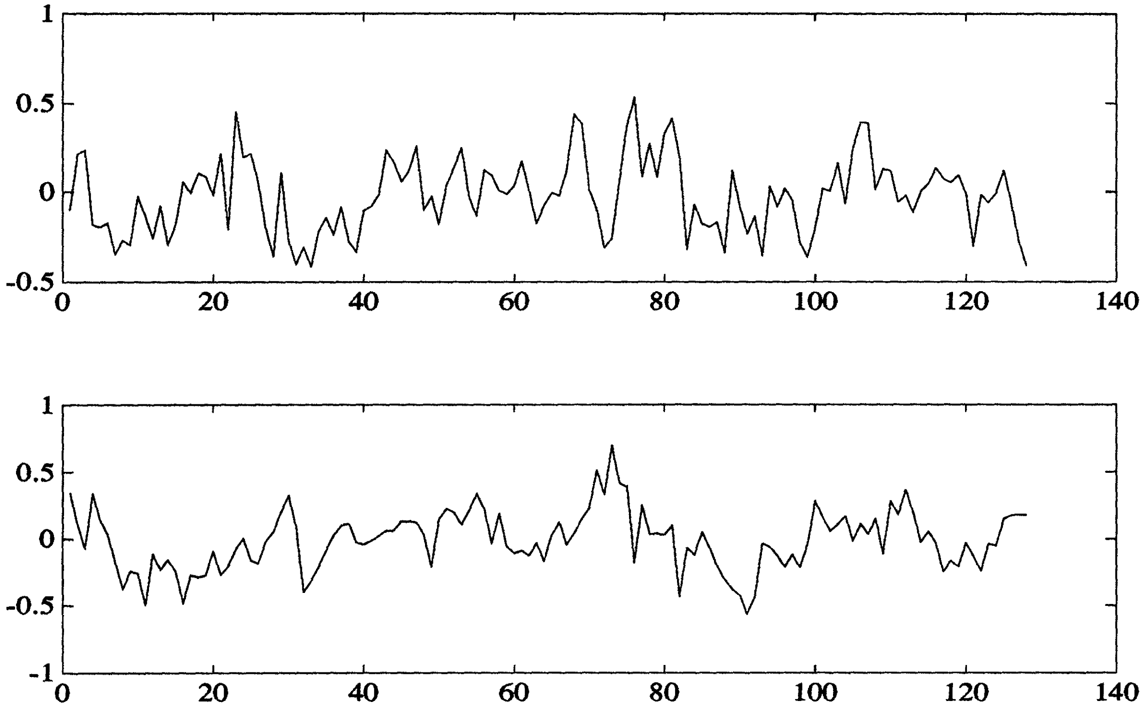

Figure 4.2.5: (a) Sample Path of a Lattice Process with Spectrum $1/|w|$ Generated Using a 4-tap Daubechies Filter (b) Sample Path of a Lattice Process with Spectrum $1/|w|$ Generated Using an 8-tap Daubechies Filter

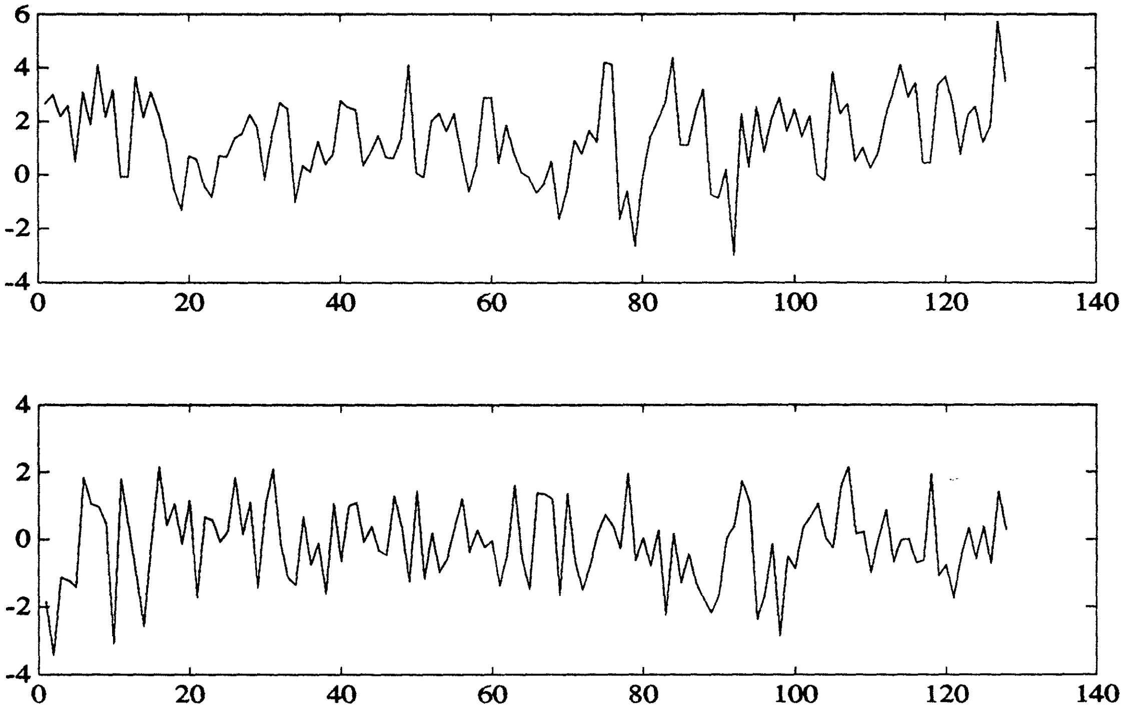

Figure 4.2.6: (a) Sample Path of a 2-parameter Scalar Tree Process with $a = .9$ (b) Sample Path of a 2-parameter Scalar Tree Process with $a = .5$

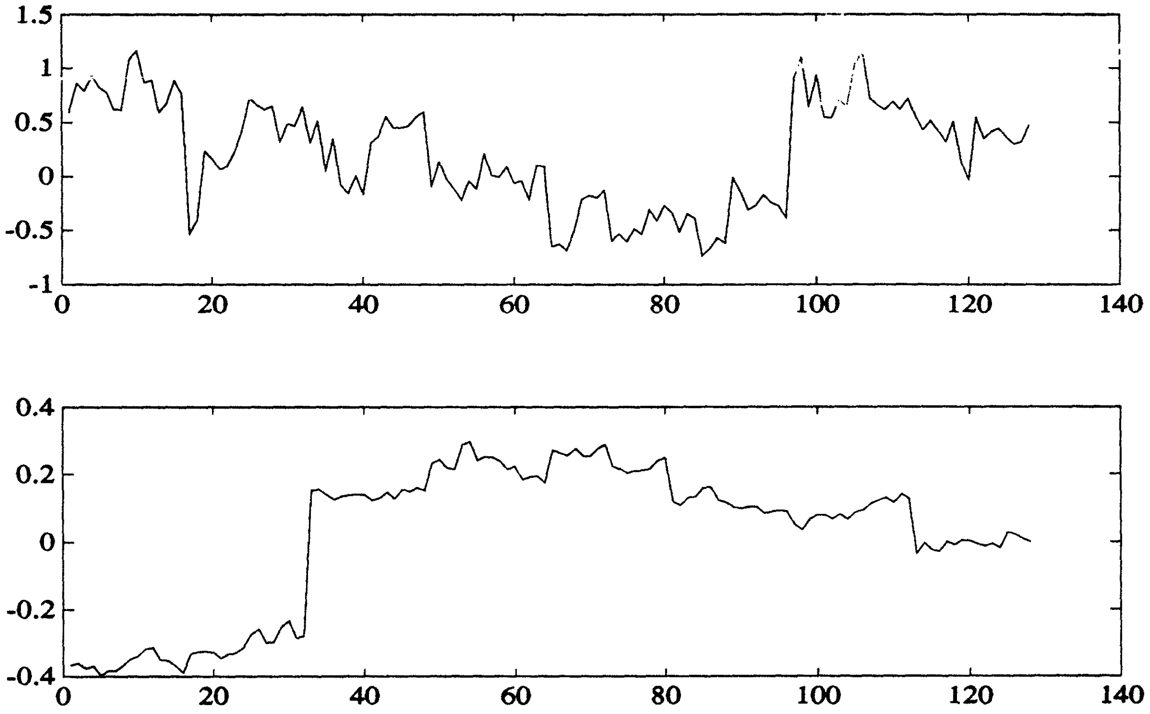

Figure 4.2.7: (a) Sample Path of a 3-parameter Scalar Tree Process with $a = .9$, $p_0$=1, $\sigma = 1$ (b) Sample Path of a 2-parameter Scalar Tree Process with $a = .9$, $p_0$=1, $\sigma = 2$

sample paths of our 3-parameter model for the case of $\delta = .9$ and the case of $\delta = .5$, where in each case $a = .9$ and $p_0 = 1$. Note that the sample path corresponding to $\sigma = 1$ appears to have more high frequency energy than the sample path with $\sigma = 2$. This suggests that our 3-parameter tree model can be used to describe processes with $1/f$-like features; in fact we will show in the following sections that this model can be used to describe both Gauss-Markov processes as well as $1/f$ processes.

## 4.3 Smoothing Processes Using Multiscale Models

In this section we present examples in which we use our models to smooth processes embedded in noise. These examples not only demonstrate the utility of our framework, but by smoothing processes outside of our class of processes we provide an indication of how rich our models are in approximating other processes. The two types of processes we consider are 1st-order Gauss-Markov processes and $1/f$ processes. We give examples showing how our models can be used to estimate rather well versions of both of these types of processes given noisy measurements. We provide several examples of sample-paths of noisy Gauss-Markov and $1/f$ processes and their smoothed versions using both lattice and tree models. Since our data is generated from processes other than those described by our model, the use of smoothers based on our multiscale models results in *suboptimal* estimates. What will become apparent in the examples is the strikingly good performance of our suboptimal smoothers, giving support to the idea that our models can well-approximate other processes.

We study the effects of performing fixed-interval smoothing of processes, both 1st-order Gauss-Markov and $1/f$-type processes, using a variety of suboptimal smoothers based on several of our multiscale models as described in the previous section. We take the *optimal* smoothed estimate of a process $x(t)$ to be the conditional mean of the process conditioned on data over a finite interval.

$$\hat{x}_s(t) \triangleq E[x(t)|Y] \tag{4.3.23}$$

where

$$\begin{aligned} y(t) &= x(t) + v(t) && (4.3.24) \\ E[v^2(t)] &= R && (4.3.25) \\ Y &= \{y(t)|t = 0, ...N-1\} && (4.3.26) \end{aligned}$$

and the optimal smoothing error is given by

$$\begin{aligned} \Sigma_{opt} &\triangleq E[(x - \hat{x}_s)(x - \hat{x}_s)^T] && (4.3.27) \\ x &= \{x(t)|t = 0, ...N-1\} && (4.3.28) \\ \hat{x}_s &= \{\hat{x}_s(t)|t = 0, ...N-1\} && (4.3.29) \end{aligned}$$

Conceptually, the idea of using a suboptimal smoother based on a model that does not exactly describe the data is as follows. Suppose we postulate a model for a zero-mean process, $z(t)$, where $z(t)$ is an approximation of $x(t)$, with the following second-order statistics (in our case $z(t)$ represents either a lattice or a tree process).

$$\begin{aligned} Z &\triangleq \{z(t)|t=0,...N-1\} &(4.3.30)\\ E[ZZ^T] &\triangleq P_Z &(4.3.31)\end{aligned}$$

Consider the following measurements of $z(t)$.

$$\begin{aligned} \overline{y}(t) &= z(t)+v(t) &(4.3.32)\\ E[v^2(t)] &= R &(4.3.33)\\ \overline{Y} &= \{\overline{y}(t)|t=0,...N-1\} &(4.3.34)\end{aligned}$$

The best estimate of $z(t)$ based on $\overline{Y}$ is the conditional mean of $z(t)$ conditioned on $\overline{Y}$. This mean is a linear function of $\overline{Y}$ of the following form.

$$\begin{aligned} \hat{z}_s(t) &\triangleq E[z(t)|\overline{Y}] &(4.3.35)\\ \hat{z}_s &\triangleq \{\hat{z}_s(t)|t=0,...N-1\} &(4.3.36)\\ &= L_z\overline{Y} &(4.3.37)\\ &= P_Z(P_Z+RI)^{-1}\overline{Y} &(4.3.38)\end{aligned}$$

Suppose now that we apply the estimation operator $L_z$ to data corresponding to our process $x(t)$. This would result in the following *suboptimal* estimator with its corresponding suboptimal error covariance.

$$\begin{aligned} \hat{x}_{sub} &\triangleq L_zY &(4.3.39)\\ x &\triangleq \{x(t)|t=0,...N-1\} &(4.3.40)\\ \Sigma_{sub} &\triangleq E[(x-\hat{x}_{sub})(x-\hat{x}_{sub})^T] &(4.3.41)\\ &= (I-L_z)P_x(I-L_z)^T+L_zRL_z^T &(4.3.42)\end{aligned}$$

The structure of $\Sigma_{sub}$ motivates one criterion for selecting the parameters of our multiscale models. In the case of our tree models our procedure for fitting model parameters to data is to optimize the parameters of our model so as to minimize the average suboptimal error variance, i.e. the trace of $\Sigma_{sub}$. The suboptimal error variance is a natural choice of optimization criterion since we are concerned ultimately with the smoothing of noisy data. In particular we choose the optimal parameters of our model as follows.

$$\theta \triangleq \text{tree model parameters} \tag{4.3.43}$$

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \quad \text{trace } \Sigma_{sub} \tag{4.3.44}$$

Note that $\Sigma_{sub}$ is a function of the true process (the process to be approximated) covariance, $P_x$, and the noise power, $R$, both of which we assume to be known. In all of the following examples, including the ones in the section on sensor fusion, we use the average suboptimal error criterion to optimize both our 2-parameter and 3-parameter tree model parameters to approximate processes, both of the Gauss-Markov as well as the $1/f$ type.

As further evidence that our models can be used effectively to approximate both Gauss-Markov processes and 1/f processes, especially in the context of estimation in noise, we provide examples using the Bhattacharyya distance measure as a way of showing the relative accuracy of our approximations. Finally, we give several examples comparing estimation performance for various order wavelets (i.e. various order QMF filters). These examples provide a partial answer to the question of whether increasing the wavelet order buys one anything in terms of modeling power and/or estimation performance.

### 4.3.1 Smoothing Gauss-Markov Processes

In this section we give a variety of numerical examples demonstrating the performance of our multiscale models in smoothing Gauss-Markov processes. We focus on the case of a single scale of data at the finest scale and consider both lattice models as well as tree models.

We begin by using our lattice models. In Fig.'s 4.3.8-4.3.14 we compare the performance of the optimal estimator, the one that minimizes the mean-square error, with the performance of our suboptimal estimator based on lattice models for both 2-tap and 8-tap Daubechies filters. In these examples the measurement noise variance $R = .5$; i.e. the data is of $SNR = 1.4142$. Note the strikingly similar performances

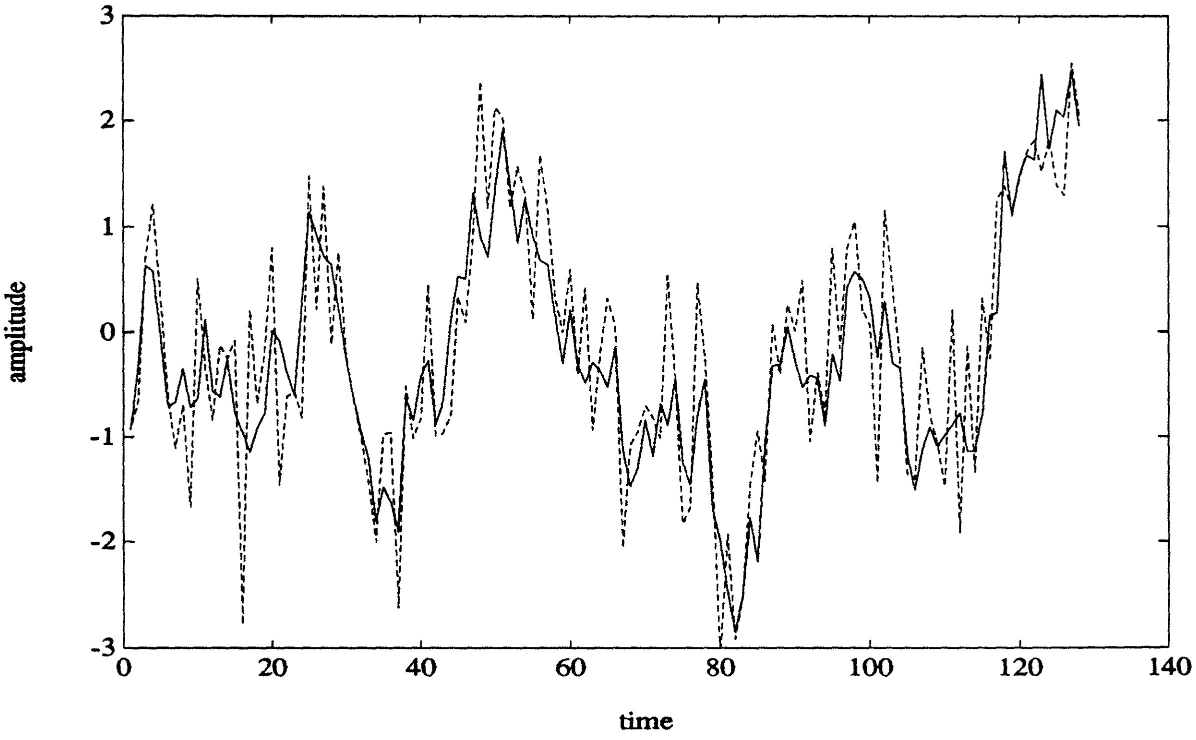

Figure 4.3.8: Sample Path of a Stationary Gauss-Markov Process (solid) and Its Noisy Version with SNR=1.4142 (dashed)

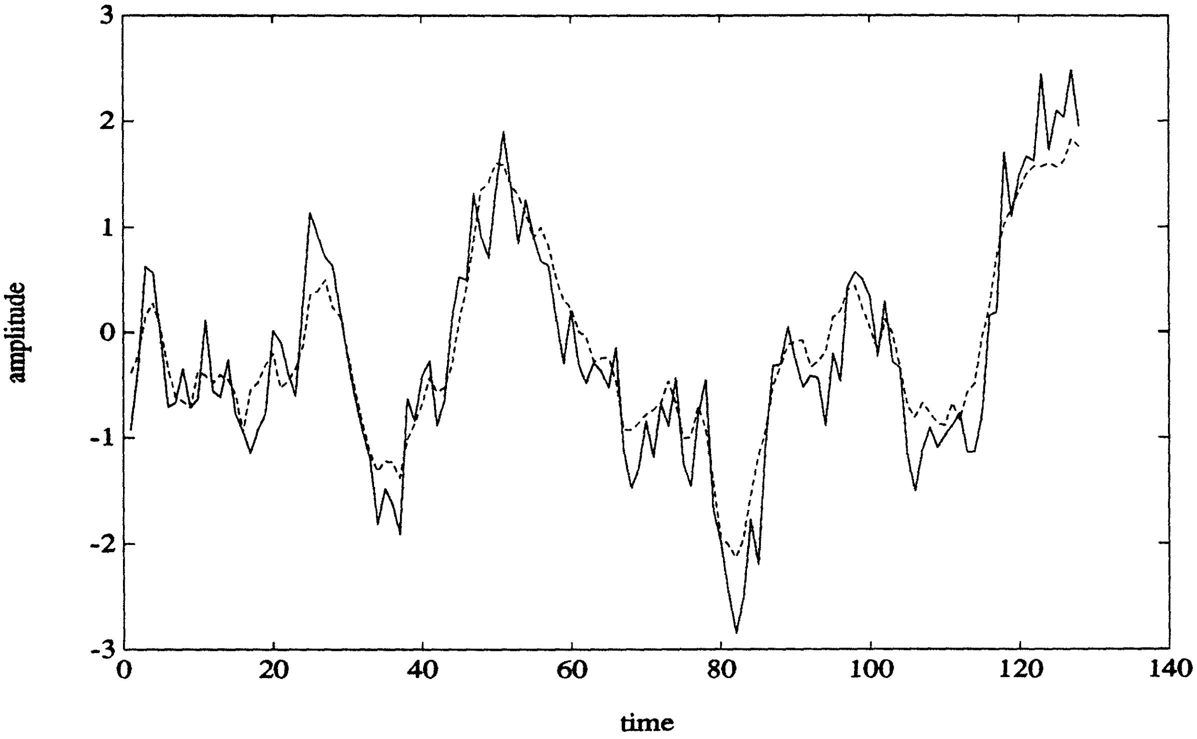

Figure 4.3.9: Stationary Gauss-Markov Process (solid) and Its Smoothed Version (dashed) Using Standard Minimum Mean-Square Error Smoother (Data of SNR=1.4142)

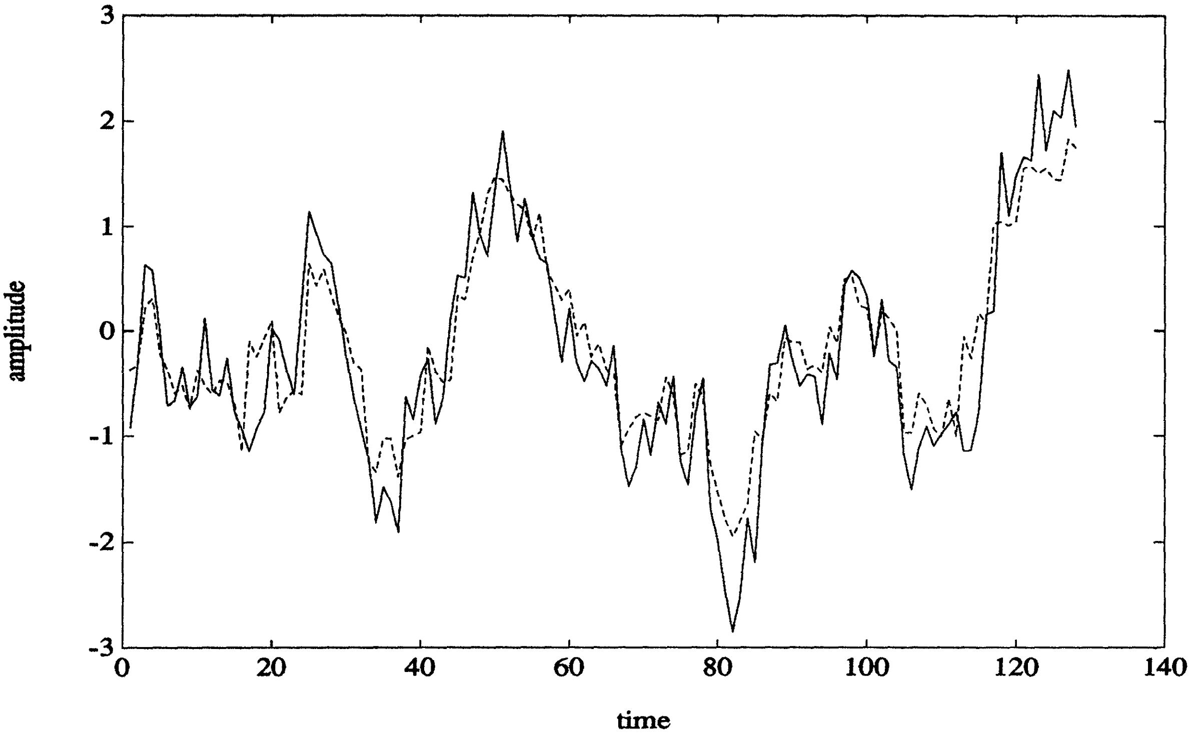

Figure 4.3.10: Stationary Gauss-Markov Process (solid) versus Multiscale Smoother Using 2-Tap (dashed) (Data of SNR=1.4142)

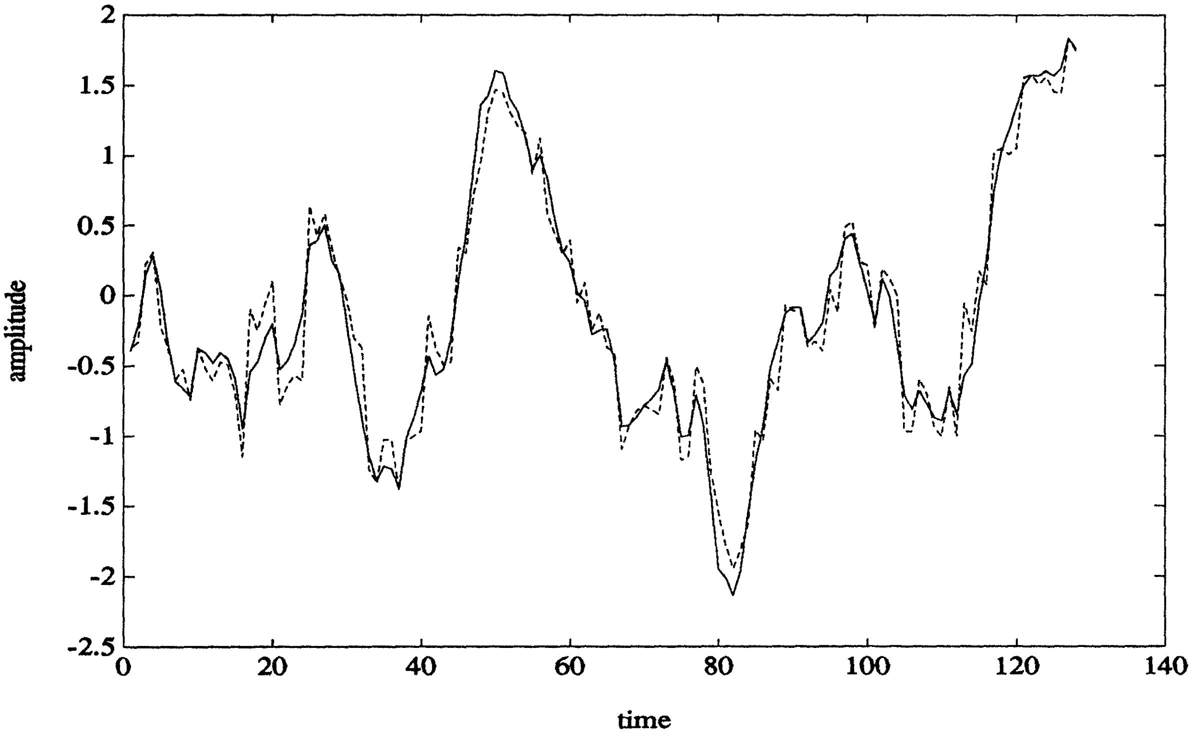

Figure 4.3.11: Standard Minimum Mean-Square Error Smoother (solid) versus Multiscale Smoother Using 2-Tap (dashed) (Data of SNR=1.4142)

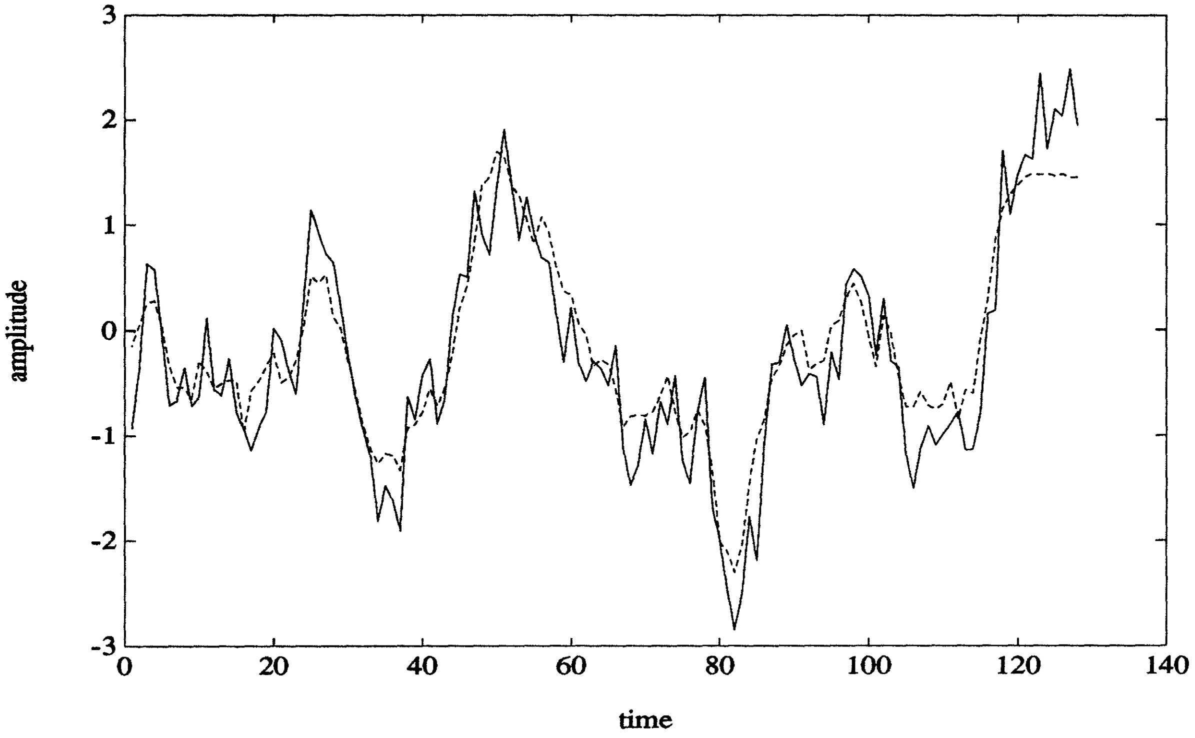

Figure 4.3.12: Stationary Gauss-Markov Process (solid) versus Multiscale Smoother Using 8-Tap (dashed) (Data of SNR=1.4142)

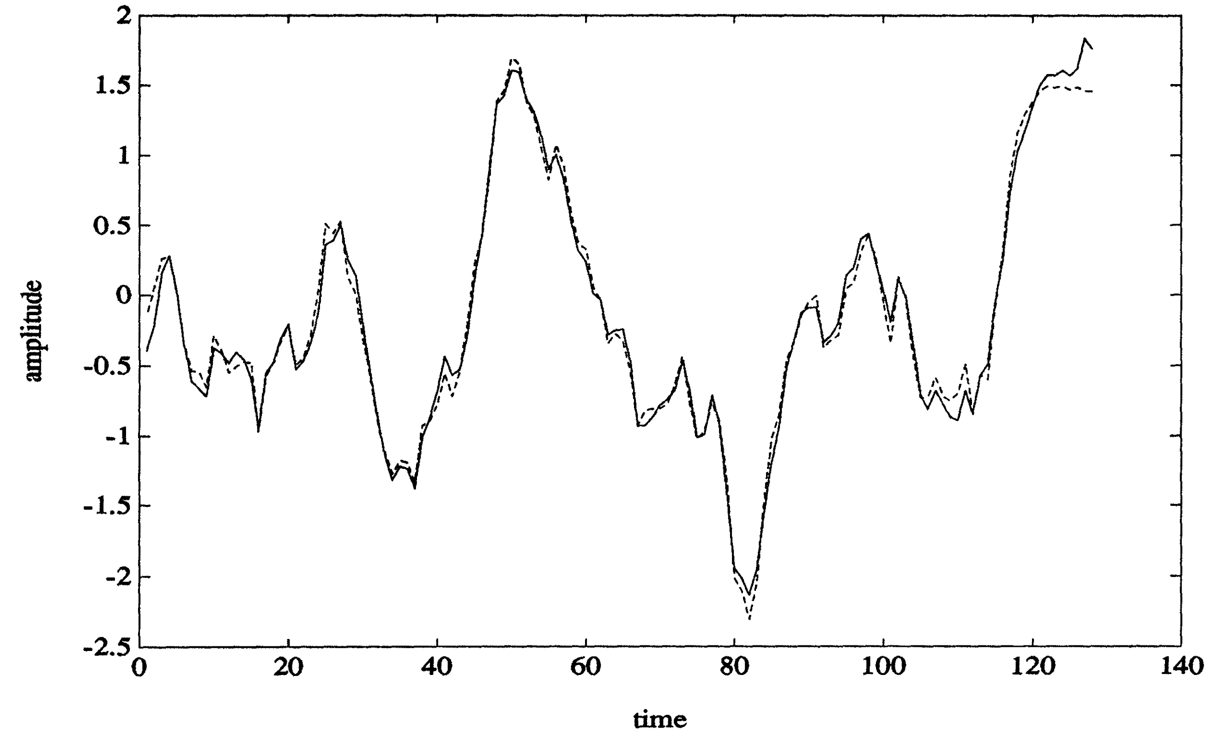

Figure 4.3.13: Standard Minimum Mean-Square Error Smoother (solid) versus Multiscale Smoother Using 8-Tap (dashed) (Data of SNR=1.4142)

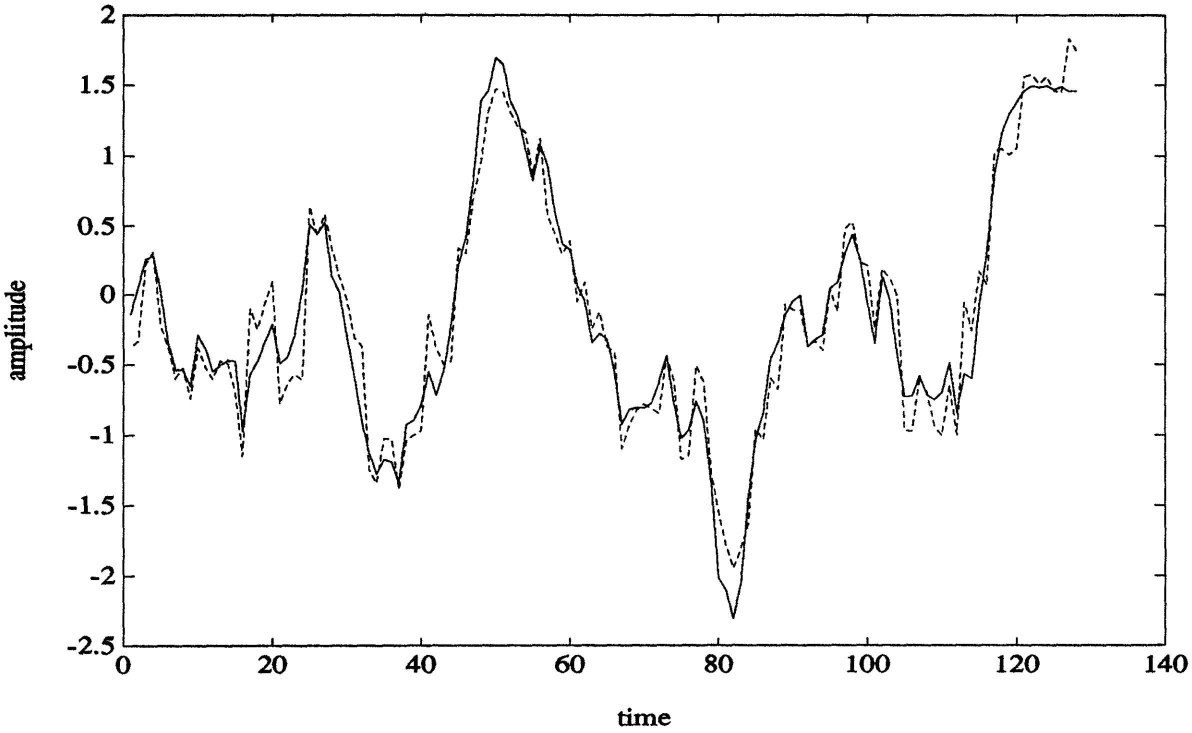

Figure 4.3.14: Multiscale Smoother Using 8-Tap (solid) versus Multiscale Smoother Using 2-Tap (dashed) (Data of SNR=1.4142)

of the optimal and suboptimal smoothers, as illustrated in Figure 4.3.11 for the case of the 2-tap lattice smoother. From visual inspection of the results of the two smoothers it is difficult to say which does a better job of smoothing the data; it seems one could make a case equally in favor of the standard smoother and the lattice-model smoother. The similarity in performance of the optimal smoother and our lattice smoothers is even more dramatic for the case of the 8-tap smoother as illustrated in Figure 4.3.13. Figure 4.3.14 compares the performance of our 8-tap and 2-tap smoothers. The difference in performance between these two suboptimal smoothers will be precisely quantified shortly.

Note that although the standard smoother results in a smaller average smoothing error (the trace of $\Sigma_{opt}$ divided by the number of points in the interval), it seems the average error of our lattice-model smoothers is not that much larger. To quantify these observations let us define the *variance reduction* of a smoother as follows.

$$\begin{aligned} \rho &\stackrel{\triangle}{=} \text{variance reduction} && (4.3.45)\\ &= \frac{p_0 - p_s}{p_0} \\ p_0 &= \text{average process variance} && (4.3.46)\\ p_s &= \text{average smoothing error variance} && (4.3.47) \end{aligned}$$

We also define the *performance degradation* resulting from using a lattice smoother as compared with using the standard smoother as follows.

$$\begin{aligned} \Delta_{\text{perf}} &\stackrel{\triangle}{=} \text{performance degradation} && (4.3.48)\\ &= \frac{\rho_{\text{standard}} - \rho_{\text{lattice}}}{\rho_{\text{standard}}} \\ \rho_{\text{standard}} &= \text{variance reduction of standard smoother} && (4.3.49)\\ \rho_{\text{lattice}} &= \text{variance reduction of lattice-model smoother} && (4.3.50) \end{aligned}$$

If we compare the variance reduction of the two smoothers, we get that the variance reduction for the standard smoother is 85.0 percent whereas the variance reduction for the 2-tap lattice model smoother is 81.9 percent. For the case of the 8-tap lattice model smoother the reduction is 83.8 percent. Figure 4.3.15 shows a plot of the smoothing errors for the various smoothers over the entire interval.

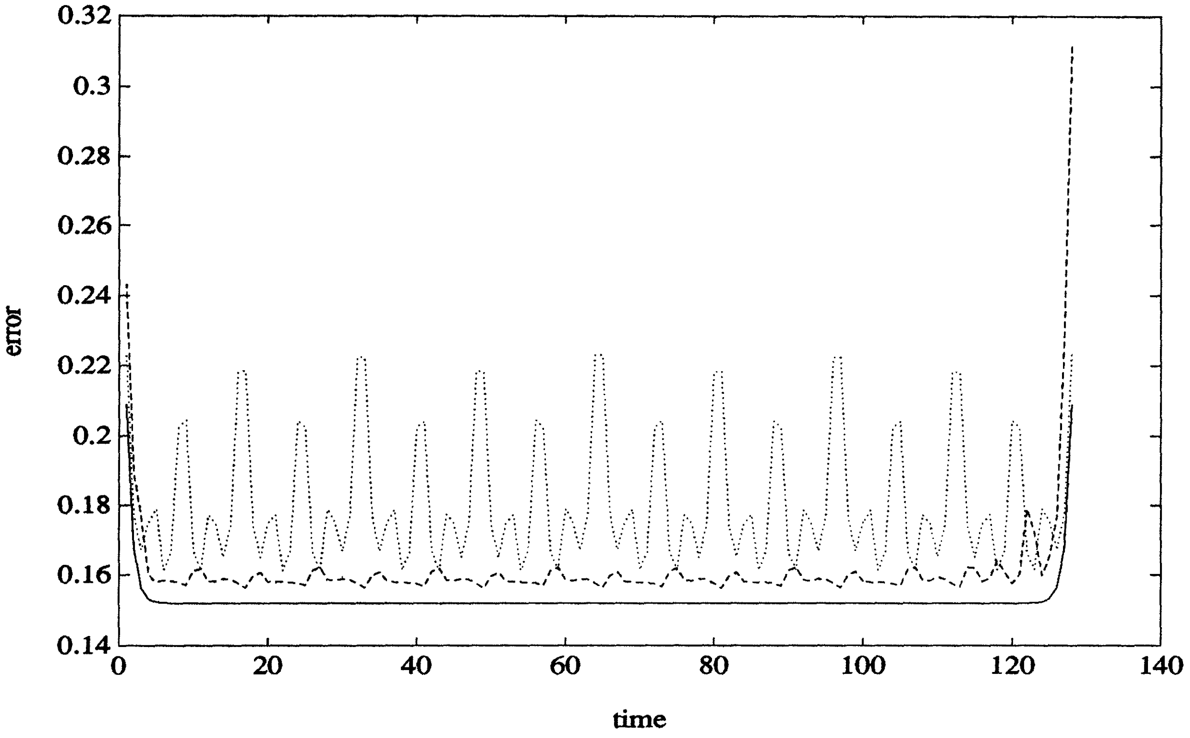

Figure 4.3.15: Standard Smoother Error (solid), 8-Tap Multiscale Smoother Error (dashed), 2-Tap Multiscale Smoother Error (dotted), (Data of SNR=1.4142)

| | 2-tap | 4-tap | 6-tap | 8-tap |
|---|---|---|---|---|
| $SNR = 2.8284$ | 1.07 % | .550 % | .402 % | .334 % |
| $SNR = 1.412$ | 3.27 % | 1.77 % | 1.24 % | 1.04 % |
| $SNR = .7071$ | 6.71 % | 4.13 % | 2.70 % | 2.33 % |
| $SNR = .5$ | 9.58 % | 6.14 % | 3.87 % | 3.27 % |

Table 4.3.1: Performance Degradation Comparison of Lattice-Model Smoothers - 2-tap, 4-tap, 6-tap and 8-tap

Note that while the 2-tap smoothing error looks highly irregular, no doubt due to the structure of the Haar transform, even at its points of highest amplitude the *relative* decrease in performance is still rather small. In particular, if we examine the point of highest amplitude for the 2-tap smoothing error in Figure 4.3.15, this corresponds to a performance degradation of only 8 percent. The strong similarity between the result of this smoother and the result of the standard smoother as shown in Figure 4.3.11 serves further to emphasize the fact that the degradation is negligible.

We now examine the performance degradation of the lattice-model when we vary the order of the QMF filter corresponding to the model. Table 4.3.1 shows the performance degradation of the lattice-model smoother relative to the standard smoother for filter tap orders 2, 4, 6, and 8 and for four different noise scenarios: 1) SNR = 2.8284 2) SNR = 1.412 3) SNR = .7071 4) SNR = .5. The variance reductions are computed using smoothing errors averaged over the entire interval.

While the degradation in performance lessens as the order of the filter increases, a great deal of the variance reduction occurs just using a 2-tap filter. For example for the case of $SNR = 1.412$ the standard smoother yields a variance reduction of 85 percent. It is arguable whether there is much to be gained in using an 8-tap filter when its relative decrease in performance degradation is only 2.23 percent over the 2-tap smoother; i.e. the variance reduction of the 8-tap smoother is 83.8 percent while the variance reduction of the 2-tap smoother is already 81.9 percent.

The performance degradation numbers for the lower SNR case (SNR = .7071) seem to suggest that the effect of raising the noise is to decrease the performance of the lattice-model smoothers. But one should keep in mind that this decrease is at most only marginal. Consider the case where the SNR = .5. In this case the data is extremely noisy, the noise power is *double* that of the case where SNR = .7071, and yet the performance degradation in using the 2-tap smoother compared with the standard smoother is 9.58 percent, up only 2.87 percent from the case of SNR = .7071. Furthermore, if one examines a plot of the smoothers, the performance of the two smoothers is as before quite comparable. Figure 4.3.16 is a plot of the sample path of the Gauss Markov process and the noisy data and Figure 4.3.17 compares the result of the standard smoother with that of the 2-tap smoother. We would expect that raising the level of the noise should somehow mask the effect of the model in smoothing the data, making our choice of model class somewhat irrelevant. The effect of noise in the data on the relative performance of the smoothers is one way of seeing how well our models do in approximating Gauss Markov processes. In the next section we use the Bhattacharyya distance as a way of gauging how well our models approximate these processes and in fact these results will confirm our intuition that noise masks the importance of the model structure one uses.

We emphasize that the average performance degradation is a *scalar* quantity, and at best gives only a rough measure of estimation performance. From this quantity it is difficult to get any idea of the qualitative features of the estimate. The plots of the sample path and its various smoothed estimates over the entire interval offer the reader much richer evidence to judge for himself what the relative differences are in the outputs of the various smoothers.

Also, as a final note recall that our fast algorithm in Chapter 2 involving the parallelization of the smoothing problem using the wavelet transform requires the same measurements to be made at all points at any particular scale. The case of missing data at a given scale for example is a situation in which this structure is violated. This is relevant to situations in which one might want to use coarse data to interpolate sparsely distributed fine data. This problem is treated in Section 4.4 using multiscale models based on homogeneous trees.

As we did for our lattice models we now demonstrate the potential utility of our tree models as the basis for designing smoothing algorithms and in particular we present numerical results that demonstrate the effectiveness of these models in

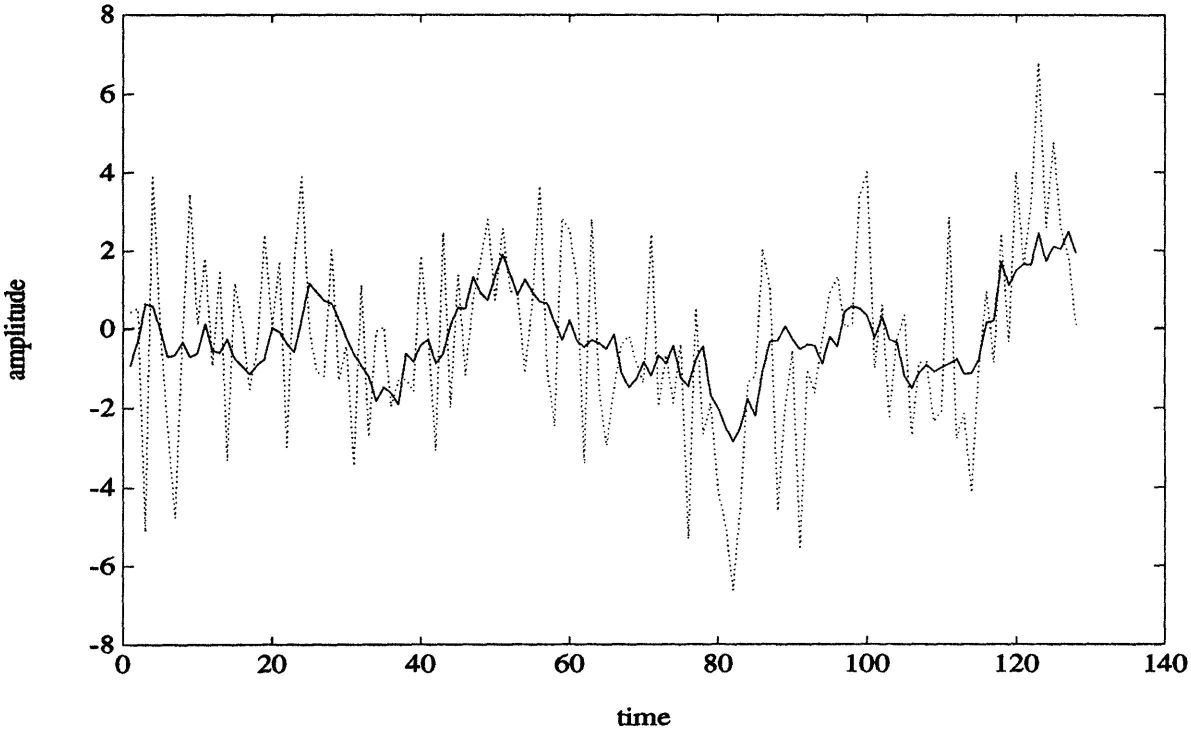

Figure 4.3.16: Sample Path of a Stationary Gauss-Markov Process (solid) and Its Noisy Version with SNR=.5 (dashed)

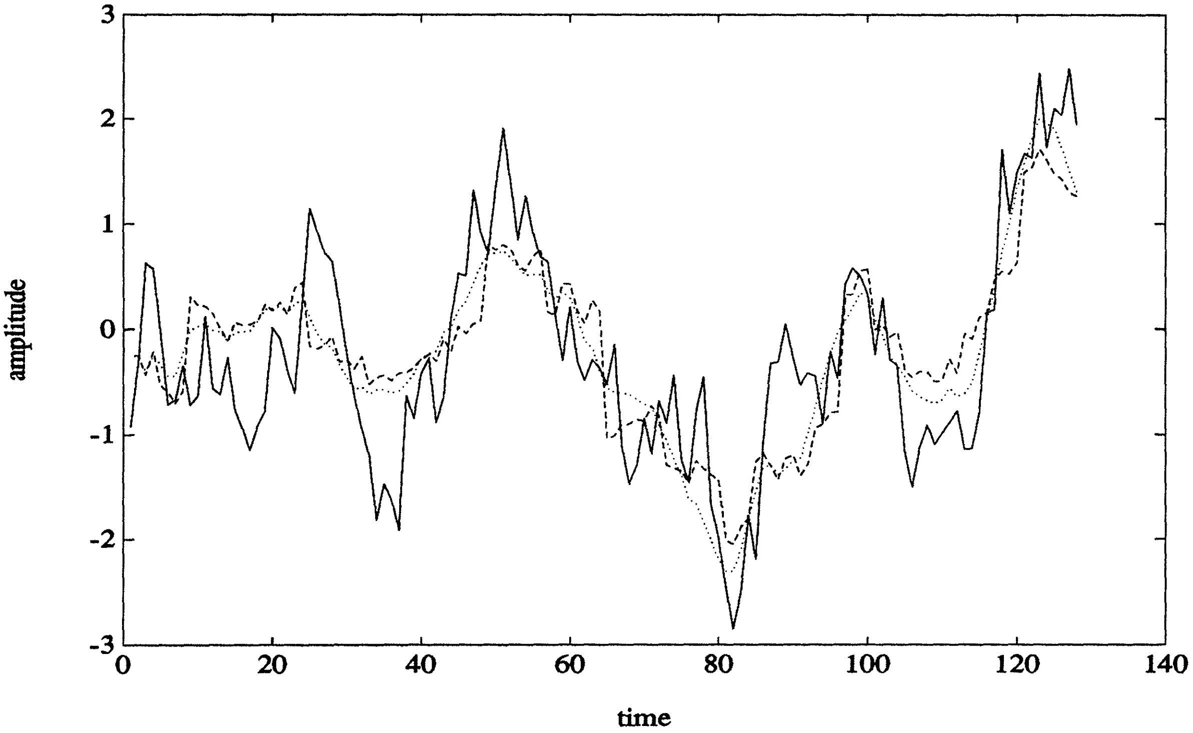

Figure 4.3.17: Sample Path of a Stationary Gauss-Markov Process (solid), Standard Smoother (dotted), 2-Tap Smoother (dashed) (Data of SNR=.5)

smoothing noisy Gauss-Markov processes. As in our examples using lattice models we use our tree model to smooth the noisy Gauss-Markov process in Figure 4.3.8 (SNR = 1.412). As described in the beginning of the section, we fit our tree parameters by choosing them to minimize the average suboptimal smoothing error variance. For the case where our measurements are of SNR = 1.412 the optimal tree parameters are as follows.

2-Parameter Tree Model:

$$a = .9905 \tag{4.3.51}$$

$$p = 6.2698 \tag{4.3.52}$$

3-Parameter Tree Model:

$$a = .9464 \tag{4.3.53}$$

$$p_0 = 7.7462 \tag{4.3.54}$$

$$\delta = .5059 \tag{4.3.55}$$

Figure 4.3.18 shows the result of our smoothed estimate using a 3-parameter model while Figure 4.3.19 compares this estimate with the result of using a standard smoother. Note again the similarity in performance between our smoother and the standard smoother. The performance degradation of the 3-parameter model smoother with respect to the standard smoother is 3.31 percent, which is comparable to the performance of the 2-tap lattice smoother. This is to be expected given the fact that both the 2-tap smoother and the tree smoother have the Haar representation as its natural basis. It is also interesting to note that by fitting the 2-parameter model in eq.(4.2.17) we get a performance degradation of 3.55 percent, only .24 percent worse than the 3-parameter model smoother. This seems to suggest that the 2-parameter model is already quite adequate in describing the 1st-order Gauss-Markov process, which is also described by two parameters. Table 4.3.1 is a comparison of the performances between the 2 and 3-parameter tree models for a variety of SNR's.

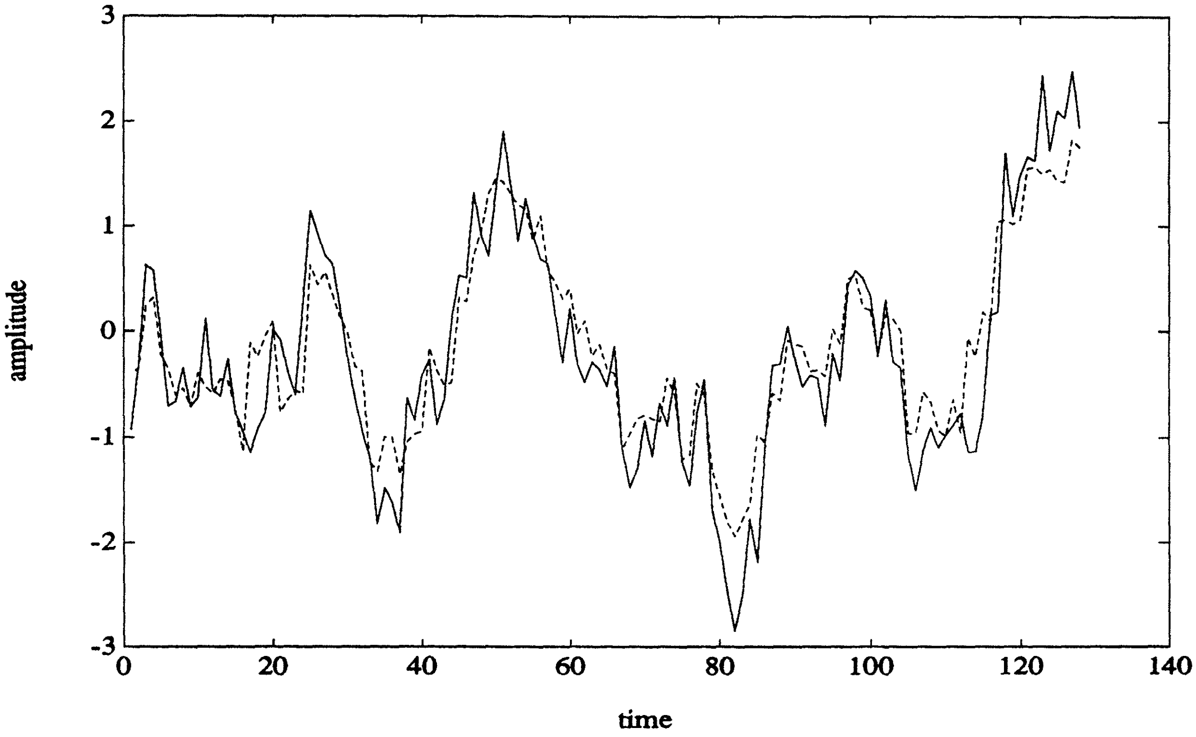

Figure 4.3.18: Stationary Gauss-Markov Process (solid) versus 3-parameter Tree-model Smoother (dashed) (Data of SNR=1.4142)

| | 2-parameter | 3-parameter |
|---|---|---|
| $SNR = 2.8284$ | 1.11 % | 1.08 % |
| $SNR = 1.412$ | 3.55 % | 3.31 % |
| $SNR = .7071$ | 7.59 % | 6.88 % |
| $SNR = .5$ | 10.52 % | 9.15 % |

Table 4.3.2: Performance Degradation Comparison of 2-parameter and 3-parameter Tree Smoothers

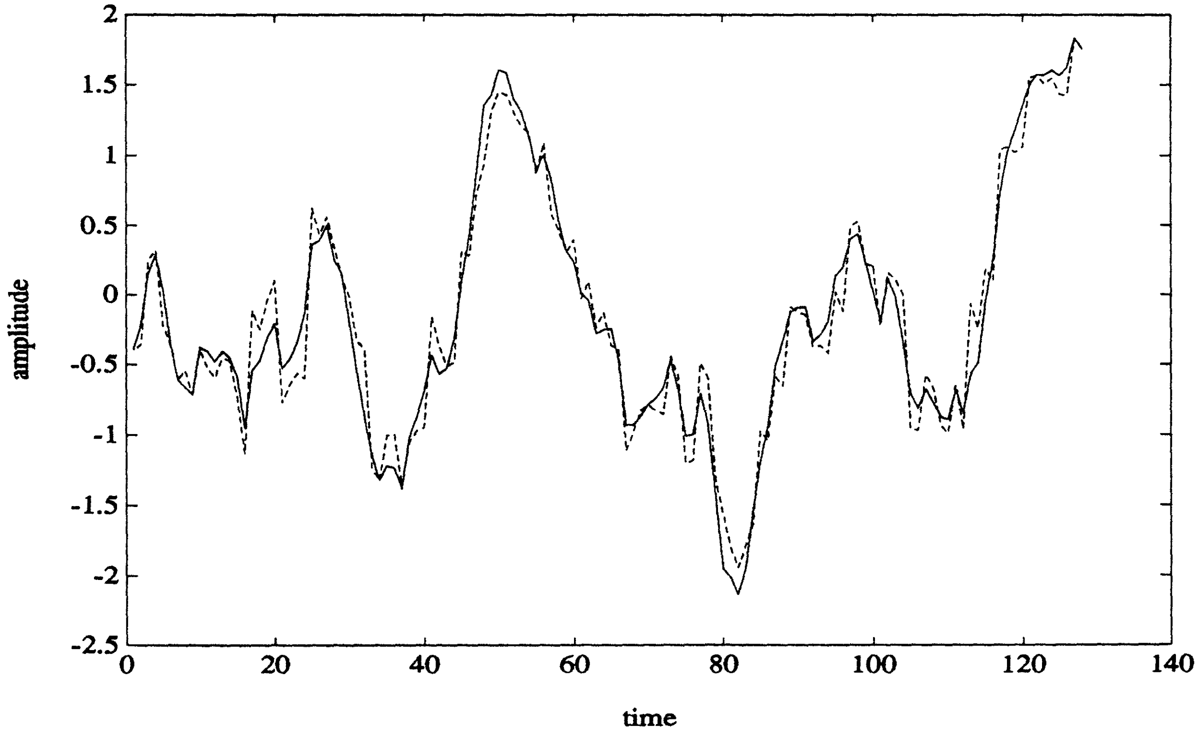

Figure 4.3.19: Standard Minimum Mean-Square Error Smoother (solid) versus 3-parameter Tree-model Smoother (dashed) (Data of SNR=1.4142)

| | 3-par. Tree | 2-par. Tree | 2-tap Lattice |
|---|---|---|---|
| $SNR = 1.4142$ | -21.94 % or 7.82 % | -23.83 % or 8.04 % | -21.83 % or 7.71 % |

Table 4.3.3: Mismatched $\alpha$ Comparison (both percentage decrease and increase in $\alpha$ given)

As a final point of comparison, we compare the performance of our non-standard smoothers with the performance of the standard smoother by seeing how much the parameter $\alpha$ of the Gauss-Markov process defined in eq.(4.2.5) must vary before the performance of the standard smoother is equal to the performance of our smoothers. Table 4.3.1 plots the value of the percentage decrease or increase over the true value of $\alpha$ such that the result of using the standard smoother with this mismatched value of $\alpha$ gives the same variance reduction as that resulting from a smoother using a 2-parameter tree model, a 3-parameter model, and a 2-tap lattice model. Note that the suboptimal error in using an incorrect value for $\alpha$ is not symmetric about the true value.

### 4.3.2 Bhattacharyya Distance

In this section we provide further evidence supporting the fact that multiscale processes are able to well-approximate stationary first order Gauss-Markov processes. In particular we use the *Bhattacharyya distance* as a measure of the distance between two stochastic processes. The Bhattacharyya distance is useful in providing bounds for the error probability in the binary hypothesis testing problem. As a way of comparing our multiscale processes with the Gauss-Markov process we consider the following problem. Suppose we are given a finite length observation, $y$, which we know to be equal to one of the following two stochastic processes in white noise.

$$y_x = x + n \tag{4.3.56}$$

$$E[xx^T] = \Sigma_x \tag{4.3.57}$$

$$y_z = z + n \tag{4.3.58}$$

$$E[zz^T] = \Sigma_z \tag{4.3.59}$$

The binary hypothesis testing problem consists of choosing one of the following hypotheses given our observation $y$: 1) $H_x : y = y_x$ 2) $H_z : y = y_z$. A decision is made based on the following threshold test.

$$\frac{Pr(y|H_x)}{Pr(y|H_z)} > \frac{P_z}{P_x} : \text{choose } H_x \tag{4.3.60}$$

$$\frac{Pr(y|H_x)}{Pr(y|H_z)} < \frac{P_z}{P_x} : \text{choose } H_z \tag{4.3.61}$$

where $P_z$ and $P_x$ are the a priori probabilities for the two hypotheses. If we consider the two hypotheses to be equally likely, the total probability of error in making a decision is as follows.

$$P_e = \frac{1}{2}P_1 + \frac{1}{2}P_2 \tag{4.3.62}$$

$$P_1 = Pr(\text{Choose} H_x | y = y_z) \tag{4.3.63}$$

$$P_2 = Pr(\text{Choose} H_z | y = y_x) \tag{4.3.64}$$

In the context of comparing stochastic processes if we let $x$ be the Gauss-Markov process and $z$ be one of our multiscale approximations the probability of error, $P_e$, gives an indication of how distinguishable two processes are in noise; i.e. a large value of $P_e$ indicates the two processes $x$ and $z$ are relatively indistinguishable from each other given some amount of noise in the observation. Note that the probability of error, $P_e$, is upper bounded by $\frac{1}{2}$ since in the worst case, i.e. if we randomly choose $H_x$(or $H_z$), $P_e = \frac{1}{2}$.

The Bhattacharyya distance between two probability densities is defined as follows.

$$B \triangleq -\ln \int (p_1(y)p_2(y))^{\frac{1}{2}} dy \tag{4.3.65}$$

As shown in [32] $B$ can be used to both upper and lower bound the probability of error $P_e$, where we take $p_1(y) = Pr(y|H_x)$ and $p_2(y) = Pr(y|H_z)$.

We now use the Bhattacharyya Distance to upper bound the error probability, $P_e$, where $y_x$ is a noisy-version of the Gauss-Markov process defined in eq.(4.2.5) and $y_z$ is a noisy-version of its lattice process approximation; i.e. $y_x$ is a Gauss-Markov process in white noise of some intensity and $y_z$ is our multiscale process, chosen to approximate the Gauss-Markov process, in white noise of the same intensity. Figure 4.3.20 plots the upper bounds for $P_e$ for several noise levels and for lattice models of various QMF filter orders. Note that adding noise to the process makes the error probabilities significantly larger, indicating that, as one would expect, noise tends to make the hypothesis testing problem more difficult. For sufficiently high noise levels the error probabilities approach .5, which is the error probability resulting from a decision based on the flip of a fair coin. This confirms our earlier intuition that the increase in noise in the data tends to mask the effects of the model, making the choice of model class less significant. The plot also indicates that for higher filter order our approximations get better as manifested by higher error probabilities. Note, however, that increasing the filter order only marginally increases performance. We also saw this in our earlier computations on performance degradation (Table 4.3.1).

Finally, as a point of comparison we note that the error probability for the 2-parameter tree model with SNR = 1.4142 is .2549 while for the case of the 3-parameter model with SNR = 1.4142, it is .2726. Under the same SNR the 2-tap lattice model yields an error probability of .2753.

### 4.3.3 Smoothing $1/f$ Processes

In this section we provide examples in which we use our models to smooth $1/f$ processes. We generate our sample path using the model of Wornell, i.e. we let the variances of our driving noise in eq.(2.2.2) vary as follows where we take our $h(n)$ to be the 4-tap Daubechies filter.

$$E[w(m)w(m)^T] = 2^{-\gamma m} \tag{4.3.66}$$

We take $\gamma$ to be equal to one, corresponding to an averaged spectrum of $1/|w|$, and we take our interval again to be 128 points. Figure 4.3.21 plots a sample path of the process and its noisy version of SNR = 1.4142. We would expect that our lattice smoother using a 4-tap model would do well in estimating this process since our model for the process exactly describes the model used to generate it. Figure 4.3.22 in fact confirms this. We also use a 3-parameter tree model to smooth the data, where the following are the optimal parameters for this case, i.e. the signal is a $1/f$ process with $\gamma = 1$ and the data is of SNR = 1.4142.

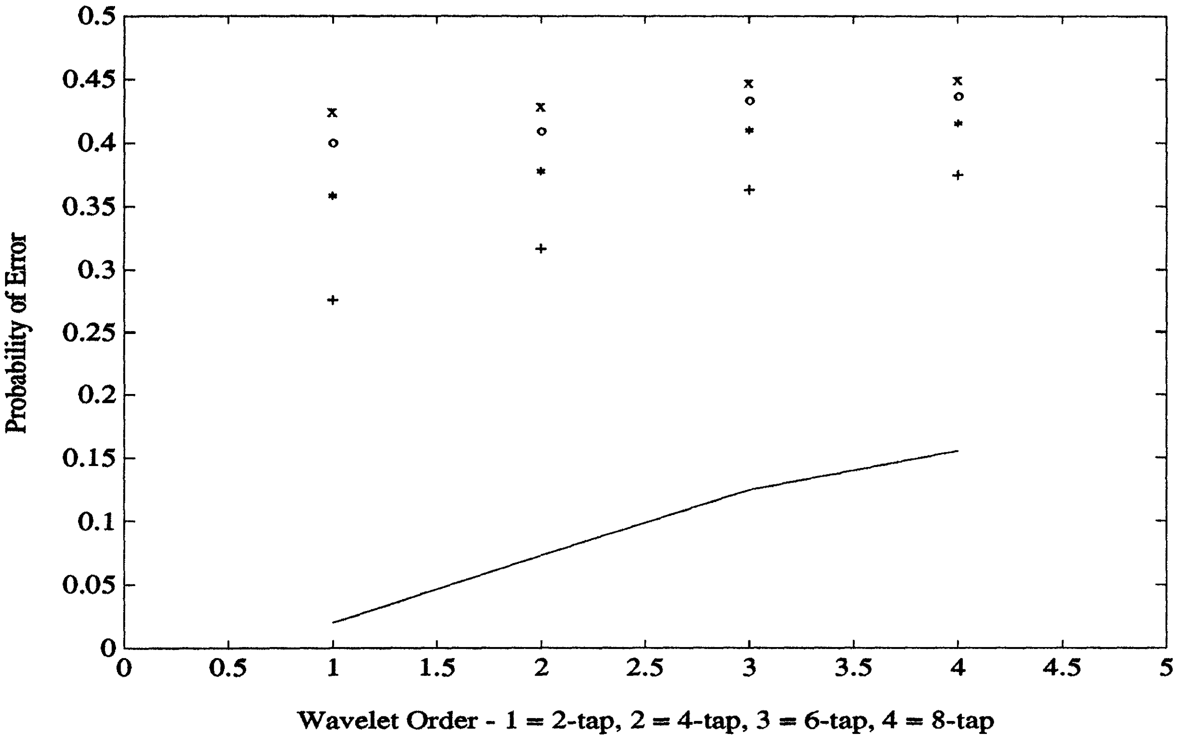

Figure 4.3.20: Error Probability - Gauss-Markov vs. Multiscale Models: R=0 (solid), R=.5 (+), R=1.0 (*), R=1.5 (o), R=2 (x)

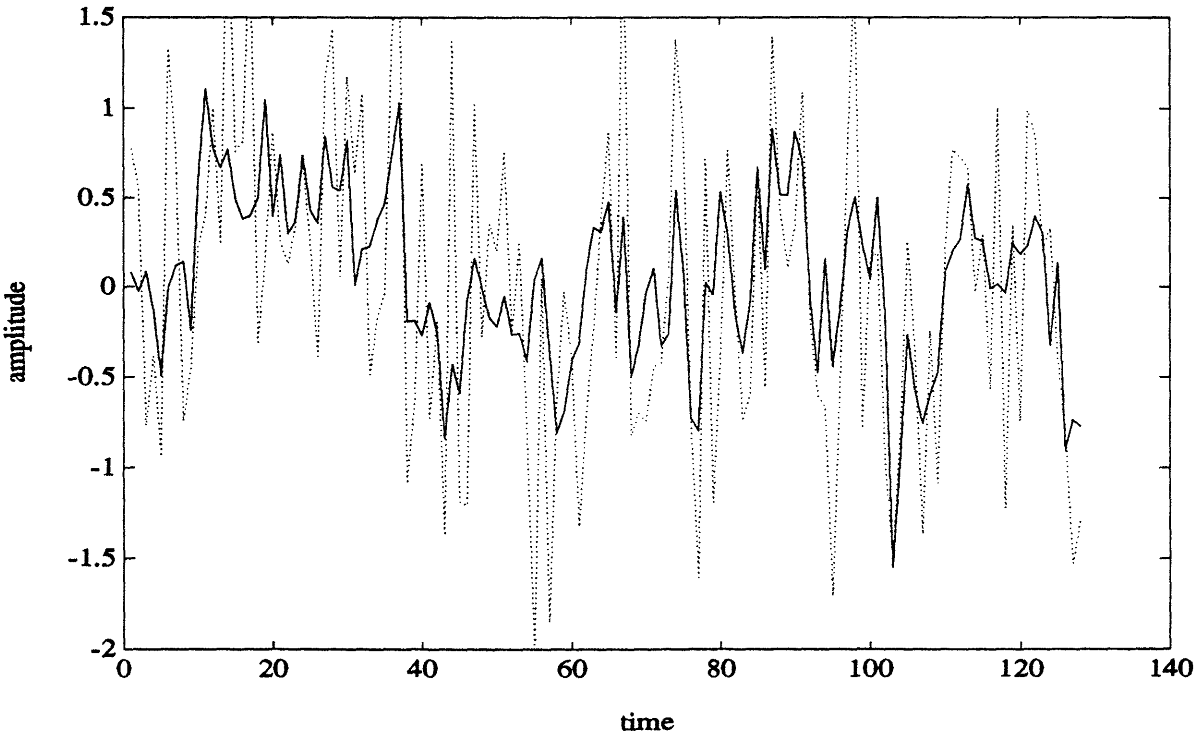

Figure 4.3.21: $1/f$ Process with $\gamma = 1$ (solid), Noisy Version of SNR = 1.4142 (dotted)

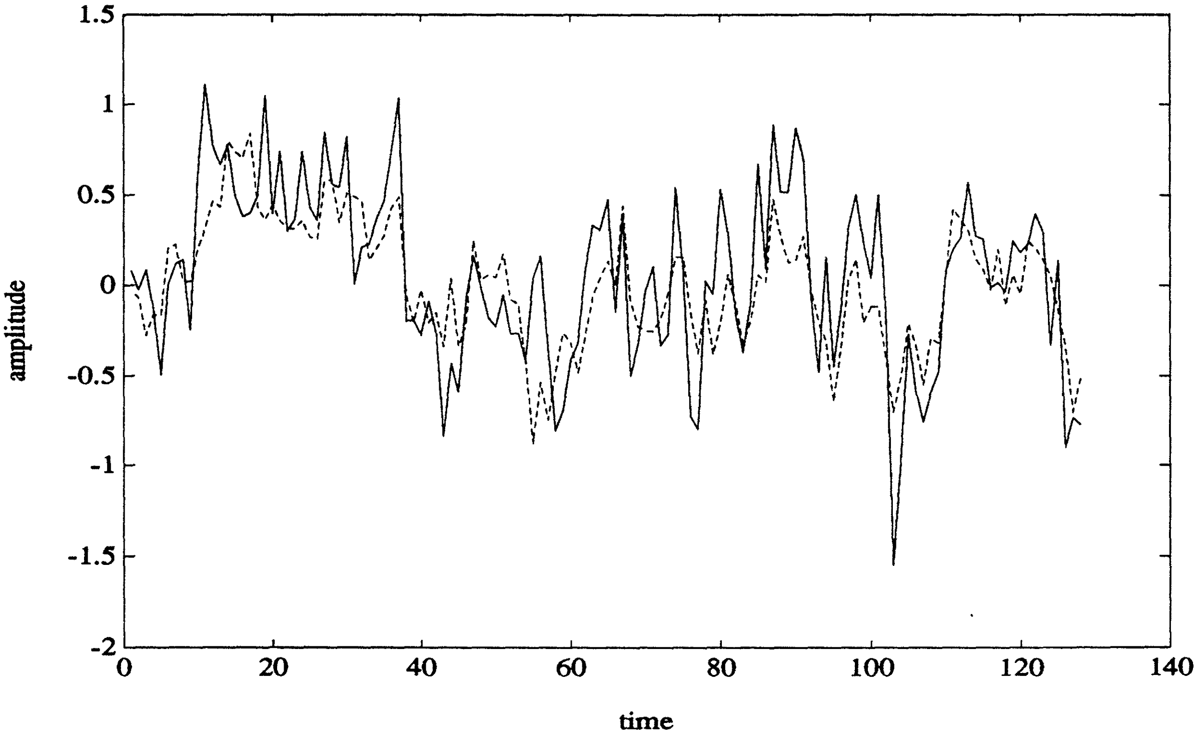

Figure 4.3.22: $1/f$ Process with $\gamma = 1$ (solid), Optimal Estimate Using 4-tap Lattice smoother (dashed)

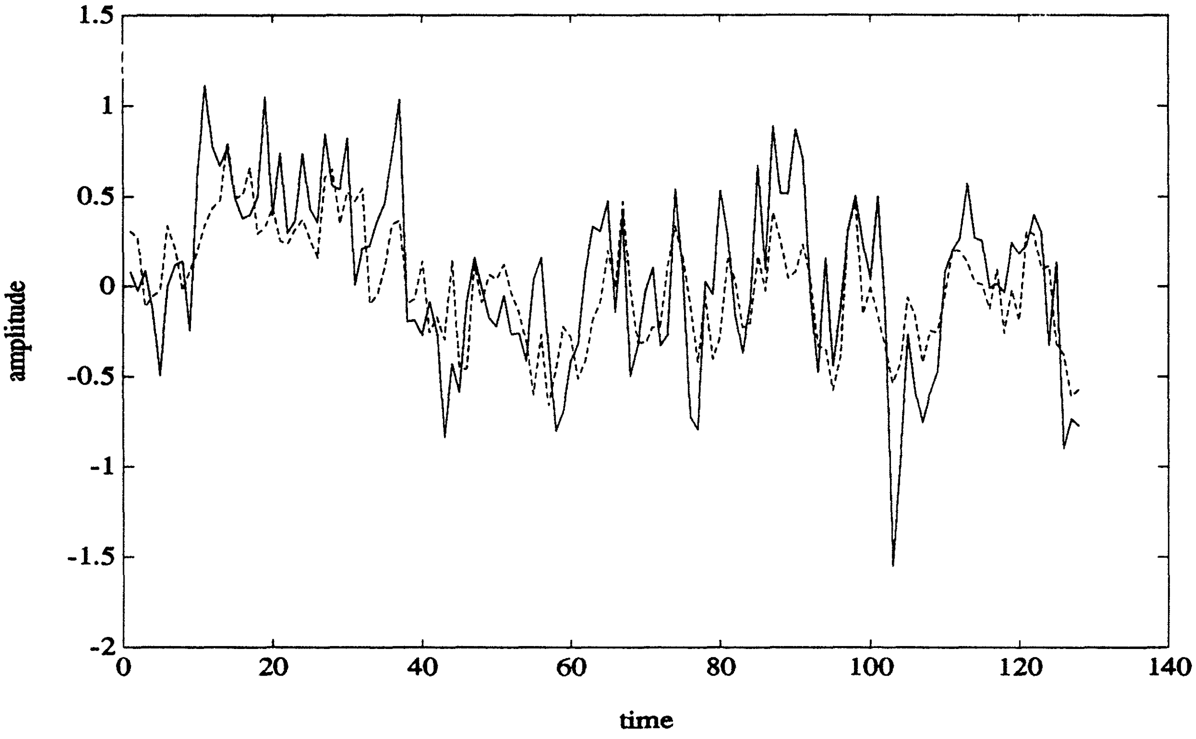

Figure 4.3.23: $1/f$ Process with $\gamma = 1$ (solid), Smoothed Using 3-parameter Tree Model (dashed)

3-Parameter Tree Model:

$$a = .6694 \tag{4.3.67}$$

$$p_0 = 1.0131 \tag{4.3.68}$$

$$\delta = .4843 \tag{4.3.69}$$

Figure 4.3.23 shows the result of using our 3-parameter tree model to smooth the data in Figure 4.3.21 while Figure 4.3.24 compares this result with the result obtained before using the 4-tap lattice model. Note the similarity in performance between the tree smoother and the 4-tap lattice smoother. The average performance degradation of the tree smoother compared with the 4-tap smoother is only 2.76%. Note, however, that for certain portions of the signal the tree smoother actually seems to do *better* in estimating the signal. This indicates that our tree models actually capture quite well $1/f$-like behavior. This brings about the question of how strong a

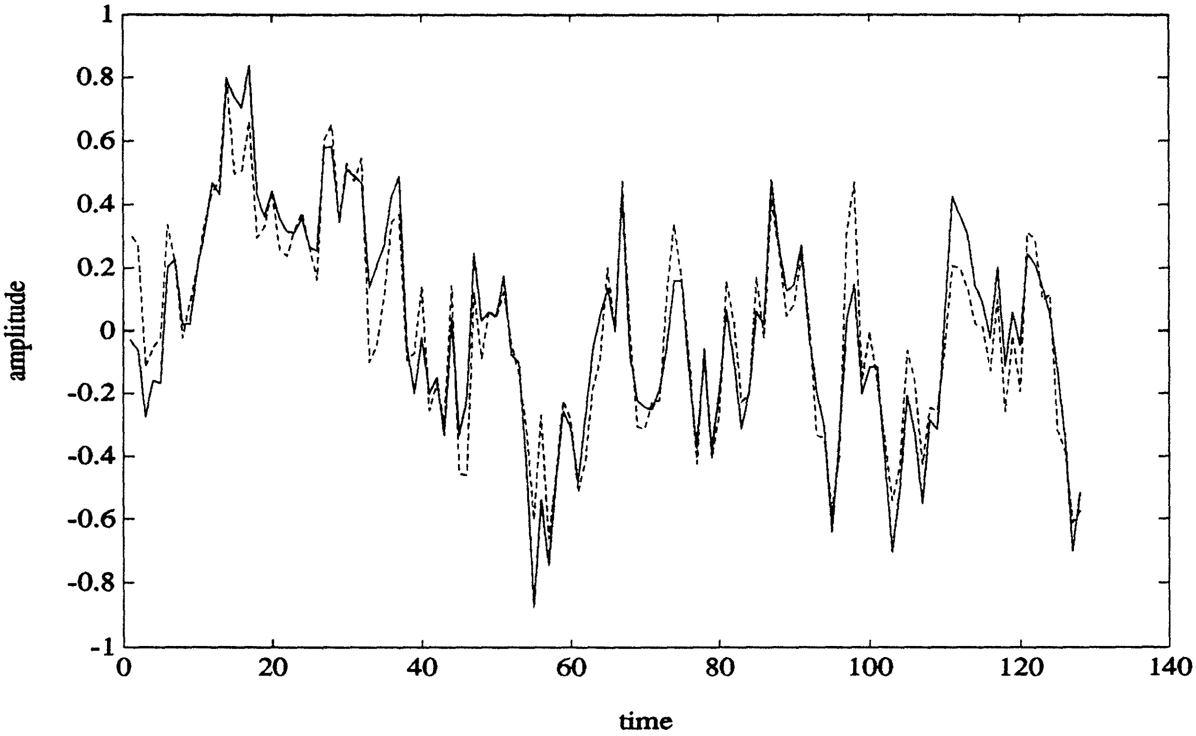

Figure 4.3.24: Smoothed Using 4-tap Lattice smoother (solid), Smoothed Using 3-parameter Tree Model (dashed)

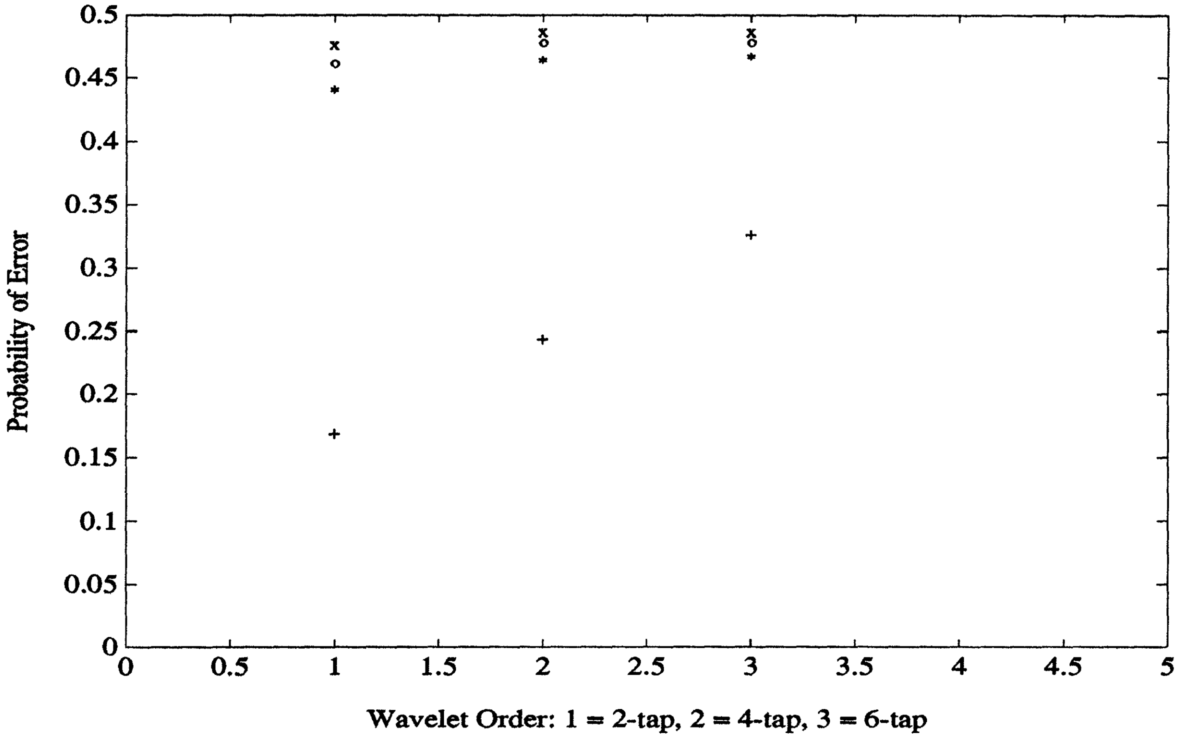

Figure 4.3.25: Error Probability - $1/f$ Process Generated Using 8-tap Model vs. Multiscale Models Generated Using 2-tap, 4-tap, and 6-tap: No Noise (+), SNR = 2 (*), SNR = 1.4142 (o), SNR = 1 (x)

factor the wavelet filter order in our lattice models is in capturing $1/f$-like features. This question can be partially answered by using the Bhattacharyya distance in the following way. We take $y_x$ to be our observed signal under the hypothesis that it is equal to a $1/f$ process with $\gamma = 1$ generated using an 8-tap filter in additive white noise. We let $y_z$ be our observed signal under the hypothesis that it is equal to our approximation to this process using either a 2-tap, 4-tap, or a 6-tap filter in additive noise of the same intensity. Figure 4.3.25 plots the upper bounds for $P_e$ for several noise levels. Note that again increasing the noise level added to the process makes the error probabilities significantly larger, making the model choice less significant. But in addition for reasonable noise scenarios the plot indicates that the 2-tap, 4-tap, and 6-tap filters perform comparably, all of them doing rather well in approximating the 8-tap model.

To further illustrate this last point we actually show the result of taking a noisy

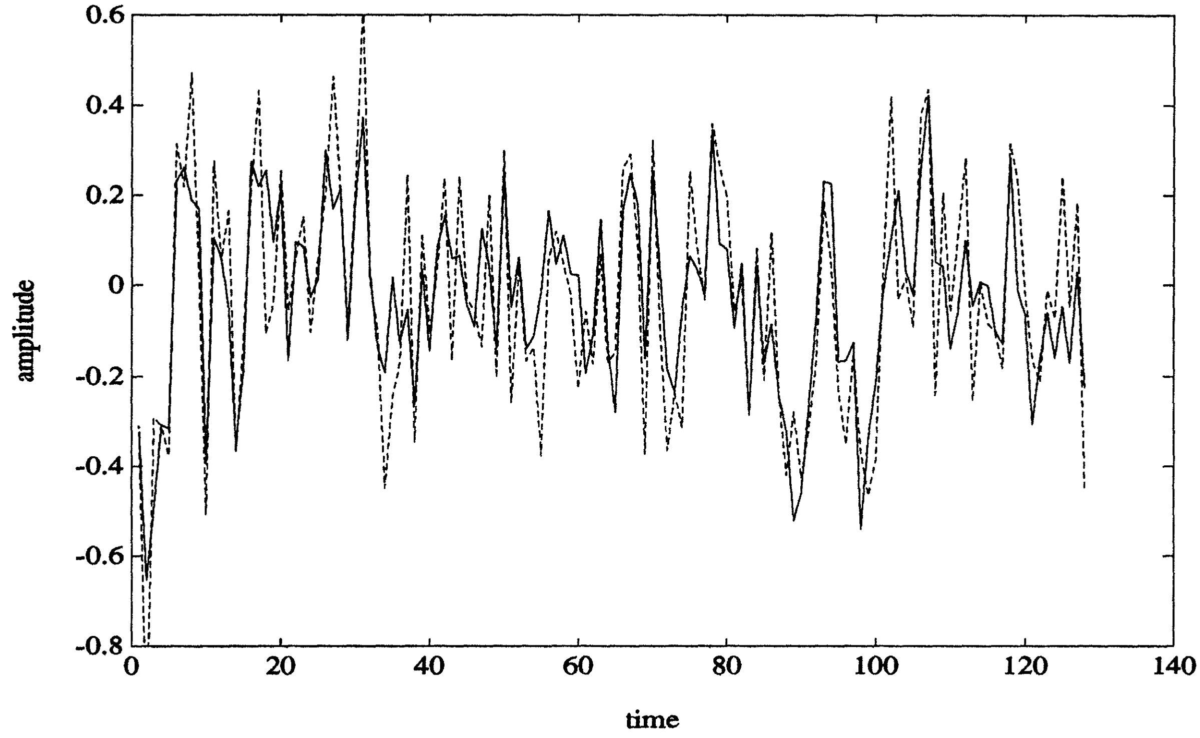

Figure 4.3.26: $1/f$ Process with $\gamma = 1$ Generated Using 8-tap $h(n)$ (solid), Noisy Version of SNR = 1.4142 (dashed)

$1/f$ process generated using an 8-tap filter and smoothing it using 6-tap, 4-tap and 2-tap lattice models. Figure 4.3.26 is a plot of a sample path of a $1/f$ process with $\gamma = 1$ which is generated using an 8-tap lattice model along with a noisy version of SNR = 1.4142. Figure 4.3.27 is a plot of the noiseless signal and its smoothed estimate using an 8-tap model, i.e. the optimal smoothed estimate using the correct model. In Figure's 4.3.28-4.3.30 we compare this optimal estimate with smoothed estimates using 6-tap, 4-tap and 2-tap lattice models, all of which correspond to models of the same $1/f$ process with $\gamma = 1$. Note in particular in Figure 4.3.30 the remarkable similarity between the estimate using an 8-tap model and the estimate using a 2-tap model. The performance degradation averaged over the interval, measured with respect to the performance of the 8-tap model, for the 2-tap, 4-tap, and 6-tap models are 3.59%, 2.15%, and 1.12%, respectively.

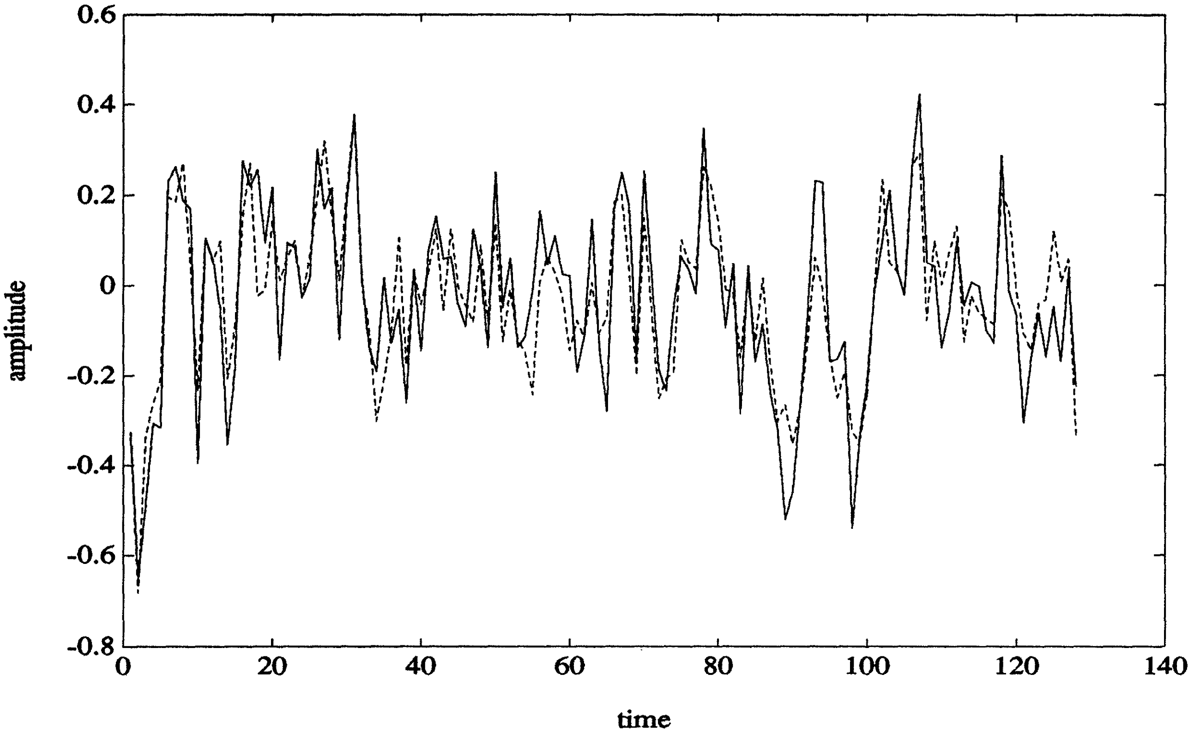

Figure 4.3.27: $1/f$ Process with $\gamma = 1$ Generated Using 8-tap $h(n)$ (solid), Estimate Using 8-tap Smoother (dashed)

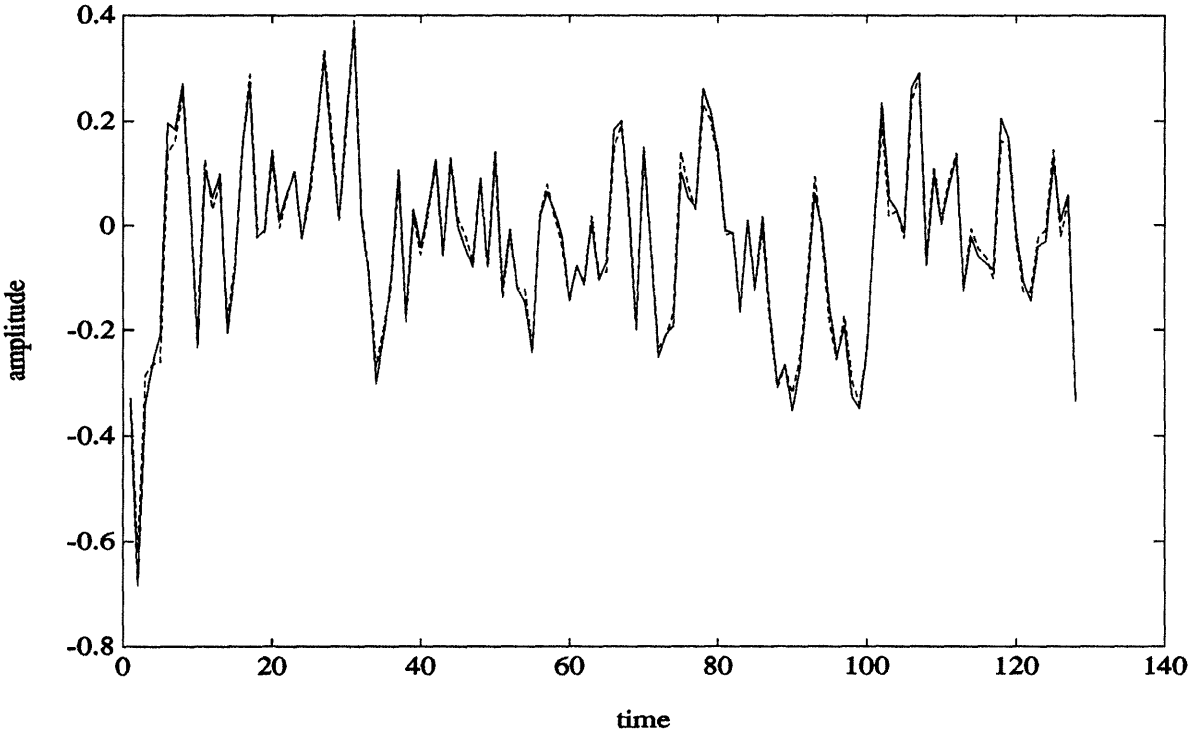

Figure 4.3.28: Estimate Using 8-tap Smoother (solid), Estimate Using 6-tap Smoother (dashed)

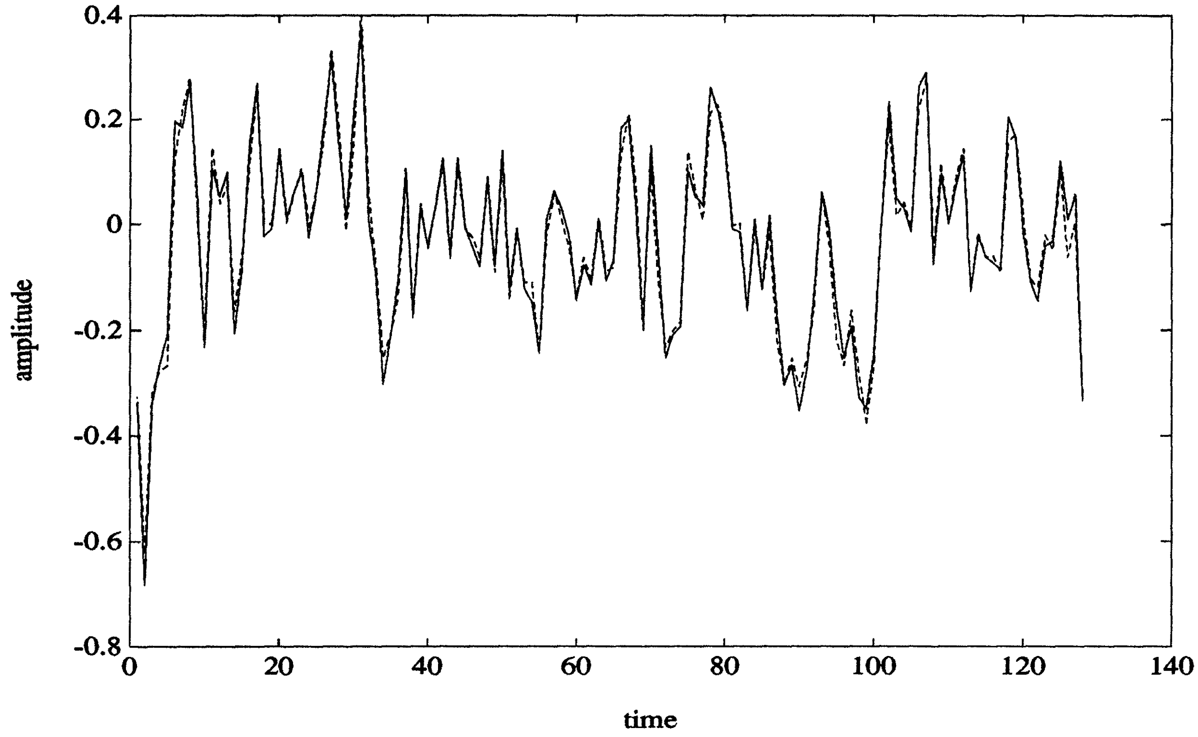

Figure 4.3.29: Estimate Using 8-tap Smoother (solid), Estimate Using 4-tap Smoother (dashed)

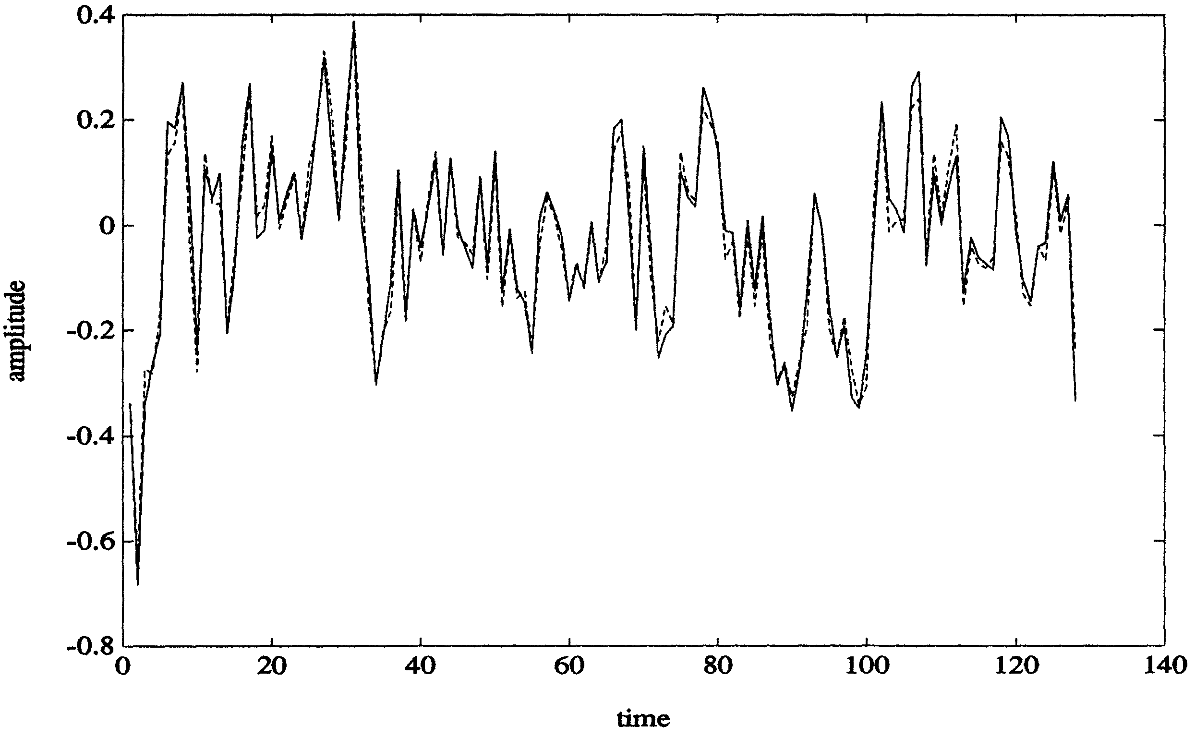

Figure 4.3.30: Estimate Using 8-tap Smoother (solid), Estimate Using 2-tap Smoother (dashed)

## 4.4 Sensor Fusion

In this section we provide examples that show how easily our framework handles the problem of fusing multiscale data to form optimal smoothed estimates. In our framework not only is there no added algorithmic complexity to the addition of multiscale measurements but it is also easy for us to *evaluate the performance* of our smoothers in using multiscale data. In the next section we present examples in which we use multiscale measurements where the coverage is full at each scale and the SNR's vary from scale to scale. In Section 4.4.2 we give examples in which the measurements at the finest scale are sparse, i.e. there are measurements which are absent at various points along the interval, but in which there are also measurements at a coarser scale which are in fact full in coverage. An example of this is the case in which sensors using both high frequency and low frequency energy are used to probe a medium. The high frequency sensor gives fine scale data in the vicinity of the boundaries of the medium but due to its short wavelength is unable to penetrate deeply into the medium. The low frequency sensor, however, is able to penetrate the medium due to its longer wavelength, giving data throughout, but at a coarser scale. Our example shows how our framework can be used to fuse the coarse scale data with the fine scale data, allowing the full-coverage coarse data to interpolate features unattainable from the sparse-coverage fine data. Note that in this case our methods using the wavelet transform to diagonalize the problem do not apply. We must use either an iterative algorithm for the case of general lattice models or our Rauch-Tung-Striebel algorithm on trees for the case of tree models.

### 4.4.1 Multiscale Measurement Performance

In this section we focus on the problem of using multiscale data to estimate a process in which the data at each of two scales is full in coverage but may vary in SNR from scale to scale. Consider the Gauss-Markov process used in our previous examples as defined in eq.(4.2.5). Again, the length of our interval is taken to be 128 points. For our coarse scale measurements we assume the following model where $M$ is the index of the finest scale containing $2^M$ points and $L$ is the index of the scale of $2^L$ points at which our coarse data resides. Referring to eq.(4.2.5) for the model of our process at the finest scale,

$$
\begin{aligned}
X_M &\triangleq \{x(t)|t = 0, 1...2^M - 1\} && (4.4.70)\\
X_L &\triangleq (\prod_{i=M-1}^{L} H_i)X_M && (4.4.71)\\
Y_{coarse} &= (\prod_{i=L}^{M-1} H_i^T)X_L + V && (4.4.72)\\
E[VV^T] &= rI && (4.4.73)
\end{aligned}
$$

That is, we take our coarse measurements to be the coarsened version of our fine scale signal, coarsened by using the wavelet coarsening operator to project the fine scale noiseless signal down to the desired scale, added with white noise. The choice of wavelet filter in the definition of the operator $H_i$ is governed by the way one chooses to model the relationship between the coarse and fine sensors. Note that in the case of the lattice models we are using, which correspond precisely to driving the wavelet synthesis equation with white noise, the projection $X_L$ is *exactly* equal to the lattice process at the $L$th scale. This is simply due to the structure of the wavelet transform and in particular the following analysis equation,

$$X_m = H_m X_{m+1} \qquad (4.4.74)$$

where $X_m$ is the state vector of our lattice process at the $m$th scale. As we discuss in the next section this is not true for the case of our lattice models as defined in eq.'s(2.2.62-2.2.64) of Chapter 2.

Consider the case where our fine scale measurements are of extremely poor quality. In particular we take the case where our data is of SNR = .3536 (the noise power is eight times the signal power) as illustrated in Figure 4.4.31. Figure 4.4.32 compares the result of using the standard smoother on this data with the result of using a 4-tap lattice model smoother on the same data. The performance of the two smoothers is comparable as we would expect from our results in the previous section.

Now we use the same data and consider fusing a higher quality coarse scale data set to form a smoothed estimate. In particular we take our coarse data to reside at

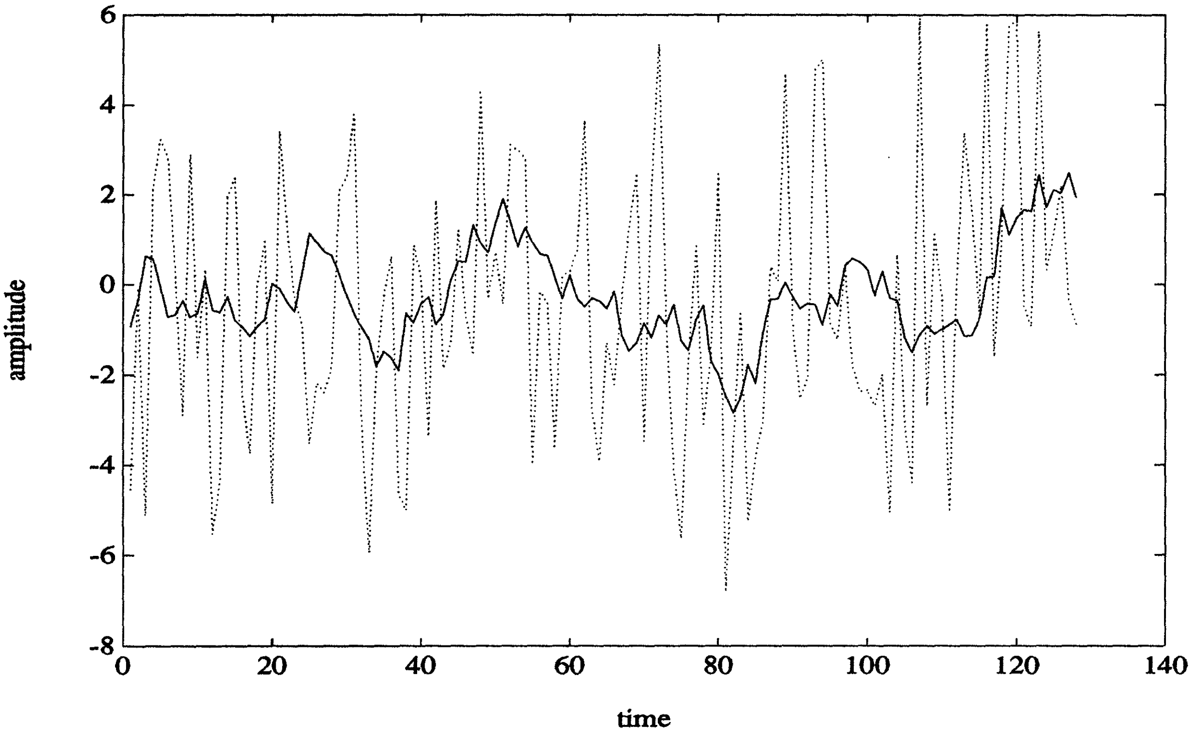

Figure 4.4.31: Sample Path of Stationary Gauss-Markov Process (solid), Poor Quality Data with SNR=.3536 (dotted)

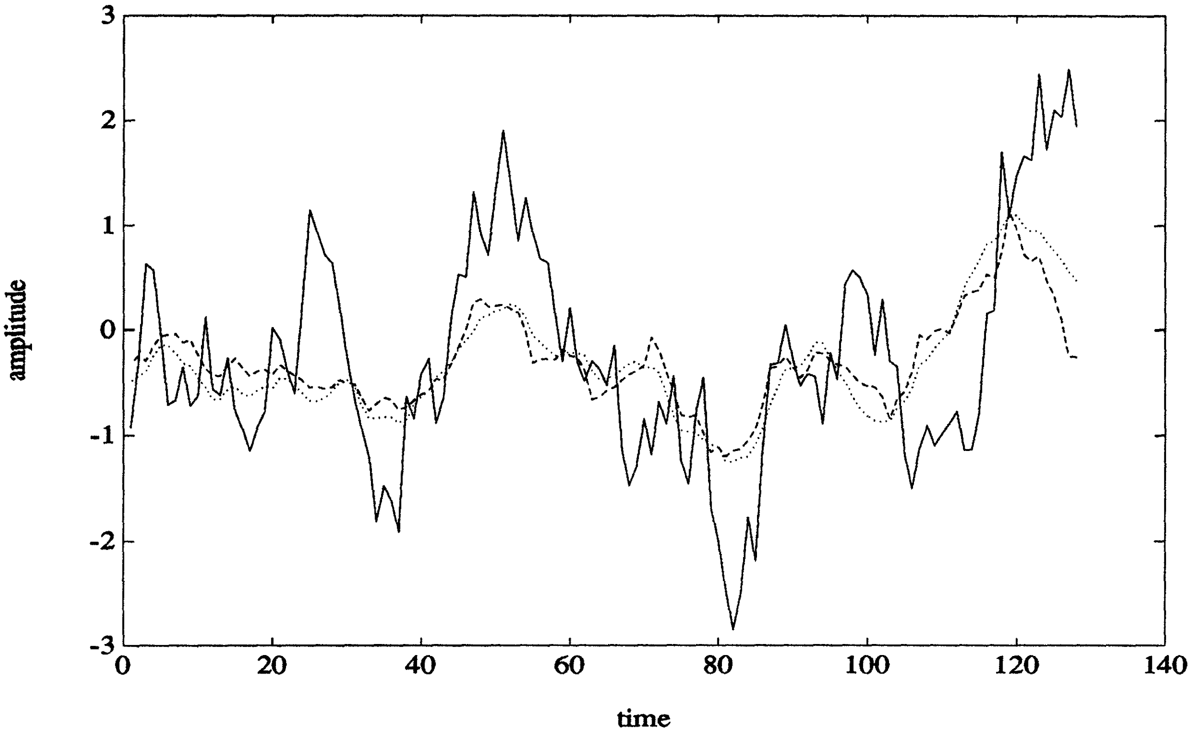

Figure 4.4.32: Sample Path of Stationary Gauss-Markov Process (solid), Result of Standard Smoother on Poor Data of SNR=.3536 (dotted), Result of 4-tap Lattice Smoother on Same Data (dashed)

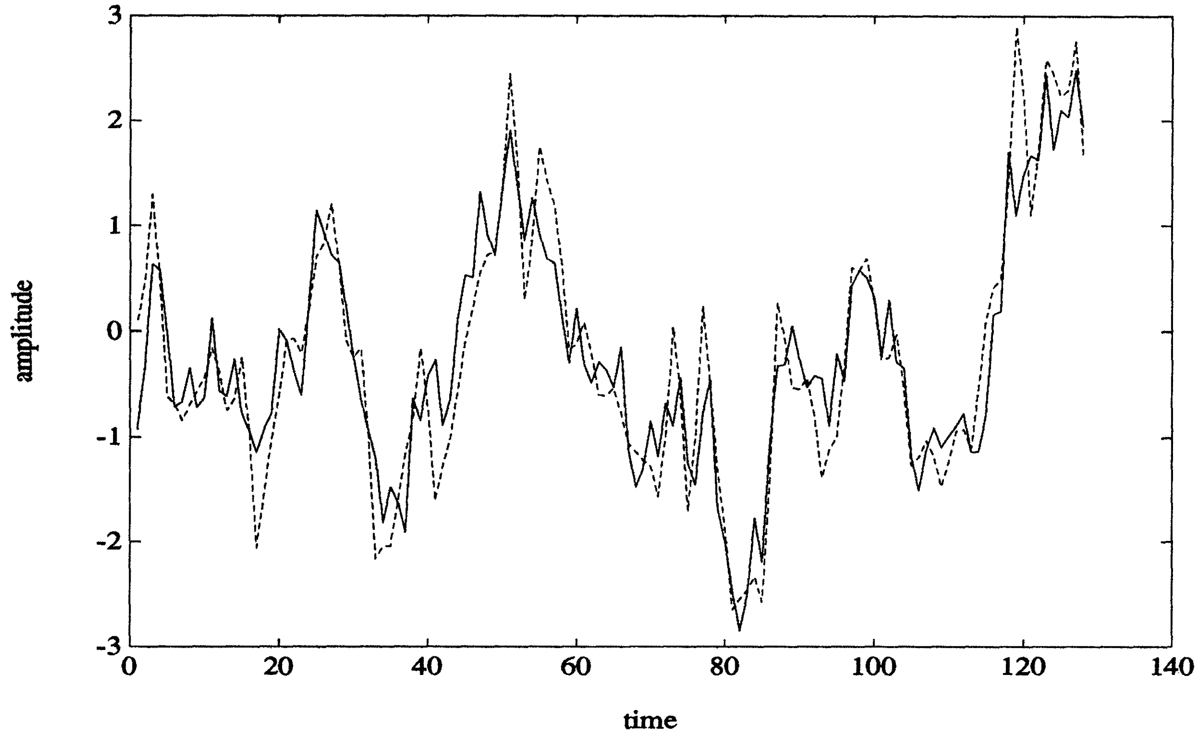

Figure 4.4.33: Sample Path of Stationary Gauss-Markov Process (solid), Coarse Data of SNR = 2 Using 4-tap Model (dashed)

the scale one level coarser than the original data (scale at which there are 64 points) and the coarsening operator, $H_i$, corresponds to a 4-tap filter. The SNR of this coarse data is equal to 2. Figure 4.4.33 is a plot of this coarse data. Figure 4.4.34 compares the result of using the standard smoother on the low quality fine scale data with the result of using our 4-tap lattice smoother to fuse this low quality data with high quality coarse data. Note that the coarse measurement aids dramatically in improving the quality of the estimate over the use of just fine-scale data alone. To quantify this recall that our smoother computes the smoothing error at each scale. We use these errors as *approximations* to the actual suboptimal errors (note that the computation of the actual error covariance from multiscale data is appreciably more complex than for the case of single scale measurements; the same is *not* true for our tree models, where the complexity of the two cases is essentially the *same*). The variance reduction in the case of fusing the two measurement sets is 97 percent versus only 36 percent for the case of using only the poor quality fine-scale data.

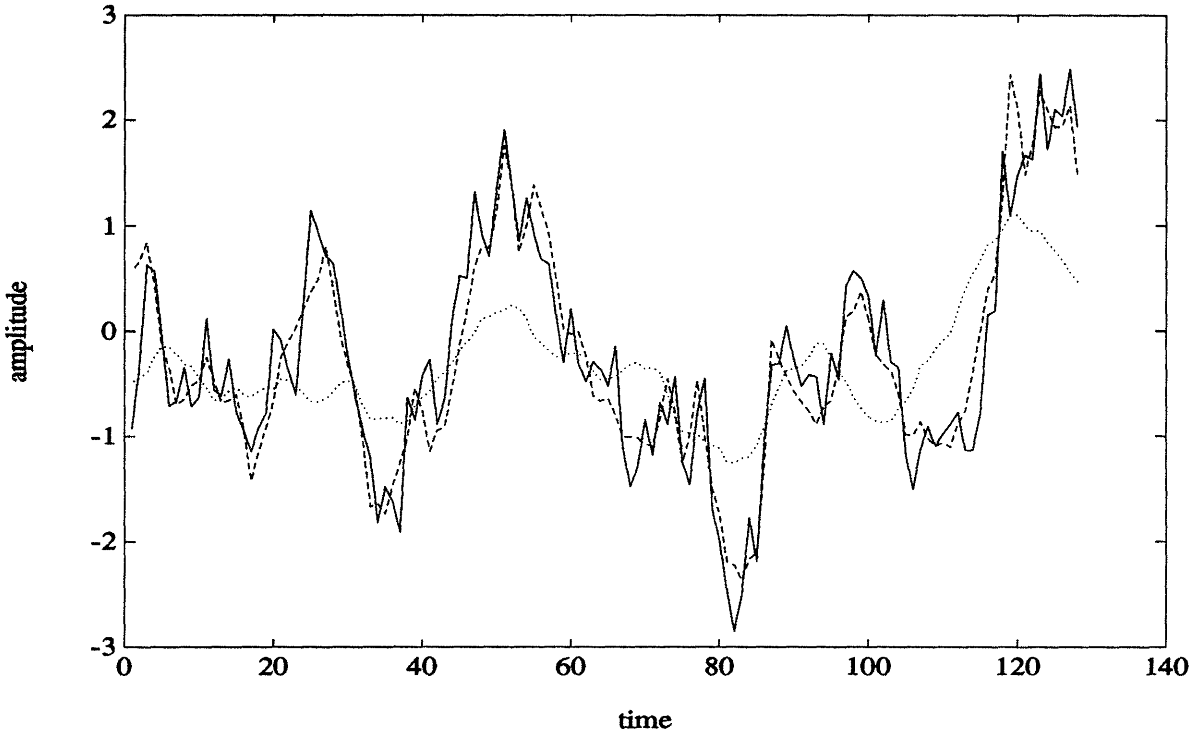

Figure 4.4.34: Sample Path of Stationary Gauss-Markov Process (solid), Result of Standard Smoother on Fine Data of SNR = .3536 (dotted), Result of 4-tap Lattice Smoother on Same Data Supplemented with Coarse Data of SNR = 2 (dashed)

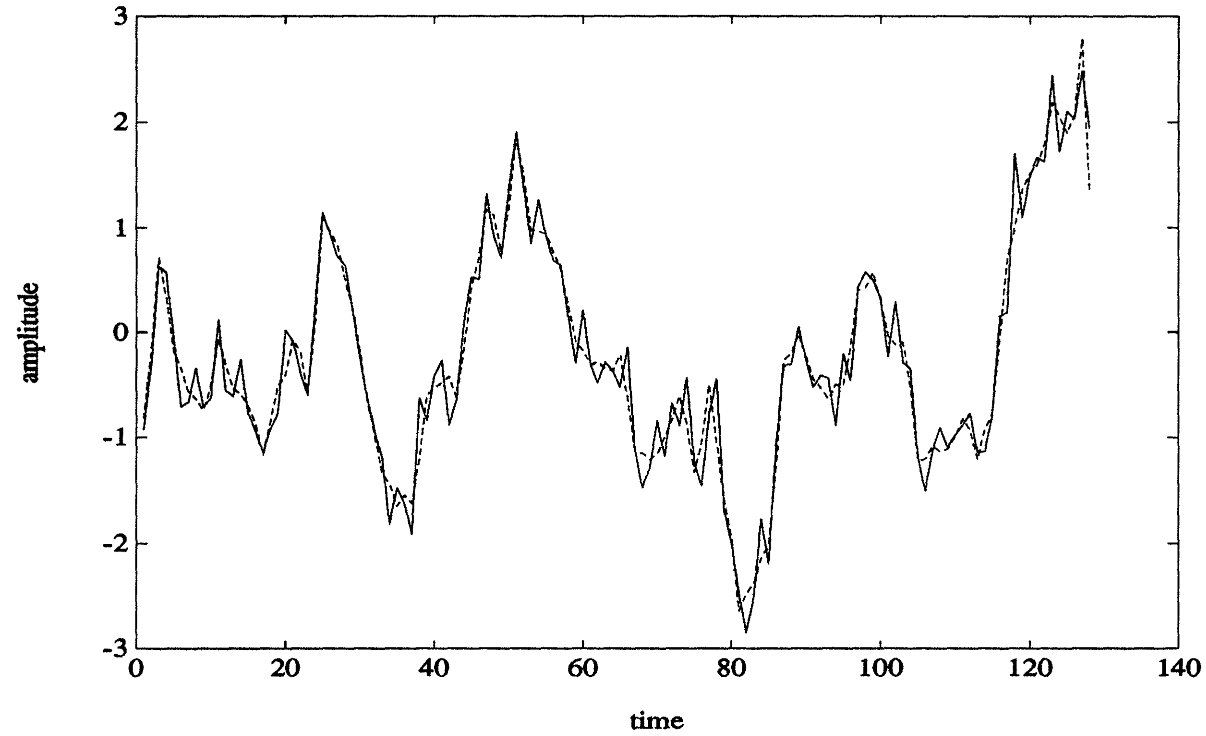

Figure 4.4.35: Sample Path of Stationary Gauss-Markov Process (solid), Results of 4-tap Lattice Smoother Using Fine Data of SNR = .3536 Supplemented with Coarse Data of SNR = 31.6: Coarse Data at 64 pt. Scale (dashed)

To explore even further the idea of fusing coarse measurements with poor quality fine measurements we compare the results of using coarse measurements of various degrees of coarseness in order to determine how the scale of the coarse data affects the resolution of the smoothed estimate. In particular, we take our fine scale data to be the same as that in Figure 4.4.34. However, we supplement this data with coarse measurements of extremely high quality (SNR = 31.6) and consider several cases: 1) the coarse data is at a scale at which there are 64 points 2) the coarse data is at a scale at which there are 32 points 3) the coarse data is at a scale at which there are 16 points. Figures 4.4.35-4.4.37 compare the original signal with its smoothed estimates using coarse data at the three different scales. Note how the coarseness of the estimates in this figure corresponds to the coarseness of the data used to produce them. All of these estimates are considerably better than the estimate using the fine data alone as illustrated in Figure 4.4.32.

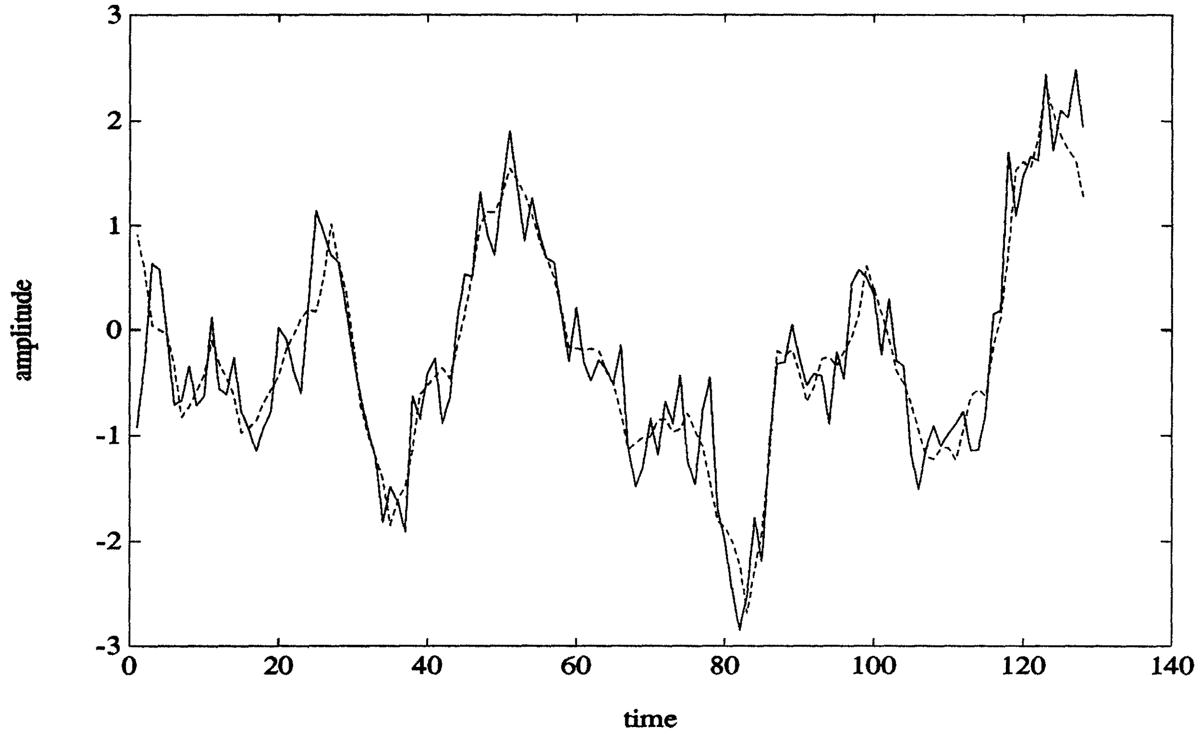

Figure 4.4.36: Sample Path of Stationary Gauss-Markov Process (solid), Results of 4-tap Lattice Smoother Using Fine Data of SNR = .3536 Supplemented with Coarse Data of SNR = 31.6: Coarse Data at 32 pt. Scale (dashed)

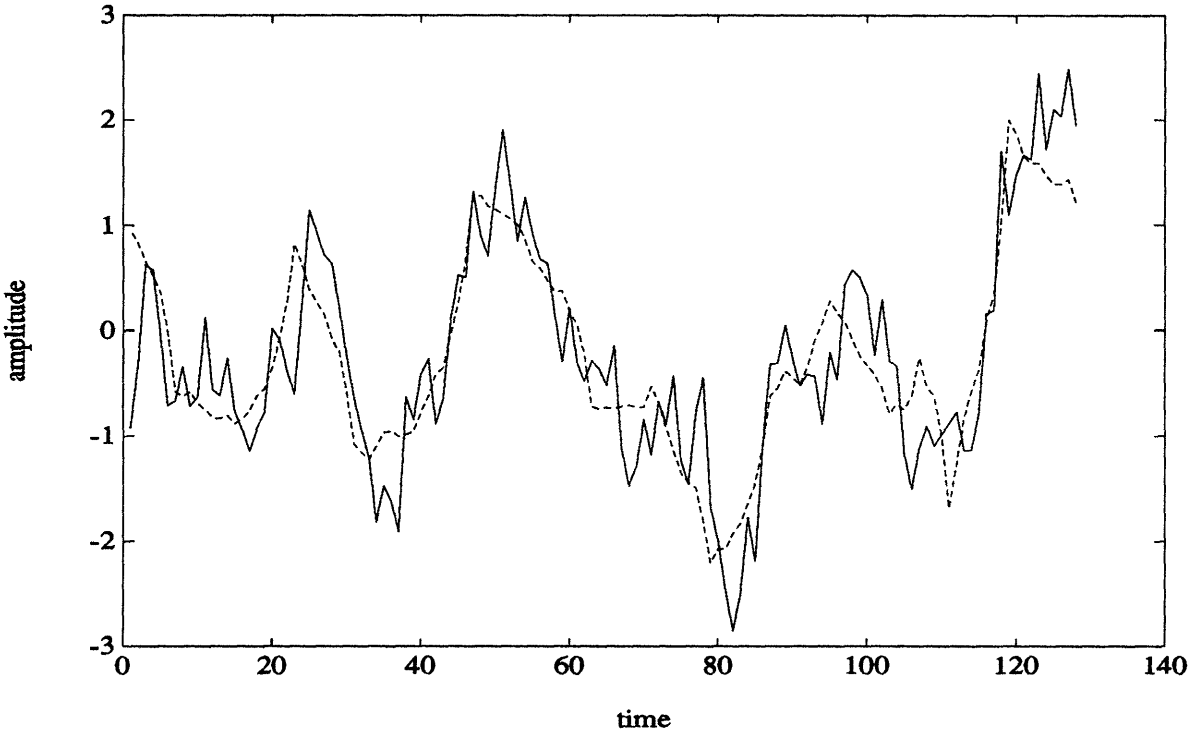

Figure 4.4.37: Sample Path of Stationary Gauss-Markov Process (solid), Results of 4-tap Lattice Smoother Using Fine Data of SNR = .3536 Supplemented with Coarse Data of SNR = 31.6: Coarse Data at 16 pt. Scale (dashed)

### 4.4.2 Limited Coverage Fine Scale, Full Coverage Coarse Scale

In this section we provide an example which corresponds to the problem of having measurements of limited coverage at the fine scale and measurements of full coverage at the coarse scale. In this case we would like to optimally fuse the two sets of measurements, where we would expect that the coarse measurements would provide information to interpolate the part of the signal not directly measured by the fine scale data. Note that since our data is not uniformly distributed, we cannot use transform based lattice smoothers to do the fusion. We instead use our Rauch-Tung-Striebel tree smoother, where the case of missing data simply amounts to setting the variable $C(t)$ equal to zero for all $t$ on the tree for which there is no measurement. In all of the examples which follow we use the 3-parameter tree smoother where the parameters are chosen to minimize the average suboptimal error resulting from using a full set of fine scale measurements of SNR $= 1.4142$. For the case of the Gauss-Markov process the parameters are as in eq.'s(4.3.53-4.3.55). For the case of the $1/f$ process which we consider later the parameters are as in eq.'s(4.3.67-4.3.69).

Consider the same Gauss-Markov process as before. We take the set of measurements of SNR $= 1.4142$ as illustrated Figure 4.3.8, and consider zeroing out the half of the measurements contained in the middle of the interval, i.e. the measurements at points 33 to 96. Figure 4.4.38 illustrates this measurement scenario. We now apply our 3-parameter tree smoother to this sparse data set the result of which is illustrated in Figure 4.4.39. As we would expect, the estimate in the middle of the interval where there is no data is very poor.

Now we consider using coarse scale data to try to improve the quality of our estimate based on sparse fine scale data. The coarse data in Figure 4.4.40 is of SNR $= 1.4142$ (the same as the fine scale data) and it is taken to be one level coarser than the fine data using the model in eq.(4.4.72) where the matrix $H_i$ is formed using the Haar 2-tap filter (i.e. this data represents pairwise averages of the original signal in additive white noise). Note, however, that unlike in the case of the lattice models we've used in this chapter which are *precisely* the synthesis equations driven by white

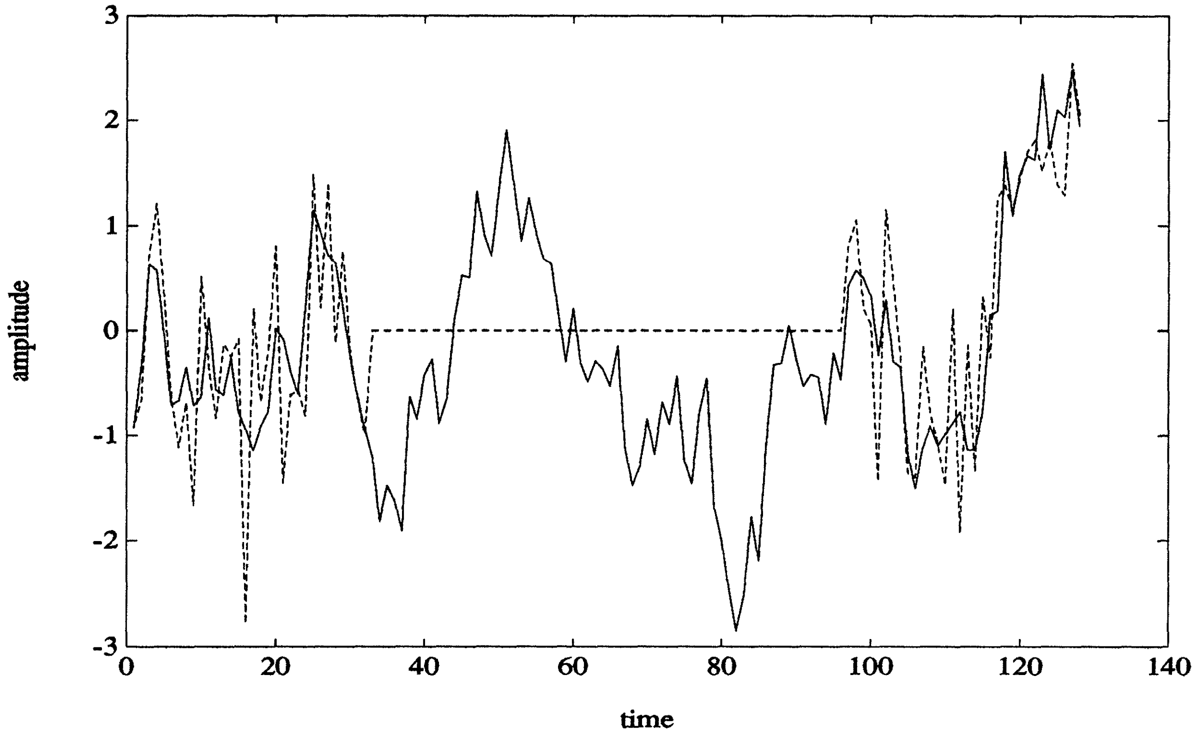

Figure 4.4.38: Sample Path of Stationary Gauss-Markov Process (solid), Noisy Sparse Data with SNR=1.4142 (dashed)

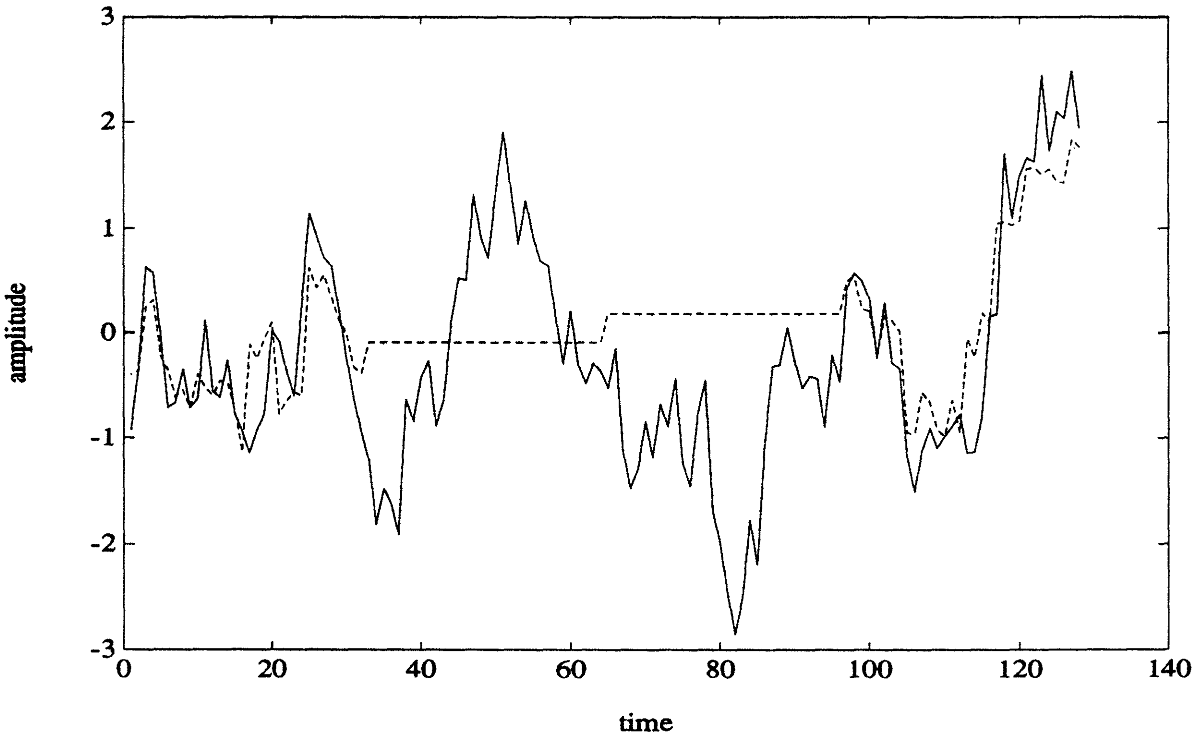

Figure 4.4.39: Sample Path of Stationary Gauss-Markov Process (solid), Result of Using Tree Smoother on Sparse Data of SNR=1.4142 (dashed)

noise as in eq.'s(2.2.1-2.2.3), for the case of our tree models we are essentially using a scalar version of the model in eq.'s(2.2.62-2.2.64) where the noises are *not* constrained by the differencing operator $G_m$. In this case our measurement equation, eq.(4.4.72), does not yield a noisy version of the tree process at a particular scale but a scaled version added with independent process noises. We show this more explicitly and for the case of a coarse scale measurement at an arbitrary scale, $L$. Note that for our 3-parameter tree model we have that

$$X(m) = (\sqrt{2}a)^{m-L}(\prod_{i=L}^{m-1} H_i^T)X(L) + \sum_{k=L+1}^{m-1} (\sqrt{2}a)^{m-k}2^{-\frac{\delta k}{2}}(\prod_{i=k}^{m-1} H_i^T)W(k) + W(m) \tag{4.4.75}$$

where $X(i)$ is the vector of points of our tree process $x(t)$ for $m(t) = i$, $W(i)$ is the vector of points of tree process noises $w(t)$ for $m(t) = i$, and $L$ is the level at which we consider our coarse data. Note that the $W(i)$'s are mutually uncorrelated and have diagonal covariance matrices, i.e.

$$E[W(i)W(j)^T] = q_j I \delta_{i-j} \tag{4.4.76}$$

Recall that our measurement at the $L$th scale is taken to be

$$X_L \triangleq (\prod_{i=M-1}^{L} H_i)X(M) \tag{4.4.77}$$

$$Y_L = (\prod_{i=L}^{M-1} H_i^T)X_L + V \tag{4.4.78}$$

$$E[VV^T] = rI \tag{4.4.79}$$

where $M$ is the finest scale. From eq.'s(4.4.75,4.4.76), the orthonormal properties of $H_i$, and our coarse scale measurement eq.(4.4.78), we have the following expression for our measurement, $Y$, at scale $L$.

$$Y_L = (\sqrt{2}a)^{m-L}(\prod_{i=L}^{M-1} H_i^T)X(L) + \tilde{W}(L) + V \tag{4.4.80}$$

$$E[\tilde{W}(L)\tilde{W}^T(L)] = (\sum_{k=L+1}^{m-1} (\sqrt{2}a)^{2(m-k)}2^{-\delta k}q_k + q_m)I \tag{4.4.81}$$

$$E[VV^T] = rI \tag{4.4.82}$$

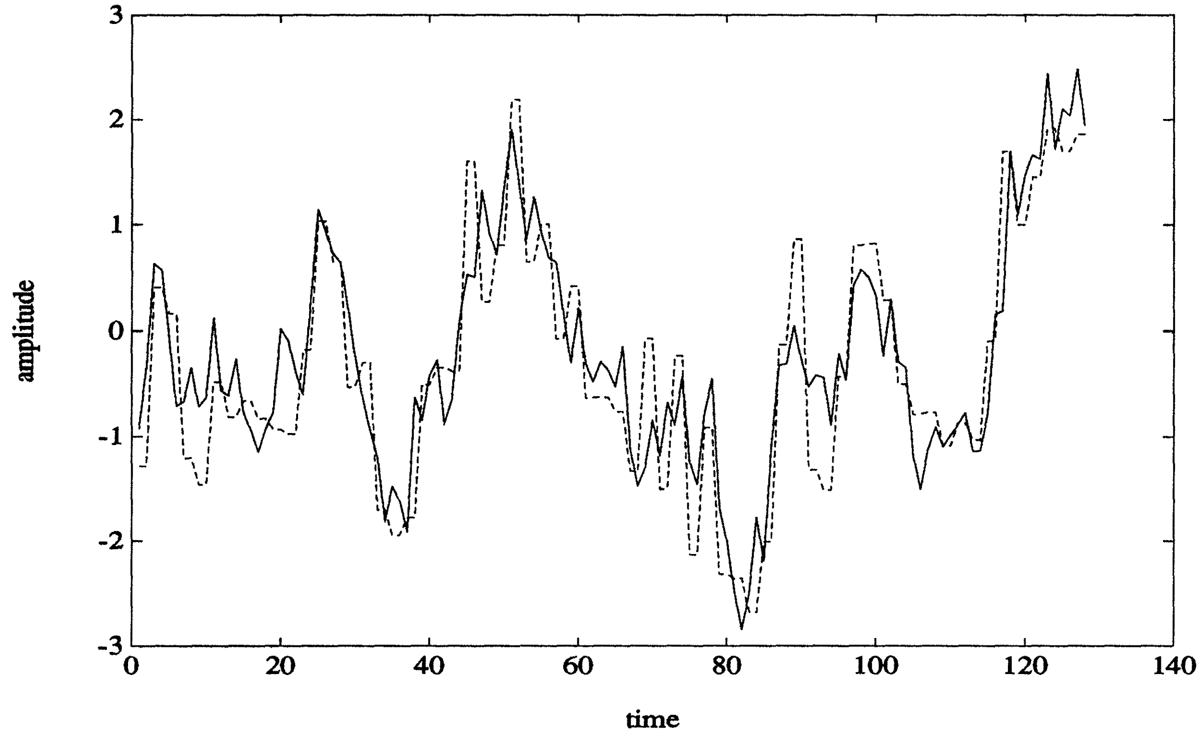

Figure 4.4.40: Sample Path of Stationary Gauss-Markov Process (solid), Coarse Data of SNR=1.4142 Modeled as Pairwise Averages of the Original Signal in Additive White Noise (dashed)

Thus, our measurement at the $L$th scale does indeed correspond to a scaled version of our tree process at the $L$th scale plus additive white noise, where the noise variance is equal to

$$\sum_{k=L+1}^{m-1} (\sqrt{2}a)^{2(m-k)} q_k + q_m + r \tag{4.4.83}$$

Note, however, that part of this noise, namely $\tilde{W}(L)$, is correlated with measurements at scales finer than $L$. Though we could use state augmentation to handle this correlation, for the sake of simplicity we have chosen in our examples to ignore this correlation.

We now show examples using our 3-parameter tree model to fuse the sparse fine scale data with the full coverage coarse data at a 64 point scale. Figure 4.4.41 illustrates the result of using our tree smoother to fuse the coarse and fine data. Note the high degree of interpolation performed by using the coarse scale data set. It is worth

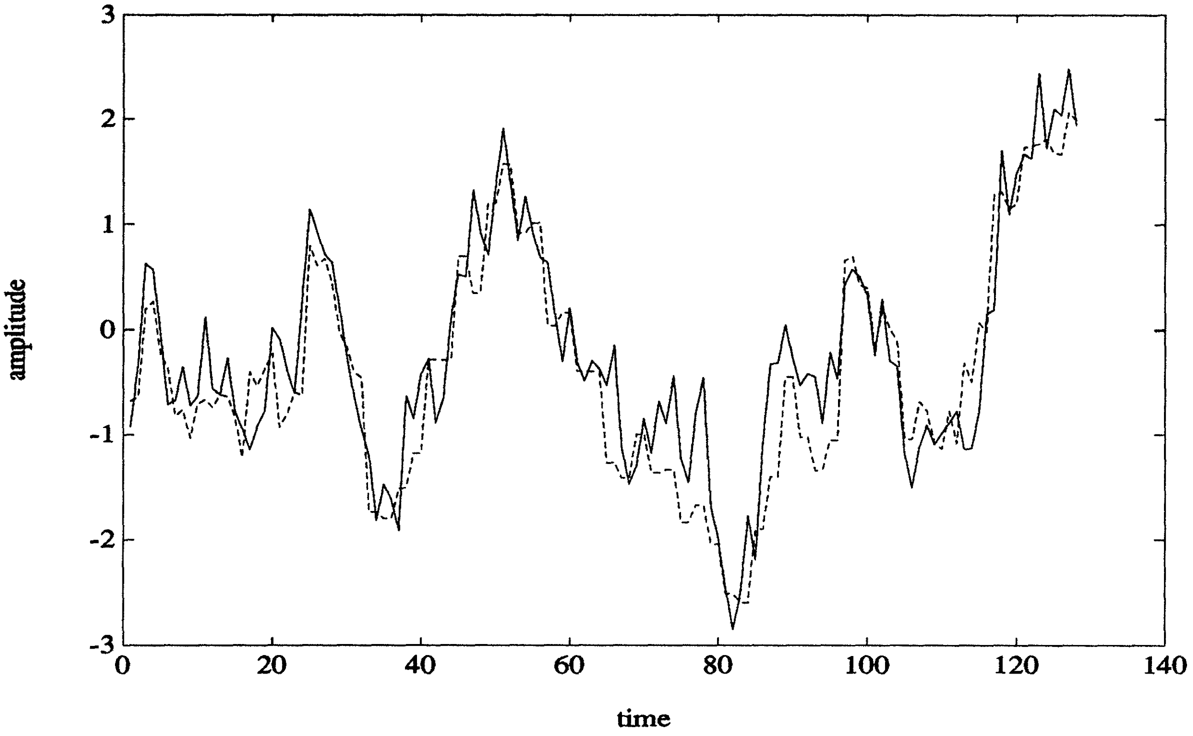

Figure 4.4.41: Sample Path of Stationary Gauss-Markov Process (solid), Result of Using Tree Smoother to Fuse Coarse (64 point) and Fine Data, Both of SNR = 1.4142 (dashed)

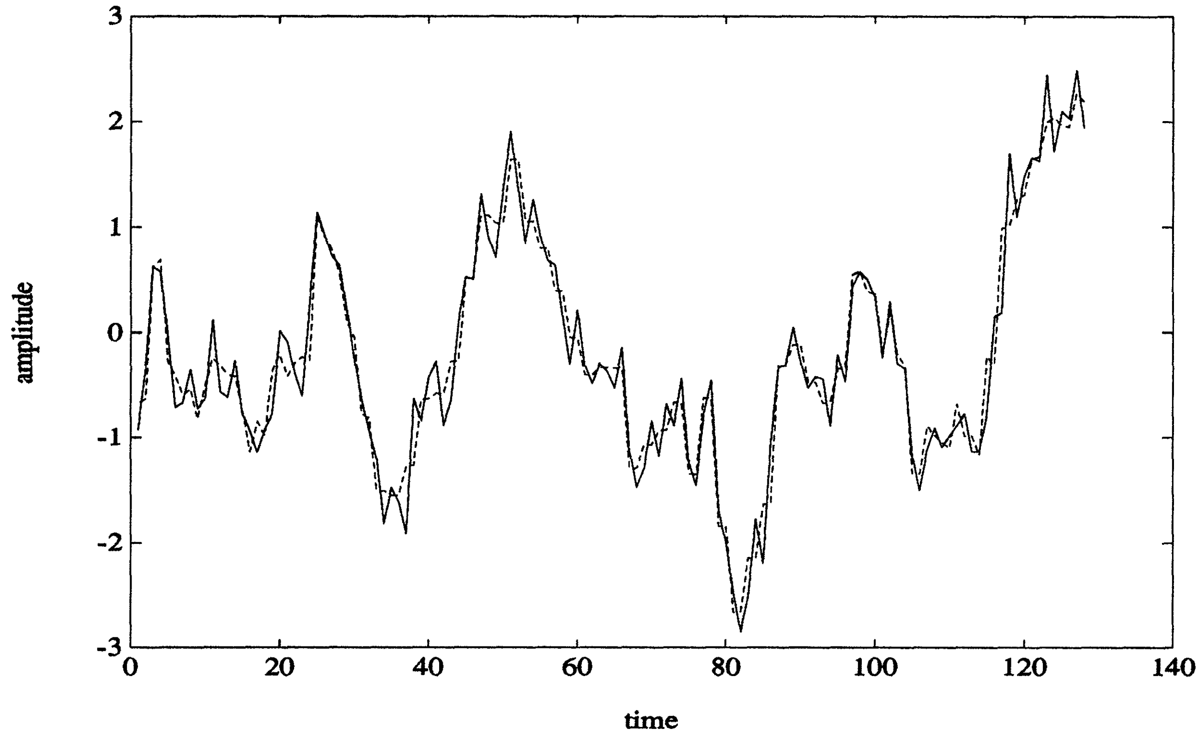

Figure 4.4.42: Sample Path of Stationary Gauss-Markov Process (solid), Result of Using Tree Smoother to Fuse High Quality Coarse Data (64 point) of SNR = 100 with Fine Data of SNR = 1.4142 (dashed)

emphasizing at this point that our smoother accommodates arbitrarily distributed data with no added computational complexity. To further illustrate the effects of coarse scale data on interpolation we provide an example in which the coarse data is of extremely high SNR; the result of this is illustrated in Figure 4.4.42. Note that in this case the interpolation is extremely good due to the near noiseless coarse data. We compare this with the example in Figure 4.4.43 in which the coarse data resides at the level of the lattice corresponding to 32 points, i.e. data which is 2 levels coarser than the fine data. Note that the estimates in the interior of the interval are resolved up to the scale given by the coarse data while the estimates near the ends are resolved much more finely due to the additional fine measurements there. This effect is even more noticeable in Figure 4.4.44 where the coarse data being fused is at a 16 point scale. To better illustrate this effect of the scale of the coarse data on the interpolation, Figure 4.4.45 compares the result of using coarse data at the three different

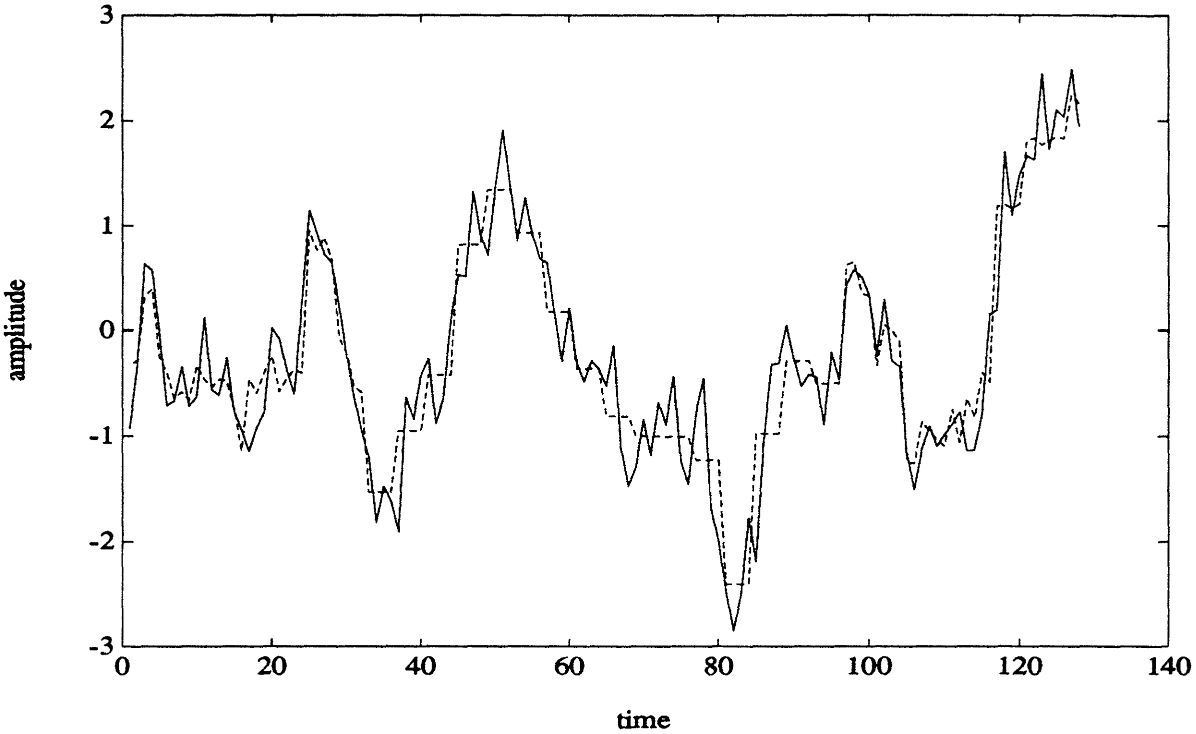

Figure 4.4.43: Sample Path of Stationary Gauss-Markov Process (solid), Result of Using Tree Smoother to Fuse High Quality Coarse Data (32 point) of SNR = 100 with Fine Data of SNR = 1.4142 (dashed)

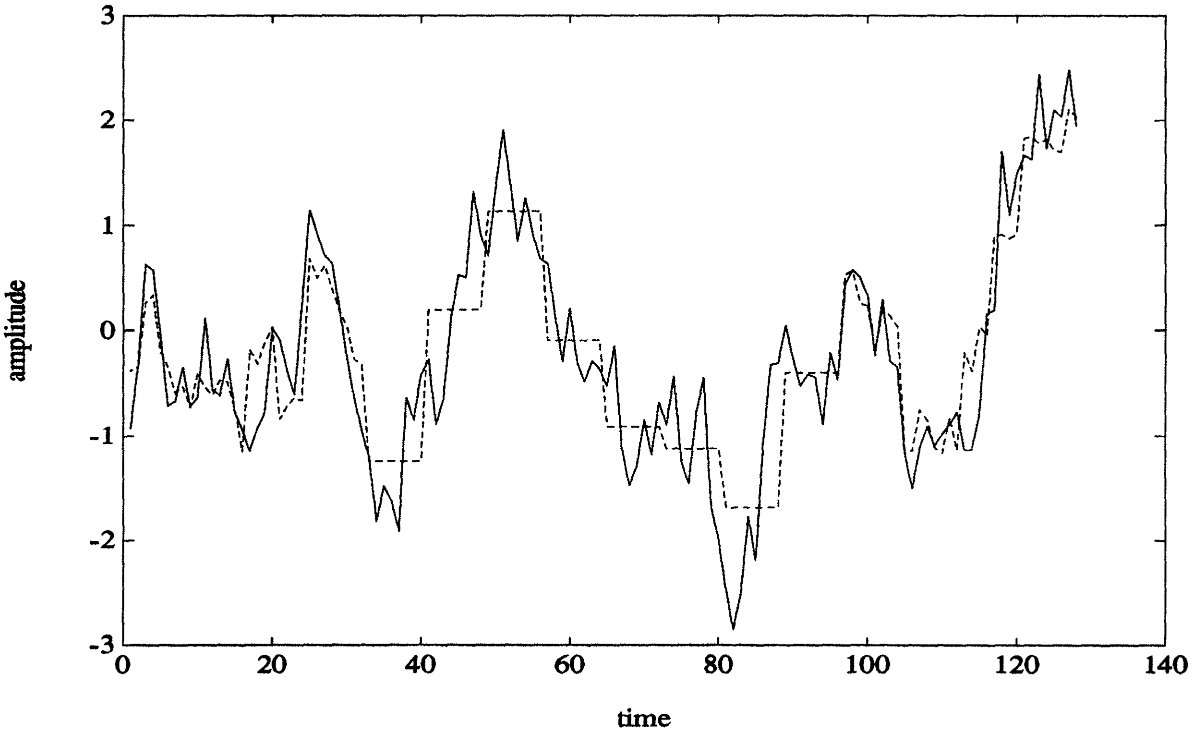

Figure 4.4.44: Sample Path of Stationary Gauss-Markov Process (solid), Result of Using Tree Smoother to Fuse High Quality Coarse Data (16 point) of SNR = 100 with Fine Data of SNR = 1.4142 (dashed)

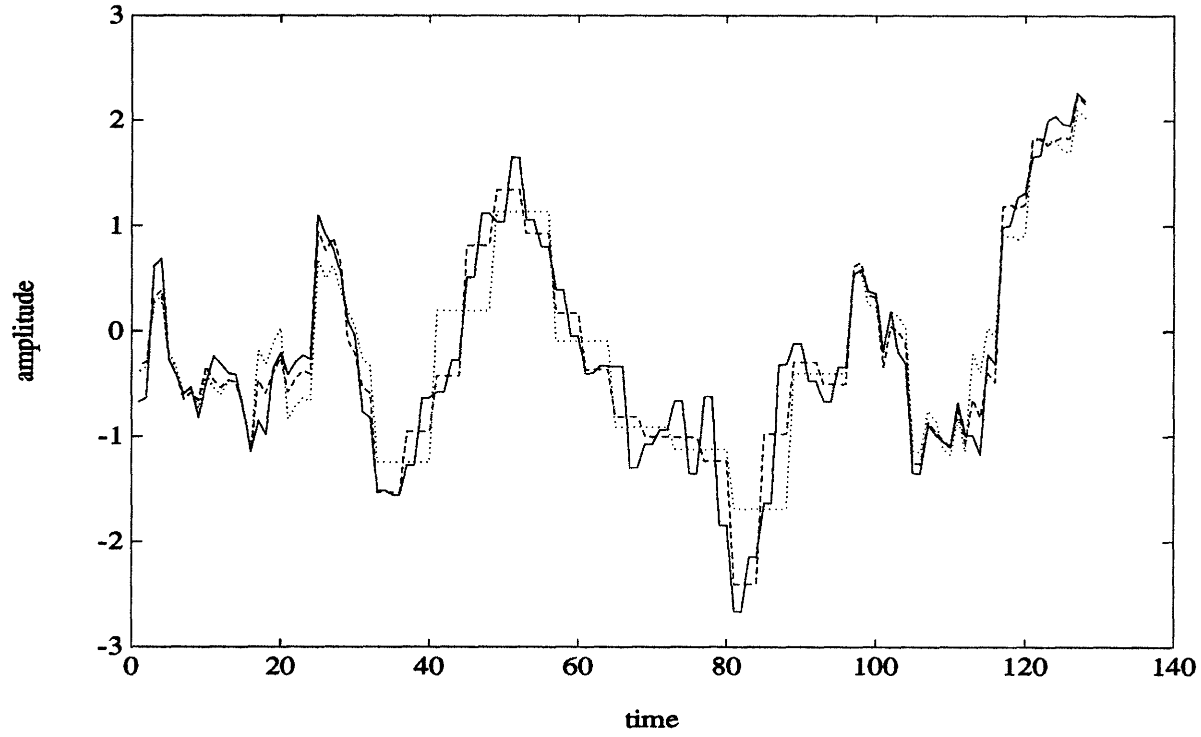

Figure 4.4.45: Result of Using Tree Smoother to Fuse High Quality Coarse Data (64 point) of SNR = 100 with Fine Data of SNR = 1.4142 (solid), Result of Using Tree Smoother to Fuse High Quality Coarse Data (32 point) of SNR = 100 with Fine Data of SNR = 1.4142 (dashed), Result of Using Tree Smoother to Fuse High Quality Coarse Data (16 point) of SNR = 100 with Fine Data of SNR = 1.4142 (dotted)

scales (64 pt., 32 pt., and 16 pt.).

In fact we can characterize the effect of coarse scale data on fine scale performance by plotting the smoothing errors for a variety of scenarios. Figure 4.4.46 is a plot of the smoothing error at the fine scale when full-coverage coarse data is fused with fine data which is sparse (zero data at at points 33 to 96). The plot shows cases where the coarse scale data ranges from one level coarser than the fine data to five levels coarser (4 points) and in each case the SNR's of the coarse data and the non-zero fine data are taken to be 1.4142. Note that although these performance plots are only approximate to the extent that our tree models approximate the Gauss-Markov process, the smoothing error computations come naturally from our Rauch-Tung-Striebel algorithm and the fact the data is distributed arbitrarily is handled at no

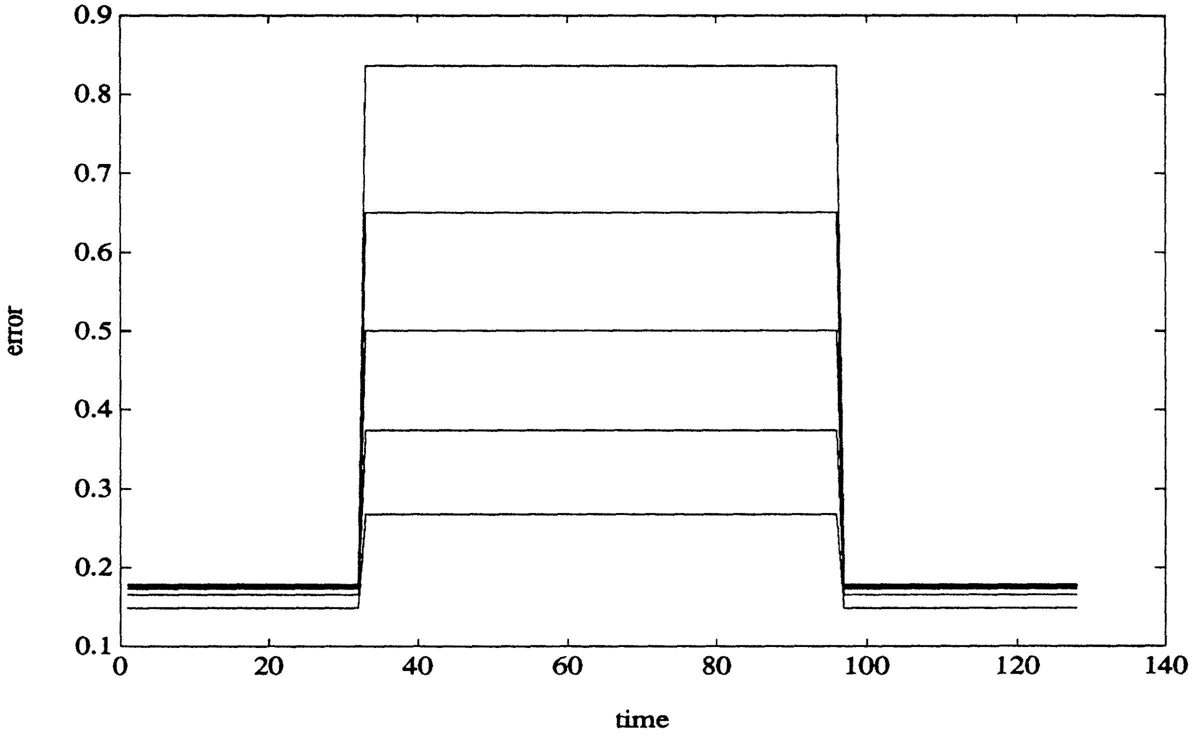

Figure 4.4.46: Plots of Performance for the Case of Full Coverage Coarse Data Fused with Sparse Fine Data Both of SNR = 1.4142. Five Plots Correspond to Coarse Data 1) One Level Coarser Than Fine Data 2) Two Levels Coarser ... 5) Five Levels Coarser

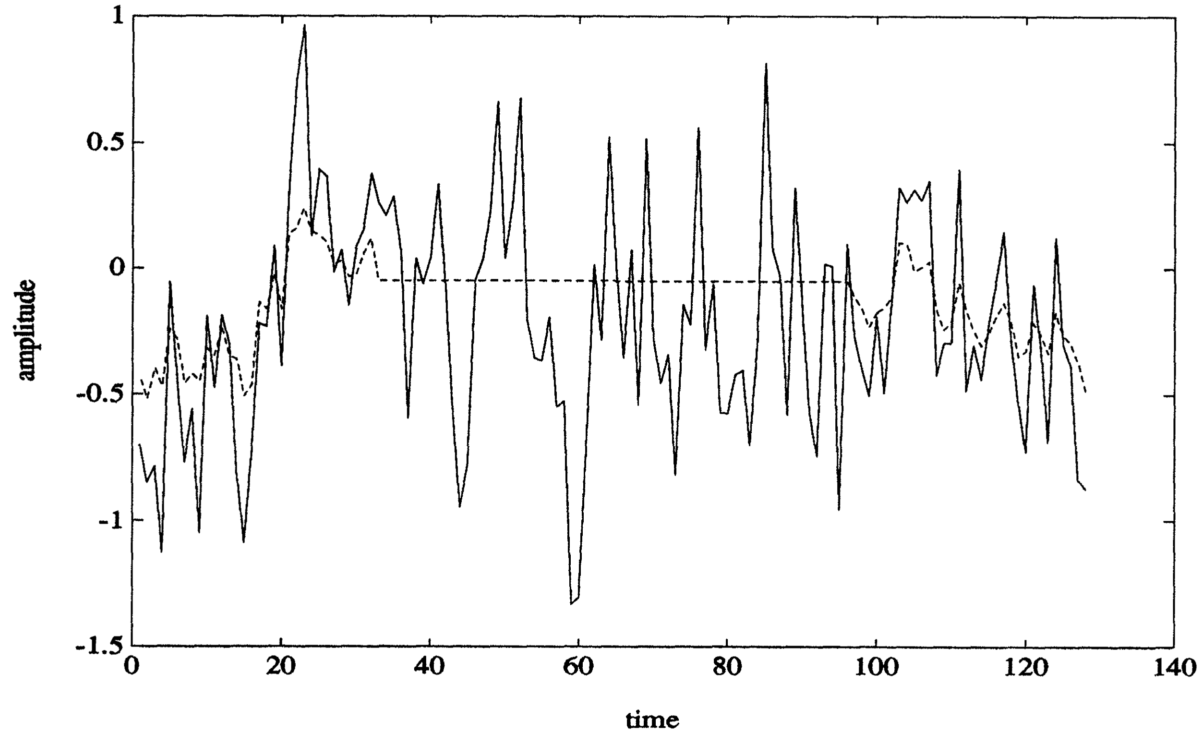

Figure 4.4.47: Sample Path of $1/f$ Process with $\gamma = 1$ (solid), Result of Smoother Using Sparse Fine Data of SNR = 1.4142 (dashed)

extra cost.

Finally, we present plots showing the results of using coarse data to interpolate sparse measurements of a $1/f$ process. We take a $1/f$ process with $\gamma = 1$ and assume again that our measurements at the finest scale are sparse; i.e. they are zero at points 33 to 96 of a 128 point sequence and of SNR = 1.4142 at the remaining points. Figure 4.4.47 shows the result of smoothing the data using just the sparse fine scale measurements. Figure 4.4.48 shows the result of fusing these sparse measurements with full-coverage coarse data, where the data is at the 64 point scale and of extremely high quality (SNR = 100). Again, we can see that the resolution of the estimate at the ends of the interval is better than that of the resolution in the middle, corresponding to the difference in resolution of the data in these two regions. This effect is further illustrated in Figures 4.4.49-4.4.51, where we compare the result of the fusion using data at 64, 32 and 16 point scales.

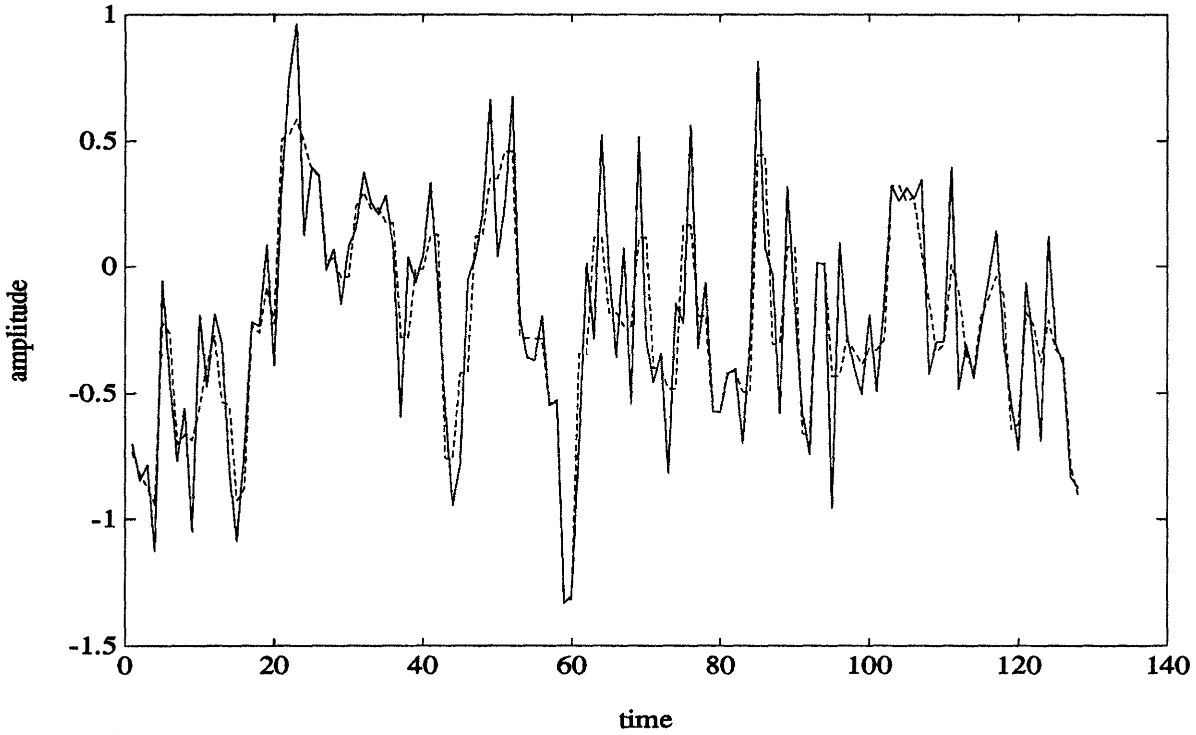

Figure 4.4.48: Sample Path of $1/f$ Process with $\gamma = 1$ (solid), Result of Fusing Sparse Data with Coarse Data of SNR = 100 at 64 point Scale (dashed)

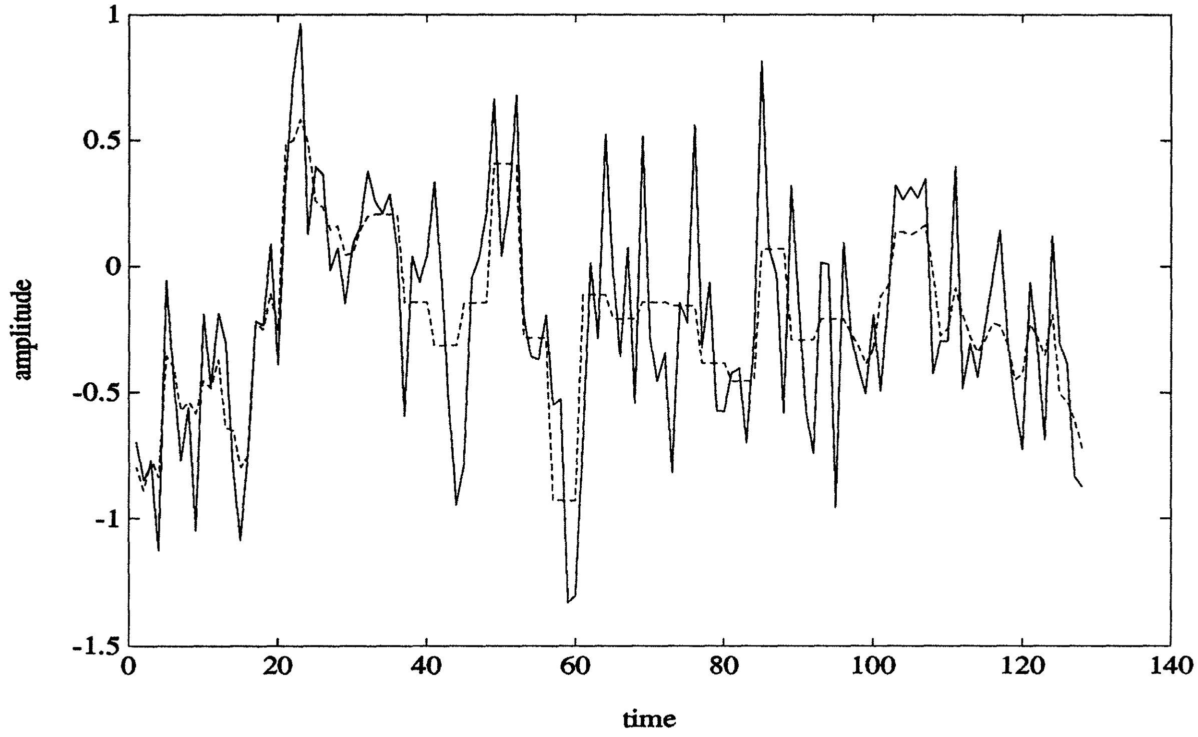

Figure 4.4.49: Sample Path of $1/f$ Process with $\gamma = 1$ (solid), Result of Fusing Sparse Data with Coarse Data of SNR = 100 at 32 point Scale (dashed)

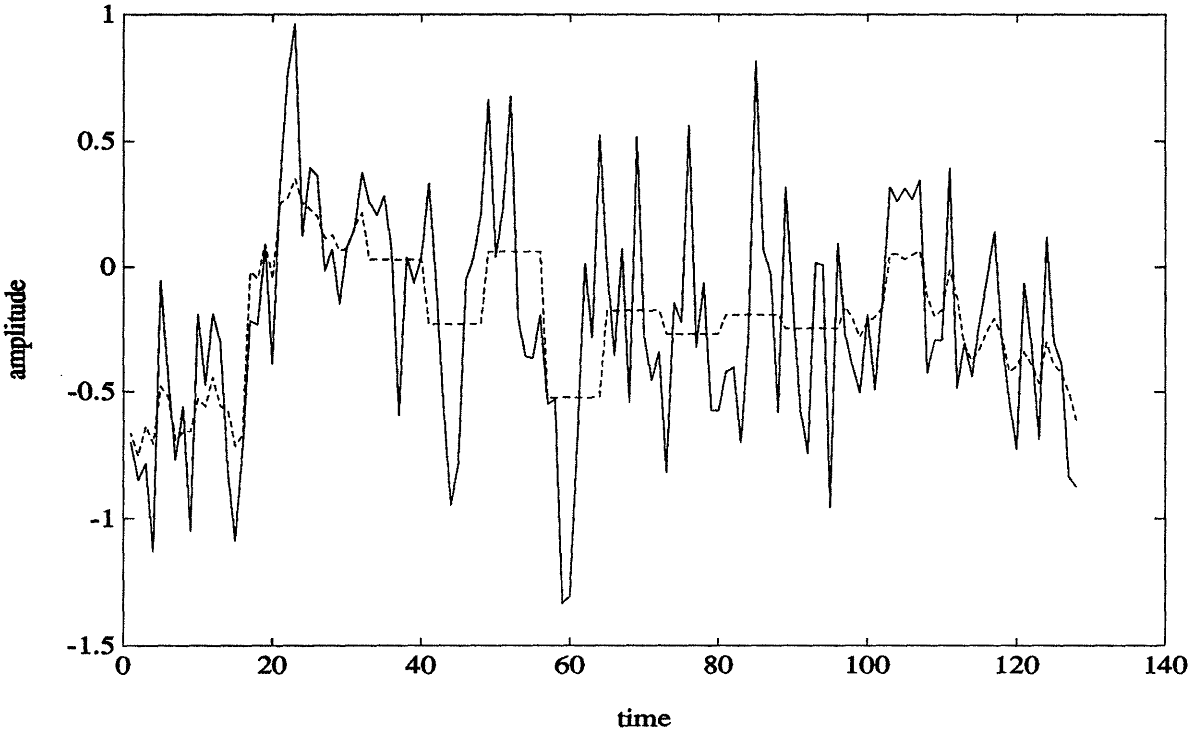

Figure 4.4.50: Sample Path of $1/f$ Process with $\gamma = 1$ (solid), Result of Fusing Sparse Data with Coarse Data of SNR = 100 at 16 point Scale (dashed)

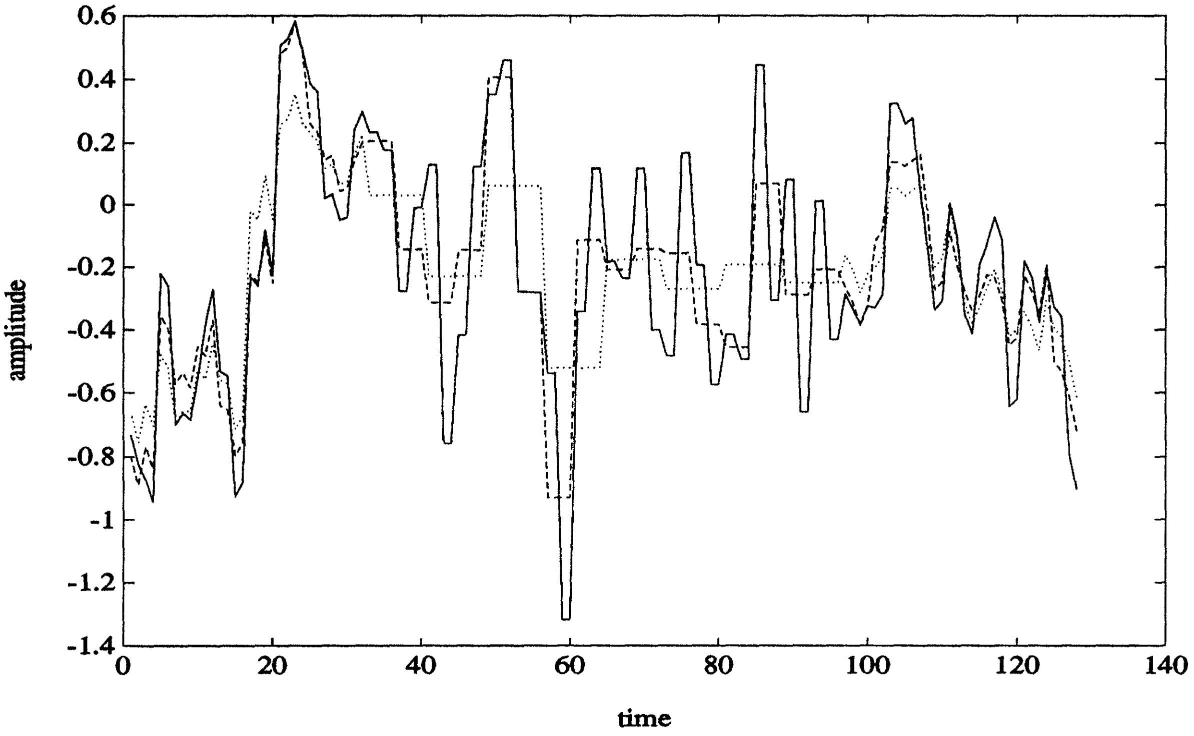

Figure 4.4.51: Result of Fusing Sparse Data with Coarse Data of SNR = 100 at 64 point Scale (solid), Result of Fusing Sparse Data with Coarse Data of SNR = 100 at 32 point Scale (dashed), Result of Fusing Sparse Data with Coarse Data of SNR = 100 at 16 point Scale (dotted)

## 4.5 Optical Flow

In this section we use our framework to formulate the problem in computer vision of determining the optical flow[30] from a sequence of images. Our purpose is to show the potential of our framework for solving this complex, computationally intensive problem. The results in this section focus on a 1D version of the problem as formulated in Horn and Schunck[30] and serve to show how our multiscale models can be used as the basis for a highly efficient, multiscale algorithm for computing optical flow. Note that everything we present in this section is extendible to 2D by using quadtrees. Typical methods for determining optical flow from 2D images are iterative, based on the idea of using relaxation algorithms as a way of handling the enormous computational complexity. Our algorithm on the other hand is a direct method based on recursions in scale which is both highly efficient and highly parallelizable.

### 4.5.1 1D Optical Flow Formulation

Let us denote the image intensity function, a function of both time and space, as $f(x,t)$. For our purposes we focus on the case where $x$ is a scalar quantity; i.e. $f(x,t)$ is a 1D image as a function of time. The constraint typically used in gradient-based approaches to motion estimation is referred to in the literature as the "brightness constraint"[30] and it amounts to assuming that the brightness of each point in the image, as we follow its movement, is constant. In other words the total derivative of the image intensity is zero, i.e.

$$\frac{Df}{dt} = 0 \tag{4.5.84}$$

This equation can be rewritten as follows.

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} = 0 \tag{4.5.85}$$

The quantity $\frac{\partial x}{\partial t}$ is referred to as the *optical flow*, which we denote as the spatial function, $v(x)$, and it is this quantity which we would like to determine from observing $f(x,t)$ sequentially in time.

We rewrite eq.(4.5.85) as follows.

$$y(x) \quad = \quad c(x)v(x) \tag{4.5.86}$$

$$y(x) = -\frac{\partial f}{\partial t} \tag{4.5.87}$$

$$c(x) = \frac{\partial f}{\partial x} \tag{4.5.88}$$

The following is the optimization problem formulation of Horn and Schunk[30] for determining $v(x)$.

$$\hat{v}(x) = ^{argmin}_{\ v} \{\mu \| y(x) - c(x)v(x)\|^2 + \|\frac{dv}{dx}\|^2\} \tag{4.5.89}$$

The second term in the cost function, $\|\frac{dv}{dx}\|^2$, is referred to as the "smoothness" constraint as it is meant to penalize large derivatives of the optical flow, constraining the solution to have a certain degree of smoothness. The variational solution to eq.(4.5.89) can be written implicitly as follows.

$$\mathcal{L}\hat{v}(x) = \mu(\int c(x)y(x) - \int c^2(x)dx) \tag{4.5.90}$$

$$\mathcal{L} = \frac{d^2}{dx^2} \tag{4.5.91}$$

Note that computing $\hat{v}(x)$ in eq.(4.5.90) is potentially daunting as the dimension of $\hat{v}(x)$ is equal to the number of pixels in the image.

As in Rougee et al[46] the optimization problem in eq.(4.5.89) can be interpreted as a stochastic estimation problem. In particular the smoothness constraint can be interpreted as the following prior model on $v(x)$.

$$\frac{dv}{dx} = w(x) \tag{4.5.92}$$

$$E[w(x_1)w(x_2)] = \delta(x_1 - x_2) \tag{4.5.93}$$

The estimation problem consists of estimating $v(x)$ based on the following observation equation.

$$y(x) = c(x)v(x) + r(x) \tag{4.5.94}$$

$$E[r(x_1)r(x_2)] = \mu^{-1}\delta(x_1 - x_2) \tag{4.5.95}$$

Note that since we are interested in estimating $v(x)$ based on $y(x)$ for all $x$, we are in fact concerned with a smoothing problem. Henceforth, we will refer to eq.'s(4.5.92,4.5.93) as the *standard model.*

The incorporation of the smoothness constraint becomes useful for several reasons. First, if the measurement in eq.(4.5.94) is extremely noisy, it is necessary to incorporate a prior term in order to provide a certain degree of regularity in the estimate of $v(x)$. Secondly, for portions of $f(x,t)$ which have zero derivative the data at these locations are missing, i.e. $c(x) = 0$, and the prior term provides a way of interpolating the sparse data. Finally, in the 2D case, where the optical flow is a 2D vector field in both the $x$ and $y$ directions, we actually have an underspecified system. That is, at each point in the image there is one measurement equation for *two* unknowns. In this case the smoothness constraint provides additional constraints for determining the solution.

Let us now examine the particular prior model associated with the smoothness constraint. From eq.'s(4.5.92,4.5.93) we see that $v(x)$ is in fact a *Brownian motion process*. Note that the Brownian motion process is a special case of the class of fractional Brownian motion processes, a class of processes that exhibit $1/f$ spectral characteristics. Since, as we showed in previous sections, our multiscale models do in fact provide good approximations to $1/f$ processes, we would expect to be able to use our multiscale models in place of eq.'s(4.5.92,4.5.93) as a prior model for $v(x)$. This would of course allow us to solve the estimation problem in our framework using the extremely efficient and highly parallelizable algorithms described in the previous chapters.

Let us consider $v(x)$ to be modeled as one of our multiscale models. Note that our observation equation, eq.(4.5.94), has a spatially varying $c(x)$, which would disallow the use of our transform method for smoothing lattice processes. Thus, if we were to consider using a lattice model for $v(x)$, we would need to solve for the smoothed estimate iteratively. If we consider $v(x)$ to be modeled by a tree model, however, our Rauch-Tung-Striebel tree smoother can be applied. Note also that the use of our tree smoother can be easily extended to the case of 2D by simply considering quadtree models as opposed to our 1D dyadic tree models. Thus, for the examples in this section we use a tree model as a model for the optical flow, $v(x)$.

### 4.5.2 Inherent Problems, Discretization

The formulation in the previous section is convenient and simple in that it consists of a linear measurement equation, eq.(4.5.86), which can be easily supplemented with prior models. In this section we point out some of the problems known to exist in using this approach on *noisy* images. Note that these problems are inherent to the problem itself, and that their treatment remains a critical question in computational vision. Our examples serve simply to show the viability of our multiscale modeling approach as an efficient method for computing optical flow and to provide a basis for further investigation of applying our approach to estimating optical flow. Thus, although we acknowledge the existence of these problems, we do not delve deeply into them in this thesis. Also, in this section we discretize the smoothing problem formulated in the previous section, providing the basis for our numerical results in the next section.

We begin by describing how noise in our measurements of $f(x,t)$ enters into our formulation. Note that what we have available as data are the measurements of $f(x,t)$ sequentially in time. It is from these measurements that we compute the partial derivatives $\frac{\partial f}{\partial t}$ and $\frac{\partial f}{\partial x}$. Consider the following model for our measured $f(x,t)$.

$$\begin{aligned} f_{measured}(x,t) &= f(x,t) + u(x,t) && (4.5.96) \\ E[u(x,s)u(y,t)] &= \epsilon\delta(x-y, s-t) && (4.5.97) \end{aligned}$$

Thus, if we use $f_{measured}(x,t)$ to compute $y(x)$ and $c(x)$, we get

$$\begin{aligned} y_{measured}(x) &= y(x) - u_2(x) && (4.5.98) \\ c_{measured}(x) &= c(x) - u_1(x) && (4.5.99) \\ u_1(x) &= -\frac{\partial u(x,t)}{\partial x} && (4.5.100) \\ u_2(x) &= \frac{\partial u(x,t)}{\partial t} && (4.5.101) \end{aligned}$$

Note that $c(x)$ is itself noisy. Furthermore, if we substitute eq.'s(4.5.98,4.5.99) into our observation equation, eq.(4.5.94), we get

$$y_{measured}(x) = (c_{measured}(x) + u_1(x))v(x) + u_2(x) + r(x) \qquad (4.5.102)$$

Thus, in terms of our measured quantities, $y_{measured}(x)$ and $c_{measured}(x)$, we have a *nonlinear* estimation problem. Note that even if we assume $u_1(x)$ and $v(x)$ to be Gaussian, $y_{measured}(x)$ is not due to their product in eq.(4.5.102). There is clearly a need for taking into account higher order moments in this case and a tractable solution requires further investigation. The typical method used to partially cirvumvent this problem is to prefilter the image $f(x,t)$ before computing $c_{measured}$, thus minimizing the variance of $u_1(x)$. For our purposes we simply wish to compare the use of our multiscale models with the use of the standard model, i.e. the model implied by the standard regularization approach. Thus, in our numerical examples we avoid this issue entirely by assuming $c(x)$ to be known exactly.

We now turn to the discretization of the standard model, eq.'s(4.5.92,4.5.93), and the observation equation, eq.(4.5.94). We need to estimate $v(x)$ based on *approximations* of both $\frac{\partial f}{\partial t}$ and $\frac{\partial f}{\partial x}$. We consider the following finite difference approximations of these partial derivatives.

$$\frac{\partial f}{\partial t}|_{x,t} \approx f(x,t+1) - f(x,t) \qquad (4.5.103)$$

$$\frac{\partial f}{\partial x}|_{x,t} \approx (f(x+1,t) - f(x-1,t))/2 \qquad (4.5.104)$$

Let us assume that our image is available at two time instants, $t$ and $t+1$, where each image is uniformly sampled over a finite interval in space; i.e. we have $f(i,t)$ and $f(i,t+1)$ where $i \in \{0,1,2,...N-1\}$. The discretized smoothing problem for our standard model is as follows.

$$v(i+1) - v(i) = w(i) \qquad (4.5.105)$$

$$E[w(i)w(j)] = \delta(i-j) \qquad (4.5.106)$$

$$y(i) = c(i)v(i) + r(i) \qquad (4.5.107)$$

$$c(i) = f(i+1,t) - f(i,t) \qquad (4.5.108)$$

$$E[r(i)r(j)] = \mu^{-1}\delta(i-j) \qquad (4.5.109)$$

$$i \in \{0,1,...N-2\} \qquad (4.5.110)$$

The following is the linear least-squares solution to the discrete smoothing problem for our standard model.

$$\begin{align}
\hat{v} &= (\mathcal{L} + \mu^{-1}C^TC)^{-1}(\mu^{-1}C^T)\bar{y} \tag{4.5.111}\\
\bar{y} &= [f(0,t+1) - f(0,t), f(1,t+1) - f(1,t), \ldots f(N-2,t+1) - f(N-2,t)]^T \nonumber\\
v &= [v(0,t), v(1,t), \ldots v(N-2,t)]^T \tag{4.5.112}
\end{align}$$

where

$$C = \begin{bmatrix} c(0) & 0 & \cdots & \cdots & 0 \\ 0 & c(1) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & c(N-3) & 0 \\ 0 & \cdots & & 0 & c(N-2) \end{bmatrix} \tag{4.5.113}$$

and

$$\mathcal{L} = \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & \cdots & 0 \\ & & & \ddots & \ddots & \ddots & \\ \vdots & & & & & & \vdots \\ 0 & \cdots & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & & 0 & -1 & 1 \end{bmatrix} \tag{4.5.114}$$

Note that the matrix $\mathcal{L}$, which is a discrete approximation of the operator $\frac{d^2}{dx^2}$, does not assume the boundary points to be known. In terms of our prior model, eq.'s(4.5.105,4.5.106), this corresponds to assuming infinite variance of the process at each endpoint. Thus, the smoother resulting from the use of this model is a Maximum-Likelihood smoother.

### 4.5.3 Numerical Examples

In this section we give numerical examples of applying our framework to the optical flow problem using the discrete formulation from the previous section, where we consider two cases. We compare the solution produced using the discretized standard model, i.e. eq.'s(4.5.105,4.5.106), with the solution produced by using one of our tree models. These examples show that our tree smoother performs comparably to the smoother resulting from the standard model. Note, however, that our smoothing algorithm, which is based on recursions in scale, is extremely efficient, highly parallelizable, and extendible to 2D. Although for the case of 1D a Kalman filtering approach can be used to compute the smoother[46], in 2D no such recursive procedure exists.

In our examples we consider our image to be a 128 point sequence; i.e. $N = 128$. The multiscale model we use as our prior model for $v(x)$, as an alternative to our discretized standard model, is the following scalar tree model.

$$\begin{aligned} z_t &= a z_{\gamma^{-1}t} + \sigma 2^{-\frac{\delta m(t)}{2}} w_t && (4.5.115) \\ E[w_t^2] &= 1 && (4.5.116) \\ E[z_{t_0}^2] &= p_0 && (4.5.117) \end{aligned}$$

where $t$ indexes the nodes of a finite tree with $N$ points at the bottom level. Thus, the covariance of this zero-mean process at the bottom level of the tree is specified entirely by the parameter vector $\theta = [a, p_0, \gamma, \sigma]$. That is,

$$\begin{aligned} E[zz^T] &= P(\theta) && (4.5.118) \\ z &= [z_0, z_1, ...z_{N-1}]^T && (4.5.119) \end{aligned}$$

where we've indexed the points at the bottom level as $0, 1, ...N - 1$.

As we pointed out earlier the standard model corresponds to the Brownian motion process, a process which our tree model does well in approximating. We focus now on the problem of using our scalar tree model to approximate the discrete standard model in eq.'s(4.5.105,4.5.106) defined for the points $i = 0, 1, ...N - 1$. Note that since the standard model has no boundary conditions specified, it does not have a well defined covariance matrix. Thus, in order to approximate this model we consider instead the discretized operator, $\mathcal{L}$, which represents the information matrix of the process described by the standard model. To choose the parameters of our tree model so as to yield an approximation to the standard model, we fit the information matrix of our tree process, $P^{-1}(\theta)$, to $\mathcal{L}$ by minimizing the matrix 2-norm (the largest singular value of the matrix) of the difference between $\mathcal{L}$ and $P^{-1}(\theta)$.

$$\theta_{fitted} = \overset{argmin}{\theta} \; |\mathcal{L} - P^{-1}(\theta)|_2 \tag{4.5.120}$$

Note that in using our multiscale smoother, as in the case with using the standard model, we would not wish to assume boundary information. Thus, we must actually use the Maximum Likelihood form of the smoother. This is achieved by assuming an infinite variance for the point of the process at the top node of the tree.

In comparing the performance of the multiscale smoother with the performance of the standard regularization method, we need a way of normalizing the problem. We define the following quantity, which can be thought of as the ratio between the information due to measurements and the information due to the model.

$$\Gamma \triangleq \frac{trace(\mu C^T C)}{trace(\mathcal{I})} \tag{4.5.121}$$

where $\mathcal{I}$ is either $\mathcal{L}$ or $P^{-1}(\hat{\theta})$. For our examples, we vary $\Gamma$ by varying $\mu$, which is the inverse of the measurement noise variance.

We begin with examples using noiseless data. Consider an image which is a sinusoid moving with constant velocity. In particular we take $v(i,t) = 3$ for all $i, t$. Figure 4.5.52 shows snapshots of the image at times $t$ and $t+1$. We now compare the performance of estimating $v$ based on the images at these two time instants using our tree smoother and using the standard regularization.

Figure 4.5.53 shows the result of estimating $v$ based on standard regularization for $\Gamma = 1, .1, .01$. Figure 4.5.54 shows the result of estimating $v$ based on our tree smoother for $\Gamma = 1, .1, .01$. Recall that the optimal solution should be a constant equal to 3. Note that the two approaches yield similar results. In fact for all three values of $\Gamma$ our tree smoother actually performs better. As we would expect by decreasing $\Gamma$, i.e. decreasing the weight $\mu$ of the measurment term in the cost function, the prior term would have more effect on the solution. In both cases the solution becomes more and more regular with increased reliance on the prior model.

We now turn to an example using images consisting of a concatenation of trapezoids which are moving with constant velocity, $v(i,t) = 3$. Figure 4.5.55 shows snapshots of the image at times $t$ and $t+1$.

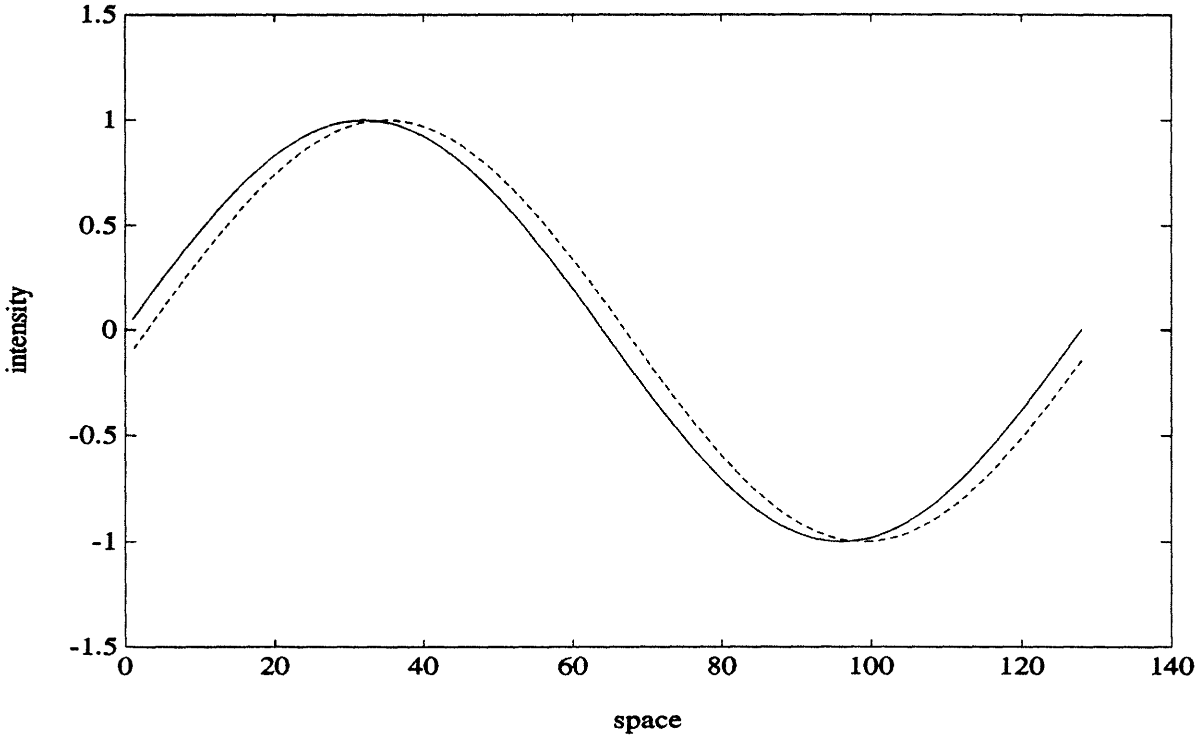

Figure 4.5.52: Noiseless Sinusoid at Time $t$ (solid) and at Time $t+1$ (dashed) ; Constant Velocity

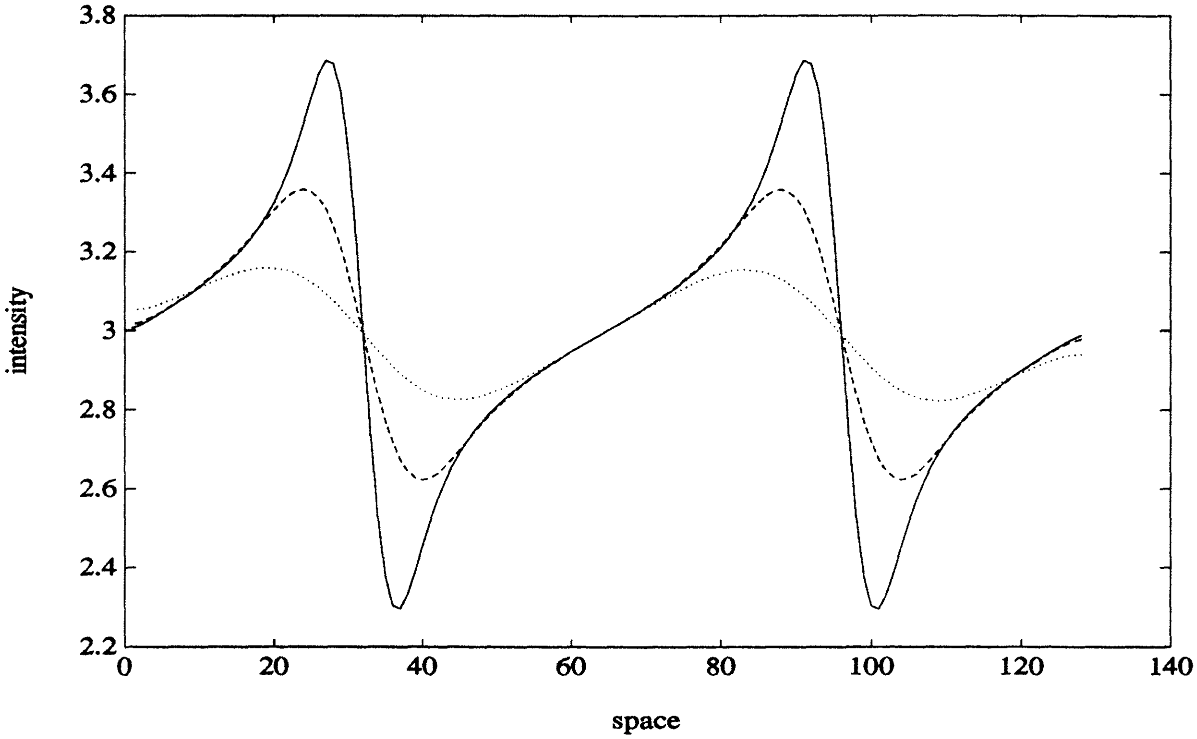

Figure 4.5.53: Estimate of $v(x)$ Using Standard Regularization for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

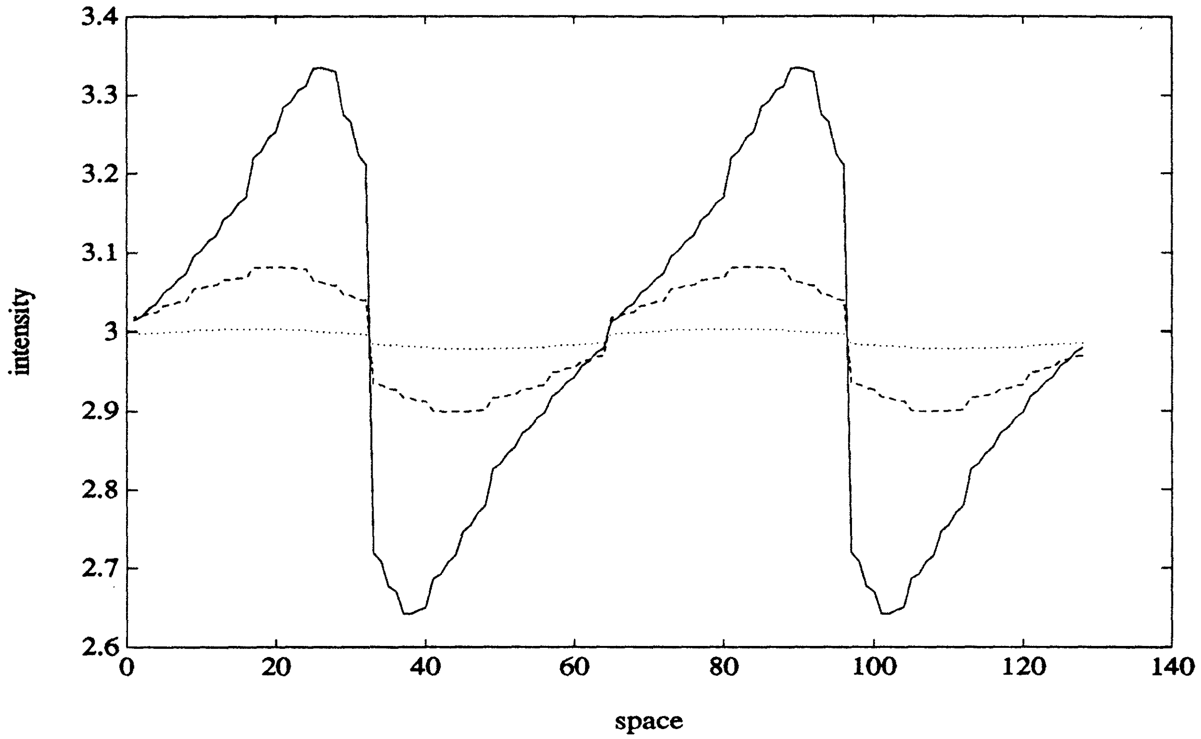

Figure 4.5.54: Estimate of $v(x)$ Using Tree Smoother for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

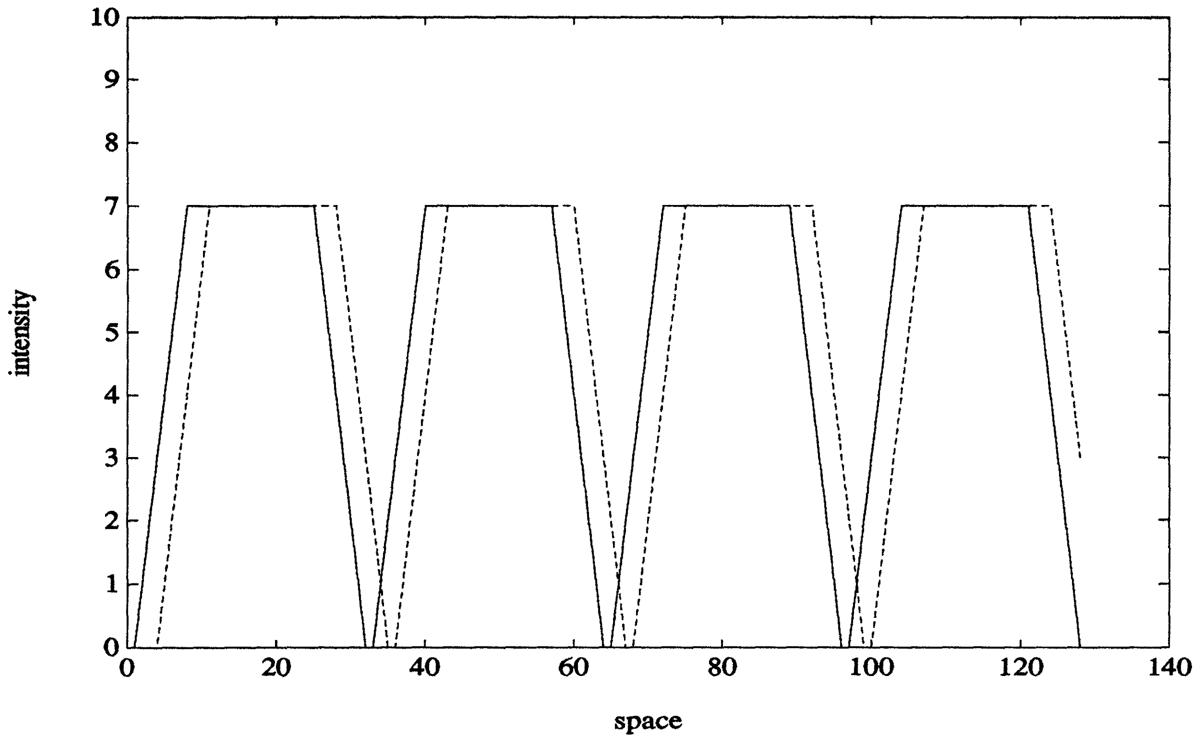

Figure 4.5.55: Noiseless Image of Shifted Trapezoids at Time $t$ (solid) and at Time $t+1$ (dashed) ; Constant Velocity

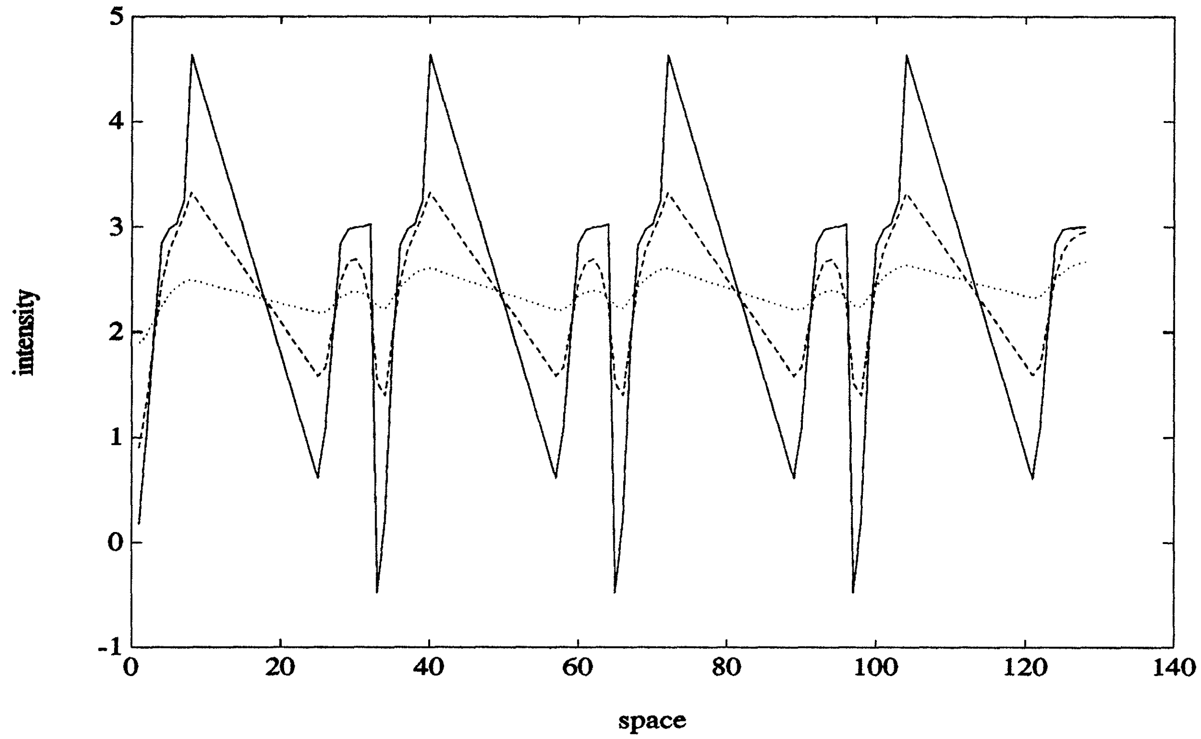

Figure 4.5.56: Estimate of $v(x)$ Using Standard Regularization for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

Figure 4.5.56 shows the result of estimating $v$ based on standard regularization for $\Gamma = 1, .1, .01$. Figure 4.5.57 shows the result of estimating $v$ based on our tree smoother for $\Gamma = 1, .1, .01$. Note that again as one increases the weight of the prior term relative to the measurement term the degree of regularity in the solution increases. In this example solutions due to both the tree smoother as well as the standard regularization are rather irregular due to the fact that the trapezoid has discontinuous derivatives. Also, the flat portions of the image which have zero derivative represent portions of the image with missing data. This is probably the source of the bias existing in the result of both smoothers.

We now turn to examples in which our images, $f(i,t)$, are noisy. In this case as mentioned in the previous section we assume the matrix of spatial derivatives $C$ to be known exactly; i.e. we compute $C$ from noiseless versions of $f(i,t)$. We do, however, take our vector of temporal derivatives, $\overline{y}$, to be formed using the noisy images at times $t$ and $t+1$. We begin with the sinusoid example moving at constant velocity,

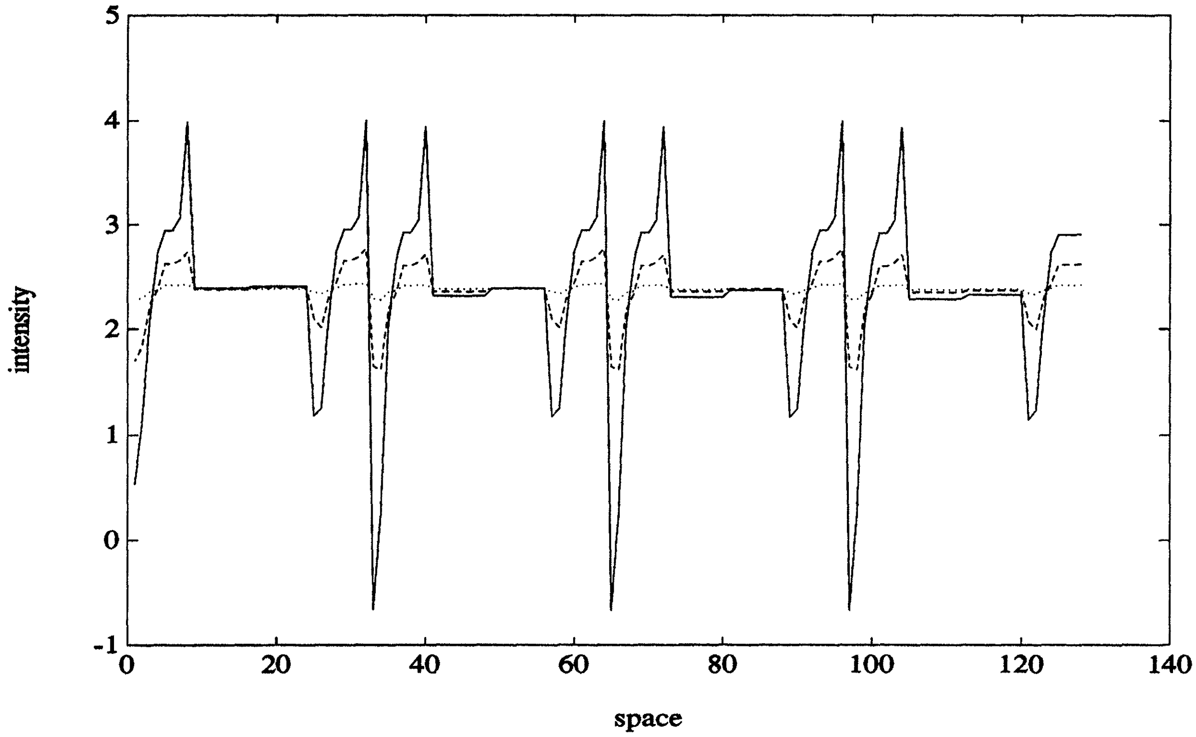

Figure 4.5.57: Estimate of $v(x)$ Using Tree Smoother for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

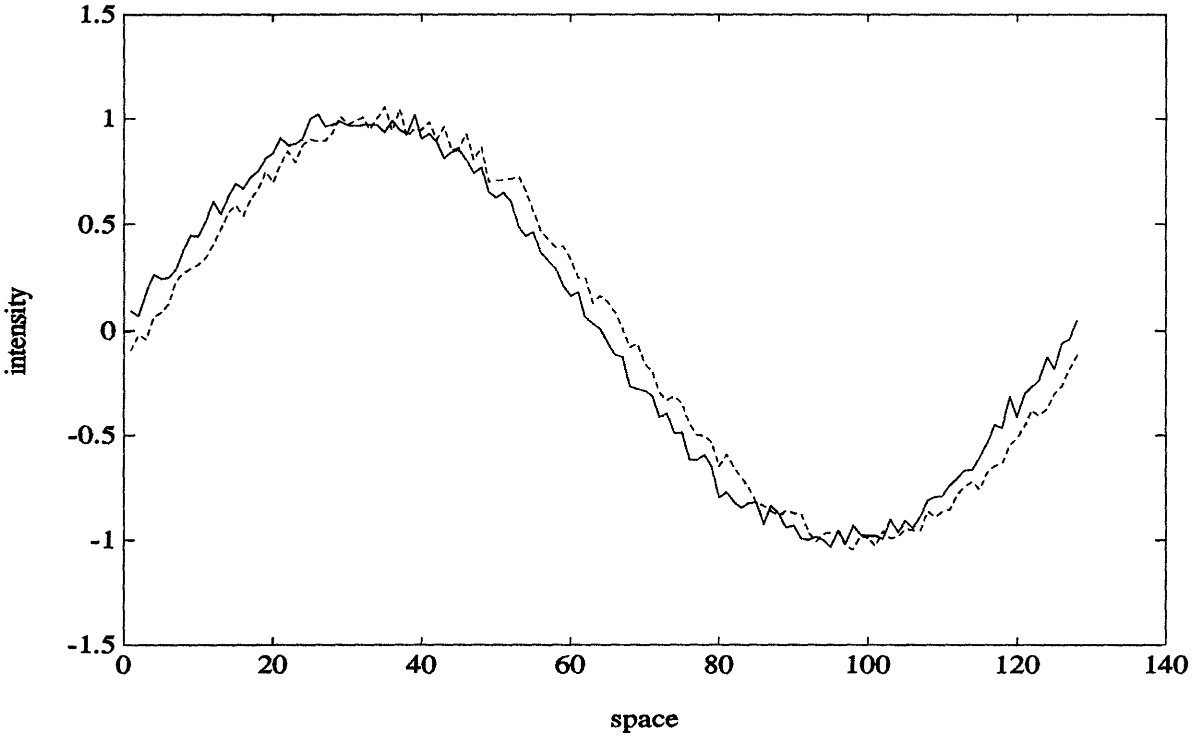

Figure 4.5.58: Noisy Sinusoid at Time $t$ (solid) and at Time $t+1$ (dashed); Noise Variance = .001, Constant Velocity

$v(i,t) = 3$. Figure 4.5.58 shows snapshots of the image at times $t$ and $t+1$.

Figure 4.5.59 shows the result of estimating $v$ based on standard regularization for $\Gamma = 1, .1, .01$. Figure 4.5.60 shows the result of estimating $v$ using our tree smoother for $\Gamma = 1, .1, .01$. In this case the noise has a definite effect on the regularity of the estimates. Note that again decreases in $\Gamma$ yield more regular estimates of $v(x)$. Note also that as in the noiseless case our smoother performs consistently better than the smoother based on the standard model.

In Figure 4.5.61 we have a noisy version of our trapezoidal image moving with velocity $v(i,t) = 3$. Figure 4.5.62 shows the result of estimating $v$ based on standard regularization for $\Gamma = 1, .1, .01$ and Figure 4.5.63 shows the result of estimating $v$ using our tree smoother for $\Gamma = 1, .1, .01$. Note that in both cases the estimates are biased. Again, this is probably due to the fact that the image has large portions where the data is missing, i.e. the spatial derivative is zero. The presence of noise degrades even further the information in the portions where the data are missing.

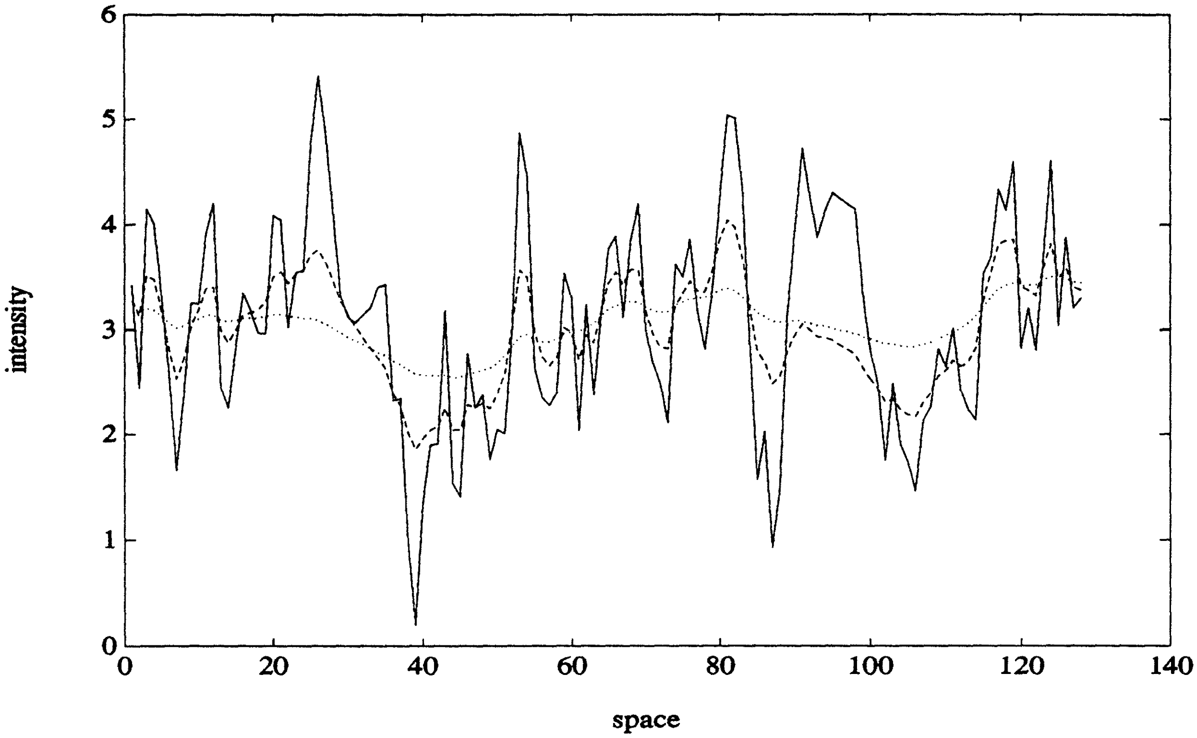

Figure 4.5.59: Estimate of $v(x)$ Using Standard Regularization for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

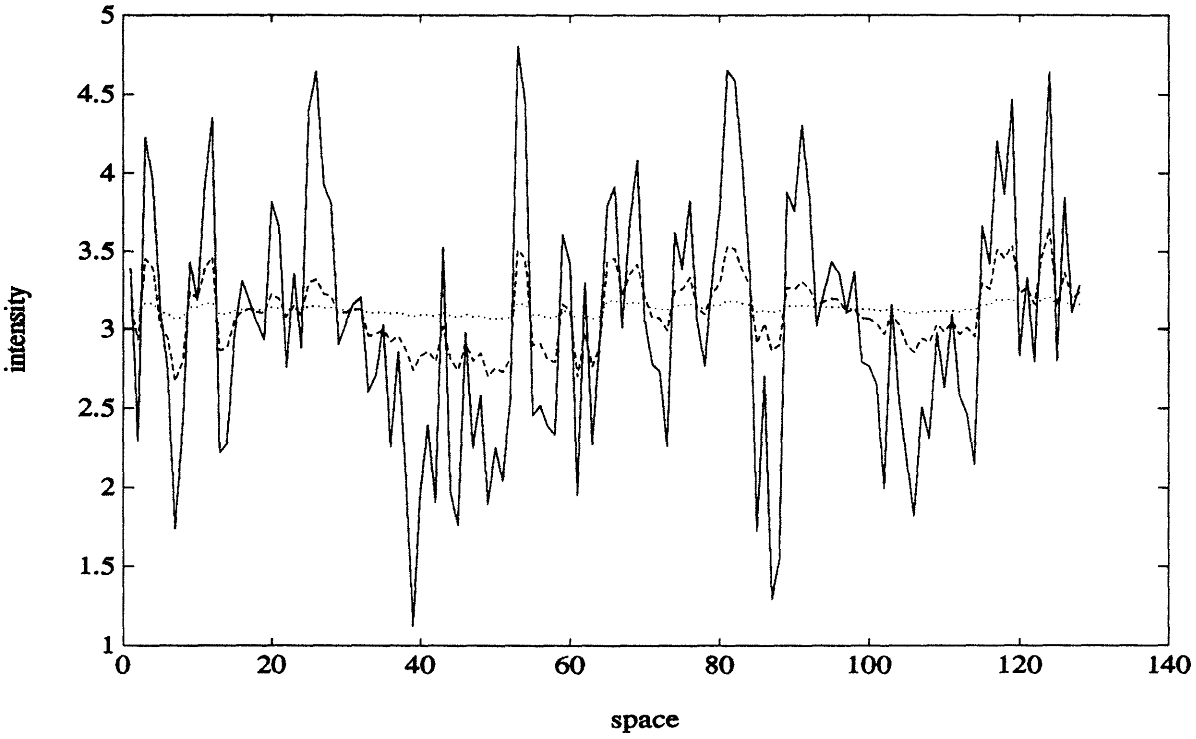

Figure 4.5.60: Estimate of $v(x)$ Using Tree Smoother for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

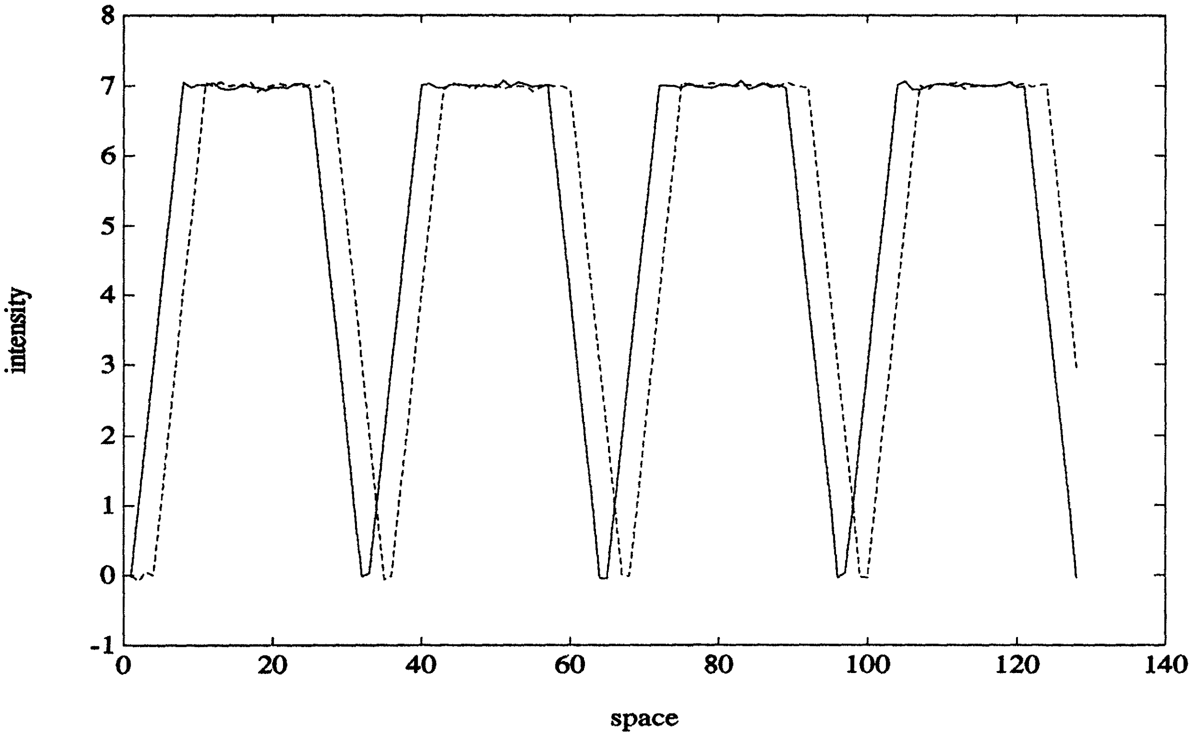

Figure 4.5.61: Noisy Trapezoid at Time $t$ (solid) and at Time $t+1$ (dashed); Noise Variance = .001, Constant Velocity

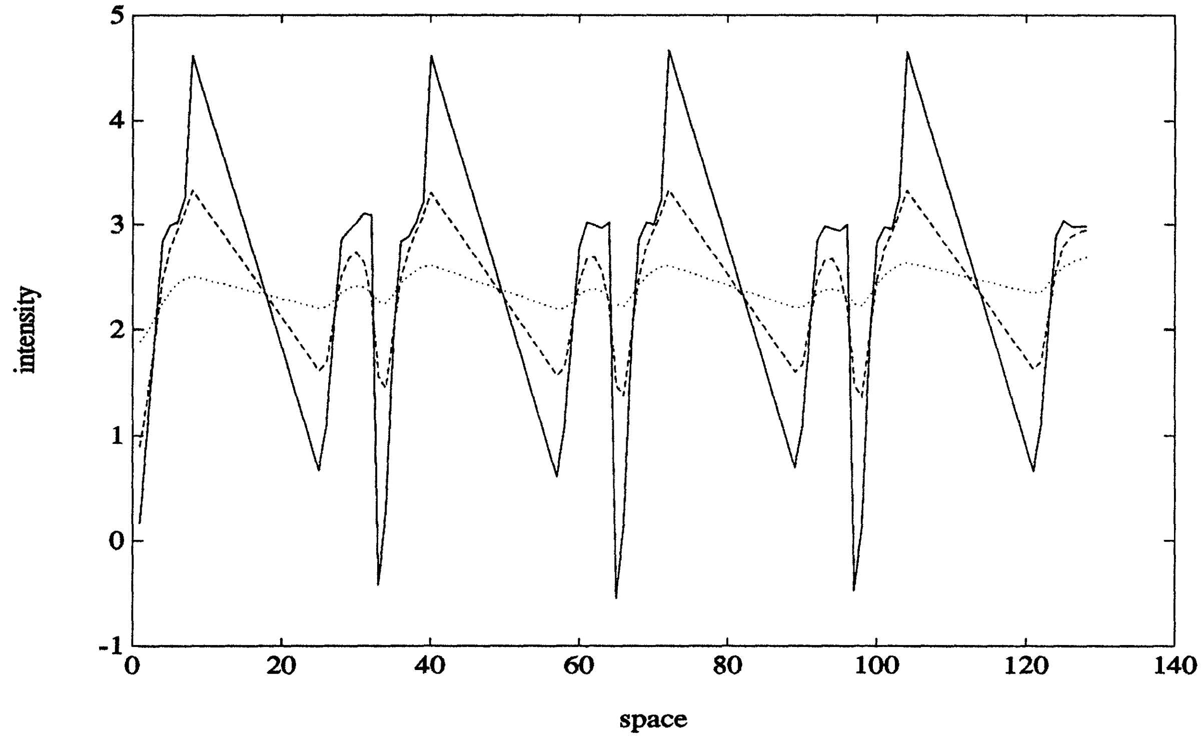

Figure 4.5.62: Estimate of $v(x)$ Using Standard Regularization for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

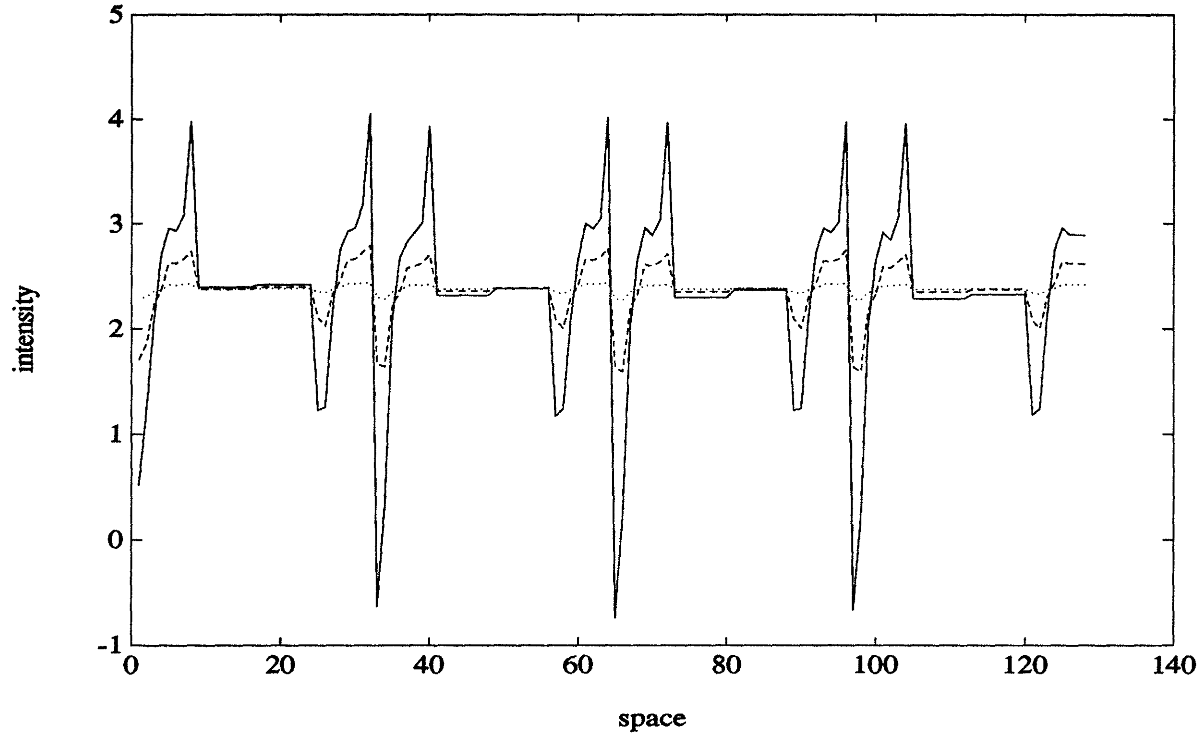

Figure 4.5.63: Estimate of $v(x)$ Using Tree Smoother for $\Gamma = 1$ (solid), for $\Gamma = .1$ (dashed), and for $\Gamma = .01$ (dotted)

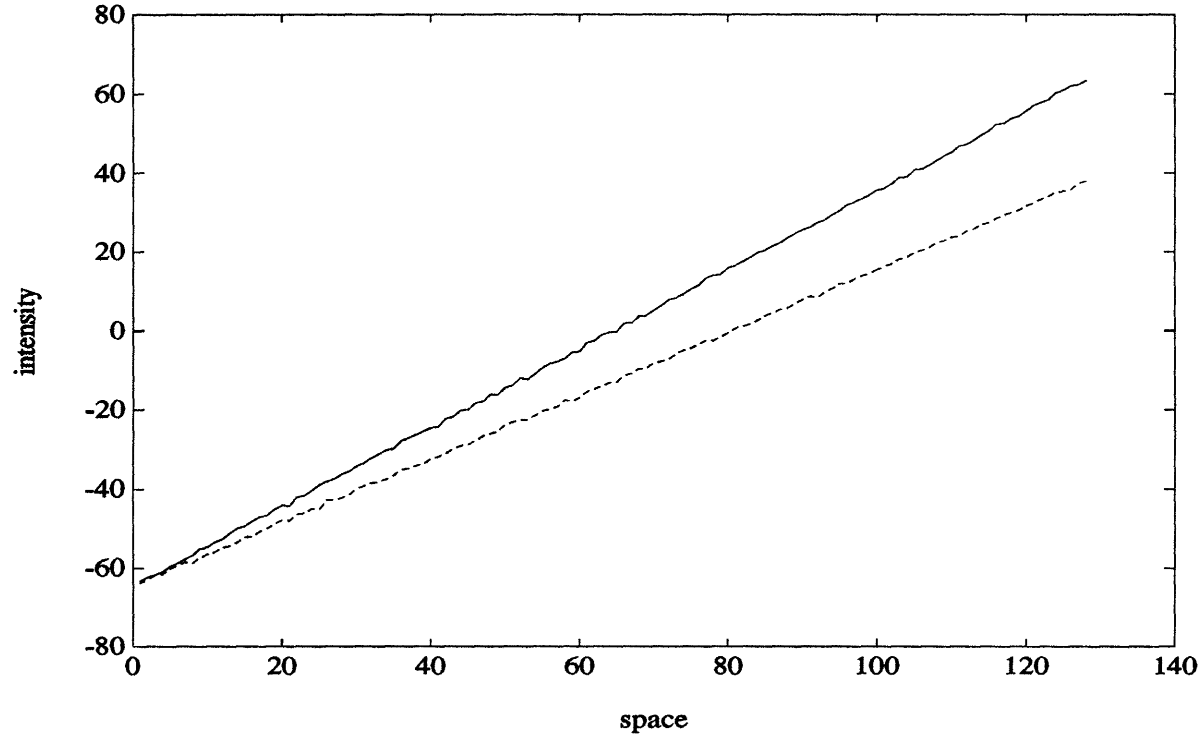

Figure 4.5.64: Noisy Line at Time $t$ (solid) and at Time $t+1$ (dashed); Noise Variance $= .01$, Velocity $v(x) = .2x$

Also, discontinuities in the spatial gradient for this particular image strongly affect the estimates.

Finally, we give an example in which the data is noisy and in which the velocity $v(x)$ is *non-constant.* In Figure 4.5.64 we have a noisy version of a ramp image moving with velocity $v(i,t) = .2i$. This corresponds to the case of linear or shear velocity. We show in Figure 4.5.65 the result of estimating $v$ based on our tree smoother and the result based on using the standard regularization.

In summary our examples have shown that our tree smoother yields optical flow estimates which are comparable or *better* in quality to the result of using standard regularization. Our smoother has the added advantage of being extremely efficient and highly parallelizable. Thus, our tree models seem promising as a way of regularizing the optical flow problem, although examples in 2D need to be looked into thoroughly. Also, the inherent problems of this gradient-based formulation as touched upon briefly in this section deserve further investigation.

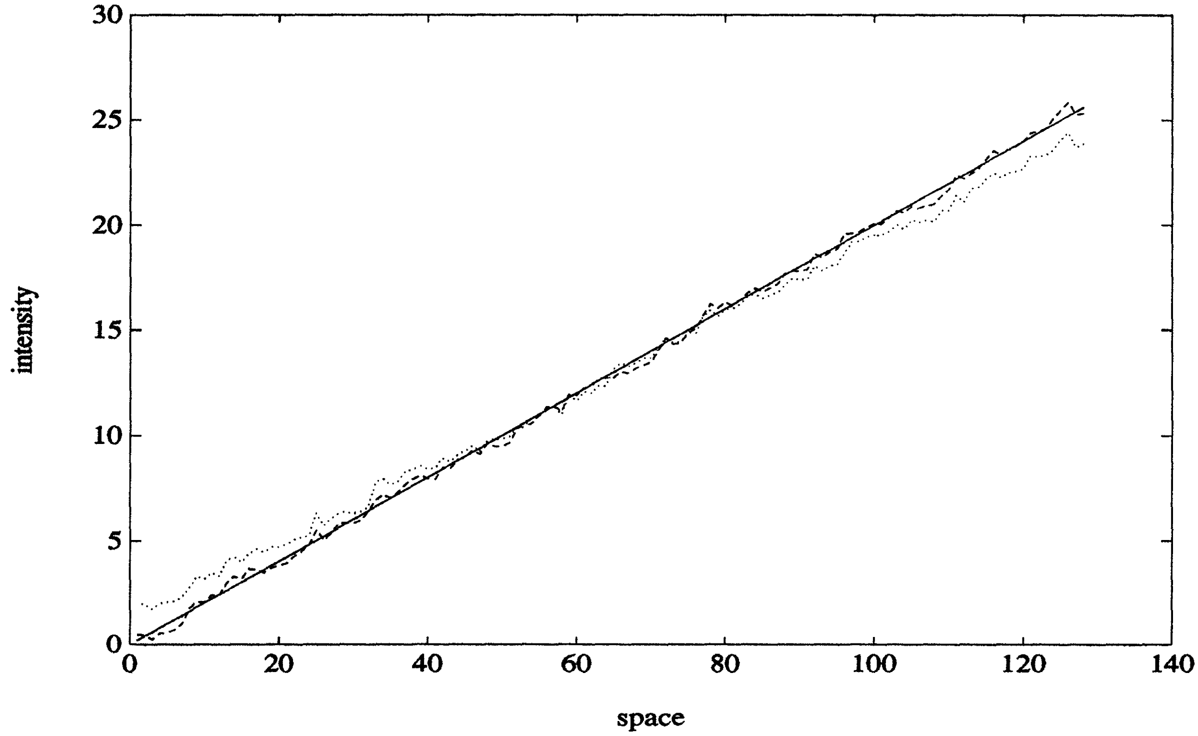

Figure 4.5.65: True $v$ (solid) $\hat{v}$ Using Standard Regularization for $\Gamma = 1$ (dashed) $\hat{v}$ Using Tree Smoother for $\Gamma = 1$ (dotted)

# Chapter 5

# Conclusion

In this chapter we summarize the major contributions of this thesis and suggest directions for future research. This thesis has been motivated by the need for developing a stochastic modeling and filtering theory to complement the already existing and quickly growing deterministic theory of multiresolution signal representations, motivated primarily by the wavelet transform. Motivated by the structure of the wavelet transform, we have introduced classes of stochastic state models defined on lattices which are Markov in scale. These models describe a rich class of phenomena, including for example a class of $1/f$ processes, as well as other processes. We have developed several algorithms for smoothing our processes based on data at multiple scales, where we've used the notion of recursion in scale to develop algorithms which are both extremely efficient and highly parallelizable. Note that our algorithms allow for the fusion of multiscale data with no added complexity. Also, recursions in scale can be performed equally well in 2D, resulting in extremely efficient algorithms for processing 2D data. The numerical examples presented in this thesis illustrate the potential of our approach in modeling processes which may or may not possess multiscale features and in fusing multiscale data efficiently. The theory developed in this thesis provides a formalism in which to investigate multiscale signal processing problems and, more importantly, we hope it provides a framework in which to understand more deeply what makes a given problem appropriate for a multiscale approach.

## 5.1 Thesis Contributions

- We have introduced a class of stochastic processes described by state models on trees and lattices motivated by the wavelet transform. These processes are defined on lattices where each level of the lattice has the interpretation of scale and where our processes defined on these lattices are Markov in scale. A specific case of our processes corresponds to the model of Wornell[55] for modeling $1/f$ processes. Our models are also successful in describing other processes as demonstrated in the numerical results in this thesis on smoothing Gauss-Markov processes. We have characterized the eigenvectors of one class of our processes (in which parameters may vary in scale but not in translation) in terms of the wavelet operators $H_i$ and $G_i$ and have developed an algorithm for smoothing our processes based on the wavelet transform. In particular the algorithm uses the wavelet transform to transform the problem into a set of independent smoothing problems each of which is solved recursively in scale. The algorithm is extremely efficient and allows for the case of smoothing multiple scale data as well as for the case of vector processes. We have provided two versions of the algorithm, one which consists of a coarse-to-fine sweep followed by a fine-to-coarse sweep and another which starts starts with a fine-to-coarse sweep and is followed by a coarse-to-fine sweep.

- We have provided a general framework for modeling multiscale processes defined over a finite-length interval by characterizing the problem of adapting the wavelet transform to finite length signals. In particular we have characterized the effect of the finite length assumption on the operators $H_i$ and $G_i$ and have provided several ways of adapting these operators to the case of finite length signals while preserving their fundamental algebraic structure. In our framework for example the commonly used method of using cyclic convolutions in place of linear convolutions in the operators $H_i$ and $G_i$ is simply one particular choice of adaptation of these operators. Models based on these operators assume that the signal is periodic in the length of the interval. We have also characterized ways of adapting the these operators that lead to models which do not assume the signal is periodic but rather the result of some form of windowing.

- We have studied in detail our class of lattice processes for the case where the operator $H_i$ corresponds to the Haar wavelet. In this case we have the class of state models on dyadic trees. We have developed a smoothing algorithm for these processes which can be viewed as a generalization of the Rauch-Tung-Striebel algorithm to trees. This algorithm is extremely efficient, highly parallelizable, and easily extendible to 2D. As in the case of our transform approach the algorithm allows for the smoothing of vector processes based on data at multiple scales. Moreover, unlike in the case of our transform algorithm, our tree algorithm can be applied to the case where the parameters of the model may vary *arbitrarily*, i.e. in both scale and translation. An example of this is the measurement scenario in which the data is missing at arbitrary points in the interval.

- We have analyzed the filtering step of our algorithm and in particular we've analyzed in detail the Riccati equations associated with the filter. This lead us to decomposing the filter into an ML filter plus a filter which propagates the prior information. We have derived the ML filter in several ways including one which involves the Hamiltonian formulation of the smoother. We have defined notions of reachability and observability on trees which allow us to provide bounds on the error covariances of the filter. We then defined a notion of $l_p$ stability for processes propagating upwards along the tree. Using Lyapunov methods we showed that our filter is $l_2$-stable under conditions of reachability and observability. We also showed the existence of a steady-state filter which is $l_2$-stable.

- Finally, we have given numerical examples demonstrating the effectiveness of our multiscale approach in smoothing processes. We've shown the richness of our models in approximating processes by giving examples of using our models to smooth both $1/f$ and Gauss-Markov processes. We've compared the performance of our smoothers with the performance of the optimal smoother (the one which is based on the true model) on these processes. Our results show, both qualitatively and quantitatively, that our smoothers do quite well for a wide range of SNR's. We've also given examples of fusing coarse scale data with fine scale data. We've shown that the resolution of the estimates produced by our smoothers at a particular point reflects the resolution of the available data at that point; i.e. the fusion of the data is optimal in both scale and translation. For example in the case where a fine scale data set of poor quality is fused with a coarse set of high quality, both of full coverage, the estimate resulting from their fusion accurately captures features of the original signal at the scale of the coarse set. Also, in the case where sparse fine scale data is supplemented with full-coverage coarse data, the optimal estimate at locations where there is fine data is at that fine scale whereas the estimate at locations where there is coarse data is at that coarse scale. Finally, we give an example in which we use our multiscale framework in the context of estimating optical flow. We show how our multiscale models can be used as a prior model for the optical flow, allowing us the use of our fast multiscale algorithms for computing the flow.

## 5.2 Directions for Future Research

- While the theory we've developed in this thesis is for general vector processes, our numerical examples focused on scalar models, leaving the full modeling power of our processes largely unexplored. An example of exploiting the modeling power of our vector processes is to create higher-order models in scale in order to capture better interscale correlations. Since the wavelet coefficients of a wide range of processes exhibit these correlations, using our vector processes to capture these correlations is analogous to using state augmentation to account for colored noise in the case of temporal processes. An example of higher-order models in the case of our tree models, is the case where we have a finite set of lag variables where the lag variables are indexed for each node $t$ on a tree by the indices $\gamma^{-k}t$ for a finite number of $k$'s.

- System identification for our multiscale processes needs to be thoroughly investigated. For the case of our tree processes we have developed a filtering theory which should prove useful for developing ideas related to likelihood functions. The possibilities for using likelihood functions to do detection of signal features in scale deserve investigation.

- The system theory we developed in Chapter 3, i.e. notions of reachability, observability, and $l_p$-stability, was for tree processes in which the model parameters were a function of scale only. It seems the notions we developed in this context should be extendible to the general case in which the parameters can vary arbitrarily. Whether this involves simple redefinition of the notions or a more fundamental rethinking of these notions remains an open question.

- The idea of a multigrid-like iterative algorithm for smoothing our multiscale processes was explored briefly in the end of Chapter 3. While the algorithm for the case of our tree models was developed fully, the details of the algorithm for the case of general lattice models still needs further investigation. Computing the weights involved in the local computations is much more difficult in this case as the correlation structure for general lattice models is not as apparent as in the case of tree models. An iterative algorithm for these lattice models would be useful since it would apply to more general classes of models than the ones considered in Chapter 2 for which the transform approach was developed.

- In general serious effort in studying applications in which a multiscale approach might be appropriate would be invaluable. The interaction between applying our approach to applications and using our applications as a guide to adapting our theory would undoubtedly lead to a deeper understanding of multiscale modeling and signal processing. The optical flow formulation at the end of Chapter 4 is an example of such an application. As discussed in our section on optical flow the issue of noise in the images leads to a nonlinear estimation problem which deserves further investigation. Also, the formulation in this section was developed for the case of 1D; applying our framework in 2D needs to be explored.

# Bibliography

[1] B. Anderson and T. Kailath, "Forwards, backwards, and dynamically reversible Markovian models of second-order processes," *IEEE Trans. Circuits and Systems*, CAS-26, no. 11, 1978, pp. 956-965.

[2] J. Arnaud, "Fonctions spheriques et fonctions definies-positives sur l'arbre homogene", C.R. Acad. Sc., Serie A, 1980, pp. 99-101.

[3] J. Arnaud and B. Letac, "La formule de representation spectrale d'un processus gaussien stationnaire sur un arbre homogene," Laboratoire de Stat. et. Prob.-U.A.-CNRS 745, Toulouse.

[4] M. Barnsley, *Fractals Everywhere*, Academic Press, San Diego, 1988.

[5] M. Basseville, A. Benveniste, and A.S. Willsky "Multiscale Autoregressive Processes, Part I: Schur-Levinson Parametrizations", submitted to *IEEE Transactions on ASSP.*

[6] M. Basseville, A. Benveniste, and A.S. Willsky "Multiscale Autoregressive Processes, Part II: Lattice Structures for Whitening and Modeling", submitted to *IEEE Transactions on ASSP.*

[7] M. Basseville, A. Benveniste, A.S. Willsky, and K.C. Chou, "Multiscale Statistical Processing: Stochastic Processes Indexed by Trees," in *Proc. of Int'l Symp. on Math. Theory of Networks and Systems*, Amsterdam, June 1989.

[8] A. Benveniste, R. Nikoukhah, and A.S. Willsky, "Multiscale System Theory", Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, December 1990.

[9] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast Wavelet Transforms and Numerical Algorithms I", to appear in *Comm. Pure and Appl. Math.*

[10] A. Brandt, "Multi-level adaptive solutions to boundary value problems," Math. Comp. Vol. 13,1977, pp.333-390.

[11] W. Briggs, "*A Multigrid Tutorial*, SIAM, Philadelphia, PA, 1987.

[12] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, vol. 31, pp. 482–540, 1983.

[13] P. Cartier, "Harmonic analysis on trees", Proc. Symos. Pure Math., Vol 26, Amer. Math. Soc. Providence, R.I., 1974, pp.419-424.

[14] P. Cartier, "Geometrie et analyse sur les arbres",Seminaire Bourbaki, 24eme annee, Expose no. 407, 1971/72.

[15] K.C. Chou, A.S. Willsky, "A Multi-resolution, Probabilistic Approach to Two-dimensional Inverse Conductivity Problems," Signal Processing, vol. 18, 1989.

[16] K.C. Chou, S. Golden and A.S. Willsky, "Modeling and Estimation of Multiscale Stochastic Processes", *Int'l Conference on Acoustics, Speech, and Signal Processing"*, Toronto, April 1991.

[17] K.C. Chou and A.S. Willsky, ''Multiscale Riccati Equations and a Two-Sweep Algorithm for the Optimal Fusion of Multiresolution Data", *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, HI, December 1990.

[18] K.C. Chou, A.S. Willsky, A. Benveniste, and M. Basseville,"Recursive and Iterative Estimation Algorithms for Multi-Resolution Stochastic Processes," Proc. 28th IEEE Conf. on Dec. and Cont., Tampa, Dec. 1989.

[19] S.C. Clippingdale and R.G. Wilson, "Least Squares Image Estimations on a Multiresolution Pyramid", Proc. of the 1989 Int'l Conf. on Acoustics, Speech, and Signal Proceeding.

[20] R.R. Coifman, Y. Meyer, S. Quake and M.V. Wickehauser, "Signal Processing and Compression with Wave Packets", preprint, April 1990.

[21] I. Daubechies,"Orthonormal bases of compactly supported wavelets", Comm. on Pure and Applied Math. 91, 1988, pp. 909-996.

[22] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Information Theory*, 36, 1990, pp. 961-1005.

[23] I. Daubechies, A. Grossman, and Y. Meyer, "Painless non-orthogonal expansions," J. Math. Phys. 27, 1986, pp. 1271-1283.

[24] J. Deyst, C. Price, "Conditions for Asymptotic Stability of the Discrete, Minimum Variance, Linear, Estimator," *IEEE Trans. on Automatic Control*, vol. 13, pp. 702- 705, Dec. 1968.

[25] P. Flandrin, "On the Spectrum of Fractional Brownian Motions", *IEEE Transactions on Information Theory*, Vol. 35, 1989, pp. 197-199.

[26] J. Goodman and A. Sokal, "Multi-grid Monte Carlo I. conceptual foundations," Preprint, Dept. Physics New York University, New York, Nov. 1988; to be published.

[27] S. Golden, *Identifying Multiscale Statistical Models Using the Wavelet Transform*, S.M. Thesis, M.I.T. Dept. of EECS, May 1991.

[28] A. Grossman and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape", SIAM J. Math. Anal. 15, 1984, pp. 723-736.

[29] W. Hackbusch and U. Trottenberg, Eds., *Multigrid Methods and Applications*, Springer-Verlag, N.Y., N.Y., 1982.

[30] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, v. 17, pp. 185-203, 1981.

[31] T. Kailath, *Linear Systems*, Prentice-Hall, New Jersey, 1980.

[32] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Trans. on Communication Tech.*, vol. Com-15, no.1, February 1967.

[33] M. S. Keshner, "$1/f$ Noise," *Proc. IEEE*, vol. 70, pp. 212-218, March 1982.

[34] M. Kim and A.H. Tewfik, "Fast Multiscale Detection in the Presence of Fractional Brownian Motions", *Proceedings of SPIE Conference on Advanced Algorithms and Architecture for Signal Processing V*, San Diego, CA, July 1990.

[35] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Anal. and Mach. Intel.*, Vol. PAMI-11, July 1989, pp. 674-693.

[36] S.G. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models", *IEEE Transactions on ASSP*, Vol. 37, December 1989, pp. 2091-2110.

[37] B. Mandelbrot, *The Fractal Geometry of Nature,* Freeman, New York, 1982.

[38] B. B. Mandelbrot and H.W. Van Ness, "Fractional Brownian Motions, Fractional Noises and Applications", *SIAM Review*, Vol. 10, October 1968, pp. 422-436

[39] S. McCormick, *Multigrid Methods*, Vol. 3 of the SIAM Frontiers Series, SIAM, Philadelphia, 1987.

[40] Y. Meyer, "L'analyse par ondelettes", Pour la Science, Sept. 1987.

[41] R. Nikoukhahm, A. Willsky, and B. Levy, "Kalman Filtering and Riccati Equations for Descriptor Systems", submitted to *IEEE Trans. on Automatic Control.*

[42] D. Paddon and H. Holstein, Eds. *Multigrid Methods for Integral and Differential Equations*, Clarendon Press, Oxford, England, 1985.

[43] A. P. Pentland, "Fast Surface Estimation Using Wavelet Bases", MIT Media Lab Vision and Modeling Group TR-142, June 1990.

[44] A. P. Pentland, "Fractal-Based Description of Natural Scenes", *IEEE Transactions on Patt. Anal. and Mach. Intel.,* Vol. PAMI-6, November 1989, 661-674.

[45] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum Likelihood Estimates of Linear Dynamic Systems," AIAA Journal, Vol. 3, No. 8, Aug. 1965, pp. 1445-1450.

[46] A. Rougee, B. Levy, and A. Willsky, "An estimation-based approach to the reconstruction of optical flow," Laboratory for Information and Decision Systems Technical Report, no. LIDS-P-1663, MIT, April 1987. 1445-1450.

[47] M.J. Smith and T.P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders", IEEE Trans. on ASSP 34, 1986, pp. 434-441.

[48] G. Strang, "Wavelets and Dilation Equations: A Brief Introduction", *SIAM Review*, Vol. 31, No. 4, December 1989, pp. 614-627.

[49] R. Szeliski, "Fast Surface Interpolation Using Hierarchical Basis Function", *IEEE Transactions on PAMI*, Vol. 12, No. 6, June 1990, pp. 513-528.

[50] D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Methods", *IEEE Transaction on PAMI*, Vol. PAMI-8, No. 2, March 1986, pp. 129-139.

[51] A.H. Tewfik and M. Kim, "Correlation Structure of the Discrete Wavelet Coefficients of Fractional Brownian Motions", submitted to *IEEE Transactions on Information Theory.*

[52] M. Todd and R. Wilson, "An Anisotropic Multi-Resolution Image Data Compression Algorithm", Proc. of the 1989 Int'l Conf. on Acoustics, Speech, and Signal Processing.

[53] T. Verghese and T. Kailath, "A further note on backward Markovian models," *IEEE Trans. on Information Theory,* IT–25, pp. 121–124, 1979.

[54] M. Vetterli, and C. Herley, "Wavelet and Filter Banks: Relationships and New Results", *Proceedings of the ICASSP*, Albuquerque, NM, 1990.

[55] G.W. Wornell, "A Karhunen-Loeve-Like Expansion for 1/f Processes via Wavelets", *IEEE Transactions on Information Theory*, Vol. 36, No. 9, July 1990, pp. 859-861.

[56] G.W. Wornell and A.V. Oppenheim, "Estimation of Fractal Signals from Noisy Measurements Using Wavelets", submitted to *IEEE Transactions on ASSP.*

[57] A.S. Willsky, K.C. Chou, A. Benveniste, and M. Basseville,"Wavelet Transforms, Multiresolution Dynamical Models, and Multigrid Estimation Algorithms", *1990 IFAC World Congress*, Tallinn, USSR, August 1990.

[58] A. Witkin, D. Terzopoulos and M. Kass, "Signal Matching Through Scale Space", *Int. J. Comp. Vision*, Vol 1, 1987, pp. 133-144.