Presented at the 1989 IEEE Conference on Decision and Control

September 1989 LIDS-P-1911

RECURSIVE AND ITERATIVE ESTIMATION ALGORITHMS FOR MULTI-RESOLUTION STOCHASTIC PROCESSES

K. C. Chou, A. S. Willsky
Dept. of Electrical Engineering and Computer Science
Lab. for Information and Decision Systems, MIT
Cambridge, MA 02139

A. Benveniste, M. Basseville
IRISA
Campus Universitaire de Beaulieu
35402 Rennes Cedex, France

ABSTRACT

A current topic of great interest is the multi-resolution analysis of signals and the development of multi-scale algorithms. In this paper we describe part of a research effort aimed at developing a corresponding theory for stochastic processes described at multiple scales and for their efficient estimation or reconstruction given partial and/or noisy measurements which may also be at several scales. The theories of multiscale signal representations and wavelet transforms lead naturally to models of signals(in one or several dimensions) on trees and lattices. In this paper we focus on one particular class of processes defined on dyadic trees. The central results of the paper are three algorithms for optimal estimation/reconstruction for such processes: one reminiscent of the Laplacian pyramid and the efficient use of Haar transforms, a second that is iterative in nature and can be viewed as a multigrid relaxation algorithm, and a third that represents an extension of the Rauch-Tung-Striebel algorithm to processes on dyadic trees and involves a new discrete Riccati equation, which in this case has *three* steps: predict, *merge*, and measurement update. Related work and extensions are also briefly discussed.

1 INTRODUCTION

The investigation of multi-scale representations of signals and the development of multi-scale algorithms has been and remains a topic of much interest in many applications.

One of the more recent areas of investigation has been the development of a theory of multi-scale representations of signals and the closely related topic of wavelet transforms [7]. These methods have drawn considerable attention and examples that have been given of such transforms seem to indicate that it should be possible to develop effective optimal processing algorithms based on these representations. The development of optimal processing algorithms—e.g. for the reconstruction of noise-degraded signals or for the detection and localization of transient signals of different durations—requires, of course, the development of a corresponding theory of stochastic processes and their estimation. The research presented in this and several other papers and reports has the development of this theory as its objective.

2 MULTISCALE REPRESENTATIONS AND STOCHASTIC PROCESSES ON TREES

2.1 Multiscale, Wavelets and Trees

As developed in [7], the multi-scale representation of a continuous-time signal $x(t)$ consists of a sequence of approximations specified in terms of a single function $\phi(t)$, where the approximation at the mth scale is given by

$$x_m(t) = \sum_{n=-\infty}^{+\infty} x(m,n)\phi(2^m t - n) \tag{2.1}$$

The function ϕ is far from arbitrary. In particular $\phi(t)$ must be orthogonal to its integer translates $\phi(t-n)$, and, in order for the $(m+1)$st approximation to be a *refinement* of the mth, we require that

$$\phi(t) = \sum_n h(n)\phi(2t-n) \tag{2.2}$$

As developed in [7], the sequence $h(n)$ must satisfy several conditions for the desired properties of $\phi(t)$ to hold and for $x_m(t)$ to converge to $x(t)$ as $m \to \infty$. The simplest example of such a ϕ, h pair is the Haar approximation in which $\phi(t) = 1$ for $t \in [0,1)$ and 0 otherwise, corresponding to $h(n) = \delta(n) + \delta(n-1)$, where $\delta(n)$ is the usual discrete impulse. As shown in [7] there is a family of FIR $h(n)$'s and corresponding compactly supported $\phi(t)$'s, where the smoothness of $\phi(t)$ increases with the length of $h(n)$.

The closely related *wavelet transform*, is based on a single function $\psi(t)$ that has the property that the full set of its scaled translates $\{2^{m/2}\psi(2^m t - n)\}$ forms a complete orthonormal basis for L^2. In [7] it is shown that if ϕ and ψ are related via

$$\psi(t) = \sum_n g(n)\phi(2t-n) \tag{2.3}$$

where $g(n)$ and $h(n)$ must form a *conjugate mirror filter* pair, then

$$x_{m+1}(t) = x_m(t) + \sum_n d(m,n)\psi(2^m t - n) \tag{2.4}$$

and indeed $x_m(t)$ is simply the partial orthonormal expansion of $x(t)$, up to scale m, with respect to the basis defined by

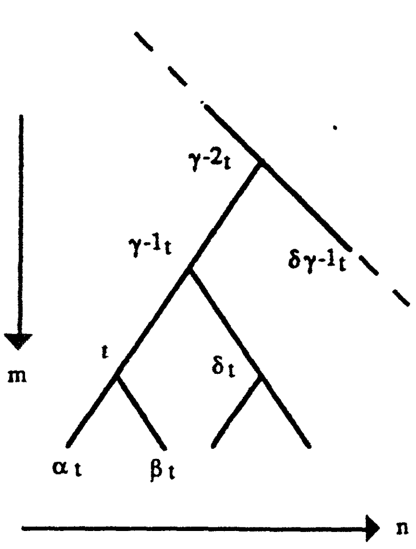

Figure 1: Dyadic Tree Representation

ψ. For example for the Haar approximation $g(n) = \delta(n) - \delta(n-1)$ and $\{2^{m/2}\psi(2^m t - n)\}$ is the *Haar basis.*

Using eqs. (2.1)–(2.4) we see that we have a ***dynamical*** relationship between the coefficients $x(m,n)$ at one scale and those at the next, defining a lattice on the points (m,n), where $(m+1,k)$ is connected to (m,n) if $x(m,n)$ influences $x(m+1,k)$. For example the Haar representation defines a dyadic tree structure on the points (m,n) in which each point has two descendents corresponding to the two subdivisions of the support interval of $\phi(2^m t - n)$.

The preceding development motivates the study of stochastic processes $x(m,n)$ defined on lattices. In our work to date we have focused attention on the case of the dyadic tree. As illustrated in Figure 1, with this and any of the other lattices, the scale index m is time-like and defines a natural direction of recursion for our representation. With increasing m denoting the forward direction, we then can define a unique backward shift γ^{-1} and two forward shifts α and β. Also, for notational convenience we denote each node of the tree by a single index t and let T denote the set of all nodes. Thus if $t = (m,n)$ then $\alpha t = (m+1, 2n)$, $\beta t = (m+1, 2n+1)$, and $\gamma^{-1}t = (m-1, [\frac{n}{2}])$ where $[x] =$ integer part of x. Also we use $m(t)$ to denote the scale (i.e. the m-component) of t. Finally, while we have described multi-scale representations for continuous-time signals on $(-\infty, \infty)$, they can also be used for signals in several dimensions on compact intervals, or in discrete-time. For example a signal defined for $t = 0, 1, \ldots, 2^{M-1}$ can be represented by M scales, and in this case the tree of Figure 1 has a bottom level, representing the samples of the signal itself, and a single root node, denoted by 0, at the top.

2.2 Dynamic Stochastic Models on Trees

The state model we consider evolves from coarse-to-fine scales on the dyadic tree:

$$x(t) = A(m(t))x(\gamma^{-1}t) + B(m(t))w(t) \qquad (2.5)$$

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. (2.5) 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 $A(m)$ is invertible for all m, this is equivalent to requiring $w(t)$ to be independent of *some* $x(\tau)$ with $\tau \neq t$, $m(\tau) < m(t)$. Note that 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. Also, while the analysis we perform is easily extended to the case in which A and B are arbitrary functions of t, we focus here on the case in which these quantities do depend only on scale. This leads to significant computational efficiencies and also, when this dependence is chosen appropriately, these models possess self-similar properties from scale to scale.

The covariance of $x(t)$ evolves according to a Lyapunov equation on the tree:

$$P_x(t) = A(m(t))P_x(\gamma^{-1}t)A^T(m(t)) + B(m(t))B^T(m(t)) \qquad (2.6)$$

Note 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 assume that this is the case and therefore 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, then $P_x(t) = P_x(m(t))$ for *all* t. Let $K_{xx}(t,s) = E[x(t)x^T(s)]$ and let $s \wedge t$ denote the least upper bound of s and t. Then

$$K_{xx}(t,s) = \Phi(m(t), m(s \wedge t))P_x(m(s \wedge t))\Phi^T(m(s), m(s \wedge t)) \qquad (2.7)$$

where $\Phi(m_1, m_2)$ is the state transition matrix associated with $A(m)$.

Consider the case when A and B are constant, A is stable, and let P_x be the solution to the algebraic Lyapunov equation

$$P_x = AP_xA^T + BB^T \qquad (2.8)$$

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

$$K_{xx}(t,s) = A^{d(t,s\wedge t)}P_x(A^T)^{d(s,s\wedge t)} \qquad (2.9)$$

Note that $d(s,t) = d(s, s\wedge t) + d(t, s\wedge t)$ and if the condition $AP_x = P_xA^T$ (which also arises in the study of reversible processes [1] and obviously holds in the scalar case) is satisfied, then $x(t)$ is an *isotropic process,* i.e. $K_{xx}(s,t)$ depends only on $d(s,t)$. We will comment on our analysis [2] of such processes in Section 4.

3 OPTIMAL ESTIMATION ON TREES

In this section we consider the estimation of $x(t), t \in T$ given the measurements are of the form

$$y(t) = C(m(t))x(t) + v(t) \qquad (3.1)$$

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 $R(m(t))$. For simplicity we assume that there is a root node 0 and M scales on which we have data and wish to focus. This model allows us to consider multiple resolution measurements of our process and includes the single resolution problem, i.e. when $C(m) = 0$ unless $m = M$. In the following three subsections we describe three different algorithmic structures for estimation problems of this type.

3.1 Noisy Interpolation and the Laplacian Pyramid

Consider the model eq. (2.5) with a single scale of measurements:

$$y(n) \;=\; Cx(M,n) + v(n) \quad n = 0, 1, \ldots 2^M - 1 \qquad (3.2)$$

where without loss of generality we assume that the covariance of $v(n)$ is I. We look first at the batch estimation of x at this finest scale, using the notation

$$\begin{aligned} Y^T &= [y^T(0), \ldots, y^T(2^M - 1)] && (3.3)\\ X_M^T &= [x^T(M,0), \ldots, x^T(M, 2^M - 1)] &&\\ V^T &= [v^T(0), \ldots, v^T(2^M - 1)] &&\\ \mathcal{C} &= diag(C, \ldots, C) && (3.4)\\ \mathcal{P}_M &= E[X_M X_M^T] && (3.5) \end{aligned}$$

The optimal estimate is given by

$$\hat{X}_M = \mathcal{P}_M \mathcal{C}^T [\mathcal{C}\mathcal{P}_M \mathcal{C}^T + I]^{-1} Y \qquad (3.6)$$

Consider next the interpolation up to higher levels in the tree. Letting X_k denote the vector of values of x_t at the kth scale, with $\mathcal{P}_k$ as its covariance, we find that

$$\begin{aligned} \hat{X}_k &= \tilde{\mathcal{P}}_{k,k+1} \mathcal{P}_{k+1}^{-1} \hat{X}_{k+1} && (3.7)\\ \tilde{\mathcal{P}}_{k,k+1} &= \mathcal{P}_k \mathcal{A}_{k+1}^T && (3.8) \end{aligned}$$

$$\mathcal{A}_{k+1} \;=\; \begin{bmatrix} A(k+1) & 0 & 0 & \cdots & 0 \\ A(k+1) & 0 & 0 & \cdots & 0 \\ 0 & A(k+1) & 0 & \cdots & 0 \\ 0 & A(k+1) & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A(k+1) \\ 0 & 0 & 0 & \cdots & A(k+1) \end{bmatrix} \qquad (3.9)$$

The computation of these coarse-scale estimates is of importance if one wishes to consider efficient coding of data possessing multiple-scale descriptions. Indeed the algorithm, eq. (3.6) and eq. (3.7), possesses structure reminiscent of the Laplacian pyramid approach [5] to multiscale coding. In particular let us examine eq. (3.7) component by component. Then from the structure of the matrices and the tree, we can deduce [6] that the contribution of $\hat{x}(t)$ with $m(t) = k + 1$ to $\hat{x}(s)$ with $m(s) = k$ depends only on $d(s, s \wedge t)$. Thus, eq. (3.7) has the following form for each node s with $m(s) = k$.

$$\begin{aligned} \hat{x}(s) &= \sum_{i=0}^{k} H(k,i) \sum_{t \in \Theta_x(k,i)} \hat{x}(t) && (3.1C)\\ \Theta_x(k,i) &= \{t' | m(t') = k+1, d(s, s \wedge t') = i\} && (3.11) \end{aligned}$$

This computation from level to level, as we successively decimate our estimated signal and in which processing from scale to scale involves averaging of values bears some resemblance to the Laplacian pyramid, although in this case the weighting function $H(k,i)$ is of full extent and in general varies from scale to scale.

Another efficient algorithm for the recursion eq. (3.7) comes from the fact that the *discrete Haar transform* block diagonalizes both $\mathcal{P}_k$ and $\tilde{\mathcal{P}}_{k,k+1}$. For simplicity let us first describe this for the case in which x and y are scalar processes.

Definition 3.1 *The discrete Haar basis is an orthonormal basis for $\mathcal{R}^N$ where $N = 2^k$. The matrix V_k whose columns form this basis consists of vectors representing "dilated, translated, and scaled" versions of the vector $[1, -1]^T$. For example for $k = 3$,*

$$V_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{2} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{2} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \end{bmatrix} \qquad (3.12)$$

The following are proven in [6].

Lemma 3.1 *Consider the case in which $x(t)$ is a scalar process. The discrete Haar matrix V_k provides a complete orthonormal set of eigenvectors for the matrix $\mathcal{P}_k$; i.e.*

$$\mathcal{P}_k = V_k \Lambda_k V_k^T \qquad (3.13)$$

where Λ_k is a diagonal matrix.

Lemma 3.2 *Given $\tilde{\mathcal{P}}_{k,k+1}$ and V_{k+1},*

$$\tilde{\mathcal{P}}_{k,k+1} V_{k+1} = \left[0 \mid V_k \tilde{\Lambda}_k\right] \qquad (3.14)$$

where $\tilde{\Lambda}_k$ is a diagonal matrix of dimension 2^k.

These results are easily extended to the case of *vector* processes $x(t)$. In this case we must consider the block version of the discrete Haar matrix, defined as in Definition 3.1 except we now consider "dilated, translated, and scaled" versions of the block matrix $[I \; - I]^T$ instead of the vector $[1, -1]^T$, where each block is of size equal to the dimension of x. It is important to note that the discrete Haar transform, i.e. the computation of $V_k z$ can be performed in an extremely efficient manner (in the block case as well), by successive additions and subtractions of pairs of elements.

Returning to eq. (3.7) we see that we can obtain an extremely efficient transform version of the recursion. Specifically, we have that

$$\hat{z}_k = V_k^T \hat{X}_k = [0 \mid \tilde{\Lambda}_k] \Lambda_{k+1}^{-1} \hat{z}_{k+1} \qquad (3.15)$$

Thus, we see that the fine scale components of $\hat{X}_k$ are unneeded at the coarser scale; i.e. only the lower half of $\hat{z}_{k+1}$, which depends only on pairwise sums of the elements of $\hat{X}_k$, enters in the computation. So, if we let

$$\begin{aligned} \Lambda_{k+1} &= diag(M_{k+1}, D_{k+1}) && (3.16) \\ \hat{z}_{k+1} &= \begin{bmatrix} \hat{z}^1_{k+1} \\ \hat{z}^2_{k+1} \end{bmatrix} && (3.17) \end{aligned}$$

where M_{k+1} and D_{k+1} each have $2^k \times 2^k$ blocks, we see that

$$\hat{z}_k = \tilde{\Lambda}_k D^{-1}_{k+1} \hat{z}^2_{k+1} \qquad (3.18)$$

Finally, while we have focused on the structure of eq. (3.7), analogous algorithmic structures exist for the initial data incorporation step eq. (3.6). Thus, once we perform a single Haar transform on the original data Y, we can compute the transformed optimal estimates $\hat{z}_M, \hat{z}_{M-1}, \ldots$ in a block-diagonalized manner as in eq. (3.18), where the work required to compute eq. (3.18) is only $O(2^k \times$ dim. of state). Also, it is possible to consider multi-scale measurements in this context, resulting in smoothing algorithms in the transform domain [6].

3.2 A Multigrid Relaxation Algorithm

In this section we define an iterative algorithm for the estimation of x given measurements at all scales. This algorithm is reminiscent of relaxation methods for multigrid partial differential equations [3,4], and, as in that context may have significant computational advantages even if only the finest level estimates are actually desired and if only fine level measurements are available. Let Y denote the full set of measurements at all scales. Then, thanks to Markovianity we have the following: For $m(t) = M$, the finest scale

$$\begin{aligned} E[x(t)|Y] &= E\left\{E[x(t)|x(\gamma^{-1}t), Y]|Y\right\} \\ &= E\left\{E[x(t)|x(\gamma^{-1}t), y(t)]|Y\right\} && (3.19) \end{aligned}$$

For $m(t) < M$

$$\begin{aligned} E[x(t)|Y] &= E\left\{E[x(t)|x(\gamma^{-1}t), x(\alpha t), x(\beta t), Y]|Y\right\} \\ &= E\left\{E[x(t)|x(\gamma^{-1}t), x(\alpha t), x(\beta t), y(t)]|Y\right\} \\ & && (3.20) \end{aligned}$$

The key now is to compute the inner expectations in eq. (3.19) and eq. (3.20). This can be done with the aid of eq. (2.5) and the reverse-time version of eq. (2.5), which assuming that $A(m)$ is invertible for all m is given by [8]:

$$\begin{aligned} x(\gamma^{-1}t) &= F(m(t))x(t) - A^{-1}(m(t))B(m(t))\tilde{w}(t) && (3.21) \\ F(m(t)) &= A^{-1}(m(t))[I - B(m(t))B^T(m(t))P_x^{-1}(m(t))] \\ & && (3.22) \end{aligned}$$

and where $\tilde{w}(t)$ is a white noise process with covariance

$$\tilde{Q}(m(t)) \triangleq I - B^T(m(t))P_x^{-1}(m(t))B(m(t)) \qquad (3.23)$$

We can then show [6] that for $m(t) = M$

$$\begin{aligned} \hat{x}(t) &= (\mathcal{P}')^{-1}\left\{C^T(m(t))R^{-1}(m(t))y(t)\right. \\ &+ \left. F^T(m(t))R_1^{-1}(m(t))\hat{x}(\gamma^{-1}t)\right\} && (3.24) \end{aligned}$$

while for $m(t) < M$

$$\hat{x}(t) = \mathcal{P}^{-1}\left\{K_1 y(t) + K_2\hat{x}(\gamma^{-1}t) + K_3\hat{x}(\alpha t) + K_4\hat{x}(\beta t)\right\} \qquad (3.25)$$

where

$$\begin{aligned} K_1 &= C^T(m(t))R^{-1}(m(t)) && (3.26) \\ K_2 &= F^T(m(t))R_1^{-1}(m(t)) && (3.27) \\ K_3 &= A^T(m(\alpha t))R_2^{-1}(m(\alpha t)) && (3.28) \\ K_4 &= A^T(m(\beta t))R_2^{-1}(m(\alpha t)) && (3.29) \\ \mathcal{P} &= P_x^{-1}(t) + K_1 C(m(t)) + K_2 F(m(t)) \\ &+ K_3 A(m(\alpha t)) + K_4 A(m(\beta t)) && (3.30) \\ \mathcal{P}' &= P_x^{-1}(t) + C^T(m(t))R^{-1}(m(t))C(m(t)) \\ &+ F^T(m(t))R_1^{-1}(m(t))F(m(t)) && (3.31) \end{aligned}$$

Thus, eq. (3.24) and eq. (3.25) are an implicit set of equations for $\{\hat{x}(t)|t \in T\}$, where the equation at each point 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, and the scale-to-scale sweeps can then be performed consecutively moving up and down the tree. The fact that the computations can now be counted in terms of scales rather that in terms of individual points reduces the size of the problem from $O(2^{M+1})$, which is the number of nodes on the tree, to $O(M)$. The following is one possible algorithm.

Algorithm 3.1 *Multigrid Relaxation Algorithm:*

```
1. Initialize $\hat{X}_0, \ldots, \hat{X}_M$ to 0.
2. Do Until Desired Convergence is Attained:
    (a) Compute in parallel eq. (3.24) for each entry of $\hat{X}_M$
    (b) For $k = M - 1$ to 0
            Compute in parallel eq. (3.25) for each entry of $\hat{X}_k$
    (c) For $k = 1$ to $M - 1$
            Compute in parallel eq. (3.25) for each entry of $\hat{X}_k$
```

Essentially, Algorithm 3.1 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. In multigrid terminology [4] this is a *V-cycle*. It is also possible to describe *W*-cycle [4] iterations.

3.3 Two-Sweep, Rauch-Tung-Striebel Algorithm

In this section we describe a recursive rather than iterative algorithm that generalizes the Rauch-Tung-Striebel (RTS) smoothing algorithm for causal state models. Our algorithm once again involves a pyramidal set of steps and a considerable level of parallelism.

Let us recall the structure of the RTS algorithm. The first step consists of a Kalman filter for computing $\hat{x}(t|t)$, predicting to obtain $\hat{x}(t+1|t)$ and updating with the new measurement $y(t)$. The second step propagates backward combining

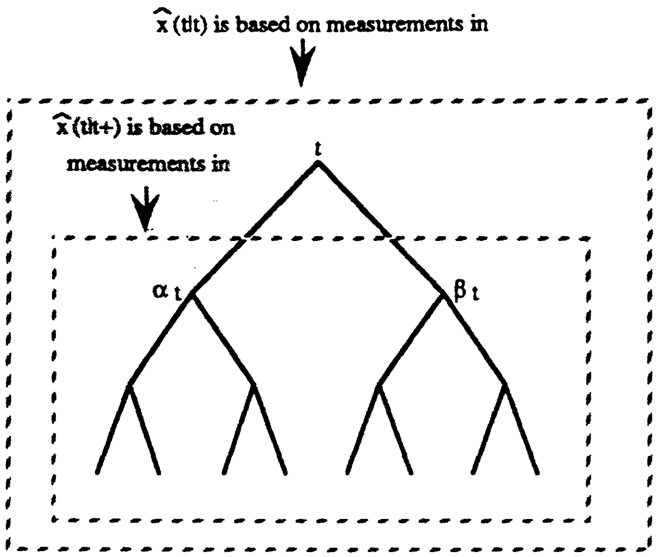
x̂(t|t) is based on measurements in
x̂(t|t+) is based on measurements in
t
αt
βt

Figure 2: Representation of Meaurement Update and Merged Estimates

the smoothed estimate $\hat{x}_s(t+1)$ with the forward estimate $\hat{x}(t|t)$ (or equivalently $\hat{x}(t+1|t)$) to compute $\hat{x}_s(t)$. 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, as it can be thought of as two parallel prediction steps, each as in RTS, followed by a *merge* step that has no counterpart for causal models. One other difference is that the forward sweep of our algorithm *must* be from fine-to-coarse.

To begin, let us define some notation(see Figure 2):

$$\begin{aligned} Y_t &= \{y(s)|s = t \text{ or } s \text{ is a descendent of } t\} \\ &= \{y(s)|s \in (\alpha,\beta)^* t \ , \ m(s) \leq M\} && (3.32) \\ Y_t^+ &= \{y(s)|s \in (\alpha,\beta)^* t \ , \ t < m(s) \leq M\} && (3.33) \\ \hat{x}(\cdot|t) &= E[x(\cdot)|Y_t] && (3.34) \\ \hat{x}(\cdot|t+) &= E[x(\cdot)|Y_t^+] && (3.35) \end{aligned}$$

We now consider the measurement update. Specifically, suppose that we have computed $\hat{x}(t|t+)$ and the corresponding error covariance, $P(m(t)|m(t)+)$, which depends only on scale. Then, standard estimation results yield

$$\begin{aligned} \hat{x}(t|t) &= \hat{x}(t|t+) + K(m(t))[y(t) - C(m(t))\hat{x}(t|t+)] && (3.36) \\ K(m(t)) &= P(m(t)|m(t)+)C^T(m(t))V^{-1}(m(t)) && (3.37) \\ V(m(t)) &= C(m(t))P(m(t)|m(t)+)C^T(m(t)) + R(m(t)) && (3.38) \end{aligned}$$

and the resulting error covariance is given by

$$P(m(t)|m(t)) = [I - K(m(t))C(m(t))]P(m(t)|m(t)+) \qquad (3.39)$$

This computation begins on the Mth level with $\hat{x}(t|t+) = 0$, $P(M|M+) = P_x(M)$.

Suppose 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 and have equal error covariances, denoted by $P(m(t)+1|m(t)+1)$. We then compute $\hat{x}(t|\alpha t)$ and $\hat{x}(t|\beta t)$ from

$$\begin{aligned} \hat{x}(t|\alpha t) &= F(m(t)+1)\hat{x}(\alpha t|\alpha t) && (3.40) \\ \hat{x}(t|\beta t) &= F(m(t)+1)\hat{x}(\beta t|\beta t) && (3.41) \end{aligned}$$

with corresponding identical error covariances $P(m(t)|m(t)+1)$ given by(for notational convenience we omit the explicit dependence of m on t)

$$\begin{aligned} P(m|m+1) &= \Gamma(m+1) + Q(m+1) && (3.42) \\ \Gamma(m+1) &= F(m+1)P(m+1|m+1)F^T(m+1) && (3.43) \\ Q(m+1) &= \Omega(m+1)\tilde{Q}(m+1)\Omega^T(m+1) && (3.44) \\ \Omega(m+1) &= A^{-1}(m+1)B(m+1) && (3.45) \end{aligned}$$

We now must merge these estimates to form $\hat{x}(t|t+)$. As shown in [6], this merge step, which has no counterpart in standard Kalman filtering is given by

$$\begin{aligned} \hat{x}(t|t+) &= P(m|m+)P^{-1}(m|m+1)[\hat{x}(t|\alpha t) + \hat{x}(t|\beta t)] && (3.46) \\ P(m|m+) &= [2P^{-1}(m|m+1) - P_x^{-1}(t)]^{-1} && (3.47) \end{aligned}$$

The interpretation of these equations is that $\hat{x}(t|\alpha t)$ and $\hat{x}(t|\beta t)$ are estimates based *almost* completely on independent information sources. However they both use the *a priori* statistics of $x(t)$ and thus this double use of prior information must be accounted for.

Finally, we must describe the downward sweep of the RTS algorithm, combining the smoothed estimate at a parent node $\hat{x}_s(\gamma^{-1}t)$ with the estimates produced during the upward sweep to produce $\hat{x}_s(t)$. Although the derivation is a bit more subtle in the tree case [6], we obtain an identical recursion to that for causal RTS smoothing:

$$\begin{aligned} \hat{x}_s(t) &= \hat{x}(t|t) + G(m(t))\left[\hat{x}_s(\gamma^{-1}t) - \hat{x}(\gamma^{-1}t|t)\right] && (3.48) \\ G(m) &= P(m|m)F^T(m)P^{-1}(m-1|m) && (3.49) \end{aligned}$$

and the smoothing error variance is given by

$$P_s(m) = P(m|m) + G(m)\left[P_s(m-1) - P(m-1|m)\right]G^T(m) \qquad (3.50)$$

4 DISCUSSION

In this paper we have introduced a class of stochastic processes defined on dyadic trees and have described several estimation algorithms for these processes. The consideration of these processes and problems has been motivated by a desire to develop multi-scale descriptions of stochastic processes and in particular by the deterministic theory of multi-scale signal representations and wavelet transforms.

In addition to open questions directly related to these models there are a number of related research problems under consideration. One of these [2] involves the modeling of scalar isotropic processes on trees. In particular, a natural extension of a classical 1D time series modeling problem is the construction of dynamic models that match a given isotropic correlation function $K_{xx}(k)$ for a specified number of lags $k = 0, 1, \ldots N$. This problem is studied in detail in [2] and in particular an extension of classical AR modeling is developed and with it a corresponding generalization of the Levinson and Schur recursions for AR models as the order N increases. A few comments about this theory are in order. First, the sequence $K_{xx}(k)$ must satisfy an even more strict set of conditions to be a valid correlation function for an isotropic tree process than it does to be the correlation function of a time series. In particular, since the sequence $x(t), x(\gamma^{-1}t), x(\gamma^{-2}t), \ldots$ is a standard time series, we see that $K_{xx}(k)$ must be a positive definite function. Moreover, considering the covariance of the three points $x(\alpha t)$, $x(\beta t)$, $x(\gamma^{-1}t)$, we conclude that :

$$\begin{bmatrix} K_{xx}(0) & K_{xx}(2) & K_{xx}(0) \\ K_{xx}(2) & K_{xx}(0) & K_{xx}(0) \\ K_{xx}(0) & K_{xx}(2) & K_{xx}(0) \end{bmatrix} \geq 0 \qquad (4.1)$$

Such a condition and many others that must be satisfied do not arise in usual time series. In particular an isotropic process $x(t)$ is one whose statistics are invariant to any isometry on the index set T, i.e. any invertible map preserving distance. For time series such isometries are quite limited: translations, $t \mapsto t + n$, and reflections $t \mapsto -t$. For dyadic trees the set of isometries is far richer, placing many more constraints on K_{xx}. Referring to the Levinson algorithm, recall that the validity of $K_{xx}(k)$ as a covariance function manifests itself in a sequence of reflection coefficients that must take values between ±1. For trees the situation is more complex: for n odd $|k_n| < 1$ while for n even $-\frac{1}{2} < k_n < 1$, $k(n)$ being the nth reflection coefficient. Furthermore, since dyadic trees are fundamentally infinite dimensional, the Levinson algorithm involves "forward" (with respect to the scale index m) and "backward" prediction filters of dimension that grows with order, as one must predict a window of values at the boundary of the filter domain. Also, the filters are not strictly causal in m. For example, while the first-order AR model is simply the scalar, constant-parameter version of the model eq. (2.5) considered here, the second order model represents a forward prediction of $x(t)$ based on $x(\gamma^{-1}t)$, $x(\gamma^{-2}t)$ *and* $x(\delta t)$, which is at the same scale as $x(t)$ (refer to Figure 1). The third-order forward model represents the forward prediction of $x(t)$ *and* $x(\delta t)$ based on $x(\gamma^{-1}t)$, $x(\gamma^{-2}t)$, $x(\gamma^{-2}t)$ *and* $x(\delta\gamma^{-1}t)$. We refer the reader to [2] for details.

ACKNOWLEDGEMENT

The work of K.C.C. and A.S.W. was supported in part by the Air Force Office of Scientific Research under Grant AFOSR-88-0032, in part by the National Science Foundation under Grant ECS-8700903, and in part by the US Army Research Office under Contract DAAL03-86-K-0171. In addition some of this research was performed while these authors were visitors at Institut de Recherche en Informatique et Systemes Aleatoires(IRISA), Rennes, France during which time A.S.W. received partial support from Institut National de Recherche en Informatique et en Automatique(INRIA). A.B. is also with INRIA, and M.B. is also with Centre National de la Recherche Scientifique(CNRS). The research of these authors was also supported in part by Grant CNRS GO134.

References

[1] B. Anderson and T. Kailath, "Forwards, backwards, and dynamically reversible Markovian models of second-order processes," *IEEE Trans. Circuits and Systems*, CAS26, no. 11, pp. 956–965, 1978.

[2] 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.

[3] A. Brandt, "Multi-level adaptive solutions to boundary value problems," *Math. Comp.*, vol. 13, pp. 333–390, 1977.

[4] W. Briggs, *A Multigrid Tutorial*, Philadephia: SIAM, 1987.

[5] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, vol. 31, pp. 482–540, 1983.

[6] K.C. Chou, *A Stochastic Modeling Approach to Multiscale Signal Processing*, MIT, Department of Electrical Engineering and Computer Science, Ph.D. Thesis, (in preparation).

[7] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. on Pure and Applied Math.*, vol. 91, pp. 909- 996, 1988.

[8] T. Verghese and T. Kailath, "A further note on backward Markovian models," *IEEE Trans. on Information Theory*, IT–25, pp. 121–124, 1979.