Computationally Efficient Steady-State Multiscale Estimation for 1-D Diffusion Processes

Terrence T. Ho

Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.

Paul W. Fieguth

System Design Engineering, University of Waterloo, Waterloo, Ontario, Canada

Alan S. Willsky

Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.

Abstract

Conventional optimal estimation algorithms for distributed parameter systems have been limited due to their computational complexity. In this paper, we consider an alternative modeling framework recently developed for large-scale static estimation problems and extend this methodology to dynamic estimation. Rather than propagate estimation error statistics in conventional recursive estimation algorithms, we propagate a more compact multiscale model for the errors. In the context of 1-D diffusion, which we use to illustrate the development of our algorithm, the resulting multiscale estimator achieves $\mathcal{O}(N \log N)$ computational complexity with near-optimal performance as compared to the $\mathcal{O}(N^3)$ complexity of the standard Kalman filter.

Key words: multiscale realization, dynamic estimation, distributed parameter systems, diffusion

Running Title: Multiscale Estimation for 1-D Diffusion

[1] This work was supported by ONR under Grant N00014-91-J-1004 and by AFOSR under Grant F49620-98-1-0349. Corresponding author: Terrence Ho, Massachusetts Institute of Technology, Room 35-425, 77 Massachusetts Avenue, Cambridge, MA 02139, U.S.A., 617-253-4874, fax 617-258-8364, hot@mit.edu.

1 Introduction

Estimation for distributed parameter systems governed by partial differential equations (PDEs), such as those found in applications ranging from pollution control (Omatu et al. (1988)) to the modeling of ecological systems and flexible structures (Banks and Kunisch (1989)), has received considerable attention in the estimation and control community in the past. While there have been successful applications of the theory of optimal estimation for such systems, it is also true that there are severe computational barriers that limit the domain in which truly optimal methods can be implemented. Indeed this is certainly the case in the field of remote sensing in which "data assimilation," i.e., the melding of data with dynamic models, represents one of the most significant current-day problems. For example, in problems of atmospheric or oceanographic data assimilation(Fieguth et al. (1995)), the dimensionality of finite-dimensional approximations to the underlying dynamics can range from hundreds of thousands to hundreds of *millions*. Given the need in such applications to produce both estimates and estimation error variances, the computational challenge is substantial. Indeed, conventional linear least squares estimation (LLSE) algorithms, such as Kalman filtering, are completely impractical for solving such large problems both for computational and for storage reasons, given the sheer size of the error covariance matrices involved. A critical aspect of these estimation problems is the requirement that estimation error statistics be computed. This necessity precludes the use of accelerated methods such as multigrid (Briggs (1987)), which do not supply such error statistics, or the FFT, which requires spatially stationary prior models and spatially regular measurement patterns, requirements that cannot be met in many applications including most, if not all, remote sensing problems.

For these reasons, it is clear that there is a need for suboptimal (that is, approximate) estimation algorithms that can deal effectively with the computational challenges. The key to doing this is to find a compact and effective representation for the statistics of the estimation errors, avoiding the storage or computation of large covariance matrices.

Consider time-recursive estimation for spatially distributed phenomena; this procedure can be viewed as an interleaved sequence of (i) temporal prediction and (ii) purely static spatial estimation problems. The standard Kalman filter approach, illustrated in Fig. 1(a), is to explicitly calculate the full covariance and the Kalman gain at each step. Each filter update step corresponds to solving a static estimation problem, namely that of estimating the errors in the one-step predicted estimates $\hat{x}(t|t-1)$ from the measurement innovations at time t. The exact Kalman filter propagation and solution of this problem corresponds to an *explicit* solution of each such static estimation problem by explicitly calculating full covariance and gain matrices, leading to associated

complexity $\mathcal{O}(N^3)$.

An alternative recursive procedure(Chin et al. (1995)) is one in which we propagate a statistical *model* for the one-step predicted estimation errors, as shown in Fig. 1(b). Such models *implicitly* specify the error statistics, although any desired element of the full error covariance can be computed (dashed lines in figure). The implicit nature of the representation leads to an *implicit* description of the optimal estimator; that is, our result is an algorithm rather than an explicit gain matrix, much as the Kalman filter is implicit and the Wiener filter explicit).

Clearly the major issue, then, is to find an implicit representation of the spatial error statistics that can be efficiently predicted and updated, improving on the $\mathcal{O}(N^3)$ complexity of the Kalman filter by orders of magnitude. In (Chin et al. (1995)) an approach was developed to use a Markov random field framework as an implicit representation for the spatial error models. Such models do indeed capture a rich class of spatial phenomena and in particular were demonstrated to lead to near-optimal estimation performance for problems in dynamic computer vision, however the actual solution of the spatial estimation problem for each measurement update using such a model is not nearly as efficient.

Instead, in this paper, we consider the use of an alternative implicit representation, namely the multiscale stochastic modeling and estimation methodology developed in (Basseville et al. (1992); Chou et al. (1994a)). These multiscale models have been demonstrated to yield extremely fast solutions to purely spatial (i.e., temporally static estimation problems), including the modeling of $1/f$ processes (Daniel and Willsky (1997a); Luettgen et al. (1993)), large distributed phenomena for remote sensing in oceanography (Fieguth et al. (1995, 1998); Menemenlis et al. (1997)) and hydrology (Daniel and Willsky (1997b)). For the class of multiscale models considered in this paper, given a multiscale model having a state dimension $d \ll N$, then the complexity to estimate a spatial process with N points is $\mathcal{O}(Nd^3)$, much less than $\mathcal{O}(N^3)$ for standard least-squares.

Taken together, the existence of the multiscale framework (a highly efficient static estimator) and the implicit modeling paradigm of Fig. 1 strongly motivate applying multiscale techniques to estimate *dynamic* or time-recursive systems, most notably systems in which the "state" is of very high dimension. This requires that we find a multiscale model for the estimation errors, and that we derive an algorithm in order to propagate the multiscale model over time:

(1) Why should multiscale models be capable of modeling the estimation errors for distributed parameter systems? A rich literature already exists for

the theory, stochastic realization, and parameter estimation of multiscale models for one-dimensional(Basseville et al. (1992); Chou et al. (1994a, 1994b); Daniel and Willsky (1997b); Fieguth and Willsky (1996); Irving (1998); Luettgen and Willsky (1995)) processes, particularly Markov ones, and two-dimensional(Chin et al. (1995); Fieguth et al. (1995, 1998); Irving et al. (1997); Luettgen et al. (1993); Menemenlis et al. (1997)) systems.

We are interested in *approximating* the statistics of a given field; that is, we intentionally sacrifice a small amount of statistical fidelity in order to obtain multiresolution models that have small state dimension d. For a surprisingly rich class of purely spatial processes, low-dimensional multiresolution models have been constructed that yield near-optimal estimation performance (that is, with statistically insignificant discrepancies).

The basics of multiscale modeling and realization are discussed in Section 2, with a multiscale model specific to the 1-D diffusion context developed in Section 4.

(2) Past work on multiscale models considered the model as *given*, and so the computational effort considered only the actual steps to compute the estimates. In the time-recursive context the estimation error statistics, and consequently the associated multiscale model, can change at each successive time if we are not in temporal steady-state. The particular problem of quickly propagating a multiscale model over time is unexplored and represents a new, significant contribution of this paper.

This second issue, the development of a fast multiscale-model prediction algorithm, is discussed in Section 3, with examples and performance comparisons shown in Section 5.

2 Multiscale Modeling and Realization

In the multiscale estimation framework of (Basseville et al. (1992); Chou et al. (1994a); Irving (1998)), random processes and random fields are modeled on tree structures. The nodes of these trees are organized into a sequence of scales, where the finest-level scale should be thought of as a discretization of the spatial domain of interest. A node s on the tree is connected to a unique parent node, $s\bar{\gamma}$, at the next coarser level, and to several child nodes $s\alpha_i$ $(i = 1, \ldots, q)$, at the next finer level. In general the number of children may vary from node to node. However, for our purposes focusing on the 1-D spatial domain, it is sufficient for us to restrict our attention to uniform $q = 2$ dyadic trees, depicted in Fig. 2.

The multiscale process is a collection of zero-mean random vectors $\boldsymbol{x}(s)$, in-

dexed by nodes s on the tree and specified by a scale-to-scale relationship of the following form

$$\boldsymbol{x}(s) = \boldsymbol{A}(s)\boldsymbol{x}(s\bar{\gamma}) + \boldsymbol{B}(s)\boldsymbol{w}(s), \tag{1}$$

where $\boldsymbol{w}(s)$ is a zero-mean unit-variance white noise process uncorrelated with $\boldsymbol{x}(0)$, the state at the root node of the tree. Measurements can be made at any node:

$$\boldsymbol{y}(s) = \boldsymbol{C}(s)\boldsymbol{x}(s) + \boldsymbol{v}(s), \tag{2}$$

where $\boldsymbol{v}(s)$ is white, zero-mean, and uncorrelated with the process $\boldsymbol{x}(s)$.

From (1), the whiteness of $\boldsymbol{w}(s)$ implies that the state $\boldsymbol{x}(s)$ conditionally decorrelates the $q+1$ subtrees connected to node s. This Markovianity property of the multiscale tree admits efficient scale-recursive smoothing algorithms (Chou et al. (1994a, 1994b)), similar to the Rauch-Tung-Striebel smoothing algorithm (Rauch et al. (1965)). The algorithm, summarized in Appendix A, is exact and has a computational complexity of $\mathcal{O}(k^3 N)$, where k is the state dimension of $\boldsymbol{x}(s)$ and N is the number of nodes at the finest scale, in order to compute the estimates *and* error covariances at *all* nodes of the tree, compared to $\mathcal{O}(N^3)$ for the standard LLSE formalism.

It is important to realize that the multiscale model (1), together with the covariance $\boldsymbol{P}(0)$ of state $\boldsymbol{x}(0)$ at the root of the tree, provides an *implicit* specification of the full covariance of the multiscale process. That is, full covariances are never explicitly stored, rather represented implicitly with appropriate choices of $\boldsymbol{A}(s), \boldsymbol{B}(s)$. The explicit covariance between any two nodes $\boldsymbol{x}(s_1)$ and $\boldsymbol{x}(s_2)$ can be easily calculated as

$$\boldsymbol{P}(s_1, s_2) \triangleq E\left[\boldsymbol{x}(s_1)\boldsymbol{x}^T(s_2)\right] = \boldsymbol{\Phi}(s_1, s_1 \wedge s_2)\boldsymbol{P}(s_1 \wedge s_2)\boldsymbol{\Phi}^T(s_2, s_1 \wedge s_2), \tag{3}$$

where $s_1 \wedge s_2$ is the first common ancestor of s_1 and s_2, $\boldsymbol{P}(s_1 \wedge s_2)$ is the covariance of $\boldsymbol{x}(s_1 \wedge s_2)$, and $\boldsymbol{\Phi}(s, \sigma)$ is the state transition matrix from any node σ to direct descendent s. For example, referring to Fig. 2,

$$\boldsymbol{P}(s\alpha_i, u) \triangleq E\left[\boldsymbol{x}(s\alpha_i)\boldsymbol{x}^T(u)\right] = \boldsymbol{A}(s\alpha_i)\boldsymbol{A}(s)\boldsymbol{P}(s\bar{\gamma})\boldsymbol{A}^T(u). \tag{4}$$

Furthermore, the covariances of $\boldsymbol{x}(s)$ at each individual node can be recursively computed from a tree-recursive Lyapunov equation:

$$\boldsymbol{P}(s) = \boldsymbol{A}(s)\boldsymbol{P}(s\bar{\gamma})\boldsymbol{A}^T(s) + \boldsymbol{B}(s)\boldsymbol{B}^T(s). \tag{5}$$

Thus the calculation of $\boldsymbol{P}(s)$ and any individual $\boldsymbol{P}(s_1, s_2)$ is computationally simple (at most $\mathcal{O}(N)$ for all of the $\boldsymbol{P}(s)$), whereas clearly the calculation of *all* of the cross-covariances $\boldsymbol{P}(s_1, s_2)$ is prohibitively complex.

Given the definition of the framework (1),(2) and the efficient algorithm of Appendix A, the issue which remains is the selection of an appropriate model. Although the estimator itself, *given* a model, is exact, normally the model is an approximation to the problem of interest. There exists a body of research (Daniel and Willsky (1997b); Irving et al. (1997); Irving (1998)) for the stochastic realization of multiscale models that exactly or approximately match a second-order specification. Let $\boldsymbol{\chi}$ denote the ideal finest-scale process we wish to realize and $\mathbf{x}$ the set of finest-scale state variables, with respective covariances $\boldsymbol{P}_{\boldsymbol{\chi}}$, $\boldsymbol{P}_{\mathbf{x}}$. Then the realization problem is to find $\boldsymbol{A}(s), \boldsymbol{B}(s)$ such that $\boldsymbol{P}_{\mathbf{x}}$ (approximately) equals $\boldsymbol{P}_{\boldsymbol{\chi}}$, and such that the states above the finest scale satisfy the tree-Markovianity property in (1).

We can interpret (1) as writing $\boldsymbol{x}(s)$ as an estimate based on its parent $\boldsymbol{x}(s\bar{\gamma})$ plus the (orthogonal) error in this estimate, implying that the parameters $\boldsymbol{A}(s)$ and $\boldsymbol{B}(s)$ are completely determined by the joint statistics of $\boldsymbol{x}(s)$ and $\boldsymbol{x}(s\bar{\gamma})$:

$$\boldsymbol{A}(s) = \boldsymbol{P}(s, s\bar{\gamma})\boldsymbol{P}^{-1}(s\bar{\gamma}), \tag{6}$$

$$\boldsymbol{B}(s)\boldsymbol{B}^T(s) = \boldsymbol{P}(s) - \boldsymbol{P}(s, s\bar{\gamma})\boldsymbol{P}^{-1}(s\bar{\gamma})\boldsymbol{P}^T(s, s\bar{\gamma}). \tag{7}$$

Thus, model construction is immediate once the multiscale states are defined *and* the parent-child second-order statistics are computed at each node. We will find it useful to define the state $\boldsymbol{x}(s)$ as a linear functional of the finest-scale process $\mathbf{x}$:

$$\boldsymbol{x}(s) = \boldsymbol{L}(s)\mathbf{x}. \tag{8}$$

Once the internal matrices $\boldsymbol{L}(s)$ are defined the required parent-child statistics follow immediately. Specifically, recalling the objective that $\boldsymbol{P}_{\mathbf{x}} = \boldsymbol{P}_{\boldsymbol{\chi}}$, we have

$$\boldsymbol{P}(s, s\bar{\gamma}) = \boldsymbol{L}(s)\boldsymbol{P}_{\boldsymbol{\chi}}\boldsymbol{L}^T(s\bar{\gamma}), \tag{9}$$

$$\boldsymbol{P}(s) = \boldsymbol{L}(s)\boldsymbol{P}_{\boldsymbol{\chi}}\boldsymbol{L}^T(s). \tag{10}$$

The design of the $\boldsymbol{L}(s)$ matrices then focuses on meeting the Markovianity property in (1). A general method for constructing $\boldsymbol{L}(s)$ to achieve the required exact or approximate decorrelation is described in (Irving (1998)), and represents a generalization of Akaike's canonical correlations algorithm (Akaike (1975)) for the stochastic realization of time series.

This method is quite complex computationally in general. However, in many cases including ours, this step can be bypassed completely. In addition to the stochastic realization research, cited earlier, there exists a body of literature (Chou et al. (1994a); Daniel and Willsky (1997a); Fieguth et al. (1998); Luettgen et al. (1993)) describing various multiscale models which have been developed. One of the simplest models, which effectively models one-dimensional Markov processes, is the "endpoint" model shown in Fig. 3; it provides an *exact* multiscale realization for any 1-D first-order MRF and has been shown to yield excellent approximate models for a wide variety of stochastic processes (Daniel and Willsky (1997b); Irving (1998)). We will adopt and further elaborate upon this end-point model in Section 4.

Finally, determining $\boldsymbol{A}(s)$ and $\boldsymbol{B}(s)$ via (6), (7) requires knowing specific fine-scale cross-covariance values contained in $\boldsymbol{P}(s)$, $\boldsymbol{P}(s\bar{\gamma})$, and $\boldsymbol{P}(s, s\bar{\gamma})$. Of course (9) and (10) show how to compute these in terms of the full fine-scale process covariance $\boldsymbol{P}_{\boldsymbol{\chi}}$, but unless this fine-scale process is of low dimension (in which case there is no reason to use this multiscale approach in the first place) computing or storing all of $\boldsymbol{P}_{\boldsymbol{\chi}}$ is out of the question. Indeed, the motivation for our approach is precisely that for problems of substantial size the implementation of exact Kalman filtering equations and in particular the computation and storage of the error covariance through the Riccati equation are out of the question because of the high-dimensionality of the process. For this reason we need an alternative to (9) and (10) to propagate the needed statistical information. This is the topic of the next section.

3 Multiscale Dynamic Estimation

3.1 General Approach

Consider a discrete-time system, whose temporal dynamics are governed by

$$\boldsymbol{z}(t+1) = \boldsymbol{A}_d \boldsymbol{z}(t) + \boldsymbol{w}_d(t), \tag{11}$$

where $\boldsymbol{w}_d(t)$ is the zero mean process noise with diagonal covariance $\boldsymbol{Q}_d$. The measurements are

$$\boldsymbol{y}_d(t) = \boldsymbol{C}_d(t)\boldsymbol{z}(t) + \boldsymbol{v}_d(t), \tag{12}$$

where $\boldsymbol{v}_d(t)$ is the measurement noise with zero mean and diagonal covariance $\boldsymbol{R}_d$. The temporal dynamics, process noise, and measurement noise are assumed to be stationary in time, i.e., $\boldsymbol{A}_d$, $\boldsymbol{Q}_d$, and $\boldsymbol{R}_d$ are independent of t.

For the applications we have in mind (see Section 4) $\boldsymbol{z}(t)$ would represent a spatially-discretized distributed parameter process and (11) would represent the corresponding temporally-discretized dynamics, so that $\boldsymbol{A}_d$ represents the discretization of a partial differential operator in space. In addition, $\boldsymbol{R}_d$ is diagonal and the components of $\boldsymbol{y}_d$ represent independent point measurements of the distributed process.

We are interested in modeling the estimation error, so we let

$$\boldsymbol{\chi}(t|\tau) = \boldsymbol{z}(t) - \hat{\boldsymbol{z}}(t|\tau). \tag{13}$$

where $\hat{\boldsymbol{z}}(t|\tau)$ denotes the estimate of $\boldsymbol{z}(t)$ based on measurements through time τ. The Kalman filter, as sketched in Fig. 1, consists of a prediction stage

$$\hat{\boldsymbol{z}}(t+1|t) = \boldsymbol{A}_d \hat{\boldsymbol{z}}(t|t), \tag{14}$$

and a measurement update stage

$$\hat{\boldsymbol{z}}(t|t) = \hat{\boldsymbol{z}}(t|t-1) + \hat{\boldsymbol{\chi}}(t|t-1). \tag{15}$$

In standard Kalman filtering the estimate $\hat{\boldsymbol{\chi}}(t|t-1)$ is calculated explicitly as

$$\hat{\boldsymbol{\chi}}(t|t-1) = \boldsymbol{P}_{\boldsymbol{\chi}}(t|t-1)\boldsymbol{C}_d^T \left(\boldsymbol{C}_d \boldsymbol{P}_{\boldsymbol{\chi}}(t|t-1)\boldsymbol{C}_d^T + \boldsymbol{R}_d\right)^{-1} \boldsymbol{y}(t), \tag{16}$$

However for the problems of interest here the dimensionality of $\boldsymbol{\chi}(t|t-1)$ makes this explicit calculation either impossible or at best exceedingly complex.

The alternative approach that we propose in this paper is to *implicitly* calculate and propagate the statistics of the estimation error as a sequence of multiscale models, illustrated in Fig. 4. Specifically, suppose that we have a multiscale model $\boldsymbol{A}(s;t|t-1)$, $\boldsymbol{B}(s;t|t-1)$ for the prediction error, defining the states as

$$\boldsymbol{x}(s;t|t-1) = \boldsymbol{L}(s)\boldsymbol{\chi}(t|t-1). \tag{17}$$

The multiscale estimation formulation in Appendix A yields the estimates $\hat{\boldsymbol{x}}(s;t|t-1)$ *and* a multiscale model $\boldsymbol{A}(s;t|t)$, $\boldsymbol{B}(s;t|t)$ for the updated estimation error $\boldsymbol{\chi}(t|t)$:

$$\boldsymbol{x}(s;t|t) = \boldsymbol{L}(s)\boldsymbol{\chi}(t|t), \tag{18}$$

where the parameters $\boldsymbol{A}(s;t|t)$ and $\boldsymbol{B}(s;t|t)$ computed as part of the multiscale estimation process. That is, if we start with a multiscale model (17) for $\boldsymbol{\chi}(t|t-$

1) we directly obtain an analogous model (18) for $\boldsymbol{\chi}(t|t)$ without explicitly calculating $\boldsymbol{P}_{\boldsymbol{\chi}}(t|t)$.

To complete one step of the recursion we need to compute a multiscale model for the next predicted errors

$$\boldsymbol{\chi}(t+1|t) = \boldsymbol{A}_d \boldsymbol{\chi}(t|t) + \boldsymbol{w}_d(t) \tag{19}$$

without explicitly calculating $\boldsymbol{P}_{\boldsymbol{\chi}}(t+1|t)$. Finding the predicted multiscale model, not provided by the multiscale estimation formulation, is explored in the following subsection and is novel to this paper.

3.2 Multiscale Prediction Step

We assume that the linear functionals $\boldsymbol{L}(s)$ have been specified and do not vary over time, although in general one might expect these linear functionals to change depending on how the statistics of the one-step prediction errors vary over time.

With the choice of the linear functionals $\boldsymbol{L}(s)$ made, we are left with the final key issue, namely determining and propagating the parameters of the multiscale model through the temporal dynamics (19).

We assume that know the predicted model $\boldsymbol{A}(s;t|t-1)$, $\boldsymbol{B}(s;t|t-1)$, $\boldsymbol{P}(s;t|t-1)$ in Fig. 4. Using the estimation and error modeling algorithms described in Appendix A we can compute the corresponding quantities $\boldsymbol{A}(s;t|t)$, $\boldsymbol{B}(s;t|t)$, and $\boldsymbol{P}(s;t|t)$ for the updated multiscale model. It remains to calculate the corresponding quantities $\boldsymbol{A}(s;t+1|t)$, $\boldsymbol{B}(s;t+1|t)$, and $\boldsymbol{P}(s;t+1|t)$ for the predicted model for $\boldsymbol{\chi}(t+1|t)$ whose states are

$$\boldsymbol{x}(s;t+1|t) = \boldsymbol{L}(s)\boldsymbol{\chi}(t+1|t). \tag{20}$$

From (6), (7), (9), (10), we see that $\boldsymbol{A}(s;t+1|t)$ and $\boldsymbol{B}(s;t+1|t)$ will be determined if we determine both the individual state covariances $\boldsymbol{P}(s;t+1|t)$ and the parent-child cross-covariances $\boldsymbol{P}(s,s\bar{\gamma};t+1|t)$. To derive expressions for the individual elements of these covariances, we first substitute the temporal dynamics (19) into (20):

$$\boldsymbol{x}(s;t+1|t) = \boldsymbol{L}(s)\boldsymbol{A}_d\boldsymbol{\chi}(t|t) + \boldsymbol{L}(s)\boldsymbol{w}_d(t). \tag{21}$$

Unless $\boldsymbol{A}_d$ and $\boldsymbol{L}(s)$ commute, the term $\boldsymbol{L}(s)\boldsymbol{A}_d\boldsymbol{\chi}(t|t)$ mixes the linear functionals used to form the states $\boldsymbol{x}(s;t|t)$ in (18). If we let $\boldsymbol{l}_i^T(s)$ denote the ith

row (linear functional) of $\boldsymbol{L}(s)$ then

$$x_i(s;t+1|t) = \boldsymbol{l}_i^T(s)\boldsymbol{A}_d\boldsymbol{\chi}(t|t) + \boldsymbol{l}_i^T(s)\boldsymbol{w}_d(t). \tag{22}$$

The term $\boldsymbol{l}_i^T(s)\boldsymbol{A}_d$ represents some linear functional of $\boldsymbol{\chi}(t|t)$, but in general it will not correspond to any of the linear functionals which we already have in $\boldsymbol{L}(s)$. However, it is always possible to write it as a *linear combination* of existing functionals, since the collection of linear functionals at the finest scale already forms a basis. Therefore we can write

$$\boldsymbol{l}_i^T(s)\boldsymbol{A}_d = \sum_{(\sigma,j)\in S} h_{\sigma,j}^{s,i}\boldsymbol{l}_j^T(\sigma), \tag{23}$$

Therefore we can write $x_i(s;t+1|t)$, not in terms of $\boldsymbol{\chi}(t|t)$, but instead based on selected model states:

$$x_i(s,t+1|t) = \sum_{(\sigma,j)\in S} h_{\sigma,j}^{s,i}x_j(\sigma,t|t) + \boldsymbol{l}_i^T(s)\boldsymbol{w}_d(t). \tag{24}$$

From (24) we can compute the quantities $\boldsymbol{P}(s;t+1|t)$ and $\boldsymbol{P}(s,s\bar{\gamma};t+1|t)$ by computing certain covariances

$$E\left[x_i(s;t+1|t)x_j(u;t+1|t)\right], \tag{25}$$

which itself is computed from the known covariances $\boldsymbol{P}(s;t|t)$ and (3). However, as we pointed out in Section 2, calculating all or even many of these cross-covariances is prohibitive, thus it is desirable, if possible, to choose among the various solutions to (23) those in which $h_{\sigma,j}^{s,i}$ are extremely sparse and in fact are nonzero only for nodes σ that are near to node s.

Since the specific properties of the $h_{\sigma,j}^{s,i}$ are highly dependent upon the dynamics, we need to study the problem in the context of specific dynamics; in the next section we will demonstrate multiscale model prediction in the context of discretized diffusion PDEs.

4 Applications to 1-D Diffusions

In this section we apply the ideas of Section 3 in detail to estimating 1-D diffusion problems. We will develop solutions to the two major issues identified in Section 3, namely the choice of the linear functionals $\boldsymbol{L}(s)$ and the propagation of the multiresolution model through the dynamic prediction step. In Section 5 we will illustrate the performance of the resulting estimator.

4.1 Problem Setup

The point of departure for this application is 1-D damped heat diffusion process on a rod or ring satisfying the following stochastic PDE:

$$\frac{\partial z(l,\tau)}{\partial \tau} = a \cdot \frac{\partial^2 z(l,\tau)}{\partial l^2} - b \cdot z(l,\tau) + c \cdot w(l,\tau), \tag{26}$$

where $z(l,\tau)$ is the temperature at location l at time τ, $w(l,\tau)$ is a white Gaussian noise with unit variance, and $l \in [0, L]$. Constant a is related to quantities such as the heat conduction coefficient, the dimensions of the rod or ring, etc; b controls the heat loss to the surrounding coolant, whose temperature is set to zero without loss of generality.

The number of free parameters in (26) can be reduced by normalizing the spatial dimension to unit length and the diffusion parameter to 1:

$$\frac{\partial z(l,\tau)}{\partial \tau} = \frac{\partial^2 z(l,\tau)}{\partial l^2} - \beta \cdot z(l,\tau) + \gamma \cdot w(l,\tau). \tag{27}$$

A number of finite difference schemes can be applied to discretize this PDE to arrive at a system of difference equations in the form of (11)

$$\boldsymbol{z}(t+1) = \boldsymbol{A}_d \boldsymbol{z}(t) + \boldsymbol{w}_d(t), \tag{28}$$

where $\boldsymbol{z}(t)$ is the vector containing the temperatures at all spatial grid points at time step t, and $\boldsymbol{w}_d(t)$ models the process noise, with covariance $\boldsymbol{Q}_d$. For our purposes such a discretized model plays two related but distinct roles: the prediction of the estimates (14), and to provide the dynamic matrix (23)–(24) to predict the estimation error statistics. As we will see, for the latter of these it is quite desirable to choose $\boldsymbol{A}_d$ to be banded with relatively small bandwidth. Such an $\boldsymbol{A}_d$ arises if we use an explicit finite-difference temporal discretization. In particular, in the results presented here we used a simple forward Euler scheme, in which case $\boldsymbol{A}_d$ is tridiagonal and $\boldsymbol{Q}_d = \sigma_w^2 \boldsymbol{I}$. Of course if we use such a scheme for the former purpose as well, i.e., in the actual prediction step (14), care must be taken to ensure that the spatial discretization step size Δl and the temporal step size $\Delta\tau$ are small enough for numerical accuracy and convergence (Strikwerda (1989)).

Obviously a better choice for this former purpose would be an implicit discretization scheme, which would result in a dense matrix $\boldsymbol{A}_d$. We can actually consider using a different choice of $\boldsymbol{A}_d$ for each of these two purposes, namely an implicit scheme for the prediction step of the estimates and an explicit

scheme in propagating the multiscale error models through the dynamic prediction step. If we view the former as the (more) "exact" model, we are then performing the estimate prediction "exactly" but are propagating the error statistics only approximately, so that the way in which new measurements are incorporated in the update step will not be optimal. However, the multiscale error model *already* introduces an approximation into the update step, and we will see that the net effect of all of these approximations is a surprisingly small loss in performance.

We assume point measurements that may be irregular in space, but stationary in time (except for the last example in Section 5):

$$\boldsymbol{y}_d(t) = \boldsymbol{C}_d \boldsymbol{z}(t) + \boldsymbol{v}_d(t), \tag{29}$$

where $\boldsymbol{C}_d$ is a selection matrix and the measurement noise $\boldsymbol{v}_d$ is white with covariance $\boldsymbol{R}_d = \sigma_v^2 \boldsymbol{I}$ and is uncorrelated with $\boldsymbol{z}(t)$ or $\boldsymbol{w}_d(t)$. Different choices for σ_w^2 and σ_v^2 with a constant ratio only scale the steady-state covariances; therefore, given a particular measurement configuration $\boldsymbol{C}_d$, we are left with only two free parameters, namely β and σ_w^2/σ_v^2.

Of course the complete specification of the model (28) also implies the specification of a specific set of boundary conditions. For the purpose of describing our methodology in this section, we will assume circular boundary conditions, $z(0,\tau) = z(L,\tau)$, physically corresponding to a thin cooling ring immersed in a coolant. In this case, the steady-state process variance σ_p^2, i.e., the diagonal elements of $\boldsymbol{P}_{\boldsymbol{z}}$, are constant as long as the process (26) has spatially constant parameters. We will use the more familiar notion of signal-to-noise ratio $\mathrm{SNR} = 10\log\left(\sigma_p^2/\sigma_v^2\right)$ instead of σ_w^2/σ_v^2. We can also adjust σ_w^2 to normalize σ_p^2 to 1. The stipulation of other boundary conditions leaves the analysis of the linear functionals and the development of the multiscale prediction algorithm unchanged and effects only the specific numerical values that result.

4.2 Linear Functionals

We are seeking a set of linear functionals which allow us to develop an accurate multiscale model for the steady-state predicted estimation errors in the context of one-dimensional diffusion. We propose to model the one-step predicted estimation errors as Markov processes. This is supported by experimental work(Chin et al. (1995)), which demonstrated cases in which estimation errors could be well-modeled by Markov random fields, and by theoretical work (Coleman (1995)), which showed that continuous-time, continuous-space heat diffusion models are Markov in steady-state. It is therefore reasonable to expect Markovianity of our model (28), except for discretization errors.

In the specific case of one-dimensional Markov processes, we have already seen in Fig. 3 an exact multiscale model(Chou et al. (1994a); Luettgen et al. (1993)), in which the coarser scale states are defined as so-called "endpoint" linear functionals, for which each state consists of the finest-scale process values taken at the endpoints of the subinterval represented by the state. Moreover, it has been demonstrated that such a choice of state is effective for many other processes as well(Daniel and Willsky (1997a)). Thus, based on this combination of previous analysis and experimental evidence, we will investigate the use of such functionals for space-time estimation problems of the type examined in our paper.

We will test our choice of linear functionals in two ways. In Section 5 we will compare our multiscale approach, based on endpoint linear functionals, with the exact Kalman filter. In this section, we will derive the best choice of linear functionals for discretized diffusion, and will compare these to the endpoint functionals.

Specifically, for small-size systems it is computationally possible to *explicitly* solve the Riccati equation for the exact steady-state error covariance. That is, we can explicitly compute the full covariance $\boldsymbol{P}_{\boldsymbol{\chi}}(t|t-1)$ for the process to be realized at the finest level of the tree. From this covariance we can then use the canonical correlations realization (CCR) algorithm (Irving (1998)) to construct multiscale realizations explicitly, that is, to find the most appropriate selection of linear functionals. The insights gained from this procedure, applied to small-size systems, may then be applied to larger systems, where neither the Riccati equation nor CCR are computationally feasible.

Consider the case of interest here, namely the multiscale modeling of the one-step predicted error process $\boldsymbol{\chi}(t|t-1)$, for example in the case of estimating a 32-dimensional process based on the measurement of a single point. We explicitly solved the Riccati equation for the steady-state prediction error covariance $\boldsymbol{P}_{\boldsymbol{\chi}}(t|t-1)$. Given this covariance, the CCR algorithm (via a singular value decomposition) produces at each node s a set of linear functionals ordered by statistical significance; that is, ordered by the singular value associated with each linear functional, measuring the degree to which it decorrelates node s from the remainder of the tree.

Except in very special cases, the Riccati error covariance is spatially nonstationary and non-Markov as well. Nevertheless, the multiresolution representation still represents an excellent choice. Fig. 3 illustrates the application of CCR to decorrelating the interval $\{1-16\}$ from $\{17-32\}$ for two different measurement locations; because of the intrinsic nonstationarity introduced by the tree, the location of the measurement has some influence on the results of CCR. The four most significant linear functionals produced by CCR are shown for the "best" (location 8) and "worst" (location 16) measurement

placements. The immediate conclusion is that the two most significant linear functionals are almost completely concentrated on the interval end-points. The relative insignificance of the third and fourth linear functionals as a function of measurement location is depicted in Fig. 6.

An alternative assessment is to use the multiscale model, based on endpoint functionals, for static estimation. Specifically we use the *fractional variance reduction* (FVR), comparing the steady-state process variance and the steady-state updated error variance, as a measure of estimator performance:

$$\text{FVR} = \frac{\text{Var(s.s. process)} - \text{Var(s.s. updated error)}}{\text{Var(s.s. process)}} \tag{30}$$

For instance, if the FVR for the optimal estimator is 0.99 and for a suboptimal estimator is 0.98, we would claim that the suboptimal estimator has done a very good job, although its error variance is twice as large as the optimal error variance.

Fig. 7 depicts results for four different measurement locations. Fig. 7(a) shows the optimal and multiscale FVRs; at the resolution of this plot all of these curves are indistinguishable. Fig. 7(b) displays the percentage difference between each of the multiscale FVRs and the optimum; the differences are very small, peaking in the *worst-case* with a measurements on a tree boundary (point 16) with an FVR of approximately 0.596, whereas the optimal estimator has an FVR of 0.6.

4.3 Multiscale Prediction Step for 1-D Diffusion

For the small examples considered thus far we could explicitly solve the Riccati equation, compute $\boldsymbol{P}_{\boldsymbol{\chi}}(t+1|t)$, and determine the multiscale model for $\boldsymbol{\chi}(t+1|t)$. For large problems, however, we must directly infer the model for $\boldsymbol{P}_{\boldsymbol{\chi}}(t+1|t)$ from the model for the updated errors $\boldsymbol{P}_{\boldsymbol{\chi}}(t|t)$, a model which is computed by the multiscale estimation algorithm. The problem is that only the child-parent cross-covariances and individual node covariances are explicitly calculated during the multiscale estimation process, whereas in general the mixing due to the dynamics $\boldsymbol{A}_d$ requires that more distant correlations be calculated (as specified by the $h^{s,i}_{\sigma,j}$ in (23)–(24)).

At this point there are two key insights. The first is that $\boldsymbol{A}_d$ is tridiagonal, therefore the mixing it induces is comparatively sparse, implying that the number of elements of $\boldsymbol{P}_{\boldsymbol{\chi}}(t|t)$ that are needed to determine the model for $\boldsymbol{\chi}(t+1|t)$ is also comparatively small. In general, these elements can be found exactly by computing the desired cross-covariance based on (3), or found

approximately by any number of covariance-extension or maximum-entropy techniques (Dempster (1972); Ho (1998); Lev-Ari et al. (1989)).

The second insight is that there are many possible specific choices of end-point linear functionals, and that the choice can greatly affect the number and complexity of cross-covariances required to be computed. In this paper we introduce a new class of end-point linear functionals, shown in Fig. 3(b), which possess the following desirable property: the left and right spatial neighbors of *any* linear functional at any node s can be found in s, the parent of s, or a child of s. Therefore, since the diffusion dynamics $\boldsymbol{A}_d$ are tridiagonal, and thus since the dynamic mixing of a given pixel is limited to its spatial left and right neighbors, therefore the cross-covariance elements required to predict the multiscale model exactly are never more than three scales apart, regardless of the overall size of the problem. Therefore the complexity, *per tree node*, to predict the model is $\mathcal{O}(1)$.

4.4 Iterative and Recursive Implementation

The complete algorithm we have just described can be used in one of two ways. One, to obtain an approximate multiscale model for the steady-state prediction error process by running the algorithm iteratively off-line until convergence is achieved. Second, to use this algorithm to provide a multiscale error model dynamically at each step of the recursive estimation procedure for the initial, transient phase of estimation or for problems temporally non-stationary. The following paragraphs comment on issues of complexity, initialization, stopping criteria, and sources of inaccuracy.

The computational complexity, per time-step, of the algorithm is as follows. Using the end-point linear functionals from Fig. 3(b), the state dimension $d \leq 3$ for any node on the tree, regardless of the problem size. Therefore the total complexity of the update step, based on the algorithm of Appendix A, is $\mathcal{O}(N)$. From the previous section, using the same linear functionals, the prediction step complexity is also $\mathcal{O}(N)$, therefore the total complexity of the dynamic multiscale estimator for discretized diffusion problems is only $\mathcal{O}(N)$ per time step!

The initialization of our algorithm takes the form of specifying a multiscale model for $\boldsymbol{\chi}(0)$, the prior estimation errors. Constructing such a model involves evaluating those elements of the prior covariance $\boldsymbol{P}_{\boldsymbol{\chi}}(0)$ in order to derive the self statistics of each tree node and the cross statistics between every node and its parent. While covariance extension and maximum-entropy methods can be used, often we can obtain these desired elements more easily using the FFT, for example, if the dynamics are space-invariant and assuming circular boundary

conditions. This latter method is used for initialization in the examples in Section 5.

If we are iteratively calculating a multiscale model for the steady-state estimation errors, then the iteration stopping criterion is a critical issue. From a theoretical view point, the convergence of the solution of the time-varying Riccati equation to its steady-state limit is controlled by the slowest time constant of the steady-state error dynamics $\boldsymbol{A}_d\,(\boldsymbol{I} - \boldsymbol{K}(\infty)\boldsymbol{C}_d)$, and thus choosing the number of iterations to be several times this time constant provides a conservative bound. However, (a) for large problems this time constant will generally be unavailable, and (b) taking this conservative approach may lead to an excessive numbers of iterations. In response to (a) an alternative, adaptive stopping criterion is to examine the diagonal elements of $\boldsymbol{P}_{\boldsymbol{\chi}}(t|t)$ and stop when these suggest convergence (e.g., when the average RMS difference between elements of the diagonal at two successive iterations falls below a specified fraction of the average of the diagonal values). In response to (b), in our experiments we restrict the number of iterations to be $\mathcal{O}(\log N)$ so that the total complexity is $\mathcal{O}(N \log N)$. Although this implies that the resulting multiscale estimator may not have converged, the results in the next section demonstrate that the performance of the resulting estimators is close to the optimal Kalman filter.

Given that the multiscale estimation algorithm of Appendix A is exact, the only sources of error lie in the realized multiscale model itself: the termination of iterations prior to convergence to model steady-state, the temporal and spatial discretization of the dynamics, and choosing end-point linear functionals as the basis for the multiscale modeling of the estimation error. Furthermore, because our model propagation assumes the updated statistics to be exact, it is possible that errors are accumulating over time. Although each of these sources of error can be reduced at the expense of additional computational complexity (e.g., using more iterations, a smaller step size, more linear functionals etc.), the results of the following section will show that the algorithm performs nearly optimally at little statistical cost.

5 Examples and Results

In this section we illustrate the application of our methodology to several examples of size $N = 64$. At this size, performance comparisons with the optimal estimator are possible because exact calculations for the optimal estimator are still feasible. The iterative multiscale algorithm described in this paper can in principle be applied more generally to any dynamic process whose steady-state error process can be adequately modeled using end-point linear functionals and whose dynamics are local. To illustrate some of this flexibility we include examples representing extensions and departures from the basic

diffusion problem described in the preceding section.

5.1 Cooling Ring

We start with a cooling ring and a single measurement. The steady-state process variance has been normalized to 1. Fig. 8(a) shows the variance reduction plots for several values of SNR and heat loss parameter $\beta = 10$. As the SNR increases, so does the percent variance reduction. In all cases, the multiscale estimator is less than 0.2% poorer than the optimal estimator in steady state, as seen in panel (b). The greatest degradation in performance occurs in regions furthest away from the measurement, i.e., where the error variances are large.

The single measurement case is, in a sense, the worse case scenario, as the system is only weakly observable. In multiple measurement cases, the performance of our multiscale estimator compared to the optimal is generally better than that shown in Fig. 8 (Ho (1998)).

5.2 Pinned Fin

In this example we introduce two variations. First, we replace the cyclic boundary condition by the more realistic condition for a cooling fin: one end of the fin is pinned to a heat source and the other end immersed in a coolant. The boundary condition at the heat source is Dirichlet: $z(0,t) = z_0$. At the free end, the heat flux is set to be equal to the heat loss, $\partial z(l,t)/\partial l = -\beta z(l,t)$. The second variation recognizes the fact that in practice the heat loss parameter β may be spatially varying if the coolant is non-homogeneous (Aihara (1997)), for example, when the fin is partially insulated or cooled by air and partially immersed in a liquid coolant. The discretized dynamic equation (28) will then have a non-circulant $\boldsymbol{A}_d$ and an extra term $\boldsymbol{B}_d\boldsymbol{u}(t)$ to account for the boundary condition:

$$\boldsymbol{z}(t+1) = \boldsymbol{A}_d\boldsymbol{z}(t) + \boldsymbol{B}_d\boldsymbol{u}(t) + \boldsymbol{w}_d(t). \tag{31}$$

With non-circular boundary conditions or spatially varying heat loss β, the steady-state diffusion process becomes spatially non-stationary. This requires a modified definition of SNR: we will use the maximum pointwise SNR, $10\log\left(\max_i \sigma_p(i)^2/\sigma_v^2\right)$.

Fig. 9(a) shows the spatially nonstationary steady-state process variance and the steady-state error variances of the estimators for the case in which measurements are available at two spatial locations. The plots of the latter two are

indistinguishable here: as shown in panel (c) their difference is only a fraction of a percent. The near equality of the true optimal error variances and the actual error variance of our suboptimal estimator demonstrate the excellent performance of our method. Of course for truly large problems we would not have access to either of these quantities because of computational limitations. What we *do* have, however, is the multiscale approximate model for $\boldsymbol{\chi}(t|t)$ which provides the values of the variances that this estimator *believes* it is achieving. This is illustrated as the dotted line in panel (a). Note that these variances are also quite accurate, although they slightly underestimate the actual error variance.

In order to test the ability of our procedure to deal with substantial mixing we performed a number of experiments in which the measurement sampling rate was substantially slower than the dynamic time step. Fig. 10 shows the same pinned fin example of Fig. 9 except that a measurement is available only every 200 prediction steps, thus allowing substantial mixing to occur between measurements. Since *many* predictions steps must be taken for every update step, the effects of the approximations in our multiscale algorithm are much more pronounced, yet in the worse case it is still within 3% of the optimal estimator.

5.3 Advection-Diffusion

In order to test the breadth of utility of our methodology we also have examined its application to a different and very important class of models, namely advection-diffusion processes. Such models have been employed in a wide variety of applications, especially in fluid dynamics, from pollution monitoring (Omatu et al. (1988)), to modeling tracer movements in oceanography (Wunsch (1988, 1987)).

The resulting dynamics

$$\frac{\partial z(l,\tau)}{\partial \tau} = \frac{\partial^2 z(l,\tau)}{\partial l^2} + \rho \cdot \frac{\partial z(l,\tau)}{\partial l} - \beta(l) \cdot z(l,\tau) + \gamma \cdot w(l,\tau), \qquad (32)$$

model a thin pipe, in which a liquid flows towards positive l from a reservoir. Fig. 11 displays the results from one such estimation problem with measurements at two spatial locations. The multiscale approximate estimator tracks the performance of the optimal estimator closely. Also, the approximate error variances captured by the multiscale model (and corresponding to the dotted line in panel (a)) provide a very good approximation to the actual error statistics.

5.4 Recursive Implementation and Temporally Nonstationary Performance

Fig. 12 illustrates the application of the recursive version of our algorithm to a temporally nonstationary situation, based on the advection-diffusion dynamics in (32), initializing the process with the nonequilibrium initial condition indicated by the solid line in Fig. 12(a). The measurements are taken once every 100 prediction steps and are highly nonstationary, the number of measurements at each measurement time being Poisson with mean 4, and the measurement locations uniformly distributed.

Since there are no steady-state performance figures to speak of, we have depicted a snapshot of the process and estimation results at time step 1300 (i.e., after the 13th update). The dash-dot line in Fig. 12(a) indicates the actual process at time 1300, while Fig. 12(b) depicts the estimation results after the measurement update. The two measurements taken at this time are indicated by the small circles. As these figures illustrate, the estimates produced by the optimal Kalman filter and by our multiscale recursive estimator are virtually identical and the differences are statistically insignificant.

6 Conclusion

In this paper we have developed a new approach to suboptimal estimation for recursive estimation for distributed parameter space-time phenomena. The point of departure for our work are the basic equations of Kalman filtering, which can be prohibitively complex because of the growth in computational complexity with the dimensions of the spatial domain of interest. Indeed this is one of the most significant challenges faced in data assimilation of remote sensing data for large-scale geophysical studies.

Our solution to this problem involves making use of the observation that each update step in recursive estimation can be viewed as a static spatial estimation problem in which the errors in the predicted estimates are estimated based on the latest measurement innovations. Thus, rather than explicitly propagating the full error covariance for this spatial prediction error field, we consider propagating a *model* for these estimation errors. In particular, rather than using standard models for these error fields such as Markov random fields, we have chosen to use a recently introduced class of multiscale models. This structure leads to extremely fast algorithms for estimation. The major challenge in applying this multiscale methodology is in developing a method for propagating multiscale error field models through the mixing due to the temporal dynamics of the process being estimated.

The estimation results obtained indicate that near-optimal performance can be achieved using this methodology. Indeed, we would argue that, compared to the intrinsic model uncertainty in many of the space-time processes of interest such as in remote sensing applications, the differences in performance between our algorithm and the Kalman filtering solution are insignificant.

While we have illustrated our results here for 1-D spatial processes, much greater benefits can be expected in 2- and 3-dimensional problems. While the basic concept of how to develop this extension is described in this paper, important issues remain in order to make this extension a reality. In particular, the choice of multiscale states in the representation of estimation error fields represents a first important problem that is currently under investigation. In addition, while diffusion and advection-diffusion problems such as those considered in this paper are of considerable practical interest in higher dimensions, it is also of considerable interest to understand how to adapt our methodology to dynamics that allow wave-like behavior. Obviously for such models we would expect that the propagation of error models over time would need to account for the modes of wave propagation. Issues such as these as well as developing a deeper understanding of how to capture temporal mixing of scales within our multiresolution framework represent clearly defined directions to be pursued in order to fully realize the promise that the results presented here suggest.

References

Shin Ichi Aihara (1997). "On Adaptive Boundary Control for Stochastic Parabolic Systems". *IEEE Transactions on Automatic Control*, 42 (3): 350–363.

H. Akaike (1975). "Markovian Representation of Stochastic Processes by Canonical Variables". *SIAM Journal of Control*, 13 (1).

H. T. Banks and K. Kunisch (1989). *Estimation Techniques for Distributed Parameter Systems*. Birkhäuser, Boston.

Michele Basseville, Albert Benveniste, Kenneth Chou, Stuart Golden, Ramine Nikoukhah, and Alan Willsky (1992). "Modeling and Estimation of Multiresolution Stochastic Processes". *IEEE Transactions on Information Theory*, 38 (2): 766–784.

William L. Briggs (1987). *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, Philadelphia.

Toshio Michael Chin, William Clem Karl, and Alan S. Willsky (1995). "A Distributed and Iterative Method for Square Root Filtering in Space-Time Estimation". *Automatica*, 31 (1): 67–82.

Kenneth Chou, Alan Willsky, and Albert Benveniste (1994a). "Multiscale Recursive Estimation, Data Fusion, and Regularization". *IEEE Transactions on Automatic Control*, 39 (3): 464–478.

Kenneth Chou, Alan Willsky, and Ramine Nikoukhah (1994b). "Multiscale Systems, Kalman Filters, and Riccati Equations". *IEEE Transactions on Automatic Control*, 39 (3): 479–492.

Jerome M. Coleman (1995). *Gaussian Spacetime Models: Markov Field Properties*. PhD Thesis, University of California at Davis, David, CA.

Michael Daniel and Alan Willsky (1997a). "Modeling and Estimation of Fractional Brownian Motion Using Multiresolution Stochastic Processes". In J. L. Vehel, E. Lutton, and C. Tricot, editors, *Fractals in Engineering*, pages 124–137. Springer.

Mike Daniel and Alan Willsky (1997b). "A Multiresolution Methodology for Signal-Level Fusion and Data Assimilation with Applications to Remote Sensing". *Proceedings of the IEEE*, 85 (1): 164–183.

A.P. Dempster (1972). "Covariance Selection". *Biometrics*, 28: 157–175.

Paul Fieguth, William Karl, Alan Willsky, and Carl Wunsch (1995). "Multiresolution Optimal Interpolation and Statistical Analysis of TOPEX/ POSEIDON Satellite Altimetry". *IEEE Transactions on Geoscience and Remote Sensing*, 33 (2): 280–292.

Paul Fieguth, Dimitris Menemenlis, Terrence Ho, Alan Willsky, and Carl Wunsch (1998). "Mapping Mediterranean Altimeter Data with a Multiresolution Optimal Interpolation Algorithm". *Journal of Atmospheric and Oceanic Technology*, 15: 535–546.

Paul Fieguth and Alan Willsky (1996). "Fractal Estimation Using Models on Multiscale Trees". *IEEE Transactions on Signal Processing*, 44 (5): 1297–1300.

Terrence T. Ho (1998). *Multiscale Modeling and Estimation of Large-Scale Dynamic Systems*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

T. Ho, P. Fieguth, and A. Willsky (1998). "Computationally Efficient Multiscale Estimation of Large-Scale Dynamic Systems," *IEEE ICIP '98*, Chicago.

William W. Irving (1998). "A Canonical Correlations Approach to Multiscale Stochastic Realization". *IEEE Transactions on Automatic Control*, In press.

William W. Irving, Paul W. Fieguth, and Alan S. Willsky (1997). "An Overlapping Tree Approach to Multiscale Stochastic Modeling and Estimation". *IEEE Transactions on Image Processing*, 6 (11): 1517–1529.

Hanoch Lev-Ari, Sydney R. Parker, and Thomas Kailath (1989). "Multidimensional Maximum-Entropy Covariance Extension". *IEEE Transactions on Information Theory*, 35 (3): 497–508.

Mark Luettgen, William Karl, Alan Willsky, and Robert Tenney (1993). "Multiscale Representations of Markov Random Fields". *IEEE Transactions on Signal Processing*, 41 (12): 3377–3395.

Mark Luettgen and Alan Willsky (1995). "Multiscale Smoothing Error Models". *IEEE Transactions on Automatic Control*, 40 (1): 173–175.

Dimitris Menemenlis, Paul Fieguth, Carl Wunsch, and Alan Willsky (1997). "Adaptation of a Fast Optimal Interpolation Algorithm to the Mapping of Oceanographic Data". *Journal of Geophysical Research*, 102 (C5): 10573–10584.

S. Omatu, J. H. Seifeld, T. Soeda, and Y. Sawaragi (1988). "Estimation of Nitrogen Dioxide Concentration in the Vincinity of a Roadway by Optimal Filtering Theory". *Automatica*, 24 (1): 19–29.

H. E. Rauch, F. Tung, and C. T. Striebel (1965). "Maximum Likelihood Estimates of Linear Dynamic Systems". *AIAA Journal*, 3 (8): 1445–1450.

T. P. Speed and H. T. Kiiveri (1986). "Gaussian Markov Distributions Over Finite Graphs". *The Annals of Statistics*, 14 (1): 138–150.

John C. Strikwerda (1989). *Finite Difference Schemes and Partial Differential Equations*. Chapman and Hall, New York.

Carl Wunsch (1987). "Using Transient Tracers: the Regularization Problem". *Tellus*, 39B: 477–492.

Carl Wunsch (1988). "Transient Tracers as a Problem in Control Theory". *Journal of Geophysical Research*, 93 (C7): 8099–8110.

Fig. 1.

Fig. 2.

Fig. 3.

Fig. 4.

Fig. 5.

Fig. 6.

Fig. 7.

Fig. 8.

Fig. 9.

Fig. 10.

Fig. 11.

Fig. 12.

A Multiscale Smoothing Algorithm

The essential equations of the multiscale smoothing algorithm are listed here. More detailed development of these equations can be found in (Chou et al. (1994a); Luettgen and Willsky (1995)).

Suppose that we are given the multiscale process and measurement equations:

$$\boldsymbol{x}(s) = \boldsymbol{A}(s)\boldsymbol{x}(s\bar{\gamma}) + \boldsymbol{B}(s)\boldsymbol{w}(s) \tag{A.1}$$
$$\boldsymbol{y}(s) = \boldsymbol{C}(s)\boldsymbol{x}(s) + \boldsymbol{v}(s) \tag{A.2}$$

where $\boldsymbol{w}(s)$ is a zero-mean unit-variance white noise process and $\boldsymbol{v}(s)$ is a zero-mean white noise process with covariance $\boldsymbol{R}(s)$. We are also given the statistics of the states at the root node: zero mean with covariance $\boldsymbol{P}(0)$. First, the prior covariances of all states at individual nodes on the tree are computed via a Lyapunov equation

$$\boldsymbol{P}(s) = \boldsymbol{A}(s)\boldsymbol{P}(s\bar{\gamma})\boldsymbol{A}^T(s) + \boldsymbol{B}(s)\boldsymbol{B}^T(s) \tag{A.3}$$

The core of the multiscale algorithm consists of an upward estimation sweep and a downward smoothing sweep, but first let us define a few quantities:

$$\boldsymbol{Y}_s = \{\boldsymbol{y}(\sigma) | \sigma \text{ is a descendant of } s\} \tag{A.4}$$
$$\hat{\boldsymbol{x}}(\sigma|s) = E\left[\boldsymbol{x}(\sigma) | \sigma \in \boldsymbol{Y}_s \cup \boldsymbol{y}(s)\right] \tag{A.5}$$
$$\hat{\boldsymbol{x}}(\sigma|s+) = E\left[\boldsymbol{x}(\sigma) | \sigma \in \boldsymbol{Y}_s\right] \tag{A.6}$$
$$\tilde{\boldsymbol{P}}(\sigma|s) = \mathrm{Cov}\left[\boldsymbol{x}(\sigma) - \hat{\boldsymbol{x}}(\sigma|s)\right] \tag{A.7}$$
$$\tilde{\boldsymbol{P}}(\sigma|s+) = \mathrm{Cov}\left[\boldsymbol{x}(\sigma) - \hat{\boldsymbol{x}}(\sigma|s+)\right] \tag{A.8}$$

The upward sweep initializes at the finest level from the prior covariances:

$$\hat{\boldsymbol{x}}(s|s+) = 0 \tag{A.9}$$
$$\boldsymbol{P}(s|s+) = \boldsymbol{P}(s) \tag{A.10}$$

It requires the following upward model, corresponding to the the downward model in (A.1),

$$\boldsymbol{x}(s\bar{\gamma}) = \boldsymbol{F}(s)\boldsymbol{x}(s) + \bar{\boldsymbol{w}}(s) \tag{A.11}$$
$$\boldsymbol{y}(s) = \boldsymbol{C}(s)\boldsymbol{x}(s) + \boldsymbol{v}(s) \tag{A.12}$$

where

$$\boldsymbol{F}(s) = \boldsymbol{P}(s\bar{\gamma})\boldsymbol{A}^T(s)\boldsymbol{P}(s)^{-1} \tag{A.13}$$

$$E\left[\bar{\boldsymbol{w}}(s)\bar{\boldsymbol{w}}(s)^T\right] = \boldsymbol{P}(s\bar{\gamma}) - \boldsymbol{P}(s\bar{\gamma})\boldsymbol{A}^T(s)\boldsymbol{P}(s)^{-1}\boldsymbol{A}(s)\boldsymbol{P}(s\bar{\gamma}) = \boldsymbol{Q}(s) \tag{A.14}$$

The upward sweep computes the best estimate of the states at a node given all measurement below that node. It consists of three steps at each scale:
a) Update step:

$$\hat{\boldsymbol{x}}(s|s) = \hat{\boldsymbol{x}}(s|s+) + \boldsymbol{K}(s)\left[\boldsymbol{y}(s) - \boldsymbol{C}(s)\hat{\boldsymbol{x}}(s|s+)\right] \tag{A.15}$$

$$\boldsymbol{P}(s|s) = \left[\boldsymbol{I} - \boldsymbol{K}(s)\boldsymbol{C}(s)\right]\boldsymbol{P}(s|s+) \tag{A.16}$$

$$\boldsymbol{K}(s) = \boldsymbol{P}(s|s+)\boldsymbol{C}^T(s)\left[\boldsymbol{C}(s)\boldsymbol{P}(s|s+)\boldsymbol{C}^T(s) + \boldsymbol{R}(s)\right]^{-1} \tag{A.17}$$

b) Prediction step:

$$\hat{\boldsymbol{x}}(s|s\alpha_i) = \boldsymbol{F}(s\alpha_i)\hat{\boldsymbol{x}}(s\alpha_i|s\alpha_i) \tag{A.18}$$

$$\boldsymbol{P}(s|s\alpha_i) = \boldsymbol{F}(s\alpha_i)\boldsymbol{P}(s\alpha_i|s\alpha_i)\boldsymbol{F}^T(s\alpha_i) + \boldsymbol{Q}(s\alpha_i) \tag{A.19}$$

c) Merge step:

$$\hat{\boldsymbol{x}}(s|s+) = \boldsymbol{P}(s|s+)\sum_{i=1}^{q}\boldsymbol{P}^{-1}(s|s\alpha_i)\hat{\boldsymbol{x}}(s|s\alpha_i) \tag{A.20}$$

$$\boldsymbol{P}(s|s+) = \left[(1-q)\boldsymbol{P}(s)^{-1} + \sum_{i=1}^{q}\boldsymbol{P}^{-1}(s|s\alpha_i)\right]^{-1} \tag{A.21}$$

The downward sweep computes the best estimate of the states at a node given all available measurements:

$$\hat{\boldsymbol{x}}(s|0) = \hat{\boldsymbol{x}}(s|s) + \boldsymbol{J}(s)\left[\hat{\boldsymbol{x}}(s\bar{\gamma}|0) - \hat{\boldsymbol{x}}(s\bar{\gamma}|s)\right] \tag{A.22}$$

$$\boldsymbol{P}(s|0) = \boldsymbol{P}(s|s) + \boldsymbol{J}(s)\left[\boldsymbol{P}(s\bar{\gamma}|0) - \boldsymbol{P}(s\bar{\gamma}|s)\right]\boldsymbol{J}^T(s) \tag{A.23}$$

$$\boldsymbol{J}(s) = \boldsymbol{P}(s|s)\boldsymbol{F}^T(s)\boldsymbol{P}^{-1}(s\bar{\gamma}|s) \tag{A.24}$$

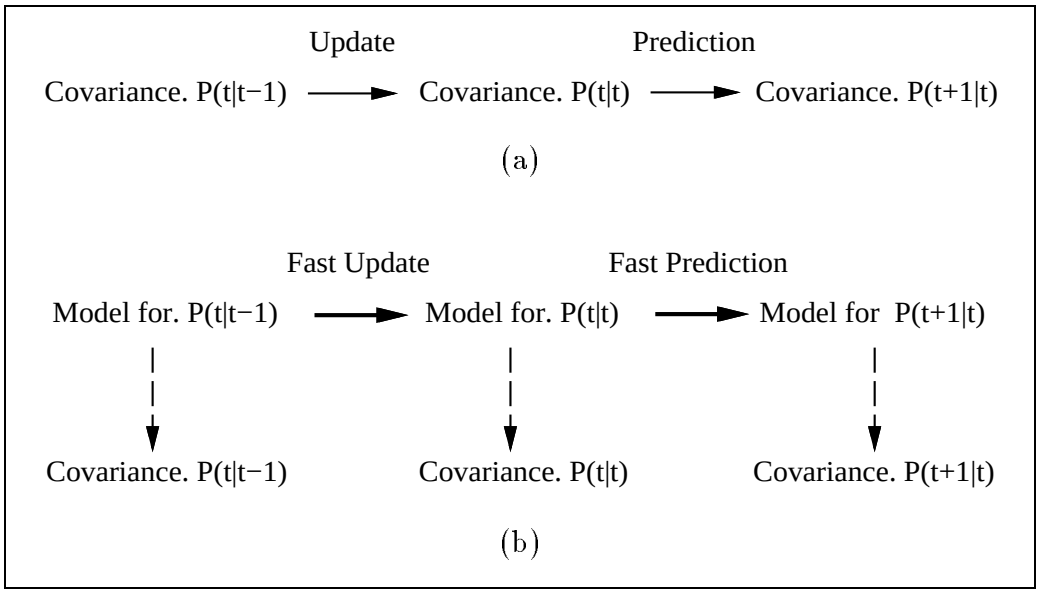
The smoothing error can be modeled as

$$\tilde{\boldsymbol{x}}(s|0) = \boldsymbol{J}(s)\tilde{\boldsymbol{x}}(s\bar{\gamma}|0) + \check{\boldsymbol{w}}(s) \tag{A.25}$$

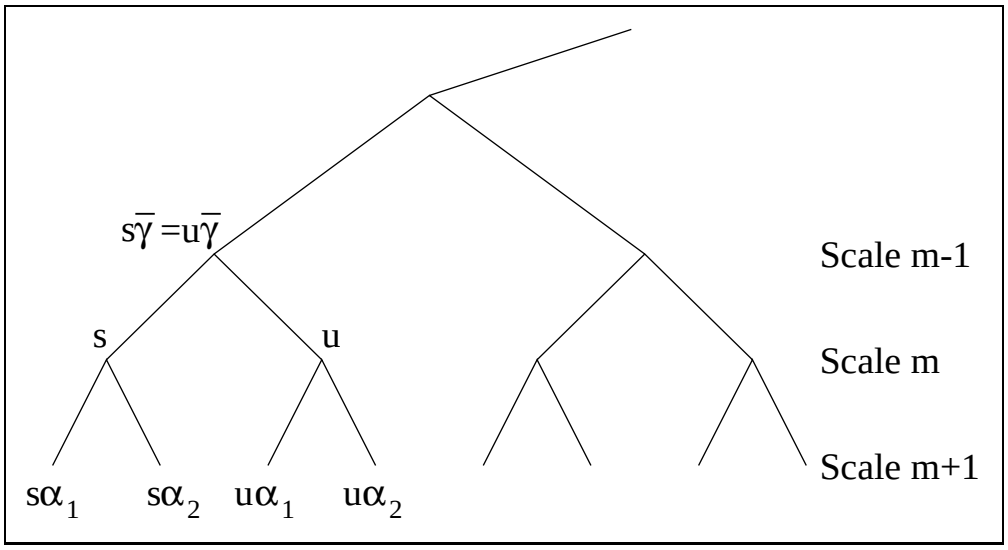
where $\tilde{\boldsymbol{x}}(s|0) = \boldsymbol{x}(s) - \hat{\boldsymbol{x}}(s|0)$, and

$$E\left[\check{\boldsymbol{w}}(s)\check{\boldsymbol{w}}(s)^T\right] = \boldsymbol{P}(s|s) - \boldsymbol{P}(s|s)\boldsymbol{F}^T(s)\boldsymbol{P}^{-1}(s\bar{\gamma}|s)\boldsymbol{F}(s)\boldsymbol{P}(s|s) \tag{A.26}$$

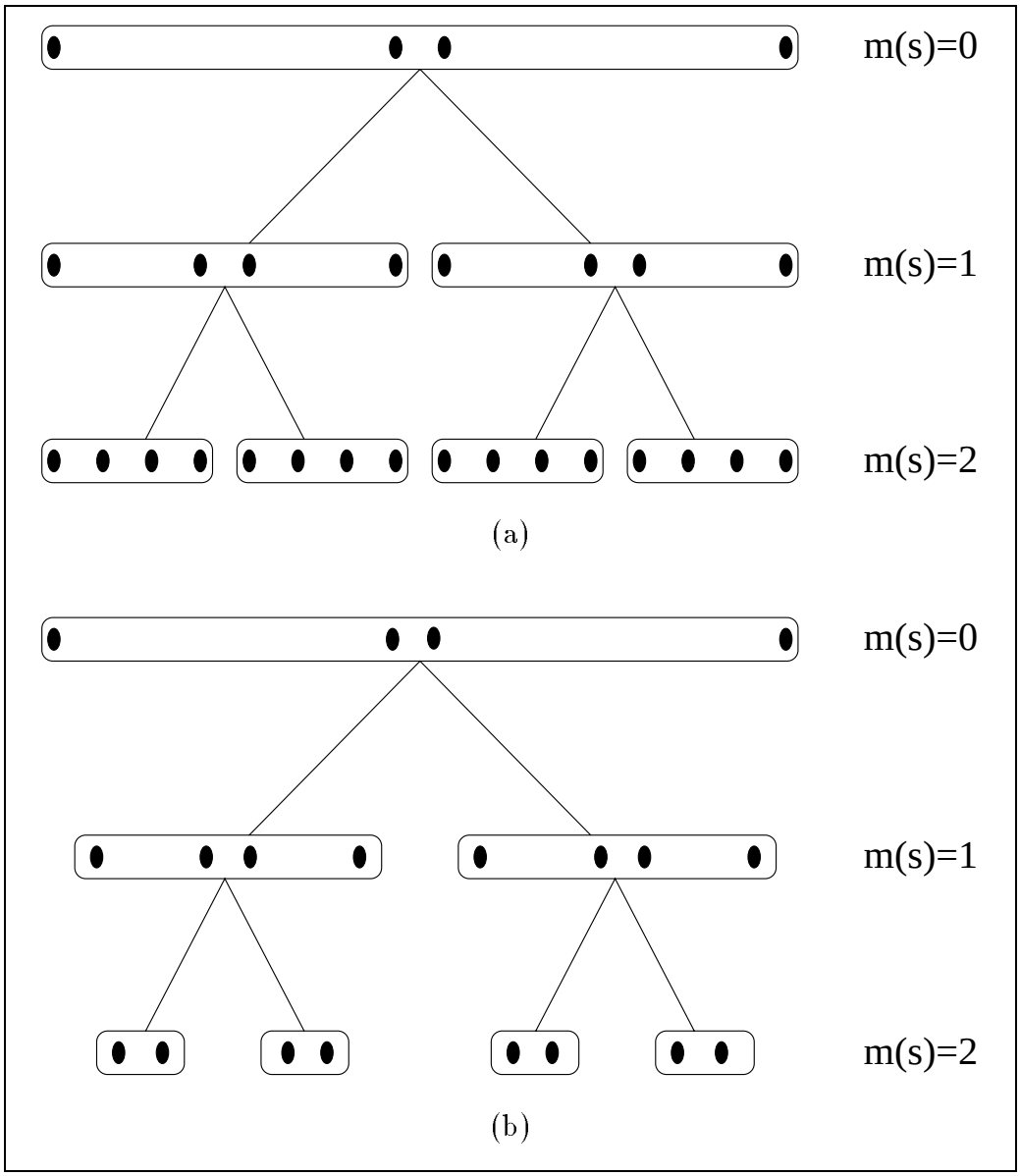
Note that the state covariances at individual nodes of the smoothing error model have already been computed in (A.24) and (A.26).



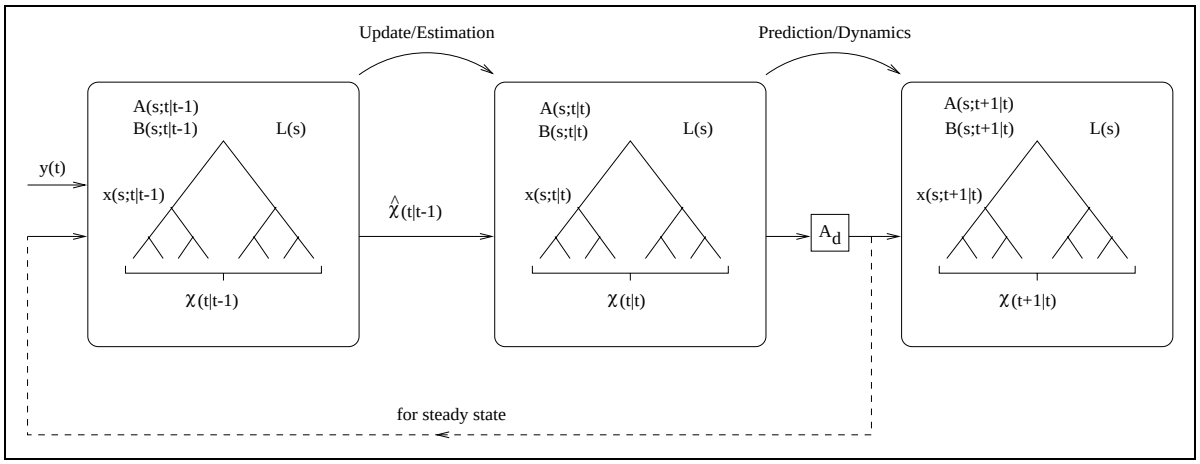
(Fig. 1 — Ho, Fieguth, Willsky)



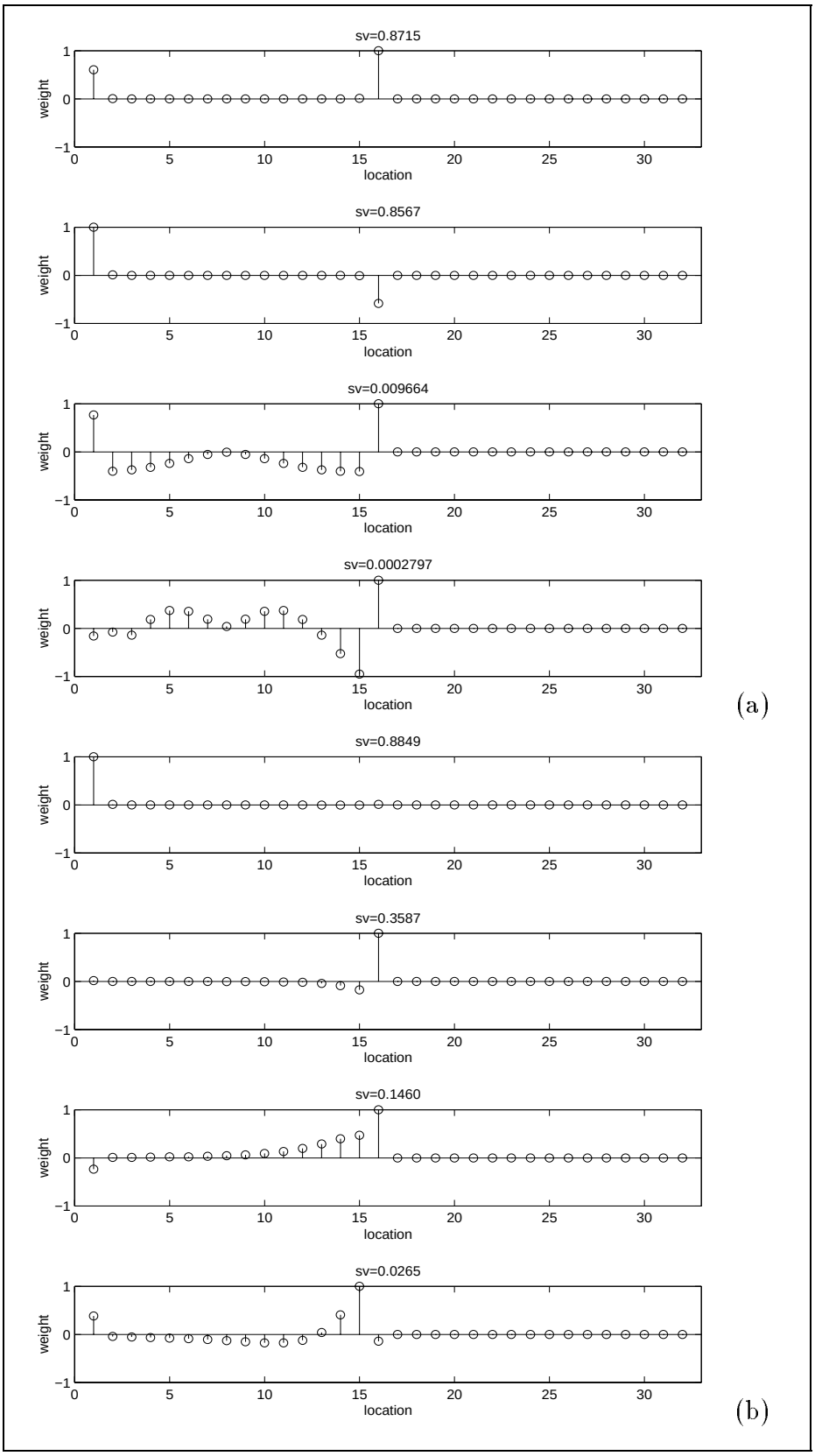
(Fig. 2 — Ho, Fieguth, Willsky)



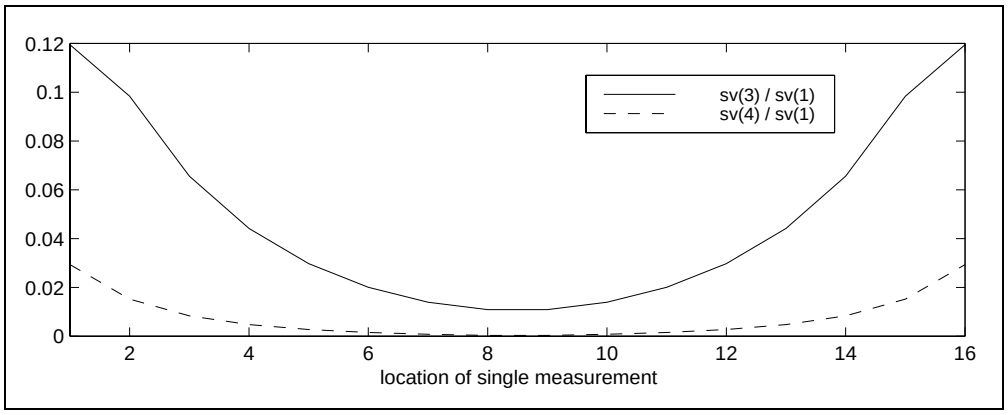
(Fig. 3 — Ho, Fieguth, Willsky)



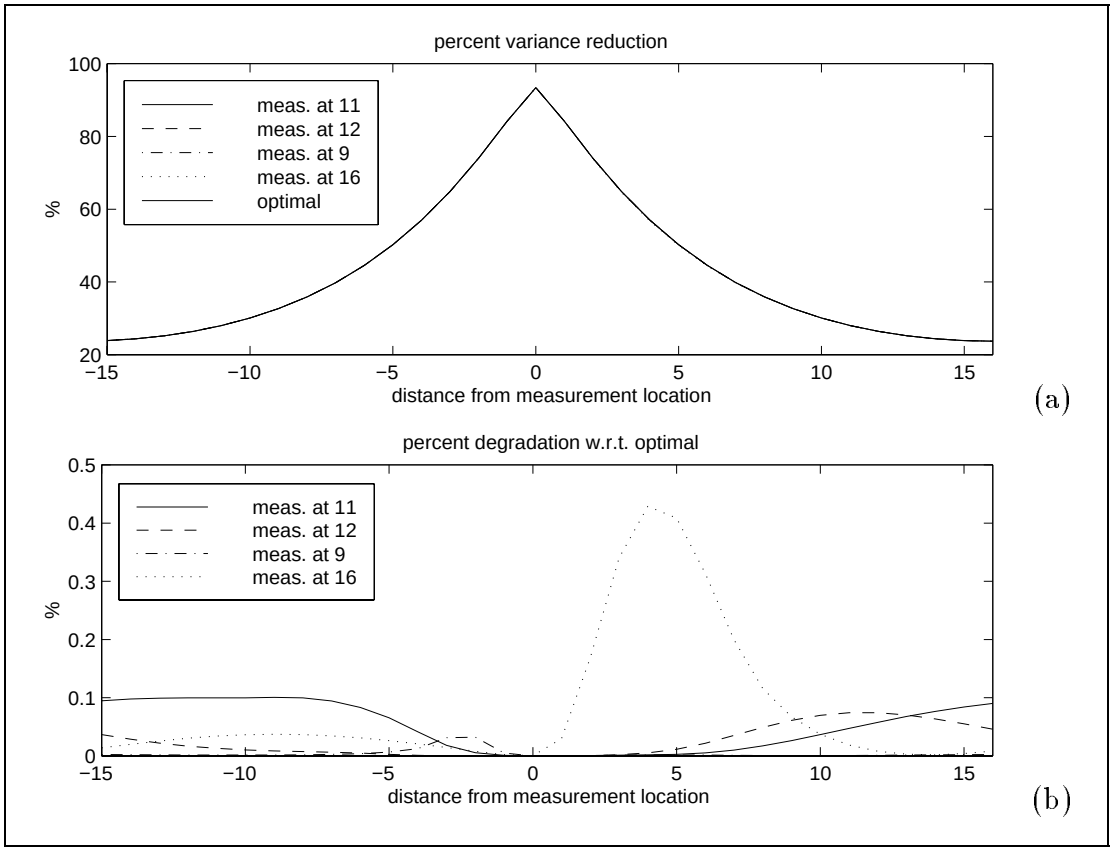
(Fig. 4 — Ho, Fieguth, Willsky)



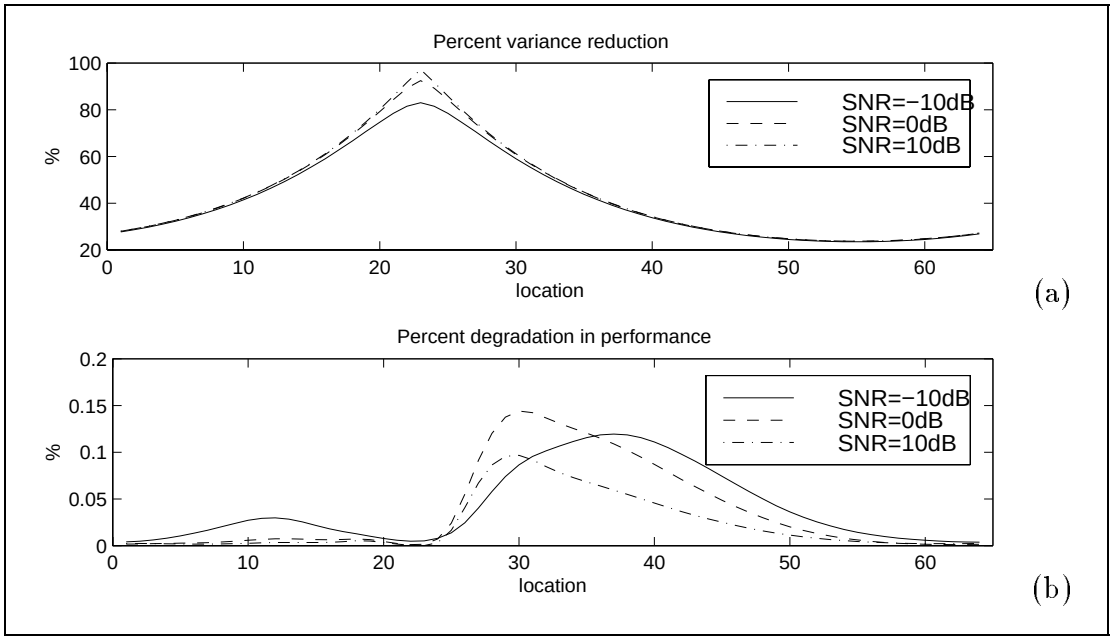
(Fig. 5 — Ho, Fieguth, Willsky)



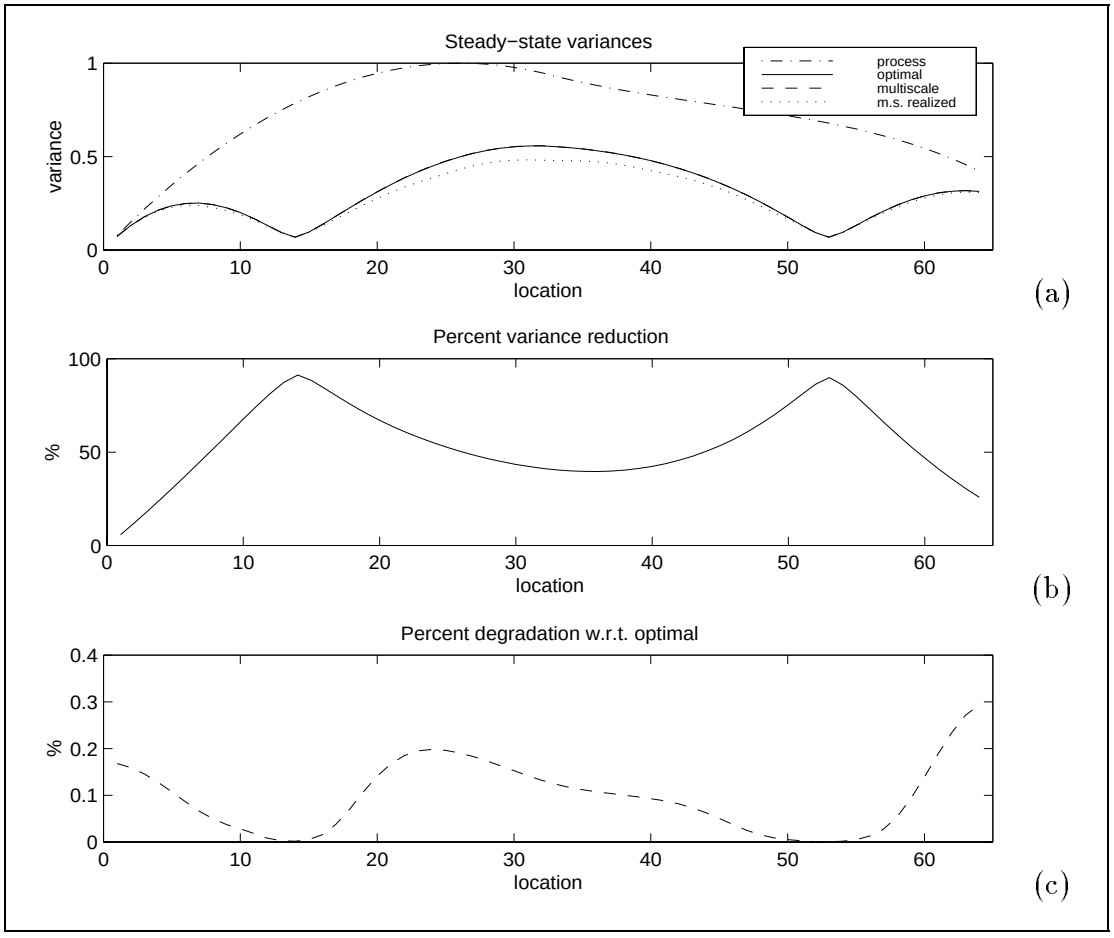
(Fig. 6 — Ho, Fieguth, Willsky)



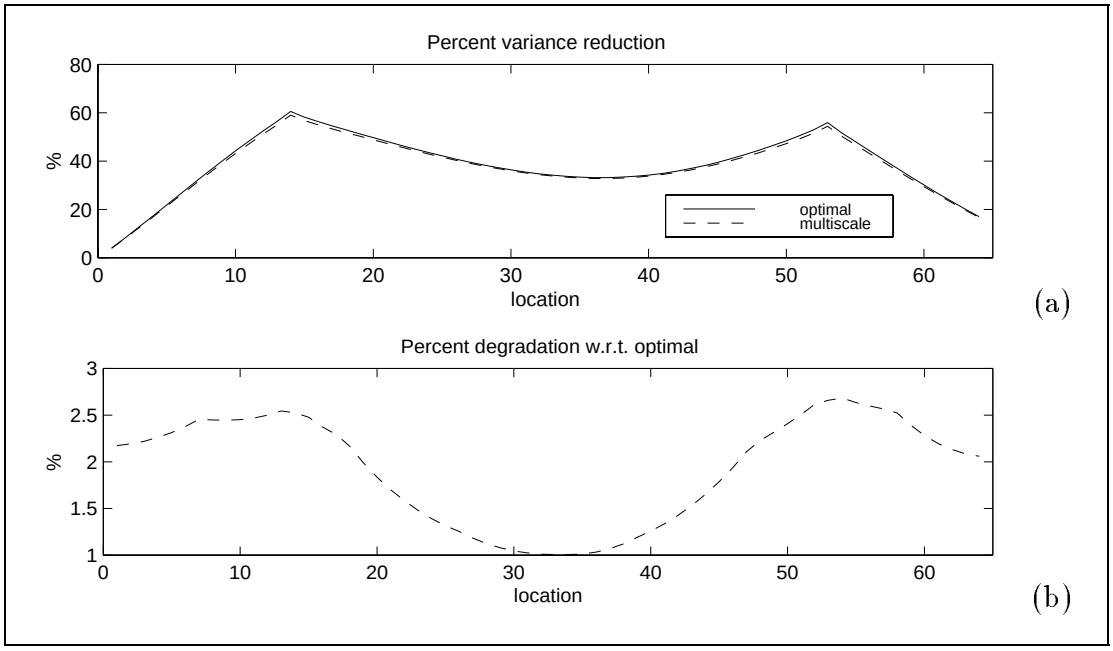
(Fig. 7 — Ho, Fieguth, Willsky)



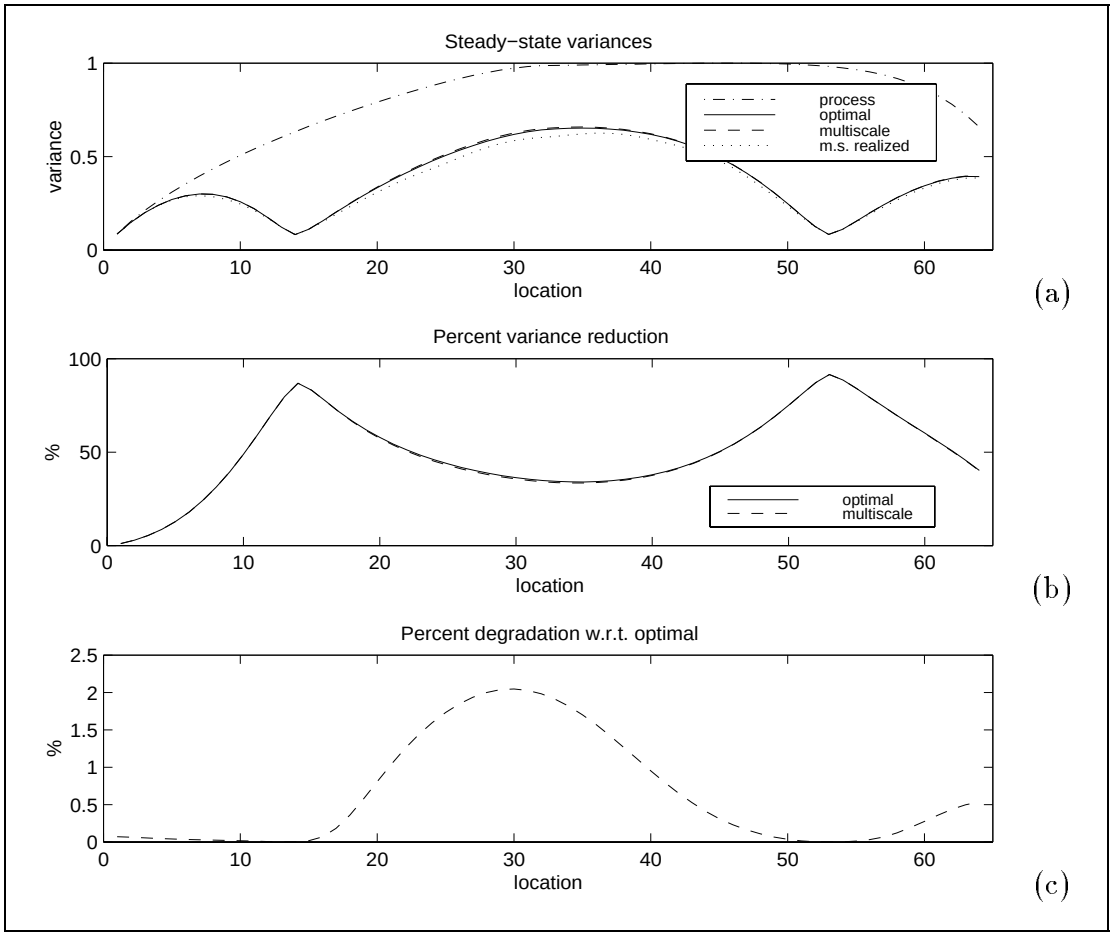
(Fig. 8 — Ho, Fieguth, Willsky)



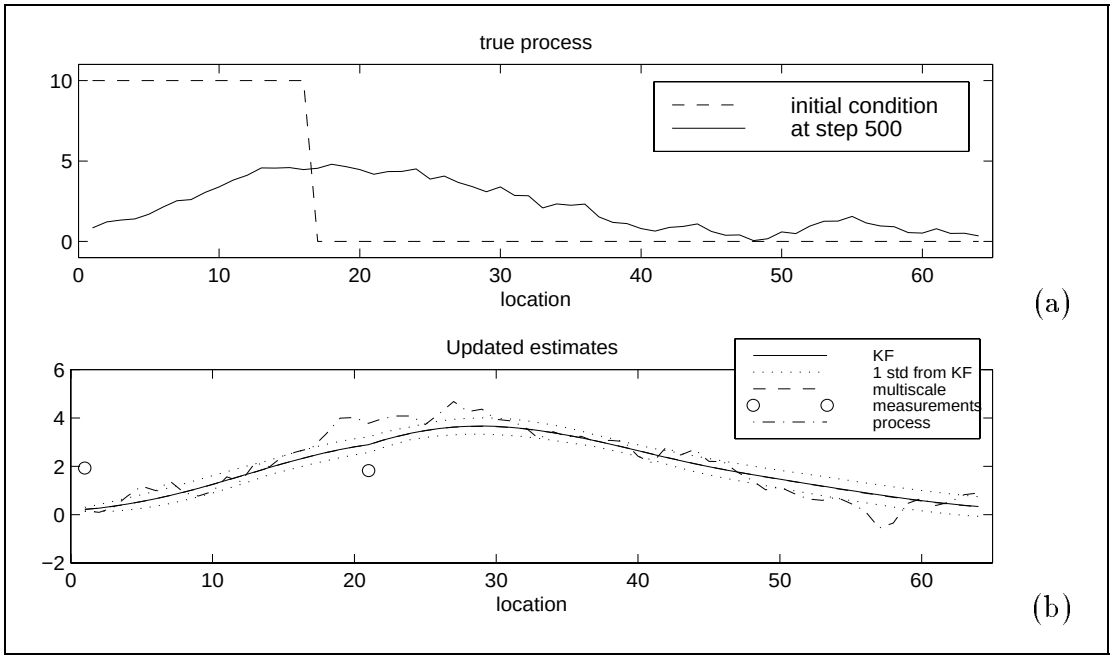
(Fig. 9 — Ho, Fieguth, Willsky)



(Fig. 10 — Ho, Fieguth, Willsky)



(Fig. 11 — Ho, Fieguth, Willsky)



(Fig. 12 — Ho, Fieguth, Willsky)

Figure Captions

Fig. 1. Two possible sequences of steps for dynamic estimation: (a) the standard Kalman filter, in which covariance matrices are propagated; (b) proposed alternative, in which *models* are propagated.

Fig. 2. A portion of a dyadic multiscale tree.

Fig. 3. Two possible 1-D Markov process realizations: (a) standard model, (b) proposed non-redundant model.

Fig. 4. A schematic of the proposed multiscale iterative method for dynamic estimation, modeled on the Kalman filter.

Fig. 5. The four most significant the linear functionals that decorrelate the steady-state predicted estimation errors at points $1-16$ from those at points $17-32$ of a 32-element diffusion process ($\beta = 10$, SNR = 0dB, $\Delta\tau = 2\times 10^{-5}$). The measurement is at pixel 8 in (a), and at pixel 16 in (b). The singular value is printed above each associated linear functional.

Fig. 6. The singular values of the third and fourth most significant linear functionals as a function of measurement location.

Fig. 7. (a) The percent variance reduction of the optimal estimator and of the multiscale estimator in steady state. (b) Percent degradation of the multiscale estimator with respect to the exact solution. ($\beta = 10$, SNR = 0dB, $\Delta\tau = 2 \times 10^{-5}$).

Fig. 8. (a) Percent variance reduction of the optimal estimator with one measurement at location 23 for $\beta = 10$ and SNR $= -10$, 0, and 10 dB. (b) Percent performance degradation with respect to optimal.

Fig. 9. Pinned fin with two measurements at locations 14 and 53. Heat loss parameter $\beta = 0$ at locations $1-32$ and $\beta = 10$ at $33-64$. (a) Steady-state process variances, steady-state estimation error variances, and the realized variances at the finest scale of the suboptimal multiscale estimator. (b) Percent variance reduction of the steady-state optimal estimators. (SNR = 0 dB) (c) Percent performance degradation of the multiscale estimator with respect to optimal.

Fig. 10. As in Fig. 10, but with one measurement update every 200 prediction steps. (a) Percent variance reduction of the steady-state estimators. (b) Percent performance degradation of the multiscale estimator with respect to the optimal.

Fig. 11. Liquid flow ($\rho = -10$) from a reservoir (location 1) through a thin

pipe, half insulated ($\beta = 0$ at locations $1 - 32$) and half exposed ($\beta = 10$ at $33 - 64$). Measurements at locations 14 and 53. SNR = 0 dB. (a) Steady-state process variances, steady-state estimation error variances, realized variances at the finest scale of the multiscale estimator. (b) Percent variance reduction of the steady-state estimators. (c) Percent performance degradation of the multiscale estimator with respect to the optimal.

Fig. 12. Cooling pipe. One update step for every 100 prediction steps. (SNR = 0 dB.) (a) True process at step 1300 and initial values of the process. (b) Updated estimates at step 1300. The optimal and the multiscale suboptimal estimates are indistinguishable at this resolution. The locations and values of the two measurements at this update step are labeled with circles. Dotted curves show the range of one standard deviation from the optimal estimates.