

**September, 1998**

LIDS- TH 2428

**Research Supported By:**

NSF Graduate Fellowship

Multiscale Modelling and Estimation of Large-Scale Dynamic  
Systems

Ho, T .T-J.



**Multiscale Modeling and Estimation of  
Large-Scale Dynamic Systems**

by

Terrence Tian-Jian Ho

S.M., Electrical Engineering and Computer Science, MIT (1994)

B.E., Electrical Engineering, The Cooper Union (1992)

Submitted to the Department of Electrical Engineering and Computer  
Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1998

© Massachusetts Institute of Technology 1998. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
August 7, 1998

Certified by .....  
Alan S. Willsky  
Professor of Electrical Engineering  
Thesis Supervisor

Accepted by .....  
Arthur C. Smith  
Chair, Departmental Committee on Graduate Students



# Multiscale Modeling and Estimation of Large-Scale Dynamic Systems

by

Terrence Tian-Jian Ho

Submitted to the Department of Electrical Engineering and Computer Science  
on August 7, 1998, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Electrical Engineering

## Abstract

Statistical modeling and estimation of large-scale dynamic systems is important in a wide range of scientific applications. Conventional optimal estimation methods, however, are impractical due to their computational complexity. In this thesis, we consider an alternative multiscale modeling framework first developed by Basseville, Willsky, et al. [6, 18].

This multiscale estimation methodology has been successfully applied to a number of large-scale *static* estimation problems, one of which is the application of the so-called  $1/f$  multiscale models to the mapping of ocean surface height from satellite altimetric measurements. A modified  $1/f$  model is used in this thesis to jointly estimate the surface height of the Mediterranean Sea *and* the correlated component of the measurement noise in order to remove the artifacts apparent in maps generated with the more simplistic assumption that the measurement noise is white.

The main contribution of this thesis is the extension of the multiscale framework to *dynamic* estimation. We introduce a recursive procedure that propagates a multiscale *model* for the estimation errors in a manner analogous to, but more efficient than, the Kalman filter's propagation of the error covariances. With appropriately chosen multiscale models, such as the new class of *non-redundant* models that we introduce, the computational gain can be substantial. We use 1-D and 2-D diffusion processes to illustrate the development of our algorithm. The resulting multiscale estimators achieve  $\mathcal{O}(N)$  computational complexity with near-optimal performance in 1-D and  $\mathcal{O}(N^{3/2})$  in 2-D, as compared to the  $\mathcal{O}(N^3)$  complexity of the standard Kalman filter.

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



# Acknowledgments

I wish to express my deepest gratitude to my thesis advisor, Prof. Alan Willsky. This thesis would not have been possible without the many great ideas he so freely and generously gave out. His intellectual prowess and his energy have been inspirational. Alan allows his student wide latitude to explore interesting research ideas, yet pays close attention to keep progress on track. His gentle encouragement and firm guidance have been instrumental in the completion of this work.

I would like to thank Prof. Paul Fieguth<sup>1</sup> for his help throughout my Ph.D. program and for serving on my thesis committee. I benefited a great deal from observing, at close range, a sharp mind at work and at play.

I would like to thank Dr. Dimitris Menemenlis and Paul for our work on the oceanography application in Chapter 6. Paul and Dimitris generously allowed me to claim a lot of their work as my own.

I would like to thank Prof. Carl Wunsch and Prof. Sanjoy Mitter for valuable discussions and for graciously serving on my thesis committee.

I would like to thank the American taxpayers for their generous financial support through three years of National Science Foundation Graduate Fellowship and three years of research assistantships underwritten by the Army. After twenty-one years in school, although without paying a cent in tuition, I am finally ready to earn my keep and repay my debt to society — in part by striving to put myself into a higher tax bracket.

My Ph.D. experience would have been less satisfying had I not had the company of the many wonderful SSG groupies, past and present. The number of good people I shall mention here bears testimony to how long it has taken me to finish my thesis (and/or how prosperous SSG has been). I would like to thank Cedric for his daily dose of insanity checks, Çedric-ism<sup>2</sup>, major contribution to the SSG Manifesto Board,

---

<sup>1</sup>formerly Dr. Paul, and more formerly just Paul

<sup>2</sup>Çedric-ism example 1: “Will it ever end?” “No.” Example 2: “Big, fat, bloated, blood sucking ticks on the back of the \$\$\$ ca\$h cow.”

and  $2 \times K_2O^3$ . I would like to thank Mike S. for our insightful discussions on, among a great variety of topics, cookie recipes, for our successful attempts at the asexual propagation of *Chlorophytum*<sup>4</sup>, and for our interesting experiment with a ChiaCat. I would like to thank Paul, now Prof. Paul, for delaying the implementation of his *Manifesto on Technology*<sup>5</sup>. I would like to thank Andy  $\Psi$ <sup>6</sup> for allowing me to pick on him, but like what Dewey would say, “Yeah, whatever.” I would like to thank AJ for giving me grief about T. Rex’ing the computers in ingenious ways from time to time. I would like to thank Frakt for many trips to the brickyard<sup>7</sup>. I would like to thank Ilya for his fine taste in chocolate and ballet. I would like to thank John-John, Rachel, Seema, Mike D., Charlie, and Bill for being far too well-adjusted and all too married to be poked fun here. To be in the company of such capable a group of people is truly a humbling experience.

My Ph.D. years in the larger context of my MIT experience had been influenced by my other personal friends and two in particular. I would like to thank Ed for his friendship and for being there with me whether it was 100 feet under the Caribbean or  $54^\circ$  under the equator. I would like to thank Sunil for his friendship, for being brave enough to step out of a Cessna at 3000 feet with me, twice, and for being even more brave to agree to sit in the right seat of my Katana.

I thank my parents for their love, support, and understanding through these years. I hope they may find some pride and joy in whatever meager achievements I have had.

With the completion of this thesis I am again at a crossroad in my life. I look back with a sense of accomplishment and pride mixed with a sense of closure and renaissance. I think of the many people who came into my life — some by design and some by accident — people who shaped and influenced who I am today. For better or for worse, it’s been an experience.

---

<sup>3</sup>Krank<sub>2</sub>O, substituted variously with Surge, Jolt, Storm, Mountain Dew, or Dr. Pepper.

<sup>4</sup>spiderplant

<sup>5</sup>The resemblance extends to facial features.

<sup>6</sup>a.k.a. P.K.

<sup>7</sup>brick, *n.*, Styrofoam box lunch from food trucks, a term coined by Mike D.

*To My Parents,  
Whose Love and Support  
Saw Me Through It All*



# Contents

<b>1</b>	<b>Introduction</b>	<b>23</b>
1.1	Problems Addressed and Contributions . . . . .	24
1.1.1	Multiscale Recursive Estimation Algorithm . . . . .	24
1.1.2	Non-Redundant Models . . . . .	27
1.1.3	Large-Scale Data Assimilation . . . . .	28
1.2	Thesis Organization . . . . .	29
<b>2</b>	<b>Fundamentals of Multiscale Modeling and Estimation</b>	<b>33</b>
2.1	Multiscale Stochastic Models . . . . .	34
2.2	Multiscale Algorithms . . . . .	38
2.2.1	Multiscale Smoothing Algorithm . . . . .	38
2.2.2	Smoothing Error Model . . . . .	40
2.2.3	Multiscale Likelihood Test . . . . .	41
2.3	Multiscale Realization . . . . .	42
2.3.1	Internal Realization Models . . . . .	43
2.3.2	Internal Models for MRFs . . . . .	45
2.3.3	Canonical Correlations Realization . . . . .	51
2.3.4	Conditional Decorrelation and Model Consistency . . . . .	54
2.3.5	External Models . . . . .	57
2.4	Concluding Remarks . . . . .	60
<b>3</b>	<b>Multiscale Time-Recursive Estimation</b>	<b>63</b>
3.1	Introduction . . . . .	63

3.2	Recursive Estimation of Dynamic Systems . . . . .	66
3.2.1	MRF-Based Recursive Estimation . . . . .	70
3.3	Multiscale Recursive Estimation . . . . .	72
3.3.1	Multiscale Prediction Step . . . . .	73
3.3.2	Model Parameters of the Predicted Error Model . . . . .	76
3.3.3	Choice of Linear Functionals . . . . .	79
3.4	Examples of Multiscale Prediction in 1-D . . . . .	81
3.4.1	End-Point Linear Functionals and Diagonal $\mathbf{A}_d$ . . . . .	81
3.4.2	End-Point Linear Functionals and Tridiagonal $\mathbf{A}_d$ . . . . .	83
3.5	Non-Redundant Models . . . . .	87
3.5.1	Non-Redundant Models for MRFs . . . . .	87
3.5.2	Multiscale Prediction in 1-D with Non-Redundant Models and Tridiagonal $\mathbf{A}_d$ . . . . .	90
3.5.3	Higher-Order Non-Redundant Models . . . . .	91
3.6	Performance Measure . . . . .	93
3.6.1	Convergence to Steady State . . . . .	93
3.6.2	Fractional Variance Reduction . . . . .	96
<b>4</b>	<b>Multiscale Dynamic Estimation of 1-D Diffusion</b>	<b>99</b>
4.1	Introduction to 1-D Diffusion . . . . .	100
4.1.1	Specifications of the Process . . . . .	100
4.1.2	Specifications of the Measurements . . . . .	104
4.2	Linear Functionals for 1-D Diffusion . . . . .	104
4.2.1	Functionals for the Steady-State Process . . . . .	105
4.2.2	Functionals for the Steady-State Prediction Error . . . . .	112
4.3	Multiscale Recursive Estimation Algorithm for 1-D Diffusion . . . . .	122
4.3.1	Multiscale Prediction Step for 1-D Diffusion . . . . .	122
4.3.2	Iterative and Recursive Implementations . . . . .	124
4.4	Examples and Results . . . . .	126
4.4.1	Cooling Ring . . . . .	126

4.4.2	Pinned Fin . . . . .	128
4.4.3	Advection-Diffusion . . . . .	131
4.4.4	Recursive Implementation and Temporally Nonstationary Performance . . . . .	133
<b>5</b>	<b>Multiscale Dynamic Estimation of 2-D Diffusion</b>	<b>137</b>
5.1	Challenges in Modeling 2-D Random Fields . . . . .	137
5.2	Linear Functionals for 2-D Diffusion . . . . .	139
5.2.1	Analysis via CCR . . . . .	141
5.3	Multiscale Realizations of the Exact Steady-State Prediction Error . .	146
5.3.1	Internal Models with Linear Functionals by CCR . . . . .	147
5.3.2	Internal Models with Boundary-Point Functionals . . . . .	152
5.3.3	Non-Redundant Models . . . . .	157
5.4	Multiscale Recursive Estimation Algorithm for 2-D Diffusion . . . . .	169
5.4.1	Multiscale Prediction Step on Internal Models . . . . .	170
5.4.2	Multiscale Prediction Step on Non-Redundant Models . . . . .	173
5.5	Examples and Results . . . . .	177
5.5.1	Iteratively Implemented Steady-State Estimators . . . . .	178
5.5.2	Recursive Implementation . . . . .	188
5.6	Concluding Remarks . . . . .	193
<b>6</b>	<b>Mapping Mediterranean Altimeter Data</b>	<b>195</b>
6.1	Altimetric Data Assimilation . . . . .	197
6.2	Multiscale Modeling of the Anomaly Signals . . . . .	199
6.2.1	Statistical Models of Altimeter Data . . . . .	199
6.2.2	Multiscale Models . . . . .	201
6.3	Experimental Results . . . . .	204
6.3.1	Prior Statistics . . . . .	205
6.3.2	Basic Interpolation . . . . .	210
6.3.3	Multiscale — No Correlated Error Model . . . . .	212
6.3.4	Multiscale — Correlated Error Model . . . . .	212

6.3.5	Suboptimal Space-Time Interpolation . . . . .	215
6.3.6	Multiscale — Joint TOPEX/POSEIDON–ERS-1 Estimation .	218
6.4	Concluding Remarks . . . . .	220
<b>7</b>	<b>Conclusion</b>	<b>223</b>
7.1	Summary of Contributions . . . . .	223
7.2	Open Problems and Future Directions . . . . .	225
7.2.1	Multiscale Prediction . . . . .	225
7.2.2	Multiscale Modeling . . . . .	227
7.2.3	Optical Flow and Hybrid Models . . . . .	230
<b>A</b>	<b>Multiscale Smoothing Algorithms</b>	<b>239</b>
<b>B</b>	<b>MATLAB Programs</b>	<b>243</b>
B.1	Data Structure . . . . .	243
B.2	Sea Level Anomaly Interpolation Code . . . . .	244
	<b>Bibliography</b>	<b>246</b>

# List of Figures

2-1	A portion of a dyadic multiscale tree. . . . .	35
2-2	An example quad tree for modeling 2-D random fields. . . . .	36
2-3	Rauch-Tung-Striebel smoothing algorithm on the tree. . . . .	39
2-4	The linear functionals of an internal realization of a first-order 1-D Markov random field on a dyadic tree with a state dimension of three. . . . .	48
2-5	The end-point linear functionals of an internal realization of a first-order 1-D Markov random field on a dyadic tree with a state dimension of four. . . . .	49
2-6	Boundary points kept as states at the root node and at the four nodes at scale 1 of an internal realization of a $16 \times 16$ first-order 2-D Markov random field. . . . .	50
2-7	An example tree that does not satisfy the conditional decorrelation requirement. . . . .	55
2-8	An example tree that does not satisfy the consistency requirement. . . . .	55
3-1	A schematic of the multiscale recursive estimation method. . . . .	72
3-2	An example tree model using end-point linear functionals that can realize non-MRFs exactly at the finest scale. . . . .	82
3-3	The cross-covariances needed in the internal model with end-point linear functionals for the updated estimation errors for multiscale prediction. . . . .	85
3-4	The entries of $\mathbf{P}_\chi(t t)$ that must be filled for multiscale prediction with end-point linear functionals. . . . .	86

3-5	The linear functionals of a non-redundant model (with state dimension $k = 3$ ) for a first-order 1-D Markov random field. . . . .	88
3-6	The cross-covariances needed in the non-redundant model for the updated estimation errors for multiscale prediction. . . . .	90
3-7	The linear functionals of a non-redundant model with state dimension $k = 4$ for a first-order 1-D Markov random field. . . . .	91
3-8	A non-redundant model with state dimension $k = 5$ for a pentagonal dynamics matrix $\mathbf{A}_d$ . . . . .	92
3-9	A non-redundant model with state dimension $k = 6$ for a second-order 1-D Markov random field. . . . .	92
4-1	A comparison of (a) the exact steady-state process covariance of a 64-element discretized diffusion process on a ring with (b) a first-order MRF approximation of the covariance, (c) a second-order MRF approximation, and (d) the covariance realized by an internal model with end-point linear functionals. . . . .	108
4-2	Two of the linear functionals that maximally decorrelate elements $\{1 - 32\}$ from elements $\{33 - 64\}$ of a 64-element diffusion process on a ring in steady state. . . . .	109
4-3	A comparison of (a) the exact steady-state prediction error covariance $\tilde{\mathbf{P}}_p$ of a 64-element discretized diffusion process on a ring with (b) a first-order MRF approximation of the covariance, (c) a second-order MRF approximation, and (d) the covariance realized by an internal model with end-point linear functionals. . . . .	113
4-4	The four most significant linear functionals generated by CCR that decorrelate the steady-state predicted estimation errors at locations $\{1 - 32\}$ from those at locations $\{33 - 64\}$ of a 64-element diffusion process on a ring with a single measurement at location 16. . . . .	114

4-5	The four most significant linear functionals generated by CCR that decorrelate the steady-state predicted estimation errors at locations $\{1 - 32\}$ from those at locations $\{33 - 64\}$ of a 64-element diffusion process on a ring with a single measurement at location 32. . . . .	115
4-6	The relative significance of the third and fourth most significant linear functionals as a function of the location of the single measurement. . . . .	115
4-7	The Bhattacharyya distances between the optimal steady-state predicted error covariance $\tilde{\mathbf{P}}_p$ and the realized covariances from the four suboptimal multiscale models as a function of the measurement location. . . . .	117
4-8	(a) The FVR of the optimal estimator in steady state for a 64-element diffusion process on a ring with one point measurement at location 32. (b)–(c) The percent degradation in FVR of the suboptimal multiscale estimators with respect to the optimal estimator. . . . .	118
4-9	The maximum percent degradation in FVR of the suboptimal multiscale models with respect to the optimal estimator as a function of measurement location. . . . .	119
4-10	(a) The FVR of the optimal estimator in steady state for a 64-element diffusion process on a pinned fin with dense point measurements and non-uniform $\beta$ . (b) The percent degradation in FVR of the suboptimal estimators with respect to the optimal estimator. . . . .	120
4-11	(a) The FVR of the optimal estimator in steady state for a 64-element diffusion process on a pinned fin with two point measurements at locations 14 and 53. (b) The percent degradation in FVR of the suboptimal estimators with respect to the optimal estimator. . . . .	121
4-12	Cooling ring example for a 64-element diffusion process with one point measurement at location 23 with SNR = $-10$ dB, $0$ dB, and $10$ dB. . . . .	127
4-13	Pinned fin example with two measurements at locations 14 and 53 with spatially varying heat loss parameter. . . . .	129
4-14	Pinned fin example as in Figure 4-13 except that there is one measurement update step for every 200 prediction steps. . . . .	130

4-15	Thin cooling pipe example of a 64-element advection-diffusion process, in which a fluid flows from a reservoir through a thin pipe that is half insulated and half exposed. . . . .	132
4-16	Thin cooling pipe example as in Figure 4-15 except that there is one measurement update step for every 100 prediction steps and that the number and the location of the measurements are random. . . . .	134
5-1	The three most significant linear functionals chosen by CCR to decorrelate elements in the “northeast” quadrant, $(1 - 8, 1 - 8)$ , from the remaining elements of a $16 \times 16$ steady-state 2-D diffusion process on a toroid. . . . .	142
5-2	The fourth to the seventh most significant linear functionals chosen by CCR to decorrelate elements $(1 - 8, 1 - 8)$ from the remaining elements of a $16 \times 16$ steady-state 2-D diffusion process on a toroid. . . . .	143
5-3	Singular values associated with the 48 most significant linear functionals produced by CCR that maximally decorrelate elements $(1 - 8, 1 - 8)$ from the remaining elements of a $16 \times 16$ steady-state 2-D diffusion process on a toroid. . . . .	144
5-4	The two most significant linear functionals chosen by CCR to decorrelate elements $(1 - 8, 1 - 8)$ from the remaining elements of a $16 \times 16$ steady-state 2-D diffusion process on a sheet. . . . .	146
5-5	The four most significant linear functionals chosen by CCR to decorrelate elements $(1 - 8, 1 - 8)$ from the remaining elements of the $16 \times 16$ steady-state prediction error field of a 2-D diffusion process on a toroid.	147
5-6	(a) The FVR of the optimal steady-state estimator for a $16 \times 16$ 2-D diffusion process on a toroid with a single point measurement at $(6, 7)$ . (b)-(d) The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators with linear functionals selected by CCR. . . . .	151

5-7	The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using a full-order internal model with dense boundary-point linear functionals. . . . .	153
5-8	The percent degradations in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using reduced-order internal models with subsampled boundary-point linear functionals and with CCR linear functionals. . . . .	155
5-9	The percent degradations in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using reduced-order internal models (of lower state dimensions than those in Figure 5-8) with subsampled boundary-point linear functionals with CCR linear functionals. . . . .	156
5-10	The linear functionals of a non-redundant model for a first-order 2-D Markov random field. . . . .	157
5-11	(a) The FVR of the optimal steady-state estimator for a $17 \times 17$ 2-D diffusion process on a toroid with a single point measurement at (6, 7). (b) The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using a full-order non-redundant model. . . . .	158
5-12	(a)–(b) The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using reduced-order non-redundant models with linear interpolation. (c)–(f) The boundary-point functionals kept at the root node and one of the nodes at scale 1. . . . .	162
5-13	The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using reduced-order non-redundant models with statistical interpolation. . . . .	164
5-14	The linear functionals kept at the root node and at the offshoot nodes connected to the root of a non-redundant model for a $17 \times 17$ field. .	165

5-15	The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using reduced-order non-redundant models with offshoot subtrees. . . . .	168
5-16	The cross-covariances needed in (a) an reduced-order and (b) a full-order internal model with boundary-point linear functionals for the updated estimation errors for multiscale prediction. . . . .	171
5-17	The cross-covariances needed in a reduced-order non-redundant model for the updated estimation errors for multiscale prediction. . . . .	174
5-18	Performance of the steady-state multiscale estimators for a 2-D diffusion process on a toroid with one point measurement at (6, 7). . . . .	179
5-19	A comparison of the steady-state updated error variance of the optimal estimator with realized error variances of the suboptimal multiscale estimators shown in Figure 5-18. . . . .	182
5-20	(a) The steady-state process variance of a 2-D diffusion process on a cooling sheet. (b) The FVR of the steady-state optimal estimator with a single point measurement at (6, 7). . . . .	183
5-21	Performance of the steady-state multiscale estimators for a 2-D diffusion process on a pinned cooling sheet with one point measurement at (6, 7). . . . .	184
5-22	Performance of the steady-state multiscale estimators for a 2-D diffusion process on a toroid with four point measurements at (6, 7), (3, 12), (10, 5), and (11, 14). . . . .	186
5-23	Performance of the steady-state multiscale estimators for a 2-D diffusion process on a pinned cooling sheet with four point measurements at (6, 7), (3, 12), (10, 5), and (11, 14). . . . .	187
5-24	Performance of the steady-state multiscale estimators for a 2-D diffusion process on a pinned cooling sheet with four point measurements and only one measurement update step for every 100 prediction steps. . . . .	189

5-25	The recursive estimation of a 2-D diffusion process on a pinned cooling sheet with temporally nonstationary measurements and only one measurement update step for every 100 prediction steps. (a) The initial values of the process. (b) The measurement locations at step 900. . .	190
5-26	A comparison of the estimation results after the measurement update at step 900 for estimating a 2-D diffusion process on a pinned cooling sheet with temporally nonstationary measurements and one measurement update step for every 100 prediction steps. . . . .	191
5-27	A comparison of the estimation results after the prediction step at step 950 for estimating a 2-D diffusion process on a pinned cooling sheet with temporally nonstationary measurements and one measurement update step for every 100 prediction steps. . . . .	192
6-1	Optimal standard deviation for the measurement noise of the TOPEX, POSEIDON, and ERS-1 altimeters. . . . .	206
6-2	Optimal oceanographic model parameters of the measured sea level anomaly signal for TOPEX data in repeat cycles 18 through 161. . .	207
6-3	Optimal standard deviation of the bias and tilt coefficients, $(\alpha_0(i))$ and $(\alpha_1(i))$ in (6.6), for each TOPEX repeat cycle. The dashed lines show four-month averages. . . . .	209
6-4	Top: Mediterranean coverage of the TOPEX altimeter for the period May 9–19, 1993 (repeat cycle 24), with each data point gridded to the finest scale of the multiscale tree. Bottom: sea level anomaly map for this period created using a simple interpolation scheme. Contour intervals are 3 cm. . . . .	211
6-5	Multiscale estimates (top) and standard deviation of uncertainty (bottom) of sea level anomaly. They do not take correlated errors into account. . . . .	213

6-6	Multiscale estimates (top) and standard deviation of uncertainty (bottom) of sea level anomaly. A first-order polynomial model has been used to remove the correlated component of measurement error along each satellite track. . . . .	214
6-7	Maps of sea-level anomaly (top) and normalized mapping error variance as a percentage of the total signal variance (bottom) for a sub-optimal space-time optimal interpolation scheme. . . . .	216
6-8	Additional coverage provided by the ERS-1 altimeter for the period May 9–19, 1993 (repeat cycle 24), concurrent with that of Figures 6-4, 6-5, 6-6, and 6-7. . . . .	218
6-9	Multiscale estimates (top) and standard deviation of uncertainty (bottom) of sea level anomaly. The maps are computed from TOPEX and ERS-1 data using the same multiscale model as that used to create Figure 6-6. . . . .	219
7-1	A multiscale internal model for optical flow field along a 1-D contour.	235
7-2	A hybrid model that achieves the three objectives of easy prediction, easy measurement placement, and accurate statistical realization. . .	237

# List of Tables

4.1	The Bhattacharrya distances between the steady-state process covariance $\mathbf{P}_z$ of a 64-element diffusion process on a ring and the realized covariances $\bar{\mathbf{P}}_z$ from first- and second-order MRF approximations for different values of $\Delta\tau$ . . . . .	107
4.2	The Bhattacharrya distances between the steady-state process covariance $\mathbf{P}_z$ of a 64-element diffusion process on a ring and the realized covariances $\bar{\mathbf{P}}_z$ from various approximate models. . . . .	110
5.1	Asymptotic computational complexity of the multiscale prediction step for several types of multiscale models. . . . .	176
6.1	Sensitivity of the multiscale estimates and error statistics to the multiscale model parameters for TOPEX repeat cycle 24. . . . .	208
6.2	Prior model parameters for multiresolution estimates of TOPEX repeat cycle 24. . . . .	210



# Chapter 1

## Introduction

This thesis primarily addresses the topic of statistical modeling and efficient estimation methods for large-scale dynamic systems within the multiscale estimation framework first developed by Basseville, Willsky, et al. [6, 18]. Since the introduction of this multiscale estimation methodology about ten years ago, a rich body of research results has accumulated on the subject. Much of the earlier research dealt with algorithmic issues; that is, developing multiscale counterparts to the conventional linear least squares formulation, such as algorithms for computing the estimates [18], for modeling the *a posteriori* estimation error statistics [69], for likelihood tests [68], etc. The bulk of the research, however, dealt with modeling issues; that is, identifying classes of random processes that admit compact multiscale models [18, 67] and developing computationally efficient methods for realizing simple and accurate multiscale models for a given random process [56, 41]. From the very beginning, the development of the multiscale methodology has been motivated by its computational efficiency. Its applications in scientific and engineering problems tackled a number of large-scale estimation problems heretofore infeasible to solve with conventional methods [33, 34].

Previous results on the multiscale methodology, however, were limited almost exclusively to the modeling and estimation of *static* random fields. The need to address issues of temporal dynamics was apparent in some applications, such as remote sensing problems in oceanography, where the underlying process is inherently dynamic.

Drawing upon the results from several areas of past research on the multiscale methodology, this thesis represents a first step towards extending the multiscale framework to *dynamic* estimation. In the area of algorithmic development, the multiscale recursive estimation algorithm introduced in this thesis can be viewed as a counterpart to the Kalman filter. In the area of modeling, a new class of non-redundant models, which results in a computationally more efficient multiscale recursive procedure than other types of multiscale models used previously, is invented. In the area of applications, this new dynamic estimation method is applied to both 1-D and 2-D diffusion processes. We also apply the multiscale methodology, though in the static estimation setting, to mapping ocean surface height of the Mediterranean Sea from satellite altimetry data. Such large-scale oceanographic data assimilation problems, i.e., the melding of data with dynamic models, provided the original motivation for our investigation into multiscale dynamic estimation techniques, although the complexity of these problems renders the application of dynamic estimation methods still a challenging goal.

## 1.1 Problems Addressed and Contributions

### 1.1.1 Multiscale Recursive Estimation Algorithm

The need for computationally efficient estimation algorithms for high-dimensional dynamic systems arises in a variety of scientific and engineering applications. Numerical approaches to solving distributed parameter systems governed by partial differential equations (PDEs), such as those found in applications ranging from pollution control [74] to the modeling of ecological systems and flexible structures [5], have been limited largely due to the high-dimensionality resulting from approximating such systems by finite-dimensional lumped systems. This is certainly the case in the field of remote sensing, in which the assimilation of remote sensing data with atmospheric [22] or oceanographic [33] dynamic models yields finite-dimensional approximations of the underlying dynamics on the order of hundreds of thousands to hundreds of *millions* of

variables. Image processing applications, such as inverse problems including optical flow computation [14] and surface reconstruction [32], for example, present another source of large-scale estimation problems.

The computational challenge is substantial. 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 [11], which do not supply such error statistics, or the fast Fourier transform (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 estimation algorithms that can deal effectively with the computational challenge. The key to doing this is to find a compact and effective representation for the statistics of the estimation errors. In particular, consider the standard recursive estimation structure, which includes the optimal Kalman filter, of alternating prediction and measurement update steps. Each measurement update step has a straightforward interpretation as solving a *static* estimation problem, namely the estimation of the error in the predicted estimates based on the innovations for the current measurements. Solving this update step requires the statistics of the prediction errors which, in standard Kalman filtering, are *explicitly* available in the form of the prediction error covariance matrix, i.e., an object with  $\mathcal{O}(N^2)$  elements for an  $N$ -dimensional state. To use this operator in the update step requires  $\mathcal{O}(N^2)$  computations, which is dwarfed by the  $\mathcal{O}(N^3)$  complexity of the Riccati equation for the propagation of the error covariance through successive prediction and update steps.

Interpreting each measurement update step as a *static* spatial estimation problem suggests the possibility of *implicitly* specifying error statistics in the form of a compact multiscale *model* for the prediction error random field. The key challenge

in developing this formalism is the development of a method to propagate multiscale models over time, i.e., developing a replacement for the Riccati equation either for the propagation of this error model as it changes after each prediction and measurement steps or for the evaluation of a steady-state error model. The full covariance matrices of the estimation errors are never explicitly computed or stored, but implicitly specified via the multiscale models. The steady-state error model is iteratively solved for rather than realized directly from the steady-state error covariance matrix, which is not available. Propagation through the measurement update step is known and is accomplished as a by-product of the estimation algorithm [69]. It is the propagation of the multiscale model through the prediction step that remains, and the challenge here is to untangle the mixing effect of the dynamics in updating the parameters of the model. Doing this requires a careful examination of exactly how the temporal dynamics mix the variables captured at each scale in the multiscale representation. The analysis will depend on both the dynamics and the multiscale models used for the estimation error.

In this thesis we perform this examination for diffusion processes in detail. As we will see, multiscale models designed to exactly model MRFs [67] can adequately capture the statistics of the estimation errors of diffusion, both in 1-D and in 2-D, resulting in multiscale estimators with near-optimal performance. For 1-D diffusion, depending on the choice of multiscale models for the estimation errors, our multiscale algorithm for the dynamic propagation of the error models is of  $\mathcal{O}(N)$  computational complexity, as compared to the  $\mathcal{O}(N^3)$  complexity of the Kalman filter, where  $N$  is the number of variables to be estimated. For 2-D diffusion, due to the added spatial dimension, multiscale models generally require that the multiscale state dimensions grow with the size of the error field. Depending on the type of models, the multiscale recursive algorithm is  $\mathcal{O}(N^{3/2})$ , as compared to, again,  $\mathcal{O}(N^3)$  of the Kalman filter. Reduction in state dimensions of the 2-D models, which results in lower computational cost, is accompanied by poorer estimator performance. We examine the trade-off for several types of reduced-order models.

### 1.1.2 Non-Redundant Models

In constructing a multiscale model for a given random field, we aim to satisfy a number of requirements. The first and the focus of much of the past research effort on realization techniques [56, 41] is the accurate reproduction of second-order statistics for the given random field of interest. Multiscale models used previously have always placed the random field of interest at the finest scale. Coarser-scale states are then designed to produce the desired correlation structure at the finest scale either exactly or approximately. For the class of *internal* models [56], such coarser-scale states can be expressed as linear functionals of the fine-scale state variables. It is important that, while maintaining a certain level of accurate reproduction of second-order statistics, the state dimensions of the multiscale model be kept low, e.g., constant such that the estimation algorithm can achieve the often claimed  $\mathcal{O}(N)$  computational efficiency. In addition, care must be also taken in the design of the linear functionals in order to maintain model consistency [25]. That is, the coarser-scale states must not only have the correct second-order statistics but must also be truly linear functions of the finest-scale random variables.

In the context of dynamic estimation, the computational complexity of the multiscale recursive algorithm depends not only on the state dimension of the multiscale model for the estimation errors but also on how the states are defined. Multiscale prediction for internal models in both 1-D and 2-D are generally more complex than multiscale smoothing, i.e., the update step in the recursive algorithm. This consideration motivated the creation of a new class of *non-redundant* models. Unlike internal models, elements of the field of interest are no longer placed at the finest scale only, but distributed among all the nodes on the tree. A state variable in a non-redundant model corresponds to an element of the random field of interest and appears only once on the tree rather than at multiple nodes as in internal models [67]. Non-redundant models, like the internal models shown in [67], can model Markov random fields (MRFs) exactly. Both produce comparable estimator performance for modeling diffusion processes. The major advantage for using the non-redundant model is

computational efficiency for multiscale prediction. For a 1-D diffusion, while the multiscale prediction step is of up to  $\mathcal{O}(N \log N)$  complexity for the more conventional internal models, it is of  $\mathcal{O}(N)$  on non-redundant models. In 2-D, the complexity is  $\mathcal{O}(N^{5/2})$  for internal models and  $\mathcal{O}(N^{3/2})$  on non-redundant models.

### 1.1.3 Large-Scale Data Assimilation

Large-scale data assimilation problems in oceanographic applications provided the original motivation for our investigation into multiscale dynamic estimation methods. In addition to the high computational cost due to the large number of variables to be estimated in these applications, the precise knowledge of the statistical structure of the ocean is often lacking, and the dynamic models of the ocean derived from the underlying physics are usually immensely complicated.

In this thesis we apply the multiscale estimation framework to mapping the sea level anomaly of the Mediterranean Sea based on satellite altimetric data. Similar to the application of multiscale methodology to mapping the ocean surface height in the Pacific [33], the oceanographic signal here is modeled as a stationary  $1/f^\mu$  process, where  $f$  is the horizontal wavenumber. Unlike previous applications, the measurement noise is modeled as the sum of a Gaussian white noise process and a correlated process of low wavenumber, the latter of which is used to model the systematic errors in the satellite orbit. The correlated measurement noise process is accommodated by augmenting the multiscale model for the oceanographic signal, and is jointly estimated along with the anomaly signal. The efficiency of the multiscale scheme allowed the testing of large sets of hypothesized statistical prior model parameters in order to determine the most likely prior model. Mapping results with low-wavenumber error corrections are free of the obvious artifacts present in mapping results without the corrections.

The issue of temporal dynamics is side-stepped by assuming the ocean to be static for the duration of one repeat cycle of the satellite measurements. Temporal dynamics are taken into account by adjusting the measurement quality, i.e., measurements further from the point in time of interest are assumed to be of higher uncertainty.

While we are addressing data fusion with a spatial model of the ocean here, fusing data with a spatial model as well as a temporal dynamics model of the ocean remains a goal for further research.

## 1.2 Thesis Organization

The remaining chapters of this thesis are organized as follows.

**Chapter 2** presents the background material on multiscale estimation and realization that is relevant to the development in later chapters. It begins with a brief review of the multiscale smoothing algorithm, the smoothing error model, and the likelihood test. The section on realization methods includes a description of the correlation structure of MRFs and the development of internal models for exact realization of MRFs on multiscale tree models. The basic approach of canonical correlations realization (CCR), a generalized method for realizing any Gaussian random field is explained next. Two simple examples then illustrate the concept of conditional decorrelation and model consistency — two important issues in multiscale internal realization. External models, and  $1/f$  models in particular, are covered next. The historical and somewhat forgotten link between wavelets and the multiscale models is revisited. The chapter concludes with a summary of the major papers written on the multiscale framework as a guide for readers who are interested in learning about the field in more detail.

**Chapter 3** begins with the standard recursive estimator and draws a parallel among the optimal Kalman filter, a MRF-based recursive estimation, and our multiscale approach. The development of the chapter focuses on a method for propagating a multiscale model for the estimation error through the prediction step. Our multiscale method is explained in general terms first, and is then illustrated with several examples of 1-D internal models with end-point linear functionals for the cases of simple temporal dynamics that results in diagonal or tridiagonal dynamic matrices.

Non-redundant models are introduced next and shown to have a computational advantage over internal models for the purpose of multiscale prediction. The chapter is concluded with a discussion on the issues of convergence of the recursive algorithm and estimator performance measure.

**Chapter 4** develops multiscale models and recursive estimators for 1-D diffusion processes in detail. A careful analysis of what linear functionals are suitable for modeling the steady-state predicted estimation errors, as well as the steady-state 1-D diffusion process, is done. The analysis demonstrates that in the case of diffusion both the internal models with end-point linear functionals and non-redundant models, built to model 1-D MRFs exactly, can adequately model the estimation errors of diffusion processes. In the last section examples for several variations of the basic diffusion process and for advection-diffusion processes show that the multiscale suboptimal estimators are not only computationally efficient but also near-optimal in performance.

**Chapter 5** examines multiscale models and recursive estimators for 2-D diffusion processes. We demonstrate that exact multiscale models built for 2-D MRFs are again highly accurate for modeling the estimation errors of diffusion. The major challenge in 2-D lies in constructing reduced-order multiscale models to deal with the higher computational complexity of the recursive algorithm due to the added spatial dimension. We propose and compare several types of reduced-order non-redundant models, and examine the trade-off between computational cost and estimator performance.

**Chapter 6** details the application of the multiscale methodology to the mapping of ocean surface height anomalies in the Mediterranean Sea using TOPEX/Poseidon and ERS-1 satellite altimetric data. First, we explain how the modifications to the basic  $1/f$  model used previously for modeling ocean surface height signals take into account the correlated measurement errors. A presentation of the results on the determination of multiscale model parameters then follows. Maps generated with several different methods — a basic interpolation scheme, multiscale estimation with

and without correcting for the correlated measurement error, a suboptimal space-time interpolation scheme, and multiscale joint TOPEX/Poseidon-ERS-1 estimation — are compared.

**Chapter 7** summarizes the major results and contributions of this thesis. It discusses several interesting areas of research beyond the scope of the main chapters of this thesis, and points out some possible future research directions.



## Chapter 2

# Fundamentals of Multiscale Modeling and Estimation

In this chapter we will lay out the major building blocks of the multiscale framework of [6, 18], upon which the material in later chapters is based. This chapter will allow readers who are not familiar with this multiscale methodology to attain a basic understanding and to put in perspective where the results in this thesis stand within the larger context of multiscale estimation.

Previous research on this multiscale methodology, summarized in this chapter, answers one of the following three basic questions:

- (i) What are the multiscale models of [6, 18]?
- (ii) How is estimation done efficiently on such models?
- (iii) How does one build multiscale models for a given random process so that estimation can be done efficiently?

The chapter is roughly organized to address each of these questions. Most of the recent research has focused on answering the last of the above three questions. Building multiscale models that not only accurately capture the desired statistics but are also compact enough to achieve the computational gains is by far the most challenging aspect of applying the multiscale framework to practical estimation problems. In

later chapters, we will extend previous results on modeling static random processes to modeling dynamic processes.

## 2.1 Multiscale Stochastic Models

In the traditional linear least squares estimation (LLSE) setting, one's prior knowledge of a given random process  $\boldsymbol{\chi}$ , a column vector, is expressed through its second-order statistics<sup>1</sup> — the mean  $\mathbf{m}_\chi = E[\boldsymbol{\chi}]$ , which is usually assumed to be zero without loss of generality, and the covariance  $\mathbf{P}_\chi = E[\boldsymbol{\chi}\boldsymbol{\chi}^T]$ . We may view the covariance matrix as a statistical model for the random process of interest  $\boldsymbol{\chi}$ .

Given a set of measurements

$$\mathbf{y}_d = \mathbf{C}_d \boldsymbol{\chi} + \mathbf{v}_d, \quad (2.1)$$

where the measurement noise  $\mathbf{v}_d$  is Gaussian with zero mean and covariance  $\mathbf{R}_d$ , the linear least squares estimate has the simple form

$$\hat{\boldsymbol{\chi}} = \mathbf{m}_\chi + \mathbf{P}_\chi \mathbf{C}_d^T (\mathbf{C}_d \mathbf{P}_\chi \mathbf{C}_d^T + \mathbf{R}_d)^{-1} (\mathbf{y}_d - \mathbf{C}_d \mathbf{m}_\chi). \quad (2.2)$$

The associated estimation error covariance is

$$\mathbf{P}_e = E[\tilde{\boldsymbol{\chi}}\tilde{\boldsymbol{\chi}}^T] = \mathbf{P}_\chi - \mathbf{P}_\chi \mathbf{C}_d^T (\mathbf{C}_d \mathbf{P}_\chi \mathbf{C}_d^T + \mathbf{R}_d)^{-1} \mathbf{C}_d \mathbf{P}_\chi, \quad (2.3)$$

where  $\tilde{\boldsymbol{\chi}} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}}$  is the estimation error. Alternatively, we may express (2.2) and (2.3) in the inverse format:

$$\hat{\boldsymbol{\chi}} = \mathbf{m}_\chi + \mathbf{P}_e \mathbf{C}_d^T \mathbf{R}_d^{-1} (\mathbf{y}_d - \mathbf{C}_d \mathbf{m}_\chi) \quad (2.4)$$

$$\mathbf{P}_e^{-1} = \mathbf{P}_\chi^{-1} + \mathbf{C}_d^T \mathbf{R}_d^{-1} \mathbf{C}_d. \quad (2.5)$$

---

<sup>1</sup>For Gaussian random variables that we deal with in this thesis, second-order statistics give a complete description of the random variables.

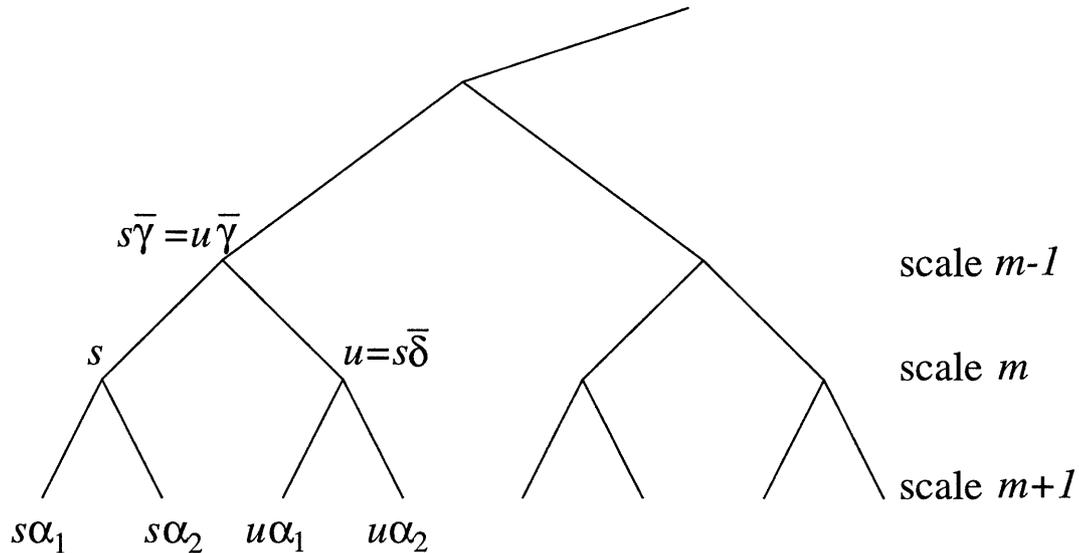


Figure 2-1: A portion of a dyadic multiscale tree.

Both forms can be used interchangeably depending on their respective computational cost.

There are two difficulties with applying LLSE directly to large systems, for example, the surface heights of an entire ocean basin on a dense grid [33, 34]:

- (i) The computational complexity of the matrix inversion involved in computing the estimates and the estimation error covariance in (2.2), (2.3), and (2.4), (2.5) grows cubically with the dimension of the number of measurements or of the number of estimates.
- (ii) The number of variables to be estimated in cases such as mapping ocean surface heights [33] can easily reach millions. The storage requirement for a million by million matrix  $\mathbf{P}_x$  is simply not feasible.

In the multiscale estimation framework of [6, 18, 56], random processes are modeled on tree structures. Figure 2-1 shows an example of a dyadic tree. The nodes of the tree are organized into a sequence of scales. A node  $s$  is connected to a unique parent node,  $s\bar{\gamma}$ , at the next coarser level, and to several child nodes  $s\alpha_i$  ( $i = 1, \dots, q$ ), at the next finer level. The multiscale process is a collection of zero-mean random vectors  $\mathbf{x}(s)$ , indexed by nodes  $s$  on the tree and specified by a scale-to-scale relationship

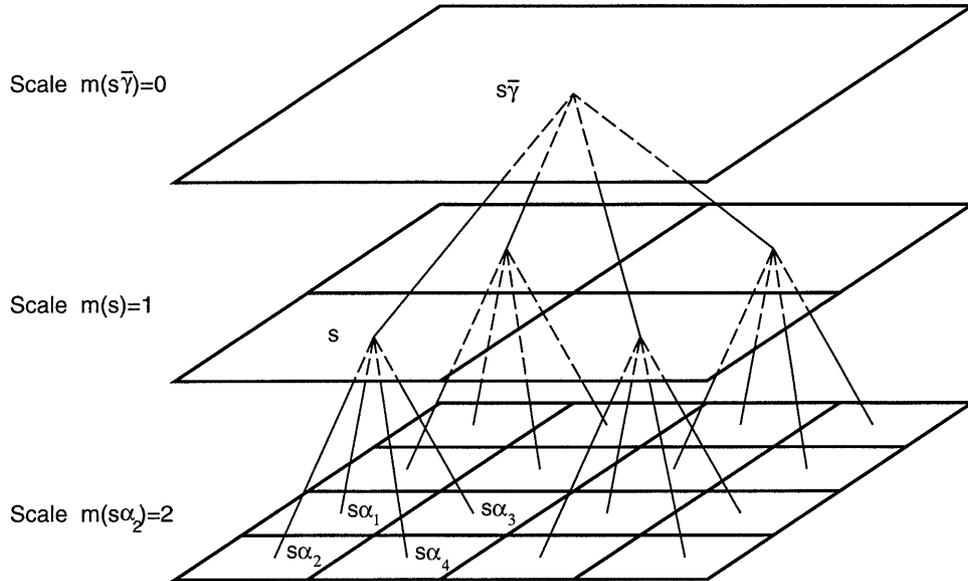


Figure 2-2: An example quad tree for modeling 2-D random fields. (Taken from [33].)

of the following form

$$\mathbf{x}(s) = \mathbf{A}(s)\mathbf{x}(s\bar{\gamma}) + \mathbf{B}(s)\mathbf{w}(s), \quad (2.6)$$

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

In general, the number of children may vary from scale to scale and in fact from node to node. For convenience, in the vast majority of cases, 1-D random processes are mapped to the finest scale of uniform dyadic trees (i.e.,  $q = 2$  for all nodes except the leaf nodes at the finest scale) as in Figure 2-1, and 2-D random fields are mapped to the finest scale of quad trees ( $q = 4$ ) as in Figure 2-2. We will show in Section 3.5 that in some cases the best way to model a random process of interest is not mapping it to the finest scale but distributing it among different nodes at different scales on the tree. How to find the appropriate  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$  given a random process of interest is a realization problem which will be covered in Section 2.3. For now, we assume that a multiscale model of (2.6) has already been defined for us.

Measurements can be made at any node:

$$\mathbf{y}(s) = \mathbf{C}(s)\mathbf{x}(s) + \mathbf{v}(s), \quad (2.7)$$

where  $\mathbf{v}(s)$  is white, zero-mean, and uncorrelated with the process  $\mathbf{x}(s)$ . Although measurements can be taken at any scale, they are restricted to be linear functions of the state  $\mathbf{x}(s)$ . Suppose that each element in the process of interest is mapped to a scalar node at the finest scale of a tree. Then only point measurements of the process are readily placeable on the finest scale of tree. Additional states can be augmented onto the tree [25] so that non-local measurements can be placed on the tree, but augmentation must be done with care to guarantee model consistency (see Section 2.3.4).

Whereas in the LLSE setting one's prior knowledge of the random process is expressed in the form of a covariance matrix  $\mathbf{P}_\chi$ , in the multiscale framework that prior knowledge is expressed through the scale-to-scale relationship of (2.6), i.e., model parameters  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$ . Equation (2.6), together with the covariance  $\mathbf{P}(0)$  of the variable  $\mathbf{x}(0)$  at the root node of the tree, provides an implicit specification of the full covariance of the multiscale process. Computing the full covariance is straightforward since the explicit covariance between any two nodes  $\mathbf{x}(s_1)$  and  $\mathbf{x}(s_2)$  can be easily calculated as

$$\mathbf{P}(s_1, s_2) \triangleq E [\mathbf{x}(s_1)\mathbf{x}^T(s_2)] = \mathbf{\Phi}(s_1, s_1 \wedge s_2)\mathbf{P}(s_1 \wedge s_2)\mathbf{\Phi}^T(s_2, s_1 \wedge s_2), \quad (2.8)$$

where  $s_1 \wedge s_2$  is the lowest-scale common ancestor of  $s_1$  and  $s_2$ ,  $\mathbf{P}(s_1 \wedge s_2)$  is the covariance of  $\mathbf{x}(s_1 \wedge s_2)$ , and  $\mathbf{\Phi}(s, \sigma)$  is the state transition matrix from any node  $\sigma$  to direct descendent  $s$ . For example, referring to Fig. 2-1,

$$\mathbf{P}(s\alpha_i, u) \triangleq E [\mathbf{x}(s\alpha_i)\mathbf{x}^T(u)] = \mathbf{A}(s\alpha_i)\mathbf{A}(s)\mathbf{P}(s\bar{\gamma})\mathbf{A}^T(u). \quad (2.9)$$

Furthermore, the covariances of  $\mathbf{x}(s)$  at each individual node can be recursively computed from  $\mathbf{P}(0)$  in a tree-recursive Lyapunov equation:

$$\mathbf{P}(s) = \mathbf{A}(s)\mathbf{P}(s\bar{\gamma})\mathbf{A}^T(s) + \mathbf{B}(s)\mathbf{B}^T(s). \quad (2.10)$$

Thus, the calculation of  $\mathbf{P}(s)$  and any individual  $\mathbf{P}(s_1, s_2)$  is computationally simple

(at most  $\mathcal{O}(N)$  for all of the  $\mathbf{P}(s)$ ). However, the calculation of *all* of the cross-covariances  $\mathbf{P}(s_1, s_2)$  is prohibitively complex for large  $N$  since there are  $\mathcal{O}(N^2)$  such covariances. Of course, the whole reason for applying multiscale models is to avoid ever computing or storing the full covariance matrix  $\mathbf{P}_\chi$ .

LLSE makes no special assumptions about the covariance structure  $\mathbf{P}_\chi$ . This generality comes with a computational penalty —  $\mathcal{O}(N^3)$  computational complexity and  $\mathcal{O}(N^2)$  storage requirements for a length  $N$  random process. Computationally more efficient methods necessarily put restrictions on the range of random processes for which such methods are applicable. FFT methods exploit the stationary correlations structure in a random process, i.e., one column of  $\mathbf{P}_\chi$  is enough to represent the whole matrix. Markov random field (MRF) [28] models exploit the conditional decorrelation properties, i.e., the inverse of the covariance matrix  $\mathbf{P}_\chi^{-1}$  is sparse and/or banded [80]. (More will be said about MRFs and multiscale realizations of MRFs in Section 2.3.2.) Similarly, multiscale models exploit certain statistical structure of a random process and express such structure in a scale-to-scale relationship of (2.6) that is not only more compact to express but also results in more efficient estimation algorithms. Two classes of random processes — MRF (see Section 2.3.2) and  $1/f$  processes (see Section 2.3.5) — have been so far identified as particularly suitable for modeling in the multiscale framework.

## 2.2 Multiscale Algorithms

### 2.2.1 Multiscale Smoothing Algorithm

From (2.6), conditioned on the state  $\mathbf{x}(s)$  the  $q + 1$  subtrees connected to node  $s$  are conditionally decorrelated (by the whiteness of  $\mathbf{w}(s)$ ). Because of this Markovianity property on the multiscale tree, the random variable  $\mathbf{x}(s)$  is referred to as the state at node  $s$ . The conditional decorrelation property also makes possible efficient scale-recursive smoothing algorithms [18, 19], similar to the Rauch-Tung-Striebel (RTS) smoothing algorithm [75] for time series. In fact, a monadic tree, i.e.  $q = 1$ , reduces

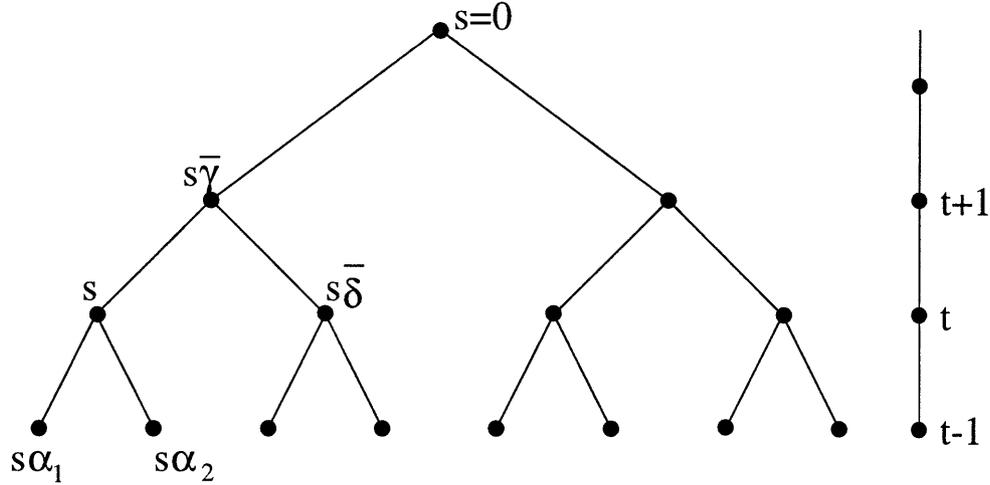


Figure 2-3: Rauch-Tung-Striebel smoothing algorithm on the tree.

to a standard 1-D Markov process as illustrated in Figure 2-3.

In the time series case, the RTS algorithm consists of a Kalman filtering estimation algorithm running forward in time and a smoothing algorithm backward in time. In alternating update and prediction steps, the Kalman filter recursively computes the best estimate of the process at time  $t$  given all measurements up to  $t$  and the best estimate of the process at time  $t + 1$  given measurements up to  $t$ . The smoothing algorithm recursively computes the best estimate of the process at time  $t$  given all measurements.

The RTS algorithm can be generalized for smoothing on the multiscale tree. The generalized algorithm consists of a single upward sweep similar to Kalman filtering and a downward smoothing sweep. The analogy with the time series case breaks down in that each node of the multiscale tree can have multiple immediate descendents, not just one. A new merge step is now required in the prediction step during the upward sweep. The best estimates of  $\mathbf{x}(s)$  given measurements on each individual subtree originating from each of its child nodes  $s\alpha_i$  are combined into an estimate of  $\mathbf{x}(s)$  given all measurements on all of these subtrees. Detailed development of the smoothing algorithm can be found in [18, 19]. The essential equations of the algorithm are summarized in Appendix A.

For models with state dimension  $k$  independent of the number of nodes at the

finest scale of the tree  $N$ , the computational complexity of the smoothing algorithm is  $\mathcal{O}(k^3N)$  for the computation of estimates *and* error covariances at *all* nodes of the tree. In comparison, the computational complexity of LLSE is  $\mathcal{O}(N^3)$ . The  $\mathcal{O}(N)$  complexity, however, hinges on a constant  $k$  independent of  $N$ . For  $k$  that grows linearly with  $N$ , e.g., exact internal models for 2-D MRFs to be seen in Section 2.3.2, only  $\mathcal{O}(N^{3/2})$  complexity is not achieved. One of the major challenges of multiscale realization is building models that do achieve  $\mathcal{O}(N)$  complexity, or at worst only slightly faster growth with respect to the problem size  $N$ .

## 2.2.2 Smoothing Error Model

In addition to the estimates  $\hat{\mathbf{x}}(s)$ , the multiscale smoothing algorithm produces, at no extra computational cost, a multiscale model for the *a posteriori* estimation errors  $\tilde{\mathbf{x}}(s) = \mathbf{x}(s) - \hat{\mathbf{x}}(s)$  [69]. The multiscale model for the estimation errors are of the exact same structure as the model for the process, but with new “ $\mathbf{A}(s)$ ” and “ $\mathbf{B}(s)$ ” matrices defining the scale-to-scale relationships:

$$\tilde{\mathbf{x}}(s) = \mathbf{P}(s|s)\mathbf{P}(s)^{-1}\mathbf{A}(s)\mathbf{P}(s\bar{\gamma})\mathbf{P}^{-1}(s\bar{\gamma}|s)\tilde{\mathbf{x}}(s\bar{\gamma}) + \check{\mathbf{w}}(s) \quad (2.11)$$

where  $\mathbf{P}(s|\sigma)$  is the estimation error covariance for estimating  $\mathbf{x}(s)$  given all measurements at node  $\sigma$  and below,  $\mathbf{P}(s)$  is the prior covariance of  $\mathbf{x}(s)$ , and  $\check{\mathbf{w}}(s)$  is the error process noise. (See [69] and Appendix A for more detail.) In *static* estimation problems, often only the estimation error variances, i.e., only the diagonals of  $E[\tilde{\mathbf{x}}(s)\tilde{\mathbf{x}}(s)^T]$  are of interest as a measure of the quality of the estimates. The multiscale model for the errors are of secondary importance. We shall see in Chapter 3 that the update step of a temporal Kalman filter can be interpreted as static estimation of the predicted estimation error while the *a posteriori* error of this static estimation step is exactly the updated estimation error. The availability of a multiscale model for the *a posteriori* estimation errors given a model for the process to be estimated is of direct interest to us.

### 2.2.3 Multiscale Likelihood Test

The multiscale tree structure also admits a fast algorithm for calculating the likelihood function for the model, i.e., the log probability density of the measurement data given the model. Such likelihood tests are used in applications such as texture discrimination [68, 70], where likelihood ratio tests are performed to decide between two multiscale models parameterized with different sets of parameters for two types of texture. The computational efficiency of the likelihood test algorithm also allows a large number of tests to be performed to identify the model parameters that maximize the likelihood function. Past applications of this type include fractal dimension identification [35] and ocean surface height model parameter identification [34, 33]. In Chapter 6, we will employ multiscale likelihood tests to determine two parameters used in multiscale models for the sea level anomaly of the Mediterranean Sea.

Suppose that all available measurements on the multiscale tree are stacked into a vector  $\mathbf{y}$ , which has a Gaussian distribution with zero-mean and covariance  $\mathbf{P}_\mathbf{y}$  that depends on a parameter vector  $\boldsymbol{\theta}$ . The likelihood function is

$$\mathcal{L}(\boldsymbol{\theta}) = \log p_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \quad (2.12)$$

$$= -\frac{m}{2} \log 2\pi - \frac{1}{2} \log |\mathbf{P}_\mathbf{y}| - \frac{1}{2} \mathbf{y}^T \mathbf{P}_\mathbf{y}^{-1} \mathbf{y}, \quad (2.13)$$

where  $m$  is the number of elements in  $\mathbf{y}$  and  $|\mathbf{P}_\mathbf{y}|$  is the determinant of  $\mathbf{P}_\mathbf{y}$ . The main difficulty with computing (2.13) directly is the matrix inversion of  $\mathbf{P}_\mathbf{y}$ , as the covariance matrix is generally full. However, if an invertible transformation  $\mathbf{T}$  that whitens  $\mathbf{y}$  can be found, so that the covariance of  $\boldsymbol{\nu} = \mathbf{T}\mathbf{y}$ ,  $\mathbf{P}_\boldsymbol{\nu} = \mathbf{T}\mathbf{P}_\mathbf{y}\mathbf{T}^T$ , is diagonal, then

$$\mathcal{L}(\boldsymbol{\theta}) = -\log |\mathbf{T}| - \frac{m}{2} \log 2\pi - \frac{1}{2} \log |\mathbf{P}_\boldsymbol{\nu}| - \frac{1}{2} \boldsymbol{\nu}^T \mathbf{P}_\boldsymbol{\nu}^{-1} \boldsymbol{\nu} \quad (2.14)$$

is easy to compute. For Markov time series, the Kalman filter performs the whitening by generating the filter innovations. This is the trivial case on the tree for  $q = 1$ . For  $q > 1$ , however, the filtering residuals  $\boldsymbol{\nu}_f(s) = \mathbf{y}(s) - \mathbf{C}(s)\hat{\mathbf{x}}(s|s+)$  generated by the Kalman filtering upward sweep of the multiscale estimation algorithm are

only partially whitened. That is, only residuals along a direct descendent line are white. Residuals at nodes across a scale are not white. Nevertheless, the partial whitening of the upward sweep can be used to advantage in a multiscale analog of the Kalman filter. The nodes on the tree are ordered according to a “pre-order” scheme in which child nodes appear before their parents. The likelihood calculation algorithm essentially provides the additional whitening required to make the residuals  $\nu$  completely uncorrelated. See [68, 70] for more detailed discussions of the algorithm.

The computational effort of the whitening algorithm is similar to that for the estimation algorithm. For multiscale models with constant state dimension, the computational complexity of the likelihood algorithm is  $\mathcal{O}(N)$ .

## 2.3 Multiscale Realization

As we have seen in Section 2.1, given a multiscale model of the form in (2.6) it is straightforward, though computationally intensive, to calculate the full covariance structure among states at all nodes on the tree. For the subset of models, for which the random process of interest is mapped to the finest scale, the covariance matrix of interest is the one for the finest scale. The converse, i.e., constructing a multiscale model given the covariance matrix of a random process of interest so that the finest-scale covariance exactly or approximately matches the given covariance, is a far more difficult task. The majority of recent work on the multiscale methodology has focused on efficient stochastic realization algorithms, motivated by the computational advantages of the multiscale framework.

All of the past results on multiscale realization have mapped the random process of interest to the finest scale of a multiscale tree. Most of the existing work focuses on building multiscale models so that the statistics of the finest-scale process match or approximate a desired finest-scale covariance, although there are motivations and some results [25] on constructing realizations under more general conditions. This differs from the usual realization problem for time series for which we wish to meet or approximate a second-order specification over the entire time index set. For multiscale

representations here we are interested in only a limited aperture of the full tree index set, namely the finest scale. To be more specific, let  $\mathbf{x}$  denote the set of all multiscale state variables  $\mathbf{x}(s)$  at the finest scale; let  $\mathbf{P}_{\mathbf{x}}$  denote its covariance. Similarly, let  $\mathbf{P}_{\chi}$  denote the desired covariance (of the finest-scale process  $\chi$ ) we wish to realize. The fine-scale realization problem is then to construct a multiscale model as in (2.6) so that  $\mathbf{P}_{\mathbf{x}}$  (approximately) equals  $\mathbf{P}_{\chi}$ .

The key to this problem is defining the state at the “hidden” levels above the finest one so that the resulting process satisfies the tree-Markovianity property, while the finest-scale covariance  $\mathbf{P}_{\mathbf{x}}$  matches the desired  $\mathbf{P}_{\chi}$  either exactly or approximately. Multiscale models developed so far fall into two main categories — internal and external models. Internal models are models where the coarser-scale states are defined explicitly as a linear function of the finest-scale states [67, 73, 57, 56, 25]. Coarse-scale states on external models, on the other hand, cannot be written explicitly as a linear function of the finest-scale states [33, 34, 35]. In either case, the coarser-scale states are created for the purpose of generating a desired covariance structure at the finest scale.

We will discuss in the remainder of this chapter the relevant past result on internal and external realizations. In Section 3.5 we will introduce a new type of *non-redundant models* where the process of interest is uniquely mapped onto the multiscale tree, i.e., there is a one-to-one correspondence between variables on the tree and variables defining the process of interest. Realization ideas described in this section will be extended in Section 3.5 to modeling Markov random fields on non-redundant models.

### 2.3.1 Internal Realization Models

Just as for time series, (2.6) can be interpreted as writing  $\mathbf{x}(s)$  as its best estimate based on its parent  $\mathbf{x}(s\bar{\gamma})$  plus the (orthogonal) error  $\mathbf{w}(s)$  in this estimate. That is,

$$\mathbf{x}(s) = E[\mathbf{x}(s)|\mathbf{x}(s\bar{\gamma})] + \mathbf{w}(s), \quad (2.15)$$

where

$$E[\mathbf{x}(s)|\mathbf{x}(s\bar{\gamma})] = \mathbf{A}(s)\mathbf{x}(s\bar{\gamma}), \quad (2.16)$$

and

$$E[\mathbf{w}(s)\mathbf{w}(s)^T] = \mathbf{B}(s)\mathbf{B}^T(s). \quad (2.17)$$

This implies that the parameters  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$  are completely determined by the joint statistics of  $\mathbf{x}(s)$  and  $\mathbf{x}(s\bar{\gamma})$ :

$$\mathbf{A}(s) = \mathbf{P}(s, s\bar{\gamma})\mathbf{P}^{-1}(s\bar{\gamma}), \quad (2.18)$$

$$\mathbf{B}(s)\mathbf{B}^T(s) = \mathbf{P}(s) - \mathbf{P}(s, s\bar{\gamma})\mathbf{P}^{-1}(s\bar{\gamma})\mathbf{P}^T(s, s\bar{\gamma}). \quad (2.19)$$

Thus, model construction is immediate once the multiscale states are defined *and* the parent-child second-order statistics are computed at each node.

For the class of internal realization the state  $\mathbf{x}(s)$  at each node can be defined as a linear functional of the finest-scale process. That is,

$$\mathbf{x}(s) = \mathbf{L}(s)\mathbf{x}. \quad (2.20)$$

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

$$\mathbf{P}(s, s\bar{\gamma}) = \mathbf{L}(s)\mathbf{P}_\chi\mathbf{L}^T(s\bar{\gamma}), \quad (2.21)$$

$$\mathbf{P}(s) = \mathbf{L}(s)\mathbf{P}_\chi\mathbf{L}^T(s). \quad (2.22)$$

The way  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$  are defined in (2.18) and (2.19) guarantees (2.21) and (2.22) to be true for any node  $s$  on the realized multiscale tree model, for any arbitrary choice of linear functionals  $\mathbf{L}(s)$ . For example, suppose that the process of interested is mapped to the finest scale where all nodes have scalar states. This means that at least the diagonal of the realized covariance  $\mathbf{P}_\mathbf{x}$  exactly matches that of the desired covariance  $\mathbf{P}_\chi$ . On the other hand,  $\mathbf{P}(s, \sigma) = \mathbf{L}(s)\mathbf{P}_\chi\mathbf{L}^T(\sigma)$ , for  $\sigma$  not equal to

$s$  or  $s\bar{\gamma}$ , does not generally hold true unless the linear functionals  $\mathbf{L}(s)$  have been designed to meet the Markovianity property on the tree. For example, the state at node  $s\bar{\gamma}$  in Figure 2-1 must decorrelate three subsets of the finest-scale process, namely  $\{\mathbf{x}(s\alpha_1), \mathbf{x}(s\alpha_2)\}$ ,  $\{\mathbf{x}(u\alpha_1), \mathbf{x}(u\alpha_2)\}$ , and all of the remainder of the finest scale. This is why we have made a distinction between what is realized on the multiscale model,  $\mathbf{P}_{\mathbf{x}}$ , and what is desired,  $\mathbf{P}_{\chi}$ .

In the following subsections on internal models, we will first describe internal models for the class of Markov random fields (MRF), which will be used heavily later on in the development of models for dynamic estimation. We will then briefly explain how a canonical correlations realization (CCR) method constructs internal models for general Gaussian random fields. We will have more to say about two important requirements we place on the linear functionals of such internal models — conditional decorrelation and consistency.

### 2.3.2 Internal Models for MRFs

A 1-D process  $\chi(t)$ ,  $t \in \mathcal{R}$ , is a Markov process (or a unilateral Markov process) [28] if for  $t > T$

$$p_{\chi(t)|\chi(\tau), \tau \in (0, T]}(\chi(t)|\chi(\tau), \tau \in (0, T]) = p_{\chi(t)|\chi(T)}(\chi(t)|\chi(T)). \quad (2.23)$$

That is, conditioned on the present,  $\chi(T)$ , the past,  $\chi(t)$ ,  $t < T$ , and the future,  $\chi(t)$ ,  $t > T$ , are uncorrelated. A 1-D process  $\chi(t)$  is a 1-D Markov random field (or a bilateral Markov process or a reciprocal process) [28] if for  $t \in (T_1, T_2)$ ,  $T_2 > T_1$ ,

$$p_{\chi(t)|\chi(\tau), \tau \in (T_1, T_2)^c}(\chi(t)|\chi(\tau), \tau \in (T_1, T_2)^c) = p_{\chi(t)|\chi(T_1), \chi(T_2)}(\chi(t)|\chi(T_1), \chi(T_2)), \quad (2.24)$$

where  $(T_1, T_2)^c$  is the complement of  $(T_1, T_2)$ . That is, conditioned on the boundary points,  $\chi(T_1)$  and  $\chi(T_2)$ ,  $\chi(t)$  within the interval  $(T_1, T_2)$  is conditionally decorrelated from the field outside of the interval. Note that a Markov process is also a 1-D MRF, but a 1-D MRF is not necessarily a Markov process. Similarly, a continuous-

space 2-D random field  $\chi(t)$ ,  $t \in \mathcal{R}^2$ , is a Markov random field if a closed boundary  $\Gamma$  conditionally decorrelates the random field inside the boundary from the field outside.

Discrete 1-D MRFs on  $t \in \mathcal{Z}$  are defined similarly with the continuous intervals replaced by discrete intervals. Discrete 2-D MRFs [28] on  $t \in \mathcal{Z}^2$  require the definition of a neighborhood structure to make precise what a closed boundary is. A first-order neighborhood of point on a lattice consists of its four nearest neighboring points; a second-order neighborhood the eight nearest, and so on. The boundary of a subset  $\Omega \in \mathcal{Z}^2$  is the set of points that are neighbors of  $\Omega$  but not elements of  $\Omega$ . The order of the MRF is the order of the neighborhood needed to define a boundary that conditionally decorrelates  $\chi(t)$  on  $\Omega$  from  $\chi(t)$  on  $\Omega^c$ . This definition of order applies to both 1-D and 2-D MRFs.

There is considerable structure in the covariance of a Markov random field. Although the covariance matrix of a MRF  $\mathbf{P}_\chi$  is generally full, its inverse  $\mathbf{P}_\chi^{-1}$  is sparse and banded [80]. Take the following 1-D Markov process for example:

$$a\chi(t) = b\chi(t-1) + w(t), \quad (2.25)$$

for  $t = 0, \dots, T$ , where  $\chi(-1) = 0$ ,  $w(t)$  is a white noise process, and  $a$  and  $b$  are constant coefficients. Written in vector form:

$$\begin{bmatrix} a & 0 & \cdots & \cdots & 0 \\ -b & a & 0 & & \vdots \\ 0 & -b & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -b & a \end{bmatrix} \begin{bmatrix} \chi(0) \\ \chi(1) \\ \vdots \\ \chi(T-1) \\ \chi(T) \end{bmatrix} = \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(T-1) \\ w(T) \end{bmatrix} \quad (2.26)$$

or

$$\mathbf{M}\boldsymbol{\chi} = \mathbf{w}, \quad (2.27)$$

where  $\mathbf{w}$  is white with diagonal covariance matrix  $\mathbf{Q}$ . Then

$$\mathbf{M}\mathbf{P}_\chi\mathbf{M}^T = \mathbf{Q}. \quad (2.28)$$

Assuming an invertible  $\mathbf{M}$ ,

$$\mathbf{P}_\chi^{-1} = \mathbf{M}^T \mathbf{Q}^{-1} \mathbf{M}. \quad (2.29)$$

Since  $\mathbf{Q}$  is diagonal,  $\mathbf{P}_\chi^{-1}$ , the inverse of the covariance of the 1-D process, is tridiagonal. For second-order Markov processes where  $\mathbf{M}$  has two lower diagonals,  $\mathbf{P}_\chi^{-1}$  is penta-diagonal.

Information about the conditional dependencies among elements of a discrete random process is in fact contained in the inverse of its covariance matrix [80]. A connected graph may be drawn according to the locations of the non-zero entries of the inverse covariance matrix. Conditional decorrelation relationships can then be deduced from the graph. (The reader is referred to [80] for more detail.) For our purposes here, it suffices to say that if the inverse of the covariance matrix is tridiagonal, then the process is a first-order Markov process. Furthermore, if the inverse is tridiagonal with two corner elements, then the process is a first-order 1-D Markov random field. If the inverse is penta-diagonal, then the process is second order, etc.

The multiscale models described in [67] for MRFs originate from the midpoint deflection algorithm for synthesizing sampled 1-D Brownian motion. The idea is that given the values of the boundaries of an interval  $I$  of a 1-D MRF, the value of a point inside the interval (not necessarily the exact midpoint) can be synthesized independently of the values of points outside of that interval. The value of the synthesized midpoint is simply the linear interpolation, i.e., the average of the values of the boundary points, plus a deflection term whose variance is the estimation error variance in estimating the midpoint given the values of the boundary points. The midpoint and the boundary points of  $I$  then form the boundary points of two subintervals. The midpoint deflections in the two subintervals are conditionally uncorrelated given the boundary points. This midpoint deflection recursion is naturally modeled on a multiscale tree with the interval boundary points defined as the states.

Figure 2-4 shows an example for a sampled 1-D MRF over a discrete finite index set, i.e., the set  $\{1 \dots 9\}$ . Each node is associated with a subinterval of  $\{1 \dots 9\}$ ; for

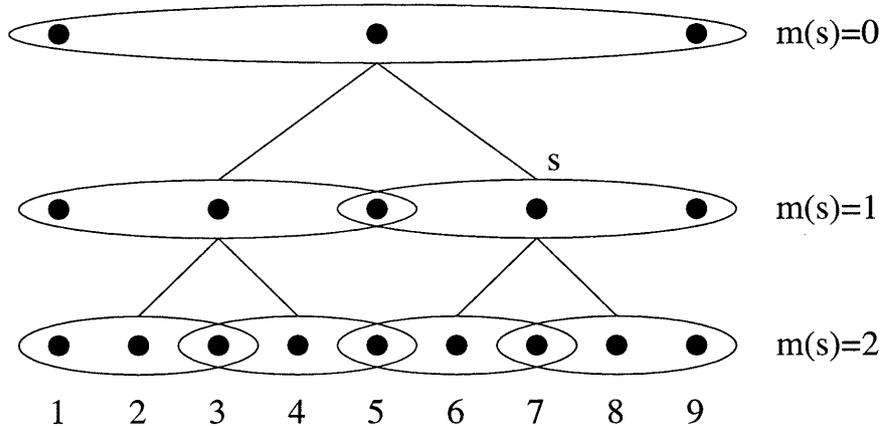


Figure 2-4: The linear functionals of an internal realization of a first-order 1-D Markov random field on a dyadic tree with a state dimension of three.

example, the node labeled  $s$  in the figure is associated with the subinterval  $\{5 \dots 9\}$ . The state at each node  $s$  consists of the set of values of the finest-scale process at the end-points and the midpoint of the subinterval, and is marked within an elongated oval in the figure. Since the states at all nodes are copies of the finest-scale variables, the multiscale model in Figure 2-4 is internal. The states so defined at any node perfectly decorrelate the states on its child subtrees and their complement, thereby satisfying the Markovianity property on tree. The multiscale dynamics (2.6) then consist of subdividing intervals at each scale and generating midpoint values.

There is considerable redundancy as most states appear at multiple nodes of the same scale as well as at multiple nodes at different scales, e.g., the variable at location 5 appears at two nodes at scale 1, and it appears at some nodes on all scales. This model redundancy can be reduced by allowing the state dimension to increase to four as shown in Figure 2-5. Each fine-scale variable appears at only one node across a scale, even though it still appears at multiple nodes at different scales. Since the linear functionals at a node consist of the end-points of the subintervals under the node, we refer to the functionals in Figure 2-5 collectively as *end-point linear functionals*, and refer to the models using such functionals as internal models with end-point linear functionals for convenience. Since each functional has a region of support over only one element of the finest-scale process, we refer to these functionals as *point functionals*. In addition, if we make the finest-scale states scalar, i.e., each node at

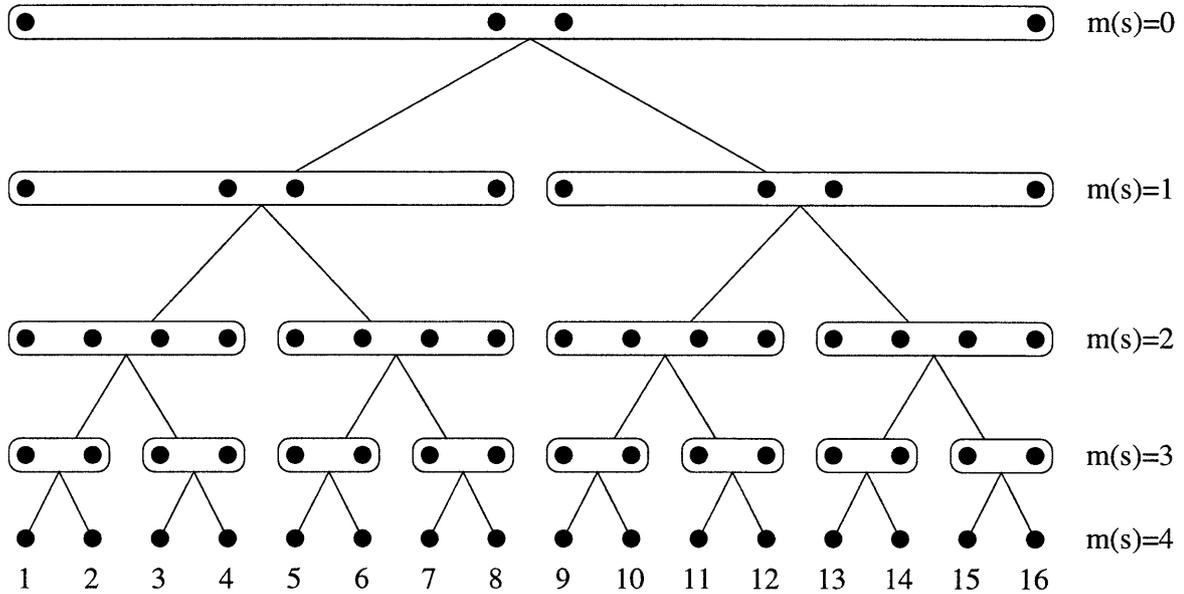


Figure 2-5: The end-point linear functionals of an internal realization of a first-order 1-D Markov random field on a dyadic tree with a state dimension of four.

the finest scale corresponds to an element of the process of interest, as in Figure 2-5, we really need only define the multiscale state to two levels above the finest scale.

The multiscale models shown in Figures 2-4 and 2-5 are both exact models for 1-D discrete Markov random fields. The midpoint deflection approach behind the construction of these models also means that they can be used to approximate continuous MRFs to arbitrary resolution by adding finer scales to the tree while keeping the same coarser-scale state definition. The random field of interest is readily available at the finest scale. In Section 3.5, we will propose a class of non-redundant models for which a state variable corresponding to an element of the random field of interest appears only once on the tree rather than at multiple nodes<sup>2</sup>. These models are still exact models for discrete MRFs, but there are subtle but important differences. (We will elaborate in Section 3.5.)

Exact realizations of 2-D MRFs are analogous to those in 1-D. Instead of end-points of subintervals, boundary points of subregions make up the states. The key

<sup>2</sup>In [70, 67], the model in Figure 2-5 is referred to as non-redundant since a state variable appears only once over a single scale, unlike the model in Figure 2-4. Our notion of non-redundancy is more stringent.

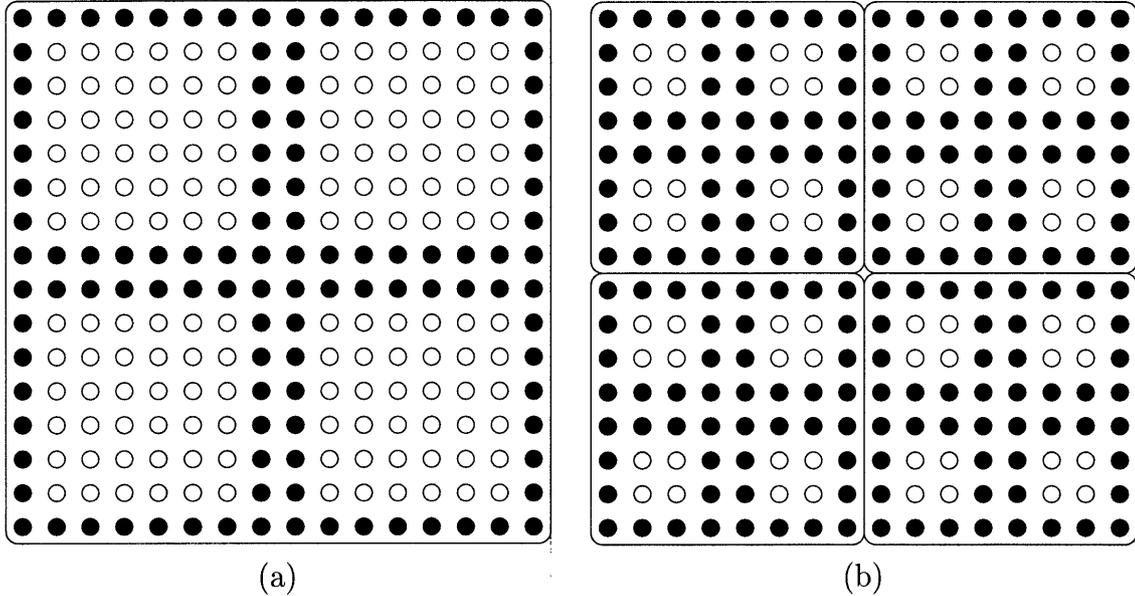


Figure 2-6: Boundary points (marked with filled circles) kept as states at (a) the root node, and (b) the four nodes at scale 1, of an internal realization of a  $16 \times 16$  first-order 2-D Markov random field.

difference and the major difficulty with exact models for 2-D MRFs is the growth of state dimension. Whereas in 1-D the state dimension is constant regardless of the size of the interval over which the process of interest is to be represented, in 2-D the state dimension grows with the length of an edge of the 2-D domain, i.e., for a square field of interest with  $N$  elements, the state dimension  $k \sim \sqrt{N}$ . The claimed  $\mathcal{O}(N)$  complexity of the multiscale smoothing algorithm hinges on a constant state dimension  $k$  independent of  $N$ . For the linear functionals shown in Figure 2-6, the state dimension at node  $s$  is  $8 \times \left( \frac{n}{2^{m(s)}} \right) - 16$ , where  $n$  the the length of one edge of the 2-D field, i.e.,  $N = n^2$ , and  $m(s)$  is the scale of node  $s$ . The total amount of computation of the smoothing algorithm is then on the order of

$$\sum_{m=0}^{\log_2 n} 4^m \times \left( 8 \times \frac{n}{2^{m(s)}} - 16 \right)^3. \quad (2.30)$$

Writing out the terms and neglecting the lower-order term, we have  $\mathcal{O}(n^3)$ . Then the computational complexity of the smoothing algorithm for exact multiscale models for 2-D MRFs is  $\mathcal{O}(N^{3/2})$ , where  $N$  is the number of variables in the field of interest. The order of computational complexity is largely determined by the state at the root

node whose dimension is  $\mathcal{O}(n)$  and whose associated computational burden is  $\mathcal{O}(n^3)$ . This is not dissimilar to the computational complexity of nested dissection [46, 77] techniques in which Gaussian eliminations on a  $n^2 \times n^2$  matrix is dominated by the inversion of an  $n \times n$  submatrix resulting in an  $\mathcal{O}(n^3)$  algorithm.

In some applications, additional structure in the correlations can be exploited to reduce the state dimensions. Rather than keeping the entire set of boundary points as the multiscale states, linear functions of the boundary points, i.e., a coordinate transformation such as 1-D Fourier series or 1-D wavelet transform, are kept. In modeling textures [67, 68], e.g., wood, which are highly correlated in one physical direction, only a few wavelet coefficients of the boundary in that direction are needed to capture most of the correlations and are kept as the multiscale state variables. The resulting multiscale model is approximate, but with much lower state dimensions. In modeling the ocean temperature field [73, 31], since nearby elements of the field are highly correlated, subsampled boundary points rather than the full boundary set are kept as the multiscale state variables.

### 2.3.3 Canonical Correlations Realization

The approximate models for Markov random fields used for modeling textures in [67] suggested the need for a way to prioritize linear functionals so that reduced-order models can be built to varying degrees of model fidelity. A general method for constructing the linear functionals  $\mathbf{L}(s)$  to achieve exact or approximate conditional decorrelation required by the tree model is the so-called canonical correlations realization (CCR) algorithm [56, 55]. It represents a generalization of Akaike’s canonical correlations algorithm [2] for the stochastic realization of time series, where one wants to define, at a point in time, a set of state variables, written as linear functionals of the past or of the future, that either exactly or approximately conditionally decorrelates the past and the future. Following the same idea, but instead of decorrelating the past from the future, the canonical correlations realization method for multiscale modeling finds states  $\mathbf{x}(s)$  that decorrelate one segment of a 1-D Gaussian process from its complement or one region of a 2-D Gaussian random field from its complement. Let

us denote  $\mathbf{x}(s)$  to be the collection of finest-scale states under node  $s$  and denote  $\mathbf{x}^c(s)$  to be the complement of  $\mathbf{x}(s)$ . Canonical correlations realization algorithm computes the set of linear functionals  $\mathbf{L}(s)$  of (2.20) so that  $\mathbf{x}(s)$  conditionally decorrelates  $\mathbf{x}(s\alpha_i)$ ,  $i = 1, \dots, q$ , and their complement.

The basic idea of CCR is the following. Given a zero-mean random vector

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix}, \quad (2.31)$$

which is written as two sub-vectors of dimensions  $n_1$  and  $n_2$ , and given its covariance matrix

$$\mathbf{P}_\eta = \begin{bmatrix} \mathbf{P}_{\eta_1} & \mathbf{P}_{\eta_1\eta_2} \\ \mathbf{P}_{\eta_1\eta_2}^T & \mathbf{P}_{\eta_2} \end{bmatrix}, \quad (2.32)$$

there exist matrices  $\mathbf{T}_1$  and  $\mathbf{T}_2$  of dimensions  $m_1 \times n_1$  and  $m_2 \times n_2$  such that

$$\begin{bmatrix} \mathbf{T}_1 & 0 \\ 0 & \mathbf{T}_2 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{\eta_1} & \mathbf{P}_{\eta_1\eta_2} \\ \mathbf{P}_{\eta_1\eta_2}^T & \mathbf{P}_{\eta_2} \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 & 0 \\ 0 & \mathbf{T}_2 \end{bmatrix}^T = \begin{bmatrix} \mathbf{I}_{m_1} & \mathbf{D} \\ \mathbf{D} & \mathbf{I}_{m_2} \end{bmatrix}. \quad (2.33)$$

The matrix  $\mathbf{D}$  has dimensions  $m_1 \times m_2$ , and is given by

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{D}} & 0 \\ 0 & 0 \end{bmatrix}, \quad (2.34)$$

where  $\hat{\mathbf{D}}$  is a unique, positive definite diagonal matrix of dimension  $m_{12}$ . Either  $\mathbf{T}_1\boldsymbol{\eta}_1$  or  $\mathbf{T}_2\boldsymbol{\eta}_2$  conditionally decorrelates  $\boldsymbol{\eta}_1$  and  $\boldsymbol{\eta}_2$  [56]. That is,

$$E[\boldsymbol{\eta}_1\boldsymbol{\eta}_2^T | \mathbf{T}_1\boldsymbol{\eta}_1] = E[\boldsymbol{\eta}_1 | \mathbf{T}_1\boldsymbol{\eta}_1] E[\boldsymbol{\eta}_2 | \mathbf{T}_1\boldsymbol{\eta}_1]. \quad (2.35)$$

The matrices  $\mathbf{T}_1$  and  $\mathbf{T}_2$  are computed via a singular value decomposition (SVD). The associated singular values are the entries of  $\hat{\mathbf{D}}$  and order the state variables by their relative significance in decorrelating  $\boldsymbol{\eta}_1$  and  $\boldsymbol{\eta}_2$ . Varying levels of approximation can be obtained naturally by discarding rows of  $\mathbf{T}_1$  and  $\mathbf{T}_2$  associated with the least

significant singular values.

To conditionally decorrelate more than two random vectors, e.g., to decorrelate  $\boldsymbol{\eta}_1$ ,  $\boldsymbol{\eta}_2$ , and  $\boldsymbol{\eta}_3$ , first compute a  $\mathbf{T}_1\boldsymbol{\eta}_1$  that conditionally decorrelates  $\boldsymbol{\eta}_1$  from  $[\boldsymbol{\eta}_2^T \boldsymbol{\eta}_3^T]^T$ . Similarly, compute  $\mathbf{T}_2\boldsymbol{\eta}_2$  that decorrelates  $\boldsymbol{\eta}_2$  from  $[\boldsymbol{\eta}_1^T \boldsymbol{\eta}_3^T]$ . The resulting

$$\begin{bmatrix} \mathbf{T}_1\boldsymbol{\eta}_1 \\ \mathbf{T}_2\boldsymbol{\eta}_2 \end{bmatrix} \quad (2.36)$$

then conditionally decorrelates  $\boldsymbol{\eta}_1$ ,  $\boldsymbol{\eta}_2$ , and  $\boldsymbol{\eta}_3$ . This concept extends naturally to  $q$ -adic trees. In this case, the set of state variables defined at node  $s$  must decorrelate the  $q + 1$  sets of variables, namely  $\mathbf{x}(s\alpha_i)$ ,  $i = 1, \dots, q$ , and  $\mathbf{x}^c(s)$ . A  $\mathbf{T}_i\mathbf{x}(s\alpha_i)$  that conditionally decorrelates  $\mathbf{x}(s\alpha_i)$  from the remaining  $q$  vectors is computed for each of the  $\mathbf{x}(s\alpha_i)$ . The linear functionals  $\mathbf{L}(s)$  at node  $s$  is then the collection of these  $\mathbf{T}_i$ .

Reduced-order models obtained by discarding some of the least significant functionals do not perform the conditional decorrelation of (2.35) perfectly. That is, the Markovianity property of the multiscale tree is not exactly satisfied. Consequently, the realized covariance structure at the finest scale  $\mathbf{P}_\mathbf{x}$  will not exactly match the true covariance  $\mathbf{P}_\mathbf{x}$  of the random field to be realized. Reduced-order models with lower state dimensions are of course computationally less costly for the multiscale estimation algorithms. The trade-off between model simplicity and fidelity generally depends on the correlation structure of the random process to be realized.

Computing the linear functionals  $\mathbf{L}(s)$  at each node involves  $q$  singular value decompositions (except at the root where only  $q - 1$  sets of variables need to be decorrelated), each of which is an SVD on the order of  $N$ . As a result, canonical correlations realization is prohibitively expensive for modeling random processes of large size. Linear functionals are computed non-recursively at each individual node, so the number of SVDs grows linearly with  $N$ , where  $N$  is the total number of variables at the finest scale. Thus, building a multiscale model using CCR carries a computational complexity of  $\mathcal{O}(N^4)$ . To lessen the computational cost, approximation techniques [56] that exploit symmetry or short correlation length of the random

process to be realized are available for special cases. However, canonical correlations realization, while conceptually general, is still too computationally expensive to be practical. For the purpose of this thesis, CCR is used only as a tool for analyzing the random processes or fields we face. The intuition gained from the analysis guides us in designing the appropriate linear functionals for modeling the random processes in which we are interested. CCR is bypassed in the actual construction of multiscale models.

A new realization method currently under investigation [40, 41, 51] deals with the drawbacks of CCR. Instead of attempting to find a  $\mathbf{T}_1$  that minimizes the residual correlation between  $\boldsymbol{\eta}_1$  and  $\boldsymbol{\eta}_2$  conditioned on  $\mathbf{T}_1\boldsymbol{\eta}_1$  as in (2.33), the new method attempts to find a  $\mathbf{T}_1$  that minimizes the mean square estimation error in estimating  $\boldsymbol{\eta}_2$  based on  $\mathbf{T}_1\boldsymbol{\eta}_1$ . In designing the states at node  $s$ , it also attempts to decorrelate the states at the child nodes  $\mathbf{x}(s\alpha_i)$  rather than the entire finest-scale process  $\mathbf{x}(s\alpha_i)$ . The method therefore can achieve  $\mathcal{O}(N^2)$  computational complexity overall.

### 2.3.4 Conditional Decorrelation and Model Consistency

There is one subtle but important point about the state definition of internal models — (2.20), which equates  $\mathbf{x}(s)$  and  $\mathbf{L}(s)\mathbf{x}$  as the same random variables, is not true for an arbitrary set of  $\mathbf{L}(s)$ , even though the way the model parameters  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$  are computed in (2.18) and (2.19) does indeed ensure that the covariance of  $\mathbf{x}(s)$  and the cross-covariance between  $\mathbf{x}(s)$  and  $\mathbf{x}(s\bar{\gamma})$  are exactly realized, i.e., (2.21) and (2.22) are guaranteed to be true. For  $\mathbf{P}_{\mathbf{x}} = \mathbf{P}_{\mathbf{x}}$  to be true, the linear functionals must satisfy a conditional decorrelation, or tree Markovianity, requirement. For  $\mathbf{x}(s)$  and  $\mathbf{L}(s)\mathbf{x}$  to be the same random variables, which is what the term internal means, the linear functionals must satisfy a consistency requirement [25]. The choice of linear functionals cannot be arbitrary. If they do not satisfy the conditional decorrelation requirement, then the resulting multiscale model is not exact; if they do not satisfy the consistency requirement, then the model is not internal.

Take the very simple tree model in Figure 2-7 as an example, where the root node  $x(0) = \chi_1 + \chi_2$  and its children are defined as  $x(1) = \chi_1$  and  $x(2) = \chi_2$ . Given the

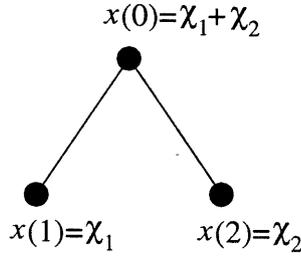


Figure 2-7: An example tree that does not satisfy the conditional decorrelation requirement.

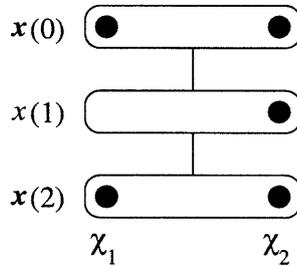


Figure 2-8: An example tree that does not satisfy the consistency requirement.

sum of  $\chi_1$  and  $\chi_2$ , the two variables are conditionally correlated. That is, if we treat  $\chi_1 + \chi_2$  as a measurement and estimate  $\chi_1$  and  $\chi_2$ , the estimation errors are correlated. In other words, the process noises  $w(1)$  and  $w(2)$  in  $x(1) = A(1)x(0) + w(1)$  and  $x(2) = A(2)x(0) + w(2)$  are correlated. A multiscale tree model with states defined as in the figure will treat  $w(1)$  and  $w(2)$  as uncorrelated and thus will realize  $P(1,2) = A(1)P(0)A(2)$  for the covariance between  $\chi_1$  and  $\chi_2$ , even though the correct covariance is  $A(1)P(0)A(2) + E[w(1)w(2)]$ . For multiscale realizations to be exact, i.e.,  $\mathbf{P}_x = \mathbf{P}_\chi$ , the linear functionals (in this case the linear functional at the root node) must be chosen so that the resulting  $w(1)$  and  $w(2)$  are indeed uncorrelated. Exact multiscale models perform the conditional decorrelation perfectly. Approximate CCR realization models [56] order the linear functionals so that for the given state dimension the resulting correlated noise term  $E[w(1)w(2)]$  is minimized.

The description of the multiscale model is complete once the model parameters  $\mathbf{A}(s)$ ,  $\mathbf{B}(s)$ , and  $\mathbf{P}(0)$  have been specified. The model does not automatically ensure that  $\mathbf{x}(s)$  are the same random variables as  $\mathbf{L}(s)\mathbf{x}$ . Consider the simple example of a monadic tree in Figure 2-8, where the states at node 0 and node 2, are defined with the

same linear functionals, i.e.,  $\mathbf{L}(0) = \mathbf{L}(2)$ . Although  $\mathbf{P}(0) = \mathbf{P}(2) = \mathbf{P}_\chi$ , where  $\mathbf{P}_\chi$  is the covariance of the random field of interest placed at node 2, the first state variables at nodes 0 and 2, denoted  $x_1(0)$  and  $x_1(2)$  respectively, are two different random variables because of the information loss going through node 1. For instance, the sample paths generated for  $x_1(0)$  and  $x_1(2)$  will be different. The issue of consistency becomes acute if we are to place a measurement of  $\chi_1$  onto the tree. If we believe (2.20), we have two equal choices:  $y(0) = [1 \ 0] \mathbf{x}(0) + v$  or  $y(2) = [1 \ 0] \mathbf{x}(2) + v$ . However, for the example in Figure 2-8, the two choices yield different estimation results. In either case, the estimates of  $x_1(0)$  and  $x_1(2)$  are different, because  $x_1(0)$  and  $x_1(2)$  are in fact two distinct random variables despite what (2.20) may lead one to believe. The model has to be consistent and no information must be lost passing from node to node, in order for  $x_1(0)$  and  $x_1(2)$  to be the same random variable.

Inconsistent models may be exact. For the example in Figure 2-8, the desired covariance is realized exactly at the finest scale. As long as measurements are not placed anywhere on the tree where the state variables are inconsistent, exact estimation results are obtained. Placing the measurement at node 2 yields the correct estimates, i.e., same as LLSE, for  $\chi_1$  and  $\chi_2$ ; placing the measurement at node 0 does not.

The issues of conditional decorrelation and consistency become important when additional states are to be augmented onto an existing tree model [25] in order to place non-local measurements that are not readily placeable at the finest scale, to which the process of interest is explicitly mapped. To preserve the conditional decorrelation property, it is sufficient to make sure that a linear functional augmented at node  $s$  has support over only an individual  $\mathbf{x}(s\alpha_i)$ , not the entire  $\mathbf{x}(s)$ . To preserve consistency, additional states should be augmented at other nodes to ensure the necessary information is propagated without loss to the descendents of node  $s$  at the finest scale [25]. For the simple example in Figure 2-8, this means augmenting  $\chi_1$  as a state at node 1. Since models whose coarser-scale states are not linear functionals of the finest-scale states are external, the simple example in Figure 2-8 is external; that is, even though (2.21) and (2.22) are true, (2.20) is not.

### 2.3.5 External Models

The most important class of multiscale external models developed to date are the so-called  $1/f$  models whose development antecedes that of internal models and draws heavily from ideas of the wavelet transform.

The term multiscale has been used loosely to mean a variety of different things. For instance, it could mean a divide-and-conquer approach to solving image processing problems [43, 42]. The term, however, brings up in many people's mind the wavelet transform, which is deterministic in most applications. We have used the term multiscale to mean stochastic processes that are organized by scale where states situated at coarser-scale nodes conditionally decorrelates states situated at finer-scale nodes. Internal realization methods focus very much on how to define the states that conditionally decorrelate well. Yet the notion of scale there is not entirely intuitive. The development of  $1/f$  models makes clear the connection between our stochastic models and the wavelet transform.

Using orthonormal wavelet transforms [26, 81], the multiresolution approximation of a signal  $f(t)$  at scale  $m$ , is a weighted sum of shifted and scaled versions of the basic scaling function  $\phi(\cdot)$ :

$$f_m(t) = \sum_{n=-\infty}^{+\infty} c_m[n] \phi(2^m t - n), \quad (2.37)$$

where  $c_m[n]$  are the scaling or approximation coefficients, and scale is indexed by  $m$ . The scaling functions are related according to the dilation equation,  $\phi(t) = \sum_n h[n] \phi(2t - n)$ . The associated wavelets are related to the scaling functions through the wavelet equation,  $\psi(t) = \sum_n g[n] \phi(2t - n)$ . The  $h[n]$  and  $g[n]$  form a conjugate mirror filter pair for orthonormal wavelets [81]. The approximation of the signal  $f(t)$  at one scale can be expressed as the sum of its approximation at one coarser scale and new details, i.e.,  $f_{m+1}(t) = f_m(t) + \sum_n d_m[n] \psi(2^m t - n)$ . The *approximation* coefficients  $c_m[n]$  and the *detail* or wavelet coefficients  $d_m[n]$  of the signal can be computed recursively in scale from fine to coarse through the analysis equations,  $c_m[n] = \sum_k h[2n - k] c_{m+1}[k]$  and  $d_m[n] = \sum_k g[2n - k] d_{m+1}[k]$ . Recursion from coarse

to fine scale is through the synthesis equation:

$$c_{m+1}[n] = \sum_k h[2k - n]c_m[k] + \sum_k g[2k - n]d_m[k]. \quad (2.38)$$

Multiscale estimation theory views the wavelet transform synthesis equation (2.38) as a dynamic relationship between scaling coefficients at one scale and those at the next. This relationship can be represented on an infinite lattice [20]. In the simplest case of the Haar wavelet transform, the lattice becomes a dyadic tree. The multiscale estimation framework relies on the assumption that the wavelet coefficients are uncorrelated from scale to scale and along a scale, as the recursive multiscale smoothing algorithm require this condition. For stochastic processes whitened by the wavelet transform such that  $d_m[n]$  at a particular scale are white and uncorrelated with  $c_m[n]$  at coarser scales, (2.38) represents a first-order recursion in scale that is driven by white noise,  $d_m[n]$ . It has been shown that fractional Brownian motion has approximately uncorrelated wavelet coefficients [37, 38, 84]. A broader class of  $1/f$  processes that display  $1/f^\mu$ -like spectra over a range of frequencies can also be synthesized with uncorrelated wavelet coefficients [61]. For this class of processes the wavelet transform acts like a Karhunen-Loève expansion, whitening the processes [92]. Analogously, the Fourier transform whitens stationary processes so that the individual frequency components of such processes are statistically uncorrelated. Frequency-domain operations such as Wiener filtering can then be applied.

In the simplest case, states on the  $1/f$  multiscale model are scalar, and the process of interest is placed at the finest scale. The state at any node is simply its parent node, i.e.,  $A(s) = 1$ , plus some injected white process noise, whose strength decreases exponentially as a function of scale:

$$x(s) = x(s\bar{\gamma}) + B_\circ 2^{(1-\mu)m(s)/2} w(s), \quad (2.39)$$

where  $m(s)$  is the scale of node  $s$  ( $m(s) = 0$  at the root node and increases as we move to finer scales), and where  $B_\circ$  and  $\mu$  controls the magnitude and the rate of decrease

of the process noise, respectively. The process noise term captures the self-similar scaling properties in that the variability of the  $1/f^\mu$  process scales geometrically with the spatial resolution at which the variations are measured [33]. A variant of (2.39), in which a higher-order model with state dimension equal to two is used to capture some of the correlations in the detail coefficients, is used in [35] to model fractional Brownian motion.

Whereas for internal models the states, defined in terms of the process of interest, have some physical meaning, e.g., a weighted average of some subset of elements of the process of interest, the meaning of the state at the coarser-scale nodes on  $1/f$  models are more abstract. For the model in (2.39), the variable at the root node may be thought of as the aggregate average over the entire region of support, although it is an average only in an abstract sense. States at coarser scales on external models, by definition, cannot be written in the form of (2.20). Moving down to finer scales, details are added as process noise  $w(s)$ . The state  $x(s)$  can be viewed as the stochastic analog of the scaling coefficients and  $w(s)$  the detail coefficients [20].

Multiscale  $1/f$  models have been successfully employed in a variety of applications. The low state dimension of the  $1/f$  models means that extremely large 2-D estimation problems can be solved very fast. The earliest application was to the problem of estimating optical flow [66], where the  $1/f$  multiscale models is used as a regularization term for dealing with the ill-posedness of the optical flow computation. In [33],  $1/f$  models were used for mapping ocean surface height in the Pacific where maps with 250,000 grid points were generated in less than a minute on the then current generation of workstations. Modified version of the same type of models will be used to model surface height anomaly signal in the Mediterranean Sea [34] in Chapter 6. Multiscale  $1/f$  models have also been used as prior models in variational surface reconstruction [32] and image segmentation [78, 76, 60].

## 2.4 Concluding Remarks

A rich body of research on multiscale estimation and modeling has been accumulated in the last ten years. In this chapter, we have briefly described those parts of the past research that are relevant to this thesis. However, the wide array of topics and papers has obscured the fact that there is considerable continuity in the theoretical development as well as the application of this multiscale framework. As a conclusion to this chapter, we will group the major topics roughly into three categories, name the main results in each, and point the reader to the relevant papers. The following list is by no means exhaustive, but it should give the reader somewhat of an index to the available literature on our multiscale framework. (Journal papers are listed where possible. More detailed exposition on a topic is usually found in the thesis by the same author.)

### Multiscale models and algorithms

- Earlier research on multiscale autoregressive model for parameterization and whitening of isotropic processes, motivated in part by the wavelet transform — Basseville, Benveniste, Nikoukhah, and Willsky [7, 8, 6].
- Multiscale model and the smoothing algorithm in the present form — Chou [18].
- System issues such as stability, reachability, observability, and steady state associated with the multiscale smoothing algorithm — Chou [19].
- Smoothing error model — Luetgen [69].
- Multiscale likelihood test
  - Algorithmic development — Luetgen [68].
  - Texture discrimination — Luetgen [68].
  - Parameter identification of fractional Brownian motion — Fieguth [35].
  - Ocean surface model parameter identification — Fieguth [33, 34].

- Overlapping trees for the removal of blocky artifacts in images due to low state dimension — Irving and Fieguth [57].

### **1/f models (External models)**

- Fractional Brownian motion — Fieguth [35].
- 1/f prior for optical flow field — Chou [18], Luetzgen [66]
- Membrane and thin-plate variational constraints interpreted as 1/f<sup>2</sup> prior statistical models for surface reconstruction — Fieguth [32].
- Ocean surface model — Fieguth [33, 34].
- Image segmentation — Schneider [78].
- Synthetic aperture radar image modeling, segmentation, and compression — Irving [58], Fosgate [39], Kim [62].

### **Internal models**

- Theoretical development
  - Models for MRFs — Luetzgen [67].
  - Canonical correlations realization — Irving [56].
  - State augmentation and non-local measurements — Daniel [25].
  - Fast realization algorithm — Frakt [40].
- Applications
  - Texture — Luetzgen [67].
  - Fractional Brownian motion — Daniel [24].
  - Ocean temperature — Fieguth [73, 31].



# Chapter 3

## Multiscale Time-Recursive Estimation

### 3.1 Introduction

Estimation of dynamic systems governed by partial differential equations (PDEs), i.e., distributed parameter systems, such as those found in applications ranging from pollution control [74] to modeling ecological systems and flexible structures [5], are of considerable scientific, as well as practical, interest. While there have been successful applications of the theory of optimal estimation for such systems, there are severe computational barriers that limit the domain in which truly optimal methods can be implemented. Numerical approaches to solving distributed parameter systems have been limited largely due to the high-dimensionality resulting from approximating such systems by finite-dimensional lumped systems [3]. 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 [33], the dimensionality of finite-dimensional approximations to the underlying dynamics can range from hundreds of thousands to hundreds of millions.

Conventional linear least squares estimation (LLSE) algorithms, such as Kalman filtering, are completely impractical for solving such large problems both for compu-

tational 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 [11], 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 estimation algorithms that can deal effectively with the computational challenge. Consider the standard recursive estimation structure of alternating prediction and measurement update steps. In standard Kalman filtering, the propagation of the estimation error statistics is done *explicitly* via the estimation error covariance matrix, an object with  $\mathcal{O}(N^2)$  elements for an  $N$ -dimensional state. The computational complexity of the Riccati equation for the propagation of the error covariance matrices through the successive prediction and update steps is  $\mathcal{O}(N^3)$  — far too large for systems with large  $N$ . The key to computationally more efficient estimation algorithms is finding a compact and effective representation for the statistics of the estimation error that *implicitly* specifies these error statistics in the form of a compact *model*. There are two major challenges in realizing this concept:

- (i) finding an appropriate random field model class that both accurately represents the error statistics and also leads to substantial reduction in complexity, and
- (ii) developing a replacement for the Riccati equation either for the propagation of this error model as it changes after each prediction and measurement step or for the evaluation of a steady-state error model.

In [15, 16, 14, 13] such an approach was developed using Markov random field (MRF) models 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. These models have several attractive features with regard to the challenges mentioned previously: they

are compact, typically requiring  $\mathcal{O}(N)$  storage, and there are several  $\mathcal{O}(N)$  algorithms for the propagation of these models. However, the actual solution of the spatial estimation problem for each measurement update using such a model is not nearly as efficient, requiring up to  $\mathcal{O}(N^2)$  computations for the calculation of estimates and the spatial error variances.

The class of multiscale models of the form (2.6) have been shown to be rich, not only yielding accurate representations in problems in which MRFs have commonly been used [67] but also in other contexts, including the modeling and analysis of  $1/f$  processes and the modeling of large distributed phenomena for remote sensing applications in oceanography [33, 34] and hydrology [25]. In applications such as mapping ocean surface height [33, 34] and computing optical flow fields [66, 15], even though the underlying physical systems are *dynamic* in such applications, the estimation problems were treated as *static* estimation problems by considering one snapshot of the system at a time. The flexibility, computational simplicity, and proven value of these multiscale models for large, but static, spatial estimation problems provide the motivation for the use of multiscale spatial models in dynamic estimation problems.

The key challenge in developing this formalism is the development of a method to propagate a multiscale model of the estimation errors over time. Each measurement update step in the standard recursive estimation structure has a straightforward interpretation as solving a static estimation problem, namely the estimation of the error in the predicted estimate based on the innovation for the current measurements. Propagation through the measurement update step is easy and is accomplished as a by-product of the multiscale estimation algorithm [69]. It is the propagation of the multiscale model through the prediction step that remains, and the challenge here is to untangle the mixing effect of the dynamics in updating the parameters of the model. Doing this requires a careful examination of exactly how the temporal dynamics mix the variables captured at each scale in the multiscale representation.

In the following sections, we first describe the basics of recursive estimation and Kalman filtering. We then outline the general approach of the multiscale dynamic

estimation algorithm and develop in more detail the multiscale prediction step. We illustrate the prediction algorithm with some simple examples in 1-D. We propose a new class of non-redundant models that are particularly suitable for modeling MRFs in the dynamic estimation context. In the next chapter, we apply the multiscale dynamic estimation algorithm to 1-D diffusion processes with both internal models and non-redundant models, and in Chapter 5 to 2-D diffusion processes.

## 3.2 Recursive Estimation of Dynamic Systems

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

$$\mathbf{z}(t+1) = \mathbf{A}_d \mathbf{z}(t) + \mathbf{w}_d(t), \quad (3.1)$$

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

$$\mathbf{y}_d(t) = \mathbf{C}_d(t) \mathbf{z}(t) + \mathbf{v}_d(t), \quad (3.2)$$

where  $\mathbf{v}_d(t)$  is the Gaussian measurement noise with zero mean and diagonal covariance  $\mathbf{R}_d$ . The temporal dynamics, process noise, and measurement noise are assumed to be stationary in time, i.e.,  $\mathbf{A}_d$ ,  $\mathbf{Q}_d$ , and  $\mathbf{R}_d$  are independent of  $t$ . For the applications we have in mind such as diffusion processes (c.f., Chapter 4)  $\mathbf{z}(t)$  would represent a spatially-discretized distributed parameter process and (3.1) would represent the corresponding temporally-discretized dynamics, so that  $\mathbf{A}_d$  represents the discretization of a partial differential operator in space. In addition, in the problems of interest here  $\mathbf{C}_d$  is a selection matrix<sup>1</sup>, and  $\mathbf{R}_d$  is diagonal, so that the components of  $\mathbf{y}_d$  represent independent point measurements of the distributed process.

Let  $\hat{\mathbf{z}}(t|\tau)$  denote the estimate<sup>2</sup> of  $\mathbf{z}(t)$  based on measurements through time  $\tau$ ,

---

<sup>1</sup>We use the term selection matrix to denote a matrix in which each row has only one nonzero element. By appropriate scaling the components of  $\mathbf{y}_d(t)$  and  $\mathbf{R}_d$  we can then, without loss of generality, assume that each nonzero value in  $\mathbf{C}_d$  equals 1.

<sup>2</sup>Linear least squares estimators require only second-order statistics of random variables. In this thesis, estimates and estimators involve only second-order statistics, and statistical stationarity implies wide sense stationarity.

and let  $\boldsymbol{\chi}(t|\tau)$  denote the corresponding estimation error

$$\boldsymbol{\chi}(t|\tau) = \mathbf{z}(t) - \hat{\mathbf{z}}(t|\tau). \quad (3.3)$$

The form of a class of recursive estimators (which includes the optimal Kalman filter) consists of a prediction stage

$$\hat{\mathbf{z}}(t+1|t) = \mathbf{A}_d \hat{\mathbf{z}}(t|t), \quad (3.4)$$

and a measurement update stage

$$\hat{\mathbf{z}}(t|t) = \hat{\mathbf{z}}(t|t-1) + \hat{\boldsymbol{\chi}}(t|t-1), \quad (3.5)$$

where  $\hat{\boldsymbol{\chi}}(t|t-1)$  is the estimate of the one-step prediction error  $\boldsymbol{\chi}(t|t-1)$  based on the innovation

$$\boldsymbol{\nu}(t) = \mathbf{y}_d(t) - \mathbf{C}_d \hat{\mathbf{z}}(t|t-1) \quad (3.6)$$

$$= \mathbf{C}_d \boldsymbol{\chi}(t|t-1) + \mathbf{v}_d(t). \quad (3.7)$$

Each measurement update step is essentially a static estimation problem of computing  $\hat{\boldsymbol{\chi}}(t|t-1)$ , but it requires the statistics of  $\hat{\boldsymbol{\chi}}(t|t-1)$ , which must be propagated through time.

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

$$\hat{\boldsymbol{\chi}}(t|t-1) = \mathbf{K}(t)\boldsymbol{\nu}(t), \quad (3.8)$$

where

$$\mathbf{K}(t) = \mathbf{P}_\chi(t|t-1)\mathbf{C}_d^T \left( \mathbf{C}_d \mathbf{P}_\chi(t|t-1)\mathbf{C}_d^T + \mathbf{R}_d \right)^{-1} \quad (3.9)$$

is the Kalman filter gain. Computing  $\hat{\boldsymbol{\chi}}(t|t-1)$  requires the availability of the predicted estimation error covariance  $\mathbf{P}_\chi(t|t-1)$ , which is propagated through time from

$\mathbf{P}_\chi(t|t-1)$  to  $\mathbf{P}_\chi(t+1|t)$  via the Riccati equation:

$$\begin{aligned} \mathbf{P}_\chi(t|t) &= \mathbf{P}_\chi(t|t-1)\mathbf{C}_d^T \left( \mathbf{C}_d\mathbf{P}_\chi(t|t-1)\mathbf{C}_d^T + \mathbf{R}_d \right)^{-1} \\ &\quad \mathbf{C}_d\mathbf{P}_\chi(t|t-1) \end{aligned} \quad (3.10)$$

$$\mathbf{P}_\chi(t+1|t) = \mathbf{A}_d\mathbf{P}_\chi(t|t)\mathbf{A}_d^T + \mathbf{Q}_d. \quad (3.11)$$

Note that (3.8) and (3.9) are of exactly the same form as the static LLSE equation in (2.2). The updated estimation error covariance  $\mathbf{P}_\chi(t|t)$  is exactly the *a posteriori* estimation error covariance from estimating  $\boldsymbol{\chi}(t|t-1)$ .

For problems of interest to us the dimensionality of  $\boldsymbol{\chi}(t|t-1)$  makes this explicit Kalman filter calculation either impossible or at best exceedingly complex. Under general conditions,

- (i) computing the predicted estimate in (3.4) is  $\mathcal{O}(N^2)$ ;
- (ii) computing the estimate of the predicted estimation error,  $\hat{\boldsymbol{\chi}}(t|t-1)$ , in (3.8) and (3.9) is  $\mathcal{O}(N^3)$ ;
- (iii) propagating the estimation error covariances through the Riccati equation in (3.10) and (3.11) is  $\mathcal{O}(N^3)$ .

For systems that approximate PDEs, the temporal dynamic matrix  $\mathbf{A}_d$  is a discretized differential operator and may be sparse and banded, depending on the discretization scheme. Computing the predicted estimate then becomes  $\mathcal{O}(N)$ . However, difficulties with (ii) and (iii) remain.

Suboptimal recursive estimators may compute  $\hat{\boldsymbol{\chi}}(t|t-1)$  and propagate the estimation error statistics in any number of ways, often aimed to achieve better computational complexity by taking advantage of additional structure in the estimation errors. The suboptimal estimator may compute the estimates approximately, or propagate the error statistics approximately, or both. Note that an equivalent filter gain  $\mathbf{G}(t)$  can be calculated for any linear suboptimal estimator if necessary. Since the estimator is a linear system with input  $\boldsymbol{\nu}(t)$  and output  $\hat{\boldsymbol{\chi}}(t|t-1)$  as in (3.8), the  $i$ -th column of  $\mathbf{G}(t)$  is the estimation result for  $\boldsymbol{\nu}(t) = \mathbf{e}_i$ , where  $\mathbf{e}_i$  is a column vector of

all zeros except a one at the  $i$ -th element. The temporal dynamics of the updated and the predicted estimation errors of the linear recursive estimator are then governed by

$$\boldsymbol{\chi}(t|t) = (\mathbf{I} - \mathbf{G}(t)\mathbf{C}_d) \boldsymbol{\chi}(t|t-1) - \mathbf{G}(t)\mathbf{v}_d(t) \quad (3.12)$$

$$\boldsymbol{\chi}(t+1|t) = \mathbf{A}_d \boldsymbol{\chi}(t|t) + \mathbf{w}_d(t), \quad (3.13)$$

determined after some algebraic manipulation<sup>3</sup>. Therefore, the updated and the predicted estimation error covariances of the suboptimal estimator satisfy the following:

$$\begin{aligned} \mathbf{P}_\chi(t|t) &= (\mathbf{I} - \mathbf{G}(t)\mathbf{C}_d) \mathbf{P}_\chi(t|t-1) (\mathbf{I} - \mathbf{G}(t)\mathbf{C}_d)^T \\ &\quad + \mathbf{G}(t)\mathbf{R}_d\mathbf{G}(t)^T \end{aligned} \quad (3.14)$$

$$\mathbf{P}_\chi(t+1|t) = \mathbf{A}_d \mathbf{P}_\chi(t|t) \mathbf{A}_d^T + \mathbf{Q}_d. \quad (3.15)$$

The suboptimal computation of  $\hat{\boldsymbol{\chi}}(t|t-1)$ , may be based on error statistics (or some implicit representation of the error statistics) propagated incorrectly, i.e., the error statistics the suboptimal estimator believes and uses in its recursion are different from the actual error statistics in (3.14) and (3.15). Only the optimal estimator, e.g., the Kalman filter for which  $\mathbf{G}(t) = \mathbf{K}(t)$ , correctly propagates the error statistics, i.e., (3.10) gives the same answer as (3.14), and (3.11) the same as (3.15). More will be said in Section 3.6.

---

<sup>3</sup>For the updated estimation error:

$$\begin{aligned} \boldsymbol{\chi}(t|t) &= \mathbf{z}(t) - \hat{\mathbf{z}}(t|t) \\ &= \mathbf{z}(t) - \hat{\mathbf{z}}(t|t-1) - \hat{\boldsymbol{\chi}}(t|t-1) \\ &= \boldsymbol{\chi}(t|t-1) - \mathbf{G}(t)\boldsymbol{\nu}(t) \\ &= \boldsymbol{\chi}(t|t-1) - \mathbf{G}(t)\mathbf{C}_d\boldsymbol{\chi}(t|t-1) - \mathbf{G}(t)\mathbf{v}_d(t) \\ &= (\mathbf{I} - \mathbf{G}(t)\mathbf{C}_d) \boldsymbol{\chi}(t|t-1) - \mathbf{G}(t)\mathbf{v}_d(t) \end{aligned}$$

For the predicted estimation error:

$$\begin{aligned} \boldsymbol{\chi}(t+1|t) &= \mathbf{z}(t+1) - \hat{\mathbf{z}}(t+1|t) \\ &= \mathbf{A}_d\mathbf{z}(t) + \mathbf{w}_d(t) - \mathbf{A}_d\hat{\mathbf{z}}(t|t) \\ &= \mathbf{A}_d\boldsymbol{\chi}(t|t) + \mathbf{w}_d(t) \end{aligned}$$

### 3.2.1 MRF-Based Recursive Estimation

Markov random field models have been used to model the errors in estimating optical flow [15, 14, 13] and geophysical flow fields [17], for which the dimension of the field of interest  $\mathbf{z}(t)$  is usually large. As we have said in Section 2.3.2, the inverse of the covariance matrix of a Markov random field is sparse. It then makes sense to express the error statistics not in terms of an error covariance matrix but in terms of the inverse of the error covariance matrix. The information form of the Kalman filter is used instead of the standard form in (3.8), (3.9), (3.10) and (3.11). The basic approach of the algorithm is the following. (See also Section 7.2.3.)

Propagating the error statistics through the update step is computationally efficient:

$$\mathbf{P}_\chi^{-1}(t|t) = \mathbf{P}_\chi^{-1}(t|t-1) + \mathbf{C}_d^T \mathbf{R}_d^{-1} \mathbf{C}_d, \quad (3.16)$$

if  $\mathbf{P}_\chi^{-1}(t|t-1)$  and  $\mathbf{C}_d^T \mathbf{R}_d^{-1} \mathbf{C}_d$  are sparse and block tridiagonal, corresponding to a first-order 2-D MRF. This is the case in the optical flow and the geophysical flow applications and in our problems (c.f., Chapters 4 and 5), where the prediction error is approximated as a MRF such that  $\mathbf{P}_\chi^{-1}(t|t-1)$  is sparse, and point or local measurements and white measurement noise result in a sparse  $\mathbf{C}_d^T \mathbf{R}_d^{-1} \mathbf{C}_d$ . The sparse and block tridiagonal structure is preserved in  $\mathbf{P}_\chi^{-1}(t|t)$ . Computing the updated estimate in (3.5) and (3.8) is done via a normal equation

$$\mathbf{P}_\chi^{-1}(t|t) \hat{\boldsymbol{\chi}}(t|t) = \mathbf{C}_d^T \mathbf{R}_d^{-1} \boldsymbol{\nu}(t). \quad (3.17)$$

The sparseness of  $\mathbf{P}_\chi^{-1}(t|t)$  allows us to use efficient iterative procedures, such as multigrid [11], to solve for  $\hat{\boldsymbol{\chi}}(t|t)$ .

Propagating the error statistics through the prediction step as expressed in (3.11), however, requires  $\mathbf{P}_\chi(t|t)$ . Remember that only the inverse covariance matrices  $\mathbf{P}_\chi^{-1}(t|t)$  and  $\mathbf{P}_\chi^{-1}(t|t-1)$  are sparse. The covariance matrices themselves are generally

full, and inverting  $\mathbf{P}_x^{-1}(t|t)$  is to be avoided. Written in the inverse form,

$$\mathbf{P}_x^{-1}(t+1|t) = \mathbf{Q}_d^{-1} - \mathbf{Q}_d^{-1} \mathbf{A}_d \left( \mathbf{A}_d^T \mathbf{Q}_d^{-1} \mathbf{A}_d + \mathbf{P}_x^{-1}(t|t) \right)^{-1} \mathbf{A}_d^T \mathbf{Q}_d^{-1}. \quad (3.18)$$

In applications where  $\mathbf{A}_d$  is sparse and  $\mathbf{Q}_d$  is diagonal, computing the predicted estimate in (3.4) is easy. On the other hand, propagating the error statistics through the prediction step in (3.18) requires a matrix inversion, and the sparse structure of  $\mathbf{P}_x^{-1}(t|t)$  is not preserved in  $\mathbf{P}_x^{-1}(t+1|t)$ . That is,  $\mathbf{P}_x^{-1}(t+1|t)$  is no longer Markov. For the case of the optical flow computation, where  $\mathbf{A}_d = \mathbf{I}$  and  $\mathbf{Q}_d = q\mathbf{I}$ :

$$\mathbf{P}_x^{-1}(t+1|t) = q^{-1} \mathbf{I} - q^{-2} \left( q^{-1} \mathbf{I} + \mathbf{P}_x^{-1}(t|t) \right)^{-1}. \quad (3.19)$$

An approximation method is used in [15, 14] to impose a sparse structure on  $\mathbf{P}_x^{-1}(t+1|t)$  by approximating the matrix inversion with an infinite series:

$$\mathbf{S}^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{\Omega} \mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{\Omega} \mathbf{D}^{-1} \mathbf{\Omega} \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{\Omega} \mathbf{D}^{-1} \mathbf{\Omega} \mathbf{D}^{-1} \mathbf{\Omega} \mathbf{D}^{-1} + \dots, \quad (3.20)$$

where  $\mathbf{S} = q^{-1} \mathbf{I} + \mathbf{P}_x^{-1}(t|t)$ ,  $\mathbf{D}$  and  $\mathbf{\Omega}$  are respectively the the block diagonal and off-diagonal parts of  $\mathbf{S}$ , such that  $\mathbf{S} = \mathbf{D} + \mathbf{\Omega}$ . The more terms in the infinite series are kept, the denser the approximated matrix is, and the more accurate the approximation is. Near-optimal estimation performance was obtained by keeping just the first two terms in (3.20) for the optical flow applications<sup>4</sup>.

The MRF models and the associated suboptimal recursive estimation method have several desirable features:

- The MRF models for the estimation errors are compact, requiring  $\mathcal{O}(N)$  storage. The statistics of the errors are expressed implicitly in terms of the inverse of the error covariances.
- The propagation of the error statistics is  $\mathcal{O}(N)$ . Propagating the error statistics

---

<sup>4</sup>In the case of the optical flow application [15], as well as others, the comparison with the optimal estimator can only be done for small problems. The whole point of using suboptimal methods is that for large systems the optimal estimator is computational infeasible.

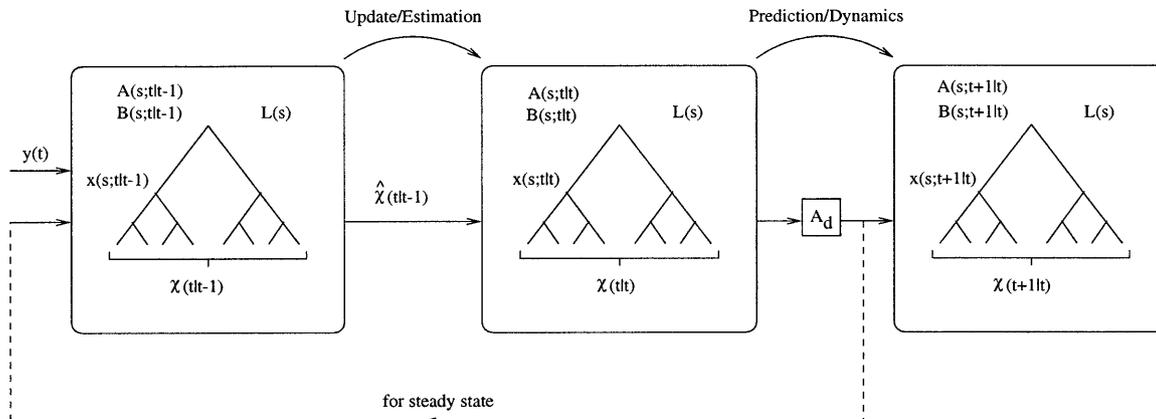


Figure 3-1: A schematic of the multiscale recursive estimation method.

through the update step is an exact calculation, although approximations must be made for the prediction step.

- Computing the updated estimates is approximate and requires iterative procedures that may reach up to  $\mathcal{O}(N^2)$  complexity. Computing the predicted estimates is simple with  $\mathcal{O}(N)$  complexity.

### 3.3 Multiscale Recursive Estimation

The alternative approach that we consider in this thesis involves the *implicit* calculation and propagation of the statistics of the estimation errors as a sequence of multiscale models, as illustrated in Figure 3-1, similar to the propagation of the error statistics as a sequence of MRFs we just discussed. Specifically, suppose that we have a multiscale model for the prediction error  $\chi(t|t-1)$  at time  $t$ . The model is completely specified by its model parameters  $\mathbf{A}(s;t|t-1)$ ,  $\mathbf{B}(s;t|t-1)$ , and  $\mathbf{P}(0;t|t-1)$ , where we use  $t|t-1$  to indicate that these are quantities associated with the prediction error model. Given the innovation of (3.7), the multiscale estimation algorithm yields not only the estimates of the prediction errors  $\hat{\chi}(t|t-1)$ , but also a multiscale model for the updated estimation error  $\chi(t|t)$ . Denote the model parameters for the updated estimation error by  $\mathbf{A}(s;t|t)$ ,  $\mathbf{B}(s;t|t)$ , and  $\mathbf{P}(0;t|t)$ , where we use  $t|t$  to indicate that these are quantities associated with the updated error model at  $t$ . The

update step of the multiscale recursive estimator is exactly a static estimation problem. That is, if we start with a multiscale model for  $\chi(t|t-1)$ , we directly obtain an analogous model for  $\chi(t|t)$  without explicitly calculating  $\mathbf{P}_\chi(t|t)$ . If the multiscale model for  $\chi(t|t-1)$  is exact, then  $\hat{\chi}(t|t-1)$  is computed exactly, i.e., optimally, and the multiscale model for  $\chi(t|t)$  is exact. For compact multiscale models with constant state dimension independent of the problem size, the update step is of  $\mathcal{O}(N)$  computational complexity. The update step of propagating a multiscale model for  $\chi(t|t-1)$  to a model for  $\chi(t|t)$  is illustrated in the first two boxes in Figure 3-1.

In contrast, the prediction step is the one that requires something new to avoid computational overload. We again assume that computing the predicted estimates of (3.4) for a sparse  $\mathbf{A}_d$  is simple. The challenge lies in the propagation of the multiscale models for the errors through the prediction step. To complete one step of the recursion we need to construct a multiscale model for the predicted estimation errors (3.13) without explicit calculating  $\mathbf{P}_\chi(t+1|t)$  as in (3.15). The prediction step is illustrated in the right two boxes in Figure 3-1. (In steady state, the model for  $\chi(t+1|t)$  becomes the same as that for  $\chi(t|t-1)$ .)

### 3.3.1 Multiscale Prediction Step

We develop the multiscale prediction step for the class of internal models as they provide an easy way to write the temporal dynamics of the coarse-scale states given dynamics of the fine-scale process, where the process of interest is placed and where the dynamics (3.13) are specified<sup>5</sup>. Thus we assume that the state at any node  $s$  in

---

<sup>5</sup>The multiscale recursive algorithm does not in principle require internal models. However, for external realization models whose coarse-scale states cannot be expressed as linear functionals of the finest-scale process, it is not clear how the coarse-scale states should evolve in time. That is, with the assumption that the process of interest is placed at the finest scale of a multiscale tree and that the temporal dynamics (3.13) are given for the the finest-scale process only, it is not clear how the coarse-scale states on the prediction error model relate to states on the updated error model as in (3.23). A brief analysis was done in [30, Appendix B] for time-varying external models for which very simple temporal dynamics are assumed for states at individual nodes such that each node on the multiscale tree evolves independently of one another. Even if one starts with a simple multiscale model with low state dimension, the exact multiscale model becomes intractable over time with growing state dimension. How external models of estimation errors can be propagated through the prediction step is an open question.

a multiscale model is a linear function of the finest-scale process, where, in general, the linear functions required may change with time. Suppose, then, that we have an internal model for the predicted estimation error at  $t$ , in which the state at node  $s$  is given by

$$\mathbf{x}(s; t|t-1) = \mathbf{L}(s; t|t-1)\boldsymbol{\chi}(t|t-1), \quad (3.21)$$

with model parameters  $\mathbf{A}(s; t|t-1)$ ,  $\mathbf{B}(s; t|t-1)$ , and  $\mathbf{P}(0; t|t-1)$ , and where we use the notation  $\mathbf{L}(s; t|t-1)$  to denote that the linear function at node  $s$  is for the predicted estimation errors at  $t$  based on data through  $t-1$ . The multiscale smoothing algorithm yields a new multiscale model with states

$$\mathbf{x}(s; t|t) = \mathbf{L}(s; t|t)\boldsymbol{\chi}(t|t), \quad (3.22)$$

for the updated error, with model parameters  $\mathbf{A}(s; t|t)$ ,  $\mathbf{B}(s; t|t)$ , and  $\mathbf{P}(0; t|t)$  computed as part of the multiscale estimation process<sup>6</sup>. The update step preserves the structure of the multiscale model as well as the state definitions, i.e.,  $\mathbf{L}(s; t|t) = \mathbf{L}(s; t|t-1)$ . The state at node  $s$  on tree model for the one-step-ahead prediction error then relates to the updated estimation error via

$$\mathbf{x}(s; t+1|t) = \mathbf{L}(s; t+1|t)\mathbf{A}_d\boldsymbol{\chi}(t|t) + \mathbf{L}(s; t+1|t)\mathbf{w}_d(t). \quad (3.23)$$

There are two principal issues involved in propagating a model of the estimation errors through the prediction step:

- (i) the specification of the linear functionals  $\mathbf{L}(s; t+1|t)$  that should be used to form the multiscale states  $\mathbf{x}(s; t+1|t)$  for modeling  $\boldsymbol{\chi}(t+1|t)$ , and
- (ii) determining and propagating the parameters of the multiscale model through the temporal dynamics (3.13).

---

<sup>6</sup>It is an abuse of notation to write  $\boldsymbol{\chi}(\cdot|\cdot)$  instead of  $\mathbf{x}(\cdot|\cdot)$ . The estimation errors  $\boldsymbol{\chi}(\cdot|\cdot)$  as defined in (3.3) evolves in time according to (3.12) and (3.13). We are attempting to model  $\boldsymbol{\chi}(\cdot|\cdot)$  on multiscale tree models. Since the models and the propagation of the models are approximate, what is realized at the finest scale of the tree,  $\mathbf{x}(\cdot|\cdot)$ , is not the same as what is to be realized,  $\boldsymbol{\chi}(\cdot|\cdot)$ .

In general, one might expect the linear functionals to change with time if the statistics of the one-step prediction errors are time varying, e.g., if the dynamics or measurements vary in time or if we wish to capture the transient behavior of the estimation process. The linear functionals in (3.21), (3.22), and (3.23), therefore, have been written to indicate that they may be time-dependent. One case in which these linear functionals should not change with time is if we are seeking to determine the form of the steady-state estimator for a time-invariant estimation problem, since in this case the prediction error statistics will be time-invariant. In this case, the recursive procedure of alternating update and prediction steps is used as an iterative method for constructing the steady-state multiscale estimator, in much the same way as using the time-varying Riccati equation to derive the steady-state solution of the Riccati equation, i.e., iterating (3.10) and (3.11) to arrive at the steady-state estimator gain  $\mathbf{K}(\infty)$ .

In the time-variant case, we may still impose a fixed set of linear functionals for modeling the estimation errors if the statistics do not change too much, or if the linear functionals can adequately, though approximately, capture the error statistics, or both. This is similar to the MRF-based recursive estimator in Section 3.2.1, where a Markov structure is imposed. In this thesis we will assume that the linear functionals defining the multiscale states for the predicted and updated estimation error models for all  $t$  are time-invariant. Thus, we simply write  $\mathbf{L}(s)$  without the second argument. The question of what these linear functionals should be may depend on the specific problem. As we will discuss in Chapters 4 and 5, the results from modeling diffusion processes show that there is a natural choice for the  $\mathbf{L}(s)$  (namely sets of boundary points of subintervals in 1-D or subregions in 2-D of the prediction errors), which works well for distributed parameter systems described by PDEs, both for the steady-state estimation problem *and* for time-varying estimation. We will also have more to say about the general properties we demand from these linear functionals in Section 3.3.3.

Once the choice of the linear functionals  $\mathbf{L}(s)$  is made, we are left with the second and final key issue, namely determining and propagating the parameters of the multiscale model through the temporal dynamics (3.13). The difficulty here, however,

is that dynamics generally introduce spatial mixing. Unraveling this mixing requires more detailed calculations which we describe next.

### 3.3.2 Model Parameters of the Predicted Error Model

Assume that we have specified a time-invariant set of linear functionals  $\mathbf{L}(s)$  to be used in the multiscale models. Assume that we have the model parameters  $\mathbf{A}(s; t|t)$ ,  $\mathbf{B}(s; t|t)$ , and  $\mathbf{P}(0; t|t)$  of the multiscale model for  $\boldsymbol{\chi}(t|t)$ . The covariances of the individual state vectors  $\boldsymbol{x}(s; t|t)$ ,  $\mathbf{P}(s; t|t)$ , and the cross-covariances between parent and child nodes,  $\mathbf{P}(s, s\bar{\gamma}; t|t)$ , are available as a result of the multiscale estimation algorithm from the previous update step, or are readily computed. What remains is to calculate the corresponding quantities  $\mathbf{A}(s; t+1|t)$ ,  $\mathbf{B}(s; t+1|t)$ , and  $\mathbf{P}(0; t+1|t)$  for the model for  $\boldsymbol{\chi}(t+1|t)$ , whose states are

$$\boldsymbol{x}(s; t+1|t) = \mathbf{L}(s)\boldsymbol{\chi}(t+1|t). \quad (3.24)$$

From (2.18), (2.19), (2.21), and (2.22), we see that  $\mathbf{A}(s; t+1|t)$  and  $\mathbf{B}(s; t+1|t)$  are determined if we have both the individual state covariances  $\mathbf{P}(s; t+1|t)$  and the parent-child cross-covariances  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$ . To derive expressions for the individual elements of these covariances, we first substitute the temporal dynamics (3.13) into (3.24):

$$\boldsymbol{x}(s; t+1|t) = \mathbf{L}(s)\mathbf{A}_d\boldsymbol{\chi}(t|t) + \mathbf{L}(s)\boldsymbol{w}_d(t). \quad (3.25)$$

It is here that we can see the spatial mixing caused by the temporal dynamics. Unless the set of linear functionals specified by  $\mathbf{L}(s)$  and the temporal dynamics commute, the term  $\mathbf{L}(s)\mathbf{A}_d$  produces a mixing of the linear functionals used to form the states  $\boldsymbol{x}(s; t|t) = \mathbf{L}(s)\boldsymbol{\chi}(t|t)$ . To unravel this further, let us examine individual components of the states in (3.25). Specifically, letting  $\boldsymbol{l}_i^T(s)$  denote the  $i$ -th row of  $\mathbf{L}(s)$ , i.e., corresponding to the  $i$ -th linear functional for  $\boldsymbol{x}(s; t+1|t)$ , we have

$$x_i(s; t+1|t) = \boldsymbol{l}_i^T(s)\mathbf{A}_d\boldsymbol{\chi}(t|t) + \boldsymbol{l}_i^T(s)\boldsymbol{w}_d(t). \quad (3.26)$$

The term  $\mathbf{l}_i^T(s)\mathbf{A}_d$  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  $\mathbf{L}(s)$ . However, it is possible to write it as a *linear combination* of existing functionals if the set of linear functionals corresponding to the rows of  $\mathbf{L}(s)$  for all nodes is a complete or an overcomplete set. This is indeed the case for internal models where the process of interest is placed at the finest scale. Since the finest scale captures the entire fine-scale process  $\boldsymbol{\chi}(\cdot|\cdot)$ , the collection of linear functionals at this scale already forms a complete basis. The collection of all linear functionals at all nodes then forms an overcomplete basis. As a consequence we can always write

$$\mathbf{l}_i^T(s)\mathbf{A}_d = \sum_{(\sigma,j) \in S} h_{\sigma,j}^{s,i} \mathbf{l}_j^T(\sigma), \quad (3.27)$$

where  $(\sigma, j)$  indexes the  $j$ -th state at node  $\sigma$  and  $S$  is the set of indices of all state components at all nodes on the tree. We can now write  $x_i(s; t+1|t)$  not in terms of the entire finest-scale process  $\boldsymbol{\chi}(t|t)$  as in (3.26), but in terms of selected states of this model:

$$x_i(s, t+1|t) = \sum_{(\sigma,j) \in S} h_{\sigma,j}^{s,i} \mathbf{l}_j^T(\sigma) \boldsymbol{\chi}(t|t) + \mathbf{l}_i^T(s) \mathbf{w}_d(t) \quad (3.28)$$

$$= \sum_{(\sigma,j) \in S} h_{\sigma,j}^{s,i} x_j(\sigma, t|t) + \mathbf{l}_i^T(s) \mathbf{w}_d(t). \quad (3.29)$$

Written in vector form for ease of notation

$$x_i(s, t+1|t) = \mathbf{h}_i^T(s) \boldsymbol{\xi}(t|t) + \mathbf{l}_i^T(s) \mathbf{w}_d(t), \quad (3.30)$$

where  $\boldsymbol{\xi}(t|t)$  is a random vector containing all the states in the multiscale tree model for  $\boldsymbol{\chi}(t|t)$ .

The representation in (3.27) and here in (3.29) are not unique for overcomplete sets of  $\{\mathbf{l}_j^T(s) | (j, s) \in S\}$ . To understand what characterizes a good choice of representation, let us examine the quantities that need to be computed, namely  $\mathbf{P}(s; t+1|t)$  and  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$ . The elements of these matrices correspond to quantities of the

form

$$E [x_i(s; t + 1|t)x_j(u; t + 1|t)] \quad (3.31)$$

for  $u = s$  or  $s\bar{\gamma}$  and for all components  $i$  and  $j$ . Examining (3.29) and (3.31) we see that such a quantity can be calculated in terms of the cross-covariances of elements of the states  $\mathbf{x}(s; t|t)$  of the model for  $\chi(t|t)$ :

$$E [x_i(s, t + 1|t)x_j^T(u, t + 1|t)] = \mathbf{h}_i^T(s)\mathbf{P}_\xi(t|t)\mathbf{h}_j(u) + \mathbf{l}_i^T(s)\mathbf{Q}_d\mathbf{l}_j(u). \quad (3.32)$$

Which cross-covariance entries of  $\mathbf{P}_\xi(t|t)$ , must be calculated depends on  $\mathbf{h}_i^T(s)$ . (The second term on the right,  $\mathbf{l}_i^T(s)\mathbf{Q}_d\mathbf{l}_j(u)$  is easy to compute as  $\mathbf{Q}_d$  is assumed to be diagonal.) By assumption we already have the covariance matrices  $\mathbf{P}(s; t|t)$  for each of the states of the updated error model individually and can, in principle, calculate cross-covariances using (2.8). However, as we pointed out in Section 2.1, calculating all or even many of these cross-covariances is prohibitive, although it is a relatively simple computational task to calculate  $\mathbf{P}(s_1, s_2; t|t)$  for  $s_1$  and  $s_2$  that are sufficiently close to each other on the tree, i.e., so that the distances between  $s_1$  and  $s_1 \wedge s_2$  and between  $s_2$  and  $s_1 \wedge s_2$  are small.

Thus, given an overcomplete set of  $\{\mathbf{l}_j^T(s) \mid (j, s) \in S\}$  that offer multiple ways of writing (3.27), it is desirable, if possible, to choose a representation of (3.27) such that the coefficients  $h_{\sigma,j}^{s,i}$  are extremely sparse and in fact are nonzero only for nodes  $\sigma$  that are near to node  $s$ . For instance, since the finest-scale linear functionals form a complete basis, (3.27) can always sum over those linear functionals. This, however, is a poor choice as it may amount to calculating the full  $\mathbf{P}_\chi(t|t)$  which we want to avoid in the first place. Whether this goal of finding sparse  $\mathbf{h}_i^T(s)$  can be achieved for all nodes on the tree depends on two related factors:

- (i) the type of linear functionals available on the tree and how they are distributed among the nodes; i.e., the  $\mathbf{l}_i^T(s)$  in (3.27), and
- (ii) the nature of dynamic mixing; i.e.,  $\mathbf{A}_d$  in (3.27).

The two factors are related as temporal dynamics determine in part the statistics of

the estimation errors and thus affect the choice of linear functionals for modeling the errors. Given the overcomplete set of linear functionals and the temporal dynamics, it is comparatively simple to decide what terms should go into the summation on the right-hand side of (3.27) so that the total amount computation for (3.31) is minimized, even though the best solution may still be computationally too costly. On the other hand, it is a major challenge to *design* linear functionals that result in computationally efficient multiscale prediction while simultaneously fulfilling their numerous other important objectives, as we discuss next. Focusing on only one objective of the linear functionals here, namely how to make the multiscale prediction step computationally efficient, we will demonstrate in Section 3.4.2 that we can perform the prediction step in  $\mathcal{O}(N \log N)$  time for an internal model with end-point linear functionals as the one shown in Figure 2-5 and for dynamics corresponding to discretized 1-D PDEs. We will show in Section 3.5.2 that for a different choice of linear functionals, i.e., non-redundant models, we can perform the prediction step in  $\mathcal{O}(N)$  time.

### 3.3.3 Choice of Linear Functionals

From the point of view of efficient propagation of a model for the estimation errors through the prediction step, the best linear functionals are the left-eigenvectors of  $\mathbf{A}_d$  such that the right-hand side of (3.27) has only one term:  $\mathbf{l}_i^T(s)\mathbf{A}_d = \lambda_i(s)\mathbf{l}_i^T(s)$ , where  $\lambda_i(s) = h_{s,i}^{s,i}$  is the left eigenvalue<sup>7</sup>, such that

$$x_i(s, t + 1|t) = \lambda_i(s)x_i(s, t|t) + \mathbf{l}_i^T(s)\mathbf{w}_d(t). \quad (3.33)$$

Note that this does not mean the states evolve independently as they are coupled through the temporal process noise. However, the left eigenvectors of  $\mathbf{A}_d$  generally have support over the entire process  $\chi(\cdot|\cdot)$ . Such linear functionals generally do not satisfy the conditional decorrelation requirement discussed in Section 2.3.4. Thus,

---

<sup>7</sup>Actually, it is not necessary that each individual linear functional be a left eigenvector for this “best” case. It is sufficient if the linear functionals at  $s$ , comprising the set of rows of  $\mathbf{L}(s)$ , are in the left eigenspace of  $\mathbf{A}_d$  such that  $\mathbf{L}(s)\mathbf{A}_d = \mathbf{\Lambda}(s)\mathbf{L}(s)$  for some  $\mathbf{\Lambda}(s)$  not necessarily diagonal.

this “best” case for model propagation will not, in general, also achieve the required decorrelation property for states in tree models. We would expect to relax the desire to have only one non-zero term in (3.27) in order to find linear functionals that achieve all of our objectives.

For instance, it is desirable that the linear functionals are “almost” in the left eigenspace of  $\mathbf{A}_d$ . By “almost” we mean the number of terms in the summation in (3.27) is very small. For example, suppose that  $\mathbf{A}_d$  is tridiagonal with constant diagonals for a 1-D dynamic system. The *point linear functionals*, i.e.,  $\mathbf{l}_i^T(s)$  that have exactly one non-zero element, are almost in the left eigenspace as the summation in (3.27) will always have exactly three terms. How the linear functionals are distributed among the nodes will have an impact on the total computational cost, since the cost of computing cross-covariances between two nodes on the tree is directly proportional to the distance between the two nodes, as we shall see shortly in Sections 3.4.2 and 3.5.2.

If  $\mathbf{A}_d$  is tridiagonal with constant diagonals, there are other linear functionals we might add to this set of point linear functionals while maintaining the sparsity of (3.27). For example, we may add functionals that are averages over a subinterval, e.g., let  $\mathbf{l}_i^T(s)$  be a 32-element row vector with ones at elements  $\{9-16\}$  and zeros elsewhere. Then,  $\mathbf{l}_i^T(s)\mathbf{A}_d = \kappa_1\mathbf{l}_i^T(s) + \sum_{j=\{8,9,16,17\}} \kappa_j\mathbf{e}_j^T$ , where  $\kappa_j$  are constant coefficients, and  $\mathbf{e}_j^T$  are row vectors with a one at the  $j$ -th element and zeros elsewhere, i.e., the point linear functionals. Thus functionals of these types (point values and averages) are attractive for the purpose of prediction for a tridiagonal  $\mathbf{A}_d$  because of the relative sparsity of (3.27). Whether these linear functionals also successfully fulfill other objectives such as accurately capturing the correlation structure of the random process of interest is another matter.

In summary, we are placing a multitude of requirements on the linear functionals, some of which may conflict with one another:

- (i) The linear functionals placed at node  $s$  must conditionally decorrelate the subtrees connected to  $s$  such that the desired statistics of the process to be modeled are accurately captured.

- (ii) The linear functionals must be almost in the left eigenspace of the temporal dynamic matrix  $\mathbf{A}_d$ , and they must be appropriately distributed among the nodes on the tree, such that multiscale prediction step is computationally efficient.
- (iii) The number of linear functionals kept at each node must be as small as possible to keep the computational cost down. They should be prioritized if possible to allow further reduction in state dimension for approximate models.
- (iv) The linear functionals must be consistent so that the models are truly internal.

Moreover, in other applications beyond the scope of this thesis there may be other requirements for these linear functionals. In particular, in some applications [25, 41] we may also require particular linear functionals to be included in the state for one of two reasons:

- (v) The linear functionals must be constructed to accommodate non-point or non-local measurements.
- (vi) The linear functionals must be constructed to accommodate non-local quantities that are to be estimated.

Choosing linear functionals that satisfy all of the above conditions is exceeding difficult. For the purpose of this thesis, we are focusing mostly on the first four requirements, as the applications that motivate our work here, such as mapping ocean surface height, often do have only point measurements and do not require non-local quantities to be estimated. Some ideas regarding requirements (v) and (vi) are included in Chapter 7.

## 3.4 Examples of Multiscale Prediction in 1-D

### 3.4.1 End-Point Linear Functionals and Diagonal $\mathbf{A}_d$

Take the example of the MRF-based recursive estimators we considered in Section 3.2.1. Let us assume that the updated estimation error process  $\chi(t|t)$  is a 1-D

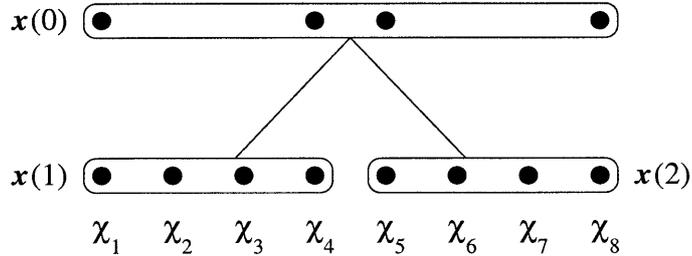


Figure 3-2: An example tree model using end-point linear functionals that can realize non-MRFs exactly at the finest scale. A broader class of random processes than 1-D MRFs can be realized by the end-point linear functionals.

MRF with a sparse  $\mathbf{P}_\chi^{-1}(t|t)$ , and assume that we have a multiscale model for  $\chi(t|t)$  with end-point linear functionals of the type shown in Figure 2-5, which exactly realizes  $\chi(t|t)$ . For the trivial case of  $\mathbf{A}_d = \mathbf{I}$ , which is used in the optical flow applications [15, 14, 13], the end-point linear functionals are in fact the left eigenvectors of  $\mathbf{A}_d$ . Thus, (3.33) applies, and to calculate  $\mathbf{P}(s; t+1|t)$  and  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$  on the predicted error model we need only  $\mathbf{P}(s; t|t)$  and  $\mathbf{P}(s, s\bar{\gamma}; t|t)$  on the updated error model.

We know from the discussion in Section 3.2.1 that the predicted estimation error  $\chi(t+1|t)$  is no longer a 1-D MRF, i.e.,  $\mathbf{P}_\chi^{-1}(t+1|t)$  is no longer sparse. Therefore, the end-point linear functionals cannot exactly capture the conditional decorrelation structure of  $\chi(t+1|t)$ , and the multiscale model for  $\chi(t+1|t)$  is not exact. In the MRF-based recursive estimation case, a MRF structure is imposed on  $\chi(t+1|t)$ . In the multiscale recursive estimation case, a fixed set of linear functionals is imposed. The two cases are subtly different. We have shown in Section 2.3.2 that 1-D MRFs are exactly modeled by the end-point linear functionals. However, the end-point linear functionals are in principle capable of realizing a broader class of random processes than just the 1-D MRFs. A MRF specifies a set of conditional decorrelation relationships. The end-point linear functionals enforces only a subset of these conditional decorrelation relationships. The rest of correlation structure is realized exactly by appropriately specifying the model parameters,  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$ . Take the simple example shown in Figure 3-2. The model can realize any correlation structure for  $\mathbf{x}(1)$ , including for example an  $\mathbf{x}(1)$  in which  $\chi_1$  and  $\chi_3$  do not conditionally decorrelate

$\chi_2$  and  $\chi_4$ . In other words, the end-point linear functionals do not impose a MRF structure on the approximate multiscale realization of  $\chi(t+1|t)$ . This is potentially advantageous as the end-point linear functionals are also convenient for multiscale prediction.

### 3.4.2 End-Point Linear Functionals and Tridiagonal $\mathbf{A}_d$

The statistical structure of the estimation errors is partially determined by the temporal dynamics as they enter in the prediction step in (3.11). The choice of linear functionals for modeling the errors will therefore depend on the dynamics. We will in Chapter 4 discuss in more detail what linear functionals are appropriate for 1-D diffusion and advection-diffusion processes. For now let us assume that the end-point linear functionals of Figure 2-5 are suitable for the  $\mathbf{A}_d$  we have. Let us further assume that the  $\mathbf{A}_d$  is tridiagonal<sup>8</sup>, e.g., the result of an explicit discretization scheme for a diffusion operator (c.f., Section 4.1). Of immediate interest here is how  $\mathbf{A}_d$  affects the spatial mixing during the prediction step.

For general internal models with the physical process placed at the finest scale, the collection of all linear functionals on the tree form an overcomplete basis. However, the end-point linear functionals, shown in Figure 2-5, in fact form only a complete basis because the coarser-scale states are merely copies of the variables at the finest scale. Let us write the basis functions as  $\mathbf{e}_j^T$ , a row vector with a one at the  $j$ -th element and zeros elsewhere. That is, the finest-scale node where the  $j$ -th element of  $\chi(\cdot|\cdot)$  is placed has  $\mathbf{e}_j^T$  as its functional. Of course, most of the  $\mathbf{e}_j^T$  also appear at coarser scales (refer to Figure 2-5). The elements of each of the cross-covariance matrix  $\mathbf{P}(s_1, s_2; \cdot|\cdot)$  are then a specific set of cross-covariances between fine-scale process values. Computing  $\mathbf{P}(s; t+1|t)$  and  $\mathbf{P}(s, s\bar{y}; t+1|t)$  to determine the model parameters of the predicted error model is then equivalent to filling certain entries of  $\mathbf{P}_\chi(t+1|t)$ .

---

<sup>8</sup>With circular boundary condition, i.e., a 1-D process on a ring, there may be two nonzero elements at the upper-right and the lower-left corners of  $\mathbf{A}_d$ . When we speak of tridiagonal  $\mathbf{A}_d$ , we also mean to include this case, even though the matrix is not strictly tridiagonal.

A tridiagonal  $\mathbf{A}_d$  introduces sparse and spatially local mixing. Let  $\mathbf{l}_i^T(s) = \mathbf{e}_j^T$  for some  $j$ . Then,  $\mathbf{l}_i^T(s)\mathbf{A}_d$  has exactly three nonzero elements and is written as the weighted sum of three linear functionals:  $\mathbf{l}_i^T(s)\mathbf{A}_d = \sum_{n=j-1,j,j+1} h_n \mathbf{e}_n^T$ . The  $\mathbf{e}_n^T \boldsymbol{\chi}(t|t)$ , for  $n = j-1, j, j+1$ , pick out three spatially neighboring elements of  $\boldsymbol{\chi}(t|t)$ . Therefore, each  $(k, l)$  entry of  $\mathbf{P}_\chi(t+1|t)$  depends on nine entries of  $\mathbf{P}_\chi(t|t)$  at  $(k+m, l+n)$ ,  $m, n \in \{-1, 0, 1\}$ . Since  $\mathbf{e}_j^T$  may appear at more than one node, the remaining question is which nodes on the updated error model are involved in computing the need entries of  $\mathbf{P}_\chi(t|t)$ . Note that the matrix  $\mathbf{P}_\chi(t|t)$  serves as a convenient way for keeping track of the necessary cross-covariances. Cross-covariances between nodes on the tree, equivalent to only a small percentage of the entries of  $\mathbf{P}_\chi(t|t)$ , are needed to define the multiscale model for the predicted error.

Suppose that we need the joint statistics between the state variables at nodes  $s$  and  $s\bar{\gamma}$  on the predicted error model, i.e.,  $\mathbf{P}(s; t+1|t)$ ,  $\mathbf{P}(s\bar{\gamma}; t+1|t)$ , and  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$ . Suppose that we have the joint statistics between state variables at nodes  $s$  and  $s\bar{\gamma}$  on the updated error model. The linear functionals at  $s$  and  $s\bar{\gamma}$  are marked as solid black circles in Figure 3-3. Spatial mixing due to the temporal dynamics means that we also need joint statistics with the variables spatially neighboring the solid circles, which are marked with  $\times$  in the figure. These neighboring elements are either already available at  $s$  and  $s\bar{\gamma}$  or they are found at some other nodes, the nearest of which are marked with shaded gray circles. Given the fixed state dimension for end-point linear functionals, the closer two nodes are, the less computation is required to calculate the cross-covariance between the two nodes, so it makes sense to locate the additional linear functionals (marked  $\times$ ) at nodes nearest  $s$  and  $s\bar{\gamma}$  so that the least amount of computation is required. Thus, to compute the joint statistics between  $s$  and  $s\bar{\gamma}$  on the predicted error model, we need to compute the joint statistics among nodes with either solid circles or shaded circles on the updated error model.

Taking all the  $\mathbf{P}(s; t+1|t)$  and  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$  that need to be computed, we map out the corresponding entries of  $\mathbf{P}_\chi(t|t)$  that must be filled in Figure 3-4. Since there are  $\mathcal{O}(N)$  nodes on the tree,  $\mathcal{O}(N)$  pairs of  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$  are to be computed. Each entry of  $\mathbf{P}_\chi(t+1|t)$  depends on at most nine entries of  $\mathbf{P}_\chi(t|t)$ , so we need

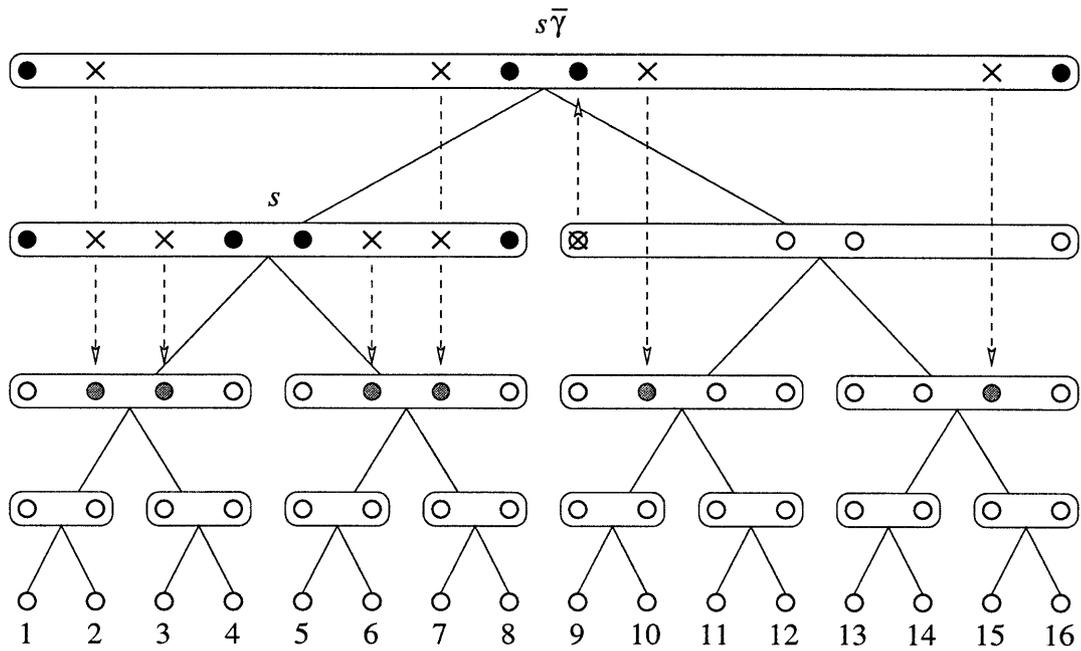


Figure 3-3: The cross-covariances needed in the internal model with end-point linear functionals for the updated estimation errors for multiscale prediction. The linear functionals are labeled with circles. The joint statistics between nodes  $s$  and  $s\bar{\gamma}$ , whose functionals are labeled with solid black circles, on the updated error model are assumed to be available. The  $\times$  marks indicate the additional functionals on the updated error model that are required for the prediction step due to temporal dynamics. The nearest locations on the tree where such linear functionals are found are marked with shaded gray circles. To compute the joint statistics between nodes  $s$  and  $s\bar{\gamma}$  on the predicted error model, joint statistics among nodes with either solid black circles or shaded gray circles on the updated error model must be computed.

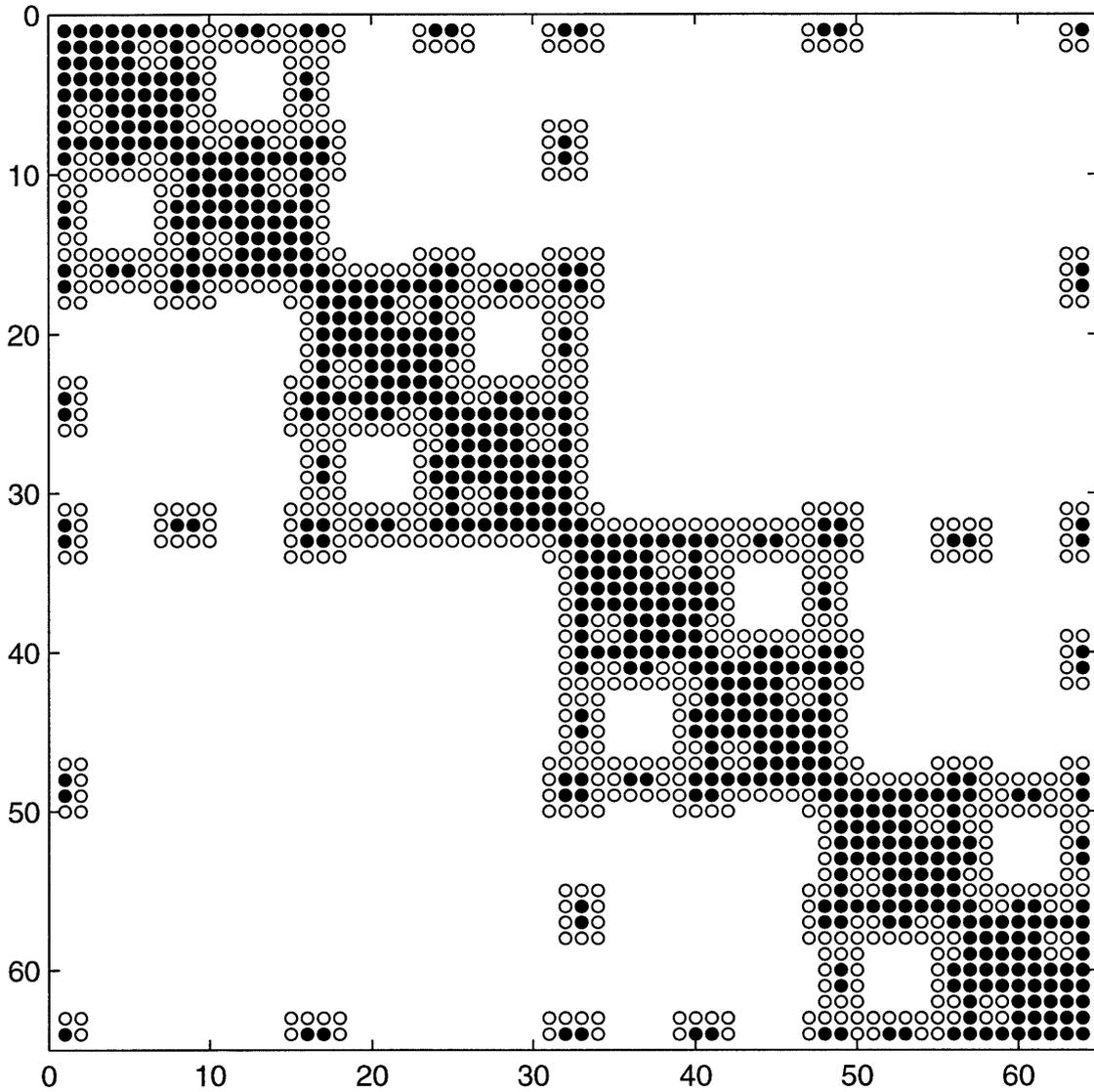


Figure 3-4: The entries of  $\mathbf{P}_\chi(t|t)$  that must be filled for multiscale prediction with end-point linear functionals are labeled with circles, solid or hollow. The solid circles mark those entries filled by computing covariances of states at individual nodes and cross-covariances between states at parent and child nodes, a by-product of the update step.

to calculate only  $\mathcal{O}(N)$  entries of the  $N \times N$  matrix  $\mathbf{P}_\chi(t|t)$ . This means that we need to compute cross-covariances of at most  $\mathcal{O}(N)$  pairs of nodes on the updated error model. Since the multiscale state dimension is fixed, the effort in computing each pair of cross-covariance is proportional to the tree distance between that pair of nodes, which is at most  $\mathcal{O}(\log N)$ . The total computations required to fill all the needed entries of  $\mathbf{P}_\chi(t|t)$  is then  $\mathcal{O}(N \log N)$ . Therefore, the recursive procedure here for multiscale models with end-point linear functionals and tridiagonal  $\mathbf{A}_d$

- requires  $\mathcal{O}(N)$  storage;
- is of  $\mathcal{O}(N)$  complexity for computing the updated estimates and for propagating the error model through the update step;
- is of  $\mathcal{O}(N)$  complexity for computing the predicted estimates and  $\mathcal{O}(N \log N)$  for propagating the error model through the prediction step.

## 3.5 Non-Redundant Models

### 3.5.1 Non-Redundant Models for MRFs

Multiscale prediction with a tridiagonal  $\mathbf{A}_d$  would be much easier and computationally efficient if the set of linear functionals could be designed such that the additional linear functionals needed for prediction due to dynamic mixing, i.e., those marked with  $\times$  in Figure 3-3 could be found at close-by nodes, e.g., within a fixed tree distance. This motivated the design of a new class of *non-redundant models* that we introduce here [50].

Multiscale realization methods and the realization of MRFs have been studied extensively [67, 70, 56, 41]. In midpoint deflection constructions of 1-D MRFs such as the multiscale models shown in Figures 2-4 and 2-5, the state at  $s$  includes exact copies of some state variables from the parent node  $s\bar{\gamma}$  plus some new state variables. The two models in Figures 2-4 and 2-5, in addition to other variations shown in [67, 70], are exact because the linear functionals so defined satisfy the conditional decorrelation

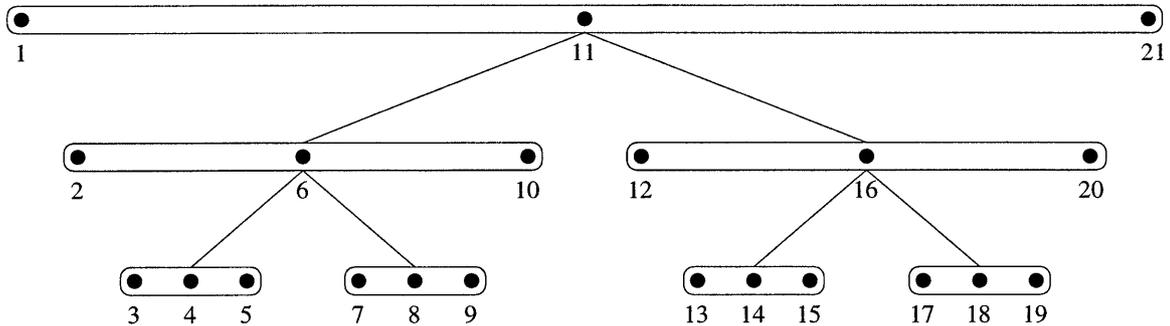


Figure 3-5: The linear functionals of a non-redundant model (with state dimension  $k = 3$ ) for a first-order 1-D Markov random field.

property or the Markovianity property on the tree (as explained in Section 2.3.4). In all multiscale models studied so far, however, the random process of interest is always mapped to the finest scale. Thus, for example, in models based on midpoint deflection coarse-scale states are simply copies of finest-scale state variables, making the multiscale model redundant. The linear functionals shown in Figure 3-5, on the other hand, are *non-redundant* as elements of the process of interest only appear once on the tree. The model still satisfies the conditional decorrelation requirement and is still exact for modeling 1-D MRFs.

There are several differences between the non-redundant models and the internal models of Figures 2-4 and 2-5. Immediately clear from the figure is that the process of interest is no longer found at the finest scale but rather is distributed among all nodes on the tree. Reassembling the process of interest from the variables on the tree requires some additional work but is straightforward. More importantly, since no element of the process of interest appears more than once on a non-redundant model, model consistency, which is important for internal models, becomes a non-issue.

The internal models in Figures 2-4 and 2-5 were designed as approximate representations of spatially continuous 1-D MRFs. These models can achieve any desired level of fidelity by extending the number of scales downward to finer resolutions. The states at the coarser levels remain unchanged as the tree grows more scales. However, for a non-redundant model, doubling the desired resolution would dramatically change the structure of the tree model, as the elements of the process of interest that

get mapped to nodes on the tree change dramatically. For example, Figure 3-5 may represent an approximation of a 1-D MRF at 21 grid points. The process at location 11 is placed at the root node; the process at the adjacent grid points at locations 10 and 12 are placed at the left and right nodes at scale 1, respectively. Doubling the resolution and modeling the 1-D MRF at 41 grid points results in the same definition for the root node. However, the grid points adjacent to location 11, which are now at locations 10.5 and 11.5, must be included in the states at scale 1, while the process at locations 10 and 12 are pushed down one scale to nodes at scale 2. Furthermore, doubling the resolution means that the number of elements of the discretized process is no longer divisible by 3. Figure 3-5 shows a regular dyadic tree on which each coarse-scale node has two children and each node has three state variables, such that the number of element of the process of interest is  $N = 3 \times (2^M - 1)$ , where  $M$  is the total number of scales. With 41 grid points an irregular tree would result. Of course, having a regular tree is mostly for convenience and makes no difference to the multiscale estimation algorithm. Non-redundant models can still approximate spatially continuous 1-D MRFs to any resolution, but they require that the resolution of the discretized process be specified ahead of time.

With the implicit assumption that the process of interest is placed at the finest scale, internal models are defined to be models whose coarse-scale states can be written as linear functionals of the finest-scale variables. Otherwise, the model is external. With this strict definition, non-redundant models appear to be external. However, if we relax the definition of internal models to mean models whose states can be written as linear functionals of the process of interest, then non-redundant models qualify. Non-redundant models do indeed resemble internal models in many ways and are well suited for multiscale prediction, as we shall see shortly in Section 3.5.2.

Non-redundant models for 2-D MRFs are analogous to the 1-D models. Instead of end-points of subintervals in 1-D, boundary points of subregions are kept as linear functionals. There are issues special to the 2-D case that are not present to 1-D. We defer the discussion of 2-D non-redundant models to Chapter 5.

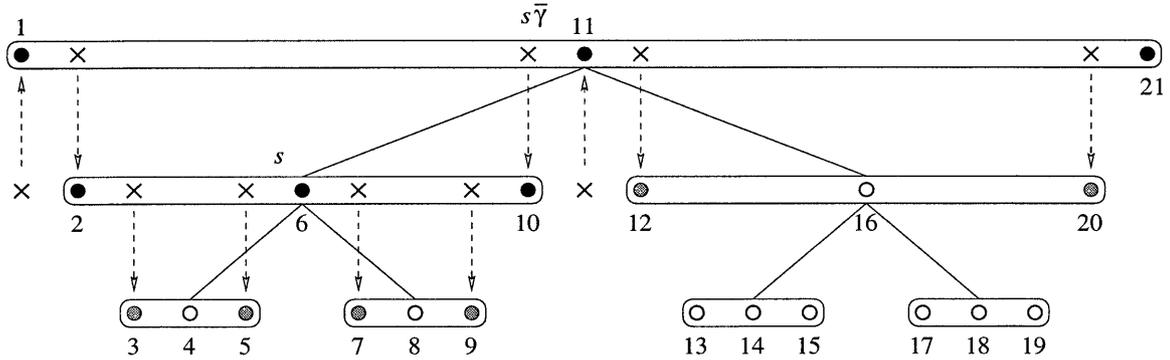


Figure 3-6: The cross-covariances needed in the non-redundant model for the updated estimation errors for multiscale prediction. The linear functionals are labeled with circles. The joint statistics between nodes  $s$  and  $s\bar{\gamma}$ , whose functionals are labeled with solid black circles, on the updated error model are assumed to be available. The  $\times$  marks indicate the additional functionals on the updated error model that are required for the prediction step due to a tridiagonal temporal dynamic matrix  $\mathbf{A}_d$ . Such linear functionals are found at either the parent or a child node of  $s$  and  $s\bar{\gamma}$ , where they are marked with shaded gray circles.

### 3.5.2 Multiscale Prediction in 1-D with Non-Redundant Models and Tridiagonal $\mathbf{A}_d$

Continuing the example in Section 3.4.2 with a tridiagonal  $\mathbf{A}_d$ , we see from Figure 3-6 that for non-redundant models in order to compute the joint statistics between nodes  $s$  and  $s\bar{\gamma}$  on the one-step prediction error model, the additional linear functionals on the updated error model needed for prediction are found at either the parent or a child node of  $s$  or  $s\bar{\gamma}$ . This means that a fixed amount of computation is required for computing the joint statistics between a pair nodes on the predicted error model. The total computational complexity of the prediction step is then  $\mathcal{O}(N)$ . Since the update step is also of  $\mathcal{O}(N)$  complexity, the resulting recursive algorithm is then  $\mathcal{O}(N)$ .

For the sole objective of computationally efficient multiscale prediction, the non-redundant models hold clear advantage over internal models with end-point linear functionals for the 1-D examples we have shown. However, as we illustrate in Section 7.2.3, internal models, or at least some level of redundancy, may be preferable in cases in which we wish to incorporate non-local measurements.

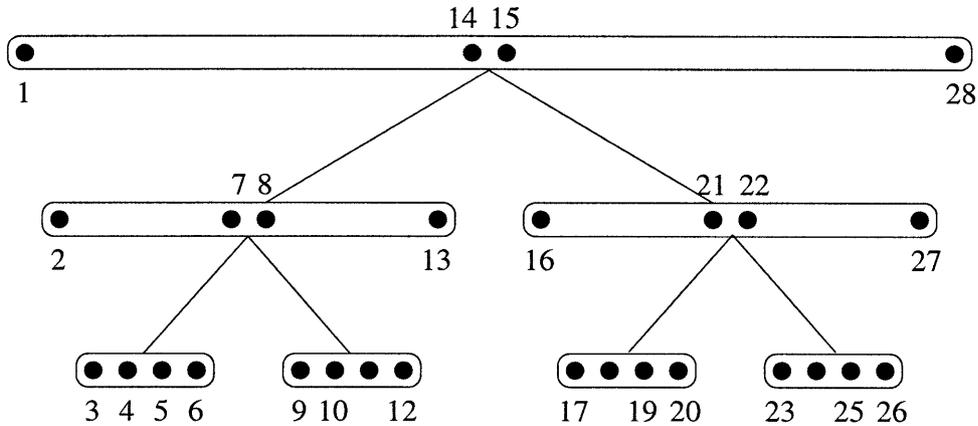


Figure 3-7: The linear functionals of a non-redundant model with state dimension  $k = 4$  for a first-order 1-D Markov random field.

### 3.5.3 Higher-Order Non-Redundant Models

There are several reasons for using higher-order, i.e., higher multiscale state dimension  $k$ , non-redundant models. We show three examples here.

We saw in Section 2.3.2 that even though an internal model with state dimension  $k = 3$ , e.g., with functionals shown in Figure 2-4, can exactly realize first-order 1-D MRFs, models with state dimension  $k = 4$ , e.g., with functionals shown in Figure 2-5, are desirable for reducing model redundancy. Although model redundancy is not an issue for non-redundant models, for estimator performance comparison purposes we shall see later in Section 4.2, it is still useful to have a higher-order non-redundant with  $k = 4$  for first-order 1-D MRFs, whose linear functionals are depicted in Figure 3-7.

Another reason for going to high state dimension is non-local dynamics. For example, Figure 3-8 shows a model for a penta-diagonal dynamics matrix  $\mathbf{A}_d$ .

The non-redundant models so far realize first-order 1-D MRFs exactly. Similar to the higher-order internal models with end-point linear functionals [67, 70], higher-order non-redundant models can be constructed for modeling second- or higher-order MRFs. Figure 3-9 shows such a non-redundant model for a second-order 1-D MRF. Because of the increased state-dimension and the selection of the functionals, this model is able to handle both tridiagonal and penta-diagonal dynamics matrices  $\mathbf{A}_d$ .

To summarize, we have shown several variations of 1-D non-redundant models:

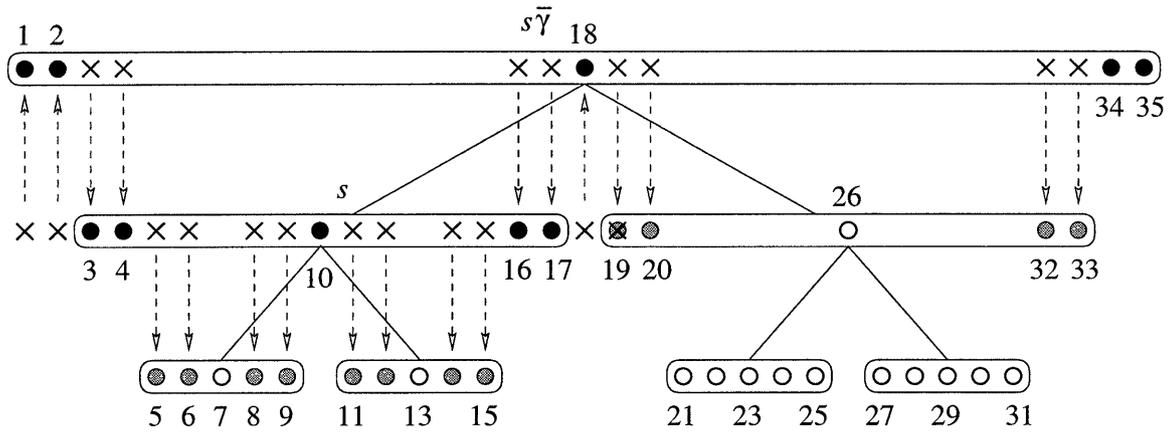


Figure 3-8: A non-redundant model with state dimension  $k = 5$  for a penta-diagonal dynamics matrix  $\mathbf{A}_d$ . The linear functionals are labeled with circles. The joint statistics between nodes  $s$  and  $s\bar{\gamma}$ , whose functionals are labeled with solid black circles, on the updated error model are assumed to be available. The  $\times$  marks indicate the additional functionals on the updated error model that are required for the prediction step due to temporal dynamics. The nearest locations on the tree where such linear functionals are found are marked with shaded gray circles.

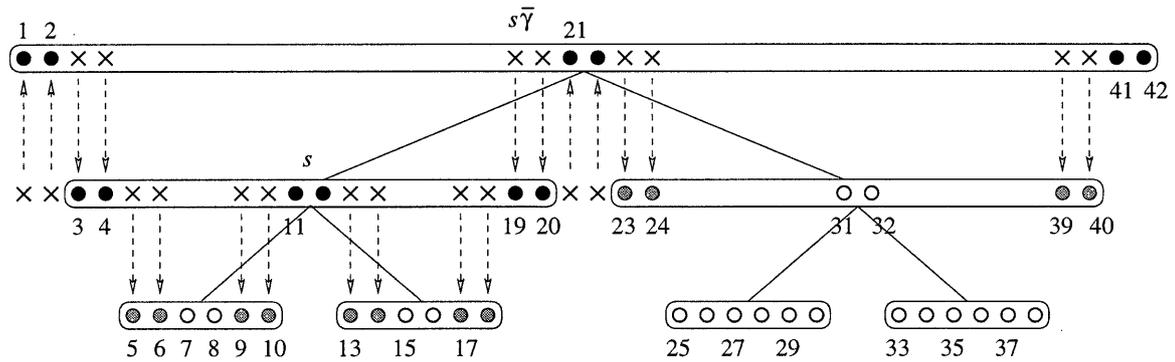


Figure 3-9: A non-redundant model with state dimension  $k = 6$  for a second-order 1-D Markov random field. The linear functionals are labeled with circles. The joint statistics between nodes  $s$  and  $s\bar{\gamma}$ , whose functionals are labeled with solid black circles, on the updated error model are assumed to be available. The  $\times$  marks indicate the additional functionals on the updated error model that are required for the prediction step due to a penta-diagonal dynamics matrix  $\mathbf{A}_d$ . The nearest locations on the tree where such linear functionals are found are marked with shaded gray circles.

- (i) Figure 3-5,  $k = 3$ , for modeling first-order MRFs.
- (ii) Figure 3-7,  $k = 4$ , for modeling first-order MRFs, design for better comparison with internal models with end-point linear functionals.
- (iii) Figure 3-8,  $k = 5$ , for modeling penta-diagonal dynamics matrix  $\mathbf{A}_d$ .
- (iv) Figure 3-9,  $k = 6$ , for modeling second-order MRFs as well as penta-diagonal  $\mathbf{A}_d$ .

## 3.6 Performance Measure

### 3.6.1 Convergence to Steady State

For the class of recursive estimators described in Section 3.2, the predicted and the updated estimates of the process of interest are computed according to (3.4) and (3.5), the latter of which is essentially a static estimation operation:

$$\hat{\chi}(t|t-1) = \mathbf{G}(t)\nu(t). \quad (3.34)$$

Any linear estimator, including our multiscale estimators, that takes  $\nu(t)$  as input and produces  $\hat{\chi}(t|t-1)$  as output, defines either explicitly or implicitly an equivalent estimator gain  $\mathbf{G}(t)$ . The optimal estimator  $\mathbf{K}(t)$  is just one possible  $\mathbf{G}(t)$ . Given dynamics  $\mathbf{A}_d$  the updated and the predicted estimation errors evolve according to (3.12) and (3.13); the updated and predicted estimation error covariances evolve according to (3.14) and (3.15).

The optimal Kalman filter specifies a time-dependent gain  $\mathbf{K}(t)$  that depends on the statistics of the estimation errors, as in (3.9). Starting from some initial condition  $\mathbf{P}_\chi(0|-1)$ , the Kalman filter converges to its steady-state gain  $\mathbf{K}(\infty)$  as the predicted error covariance converges to  $\tilde{\mathbf{P}}_p$  and the updated error covariance to  $\tilde{\mathbf{P}}_u$ . This optimal steady-state estimator  $\mathbf{K}(\infty)$  can be iteratively computed or it can be solved explicitly with, in addition to (3.9), the following steady-state Riccati

equation:

$$\tilde{\mathbf{P}}_p = \mathbf{A}_d \tilde{\mathbf{P}}_p \mathbf{A}_d^T + \mathbf{Q}_d - \mathbf{A}_d \tilde{\mathbf{P}}_p \mathbf{C}_d^T (\mathbf{C}_d \tilde{\mathbf{P}}_p \mathbf{C}_d^T + \mathbf{R}_d)^{-1} \mathbf{C}_d \tilde{\mathbf{P}}_p \mathbf{A}_d^T, \quad (3.35)$$

which follows from (3.10) and (3.11). If the steady-state estimator gain  $\mathbf{K}(\infty)$  is known beforehand, we may use this  $\mathbf{K}(\infty)$  as a fixed estimator without computing a new  $\mathbf{K}(t)$  at every time step, and the estimation error covariances will converge to the same  $\tilde{\mathbf{P}}_p$  and  $\tilde{\mathbf{P}}_u$ .

A suboptimal steady-state estimator with an equivalent gain  $\mathbf{G}$  can be run indefinitely starting from some initial condition. The predicted estimation error covariance, if it converges, converges to the solution to the following steady-state Lyapunov equation:

$$\tilde{\mathbf{P}}_{p,G} = \mathbf{A}_d \left( (\mathbf{I} - \mathbf{G}\mathbf{C}_d) \tilde{\mathbf{P}}_{p,G} (\mathbf{I} - \mathbf{G}\mathbf{C}_d)^T + \mathbf{G}\mathbf{R}_d\mathbf{G}^T \right) \mathbf{A}_d^T + \mathbf{Q}_d, \quad (3.36)$$

which follows from (3.14) and (3.15). The second subscript in  $\tilde{\mathbf{P}}_{p,G}$  indicates that  $\tilde{\mathbf{P}}_{p,G}$  is the steady-state predicted error covariance of the suboptimal estimator  $\mathbf{G}$ . Note that the Lyapunov equation of (3.36) computes the steady-state estimation error covariance for a given time-invariant estimator  $\mathbf{G}$ , whereas the Riccati equation of (3.35) computes the steady-state estimation error covariance of the optimal estimator *and* tells what the optimal estimator  $\mathbf{K}(\infty)$  should be (via (3.9)). Performance of the suboptimal steady-state estimator can be measured using either  $\tilde{\mathbf{P}}_{p,G}$  or  $\tilde{\mathbf{P}}_{u,G}$ , which are clearly not the same as  $\tilde{\mathbf{P}}_p$  and  $\tilde{\mathbf{P}}_u$ .

Suboptimal steady-state estimators  $\mathbf{G}$  that are of interest may include the following:

- (i) We may run the Kalman filter for a finite number of time steps to  $t = T$ , where the predicted error covariance is  $\mathbf{P}_\chi(T|T-1)$ , and declare the estimator  $\mathbf{K}(T)$  to be an approximate steady-state estimator.
- (ii) We may run the multiscale recursive algorithm for  $T$  iterations to reach an estimator with an equivalent  $\mathbf{G}(T)$ , which is then used as a steady-state es-

imator. Denote the realized predicted error covariance on the tree model by  $\mathbf{P}_X(T|T-1)$ .

- (iii) We may attempt to realize the optimal steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$  with a tree model. Since the linear functionals may not be exact, the realized covariance  $\mathbf{P}_{X,K}$  is not the same as  $\tilde{\mathbf{P}}_p$ . We denote the resulting estimator by  $\mathbf{G}_K$ .

For small problems, we may compare the performance of a suboptimal estimator with that of the optimal by solving for the respective steady-state error covariances via (3.36) and (3.35). For large problems, it is not computationally feasible to calculate these quantities. We may run simulations to experimentally determine the steady-state error variance. An alternative is to use the statistics assumed in the construction of the suboptimal estimator  $\mathbf{G}$  as a proxy for performance measure in some cases. For instance, to assess the quality of the approximate multiscale estimator  $\mathbf{G}(T)$  in (ii) above, we may use the diagonal of the realized  $\mathbf{P}_X(T|T-1)$ . Although computing the full  $\mathbf{P}_X(T|T-1)$  is computationally infeasible, the diagonal is readily computed in the tree model. However, since the multiscale recursive algorithm propagates the error statistics only approximately, the realized error statistics are what the estimator believes the error statistics to be, not the actual error statistics. Therefore, when using the realized variance as a proxy performance measure, we must treat it with caution.

It is not clear under what conditions the multiscale recursive estimation algorithm converges, and when it does, to what  $\mathbf{G}(\infty)$  it converges. There is no equivalent Riccati equation for solving for the steady-state estimation error statistics  $\tilde{\mathbf{P}}_{p,G(\infty)}$  and the steady-state estimator  $\mathbf{G}(\infty)$ . One possible candidate is the multiscale estimator  $\mathbf{G}_K$  realized from  $\tilde{\mathbf{P}}_p$  in (iii) above. However, we know that  $\mathbf{G}_K$  realizes at least the diagonal of  $\tilde{\mathbf{P}}_p$  exactly, but we have no compelling reason to believe that the multiscale recursive procedure should converge to a  $\mathbf{G}(\infty)$  that can exactly realize the diagonal of  $\tilde{\mathbf{P}}_p$ . Therefore,  $\mathbf{G}_K$  is most likely not the steady-state multiscale estimator  $\mathbf{G}(\infty)$ . The convergence properties of the multiscale recursive estimation algorithm remain an open theoretical problem.

### 3.6.2 Fractional Variance Reduction

To assess the quality of a suboptimal steady-state estimator, we use as a measure of performance the steady-state error variances, i.e., after running the approximate steady-state estimator until the estimation error statistics have converged. For systems small enough, we can calculate the equivalent gain  $\mathbf{G}$  for the suboptimal estimator and then solve the steady-state Lyapunov equation (3.36) explicitly for the steady-state error covariance matrix  $\mathbf{P}_{x,G}(\infty|\infty)$ .

Comparing directly the steady-state error variances of two estimators, however, may be misleading. For instance, if the estimation error variance of estimator A is 0.01 and the variance of estimator B is 0.02. It may appear that estimator B performs twice as poorly. Suppose that the process variance is 1. The process variance represents our prior knowledge of the process we are estimating and is equal to the estimation error variance when there is no measurement or when the measurement noise approaches infinity. With the measurement estimator A reduces the uncertainty from 1 down to 0.01, a 99% reduction, while estimator B reduces the uncertainty to 0.02, a 98% reduction. The performance of the two estimators should be considered to be comparable as they both reduce the estimation error variance from the prior by similar amounts. Therefore, we use the *fractional variance reduction* (FVR) as a measure of estimator performance:

$$\text{FVR} = \frac{\text{Var}(\text{s.s. process}) - \text{Var}(\text{s.s. updated estimation error})}{\text{Var}(\text{s.s. process})}. \quad (3.37)$$

For multi-dimensional systems, an FVR value can be computed for each element of the process. FVR encounters a similar problem as error variance when two estimators both perform poorly with respect to the prior. For example, if the error variances of estimators A and B are 0.99 and 0.98 respectively, then the FVR of the two estimator are 0.01 and 0.02 respectively. This seems to suggest that estimator B is twice as good as A, even though the two perform comparably poorly.

If we are to compare the performance of a suboptimal estimator with the optimal by taking the difference between some performance measure of the two and divide by

the performance of the optimal, then neither error variance nor FVR is adequate for the whole range of values. Error variance is a more meaningful measure where the two estimators both perform poorly, while FVR is more meaningful where both perform well. Since generally we are more interested in regions where there are measurements, and since an estimator generally performs better where there is a measurement than where there is none, we will adopt FVR as the performance measure. It is convenient that FVR is less reliable in areas we often care less about.



## Chapter 4

# Multiscale Dynamic Estimation of 1-D Diffusion

In this chapter we apply the multiscale recursive estimation algorithm developed in Chapter 3 to the estimation of 1-D dynamic systems. In particular, we choose to study diffusion processes in detail because they are fairly well understood, extensively studied, and have been employed in a wide variety of applications [5, 1]. The general approach to multiscale modeling and estimation of diffusion can be generalized to other dynamic processes. Indeed at the end of this chapter we will show estimation results for advection-diffusion processes, which find a number of useful applications especially in fluid dynamics such as monitoring pollution [74] and modeling tracer movements in the ocean [95, 94].

In the following sections, after setting up the basic equations, we first discuss in some detail the key issue of identifying suitable linear functionals  $\mathbf{L}(s)$  for modeling the estimation errors in the recursive estimation of diffusion processes. We demonstrate that both internal models with end-point linear functionals and non-redundant models can adequately represent the estimation errors. The resulting multiscale recursive estimators are not only computationally efficient, achieving  $\mathcal{O}(N \log N)$  complexity for end-point internal models and  $\mathcal{O}(N)$  for non-redundant models, as compared to  $\mathcal{O}(N^2)$  for MRF-based methods and  $\mathcal{O}(N^3)$  for the full Kalman filter, but also nearly optimal in performance. In the last section, we show a number of examples for 1-D

diffusion and advection-diffusion processes under a variety of system configurations.

## 4.1 Introduction to 1-D Diffusion

### 4.1.1 Specifications of the Process

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

$$\frac{\partial z(l, \tau)}{\partial \tau} = a \cdot \frac{\partial^2 z(l, \tau)}{\partial l^2} - b \cdot z(l, \tau) + c \cdot w(l, \tau), \quad (4.1)$$

where  $z(l, \tau)$  is the temperature at location  $l$  at time  $\tau$ ,  $w(l, \tau)$  is a white Gaussian noise with unit variance, and  $l \in [0, L]$ . The constants  $a$  and  $b$  are related to quantities such as the heat conduction coefficient, the dimensions of the rod or ring, etc. The first term on the right-hand side represents heat conduction within the ring, and the second term represents the heat loss to the surrounding coolant. The temperature of the coolant is set to zero, without loss of generality. The number of free parameters in (4.1) can be reduced by normalizing the spatial dimension to unit length and the diffusion parameter to 1<sup>1</sup>:

$$\frac{\partial z(l, \tau)}{\partial \tau} = \frac{\partial^2 z(l, \tau)}{\partial l^2} - \beta \cdot z(l, \tau) + \gamma \cdot w(l, \tau), \quad (4.5)$$

where  $\beta$  and  $\gamma$  are constants<sup>2</sup>.

---

<sup>1</sup>Starting with (4.1), we first normalize the spatial dimension to unit length by letting  $\bar{l} = l/L$ :

$$\frac{\partial z(\bar{l}, \tau)}{\partial \tau} = \left(\frac{a}{L^2}\right) \frac{\partial^2 z(\bar{l}, \tau)}{\partial \bar{l}^2} - b \cdot z(\bar{l}, \tau) + c \cdot w(\bar{l}, \tau). \quad (4.2)$$

Letting  $\bar{\tau} = (a/L^2) \tau$  normalizes the diffusion parameter to 1:

$$\left(\frac{a}{L^2}\right) \frac{\partial z(\bar{l}, \bar{\tau})}{\partial \bar{\tau}} = \left(\frac{a}{L^2}\right) \frac{\partial^2 z(\bar{l}, \bar{\tau})}{\partial \bar{l}^2} - b \cdot z(\bar{l}, \bar{\tau}) + c \cdot w(\bar{l}, \bar{\tau}), \quad (4.3)$$

and

$$\frac{\partial z(\bar{l}, \bar{\tau})}{\partial \bar{\tau}} = \frac{\partial^2 z(\bar{l}, \bar{\tau})}{\partial \bar{l}^2} - \left(\frac{bL^2}{a}\right) z(\bar{l}, \bar{\tau}) + \left(\frac{cL^2}{a}\right) w(\bar{l}, \bar{\tau}). \quad (4.4)$$

Re-labeling the coefficients and recycling the use of  $l$  and  $\tau$  give us (4.5).

<sup>2</sup>We will allow  $\beta$  to be a function of space for some examples shown in Section 4.4.

A number of finite difference schemes can be applied to discretize (4.5) to arrive at a system of difference equations in the form of (3.1),

$$\mathbf{z}(t+1) = \mathbf{A}_d \mathbf{z}(t) + \mathbf{w}_d(t), \quad (4.6)$$

where  $\mathbf{z}(t)$  is the vector containing the temperatures at all spatial grid points at time step  $t$ , and  $\mathbf{w}_d(t)$  models the process noise with covariance  $\mathbf{Q}_d$ . For our purposes such a discretized model plays two related but distinct roles:

- (i) to define the prediction of the estimates, as defined in (3.4), and
- (ii) to provide the dynamic matrix needed in (3.27)–(3.29) to predict the estimation error statistics by propagating the multiscale error model.

As we have seen in Sections 3.4 and 3.5, for the latter of these it is desirable to choose  $\mathbf{A}_d$  to be banded with a relatively small bandwidth such that the mixing of linear functionals is relatively local during each prediction step. Such an  $\mathbf{A}_d$  arises if we use an explicit finite-difference temporal discretization scheme. In particular, in the results presented here we use a simple forward Euler scheme<sup>3</sup>, in which case  $\mathbf{A}_d$  is

---

<sup>3</sup>Discretization schemes for PDEs can be found a number of books including [82]. We first discretize (4.5) in space to get

$$\frac{d}{d\tau} \mathbf{z} = \mathbf{D} \mathbf{z} + \mathbf{w}, \quad (4.7)$$

where elements of the vector  $\mathbf{z}$  are continuous-time variables at the spatial grid points, and  $\mathbf{D}$  is a banded matrix approximating the differential operators on the right-hand side of (4.5). With forward Euler temporal discretization,

$$\frac{1}{\Delta\tau} (\mathbf{z}(t+1) - \mathbf{z}(t)) = \mathbf{D} \mathbf{z}(t) + \mathbf{w}(t), \quad (4.8)$$

and

$$\mathbf{z}(t+1) = (\mathbf{I} + \Delta\tau \mathbf{D}) \mathbf{z}(t) + \Delta\tau \mathbf{w}(t), \quad (4.9)$$

$\mathbf{A}_d$  is then tridiagonal. With backward Euler temporal discretization,

$$\frac{1}{\Delta\tau} (\mathbf{z}(t) - \mathbf{z}(t-1)) = \mathbf{D} \mathbf{z}(t) + \mathbf{w}(t), \quad (4.10)$$

and

$$\mathbf{z}(t) = (\mathbf{I} - \Delta\tau \mathbf{D})^{-1} \mathbf{z}(t-1) + (\mathbf{I} - \Delta\tau \mathbf{D})^{-1} \Delta\tau \mathbf{w}(t), \quad (4.11)$$

$\mathbf{A}_d$  is generally full.

tridiagonal<sup>4</sup> and  $\mathbf{Q}_d = \sigma_w^2 \mathbf{I}$ . Of course when we use such a scheme for the former purpose as well, i.e., in the actual prediction step (3.4), care must be taken to ensure that the spatial discretization step size  $\Delta l$  and the temporal step size  $\Delta\tau$  are small enough for numerical accuracy and convergence [82, 52]. Obviously a better choice for this former purpose would be an implicit discretization scheme, but it would result in a dense matrix  $\mathbf{A}_d$ .

We can actually consider using a different choice of  $\mathbf{A}_d$  for each of these two purposes, namely an implicit scheme for the prediction step of the estimates and an explicit scheme for propagating the multiscale error models through the dynamic prediction step. We may even consider using two different time step sizes for each of the discretization schemes, e.g., a larger  $\Delta\tau$  for the implicit scheme. A dense but more accurate  $\mathbf{A}_d$  from the implicit scheme may be used to compute the predicted estimate, while a banded  $\mathbf{A}_d$  from the explicit scheme is used multiple times to propagate the error model over the same amount of real time. 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, such that new measurements are not incorporated optimally in the update step. However, the multiscale error model *already* introduces approximations into modeling the update error, and we will see in Section 4.4 that the net effect of all of these approximations is a surprisingly small loss in performance. For the purposes of this thesis, we will simply adopt the explicit forward Euler temporal discretization scheme and assume a tridiagonal  $\mathbf{A}_d$ .

The complete specification of the model (4.6) of course also implies the specification of a specific set of boundary conditions. There are four typical boundary conditions [54]:

- (i) circular boundary condition:  $z(0, \tau) = z(1, \tau)$ , corresponding to a thin cooling ring;
- (ii) Dirichlet boundary condition:  $z(0, \tau) = z_0$  (or  $z(1, \tau) = z_1$ ), corresponding to a

---

<sup>4</sup>As we have said in Chapter 3, for a process on a ring there may be nonzero elements at the upper-right and the lower-left corners of  $\mathbf{A}_d$ . When we speak of tridiagonal or penta-diagonal  $\mathbf{A}_d$ , we also mean to include this case, even though the matrix is not strictly tridiagonal or penta-diagonal.

- thin rod where one end is pinned to a heat source at temperature  $z_0$  (or  $z_1$ );
- (iii) Von Neumann boundary condition:  $\frac{\partial}{\partial t}z(0, \tau) = r_0$  (or  $\frac{\partial}{\partial t}z(1, \tau) = r_1$ ), corresponding to a thin rod where one end is connected to a conductive material with the rate of heat flow equal to  $r_0$  (or  $r_1$ );
- (iv) Type 3 homogeneous boundary condition:  $\frac{\partial}{\partial t}z(0, \tau) = -\beta z(0, \tau)$  (or  $\frac{\partial}{\partial t}z(1, \tau) = -\beta z(1, \tau)$ ), corresponding to a thin rod where one end is freely immersed in the coolant.

For the purpose of describing our methodology in the following sections, we for convenience assume circular boundary conditions. However, as we illustrate via examples, our approach easily applies to the other cases as well. Moreover, the specification of other boundary conditions leaves the analysis of the linear functionals and the development of the multiscale estimators unchanged and effects only the specific numerical values that result. The examples shown in Section 4.4 assume a variety of boundary conditions depending on the physical setup of the system.

Since we are interested in constructing the multiscale steady-state estimator, the optimal steady-state process covariance  $\mathbf{P}_z$  is of interest to us. The process covariance represents our prior knowledge of the steady-state errors without measurements. It is also the limit of the steady-state error covariances when the measurement noise variance approaches infinity. It is solved by the following steady-state Lyapunov equation:

$$\mathbf{P}_z = \mathbf{A}_d \mathbf{P}_z \mathbf{A}_d + \mathbf{Q}_d. \quad (4.12)$$

The process noise variance  $\sigma_w^2$  merely scales the magnitude of the steady-state process covariance. In the case of a process with circular boundary condition and spatially constant system parameter  $\beta$ , the steady-state process variance is spatially uniform. Let the diagonal elements of  $\mathbf{P}_z$  in this case be  $\sigma_p^2$ . We may choose a  $\sigma_w$  such that the steady-state process variance  $\sigma_p^2$  is normalized to 1. For spatially non-uniform steady-state variance, we may for convenience choose a  $\sigma_w$  that normalizes the maximum of the steady-state variance to 1.

## 4.1.2 Specifications of the Measurements

We assume point measurements that may be irregular in space, but stationary in time<sup>5</sup>,

$$\mathbf{y}_d(t) = \mathbf{C}_d \mathbf{z}(t) + \mathbf{v}_d(t), \quad (4.13)$$

where  $\mathbf{C}_d$  is a selection matrix, and the measurement noise  $\mathbf{v}_d$  is white with covariance  $\mathbf{R}_d = \sigma_v^2 \mathbf{I}$  and uncorrelated with  $\mathbf{z}(t)$  or  $\mathbf{w}_d(t)$ .

The optimal steady-state predicted estimation error covariance  $\tilde{\mathbf{P}}_p$  satisfies the Riccati equation (3.35). Different choices of  $\sigma_w^2$  and  $\sigma_v^2$  with a constant ratio only scale the steady-state error covariances. Therefore, given a particular measurement configuration  $\mathbf{C}_d$ , we are left with only two free parameters, namely  $\beta$  and  $\sigma_w^2/\sigma_v^2$ . We use the more familiar notion of signal-to-noise ratio  $\text{SNR} = 10 \log(\sigma_p^2/\sigma_v^2)$  instead of  $\sigma_w^2/\sigma_v^2$ . For spatially uniform and normalized steady-state process variance, i.e.,  $\sigma_w^2$  is chosen such that  $\sigma_p^2 = 1$ , the SNR is simply  $10 \log(1/\sigma_v^2)$ . For normalized but spatially non-uniform process variance, i.e.,  $\sigma_w^2$  is chosen such that the maximum of  $\sigma_p^2$  is 1, a point-wise SNR can be defined at every spatial grid point. However, for convenience, we simply use the SNR value at the grid point where the maximum  $\sigma_p^2$  is.

## 4.2 Linear Functionals for 1-D Diffusion

Our multiscale recursive estimation methodology discussed in Chapter 3 for solving the Riccati equation assumes that we have the “right” linear functionals for modeling the steady-state predicted estimation error. For small-size systems, it is computationally possible to explicitly solve the Riccati equation (3.35) for the exact steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$  and the Lyapunov equation (4.12) for the exact steady-state process covariance  $\mathbf{P}_z$ . From these we then apply the canonical correlations realization (CCR) algorithm [56] described in Section 2.3.3 to construct

---

<sup>5</sup>The temporal stationarity assumption is relaxed for the last example in Section 4.4, i.e.,  $\mathbf{C}_d$  in (4.13) is a function of time.

realizations explicitly. Using the intuition gained from these small examples, we can then select the appropriate linear functionals for larger systems, where neither the Riccati equation nor CCR is computationally feasible.

### 4.2.1 Functionals for the Steady-State Process

It was shown in [21] that the continuous-time, continuous-space 1-D heat diffusion process of (4.5) on a ring in steady state is a 1-D MRF. It is reasonable to expect our discretized model (4.6) to be close to 1-D MRF except for errors introduced by the discretization. It is, therefore, also reasonable to believe that the end-point internal models of Figure 2-5 and the non-redundant models of Figure 3-5, which exactly realize 1-D MRFs as we discussed in Sections 2.3.2 and 3.5.1, can adequately model the steady-state diffusion process. To test this we take as an example a 64-dimensional discretized diffusion process with circular boundary condition and spatially uniform diffusion and heat loss parameters, and explicitly compute its steady-state process covariance  $\mathbf{P}_z$ . We are interested in modeling the steady-state process mainly as a limiting case of the steady-state errors. We will discuss in the next subsection the modeling of the steady-state prediction errors.

We know from Section 2.3.2 that the information about the conditional correlation relations among elements of a discrete random process is contained in the inverse of its covariance matrix, and that processes with tridiagonal and penta-diagonal inverse covariance matrices are first- and second-order 1-D MRFs. The inverse of the process covariance  $\mathbf{P}_z$  of our discretized diffusion process is found to be penta-diagonal in general (with non-zero corners to account for circular boundary conditions). That is, the discretized steady-state process can essentially be modeled exactly as second-order MRFs, although the second off-diagonals are sometimes orders of magnitude smaller than the first off-diagonals. The second off-diagonals also become increasingly insignificant as the temporal discretization step  $\Delta\tau$  decreases. For a different value of  $\Delta\tau$ , we have a different discretized dynamics matrix  $\mathbf{A}_d$ , for which a steady-state process covariance  $\mathbf{P}_z$  can be computed from (4.12). As  $\Delta\tau$  decreases,  $\mathbf{P}_z$  is increasingly better approximated as a first-order 1-D MRF. In the limit as  $\Delta\tau \rightarrow 0$ , there

exists a continuous-time Lyapunov equation:

$$0 = \mathbf{D}\mathbf{P}_z + \mathbf{P}_z\mathbf{D}^T + \mathbf{Q}_w, \quad (4.14)$$

where  $\mathbf{D}$  is the continuous-time dynamics matrix from (4.7) and  $\mathbf{Q}_w$  is the covariance of  $\mathbf{w}$  in (4.7).

First- and second-order MRF approximations of the steady-state covariance are obtained by zeroing out the elements of  $\mathbf{P}_z^{-1}$  that are not in the tridiagonal or the penta-diagonal bands, and inverting the truncated inverse covariance. Denote the approximately realized covariance by  $\bar{\mathbf{P}}_z$ . To measure how accurate a first- or second-order MRF approximation is, i.e., how close the approximated covariance  $\bar{\mathbf{P}}_z$  is to the desired covariance  $\mathbf{P}_z$ , we use the Bhattacharyya distance [59, 55], defined as

$$d(\mathbf{P}_z, \bar{\mathbf{P}}_z) = \frac{1}{2} \ln \left( \frac{|\frac{1}{2}\mathbf{P}_z + \frac{1}{2}\bar{\mathbf{P}}_z|}{|\mathbf{P}_z|^{1/2} |\bar{\mathbf{P}}_z|^{1/2}} \right). \quad (4.15)$$

The Bhattacharyya distance is closely related to the probability of error in binary hypothesis tests for distinguishing between two realizations of a random vector. If the distance is large, then a reasonable decision rule for the hypothesis test will have a low probability of error. Table 4.1 shows that the discretized diffusion process is well approximated as a second-order MRF, while  $\Delta\tau$  has to be sufficiently small for it to be adequately approximated as first-order. The accuracy of the first-order MRF approximation improves as the temporal discretization step size decreases. However, the slowest time constant of the diffusion process used in Table 4.1 is 0.1 sec<sup>6</sup>. Even at a reasonably small  $\Delta\tau = 2 \times 10^{-5}$  sec the first-order MRF approximation is still a fairly poor one, while the second-order MRF approximation at this  $\Delta\tau$  is nearly exact, as seen in Figure 4-1. A reasonable question we might ask is whether the internal model with end-point linear functionals in Figure 2-5 or the non-redundant model in Figure 3-5, both of which are designed to model first-order MRFs exactly,

---

<sup>6</sup>The time constants of a discrete-time dynamic system is found via the eigenvalues of  $\mathbf{A}_d$ . The largest eigenvalue  $\lambda_{\max}$  corresponds to the slowest time constant  $-\Delta\tau/\ln(\lambda_{\max})$ .

$\Delta\tau$	Bhattacharrya distance between $\mathbf{P}_z$ from (4.12) and $\bar{\mathbf{P}}_z$ from	
	first-order MRF approximation	second-order MRF approximation
$2 \times 10^{-5}$	2.48	$1.11 \times 10^{-16}$
$1 \times 10^{-5}$	1.66	$1.11 \times 10^{-16}$
$2 \times 10^{-6}$	0.44	0
$\rightarrow 0$	0	0

Table 4.1: The Bhattacharrya distances between the steady-state process covariance  $\mathbf{P}_z$  of a 64-element diffusion process on a ring ( $\beta = 10$ ) and the realized covariances  $\bar{\mathbf{P}}_z$  from first- and second-order MRF approximations for different values of  $\Delta\tau$ . The numbers reported in the column under second-order MRF approximation are on the order of numerical errors. The Bhattacharrya distance may be so small that the result is below machine precision, for which case we list the answer as 0, even though there might be very slight differences between  $\mathbf{P}_z$  and  $\bar{\mathbf{P}}_z$ .

can in fact adequately realize the steady-state process covariance  $\mathbf{P}_z$ .

We first turn to canonical correlations realization method for identifying the linear functionals to be used in constructing a multiscale internal model for which the steady-state diffusion process is mapped to the finest scale. Recall from the discussion in Section 2.3.1 that at each node  $s$  on the tree we would like to keep linear functionals that decorrelate the sets of values of the fine-scale process on the disjoint parts of the tree separated by the node  $s$ . The CCR algorithm produces a set of linear functionals ordered by their relative statistical significance in decorrelating the subsets of fine-scale values. For instance, in our 64-element diffusion example the linear functionals at the root node must decorrelate elements  $\{1 - 32\}$  from elements  $\{33 - 64\}$ ; the functionals at the left node at scale 1, i.e., the left child node of the root node whose descendents at the finest scale are elements  $\{1 - 32\}$ , must decorrelate the three subsets of the fine-scale process: elements  $\{1 - 16\}$ ,  $\{17 - 32\}$ , and  $\{33 - 64\}$ ; the functionals at the right node at scale 1 must decorrelate  $\{1 - 32\}$ ,  $\{33 - 48\}$ , and  $\{49 - 64\}$ ; and so on. Figure 4-2 shows the two most significant linear functionals chosen by CCR to decorrelate element  $\{1 - 32\}$  from  $\{33 - 64\}$  of the steady-state process in our 64-element example. Only the first two linear functionals shown in the

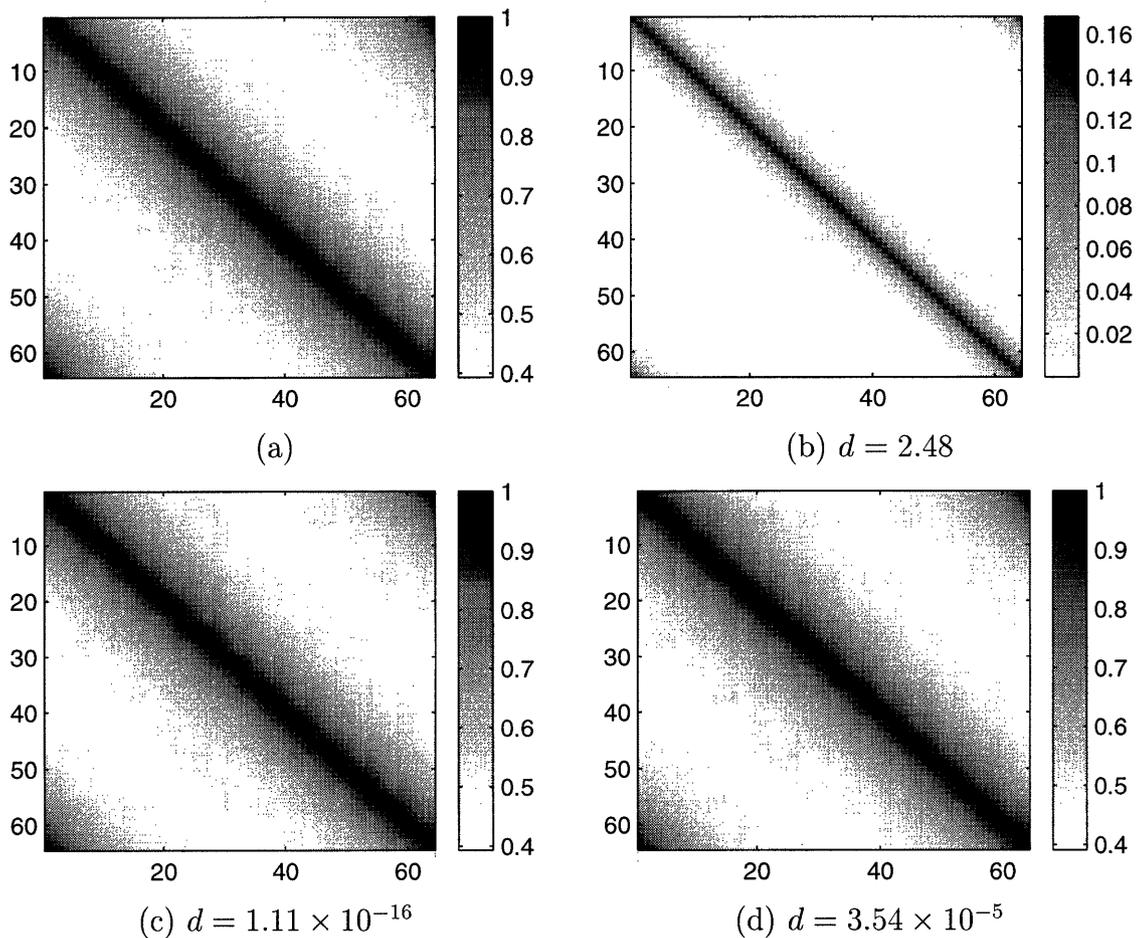


Figure 4-1: A comparison of (a) the exact steady-state process covariance of a 64-element on a ring discretized diffusion process with (b) a first-order MRF approximation of the covariance, (c) a second-order MRF approximation, and (d) the covariance realized by an internal model with end-point linear functionals. The difference between the approximations and the exact covariance is measured in terms of the Bhattacharyya distance, labeled  $d$  under each panel. ( $\beta = 10$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

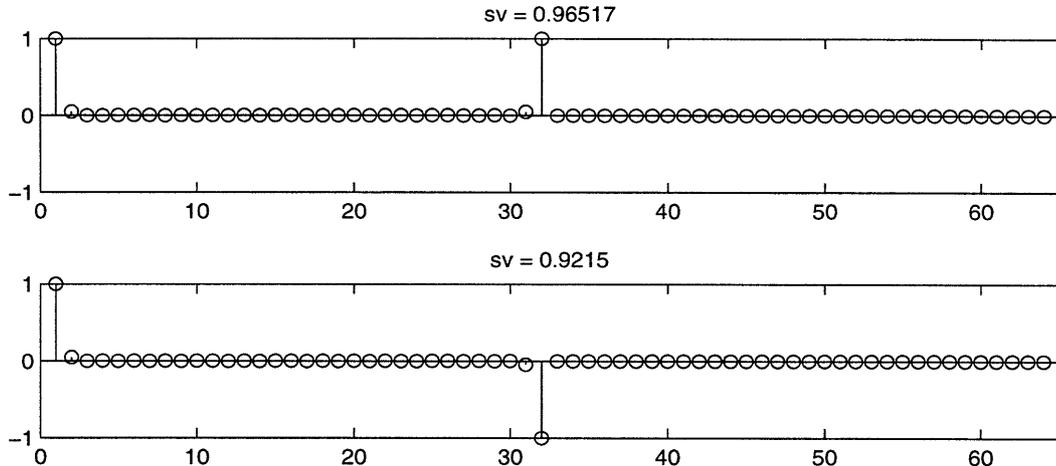


Figure 4-2: Two of the linear functionals that maximally decorrelate elements  $\{1-32\}$  from elements  $\{33-64\}$  of a 64-element diffusion process on a ring in steady state. The maximum magnitude of the functionals is normalized to 1. The region of support for the functionals is points  $\{1-32\}$ . The relative significance of the functionals is indicated by the associated singular values (sv). ( $\beta = 10$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

figure have significant singular values, implying that keeping only these two achieves near perfect decorrelation. Furthermore, the residual correlation decreases as the time step size  $\Delta\tau$  is decreased. This is consistent with the observation that the first-order MRF approximation is increasingly more accurate with decreasing time step size. Moreover, the two so-identified linear functionals have tiny non-zero values at points other than 1 and 32. Thus, a good approximation to the information provided by these two linear functionals is simply the values of the fine-scale process at end-points 1 and 32. In other words, the approximate functionals resulting from reducing the region of support for the two functionals in Figure 4-2 from elements  $\{1-32\}$  to just elements 1 and 32 are equivalent to two point linear functionals at 1 and 32 via a simple basis transform.

In Table 4.2 we compare the fidelity of the realized steady-state process covariance  $\bar{\mathbf{P}}_z$  from several approximate models with the desired covariance matrix  $\mathbf{P}_z$ . As we have already seen from the images of  $\mathbf{P}_z$  and  $\bar{\mathbf{P}}_z$  in Figure 4-1, the first-order 1-D MRF approximation to the steady-state process covariance is rather poor, while the second-order MRF approximation is nearly perfect. We compare the realized steady-state process covariance from four multiscale models:

Approximate model	Bhattacharrya distance between the exact $\mathbf{P}_z$ from (4.12) and the realized $\bar{\mathbf{P}}_z$
first-order MRF	2.48
second-order MRF	$1.11 \times 10^{-16}$
CCR internal model, $k = 4$	$1.18 \times 10^{-10}$
end-point internal model, $k = 4$	$3.54 \times 10^{-5}$
non-redundant model, $k = 4$	$5.82 \times 10^{-5}$
non-redundant model, $k = 3$	$1.34 \times 10^{-4}$

Table 4.2: The Bhattacharrya distances between the steady-state process covariance  $\mathbf{P}_z$  of a 64-element diffusion process on a ring and the realized covariances  $\bar{\mathbf{P}}_z$  from various approximate models. The state dimension for the multiscale models is denoted by  $k$ . ( $\beta = 10$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

- (i) The internal model with the linear functionals identified by CCR at all nodes. In Figure 4-2 we showed the linear functionals selected by CCR to decorrelate elements  $\{1 - 32\}$  from  $\{33 - 64\}$ . Similar results are obtained for decorrelating any subset of the fine-scale process from its complement, i.e., the two most significant linear functionals capture essentially all the correlation between the two subsets. As discussed in Section 2.3.3, at each node on the tree, we must decorrelate three random vectors. For example, at the left child node of the root node, we must decorrelate elements  $\{1 - 16\}$ ,  $\{17 - 32\}$ , and  $\{33 - 64\}$ . The linear functionals kept by CCR at this node consist of the two linear functionals from decorrelating  $\{1 - 16\}$  from  $\{17 - 64\}$  and the two from decorrelating  $\{17 - 32\}$  from  $\{1 - 16, 33 - 64\}$ . Thus the multiscale state dimension is  $k = 4$ .
- (ii) The internal model with end-point linear functionals of Figure 2-5. The linear functionals kept on this model are equivalent to the linear functionals generated by CCR but with their region of support restricted to the end-points. Not surprisingly the internal model with end-point linear functional realizes a covariance less accurate than the one by CCR method, although the degradation

in accuracy is insignificant, since the functionals generated by CCR have only tiny values away from the end-points, as shown in Figure 4-2. The state dimension here is also  $k = 4$ . The image of the realized covariance at the finest scale, shown in Figure 4-1(d), looks identical to the desired  $\mathbf{P}_z$ .

- (iii) The non-redundant model of Figure 3-5 with state dimension  $k = 3$ . Although both the internal model in Figure 2-5 and the non-redundant model in Figure 3-5 are designed to model first-order MRFs exactly, the former should yield a better approximation of second-order MRFs than the latter, since it keeps one more mid-point than the latter. This allows it to better capture some additional correlations that the latter model is not able to. It is then reasonable to expect that the non-redundant model will be less accurate than the internal model. A 64-element process placed on a non-redundant model results in an irregular tree, where some nodes have only one child and/or some nodes have state dimensions less than 3.
- (iv) The non-redundant model of Figure 3-7 with state dimension  $k = 4$ . This model makes a better comparison with the models in (i) and (ii), since these three models now all have the same state dimension. Again, the non-redundant model with  $k = 4$  is irregular for modeling a 64-element process.

It is not surprising that the covariances realized by the multiscale models are of comparable accuracy, since the differences among the four are minor. On the other hand, while these multiscale models are designed to model 1-D MRFs exactly, their approximations are much better than a simple 1-D MRF approximation of the steady-state process covariance. This observation further validates the assertion we made earlier in Section 3.4.1 that the end-point linear functionals are capable of realizing a broader class of random processes than just the first-order 1-D MRFs.

Under non-circular boundary conditions and/or spatially varying system parameters, the steady-state process covariance matrix is no longer spatially stationary. In simulations we have observed very similar results for the spatially nonstationary cases; that is, the various approximate models realize the steady-state process covariance to

accuracies on the same order as those shown in Table 4.2. We will not elaborate on the different cases.

### 4.2.2 Functionals for the Steady-State Prediction Error

Consider now the case of principal interest here, namely the multiscale modeling of the one-step predicted error process  $\chi(t|t-1)$ . Except in very special cases, this error process is spatially nonstationary and spatially non-Markov. Nevertheless, we shall see that the choice of multiscale states we just considered for the steady-state process still represent excellent choices for the steady-state error.

Consider the estimation of the same 64-dimensional process just discussed based on a single point measurement. For this small problem we explicitly solved the Riccati equation (3.35) for the optimal steady-state one-step prediction error covariance  $\tilde{\mathbf{P}}_p$ . The inverse of this covariance is full. However, the off-diagonal elements that are outside of the penta-diagonal band corresponding to a MRF structure are generally small, implying that although the steady-state error is not Markov, a second-order MRF approximation is still reasonable, as shown in Figure 4-3 for the case of a single measurement at location 23.

Consider again using CCR to decorrelate the interval  $\{1-32\}$  from  $\{33-64\}$  when the measurement is located is at some point in the former interval. Because of the intrinsic nonstationarity introduced by the tree, the location of the measurement has some influence on the results of CCR. This is illustrated in Figures 4-4 and 4-5, which depict the four most significant linear functionals produced by CCR for two measurement locations: 16 and 32. As before, the two most significant linear functionals are almost completely concentrated on the interval end-points, although the distribution of their values are different for the two cases. Furthermore, the relative significance of the third and fourth linear functionals is a function of measurement location, as shown in Figure 4-6, which clearly indicates that dropping all but the first two linear functionals is a much better approximation for certain measurement locations. In

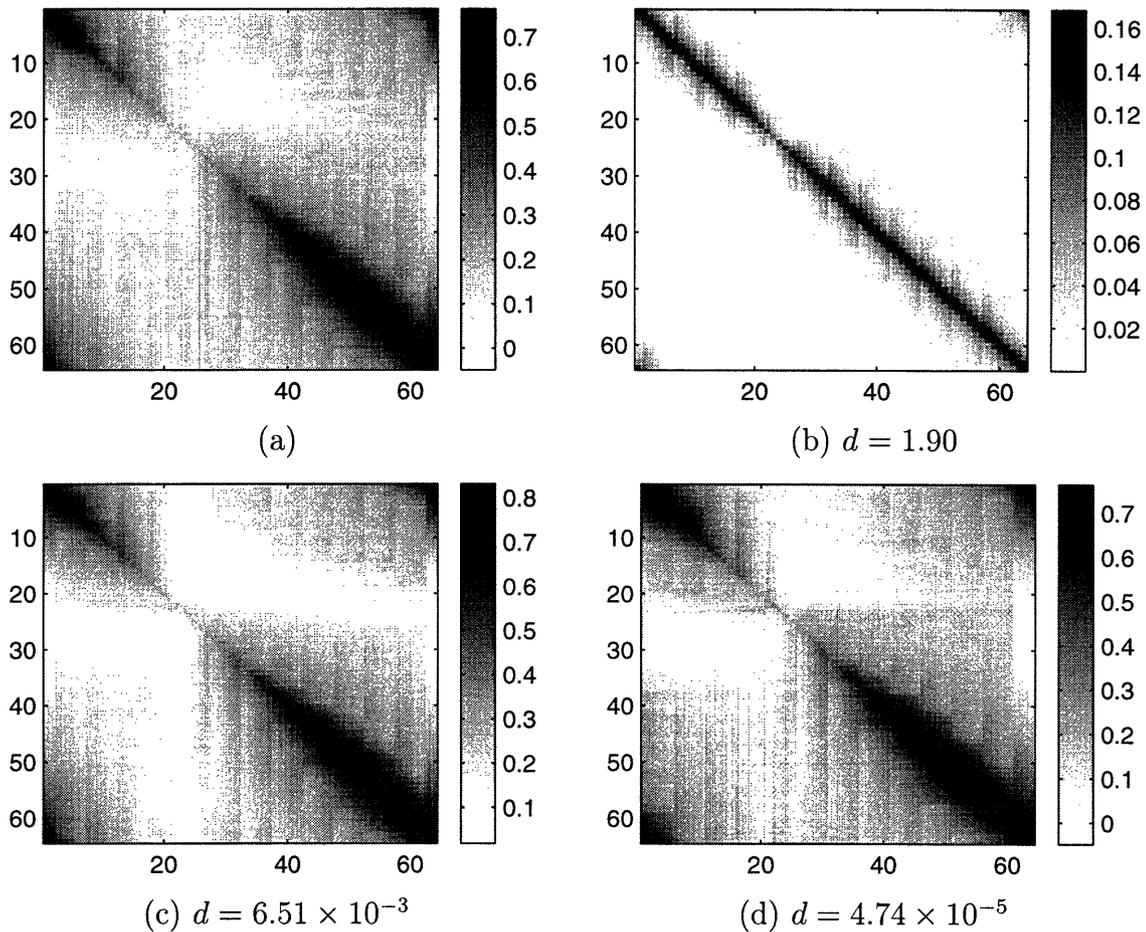


Figure 4-3: A comparison of (a) the exact steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$  of a 64-element discretized diffusion process on a ring with (b) a first-order MRF approximation of the covariance, (c) a second-order MRF approximation, and (d) the covariance realized by an internal model with end-point linear functionals. A single measurement is placed at location 23 (SNR = 0 dB). The Bhattacharyya distance between the realized and the exact covariance is displayed under each panel. ( $\beta = 10$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

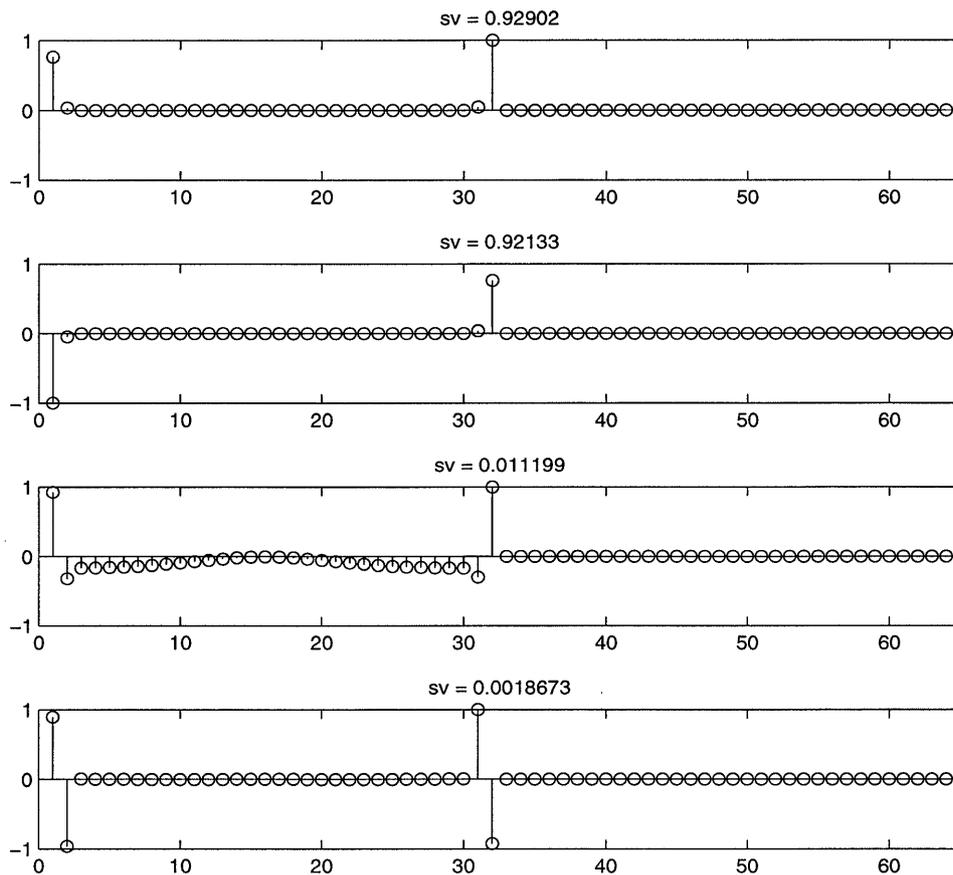


Figure 4-4: The four most significant linear functionals generated by CCR that decorrelate the steady-state predicted estimation errors at locations  $\{1 - 32\}$  from those at locations  $\{33 - 64\}$  of a 64-element diffusion process on a ring with a single measurement at location 16 ( $\beta = 10$ ,  $\text{SNR} = 0$  dB,  $\Delta\tau = 2 \times 10^{-5}$ ). The singular values associated with the linear functionals are displayed above the plots.

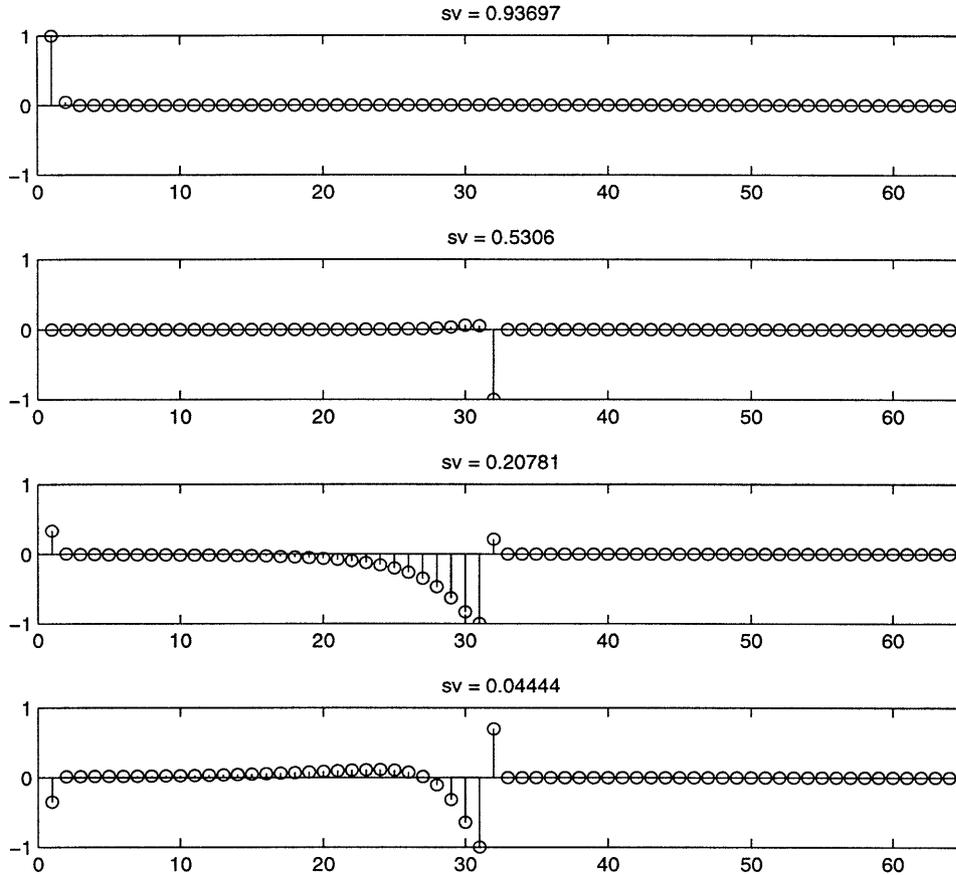


Figure 4-5: The four most significant linear functionals generated by CCR that decorrelate the steady-state predicted estimation errors at locations  $\{1 - 32\}$  from those at locations  $\{33 - 64\}$  of a 64-element diffusion process on a ring with a single measurement at location 32 ( $\beta = 10$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 2 \times 10^{-5}$ ). The singular values associated with the linear functionals are displayed above the plots.

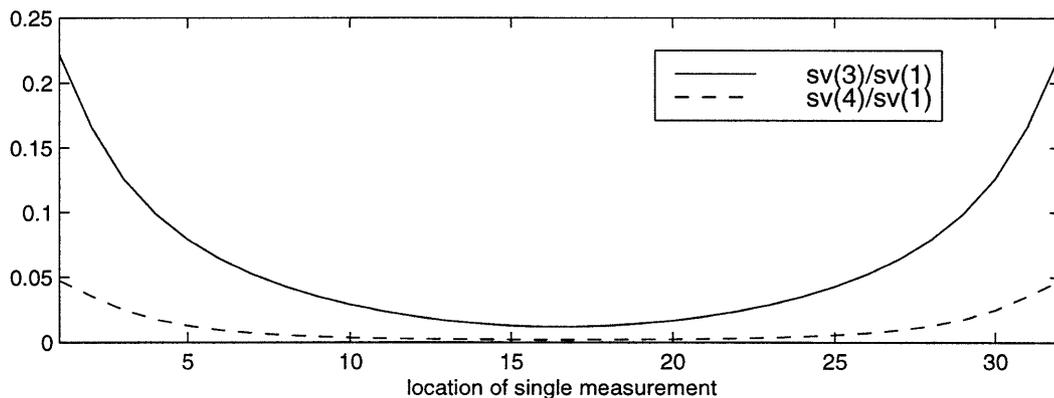


Figure 4-6: The relative significance of the third and fourth most significant linear functionals, measured as the ratio between their associated singular values and the singular value associated with the most significant functional, as a function of the location of the single measurement.

particular, placing measurements near major tree boundaries<sup>7</sup> — in this case near 1 or 32 — is different from placing measurements near the middle of the interval<sup>8</sup>. However, as we now demonstrate, keeping only the first two linear functionals yields excellent results for our purposes regardless of measurement location.

The resulting approximate models for  $\chi(t|t-1)$  can be assessed in at least two different ways. The first is to examine, as we did for the steady-state process, how closely the actual realized covariance of the multiscale model approximate the optimal steady-state predicted error covariance  $\tilde{\mathbf{P}}_p$  computed from the Riccati equation (3.35). As an example, the covariance realized by the end-point linear functionals model in Figure 4-3(d) is compared with the exact covariance in Figure 4-3(a) for the case of a point measurement at location 23. We plot the Bhattacharyya distances between the realized covariances of the four multiscale models and the optimal  $\tilde{\mathbf{P}}_p$  as a function of the measurement location in Figure 4-7. We observe a general trend of increasing fidelity as the measurement is placed further away from major tree boundaries. The Bhattacharyya distance peaks near 1 and 32, becomes smaller near 16, and even smaller near 8 and 24. Note that the peaks for the internal models and those for the non-redundant models do not match exactly as the non-redundant models are irregular and asymmetric for modeling a 64-element process. We see that the CCR internal model realizes the most accurate error covariance. The covariances realized by the internal model with end-point linear functionals and by the non-redundant model of the same state dimension are of comparable quality, while the non-redundant model

---

<sup>7</sup>If the tree distance between two nodes at the finest scale of an internal model is large, i.e., the distance one must traverse on the tree to go from one node to the other is large, then the two fine-scale nodes sit on a major tree boundary. For example in Figure 2-5, points 8 and 9 sit on a major tree boundary, while 6 and 7 are far away from any major tree boundaries.

<sup>8</sup>The reason for this can be understood in terms of the 16-point process depicted in Figure 2-5. Remembering that we are representing processes with cyclic boundary conditions (so that points 1 and 16 are next to each other), we see that the pairs  $\{8, 9\}$  and  $\{1, 16\}$ , while very close spatially, are distant on the tree, so that the strong correlations between these neighboring points must be captured in the state at the root node which separates each of these pairs.

On non-redundant models, e.g., the one depicted in Figure 3-5, although adjacent elements of the physical process are always found at a parent or child node, near major tree boundaries neighboring points of the physical process are scattered over nodes that are farther apart on the tree. For example, elements  $\{9 - 11\}$  are found at nodes over three scales, while  $\{7 - 9\}$  are found at a single scale.

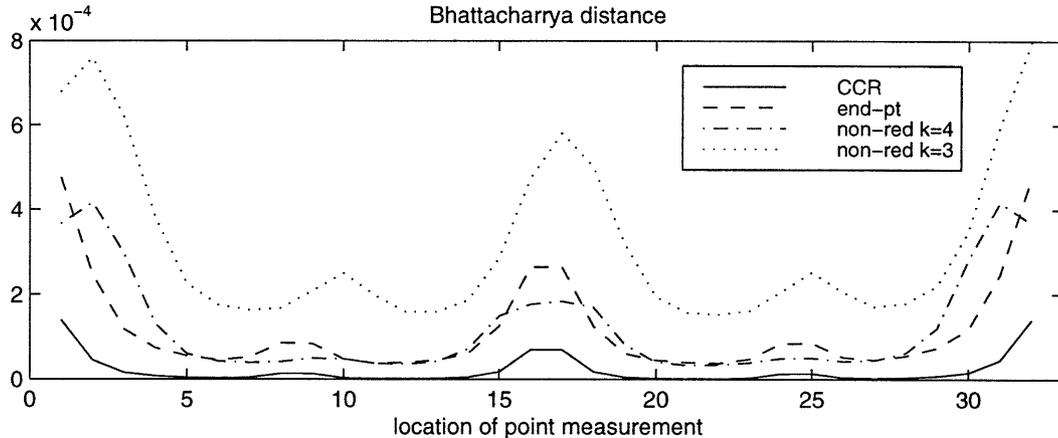


Figure 4-7: The Bhattacharrya distances between the optimal steady-state predicted error covariance  $\tilde{\mathbf{P}}_p$  and the realized covariances from the four suboptimal multiscale models (internal model with linear functionals selected by CCR, internal model with end-point linear functionals, non-redundant model with  $k = 4$ , and non-redundant model with  $k = 3$ ) for a 64-element diffusion process on a ring, plotted as a function of the location of the single measurement on  $\{1 - 32\}$ . ( $\beta = 10$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

of a lower dimension ( $k = 3$ ) is the least accurate. This observation is consistent with what we saw before for the steady-state process (c.f., Table 4.2).

The second and more meaningful assessment is the estimation performance loss incurred by using the approximate multiscale models, using fractional variance reduction (FVR) as a measure of estimator performance as we discussed in Section 3.6. We first calculate the equivalent estimator gain for the approximate estimators and then solve for their steady-state error covariance via (3.36). Figure 4-8 depicts results for two different measurement locations, both far from a major tree boundary (where we would expect our approximation to work best, e.g., at location 23) and right on such a boundary (e.g., at location 32). Figure 4-8(a) shows the optimal estimator FVR (whose performance is invariant to measurement location due to the circular boundary condition). The variance reduction plots for the suboptimal estimators are nearly indistinguishable from the optimal one at the resolution, and are therefore not plotted. The differences between the FVRs of the suboptimal multiscale estimators and the optimal estimator are displayed in panel (b) for a single measurement at location 32 and in (c) for a single measurement at 23. In Figure 4-9 we plot the max-

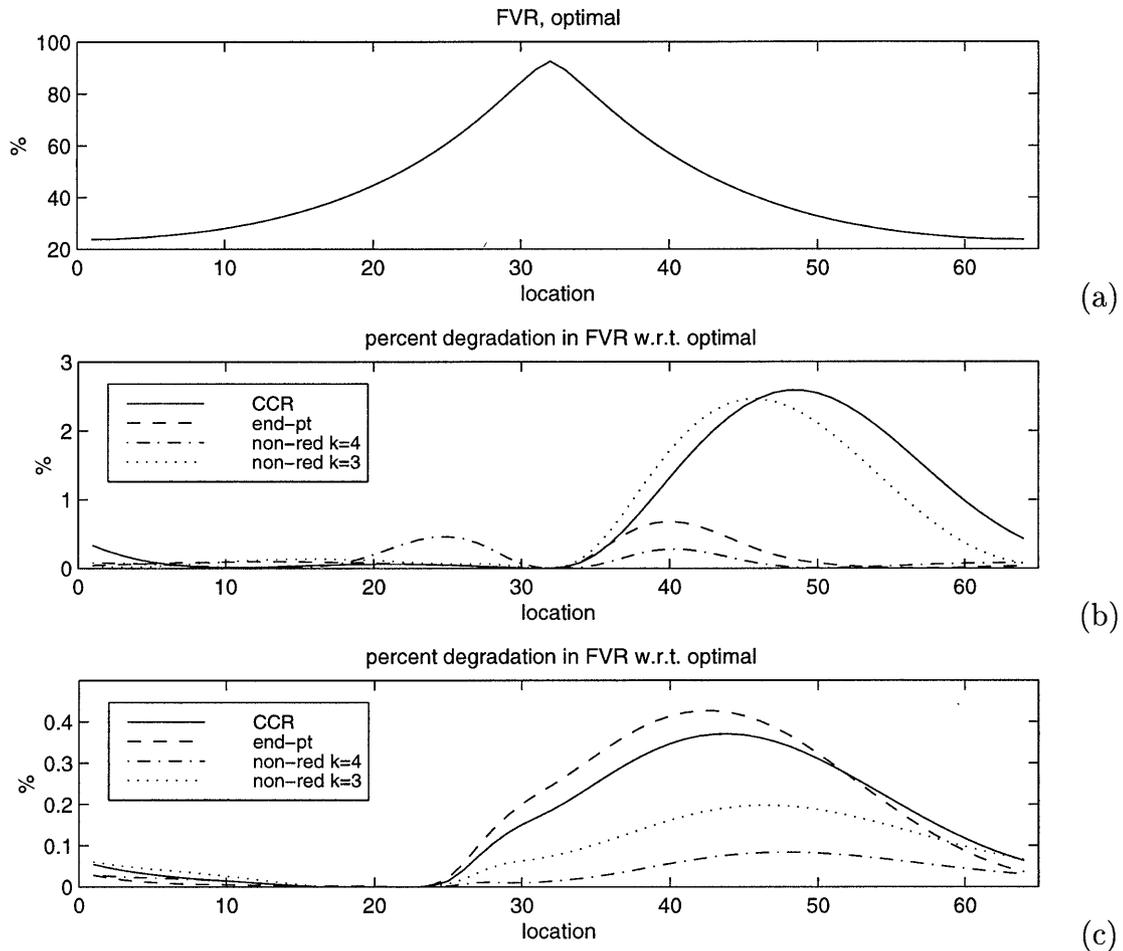


Figure 4-8: (a) The FVR of the optimal estimator in steady state for a 64-element diffusion process on a ring with one point measurement at location 32. FVRs of the optimal steady-state estimator for single point measurements at other locations are simply shifted versions of (a), since the performance of the optimal estimator is shift invariant. (b) The percent degradation in FVR of the suboptimal multiscale estimators with respect to the optimal estimator for a single measurement at location 32. (c) The percent degradation in FVR for a single measurement at location 23. ( $\beta = 10$ ,  $\text{SNR} = 0$  dB,  $\Delta\tau = 2 \times 10^{-5}$ .)

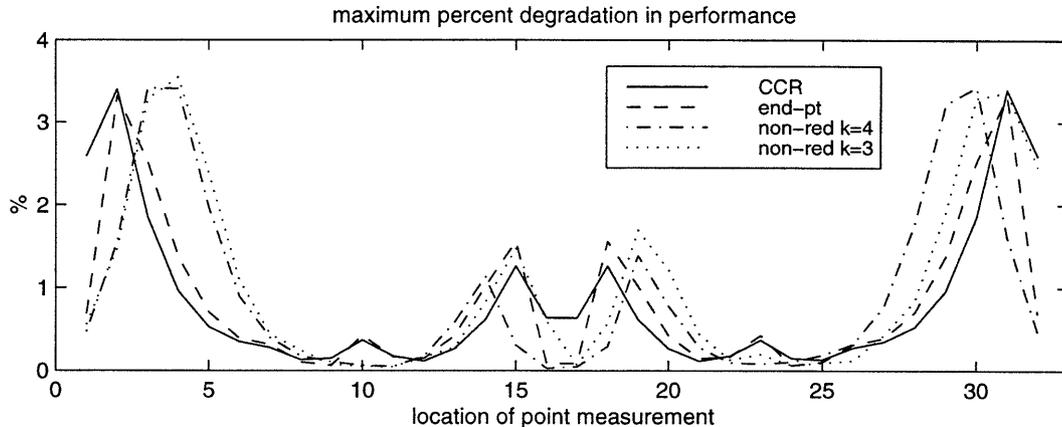


Figure 4-9: The maximum percent degradation in FVR of the suboptimal multiscale models (internal model with linear functionals selected by CCR, internal model with end-point linear functionals, non-redundant model with  $k = 4$ , and non-redundant model with  $k = 3$ ) with respect to the optimal estimator for a 64-element diffusion process on a ring, plotted as a function of the location of the single measurement on  $\{1 - 32\}$ . ( $\beta = 10$ ,  $\text{SNR} = 0$  dB,  $\Delta\tau = 2 \times 10^{-5}$ .)

imum percent degradation, i.e., the maxima of the curves in Figures 4-8(b) and (c), of the suboptimal estimators for each of the measurement locations over  $\{1 - 32\}$ . We observe the same general shape as in Figure 4-7, although the performance of the various multiscale suboptimal estimators here do not differ a great deal. Since the physical process is mapped differently onto the non-redundant models than the internal models, the worst case performance for the different estimators occur at slightly different measurement locations. As expected, the suboptimal estimator has greater degradations in performance for a measurement near major tree boundaries. However, even in this *worst case* the suboptimal estimators have FVRs of no more than a few percent worse than the optimal FVR.

While the analysis done so far has focused on the case of a single point measurement for only a single value of  $\beta$  and SNR, similar results are obtained for other boundary conditions, measurement configurations, SNRs, etc. that we have tested. We show two additional examples for the cases of dense measurements and two point measurements in Figures 4-10 and 4-11 (with non-uniform  $\beta$ , different values of SNR, and non-circular boundary conditions). For large systems, CCR is computationally impractical, but both the end-point linear functional models and non-redundant mod-

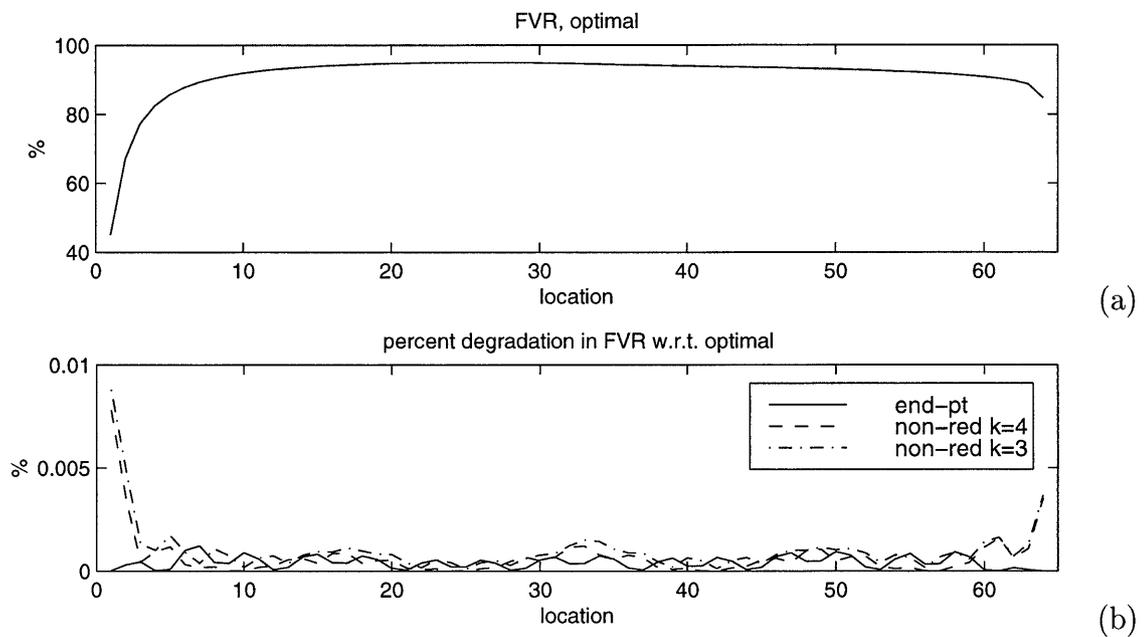


Figure 4-10: (a) The FVR of the optimal estimator in steady state for a 64-element diffusion process on a pinned fin with dense point measurements and non-uniform  $\beta$ :  $\beta = 0$  at locations  $\{1 - 32\}$  and  $\beta = 10$  at  $\{33 - 64\}$ . (b) The percent degradation in FVR of the suboptimal estimators (internal model with end-point linear functionals, non-redundant model with  $k = 4$ , and non-redundant model with  $k = 3$ ) with respect to the optimal estimator. (SNR = 0 dB,  $\Delta\tau = 2 \times 10^{-5}$ .)

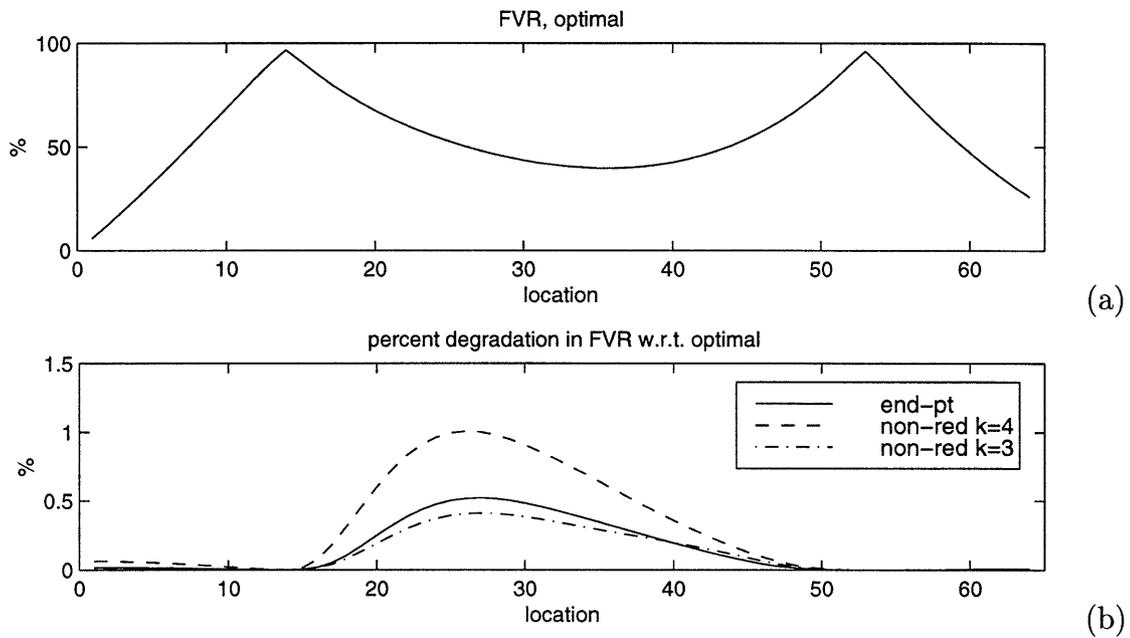


Figure 4-11: (a) The FVR of the optimal estimator in steady state for a 64-element diffusion process on a pinned fin with two point measurements at locations 14 and 53 with  $\text{SNR} = 10$  dB. (b) The percent degradation in FVR of the suboptimal estimators (internal model with end-point linear functionals, non-redundant model with  $k = 4$ , and non-redundant model with  $k = 3$ ) with respect to the optimal estimator. ( $\beta = 0$  at locations  $\{1 - 32\}$ ,  $\beta = 10$  at  $\{33 - 64\}$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

els easily generalize to 1-D processes of any size. Thus, in Figures 4-10 and 4-11, we compare only the estimator performances of the end-point linear functional model and the non-redundant models. The multiscale estimators generally perform comparably. In the following sections, we will consider only the end-point internal model and the standard non-redundant model with  $k = 3$  for comparisons. The analyses and the results discussed in this section lead us to conclude that both the end-point linear functional models and the non-redundant models are more than adequate choices for estimating 1-D diffusion processes.

## 4.3 Multiscale Recursive Estimation Algorithm for 1-D Diffusion

### 4.3.1 Multiscale Prediction Step for 1-D Diffusion

For the examples in the preceding section we explicitly computed the optimal steady-state predicted error covariance and consequently directly determined the linear functionals and the parameters of the multiscale model for the steady-state prediction error  $\chi(t+1|t)$ . The purpose of the analysis in Section 4.2 was to discover the appropriate linear functionals and multiscale models. We showed that both the internal model with end-point functionals and the non-redundant models can adequately capture the steady-state prediction error covariance as well as the steady-state process covariance. For large problems, however, we must adopt the recursive procedure, described in Chapter 3, to construct models for the prediction error process, without assuming the availability of any full covariance matrices.

Recall from Chapter 3 that after the measurement update step, the multiscale estimation algorithm gives us a multiscale model for the updated estimation error  $\chi(t|t)$ , which implicitly defines  $\mathbf{P}_\chi(t|t)$ , the node covariances  $\mathbf{P}(s; t|t)$ , and child-parent covariances  $\mathbf{P}(s, s\bar{\gamma}; t|t)$ . To construct the model for the one-step-ahead predicted estimation error  $\chi(t+1|t)$  we need the joint parent-child statistics for the predicted error model,  $\mathbf{P}(s; t+1|t)$  and  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$ , which can be specified in terms of cross-

covariances among nodes on the updated model. The mixing due to the dynamics  $\mathbf{A}_d$  requires that more distant correlations than just  $\mathbf{P}(s; t|t)$  and  $\mathbf{P}(s, s\bar{\gamma}; t|t)$  on the updated model be calculated.

We have already discussed in some detail in Sections 3.4.2 and 3.5.2 the multiscale prediction step for internal models with end-point linear functionals and for non-redundant models in the case of a tridiagonal  $\mathbf{A}_d$ . This is exactly the situation we have here. The discretized diffusion operator  $\mathbf{A}_d$  is tridiagonal from a forward Euler temporal discretization scheme and both the end-point internal model and the non-redundant model have been found to be appropriate for modeling the steady-state prediction error process. For 1-D problems, using non-redundant models holds clear advantage. While estimators based on the end-point internal model is of  $\mathcal{O}(N \log N)$  for the prediction step<sup>9</sup>, those based on non-redundant models are faster with  $\mathcal{O}(N)$  complexity. The trade-off between accuracy and computational complexity is small. We shall see in Section 4.4 that estimators based on the end-point internal model (with

---

<sup>9</sup>For end-point internal models, unlike non-redundant models (c.f., Section 3.5.2), keeping track of which cross-covariances between nodes on the updated tree must be computed for the multiscale prediction step can be complicated. Although the results and examples in Section 4.4 are produced with all the needed entries computed exactly, for implementation reasons it can be advantageous at times to compute only a subset of the cross-covariance pairs exactly, e.g., among nodes along direct descendant lines, and approximate the remainder [48]. This essentially becomes a covariance extension problem: we have a subset of the entries of  $\mathbf{P}_\chi(t|t)$  and from this knowledge we want to estimate some of the other entries in this matrix. Because of the unusual pattern of the known entries, a global solution to this problem is complex. However, our need for these missing entries is simply to allow us to compute the parameters in the multiscale model for  $\chi(t+1|t)$  using (2.18), (2.19). Consequently we may adopt the approach of computing each of the needed elements independently and using only local information.

For example, suppose that we need to compute the  $(k, l)$  element  $p_{kl}$  of  $\mathbf{P}_\chi(t|t)$ , i.e., the covariance between  $\chi_k(t|t)$  and  $\chi_l(t|t)$ . One approach assumes local stationarity: if  $p_{k-1, l-1}$  or  $p_{k+1, l+1}$  or both are available, we might approximate  $p_{kl}$  as being equal to either of these or to their average. An alternative, more general approach goes as follows: suppose we can find a third component  $\chi_n(t|t)$  so that the cross-covariances between this component and both  $\chi_k(t|t)$  and  $\chi_l(t|t)$  are known, i.e., so that only  $p_{kl}$  is unknown in the  $3 \times 3$  covariance matrix

$$\begin{bmatrix} p_{kk} & p_{kn} & p_{kl} \\ p_{kn} & p_{nn} & p_{nl} \\ p_{kl} & p_{nl} & p_{ll} \end{bmatrix}$$

Then we choose for  $p_{kl}$  the maximum entropy completion [27, 65] of this covariance:  $p_{kl} = p_{kn}p_{nl}/p_{nn}$ . An interpretation of this choice is that it is equivalent to assuming that  $\chi_k(t|t)$  and  $\chi_l(t|t)$  are conditionally decorrelated given  $\chi_n(t|t)$ ; that is, if  $n$  is between  $k$  and  $l$  this assumes that the components of  $\chi(t|t)$  form a linearly-ordered 1-D MRF, which is consistent with our choice of end-point linear functionals.

$k = 4$ ) often perform better than estimators based on the non-redundant model with  $k = 3$ , although the difference in performance is mostly attributable to the difference in state dimension and measurement configurations, and largely disappears for the non-redundant model with  $k = 4$ .

### 4.3.2 Iterative and Recursive Implementations

The multiscale recursive algorithm can be used in one of two ways. Our original motivation for using this approach was to obtain an approximate multiscale model for the steady-state one-step prediction error process. For that purpose, the algorithm is run iteratively off-line until convergence is achieved. Alternatively, we can use this algorithm to provide a multiscale error model dynamically at each step of the recursive estimation procedure for the transient phase of estimation or for problems with temporal nonstationarities. Examples in Section 4.4 will cover both of these cases.

#### Initialization

The initialization of our algorithm takes the form of specifying a multiscale model for the prior estimation errors. To iteratively approximate the steady-state error statistics, the choice of model is not so critical. For recursive estimation during the transient phase or for time-varying problems we are more interested in using an initial model that accurately reflects the prior statistics. For example, a common choice for the initial covariance is the steady-state process covariance which are easily calculated using the FFT, if the dynamics are space-invariant and circular boundary conditions are assumed.

#### Stopping Criterion for the Iterative Algorithm

For the iterative calculation of a multiscale model for the steady-state estimation errors, a critical issue is the iteration stopping criterion. 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  $\mathbf{A}_d(\mathbf{I} - \mathbf{K}(\infty)\mathbf{C}_d)$ , and thus choosing the number of iterations to be several times this time constant provides a conservative bound. For the examples shown later in this section, for which we can compute the slowest time constants of the steady-state error dynamics  $\text{TC}_{\mathbf{K}(\infty)}$ , we terminate the iterations at about one time constant. Although the estimator may not have converged, the results there demonstrate that the performance of the resulting estimators is surprisingly close to the optimal Kalman filter.

However, (i) for large problems this time constant will generally be unavailable, and (ii) taking this conservative approach may lead to an excessive numbers of iterations. In response to (i) an alternative, adaptive stopping criterion is to examine the diagonal elements of  $\mathbf{P}_x(t|t)$  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 (ii), we may restrict the number of iterations to be  $\mathcal{O}(\log N)$  so that the total complexity is  $\mathcal{O}(N \log N)$ . This, of course, implies that the resulting multiscale estimator may not have converged and is a poorer approximate steady-state estimator.

### Accuracy

There are several sources of inaccuracy in the realized error covariance on the multiscale tree. One source is clearly the stopping of the iterations prior to convergence. Two other sources are errors due to the temporal discretization of the dynamics and choosing end-point linear functionals as the basis for the multiscale modeling of the estimation error. While 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, higher-order multiscale models with more linear functionals, we have found through extensive testing that the choice made here provide excellent results while preserving the computational advantages of our approach.

## 4.4 Examples and Results

In this section we illustrate the application of our multiscale 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. We emphasize again that the performance figures we saw before were for suboptimal multiscale estimators built from the exact steady-state error covariance for the purpose of designing the appropriate linear functionals and multiscale models. The steady-state suboptimal multiscale estimators shown here are constructed iteratively using the multiscale recursive algorithm.

The multiscale recursive algorithm can in principle be applied more generally to any dynamic process whose steady-state error process is 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 sections.

### 4.4.1 Cooling Ring

We start with a cooling ring and a single measurement, which has been used as an example in the previous sections. The steady-state process variance has been normalized to 1. Figure 4-12(a) shows the variance reduction plots for several values of SNR and heat loss parameter  $\beta = 10$ . On the whole, as the signal to noise ratio increases, so does the percent variance reduction. In all cases, when compared to the optimal estimator in steady state, the multiscale estimators perform less than one percent poorer, as seen in panel (b) for an estimator using an internal model with end-point linear functionals and in panel (c) for an estimator using a non-redundant model with  $k = 3$ . The greatest degradation in performance occurs in regions furthest away from the measurement, i.e., regions where the error variances are expected to be large anyway.

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

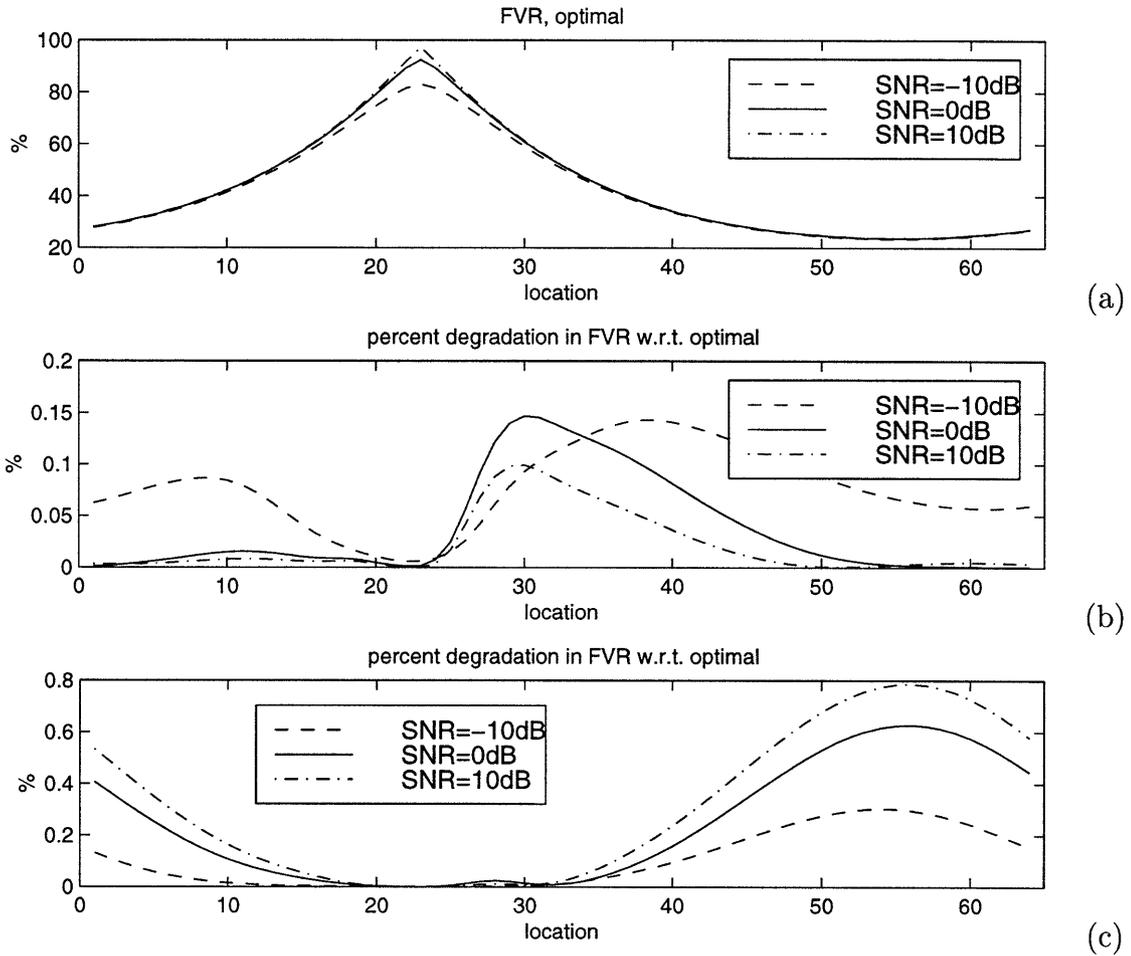


Figure 4-12: Cooling ring example for a 64-element diffusion process with one point measurement at location 23 with SNR = -10 dB, 0 dB, and 10 dB. (a) The FVR of the optimal estimator in steady state. (b) The percent degradation in FVR of the multiscale suboptimal estimator with end-point linear functionals with respect to the optimal. (c) The percent degradation in FVR of the multiscale estimator with non-redundant model with  $k = 3$  with respect to the optimal. ( $TC_{K(\infty)} \approx 1730$  steps = 0.0346 sec, multiscale iterations terminated at step 2000,  $\beta = 10$ ,  $\Delta\tau = 2 \times 10^{-5}$ .)

steady-state multiscale estimator as compared to the optimal is generally better than those shown in Figure 4-12; such plots are not repeated here.

#### 4.4.2 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, i.e.,  $z(0, \tau) = z_0$ . At the free end, the heat flux is set to be equal to the heat loss,  $\partial z(1, \tau)/\partial l = -\beta z(1, \tau)$ . The second variation recognizes the fact that in practice the heat loss parameter  $\beta$  may be spatially varying if the coolant is non-homogeneous [1], for example, when the fin is partially insulated or cooled by air and partially immersed in a liquid coolant. The discretized dynamic equation (4.6) will then have a non-circulant  $\mathbf{A}_d$  and an extra term  $\mathbf{B}_d u(t)$  to account for the boundary condition:

$$\mathbf{z}(t+1) = \mathbf{A}_d \mathbf{z}(t) + \mathbf{B}_d u(t) + \mathbf{w}_d(t), \quad (4.16)$$

where  $u(t)$  contains the boundary conditions which may be time varying.

While the development in the preceding sections demonstrated the adequacy of the end-point linear functionals mostly for the cooling ring case, we find that for the pinned fin and other examples in this section the end-point linear functional models and the non-redundant models are robust enough to model the steady-state error process accurately. Figure 4-13(a) shows the spatially nonstationary steady-state process variance and the steady-state error variance of the optimal estimator for the case of two point measurements. The steady-state error variances of the suboptimal multiscale estimators are so close to that of the optimal estimator that they would be indistinguishable in the plots and are therefore not plotted in either panel (a) or (b). The performance degradation of the suboptimal estimators as compared to the optimal is only a fraction of a percent as seen in panel (c). The near equality of the true optimal error variance and the actual error variances of our suboptimal estimators

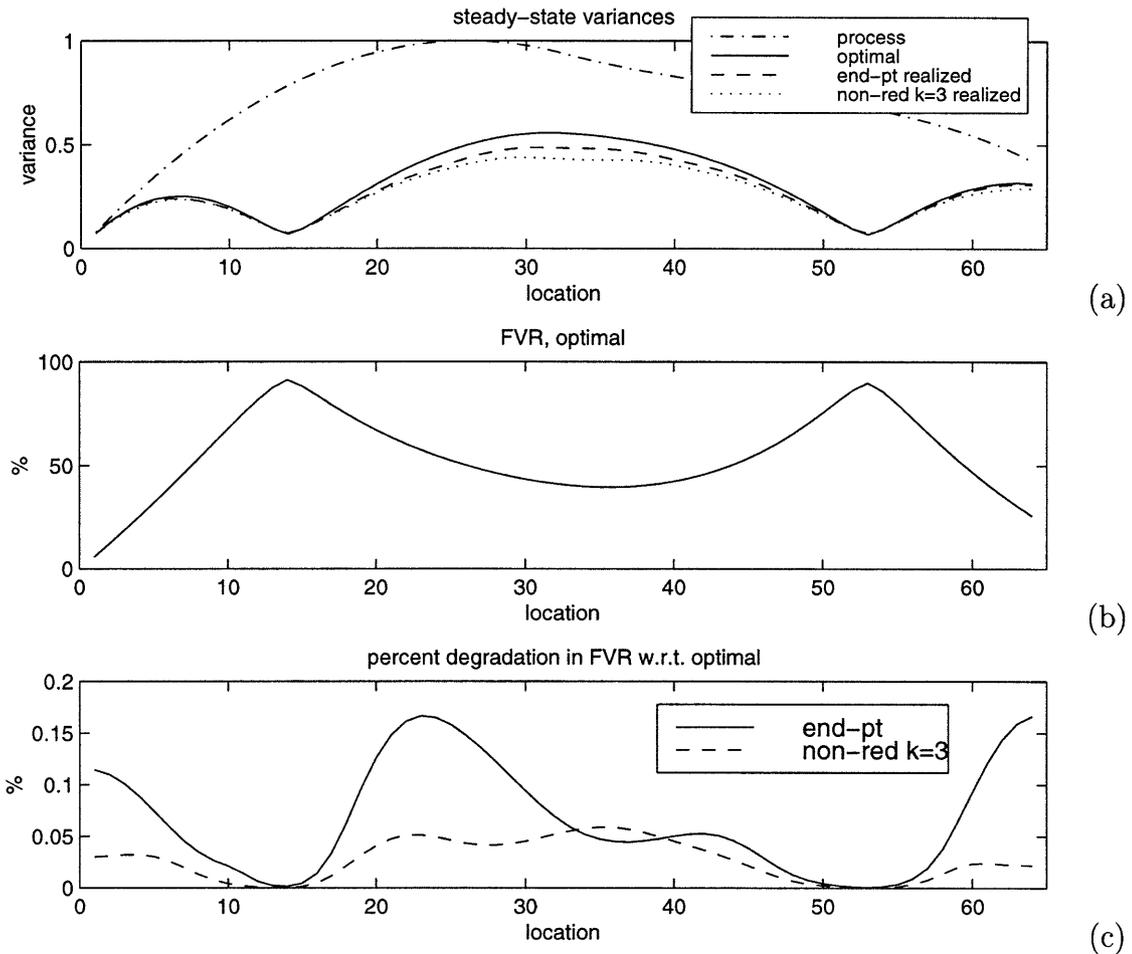


Figure 4-13: Pinned fin example with two measurements at locations 14 and 53 with spatially varying heat loss parameter  $\beta = 0$  at locations  $\{1 - 32\}$  and  $\beta = 10$  at  $\{33 - 64\}$ . (a) The steady-state process variance, steady-state updated estimation error variance of the optimal estimator, and the updated error variances realized by the suboptimal multiscale estimators (internal model with end-point linear functionals and non-redundant model with  $k = 3$ ). The realized error variances are what the multiscale estimators believe they are achieving. Both multiscale estimators underestimate the actual error variances. The true steady-state error variances of the suboptimal multiscale estimators are not plotted, as they are so close that of the optimal that they would be indistinguishable in the plot. (b) The FVR of the steady-state optimal estimator. (c) The percent degradation in FVR of the multiscale estimators with respect to the optimal. ( $TC_{K(\infty)} = 1017$  steps = 0.0203 sec, multiscale iterations terminated at step 1017, SNR = 0 dB,  $\Delta\tau = 2 \times 10^{-5}$ .)

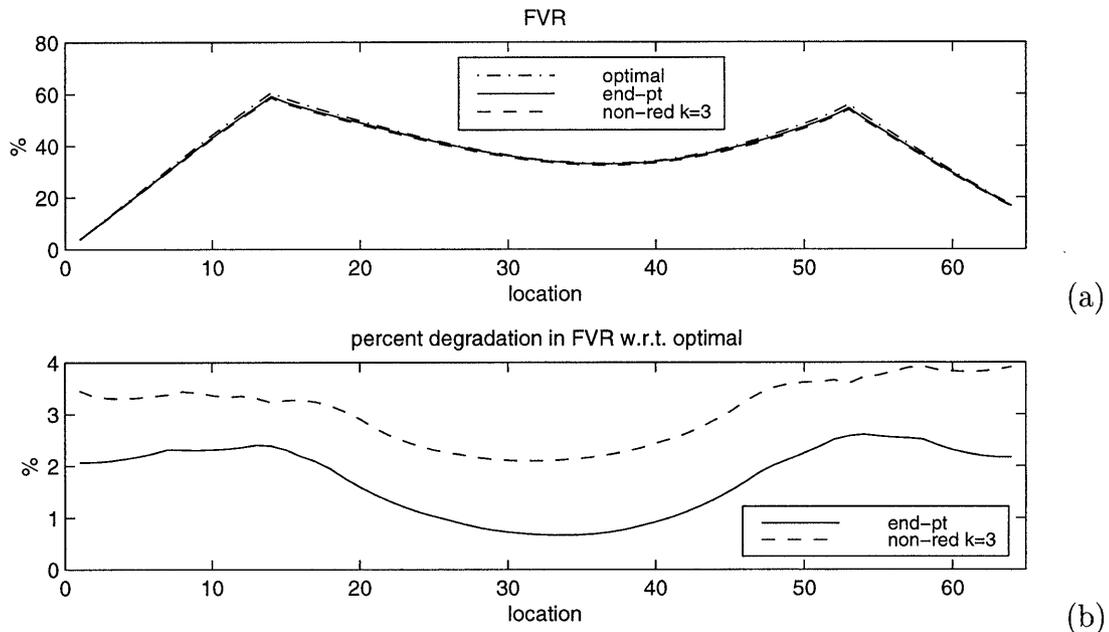


Figure 4-14: Pinned fin example as in Figure 4-13 except that there is one measurement update step for every 200 prediction steps. (a) The FVRs of the steady-state estimators. (b) The percent degradation in FVR of the suboptimal multiscale estimators with respect to the optimal. ( $TC_{K(\infty)} = 1216$  steps = 0.0243 sec, multiscale iterations terminated at step 1200 after the measurement update there,  $\beta = 0$  at  $\{1 - 32\}$  and  $\beta = 10$  at  $\{33 - 64\}$ , SNR = 0 dB,  $\Delta\tau = 2 \times 10^{-5}$ .)

demonstrate the excellent performance of our multiscale 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  $\chi(t|t)$  which provides the values of the variances that this estimator *believes* it is achieving. The realized updated estimation error variances are illustrated in Figure 4-13(a). Note that these variances are also fairly accurate, although they underestimate the actual error variance.

As we discussed in Section 4.1, in order to simplify the procedure for constructing multiscale models for prediction errors we used a tridiagonal model for the discretized dynamic matrix  $\mathbf{A}_d$ . This model produces limited mixing among the scales in our error models and consequently simplifies the model construction process during the prediction step. In order to test the impact of these approximations and the ability of our procedure to deal with substantial mixing we show an example here in which the measurement sampling rate is substantially slower than the update rate for the

dynamics. Figure 4-14 shows the same pinned fin example of Figure 4-13 except that a measurement is only available every 200 prediction steps thus allowing substantial mixing to occur between measurement updates. Since many predictions steps must be taken for every update step, we expect that the effects of the approximations built in to our multiscale recursive algorithm would be more pronounced. As expected, the approximate multiscale estimators perform poorer now than in the previous case. Yet in the worse case, it is still within 4% of the optimal estimator<sup>10</sup>.

### 4.4.3 Advection-Diffusion

In order to test the breadth of utility of our methodology we extend 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 [74] to modeling tracer movements in the ocean [95, 94]. Although the end-point linear functional models and the non-redundant models were justified by a detailed examination of diffusion in Section 4.2, the robustness of our multiscale estimator gives us sufficient confidence to apply it to advection-diffusion processes as well. Adding an advection term to (4.5):

$$\frac{\partial z(l, \tau)}{\partial \tau} = \frac{\partial^2 z(l, \tau)}{\partial l^2} + \rho \cdot \frac{\partial z(l, \tau)}{\partial l} - \beta(l) \cdot z(l, \tau) + \gamma \cdot w(l, \tau), \quad (4.19)$$

which models the temperature distribution along a thin pipe. Note that we write the heat loss parameter as  $\beta(l)$  here to indicate possible spatial dependency.

---

<sup>10</sup>To calculate the optimal steady-state estimator, we can write an augmented dynamic equation as follows. Given the one-step temporal dynamic equation,  $z(t+1) = \mathbf{A}_d z(t) + \mathbf{w}_d(t)$ , where the process covariance is  $\mathbf{Q}_d(t)$ , over two steps we may write

$$z(t+2) = \mathbf{A}_d^2 z(t) + \mathbf{A}_d \mathbf{w}_d(t) + \mathbf{w}_d(t+1). \quad (4.17)$$

Over  $i$  steps,

$$z(t+i) = \mathbf{A}_c z(t) + \mathbf{G}_c \mathbf{w}_c(t), \quad (4.18)$$

where  $\mathbf{A}_c = \mathbf{A}_d^i$ ,  $\mathbf{w}_c(t) = [\mathbf{w}_d(t)^T \ \mathbf{w}_d(t+1)^T \ \dots \ \mathbf{w}_d(t+i)^T]^T$ ,  $\mathbf{G}_c = [\mathbf{A}_d^{i-1} \ \mathbf{A}_d^{i-2} \ \dots \ \mathbf{A}_d^0]$ , and the covariance of  $\mathbf{w}_c(t)$ ,  $\mathbf{Q}_c(t)$  is block diagonal with blocks  $\mathbf{Q}_d(t)$ ,  $\mathbf{Q}_d(t+1)$ ,  $\dots$ ,  $\mathbf{Q}_d(t+i)$ . The matrices  $\mathbf{A}_c$ ,  $\mathbf{G}_c$ , and  $\mathbf{Q}_c(t)$  are used in the Riccati equation (3.35) to solve for the steady-state error covariance of the optimal estimator.

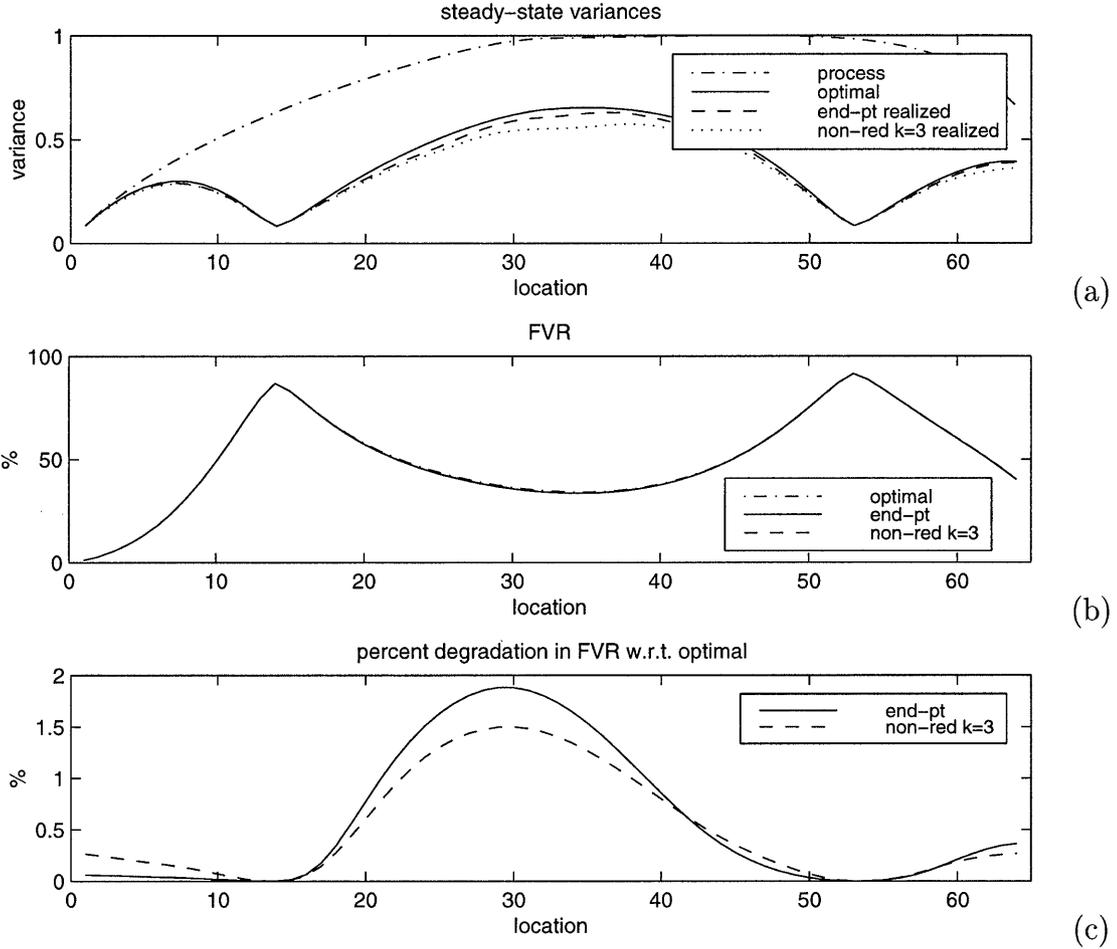


Figure 4-15: Thin cooling pipe example of a 64-element advection-diffusion process, in which a fluid flows to the right ( $\rho = -10$ ) from a reservoir at  $l = 0$  through a thin pipe that is half insulated ( $\beta = 0$  at locations  $\{1 - 32\}$ ) and half exposed ( $\beta = 10$  at  $\{33 - 64\}$ ). (a) The steady-state process variance, steady-state updated estimation error variance of the optimal estimator, and the updated error variances realized by the suboptimal multiscale estimators (internal model with end-point linear functionals and non-redundant model with  $k = 3$ ). (b) The FVRs of the steady-state optimal estimator and the multiscale estimators. (c) The percent degradation in FVR of the suboptimal estimators with respect to the optimal. (Two measurements at locations 14 and 53, SNR = 0 dB,  $TC_{K(\infty)} = 551$  steps = 0.011 sec, multiscale iterations terminated at step 551,  $\Delta\tau = 2 \times 10^{-5}$ .)

Figure 4-15 displays the results from a thin pipe example, in which a fluid flows to the right from a reservoir at  $l = 0$  through a thin pipe that is half insulated and half exposed. For  $\rho = -10$  in this example, it takes 0.1 sec or 5000 steps for a particle to flow from one end of the pipe to the other end. Two point measurements are available at locations 14 and 53. The multiscale approximate steady-state estimators again track the performance of the optimal estimator closely. In addition, the error variances realized by the multiscale models (shown in Figure 4-15(a)) provide a fairly good approximation to the actual error statistics.

#### 4.4.4 Recursive Implementation and Temporally Nonstationary Performance

Finally, we illustrate the performance of the recursive version of our algorithm in order to see how well it does in temporally nonstationary situations. In particular, in Figure 4-16 we illustrate results for one such experiment using the advection-diffusion dynamics in (4.19). We start this process from a nonequilibrium initial condition, as shown in Figure 4-16(a), which may, for instance, represent a segment of warmer gas in the leftmost quarter of the pipe. As time goes on, this segment of warm gas moves to the right and cools down. The measurements are taken once every 100 prediction steps and are highly nonstationary in time: the number of measurements at each measurement time is randomly generated as a Poisson process with a mean value of 4, and the measurement locations are 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 500 (i.e., after the fifth update) for a particular simulation. The actual process at this time (displaying both advection to the right and diffusion), is depicted in Figure 4-16(a), while estimation results just after the measurement update at this time are shown in Figure 4-16(b). There are three measurements taken at this particular update step with the locations and values indicated by the small circles. As these figures illustrate, the estimates produced by the optimal Kalman filter and by our multiscale recursive estimators are

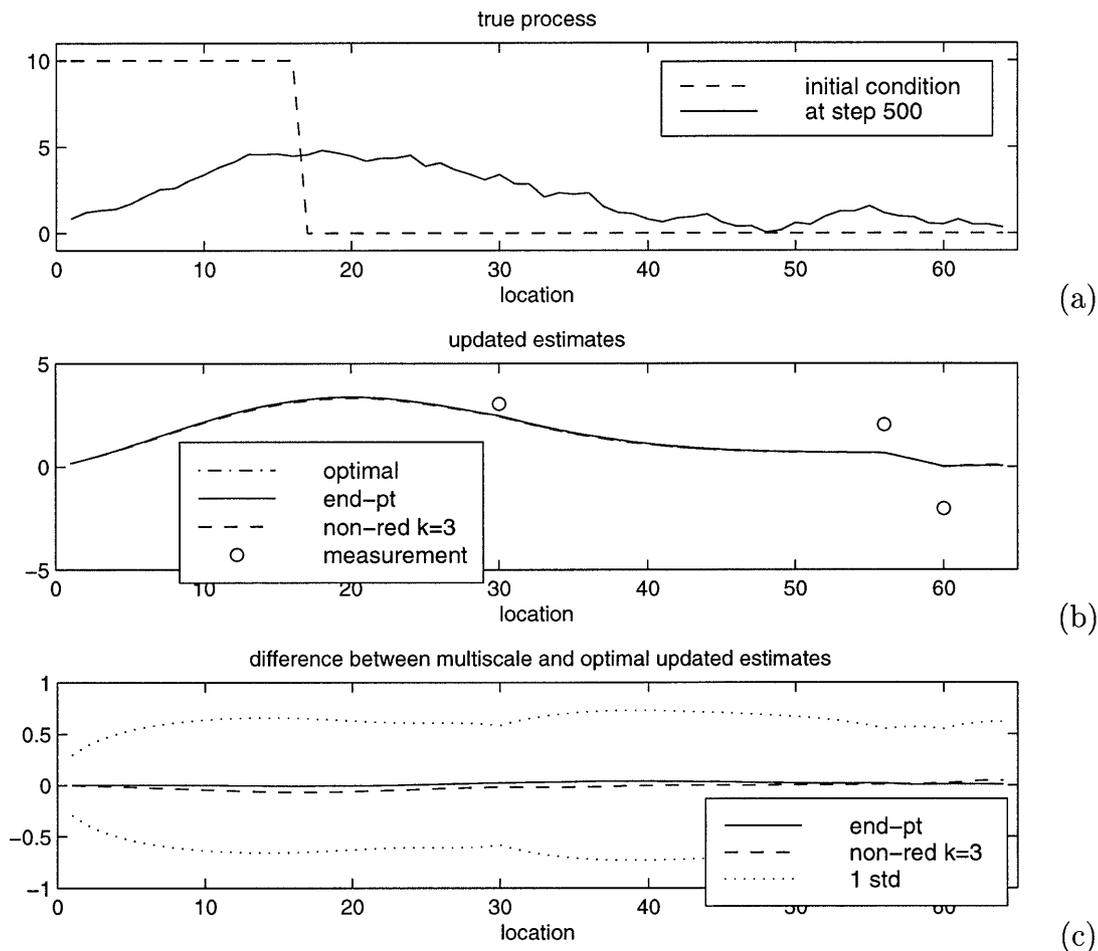


Figure 4-16: Thin cooling pipe example as in Figure 4-15 except that there is one measurement update step for every 100 prediction steps. The number of measurements at each update step is a Poisson distribution with mean 4. The locations of the measurements are uniformly distributed. (a) The initial values of the process and the true process at step 500. (b) The updated estimates of the optimal and the suboptimal multiscale estimators (internal model with end-point linear functionals and non-redundant model with  $k = 3$ ) at step 500. The locations and values of the measurements at this particular update step are labeled with circles. (c) Differences between the updated estimates from the multiscale estimators and those from the optimal. The dotted curves show the magnitude of one standard deviation of the updated estimation error of the optimal estimator at step 500. (SNR = 0 dB, boundary condition  $z_0 = 0$ .)

virtually identical. In fact the differences between the estimates are statistically insignificant when compared to the estimation error standard deviations of the Kalman filter, as seen in Figure 4-16(c), where the differences between the multiscale estimates and the optimal estimates are plotted in comparison with the error standard deviations of the Kalman filter.



# Chapter 5

## Multiscale Dynamic Estimation of 2-D Diffusion

### 5.1 Challenges in Modeling 2-D Random Fields

While the multiscale modeling and recursive estimation formalism in the preceding chapters was developed and illustrated with examples of 1-D spatial processes, our ultimate goal is the application of this formalism to dynamic systems in two or even three spatial dimensions, where we encounter truly large-scale problems and where the real potential benefit of our approach is to be reaped. Although the basic concept of multiscale modeling and recursive estimation is general, the added spatial dimension means that the computational efficiency and near-optimal performance achieved in 1-D cannot be easily duplicated in 2-D without resolving some important issues. The choice of multiscale states in the representation of estimation error fields represents a first important problem.

We saw in Section 2.3.2 that the direct extension to 2-D Markov random fields (MRFs) of the exact internal realizations of 1-D MRFs, which keep mid- and end-points of subintervals as linear functionals (depicted in Figure 2-5), requires keeping dense boundary and midline points of subregions as linear functionals (depicted in Figure 2-6). We refer to the functionals in Figure 2-6 as *boundary-point linear functionals*, just as we referred to the functionals shown in Figure 2-5 as *end-point linear*

functionals. The major difference between the 1-D and 2-D exact realizations of MRFs is the dependence of the multiscale state dimension  $k$  on the scale of node  $s$ ,  $m(s)$ , and on the size of the field of interest  $N$ . In 1-D models,  $k$  is constant, independent of  $m(s)$  and  $N$ , so that estimation on the tree with the multiscale smoothing algorithm has computational complexity  $\mathcal{O}(N)$ . In 2-D, the state dimension of the exact model is a function of  $m(s)$  and  $N$  such that the computational complexity becomes  $\mathcal{O}(N^{3/2})$ . The major challenge in multiscale modeling of 2-D random fields is the construction of accurate models with low state dimensions to achieve better computational efficiency, either with a lower asymptotic complexity, e.g.,  $\mathcal{O}(N \log N)$ , or of the same  $\mathcal{O}(N^{3/2})$  but with a significantly smaller constant of proportionality requiring less computation.

Previous efforts in realizing 2-D Markov random fields [67, 73] have had to grapple with the trade-off between state selection and computational complexity. Since the linear functionals at a node conditionally decorrelate the random variables on the subtrees connected to that node, the number of linear functionals needed for decorrelation reflects, in a sense, the correlatedness of the subtrees. Reduction in state dimension is possible if there exists additional structure in the correlations. The texture modeling examples in [67, 70] exploit additional correlations by keeping the coefficients of the wavelet transform of the boundary point set as states. Depending on the type of texture or, equivalently, the correlation structure, a much smaller number of coefficients than the number of boundary and midline points are needed to capture most of the conditional correlations.

For more general Gaussian random fields, the CCR method provides a general method for determining the set of suitable linear functionals and for ranking their significance for conditional decorrelation. However, more so in 2-D than 1-D, the computational complexity of CCR at  $\mathcal{O}(N^4)$  is simply infeasible for anything but the smallest 2-D random fields. In applications where we have reasons to believe that most of the conditional correlation is captured by the boundary point set and that neighboring points along each such boundary are highly correlated, we may choose to keep only a subsampled set of the boundary points as the linear functionals at

that node. This approach is used to model ocean hydrography in [73, 31]. For a fixed subsampling rate, the number of linear functionals kept at a node, i.e., the state dimension, still grows with the size of the random field,  $N$ , such that the asymptotic computational complexity is no different from the dense boundary point case. Nevertheless, the amount of computation may be reduced to the point that problems too large to be solved by other methods can now be solved.

In the following sections, we first analyze what type of linear functionals and multiscale models are suitable for modeling the steady-state 2-D diffusion process as well as the steady-state prediction errors. The multiscale models we consider include internal models with functionals selected by a CCR analysis, internal models with boundary points of 2-D subregions, and non-redundant models. We examine how accurately each of these types of models and their reduced-order versions capture the steady-state prediction error statistics. More importantly we are interested in understanding how difficult the prediction step in the multiscale recursive algorithm is for each type of model and what the trade-off is between computational complexity and performance of the multiscale recursive estimators. We conclude the chapter with several examples.

## 5.2 Linear Functionals for 2-D Diffusion

Consider a normalized isotropic 2-D diffusion process, governed by a PDE similar to (4.5):

$$\frac{\partial z(l_1, l_2, \tau)}{\partial \tau} = \frac{\partial^2 z(l_1, l_2, \tau)}{\partial l_1^2} + \frac{\partial^2 z(l_1, l_2, \tau)}{\partial l_2^2} - \beta \cdot z(l_1, l_2, \tau) + \gamma \cdot w(l_1, l_2, \tau), \quad (5.1)$$

where  $z(l_1, l_2, \tau)$  is the temperature at location  $(l_1, l_2)$  at time  $\tau$ ,  $w(l_1, l_2, \tau)$  is the white Gaussian process noise with unit variance. Similar to what was done in the 1-D case, we discretize this PDE into a system of difference equations in the form of (3.1) or (4.6):

$$\mathbf{z}(t+1) = \mathbf{A}_d \mathbf{z}(t) + \mathbf{w}_d(t), \quad (5.2)$$

where the process vector  $\mathbf{z}(t)$  is a lexicographically ordered vector containing the elements of the 2-D field. A larger heat loss term shortens the correlation length of the diffusion process and needs finer spatial resolution. Since the linear dimension for the 2-D examples shown in this chapter, at 16 or 17 (i.e.,  $16 \times 16$  or  $17 \times 17$  2-D fields), is smaller than the 64-dimensional 1-D examples, a smaller heat loss parameter  $\beta = 1$  is used here. Also, for the examples in this chapter, we often assume, for symmetry, circular boundary conditions, in which case the 2-D field represents the temperature distribution on a toroid. In other cases, for the more realistic setup of a cooling sheet we assume Dirichlet conditions, where the temperature is fixed, and type 3 homogeneous boundary conditions, in which the temperature gradient is equal to the heat loss term  $\beta z$  along the edge.

A number of properties associated with 1-D diffusion can be generalized to the 2-D case. Of importance is the near Markov spatial correlation structure of the steady-state diffusion process. Generalizing from 1-D diffusion, we expect the steady-state process of the discretized 2-D diffusion process to be nearly a first-order Markov random field. Indeed the inverse covariance matrix  $\mathbf{P}_z^{-1}$  of the steady-state process is found to be almost exactly block penta-diagonal, where the main diagonal blocks are penta-diagonal submatrices, the first off-diagonal blocks are tridiagonal submatrices, and the second off-diagonal blocks are diagonal submatrices. This pattern corresponds to a second-order Markov structure, although the entries in the second off-diagonal blocks and the second off-diagonal entries of the main diagonal blocks are much smaller compared to the other entries. Furthermore, with decreasing time step size, entries of  $\mathbf{P}_z^{-1}$  corresponding to a second-order MRF become smaller and the steady-state process approaches a first-order MRF.

Similar to the 1-D case, the steady-state prediction error field in 2-D is not Markov. However, entries of the inverse covariance of the steady-state predicted estimate error  $\tilde{\mathbf{P}}_p^{-1}$  outside of the set of non-zero entries associated with a second-order MRF are small. The earlier results on 1-D diffusion gives us some confidence that the MRF approximation and multiscale models built for MRFs may be adequate for 2-D diffusion too. We will illustrate our analysis with a relatively small  $16 \times 16$  field in the

following discussion. Note that (5.2) is already a 256-dimensional state equation in time at this size. Even though our multiscale algorithms are aimed at solving much large problems, our choice of  $16 \times 16$  examples is mainly for comparison with the optimal solutions, which are computationally infeasible for much larger sizes.

### 5.2.1 Analysis via CCR

We first apply the canonical correlations realization method to identify the linear functionals to be used for constructing a multiscale model for the steady-state process. In particular, we examine the linear functionals chosen by CCR to decorrelate the “northeast” quadrant,  $(1 - 8, 1 - 8)$ , from the remaining elements of a  $16 \times 16$  2-D diffusion process in steady state. Circular boundary conditions are used so that the process covariance is spatially stationary and symmetric. Figures 5-1 and 5-2 show the seven most significant linear functionals chosen by CCR.

We note several features of the linear functionals produced by CCR. First, the singular values associated with the linear functionals, shown in Figure 5-3 in decreasing order, exhibit a jump after the first 28 functionals and another after the next 20 linear functionals. The singular values associated with the most significant 28 functionals are on the order of 1, as seen in Figure 5-3(a), while those associated with the next 20 functionals are on the order of  $10^{-5}$ , as seen in panel (b). The singular values associated with the remaining functionals are on the order of round-off errors. That is, the first 28 linear functionals essentially capture all the conditional correlations, while the additional 20 capture what little that remains. Notice that the number of points along a border of width 1 of an  $8 \times 8$  field is 28, and is 48 along a border of width 2. Secondly, although the region of support for all the linear functionals is the entire  $(1 - 8, 1 - 8)$  quadrant, the values of the first 28 linear functionals are significant only along the boundary, i.e., the four edges. Seven of the most significant functionals are depicted in Figures 5-1 and 5-2. The values of the next 20 linear functionals are significant only along the inside rim of a border of width 2.

From Section 2.3.2 we know that the 28 boundary points of  $(1 - 8, 1 - 8)$  are sufficient to exactly decorrelate  $(1 - 8, 1 - 8)$  from the remainder of a  $16 \times 16$  first-

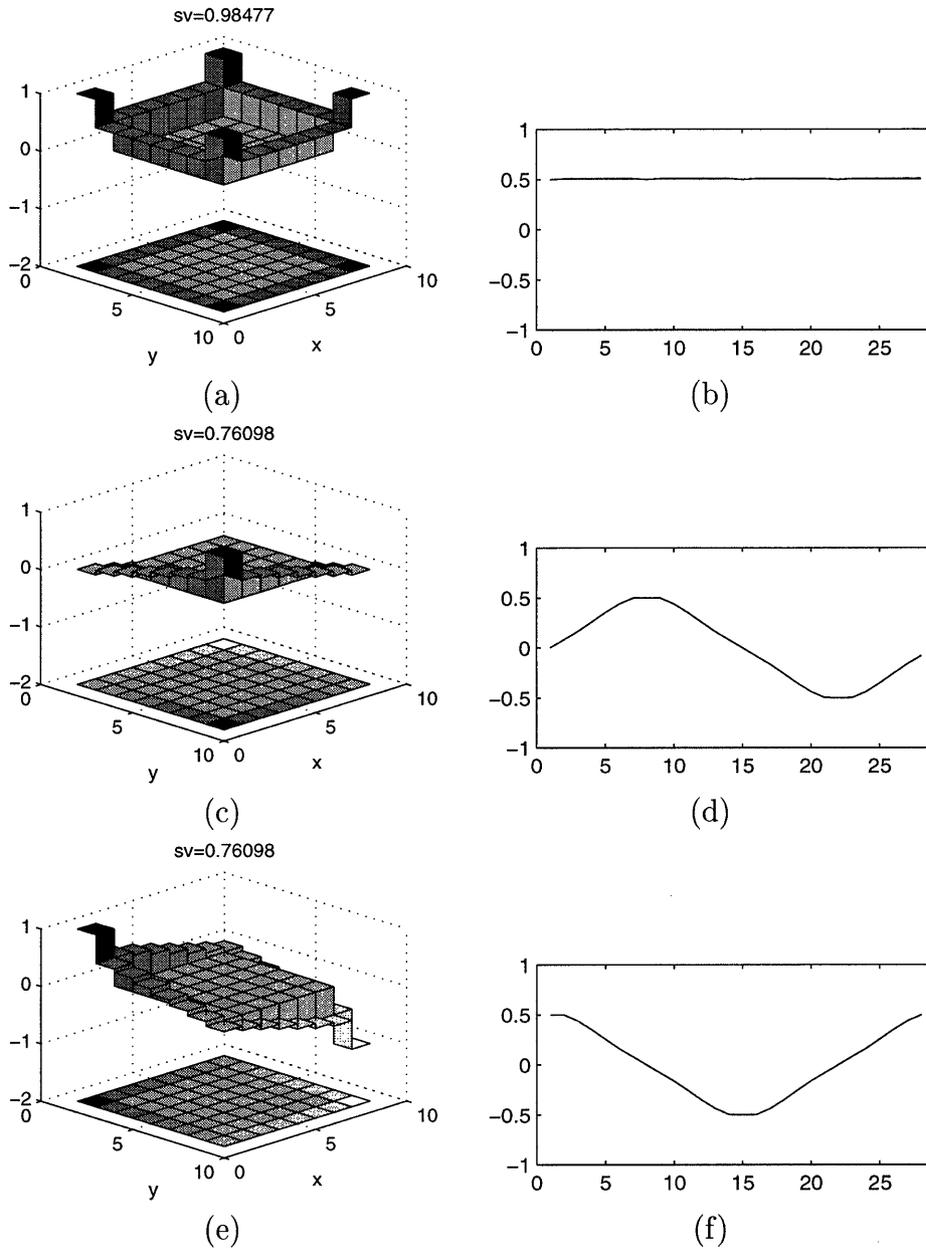
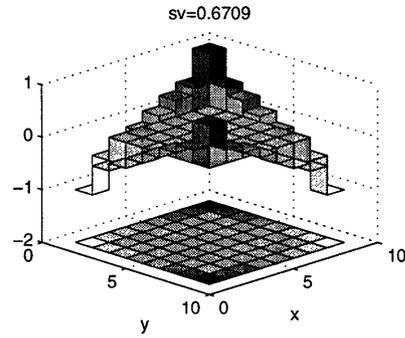
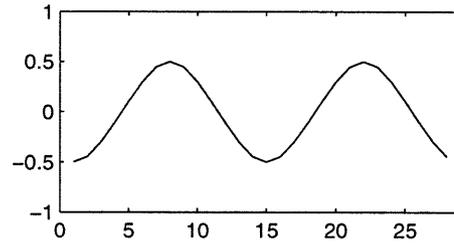


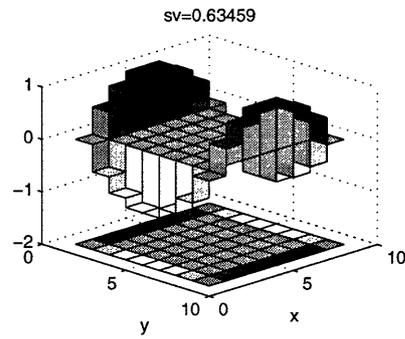
Figure 5-1: The three most significant linear functionals chosen by CCR to decorrelate elements in the “northeast” quadrant,  $(1-8, 1-8)$ , from the remaining elements of a  $16 \times 16$  steady-state 2-D diffusion process on a toroid. ( $\beta = 1$ ,  $\Delta\tau = 4 \times 10^{-5}$ .) The maximum magnitude of the linear functionals is normalized to 1. The panels on the left show the values of the linear functional over its region of support on  $(1-8, 1-8)$ . The singular value (sv) associated with each linear functional is displayed above the plots. White is  $-1$ ; black is 1. The panels on the right show plots of the values of the linear functionals along the boundary of  $(1-8, 1-8)$ , starting at  $(1, 1)$  and circling the quadrant counterclockwise. The values at the four corners, i.e., at  $(1, 1)$ ,  $(1, 8)$ ,  $(8, 8)$ , and  $(8, 1)$ , are plotted at half of their actual values to account for the corner effects.



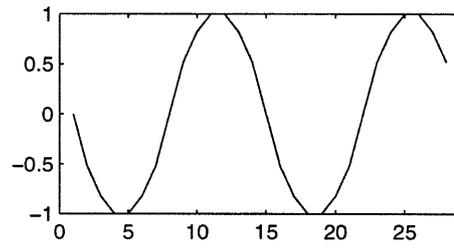
(a)



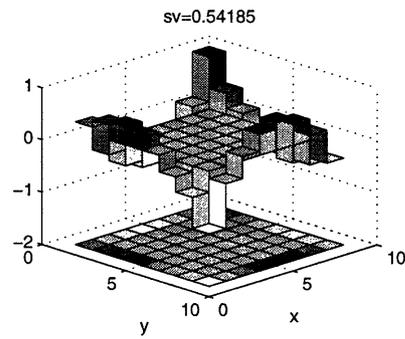
(b)



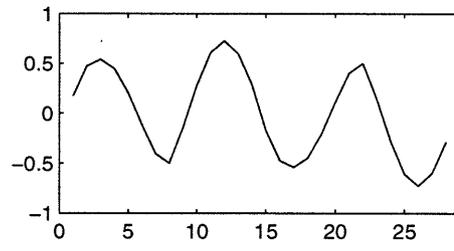
(c)



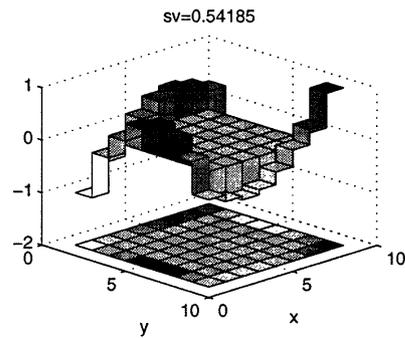
(d)



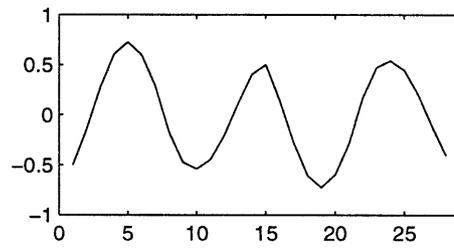
(e)



(f)



(g)



(h)

Figure 5-2: The fourth to the seventh most significant linear functionals chosen by CCR to decorrelate elements in the “northeast” quadrant,  $(1 - 8, 1 - 8)$ , from the remaining elements of a  $16 \times 16$  steady-state 2-D diffusion process on a toroid. The plots are similar to those in Figure 5-1.

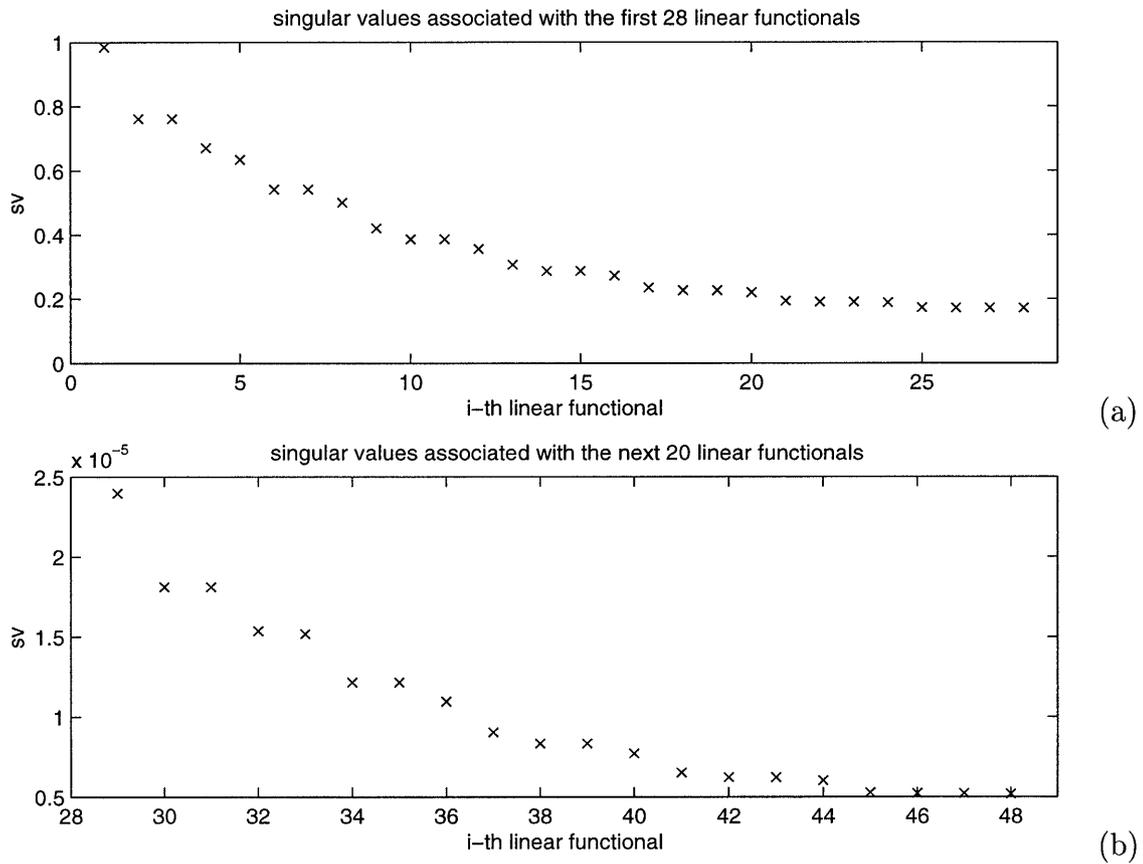


Figure 5-3: Singular values associated with (a) the 28 most significant linear functionals and (b) the next 20 linear functionals produced by CCR that maximally decorrelate elements  $(1 - 8, 1 - 8)$  from the remaining elements of a  $16 \times 16$  steady-state 2-D diffusion process on a toroid. ( $\beta = 1$ ,  $\Delta\tau = 4 \times 10^{-5}$ .)

order MRF. To exactly decorrelate  $(1 - 8, 1 - 8)$  of a second-order MRF, boundary points along the border of  $(1 - 8, 1 - 8)$  of width 2 are sufficient, i.e., requiring another 20 functionals comprised of the inside rim of the border. The structure of  $\mathbf{P}_z^{-1}$  tells us that the steady-state diffusion process on a toroid is essentially a first-order MRF, but is better approximated as a second-order MRF mostly due to the discretization effects. It is then not surprising that the linear functionals chosen by CCR are equivalent to the functionals chosen for modeling MRFs. The set of the 28 most significant linear functionals and the set of the next 20 most significant functionals selected by CCR are equivalent to, i.e., new bases for, the set of points on the outside and the inside rim of a width-2 border of  $(1 - 8, 1 - 8)$ , respectively.

Thirdly, and more interestingly, are the functional shapes of the CCR linear functionals. Focusing on the distribution of the values along the boundary points of the linear functionals shown in Figures 5-1 and 5-2, we see that the functionals look almost like the Fourier basis functions, except at the corner points at  $(1, 1)$ ,  $(1, 8)$ ,  $(8, 8)$ , and  $(8, 1)$ , where the values seem to be doubled. The functional in Figure 5-1(a) is almost a constant function; in Figures 5-1(c) and (e), we have two sines of period equal to the total length of the boundary and out of phase; and in Figures 5-2(a) and (c), two sines of period equal to half the length of the boundary and out of phase, etc. The panels in the right column show plots of the values of the linear functionals along the boundary of  $(1 - 8, 1 - 8)$ , starting at  $(1, 1)$  and circling the quadrant counterclockwise. The values at the four corners are plotted at half of their actual values to account for the corner effects. Given the symmetry of the setup, the 1-D process along the edges of a square region is spatially stationary except at the corners and is thus approximately diagonalizable with a Fourier decomposition.

With different boundary conditions, for instance Dirichlet boundary conditions, or type 3 homogeneous boundary conditions [54], the steady-state diffusion process becomes spatially nonstationary, although it still appears to be approximately Markov. The number of linear functionals needed for decorrelating a subregion on the edge of the 2-D field is now less than the full boundary point set [49] because of the non-circular boundary conditions. For instance, with type 3 boundary conditions

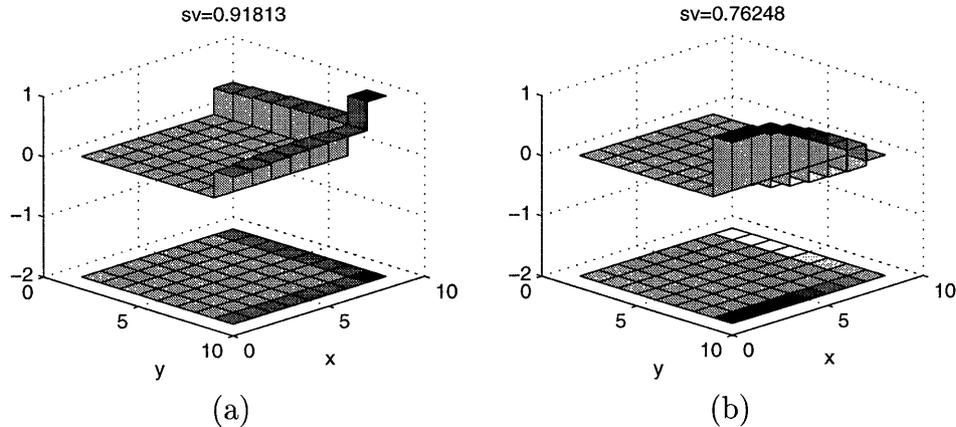


Figure 5-4: The two most significant linear functionals chosen by CCR to decorrelate elements in the “northeast” quadrant,  $(1 - 8, 1 - 8)$ , from the remaining elements of a  $16 \times 16$  steady-state 2-D diffusion process on a sheet ( $\beta = 1$ , type 3 homogeneous boundary conditions,  $\Delta\tau = 4 \times 10^{-5}$ ). The maximum magnitude of the linear functionals is normalized to 1. White is  $-1$ ; black is 1.

corresponding to a square sheet rather than a toroid, decorrelating  $(1 - 8, 1 - 8)$  from the rest of the  $16 \times 16$  field requires only 15 functionals, equal to the number of boundary points of the  $(1 - 8, 1 - 8)$  quadrant that do not fall on the edges of the  $16 \times 16$  field. As an illustration, Figure 5-4 depicts two most significant linear functionals produced by CCR for the case of a cooling sheet.

### 5.3 Multiscale Realizations of the Exact Steady-State Prediction Error

The exact steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$  solved by the Riccati equation (3.35) is generally not only nonstationary but also non-Markov. Similar to the 1-D case, we claim that the linear functionals and the multiscale models designed to exactly model MRFs can also adequately model the non-Markov steady-state prediction errors. To validate this proposition, we test several types of multiscale models and examine their performance. The purpose here is to determine what linear functionals and multiscale models are suitable for capturing the steady-state prediction error statistics. In the next section, we will examine whether these models are suitable

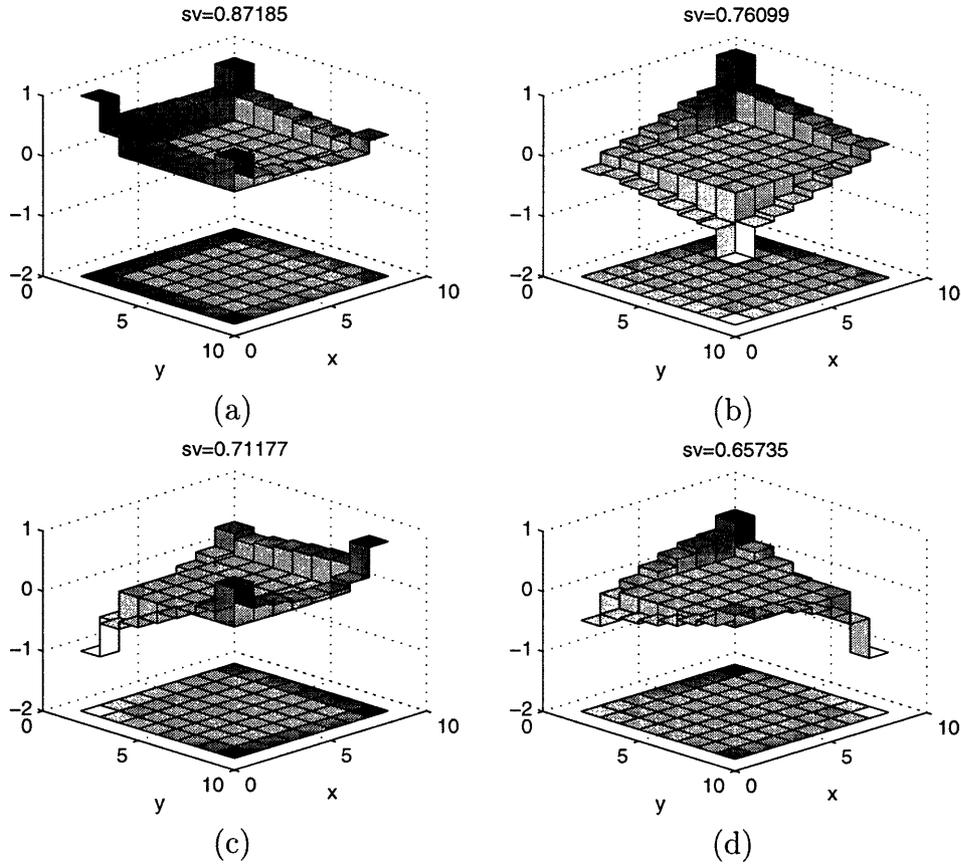


Figure 5-5: The four most significant linear functionals chosen by CCR to decorrelate elements  $(1-8, 1-8)$  from the remaining elements of the  $16 \times 16$  steady-state prediction error field of a 2-D diffusion process on a toroid with a single point measurement at  $(6, 7)$ . ( $\beta = 1$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 4 \times 10^{-5}$ .) The maximum magnitude of the linear functionals is normalized to 1. White is  $-1$ ; black is 1.

for the multiscale recursive algorithm.

### 5.3.1 Internal Models with Linear Functionals by CCR

The linear functionals generated by CCR for an exact multiscale model of the steady-state prediction error as compared to those for the steady-state process exhibit several features similar to what was observed in the 1-D case in Section 4.2.2. Take the example of estimating the  $16 \times 16$  diffusion process with a single measurement at  $(6, 7)$ , whose four most significant functionals are depicted in Figure 5-5.

- (i) The first 28 most significant functionals still dominate. However, the drop in the singular values associated with the functionals after the first 28 is less drastic.

The singular values associated with the next 20 most significant functionals are on the order of  $10^{-2}$ . This means that the residual correlation not captured by the first 28 functionals is larger now for modeling the error field than before for the process.

- (ii) The values of the 28 most significant functionals are still concentrated on the boundary. Their values at the interior points, although still tiny, are larger than those functionals for the steady-state process.
- (iii) The spatial nonstationarity of the error field results in a distribution of the values of the linear functionals that appear less like sinusoids along the boundary. On the other hand, when the functionals are arranged in decreasing order of their associated singular values, as depicted in Figure 5-5, the general feature of increasing spatial frequencies of the functionals along the boundary is still evident.

### Full-Order Internal Models with CCR Functionals

The results from CCR show that for the single measurement example<sup>1</sup> essentially all the conditional correlations between any  $n_x \times n_y$  subregion and the remainder of the 2-D steady-state diffusion process or the error field are captured by the  $2(n_x + n_y) - 4$  most significant linear functionals generated by CCR, which equals the number of boundary elements of the  $n_x \times n_y$  block. In addition, the region of support for these functionals concentrate predominantly on the boundary. The actual state dimension at node  $s$  on an internal model with CCR generated functionals for an  $n \times n$  2-D field is  $k(s) = 8 \times (n/2^{m(s)}) - 16$ , where  $m(s)$  is the scale of node  $s$ . For example, the state dimension at the root node of an internal model for  $16 \times 16$  field at the finest scale is  $4 \times 28$  accounting for the four sets of linear functionals that decorrelate each of the four quadrants. We refer to such internal models with state dimensions equal

---

<sup>1</sup>Since the system is only weakly observable, the single measurement case is, in a sense, a severe test. We use the single measurement example throughout this chapter to illustrate the development of our multiscale methods. We do, however, show examples with other system setups in the results section in Section 5.5.

to the number of boundary points as *full-order* models<sup>2</sup>.

As we have said in our discussion in Section 2.3.2 about exact multiscale models for 2-D MRFs, which are full-order models, if the state dimension grows with the number of variables,  $N = n^2$ , of the  $n \times n$  field of interest, the computational complexity of the multiscale smoothing algorithm is  $\mathcal{O}(N^{3/2})$ . Reduced-order models may aim to cap the state dimension to a constant independent of  $s$  or  $N$ , in which case the computational complexity becomes the far more desirable  $\mathcal{O}(N)$ . However, reduction in state dimension is almost invariably accompanied by a decrease in model fidelity and consequently estimator performance. The trade-off depends on whether a few of the linear functionals capture most of the conditional correlations. Capping the state dimensions to a constant is an especially stringent requirement for modeling 2-D random fields. It is reasonable to expect that at coarser scales of the tree model, more linear functionals must be kept to capture the conditional correlations over a large spatial area. The relationship between state dimension and the size of the 2-D field depends on the structure in the correlations. For exact models of 2-D MRFs, the state dimension, i.e., the number of functionals, is proportional to  $\sqrt{N}$ . It was asserted in [67] that with additional structure in the correlations of certain textures, the state dimension may be independent of  $N$ . However, the computational complexity properties we have spoken so far are asymptotic properties. For practical purposes on images of possibly large but fixed size, it may be good enough to reduce the state dimension, for instance by subsampling of the boundary points [73] to a fixed percent of the number of boundary points. The resulting multiscale model and smoothing algorithm are more efficient than conventional estimation methods, even though the asymptotic complexity remains the same as the full-order model at  $\mathcal{O}(N^{3/2})$ . From a computational point of view, a desirable compromise may be a slow growth of state dimension, e.g.,  $\log N$ , while capturing most of the conditional correlations.

---

<sup>2</sup>Full-order models, without additional qualification, are simply models whose state dimension is equal to the number of boundary points. Full-order models do not necessarily imply exact models.

## Reduced-Order Internal Models with CCR Functionals

Linear functionals produced by CCR are naturally ordered according to the singular values associated with the functionals. However, the decrease in the singular values for the 2-D diffusion process is slow and gradual as seen in Figure 5-3(a) before the jump at the full-order size, which equals the number of boundary points. There is not an obvious cut-off point where a reduced-order model is particularly suitable. We expect a trade-off between lower state dimension, thus computational complexity, and performance. However, it is not immediately clear how low we may limit the state dimensions and still obtain reasonable performance figures. As an example, we show some results for the simple case of one point measurement at (6, 7).

The purpose here is to examine whether the linear functionals produced by CCR and the resulting full- and reduced-order internal models can adequately model the steady-state prediction errors of 2-D diffusion for the purpose of estimation. Performance is measured via the fractional variance reduction (FVR) of the suboptimal multiscale estimators using the internal models. We first solve for the exact steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$  via (3.35). We then apply CCR to find the linear functionals and keep a subset of the most significant functionals on the multiscale model for the prediction error. The steady-state error covariance of the resulting suboptimal multiscale estimator is found via the Lyapunov equation of (3.36). The FVR of the optimal steady-state estimator for the case of a single point measurement at (6, 7) is shown in Figure 5-6(a). In panels (b)–(d), we compare the steady-state performance of the suboptimal multiscale estimators using a full-order model and two reduced-order models. Note that the full-order model simply keeps the same number of functionals as the number of boundary points. It is not exact, but the multiscale estimator using the full-order model is nearly optimal, as seen in Figure 5-6(b). The degradation in performance with respect to the optimal estimator of two reduced-order estimators are shown in panels (c) and (d). The state dimensions of the full-order model are 112, 48, 16, 4, 1 from coarse to fine at scales 0 – 4. The reduced-order model in panel (c) keeps half as many functionals as the full-order model (except

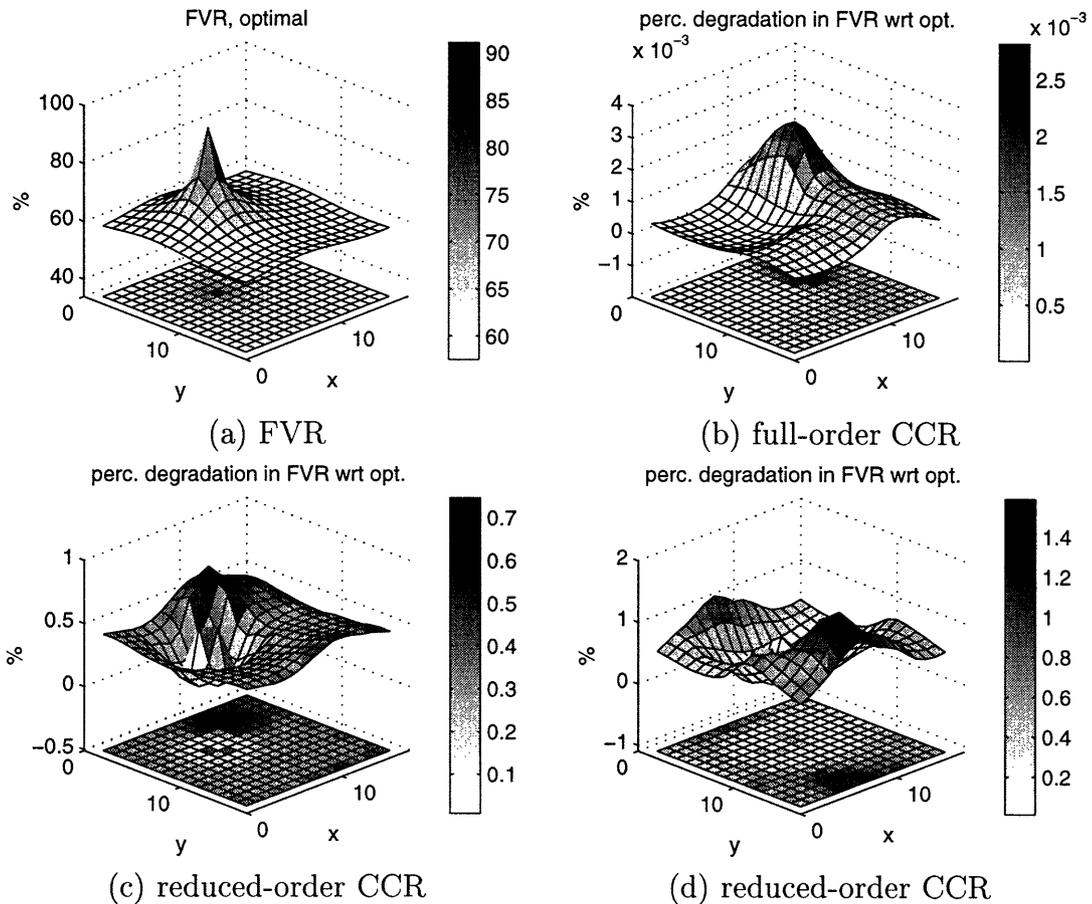


Figure 5-6: (a) The FVR of the optimal steady-state estimator for a  $16 \times 16$  2-D diffusion process on a toroid with a single point measurement at  $(6, 7)$ . FVRs of the optimal steady-state estimator for single point measurements at other locations are simply shifted versions of (a), since the performance of the optimal estimator is shift invariant. The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators with linear functionals selected by CCR are shown in (b) for a full-order model with state dimensions equal to 112, 48, 16, 4, 1 at scales 0 – 4, in (c) for a reduced-order model with state dimensions 56, 24, 8, 4, 1, and in (d) for a reduced-order model with state dimension 12, 8, 4, 4, 1. ( $\beta = 1$ ,  $\text{SNR} = 0$  dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

at scales near the finest). The model in panel (d) keeps roughly 10% as many; its state dimensions are 12, 8, 4, 4, 1. That is, only three functionals, instead of 28, are used to decorrelate  $(1 - 8, 1 - 8)$  from the rest of the  $16 \times 16$  field at the root node, for example. In relative terms, the reduced-order estimator performed two to three orders of magnitude worse than the full-order estimator. However, in absolute terms, the reduced-order estimator, even at severely restricted state dimensions, performed at most a few percent worse in FVR than the optimal estimator.

The results here suggest that in principle a small number of well chosen linear functionals, far fewer than the number of boundary points, are enough to capture most of the conditional correlations of the steady-state error field of 2-D diffusion, such that the resulting approximate multiscale model and estimator perform more than adequately. However, computational complexity of CCR and of solving the Riccati equation prevents us from blindly applying this approach. Thus, finding CCR functionals for large problems is prohibitively complex. Moreover, with an eye towards the multiscale recursive algorithm as we shall see shortly in Section 5.4.1, we note that the multiscale prediction step is computationally costly with the CCR functionals even if we could compute them.

### 5.3.2 Internal Models with Boundary-Point Functionals

#### Full-Order Internal Models with Boundary-Point Functionals

We saw that the values of the linear functionals produced by CCR for decorrelating a subregion of either the steady-state diffusion process or the estimation error field from the rest of the 2-D field are small except at the boundary points of the subregion. We know that the full set of boundary points and any full-order set of non-local functionals whose region of support is exactly the boundary point set are equivalent via a basis transform. Therefore, we would expect a full-order internal model with boundary-point linear functionals, an example of which is shown in Figure 2-6, to be almost the same as a full-order model with CCR linear functionals whose region of support are nearly the boundary point set. Since the CCR functionals do have

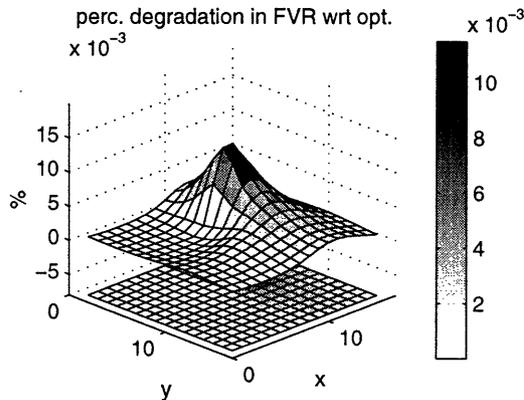


Figure 5-7: The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using a full-order internal model with dense boundary-point linear functionals, whose with state dimensions equal to 112, 48, 16, 4, 1 at scales 0 – 4. ( $\beta = 1$ , circular boundary conditions, one point measurement at (6, 7),  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 4 \times 10^{-5}$ .)

nonzero values at element other than the boundary set, they are expected to result in a slightly more accurate model and better estimator. Figure 5-7 shows the degradation in FVR of the suboptimal multiscale estimator using a full-order model with boundary-point functionals. Compare Figure 5-7 with Figure 5-6(b), and we see that the suboptimal estimator using full-order model with boundary point functionals performed somewhat poorer than the estimator with CCR linear functionals, although both are nearly the same as the optimal estimator.

### Reduced-Order Internal Models with Boundary-Point Functionals

Constructing reduced-order models using CCR selected linear functionals is relatively straightforward, since the CCR algorithm orders the functionals. It is less clear which boundary points should be chosen and which discarded for reduced-order models with boundary-point linear functionals, since it is not clear which boundary points are more important. One method is to subsample the boundary points, as is done in [73, 31] for modeling ocean hydrography. The method is based on the simple premise that neighboring points in the random field are usually highly correlated and that the subsampled points capture the essential correlation information. We surmise that points at the corners of subregions and around areas with more detailed

features in the correlation structure of the underlying field, such as the area around a measurement, are more important. If we have a fixed budget of state dimension, then boundary points falling in the more important areas should be more densely sampled while those in relatively featureless areas may be more sparsely sampled. Aside from the freedom to choose what subset of boundary points should be kept as functionals, we also have the freedom to choose the state dimensions. For problems of the size we are considering here most of the order reduction occurs at the root and scale 1. Nodes at the finer scales keep the full-order set of functionals. The real computational advantage is to be seen for much larger problems.

In Figures 5-8 and 5-9 we show two examples of reduced-order models with subsampled boundary-point functionals. Our purpose here is to see whether models using subsampled boundary-point functionals yield adequate approximate estimators. In Figure 5-8(a) we show the performance of a suboptimal multiscale estimator using a reduced-order model with subsampled boundary-point linear functionals. The functionals kept at the root node of the quad tree and those kept at one of the nodes at scale 1 are depicted in panels (c) and (d), respectively. Boundary-point functionals kept at the other three nodes at scale 1 have the same pattern as those shown in (d) but are located in the other three quadrants of the 2-D field. Dense boundary points are kept at the finer scales. The state dimensions are 48, 48, 16, 4, 1 at scales 0 – 4. To make the comparison with estimators using CCR generated functionals more meaningful, the performance of a reduced-order model with CCR functionals of the same state dimensions are shown in panel (b). A lower dimensional example with state dimension 16, 16, 16, 4, 1 is depicted in Figure 5-9. Even in the latter example, the reduced-order estimator performed no more than 2% worse than the optimal estimator. Not surprisingly, the reduced-order estimators with subsampled boundary points performed worse than those with CCR linear functionals. Given a fixed state dimension, the CCR algorithm generates an optimal set of functionals in the sense that they maximally decorrelate the subregions under consideration. No other set of functionals does a better job at the decorrelation. Thus, it is reasonable to expect the resulting multiscale model based on the CCR functionals to be the most accurate and

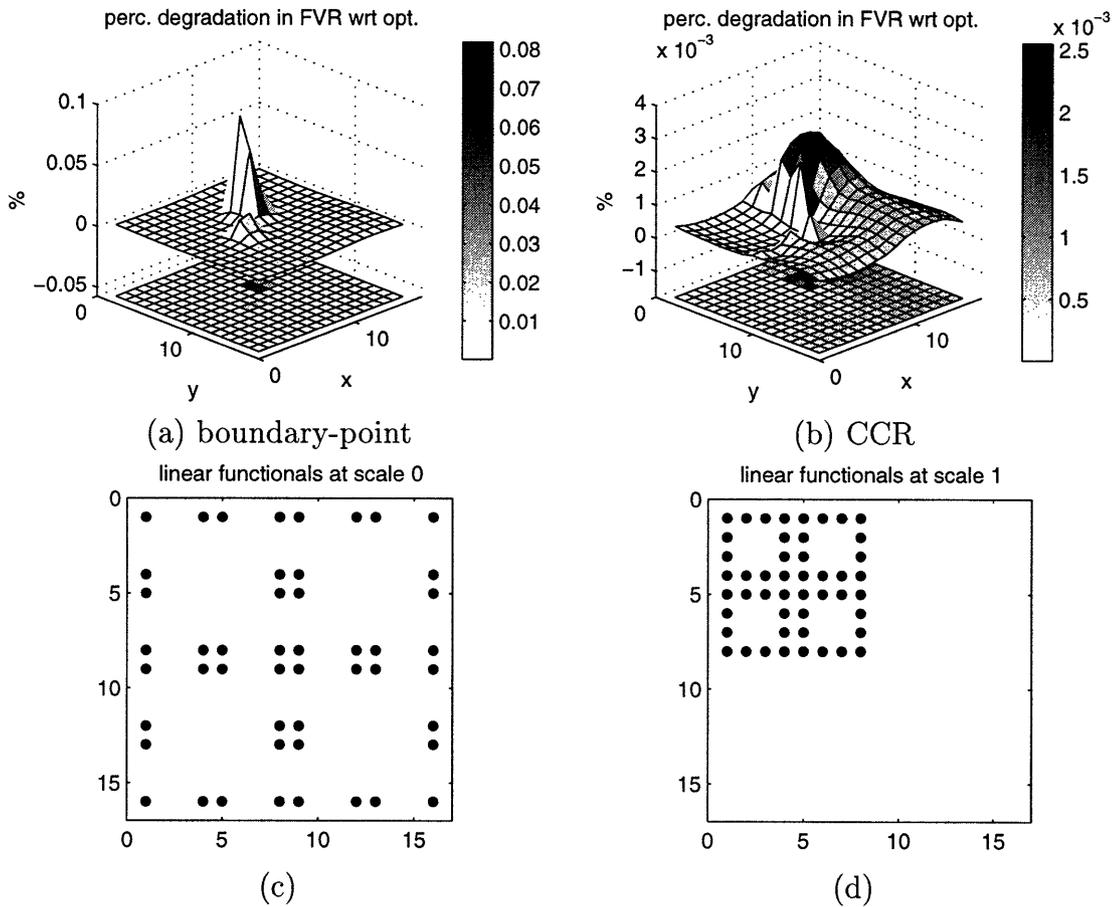


Figure 5-8: The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using (a) a reduced-order internal model with subsampled boundary-point linear functionals and (b) a reduced-order internal model with CCR linear functionals of the same state dimensions. The functionals kept at the root node and one of the nodes at scale 1 of the model for (a) are shown in (c) and (d). The state dimensions are 48, 48, 16, 4, 1 at scales 0 – 4. ( $\beta = 1$ , circular boundary conditions, one point measurement at (6, 7), SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

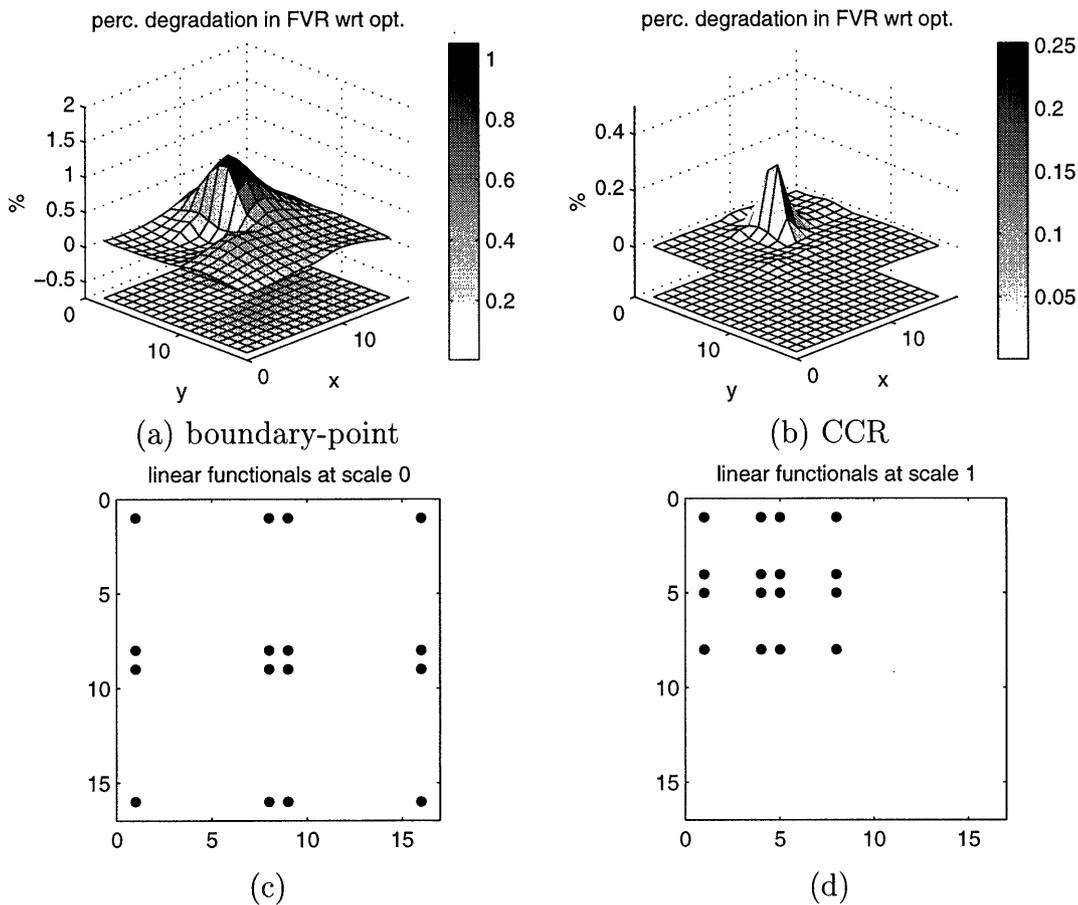


Figure 5-9: The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using (a) a reduced-order model with subsampled boundary-point linear functionals and (b) a reduced-order model with CCR linear functionals of the same state dimensions. The functionals kept at the root node and one of the nodes at scale 1 of the model for (a) are shown in (c) and (d). The state dimensions are 16, 16, 16, 4, 1 at scales 0 – 4. ( $\beta = 1$ , circular boundary conditions, one point measurement at (6, 7),  $\text{SNR} = 0$  dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

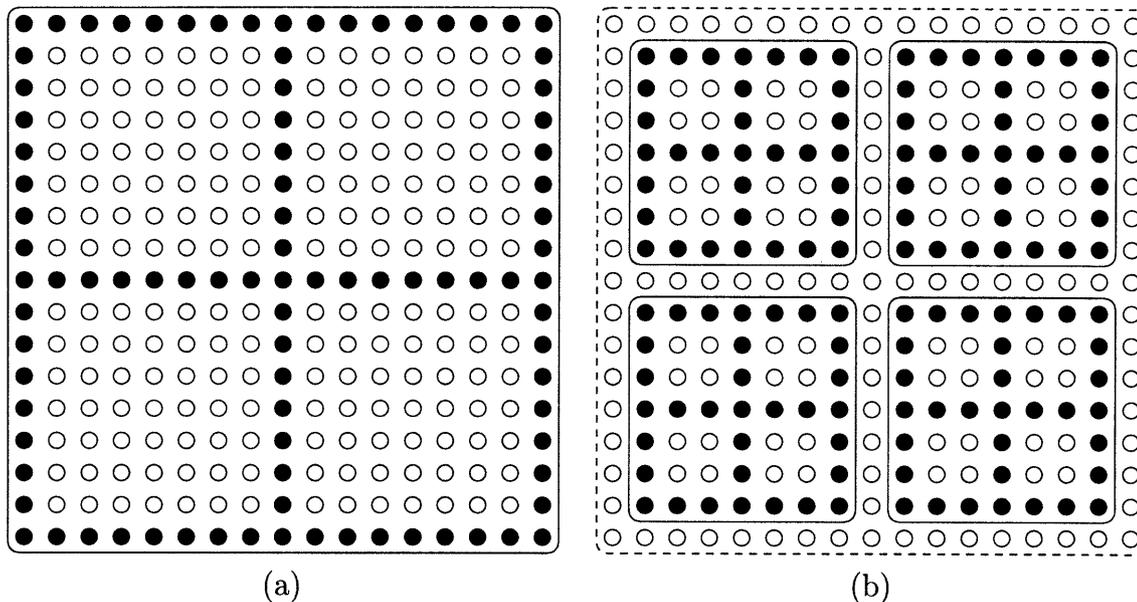


Figure 5-10: The linear functionals of a non-redundant model for a first-order 2-D Markov random field. The elements marked with filled circles are the states kept at (a) the root node and (b) the four nodes at scale 1 of a  $17 \times 17$  2-D MRF.

in turn the resulting estimator to perform the best. Of course, the construction of models using CCR functionals require more computation since the CCR functionals must be computed via SVDs.

### 5.3.3 Non-Redundant Models

#### Full-Order Non-Redundant Models

In Section 3.5 we introduced the class of non-redundant models for 1-D random processes primarily for modeling 1-D MRFs. Exact non-redundant models for 2-D MRFs are directly analogous to the 1-D models. An example for a  $17 \times 17$  field is shown in Figure 5-10. The linear functionals kept at the root node consist of the edges and the midlines of the 2-D field, instead of the end- and midpoints in the 1-D case. Functionals at the first-scale nodes similarly keep the edges and the midlines of the four subregions under the root node. Since dense boundary and midline points are kept as functionals, we will refer to a model with functionals shown in Figure 5-10

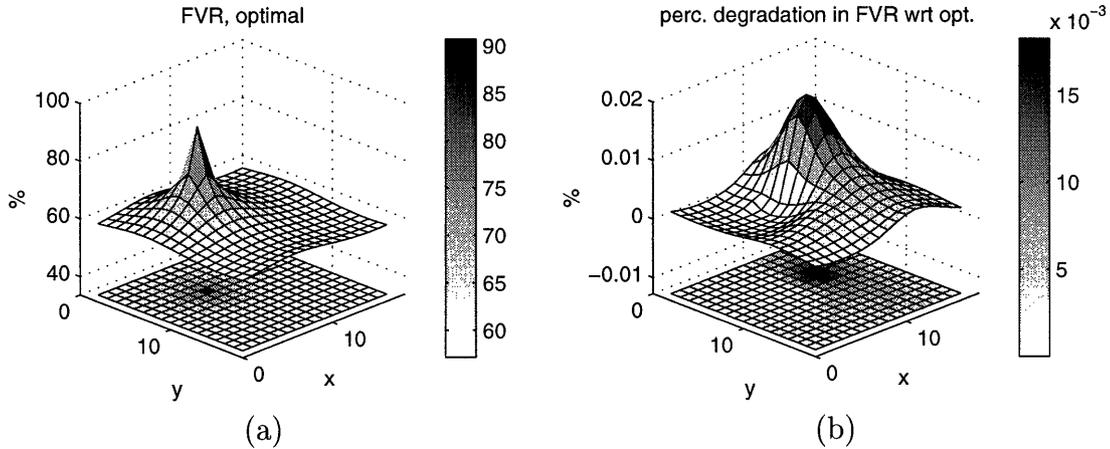


Figure 5-11: (a) The FVR of the optimal steady-state estimator for a  $17 \times 17$  2-D diffusion process on a toroid with a single point measurement at  $(6, 7)$ . (b) The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimator using a full-order non-redundant model with state dimensions 93, 33, 4 at scales 0 – 2. ( $\beta = 1$ ,  $\text{SNR} = 0$  dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

as a full-order non-redundant model<sup>3</sup>. Unlike the functionals of the boundary-point internal models (c.f., Figure 2-6), the point functionals kept on non-redundant models are not repeated at multiple nodes.

For convenience, we choose a 2-D field of size  $17 \times 17$ , so that the distribution of the elements of the field of interest in the non-redundant model is symmetric. Similar to 1-D models in Section 3.5.1, a 2-D field of any size is just as easily placed on the non-redundant model, or for that matter, is just as easily placed at the finest scale of an internal model. The two different sizes for internal model examples and for non-redundant models examples are close enough for comparisons between them to be meaningful. In Figure 5-11(a), we show FVR of the optimal steady-state estimator for a  $17 \times 17$  2-D diffusion process on a toroid with a single point measurement at  $(6, 7)$ , which resembles the  $16 \times 16$  result shown in Figure 5-6(a). The largest variance reduction occurs near the measurement. The percent degradation in FVR of the suboptimal steady-state multiscale estimator using a full-order non-redundant model is shown in Figure 5-11(b). The performance of the estimator using a non-redundant model is nearly optimal and compares favorably with estimators using internal models

<sup>3</sup>Full-order simply means that the state dimension is equal to the number of boundary points. It does not necessarily imply optimality.

with either CCR linear functionals or boundary-point linear functionals as depicted in Figures 5-6(b) and 5-7.

### **Reduced-Order Non-Redundant Models**

A critical difference between 2-D non-redundant models and internal models arises when we construct reduced-order models by keeping subsampled boundary points as state variables rather than the full dense boundary point set. For reduced-order internal models, the reduced-order state at node  $s$ , simply no longer decorrelates the subtrees connected to  $s$  as effectively as a higher-order state. Whether coarse-scale linear functionals are local or non-local, the field of interest is mapped to the finest scale on internal models. Therefore, the estimates and error variances for individual elements of the field of interest and the point measurements of the field are always available at the finest scale. If the coarse-scale functionals are non-local, then estimates and measurements of these non-local quantities are also available, and the state variables at all nodes on the tree form an overcomplete basis for the field of interest.

On the other hand, reduced-order non-redundant models face, in addition to less effective conditional decorrelation, a problem with representing all elements of the field of interest on the tree. Because of the non-redundancy property, subsampling the boundary points in non-redundant models means that some elements of the field of interest simply do not appear in the tree model anymore. The state variables in non-redundant models forms a complete basis only in the full-order version. The number of state variables in a reduced-order model is less than the number of elements in the field of interest. Therefore, if applied directly, we encounter two potential problems:

- (i) no estimates or error statistics for these elements are directly available, and
- (ii) point measurements of these elements can not be placed on the tree (although this problem can be solved by simply adding such elements onto the tree).

The issues at hand with representing the random field of interest in reduced-order non-redundant models are similar in spirit to the last two requirements we laid out

in Section 3.3.3 for the choice of linear functionals, i.e., the ability of the functionals to represent the quantities for which we want estimates and quantities for which we have measurements. In the following subsections, we propose several ways of dealing with elements of the field of interest that are not modeled on the tree.

### Reduced-Order Non-Redundant Models with Deterministic Interpolation

Denote the field of interest by  $\chi$ . Denote the subset of  $\chi$  that is modeled in the reduced-order model by  $\xi$ , i.e.,  $\xi$  is a vector containing all the state variables in the reduced-order model. We may write  $\xi = \mathbf{M}\chi$  and use a deterministic interpolation scheme to compute the entire  $\chi$  from  $\xi$ :  $\chi = \mathbf{M}^\# \xi$ , where  $\mathbf{M}^\#$  is the interpolation matrix and also a pseudoinverse of  $\mathbf{M}$ . This interpolation can be done node by node. Let  $\zeta(s)$  denote the set of boundary points that would normally be the state at node  $s$  in a full-order model. The state  $\mathbf{x}(s)$  in the reduced-order model is then related to  $\zeta(s)$  by  $\mathbf{x}(s) = \mathbf{M}(s)\zeta(s)$ , where  $\mathbf{M}(s)$  is a selection matrix. At each node we may write the interpolation equation

$$\zeta(s) = \mathbf{M}^\#(s)\mathbf{x}(s). \quad (5.3)$$

A simple interpolation scheme that results in very sparse  $\mathbf{M}^\#(s)$  is a one-dimensional linear interpolation method. A boundary point that is not represented in the reduced-order model is simply interpolated from the two nearest boundary points that are represented.

Denote the estimate of  $\mathbf{x}(s)$  by  $\hat{\mathbf{x}}(s)$  and its estimation error covariance by  $\tilde{\mathbf{P}}(s)$ . Then, the estimate of  $\zeta(s)$  is  $\hat{\zeta}(s) = \mathbf{M}^\#(s)\hat{\mathbf{x}}(s)$ , and the covariance of the estimation error,  $\zeta(s) - \hat{\zeta}(s)$ , is  $\tilde{\mathbf{P}}_\zeta(s) = \mathbf{M}^\#(s)\tilde{\mathbf{P}}(s)\left(\mathbf{M}^\#(s)\right)^T$ . Of course, the sparser  $\mathbf{x}(s)$  is with respect to  $\zeta(s)$ , the less accurate the estimator using such a model is, and the more pronounced the artifacts are in the resulting interpolated estimates for the field of interest.

Highly subsampled reduced-order models also make placing point measurements more difficult. If a point measurement of  $\chi$  is not modeled in the tree, we may simply

add that element to the state at the node where it would normally be placed on a full-order model. This is adequate as long as we have only a small number of point measurements that need to be augmented, since each augmentation increases the state dimension by 1. However, if the number is large, for example in the case of dense measurements, adding state variables to accommodate all measurements reverts the augmented model to a full-order model. This is similar to the state augmentation problem studied in [25], where the added state dimension roughly equals the number of measurements to be augmented. Alternatively, we may model the point measurements not modeled on the tree using the interpolation equation of (5.3),

$$\mathbf{y}(s) = \mathbf{C}(s)\boldsymbol{\zeta}(s) + \mathbf{v}(s) = \mathbf{C}(s)\mathbf{M}^\#(s)\mathbf{x}(s) + \mathbf{v}(s). \quad (5.4)$$

We found that for 2-D diffusion, the simple linear interpolation scheme works well in many cases. More sophisticated deterministic interpolation scheme generally do not provide better results or less noticeable artifacts. Figure 5-12 shows the performance results and the linear functionals for estimating a  $17 \times 17$  2-D diffusion process on a toroid with a single point measurement at (6, 7). The suboptimal multiscale estimators are realized from the exact steady-state prediction error covariance  $\hat{\mathbf{P}}_p$  using non-redundant models with subsampled boundary-point linear functionals. We show two different sampling rates. As depicted in Figure 5-12(c) and (d) respectively, every fourth and every eighth boundary-point functional is kept at the root node. At scale 1, the size of the subregion is small enough such that we use the same sampling rate of 3 in both cases. At scale 2, full-order sets of functionals are kept. As shown in panels (a) and (b), the resulting suboptimal multiscale estimators perform no worse than within two percent of the optimal. The results here give us sufficient confidence to use reduced-order non-redundant models with linear interpolation for multiscale recursive estimation later in Section 5.4.

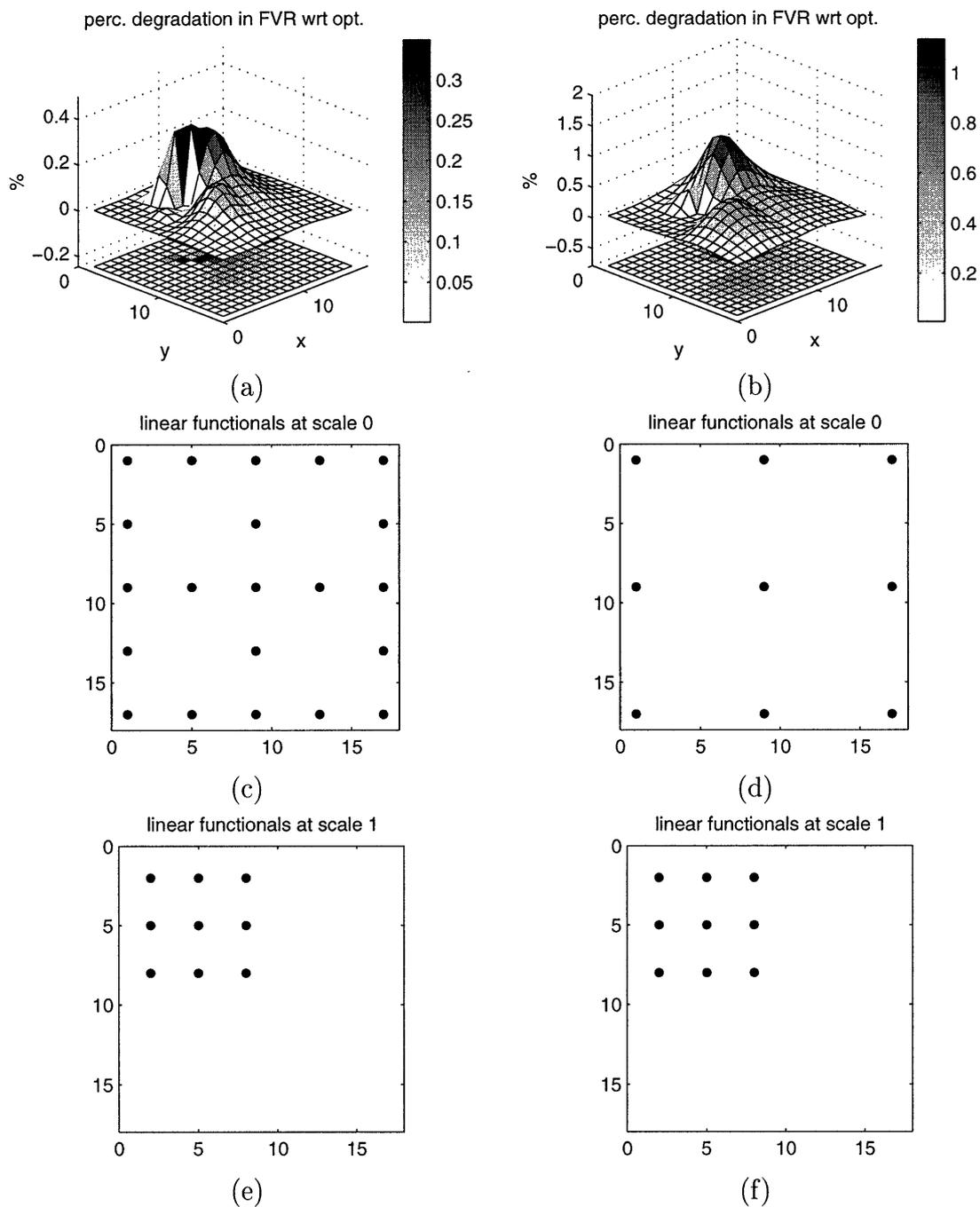


Figure 5-12: (a)–(b) The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using reduced-order non-redundant models with linear interpolation. The boundary-point linear functionals are sampled at a rate of 4 in (a) and 8 in (b) at the root node. The sampling rate at the scale 1 is 3, and at scale 2 is 1. The boundary-point functionals kept at the root node and one of the nodes at scale 1 for (a) are shown in (c) and (e); and for (b) in (d) and (f). ( $\beta = 1$ , circular boundary conditions, one point measurement at (6, 7), SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

## Reduced-Order Non-Redundant Models with Statistical Interpolation

Deducing  $\chi$  from  $\xi$  using linear interpolation ignores the statistical structure of  $\chi$ . We may treat  $\xi = \mathbf{M}\chi$  as a measurement equation and compute the LLSE of  $\chi$  given observation  $\xi$ . This is possible if we have the joint statistics between  $\xi$  and  $\chi$ , which is the case so far with the assumption that we have the full covariance of the field of interest. Of course, for large-size problems, computing the estimate of  $\chi$  globally is computationally infeasible, which is the reason for using multiscale methods in the first place. Done node by node, we may write  $\zeta(s)$  as its best estimate based on  $\mathbf{x}(s)$  plus the orthogonal error in this estimate. That is,

$$\zeta(s) = \mathbf{M}^\#(s)\mathbf{x}(s) + \mathbf{w}_\zeta(s\alpha_\zeta), \quad (5.5)$$

where  $\mathbf{M}^\#(s) = \mathbf{P}_{\zeta,x}(s)\mathbf{P}^{-1}(s)$  and the covariance of  $\mathbf{w}_\zeta(s\alpha_\zeta)$  is  $\mathbf{Q}_\zeta(s\alpha_\zeta) = \mathbf{P}_\zeta(s) - \mathbf{P}_{\zeta,x}(s)\mathbf{P}^{-1}(s)\mathbf{P}_{\zeta,x}^T(s)$ . Here  $\mathbf{P}_\zeta(s)$ ,  $\mathbf{P}(s)$ , and  $\mathbf{P}_{\zeta,x}(s)$ , respectively denote the covariance of  $\zeta(s)$ , the covariance of  $\mathbf{x}(s)$ , and the cross-covariance between the two. The particular values of  $\mathbf{P}_\zeta(s)$ ,  $\mathbf{P}(s)$ , and  $\mathbf{P}_{\zeta,x}(s)$  depend on the covariance of the field of interest  $\mathbf{P}_\chi$ .

Similar to deterministic interpolation, the estimate of  $\zeta(s)$  is  $\hat{\zeta}(s) = \mathbf{M}^\#(s)\hat{\mathbf{x}}(s)$ , but the error covariance is corrected with an additional noise covariance term:  $\tilde{\mathbf{P}}_\zeta(s) = \mathbf{M}^\#(s)\tilde{\mathbf{P}}(s)\left(\mathbf{M}^\#(s)\right)^T + \mathbf{Q}_\zeta(s\alpha_\zeta)$ . The cross-covariance between  $\hat{\zeta}(s)$  and  $\hat{\mathbf{x}}(s)$  is  $\mathbf{M}^\#(s)\tilde{\mathbf{P}}(s)$  since the noise  $\mathbf{w}_\zeta(s\alpha_\zeta)$  is uncorrelated with  $\mathbf{x}(s)$ .

Figure 5-13 shows the performance results of suboptimal multiscale estimators using statistical interpolation, for the same system in Figure 5-12. The interpolation parameters,  $\mathbf{M}^\#(\cdot)$  and  $\mathbf{Q}_\zeta(\cdot)$  are computed from the exact steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$ . The estimators in Figure 5-13(a) and (b) have the same state definitions as those in Figure 5-12(a) and (b), respectively. The suboptimal multiscale estimators with statistical interpolation perform comparably to, although somewhat poorer than, those with linear interpolation. We shall see later in Section 5.5 that for the iteratively implemented multiscale steady-state estimators, statistical interpolation scheme yields better estimator performance results.

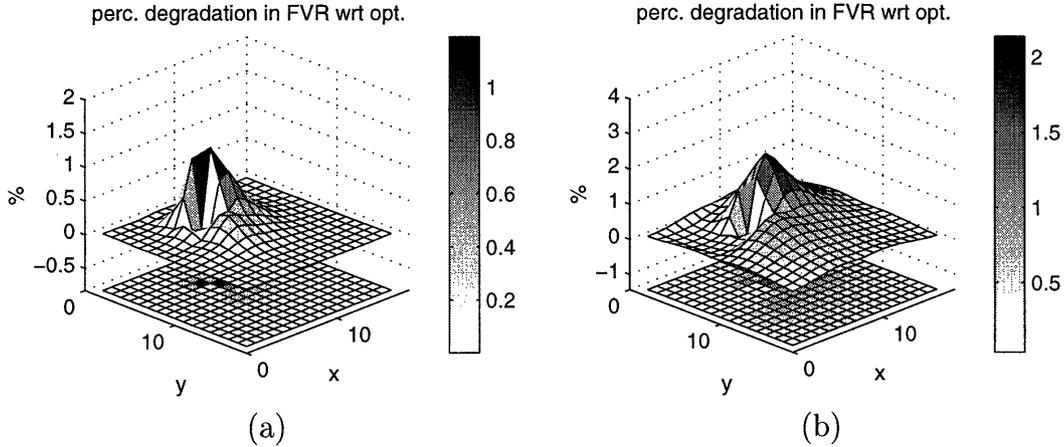


Figure 5-13: The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using reduced-order non-redundant models with statistical interpolation. The state definitions of the estimators in (a) and (b) are the same as those in Figure 5-12(a) and (b), respectively. ( $\beta = 1$ , circular boundary conditions, one point measurement at (6, 7), SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

### Non-Redundant Models with Offshoot Nodes

Equation (5.5) looks very much like the multiscale dynamic equation (2.6). Let  $\mathbf{x}_\zeta$  denote the elements of  $\zeta(s)$  that are not part of  $\mathbf{x}(s)$ , we may rewrite (5.5) as  $\mathbf{x}_\zeta = \mathbf{A}_\zeta(s\alpha_\zeta)\mathbf{x}(s) + \mathbf{w}_\zeta(s\alpha_\zeta)$ . We may create a new child node  $s\alpha_\zeta$  for  $s$  and put  $\mathbf{x}_\zeta$  there, i.e., let  $\mathbf{x}(s\alpha_\zeta) = \mathbf{x}_\zeta$ . The subscript  $\zeta$  indicates that these are quantities associated with the new *offshoot* nodes. For models with large subsampling rate, the dimension of  $\mathbf{x}(s\alpha_\zeta)$  may be much larger than  $\mathbf{x}(s)$ , in which case we may replace the statistical model represented by (5.5) with a multiscale model for the offshoot process.

Let us take the example of the root node of a  $17 \times 17$  2-D field, as shown in Figure 5-14. Suppose that nine subsampled boundary-point functionals are kept at the root node. These are labeled as solid black circles. On an ordinary reduced-order non-redundant model, the skipped boundary points, labeled as shaded gray circles, are not represented on the tree. We have said earlier that the steady-state process and prediction error of 2-D diffusion are approximately 2-D MRFs. This does not necessarily mean that rows or columns of the 2-D field are themselves 1-D MRFs. We found by examining the correlation structure of the field along single rows or columns

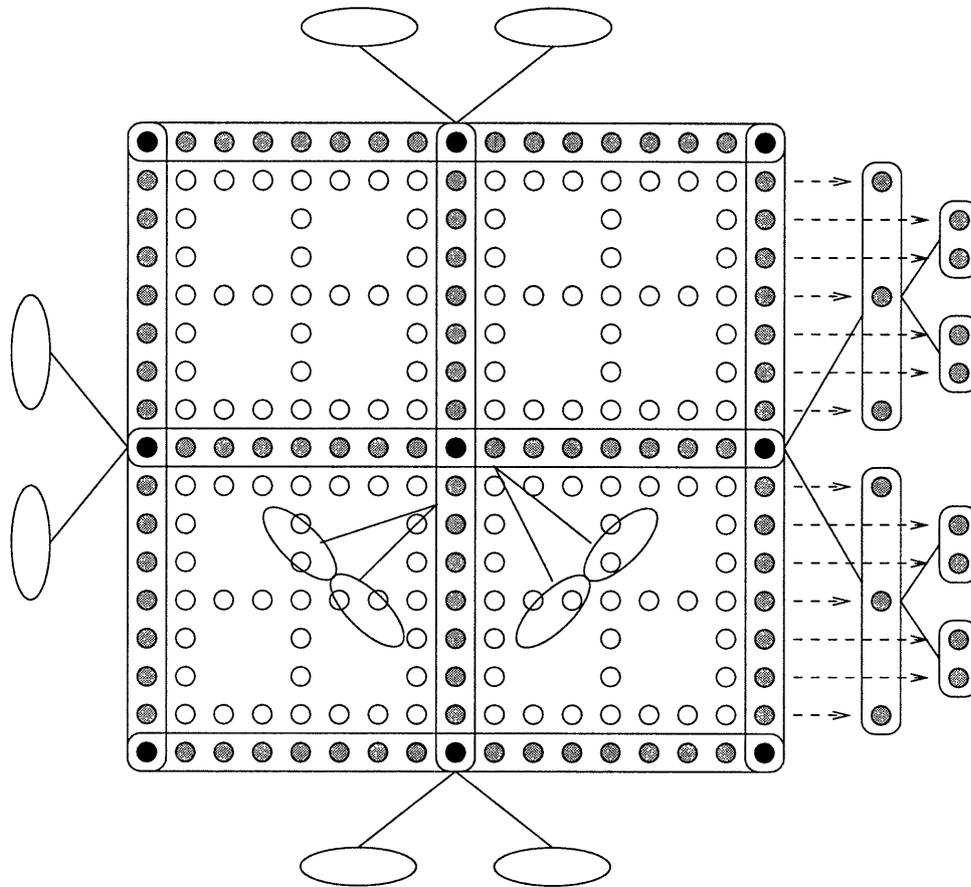


Figure 5-14: Linear functionals kept at the root node and at the offshoot nodes connected to the root of a non-redundant model for a  $17 \times 17$  field. The subsampled boundary-point functionals kept at the root node are labeled as solid black circles. The shaded gray circles indicate the boundary points that are to be kept on offshoot nodes. The offshoot tree modeling the right vertical edge is drawn in more detail. The arrowed dash lines indicate on which offshoot node the skipped boundary points are kept. Similar 1-D offshoot trees, illustrated with ovals, are grown for each of the other five vertical and horizontal edges consisting of the boundary point set. The region of support for the four regular child nodes of the root are marked with empty circles. Functionals kept at these child nodes may consist of all or a subsampled set of these points.

that they are approximately 1-D Markov, although the quality of the approximation is generally worse than those obtained in Chapter 4 for 1-D diffusion. Nevertheless, suppose that we assert that the rows and columns of the 2-D field can be modeled by 1-D non-redundant models. For example, take the right-most vertical edge which is shown in more detail in Figure 5-14. We assert that the top, middle, and bottom three boundary points, labeled as solid black circles, conditionally decorrelate the two segments of skipped points, marked as shaded gray circles. Furthermore, the end- and midpoints of those individual segments also conditionally decorrelate their respective subintervals. That edge is modeled by a 1-D non-redundant model attached to the main quad tree. Similar 1-D offshoot trees are grown for each of the other five vertical and horizontal edges consisting of the boundary point set. The region of support for the four regular children of the root node are marked with empty circles. Functionals kept at the children may consisted of all of or a subsampled set of these points, and are not specifically shown in the figure.

A reduced-order non-redundant model with offshoot nodes is a non-redundant model with an irregular structure, in which the number of children at each node varies from node to node. For instance, the root node of the example in Figure 5-14 has a total of 16 child nodes. With the offshoot nodes, we reduce the state dimensions of the nodes on the main quad tree by adding more nodes. We are claiming that the subsampled boundary-points at the root node can decorrelate not only its regular main subtrees, but also the offshoot subtrees. Similar to the full-order non-redundant model, the non-redundant model with offshoot subtree models all elements of the field of interest. Computing estimates and error statistics on offshoot trees are no different from computing these quantities on the main quad tree. Point measurements are easily modeled on the non-redundant model with offshoot trees, unlike the situation for reduced-order models with interpolation. Full representation of the field of interest is a main advantage of the offshoot model.

It may seem that the offshoot model is the same as a reduced-order model with statistical interpolation. The difference become apparent when we use these models in the multiscale recursive algorithm. The statistical interpolation parameters  $\mathbf{M}^\#(\cdot)$

and  $\mathbf{Q}_\zeta(\cdot)$ , once computed, do not change over time even though the statistics of the error field evolve in time. The model parameters  $\mathbf{A}(\cdot)$  and  $\mathbf{B}(\cdot)$  on the offshoot trees, on the other hand, are updated along with the parameters on the main tree.

The computational complexity of the offshoot non-redundant models depends on whether measurements are placed on the offshoot nodes. When measurements are only present on the main tree, the complexity of the smoothing algorithm is no different from the reduced-order models using interpolation. On the other hand, if measurements are placed on the offshoot nodes, the asymptotic computational complexity of the multiscale smoothing algorithm in offshoot non-redundant models can reach up to  $\mathcal{O}(N^2)$ , higher than even the full-order model. The key reason is the merge step of (A.15) and (A.16) in the smoothing algorithm. The number of matrix inversions, each of which is on the order of the state dimension  $k(s)$ , in the merge step is equal to the number of child subtrees of  $s$  on which there are measurements. On normal trees, whether dyadic or quad, and in offshoot models with no measurements on the offshoot nodes, this number is constant independent of  $N$ . In offshoot models with measurements on the offshoot nodes, the number of matrix inversions can be equal to the number of offshoot nodes at a node on the main tree  $s$ . Designed in the fashion shown in Figure 5-14, an offshoot tree is spawned between every two point functionals at a node the main tree, so that the number of offshoot tree is equal to the state dimension  $k(s)$  itself. Therefore, at each node on the main tree, the computational complexity may reach  $k(s)^3 \cdot k(s)$  rather than  $k(s)^3 \cdot 4$  for normal quad trees or offshoot tree without measurements on offshoot nodes. The overall complexity of the smoothing algorithm<sup>4</sup> may then reach  $\mathcal{O}(N^2)$ . However, the asymptotic figure is somewhat misleading. The  $\mathcal{O}(N^2)$  complexity of offshoot models has a much smaller constant factor of proportionality in front of  $N^2$  than the constant factor for the  $\mathcal{O}(N^{3/2})$  of full-order models, so for 2-D fields of practical size, the higher asymptotic complexity of the offshoot non-redundant model

---

<sup>4</sup>Summing over all nodes on the main tree, the smoothing algorithm costs on the order of  $\sum_{m=0}^{\log_2 \sqrt{N}} 4^m k(s)^4$ . For a constant state dimension, the sum comes out to be  $\mathcal{O}(N)$ . For  $k(s) = c\sqrt{N}/2^{m(s)}$ , the sum is  $\mathcal{O}(N^2)$ . The additional computational cost for smoothing on the offshoot nodes in either case is  $\mathcal{O}(N)$ .

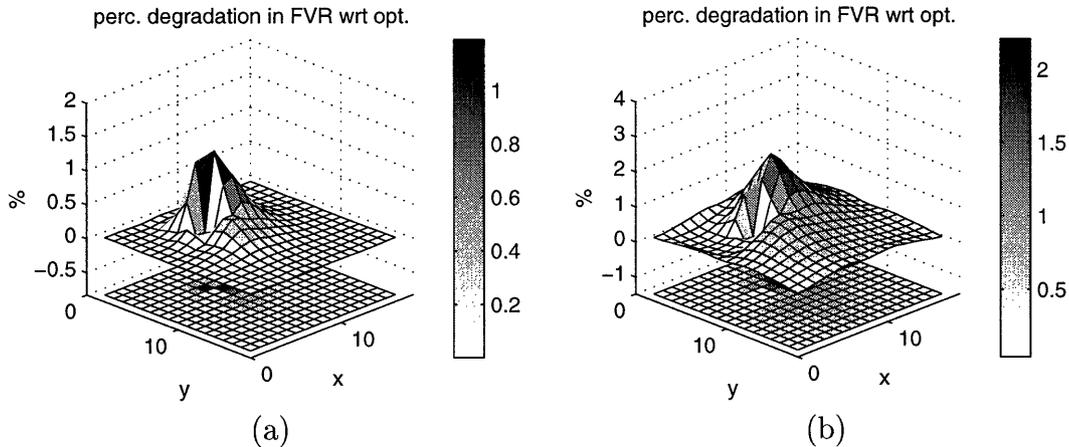


Figure 5-15: The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using reduced-order non-redundant models with offshoot subtrees. The functionals kept at the root node are subsampled boundary-point functionals sampled at a rate of 4 in (a) and 8 in (b). The functionals at scale 1 of the main tree are subsampled at a rate of 3 in both cases. The functionals kept on the main tree are the same as those shown in Figure 5-12(c-f). ( $\beta = 1$ , circular boundary conditions, one point measurement at (6, 7), SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

is essentially irrelevant<sup>5</sup>.

The steady-state performance plots of suboptimal multiscale estimators using offshoot non-redundant models built from the exact steady-state prediction error covariance are shown in Figure 5-15. The state definitions on the main tree for the two cases in Figure 5-15 are the same as those depicted in Figure 5-12. The structure of the offshoot trees are similar to the example shown in Figure 5-14. The offshoot non-redundant models perform comparable to, although worse than, the reduced-order

<sup>5</sup>Suppose that the state dimension at node  $s$  of the full-order model, which is a regular quad tree, is  $k$ . Suppose that the number of operations for each of the matrix inversions in (A.16) is  $k^3$ , so that on the quad tree the computational cost at  $s$  is  $4k^3$ . If we keep every other point along the boundary as the state variables on the main tree, and grow  $k/2$  offshoot nodes, the number of child nodes becomes  $4 + k/2$ . The cost of the smoothing algorithm at node  $s$  is then  $(k/2)^3(4 + k/2) = k^4/16 + k^3/2$ . Comparing this quantity with  $4k^3$ , we see that for  $k < 56$ , the offshoot non-redundant model is less computationally costly than the full-order model. If we keep every fourth boundary point as the state variable on the main tree, i.e., with a subsampling rate of 4, this break-even point becomes 1008, corresponding roughly to a  $169 \times 169$  2-D field, which is much larger than the size we are showing here. If every eighth point along the boundary is kept on the main tree, smoothing in offshoot models takes less computation than in full-order models for fields smaller than  $2727 \times 2727$ . Since offshoot non-redundant models should really be used only when sparse sampling can be used for reducing the state dimensions on the main tree, for 2-D fields of practical size, the higher asymptotic complexity of the offshoot non-redundant model is essentially irrelevant.

models with linear interpolation. The poorer performance results we see here for the offshoot model suggests that the 1-D MRF approximation is actually less accurate than simple linear interpolation.

Up to this point we have looked at several types of multiscale models for modeling the steady-state prediction error of 2-D diffusion. These include internal models with CCR selected linear functionals, internal models with boundary-point linear functionals, non-redundant models, and their reduced-order versions. The steady-state suboptimal multiscale estimators built directly from the exact steady-state prediction error covariance of the optimal estimator are all of acceptable quality, performing not more than a few percent worse than the optimal estimator in the examples that we have shown. Whether these multiscale models lead to computationally efficient and effective multiscale recursive algorithms is examined next.

## 5.4 Multiscale Recursive Estimation Algorithm for 2-D Diffusion

Recall from Chapter 3 that the multiscale prediction step propagates a model for the updated estimation error  $\chi(t|t)$  to a model for the one-step-ahead predicted estimation error  $\chi(t+1|t)$ . The model structure and the definition of the linear functionals remain the same through the prediction step. Model parameters for the prediction error model,  $\mathbf{A}(s; t+1|t)$ ,  $\mathbf{B}(s; t+1|t)$ , and  $\mathbf{P}(0; t+1|t)$ , are computed from the joint statistics,  $\mathbf{P}(s; t+1|t)$  and  $\mathbf{P}(s, s\bar{y}; t+1|t)$ , between parent and child nodes in the prediction error model, which are in turn computed from the joint statistics between parent and child nodes, as well as additional pairs of nodes, on the updated error model. Recall that a state variable anywhere in the predicted error model is related to the finest-scale variables of the update error model through (3.26):  $x_i(s; t+1|t) = \mathbf{l}_i^T(s)\mathbf{A}_d\chi(t|t) + \mathbf{l}_i^T(s)\mathbf{w}_d(t)$ . We attempt to write  $x_i(s; t+1|t)$  more succinctly in terms of the state variables, rather than just the finest-scale variables, on the updated error model via (3.30):  $x_i(s, t+1|t) = \mathbf{h}_i^T(s)\boldsymbol{\xi}(t|t) + \mathbf{l}_i^T(s)\mathbf{w}_d(t)$ , where

$\boldsymbol{\xi}(t|t)$  is the vector containing all the states on the tree model for  $\boldsymbol{\chi}(t|t)$ . The effort in computing the joint statistics between parent-child pairs in the predicted error model and the complexity of the prediction step depends on how sparse and structured  $\mathbf{h}_i^T(s)$  is, which is determined by the particular temporal dynamics  $\mathbf{A}_d$  and the choice of linear functionals  $\mathbf{l}_i^T(s)$ . In 1-D, the computational complexity of the prediction step is  $\mathcal{O}(N \log N)$  for internal models with end-point linear functionals and is  $\mathcal{O}(N)$  on non-redundant models. We use the same general approach, laid out in Section 3.3, for multiscale prediction in 2-D, and analyze the four types of multiscale models discussed so far in this chapter.

### 5.4.1 Multiscale Prediction Step on Internal Models

#### Internal Models with Boundary-Point Linear Functionals

We had assumed earlier for 1-D diffusion, that the dynamics matrix  $\mathbf{A}_d$  is tridiagonal resulting from an explicit temporal discretization scheme. The 2-D version of  $\mathbf{A}_d$  is a block-tridiagonal matrix of tridiagonal blocks. This  $\mathbf{A}_d$  introduces sparse and spatially local mixing. We first consider multiscale prediction for internal models with boundary-point functionals and then those with non-local functionals, e.g., functionals calculated by CCR.

For point linear functionals  $\mathbf{l}_i^T(s)$ ,  $\mathbf{l}_i^T(s)\mathbf{A}_d$  has exactly five nonzero elements and can be written as the weighted sum of five linear functionals, consisting of spatially neighboring elements of  $\boldsymbol{\chi}(t|t)$ . Suppose that we need the covariance  $\mathbf{P}(0; t+1|t)$  at the root node of an internal model with boundary-point linear functionals for the prediction error. The linear functionals at the root node are labeled as solid circles in Figure 5-16 for an example of a reduced-order model in panel (a) and a full-order model in (b). Assume that we have the covariance among these linear functionals,  $\mathbf{P}(0; t|t)$ , on the updated error model, from the measurement update step. Due to dynamic mixing, computing  $\mathbf{P}(0; t+1|t)$  on the predicted error model requires the covariance among not only those functionals marked with solid circles but also their neighboring elements marked with  $\times$ . Compare with pictures of the

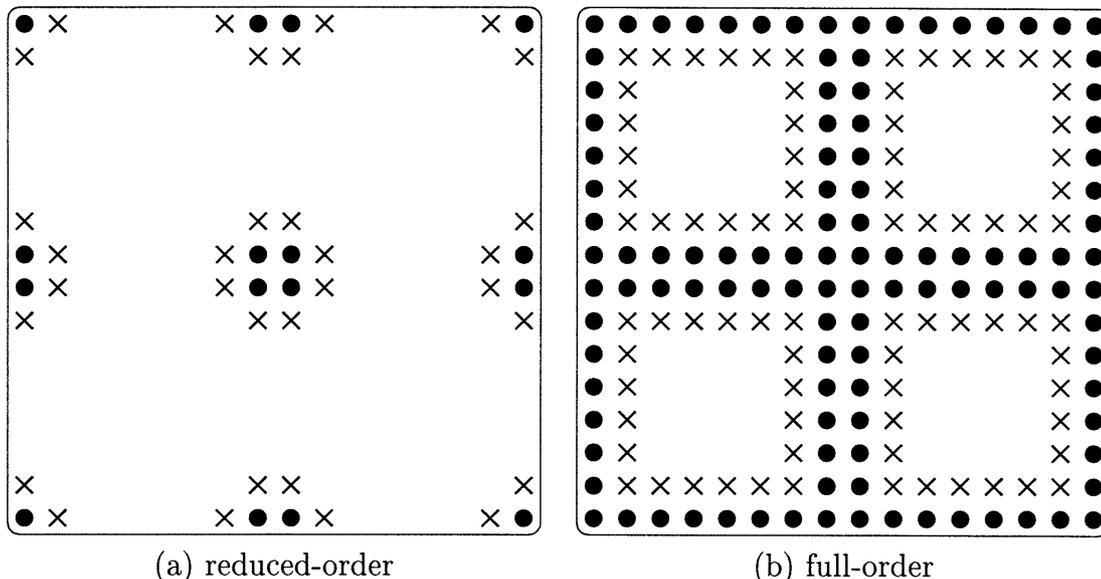


Figure 5-16: The cross-covariances needed in (a) an reduced-order and (b) a full-order internal model with boundary-point linear functionals for the updated estimation errors for multiscale prediction. The linear functionals kept at the root node are labeled as solid circles. The covariance among these functionals are assumed available. The  $\times$  marks indicate the additional functionals on the updated error model that are required for the prediction step due to temporal dynamics. Some of these additional functionals can be found at scale 1; others are found at the finest scale. To compute the covariance at the root node of the predicted error model, joint statistics among nodes with either solid circles or  $\times$  on the updated error model must be computed.

linear functionals kept at scale 1 for the reduced-order model and the full-order model shown in Figures 5-9 and 2-6. For the reduced-order model in Figure 5-16(a), the additional linear functionals due to dynamic mixing are found only at the finest scale (or at the third finest scale, which is scale 2 for the  $16 \times 16$  example, since the finest two scales on the internal model replicates the state variables at the third-finest scale to make finest-scale states scalar, as illustrated in Figure 3-3). For the full-order model, some of the additional point functionals can be found at the immediate child nodes of the root, while most others are found at finer scales.

In general, computing the joint statistics between parent and child nodes in the predicted error model requires the cross-covariances between node pairs that are often far from each other in the updated error model. The computational complexity of calculating all the needed pairs of cross-covariances depend on the state dimensions of the tree model. For a reduced-order model whose state dimension  $k$  is constant,

independent of the size of the field of interest  $N$ , the total cost of the prediction step<sup>6</sup> is then upper-bounded by  $\mathcal{O}(N \log N)$ . This is the same result as that obtained in Section 3.4.2 for 1-D models with end-point linear functionals.

The accounting is more difficult for the case of variable state dimension that is a function of  $N$ . Suppose that we use a fixed subsampling rate for the boundary-point linear functionals, which includes the case of full-order model (for a sampling rate of 1). The state dimension is then proportional to the length of an edge of the 2-D field and decreases at finer scales:  $k(s) = c\sqrt{N}/2^{m(s)}$ . The total cost of computing all the necessary cross-covariances in the updated model for the prediction step<sup>7</sup> is  $\mathcal{O}(N^{5/2})$ .

### Internal Models with Non-Local Linear Functionals

For linear functionals  $\mathbf{l}_i^T(s)$  with spatially non-local region of support, such as those produced by CCR, it is not clear whether or how  $\mathbf{l}_i^T(s)\mathbf{A}_d$  can be succinctly written as the sum of a few functionals on the tree as in (3.27). It is, of course, always possible to write  $\mathbf{l}_i^T(s)\mathbf{A}_d$  as the sum of point linear functionals at the finest scale. If the non-local functionals have region of support consisting of the boundary and midline

---

<sup>6</sup>There are  $\mathcal{O}(k^2)$  entries in  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$ . Since each state variable in the predicted error model is being expressed as a weighted sum of at most five state variables in the updated error model, the cross-covariances of a fixed number (at most 25) of pairs of state variables need to be calculated in the updated error model. The cost of computing the cross-covariance between each pair is no more than  $\mathcal{O}(k^3 \log \sqrt{N})$ , the product of computing cross-covariance across one scale (c.f., the Lyapunov equation (2.10)) and the maximum number of scale between any pair of nodes. Since the number of nodes at scale  $m$  is  $4^m$ , the total cost of the prediction step is then upper-bounded by  $\sum_{m=0}^{\log_2 \sqrt{N}} 4^m k^2 (k^3 \log \sqrt{N})$ , which is  $\mathcal{O}(N \log N)$ .

<sup>7</sup>There are  $k(s)k(s\bar{\gamma}) = c^2 N/2^{m(s)+m(s)-1} \propto N/2^{m(s)}$  number of entries in  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$ , each of which can be derived from a fixed number of pairs of cross-covariances of in the updated error model. Computing cross-covariance across several scales between nodes  $s$  and direct ancestor is  $\sigma$  is proportional to  $\sum_{u=s}^{\sigma} k(u)k(u\bar{\gamma})^2$ . For  $k(s)$  that geometrically decreases as a function of  $m(s)$ , the sum becomes  $\sum_{u=s}^{\sigma} c^3 N^{3/2} / (2^{m(u)} 2^{2(m(u)-1)}) \propto N^{3/2} \sum_{u=s}^{\sigma} 1/2^{3m(u)}$ . This last sum is proportional to  $1/2^{3m(s \wedge \sigma)}$ , i.e., the cost of computing the cross-covariance between two nodes  $s$  and  $\sigma$  is dominated by computing the cross-covariance at their lowest-scale common ancestor  $s \wedge \sigma$  and its children. Suppose that computing  $\mathbf{P}(s, s\bar{\gamma}; t+1|t)$  at node  $s$  involves only new computation of cross-covariances among nodes below  $s$ , because the cross-covariances among nodes above  $s$  have already been computed for  $\mathbf{P}(u, u\bar{\gamma}; t+1|t)$  for some node  $u$  at a coarser scale than  $s$ . Summing over all nodes, the total cost of computing all the needed cross-covariances is then  $\propto \sum_{m=0}^{\log_2 \sqrt{N}} 4^m (N/2^{2m}) (N^{3/2}/2^{3m})$ , i.e.,  $\mathcal{O}(N^{5/2})$ .

points only, and  $\mathbf{l}_i^T(s)\mathbf{A}_d$  cannot be written accurately as the sum of only a few functionals on the tree as in (3.27) but only as the sum of point linear functionals at the finest scale, then the amount of computation required for multiscale prediction is the same as that for the full-order model with boundary-point functionals, regardless of whether the model is of full-order or reduced-order. While non-local functionals, such as those computed by CCR, can better capture conditional correlations as we saw in Section 5.3.1, the non-local support of these functionals makes the multiscale prediction step computational costly.

## 5.4.2 Multiscale Prediction Step on Non-Redundant Models

### Full-Order Non-Redundant Models

Multiscale prediction on non-redundant models is much simpler than prediction on internal models in terms of both computational complexity and computer implementation. The reason is the same as for 1-D models discussed in Section 3.5.2. For full-order non-redundant models, the additional functionals on the updated error model needed for prediction due to dynamic mixing (with an  $\mathbf{A}_d$  that is block-tridiagonal with tridiagonal blocks) are always found at either a parent or a child node. (Refer to Figure 5-10 for a graph of the linear functionals at the root node and the nodes at scale 1 for a  $17 \times 17$  example.) Thus, to compute either the covariance  $\mathbf{P}(s; t + 1|t)$  or the cross-covariance  $\mathbf{P}(s, s\bar{\gamma}; t + 1|t)$  at node  $s$  on the predicted error model, the cross-covariances of no more than  $s\bar{\gamma}\bar{\gamma}$ ,  $s\bar{\gamma}$ ,  $s$ , and  $s\alpha_i$  on the updated error model need to be computed. Therefore, the cost of computing  $\mathbf{P}(s, s\bar{\gamma}; t + 1|t)$  at a node  $s$  is  $\mathcal{O}(k(s)^3)$ . For full-order models, where  $k(s) = 6\sqrt{N}/2^{m(s)} - 9$ , the total computational cost of the prediction step is  $\propto \sum_{m=0}^{\log_2 \sqrt{N}} 4^m \left(\sqrt{N}/2^m\right)^3 \propto N^{3/2}$ , the same computational complexity as the smoothing algorithm. Furthermore, keeping track of which cross-covariance pairs in the updated error model must be computed is much simpler for a non-redundant model, lessening overhead and making implementation easier.

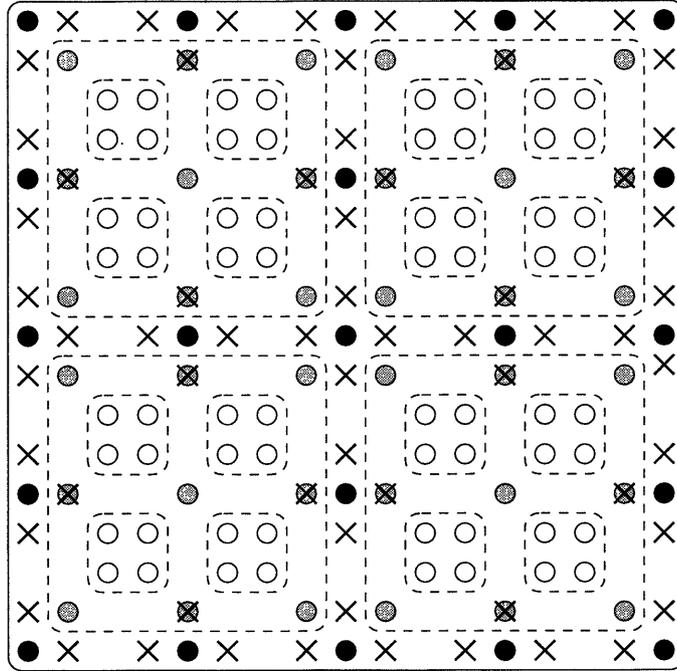


Figure 5-17: The cross-covariances needed in a reduced-order non-redundant model for the updated estimation errors for multiscale prediction. The linear functionals kept at the root node are labeled as solid black circles. The covariance among these functionals are assumed available. The  $\times$  marks indicate the additional functionals on the updated error model that are required for the prediction step due to temporal dynamics. Some of these additional functionals can be found at scale 1, where the functionals are labeled as shaded gray circles. Functionals at scale 2 are labeled as unfilled circles. To compute the covariance at the root node of the predicted error model, joint statistics among nodes with either solid black circles or  $\times$  on the updated error model must be computed.

### Reduced-Order Non-Redundant Models with Interpolation

For reduced-order non-redundant models with interpolation, since some elements of the field of interest are not represented on the tree model, some of the additional linear functionals required for prediction due to dynamics mixing are not found on the tree. In Figure 5-17 the functionals at the root node are labeled as solid black circles; functionals at scale 1 are labeled with shaded gray circles; and those at scale 2 with unfilled circles. Since the model is non-redundant, we can plot the boundary-point functionals at more than one scale in the same figure without any confusion. The additional functionals needed for the updated model for computing the covariance at the root node of the predicted error model  $\mathbf{P}(0; t+1|t)$  are marked with  $\times$ . Some of these  $\times$  do not

overlap the shaded gray circles, i.e., these additional functionals are not represented on the tree. However, for the purpose of the prediction step, we may approximate their statistics via the interpolation equations of (5.3) and (5.5). That is, for linear interpolation the covariance of the entire boundary  $\zeta(s; t|t)$  is  $\mathbf{M}^\#(s)\mathbf{P}(s; t|t)\left(\mathbf{M}^\#(s)\right)^T$ , and for statistical interpolation,  $\mathbf{M}^\#(s)\mathbf{P}(s; t|t)\left(\mathbf{M}^\#(s)\right)^T + \mathbf{Q}_\zeta(s\alpha_\zeta)$ .

The interpolation matrix  $\mathbf{M}^\#(s)$  is sparse for linear interpolation and generally dense for statistical interpolation. Let the state dimension be  $k(s)$  at node  $s$  and the number of elements on the boundary be  $d(s)$ , which is proportional to  $\sqrt{N}/2^{m(s)}$ . Then, computing the covariance of the entire boundary  $\zeta(s; t|t)$  is on the order of  $d(s)k(s)^2$  for a dense interpolation matrix  $\mathbf{M}^\#(s)$  and on the order of  $d(s)$  for a sparse matrix. In either case, for a constant state dimension  $k$  the cost of interpolation<sup>8</sup> is  $\mathcal{O}(N)$ . For a variable state dimensions  $k(s) = c\sqrt{N}/2^{m(s)}$ , the cost of interpolation for a dense  $\mathbf{M}^\#(s)$  is  $\propto \sum_{m=0}^{\log_2 \sqrt{N}} 4^m \left(\sqrt{N}/2^m\right)^3 \propto N^{3/2}$ , and for a sparse  $\mathbf{M}^\#(s)$  is  $\propto \sum_{m=0}^{\log_2 \sqrt{N}} 4^m \left(\sqrt{N}/2^m\right) \propto N$ . The complexity of interpolation, therefore, does not exceed the cost associated with computing covariances and cross-covariances among the states in the reduced-order non-redundant model itself, which is similar to the smoothing algorithm at  $\mathcal{O}(N)$  for a fixed  $k(s)$  and at  $\mathcal{O}\left(N^{3/2}\right)$  for a variable  $k(s)$ .

### Non-Redundant Models with Offshoot Nodes

Non-redundant models with offshoot trees reduce the state dimensions by keeping subsampled boundary point functionals on the main tree and moving the skipped ones onto the offshoot tree connected to the same node, while maintaining full representation of all elements of the field of interest. Some of the state variables that were modeled at a node on the full-order model are now distributed on the offshoot trees connected to the node. To derive the model parameters on the predicted error model, we need parent-child joint statistics at all nodes, both on the main tree and on the offshoot trees. Such statistics are derived from the updated error model involving some additional functionals due to dynamic mixing. For a node on the main

---

<sup>8</sup>For a dense  $\mathbf{M}^\#(s)$ , the total cost of interpolation is on the order of  $\sum_{m=0}^{\log_2 \sqrt{N}} 4^m \sqrt{N}/2^m k^2 \propto N$ . For a sparse  $\mathbf{M}^\#(s)$ , the cost is on the order of  $\sum_{m=0}^{\log_2 \sqrt{N}} 4^m \sqrt{N}/2^m$ , again  $\mathcal{O}(N)$ .

multiscale models	full-order	reduced-order	
		constant $k$	$k(s) \propto \sqrt{N}/2^{m(s)}$
internal, boundary-point	$N^{5/2}$	$N \log N$	$N^{5/2}$
internal, non-local	$N^{5/2}$	$N^{5/2}$	$N^{5/2}$
non-redundant	$N^{3/2}$	—	—
non-red., interpolation	—	$N$	$N^{3/2}$
non-red., offshoot	—	$N$	$N^{3/2}$

Table 5.1: Asymptotic computational complexity of the multiscale prediction step for several types of multiscale models.  $N$  is the size of the 2-D random field.  $k(s)$  is the state dimension at node  $s$ .

tree these additional linear functionals can be found at either a parent or child node or on the offshoot trees connected to those nodes. For a node on the offshoot tree connected to  $s$ , the additional functionals can be found either somewhere on the same offshoot or on the offshoot trees connected to  $s\bar{\gamma}$  or  $s\alpha$ . That is, computing the joint statistics between a parent and child pair on the predicted error model requires the joint statistics of only a fixed number of nodes on the main tree and their offshoot trees on the updated error model.

The cost of computing the needed covariances and cross-covariances on the main tree is  $\mathcal{O}(N)$  for a fixed  $k(s)$  and  $\mathcal{O}(N^{3/2})$  for a variable  $k(s)$ . The cost of computing the necessary statistics on the offshoot trees<sup>9</sup> is also  $\mathcal{O}(N)$  for a fixed  $k(s)$  and at  $\mathcal{O}(N^{3/2})$  for a variable  $k(s)$ .

To summarize, in Table 5.1 we list the asymptotic computational complexity of

<sup>9</sup>Computing the needed cross-covariances between parent and child node pairs on the main tree of the updated error model is of order  $k(s)k(s\alpha)^2$ . Since the offshoot trees have a constant state dimension of 3, computing the cross-covariance between node  $s$  on the main tree and its children is of order  $3k(s)^2$ . Subsequent computation of cross-covariances between  $s$  on the main tree and the finer-scale offshoot nodes is of order  $9k(s)$ . Given that there are  $\propto \sqrt{N}/2^{m(s)}/k(s)$  nodes on the offshoot tree connected to  $s$ , the total cost of computing cross-covariances on the offshoot tree is  $3k(s)^2 + 9\sqrt{N}/2^{m(s)}$ . Summing over all nodes, the cost of the prediction step, for a fixed  $k$ , is  $\propto \sum_{m=0}^{\log_2 \sqrt{N}} 4^m k^3 + \sum_{m=0}^{\log_2 \sqrt{N}} 4^m (3k^2 + 9\sqrt{N}/2^m)$ , which is  $\mathcal{O}(N)$ . For a variable  $k(s)$ , the total cost is  $\propto \sum_{m=0}^{\log_2 \sqrt{N}} 4^m N^{3/2}/2^{3m} + \sum_{m=0}^{\log_2 \sqrt{N}} 4^m (3N/2^{2m} + 9\sqrt{N}/2^m) \propto N^{3/2} + N \log N + N$ , which is  $\mathcal{O}(N^{3/2})$ .

the multiscale prediction step for the several types of multiscale models considered in this section. Modeling a 2-D field with constant state dimension without substantial sacrifice in model fidelity is difficult. Decorrelating large 2-D regions, especially at the coarser nodes on the tree, generally requires more state variables. Therefore, the cost figures in the rightmost column in Table 5.1 are of more relevance to us. It is clear that non-redundant models are computationally more efficient for multiscale prediction. Asymptotic computation complexity, however, does not paint the full picture. For instance, reduced-order models are certainly less computationally costly than full-order models, even though both may be of the same order. The asymptotic complexity figures ignore the constant factor of proportionality in computational cost associated with each algorithm, i.e., whether an algorithm grows as  $10N$  or  $100N$ . Because of this constant factor and implementation overhead, when comparing models and algorithms with different growth rate, it is possible that the size of the problem may need to be very large for a slower growing method to be less costly, e.g., regarding the complexity of the smoothing algorithm for offshoot non-redundant models discussed towards the end of Section 5.3.3. In practice, applications in 2-D often focus more on what size problem can be solved in reasonable amount of time rather than what the asymptotic behavior of the particular model or algorithm used. Among the several types of non-redundant models, the reduced-order models with linear interpolation are the fastest. Non-redundant models with offshoot nodes are also contenders as they hold the advantage of full representation of field of interest. In the next section, we will focus on these two types of models and show some results for the iterative and recursive implementation of our multiscale dynamic estimation algorithm.

## 5.5 Examples and Results

In Section 4.4, we had shown 1-D examples of size  $N = 64$ . The added spatial dimension for 2-D problems makes the comparison of our multiscale estimators with optimal ones much more difficult. Solving the Riccati equation for the optimal estimator and the steady-state error covariance or the Lyapunov equation for the steady-state per-

formance of the suboptimal multiscale estimator are computationally intensive for systems not much larger than the  $17 \times 17$  examples we have shown, even though the benefits and advantages of using multiscale estimators are to be seen for much larger problems.

In this section, we showcase two diffusion examples in more detail. The first is the diffusion example on a toroid, which we have used so far in this chapter. The other is a pinned cooling sheet, which represents a more realistic setup, for instance, for a radiator fin. We continue with the single measurement case, but add a multiple measurement case and other variations, such as temporally subsampled measurements. We show results for the iterative implementation of steady-state multiscale estimators as well as recursive implementation results for time-varying systems. We compare the several types of non-redundant models that we discussed before.

### 5.5.1 Iteratively Implemented Steady-State Estimators

First, we show several examples of suboptimal steady-state multiscale estimators that are implemented iteratively using the multiscale recursive algorithm. The iterative procedure does not assume the availability of the exact steady-state prediction error covariance  $\tilde{\mathbf{P}}_p$ . It starts from a model realized from the steady-state process covariance  $\mathbf{P}_z$  and iteratively computes the predicted error models. The procedure is terminated at about one time constant of the slowest mode of the optimal steady-state estimator. We then solve the Lyapunov equation (3.36) for the steady-state performance of the multiscale suboptimal estimator. The optimal steady-state error covariance  $\tilde{\mathbf{P}}_p$  is explicitly solved to allow comparisons with the suboptimal estimators.

#### Single Measurement

We continue the  $17 \times 17$  examples shown previously for a 2-D diffusion process on a toroid with a single measurement at (6, 7). The FVR of the optimal steady-state estimator was shown before in Figure 5-11(a). Here, we compare the steady-state

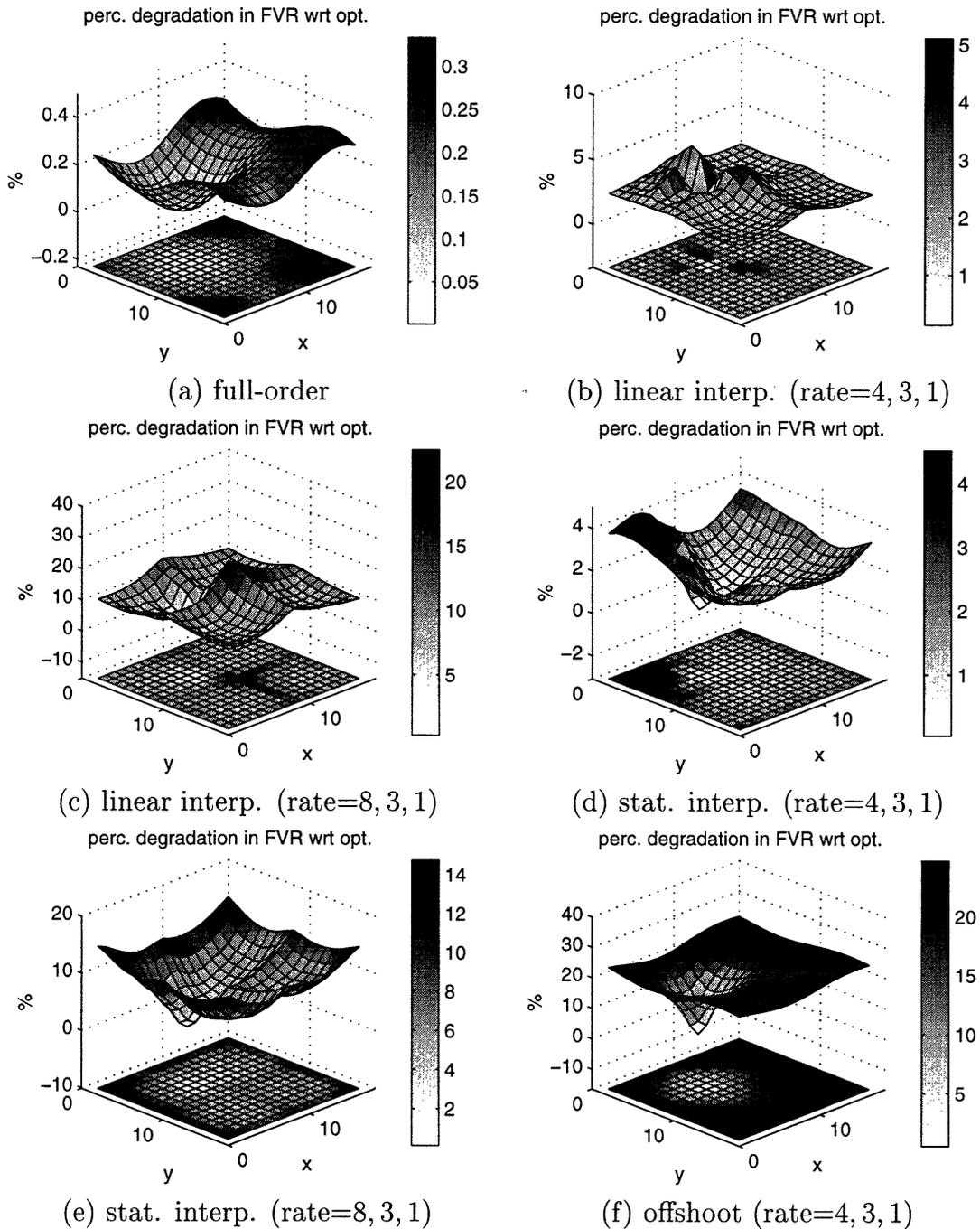


Figure 5-18: Performance of the steady-state multiscale estimators for a 2-D diffusion process on a toroid with one point measurement at  $(6, 7)$ . The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using (a) a full-order non-redundant model, (b)–(c) a reduced-order model with linear interpolation, (d)–(e) a reduced-order model with statistical interpolation, and (f) a non-redundant model with offshoot nodes. The boundary points modeled on the reduced-order models or the main tree of the offshoot model are subsampled at rate of 4, 3, 1 or 8, 3, 1 at scales 0–2, as indicated. ( $TC_{K(\infty)} \approx 1654$  steps = 0.0661 sec, iterations terminated at step 1800,  $\beta = 1$ , SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

performance of several suboptimal multiscale estimators. Figure 5-18 depicts the percent degradation in FVR of the iterative derived suboptimal multiscale estimators with respect to the optimal. In panel (a), we see that the multiscale estimator using a full-order non-redundant model performs no more than 1% worse than the optimal. This result, as well as later ones for the full-order models, confirms our findings that the exact multiscale models built for MRFs and the multiscale recursive algorithm yield high-quality suboptimal steady-state multiscale estimators for diffusion processes. The iterative procedure for constructing a multiscale steady-state estimator works well given that the full-order model adequately captures the estimation error statistics.

The more interesting question in 2-D is how much we may reduce the state dimensions of the multiscale models and still obtain reasonable steady-state estimator performance. In Figures 5-18(b)–(e), we show the performance results of four estimators using reduced-order non-redundant models whose linear functionals are spatially subsampled boundary points, the same as those used in Figure 5-12. We use the sampling rate as a label to distinguish the different models in the figure. The skipped boundary points are interpolated with a simple linear interpolation scheme for the estimators in panel (b) and (c) and with a statistical interpolation scheme, whose parameters are computed from the initial condition  $\mathbf{P}_z$ , in panels (d) and (e). We observe that the two interpolation methods yield comparable performance results. The statistical interpolation method is more accurate since its interpolation of the elements of the error field not modeled are better tailored to the statistical structure of the error process. For both interpolation methods, the lower-dimensional models, as expected, yield poorer results than the higher-dimensional models, and show more pronounced artifacts due to interpolation, especially at the root node where the boundary points are more sparsely sampled. If we use the steady-state process covariance  $\mathbf{P}_z$ , which has correlation length of about four or five pixels, as a guide, it seems reasonable that a sampling rate of 4 for the reduced-order model yields acceptable multiscale estimators that are no more than 5% worse than the optimal, as depicted in Figures 5-18(b) and (d). On the other hand, a rate of 8 yields estimators

that are much worse.

The performance result for an estimator using the offshoot tree model whose functionals on the main tree are subsampled at 4, 3, 1 is depicted in Figure 5-18(d). Even though we already knew from Figures 5-12 and 5-15 that estimators realized from the exact  $\tilde{P}_p$  on offshoot tree models yielded poorer performance results, it is still surprising that the estimator using this more sophisticated offshoot model performs poorer than simple interpolation in panels (b) and (d). It must be noted, however, that around the measurements, where one cares most about the estimation quality, the degradation in performance for the estimator using the offshoot model is only a few percent, although in areas far from the measurements, the estimator performs much worse. The offshoot model may be thought of as a statistical interpolation method whose interpolation parameters  $M^\#(\cdot)$  and  $Q_\zeta(\cdot)$  are time varying to better match the statistics of the error field. The use of offshoot trees is based on the assumption that sampled boundary points kept on the main tree can cleanly conditionally decorrelate the offshoot trees. For it to be valid,  $Q_\zeta(\cdot)$  would have to be block diagonal. By examining the models using statistical interpolation in Figure 5-18(d) and (e), we find that the  $Q_\zeta(\cdot)$  are not close to block diagonal. It is then not so surprising that the offshoot models do not perform as well as the statistical interpolation. Rather, it is surprising that simple linear interpolation can work so well.

For large systems, it is not computationally feasible to solve the Lyapunov equation (3.36) for the steady-state performance figures for the suboptimal multiscale estimators. The error variances realized by the multiscale estimator, i.e., what the estimator believes it is achieving, is used as a proxy. In Figures 5-19 we compare the optimal steady-state updated error variance with those realized by the multiscale estimators. The full-order model captures the error variances essentially correctly as seen in panel (b). The reduced-order models all show obvious artifacts. The variances realized by the two reduced-order models with linear interpolation are shown in panels (c) and (d). Both of these overestimate the true magnitude of the error variances. The variances realized by the reduced-order model with statistical interpolation, shown in panel (e), are the most accurate among the reduced-order models.

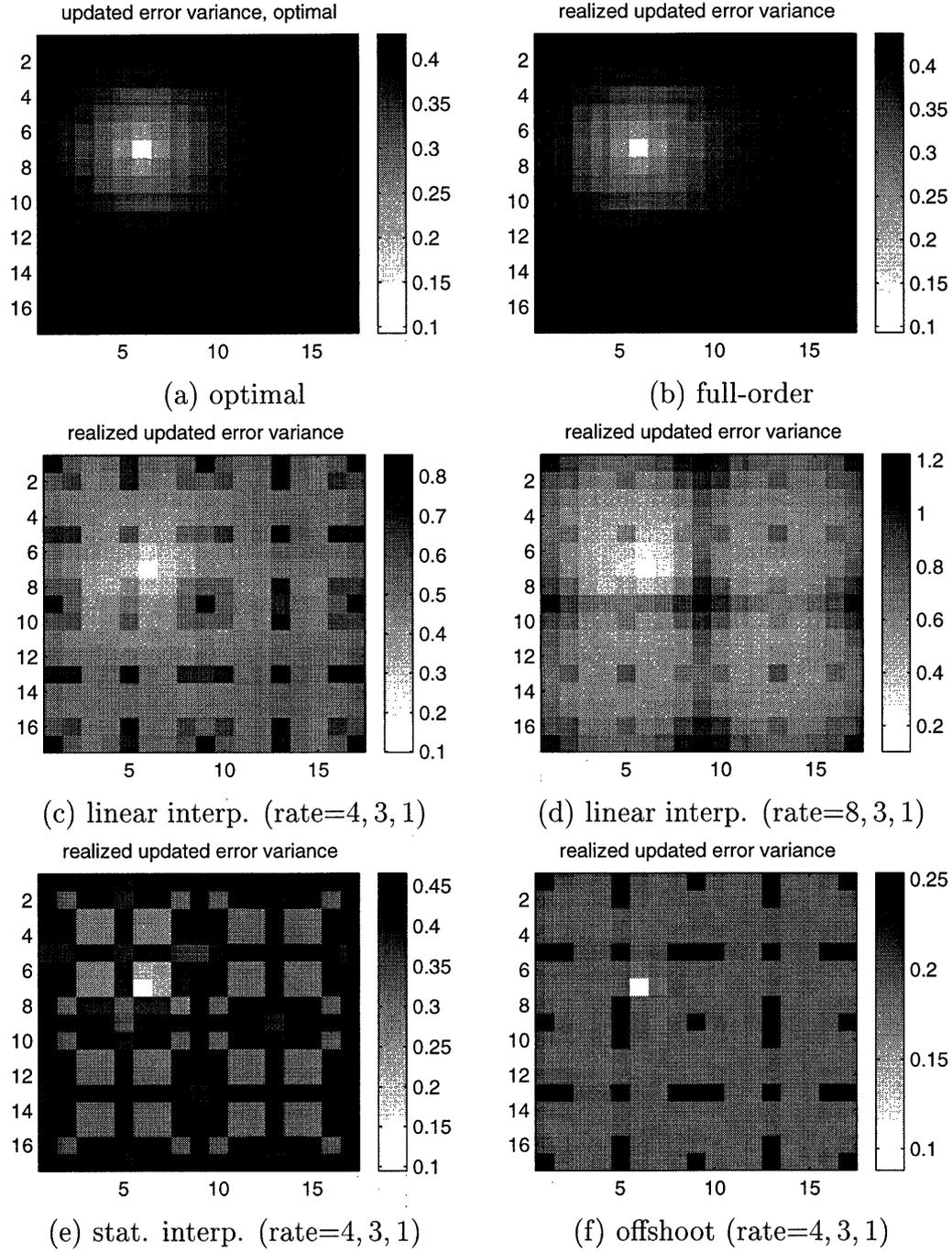


Figure 5-19: A comparison between the steady-state updated error variance of the optimal estimator in (a) and the realized error variances of the suboptimal multiscale estimators in Figure 5-18. The realized error variances are for multiscale estimators using (b) the full-order non-redundant model, (c)–(d) the reduced-order model with linear interpolation, (e) the reduced-order model with statistical interpolation, and (f) the offshoot model.

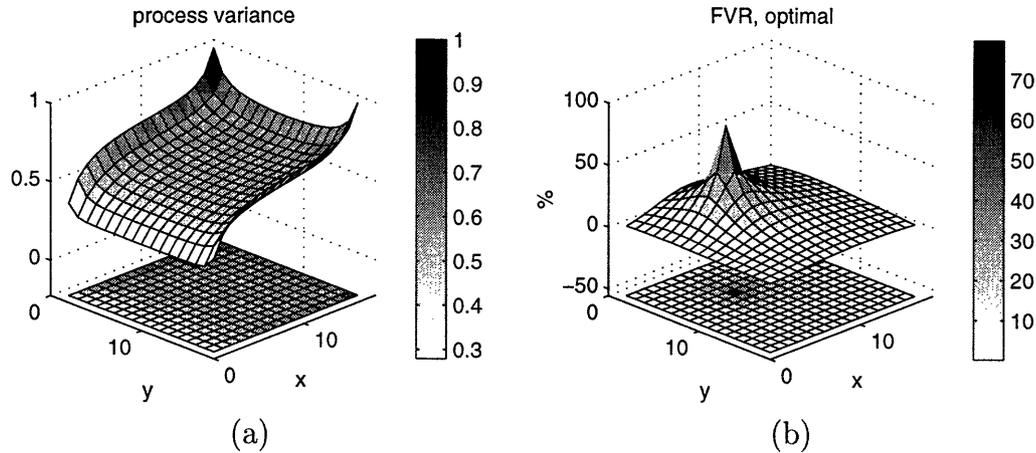


Figure 5-20: (a) The steady-state process variance of a 2-D diffusion process on a cooling sheet. (b) The FVR of the steady-state optimal estimator with a single point measurement at  $(6, 7)$ . ( $\beta = 1$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 4 \times 10^{-5}$ .)

Not surprisingly, the estimator also had the best performance. The offshoot model underestimates the true magnitude of the error variances.

The performance of the multiscale estimators hinges on whether the models can accurately capture the error statistics. Since the error statistics depend on system parameters, such as number and quality of measurements, boundary conditions, etc., the performance of the multiscale estimator is influenced by the same parameters. For example, the same multiscale estimators we have examined in Figure 5-18 perform worse for estimating a diffusion process on a cooling sheet, such as a radiator fin. The steady-state process is no longer spatially stationary since one edge of the cooling sheet is at a fixed temperature and the other three are freely immersed in the coolant. The steady-state process variance, shown in Figure 5-20(a), is lower near the pinned-down edge, and higher along the other three edges. The steady-state updated error variance has a similar shape but dips to lower values near the measurement. The FVR of the optimal estimator is depicted in Figure 5-20(b).

The iteratively constructed multiscale model using a full-order model is, in fact, highly accurate. Its percent degradation in FVR with respect to the optimal, shown in Figure 5-21(a) is less than 1%. However, the reduced-order models fared worse than for the case of toroidal boundary conditions. This is essentially a modeling issue, namely that the same reduced-order models are not as good at capturing the error

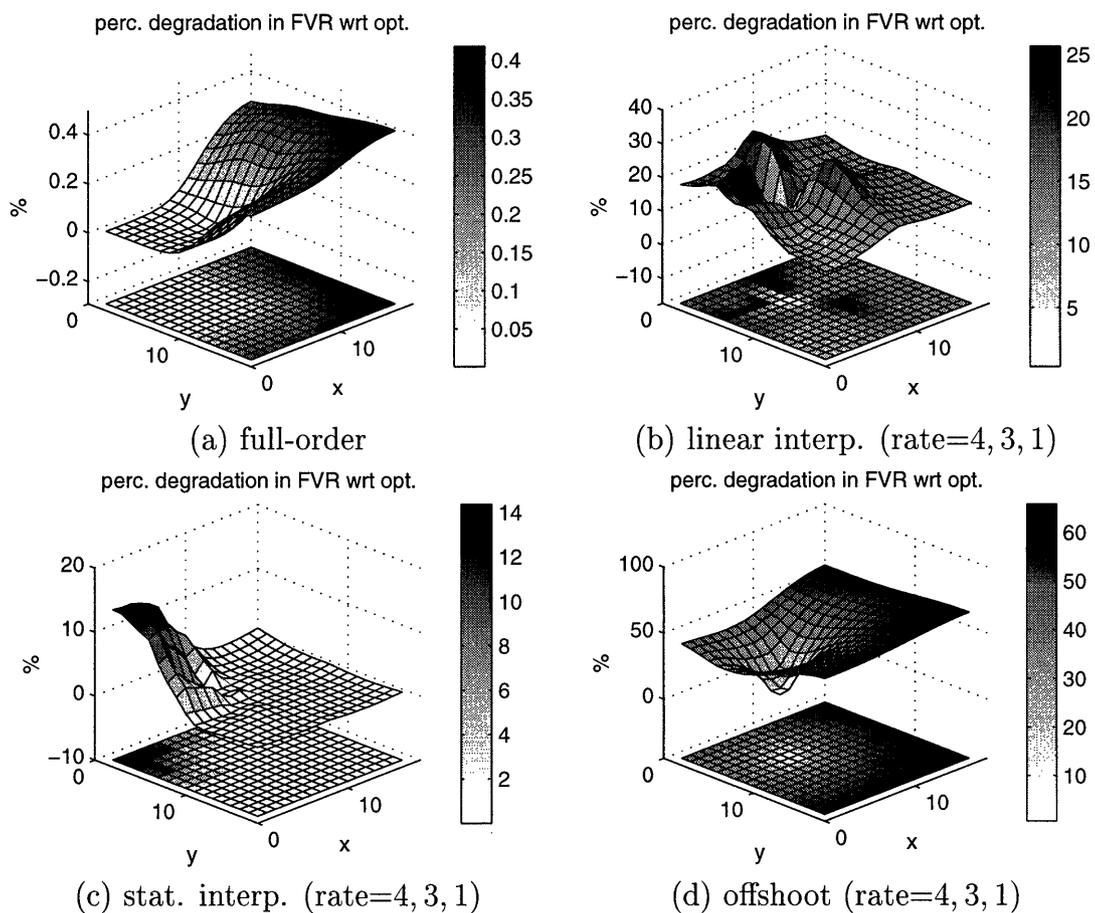


Figure 5-21: Performance of the steady-state multiscale estimators for a 2-D diffusion process on a pinned cooling sheet with one point measurement at (6, 7). The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using (a) a full-order non-redundant model, (b) a reduced-order model with linear interpolation, (c) a reduced-order model with statistical interpolation, and (d) a non-redundant model with offshoot nodes. ( $TC_{K(\infty)} \approx 1874 \text{ steps} = 0.0750 \text{ sec}$ , iterations terminated at step 1800,  $\beta = 1$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 4 \times 10^{-5}$ .)

statistics on a sheet with the added asymmetry due to the non-cyclic boundary conditions. Note that the statistical interpolation method still yields acceptable results, as seen in panel (c), while the percent degradations in FVR of the estimator using linear interpolation in panel (b) and the estimator using offshoot model in panel (d) are only acceptable near the measurement.

With a single point measurement the system is very weakly observable. It is, in a sense, the worst case situation. The performance of the multiscale steady-state estimators improve with more measurements, as the examples we show next indicate.

### Multiple Measurements

First, we illustrate the estimator performance results with a four-measurement example on a toroid. The measurement locations, (6, 7), (3, 12), (10, 5), and (11, 14), are selected such that they are all modeled in the reduced-order models. The FVR of the optimal estimator has its peaks, i.e., the most reduction in uncertainty in our knowledge about the process due to measurements, near the measurements, and is not plotted here. Figure 5-22 shows the steady-state performance degradation of our sub-optimal multiscale estimators with respect to the optimal estimator. All four perform better than their counterparts for the one measurement case in Figure 5-18. Again, the full-order model yields the best and near-optimal estimator performance. The reduced-order estimators, including the offshoot model, are all within a few percent worse than the optimal estimator.

The steady-state performance results of multiscale estimators for estimation on a pinned cooling sheet with four measurements are shown in Figure 5-23. Again, all four multiscale estimators perform better than their counterparts for the single-measurement case in Figure 5-18. As in the single-measurement case, estimation results are poorer on the cooling sheet than those on the toroid. The full-order model, shown in Figure 5-23(a), again yields the best and near-optimal estimator performance, validating the iterative method itself. The reduced-order estimators with interpolations, in panels (b) and (c), show acceptable performance; while the offshoot model in panel (d) exhibits the worse performance numbers.

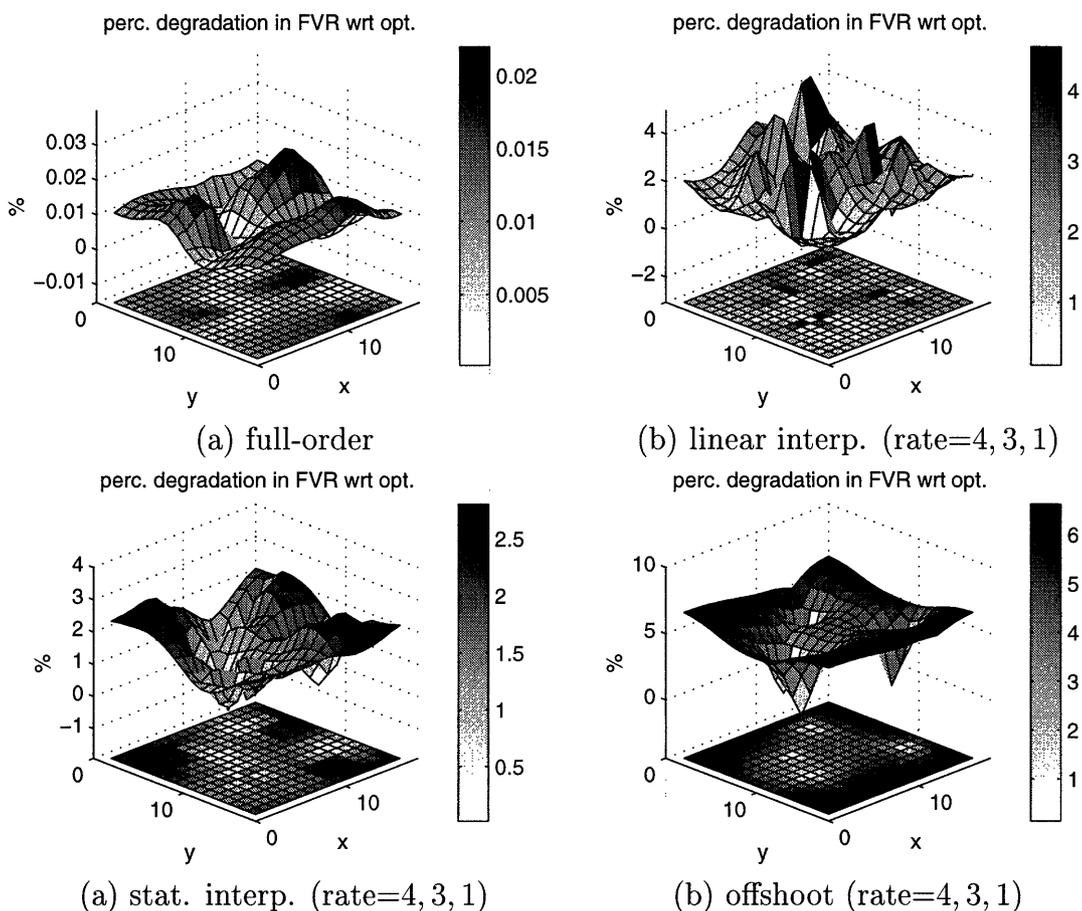


Figure 5-22: Performance of the steady-state multiscale estimators for a 2-D diffusion process on a toroid with four point measurements at  $(6, 7)$ ,  $(3, 12)$ ,  $(10, 5)$ , and  $(11, 14)$ . The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using (a) a full-order non-redundant model, (b) a reduced-order model with linear interpolation, (c) a reduced-order model with statistical interpolation, and (d) a non-redundant model with offshoot nodes. ( $TC_{K(\infty)} \approx 782$  steps = 0.0313 sec, iterations terminated at step 1000,  $\beta = 1$ , SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

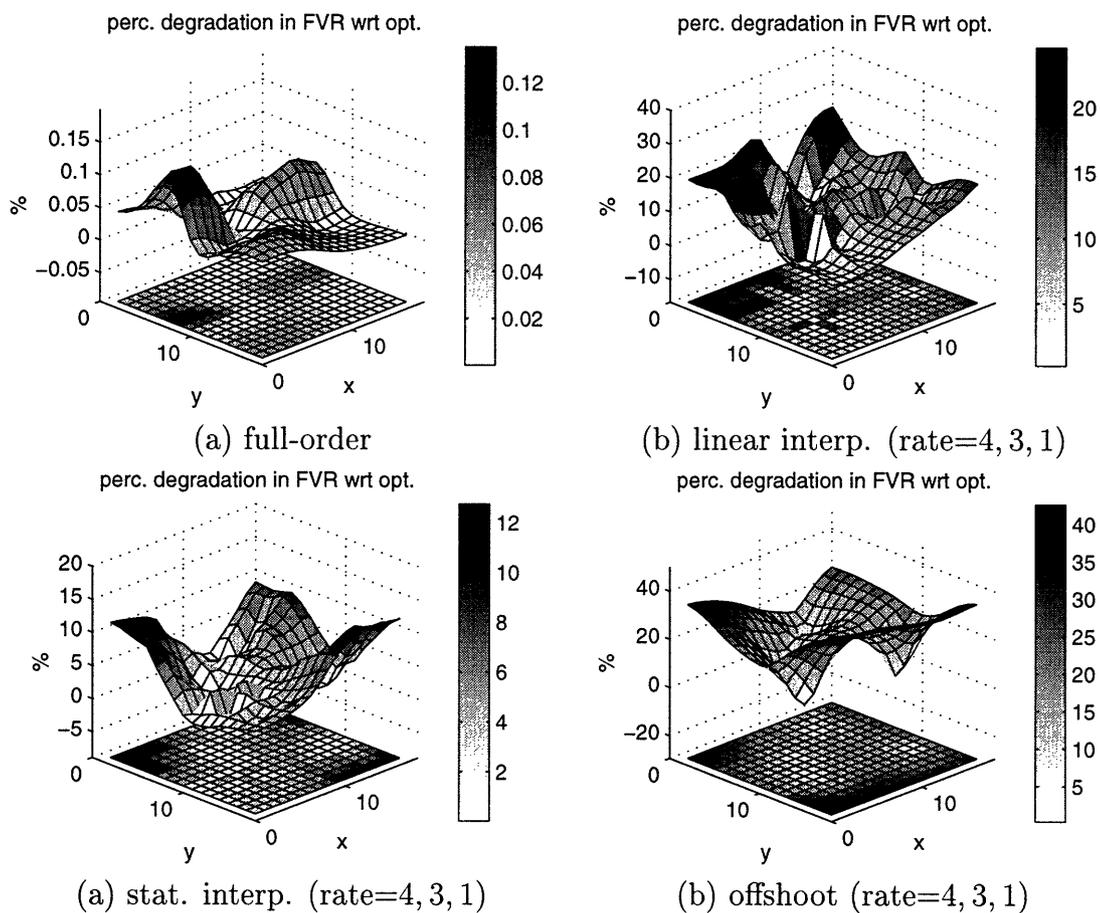


Figure 5-23: Performance of the steady-state multiscale estimators for a 2-D diffusion process on a pinned cooling sheet with four point measurements at  $(6, 7)$ ,  $(3, 12)$ ,  $(10, 5)$ , and  $(11, 14)$ . The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using (a) a full-order non-redundant model, (b) a reduced-order model with linear interpolation, (c) a reduced-order model with statistical interpolation, and (d) a non-redundant model with offshoot nodes. ( $TC_{K(\infty)} \approx 766$  steps = 0.0306 sec, iterations terminated at step 1000,  $\beta = 1$ , SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

## Multiple Prediction Steps for Each Update Step

The results so far on the iteratively implemented steady-state multiscale estimators suggest that the poorer performance of some of the estimator examples are mostly due to the reduced-order models' inability to fully capture the error statistics rather than the effectiveness of the recursive procedure itself. Similar to what we did for 1-D diffusion process, we examine the cumulative effect of the multiscale prediction steps by temporally subsampling the measurements. Inaccuracies in the multiscale prediction steps would be amplified by having multiple prediction steps for each update step. Furthermore, the rate of available measurements in practical problems may indeed be much slower than the particular temporal discretization step size we use for modeling the dynamics. In Figure 5-24, we show the results for the same cooling sheet example as in Figure 5-23 except that there is only one measurement update step for every 100 prediction steps. With fewer measurement updates per unit time, the reduction in uncertainty in our knowledge about the process due to measurements is less, as indicated by the lower FVR of the optimal estimator in Figure 5-24(a). In panels (b), we show the percent degradation in FVR of the estimator using a full-order non-redundant model. While its performance is not as good as before with temporally dense measurements, the estimator using the full-order model is still only a few percent worse than the optimal estimator. The performance plots of the two estimators using reduced-order models with interpolation, depicted in panels (c) and (d), show comparable results as before with temporally dense measurements. The results here confirm effectiveness of the iterative algorithm and that the quality of the steady-state multiscale estimators depends mostly on how accurately the models capture the statistics of the estimation error field.

### 5.5.2 Recursive Implementation

Lastly, we show an example of recursive implementation of our multiscale algorithm to estimate a 2-D diffusion process using temporally nonstationary measurements. We compare the estimation results from the Kalman filter with those from estimators

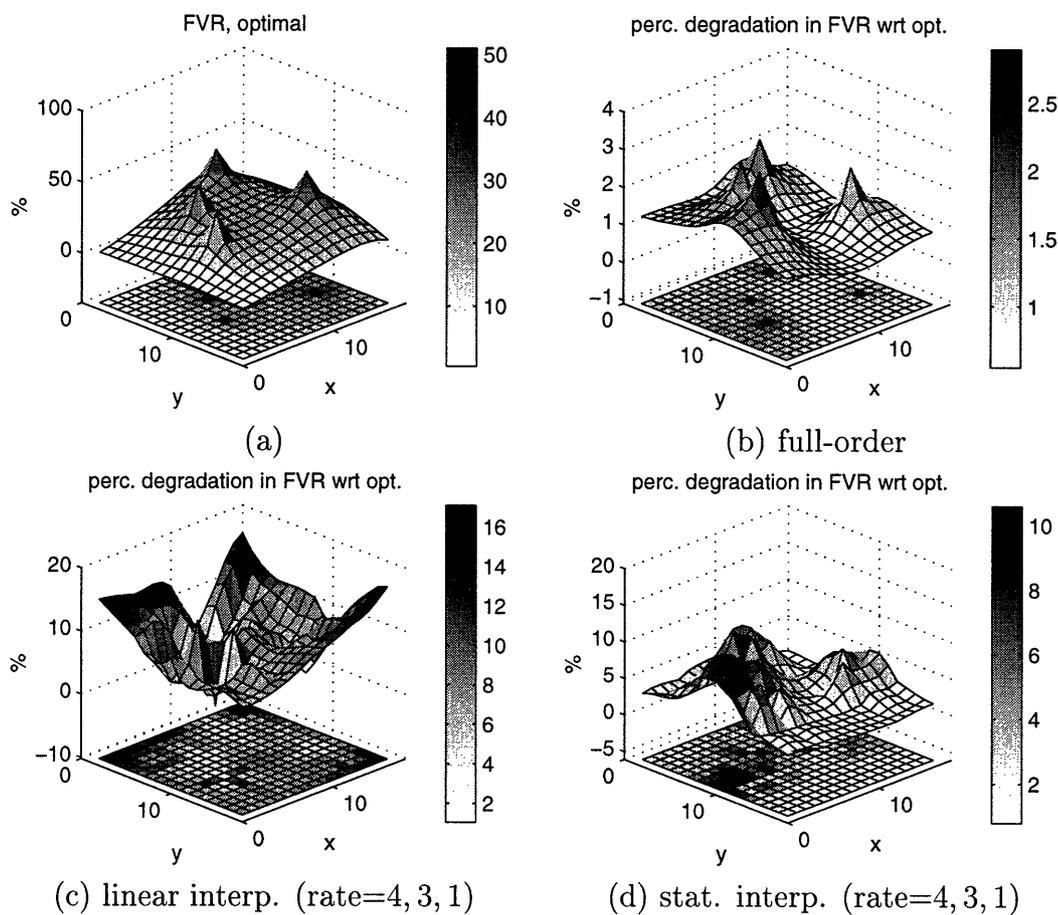


Figure 5-24: Performance of the steady-state multiscale estimators for a 2-D diffusion process on a pinned cooling sheet with four point measurements at  $(6, 7)$ ,  $(3, 12)$ ,  $(10, 5)$ , and  $(11, 14)$  and only one measurement update step for every 100 prediction steps. (a) The FVR of the optimal steady-state estimator. The percent degradation in FVR with respect to the optimal estimator of the suboptimal steady-state multiscale estimators using (b) a full-order non-redundant model, (c) a reduced-order model with linear interpolation, and (d) a reduced-order model with statistical interpolation. ( $TC_{K(\infty)} \approx 927$  steps = 0.0371 sec, iterations terminated at step 1000,  $\beta = 1$ , SNR = 0 dB,  $\Delta\tau = 4 \times 10^{-5}$ .)

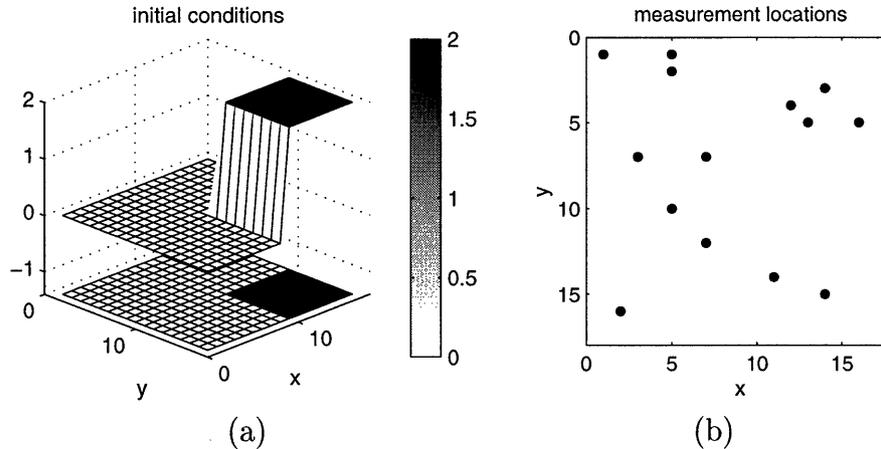


Figure 5-25: The recursive estimation of a 2-D diffusion process on a pinned cooling sheet with temporally nonstationary measurements and only one measurement update step for every 100 prediction steps. (a) The initial values of the process. (b) The measurement locations at step 900. The number of measurements at each update step has a Poisson distribution with mean 12. Estimates are shown in Figures 5-26 and 5-27.

using the reduced-order non-redundant models with interpolation. We set the non-equilibrium initial condition to the temperature distribution shown in Figure 5-25(a), where one corner of the 2-D field is set to a higher temperature. The measurements are available once every 100 prediction steps. The number of measurements at each measurement time is randomly generated as a Poisson process with a mean of 12. At step 900, for example, there are 14 point measurements whose locations are plotted in Figure 5-25(b). For convenience, the measurement locations are randomly selected from the set of elements that are represented on the reduced-order models.

Starting from the same initial conditions, we run a Kalman filter in parallel with the multiscale estimators using reduced-order non-redundant models with linear and statistical interpolation. A comparison of the updated estimates after the measurement update step at step 900 is shown in Figure 5-26. The optimal updated estimates produced by the Kalman filter is displayed in panel (a). The updated estimates from the multiscale estimators are shown in panels (c) and (e). The differences between the multiscale and the optimal estimates, shown in panels (d) and (f), exhibit some artifacts due to interpolation. The estimates from the multiscale estimator using statistical interpolation are more accurate than those from the estimator using linear

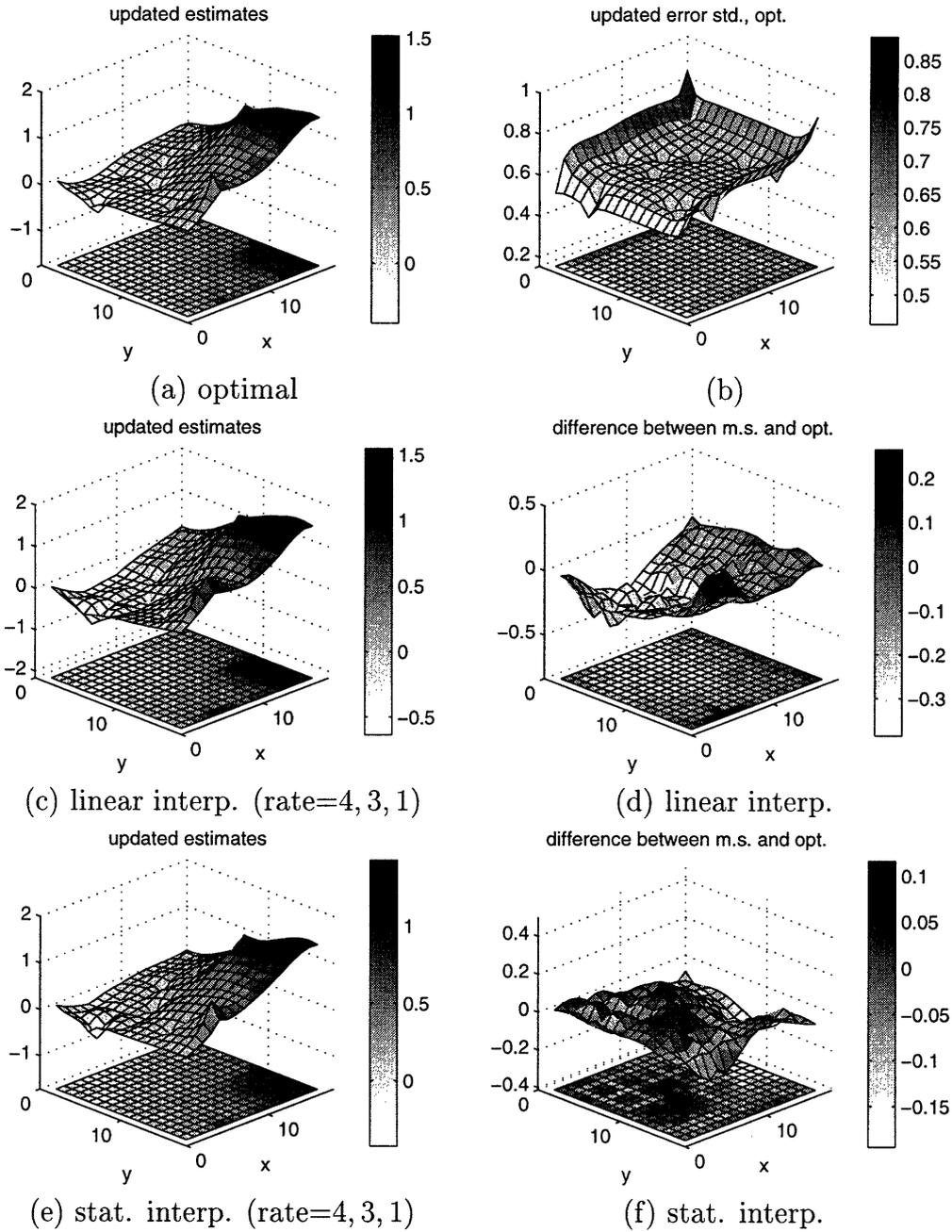


Figure 5-26: A comparison of the estimation results after the measurement update at step 900 for estimating a 2-D diffusion process on a pinned cooling sheet with temporally nonstationary measurements and one measurement update step for every 100 prediction steps. (a) Updated estimates of the optimal Kalman filter. (b) Standard deviation of the optimal updated estimation errors. (c) Updated estimates the sub-optimal multiscale estimator using a reduced-order non-redundant model with linear interpolation. (d) The different between the multiscale and the optimal estimates. (e) Updated estimates the multiscale estimator using a reduced-order model with statistical interpolation. (f) The different between the multiscale and the optimal estimates. ( $\beta = 1$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 4 \times 10^{-5}$ .)

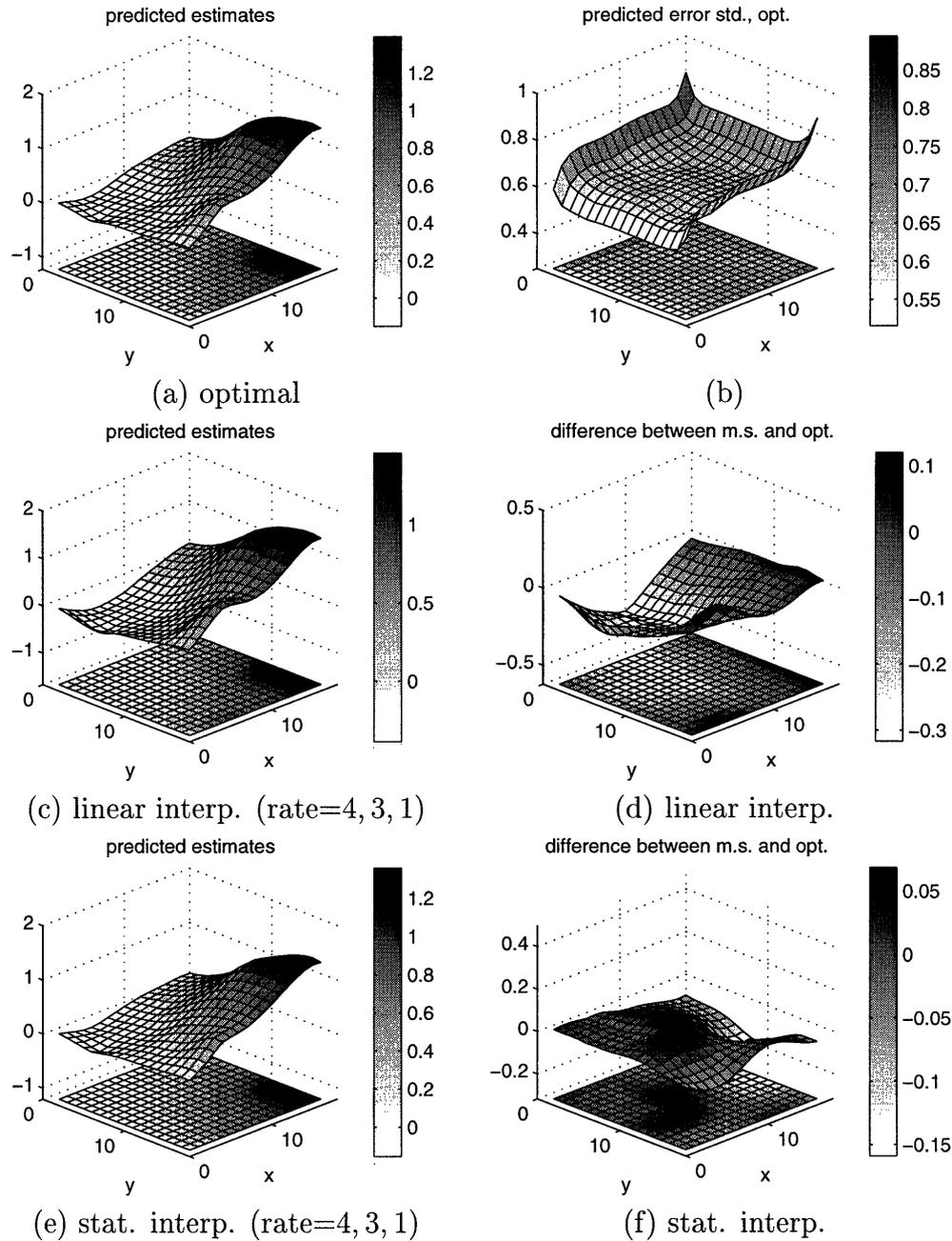


Figure 5-27: A comparison of the estimation results after prediction at step 950 for estimating a 2-D diffusion process on a pinned cooling sheet with temporally nonstationary measurements and one measurement update step for every 100 prediction steps. (a) Predicted estimates of the optimal Kalman filter. (b) Standard deviation of the optimal predicted estimation errors. (c) Predicted estimates the sub-optimal multiscale estimator using a reduced-order non-redundant model with linear interpolation. (d) The different between the multiscale and the optimal estimates. (e) Predicted estimates the multiscale estimator using a reduced-order model with statistical interpolation. (f) The different between the multiscale and the optimal estimates. ( $\beta = 1$ ,  $\text{SNR} = 0 \text{ dB}$ ,  $\Delta\tau = 4 \times 10^{-5}$ .)

interpolation as seen in panels (d) and (f), but both are well within the standard deviation of the estimation errors of the Kalman filter.

A similar comparison of the predicted estimates at step 950, which is half way between the measurement updates at steps 900 and 1000, is shown in Figure 5-27. The multiscale estimates are exactly propagated in time while the statistics of the errors are propagated approximately. The diffusion dynamics smooth the predicted estimates over time. Indeed, the estimates, as plotted in Figures 5-27(a), (c), and (e), at 50 prediction steps after the previous measurement update, are much smoother than the updated estimates. The differences between the multiscale estimates and the Kalman filter estimates are also smoother, as seen in panels (d) and (f), and are well within one standard deviation of the prediction errors of the optimal filter. The difference is again smaller for the multiscale estimator using statistical interpolation than the one using linear interpolation.

## 5.6 Concluding Remarks

In Chapter 4 we showed that for 1-D diffusion, multiscale models built for exact 1-D MRFs can accurately model the estimation errors and result in nearly optimal multiscale estimators. In this chapter, we analyzed analogous multiscale models for 2-D diffusion. The direct extension from 1-D to 2-D are full-order models designed to model 2-D MRFs exactly. The performance of multiscale estimators using such full-order models in all cases are nearly optimal. These results validate the multiscale recursive estimation algorithm discussed in Chapter 3.

The chief challenge in 2-D is the design of reduced-order models to lessen the computational cost, while maintaining a minimum level of performance. Unlike 1-D models whose state dimensions are constant, the state dimensions of full-order 2-D models grows with the size of the field of interest. This dependence increases the computational complexity from  $\mathcal{O}(N)$  in 1-D to  $\mathcal{O}(N^{3/2})$  in 2-D for the multiscale smoothing algorithm. It also rendered the multiscale prediction algorithm on internal models, still manageable at  $\mathcal{O}(N \log N)$  in 1-D, infeasible in 2-D. This leaves non-

redundant models the only known alternative.

We examined two approaches to reducing state dimensions on non-redundant models. The first approach models only subsampled boundary points as functionals and interpolates the skipped ones using either a simple linear interpolation scheme or a more sophisticated statistical interpolation method by taking advantage of our knowledge of the statistics of the field of interest. The second reduces state dimensions by extending the tree model with additional offshoot nodes of lower state dimension. Our experiments in fact show that the former approach is not only faster but also more accurate. In general, whether and how low-dimensional linear functionals can be chosen to accurately model a given random field but still allow efficient and effective multiscale prediction, remains an open research problem.

## Chapter 6

# Mapping Mediterranean Altimeter Data

Applications in oceanography provided the original motivation for this investigation into multiscale modeling and estimation of large-scale dynamic systems. Data assimilation with ocean circulation and climate models presents particularly daunting challenges to the data analysts for several reasons:

- The number of variables to be estimated, given the size of the ocean basins and the desired resolution, can easily reach hundreds of thousands if not hundreds of millions, and is too large for conventional statistical estimation methods.
- Precise knowledge of the statistical structure of the ocean is lacking. For data analysis purposes, the oceanographers often assume a simple Gaussian function for spatial as well as temporal correlations.
- The dynamic models of the ocean derived from the underlying physics are immensely complicated. However, statistical methods for data assimilation in this area is so lacking that even the assumption of a simple dynamic model such as diffusion-advection represents a significant step forward.

Multiscale models and estimation algorithms have previously been employed with great success in oceanographic applications —  $1/f$  models in mapping altimeter

data in the North Pacific Ocean [33], and internal models in mapping hydrographic data [73]. The principal advantages of the multiresolution scheme are its high computational efficiency, the specification of explicit statistical models for the oceanographic signal and the measurement errors, and the production of error variances for all estimates at multiple scales.

This chapter details the application of a modified  $1/f$  multiscale model for mapping the sea level anomaly of the Mediterranean Sea based on TOPEX/POSEIDON and ERS-1 data. Similar to [33], the oceanographic signal is modeled as a stationary  $1/k^\mu$  process, where  $k$  is the horizontal wavenumber<sup>1</sup>. Unlike in previous applications, measurement noise is modeled as the sum of two separate random processes: a Gaussian white noise process, and a correlated process of low wavenumber representing the uncertainties in the orbital position of the satellite and in the atmospheric load corrections. The correlated measurement noise process is augmented onto the multiscale model for the oceanographic signal, and is jointly estimated along with the anomaly signal. Section 6.2 details the specification of the multiscale model employed. The efficiency of the multiscale scheme allowed the testing of more than 16,000 sets of hypothesized statistical prior model parameters in order to determine the most likely parameters. Mapping results with and without low-wavenumber error corrections are presented and compared in Section 6.3.

The issue of temporal dynamics is sidestepped by assuming the ocean to be static for the duration of one repeat cycle of the satellite measurements. In [33], temporal dynamics are taken into account by adjusting the measurement quality, i.e., measurements further from the point in time of interest is assumed to be of a higher uncertainty. While we are addressing data fusion with a spatial model of the ocean here, fusing data with a spatial model as well as a temporal dynamics model of the ocean remains a goal for further research.

---

<sup>1</sup>In oceanography literature, the spatial frequency is denoted  $k$  and referred to as wavenumber, rather than  $f$  as we have used in the previous chapters.

## 6.1 Altimetric Data Assimilation

The study of ocean circulation had traditionally been hampered by the lack of data. Significant progress has been made in the last ten years thanks in large part to the accurate satellite data such as those provided by the joint U.S.-French altimetric mission TOPEX/POSEIDON and the European Space Agency ERS-1 satellites. Measurement data from TOPEX/POSEIDON and ERS-1 exhibit different characteristics. The repeat cycles of the two satellites are respectively 10 days and 35 days and the cross-track distances, i.e., distance between two parallel satellite tracks<sup>2</sup>, are  $2.8^\circ$  and  $0.7^\circ$ . The spatial coverage of ERS-1 is denser while the TOPEX/POSEIDON are more frequent. Further, because of poor tracking, the ERS-1 orbit is determined with an accuracy of 15 cm, whereas the TOPEX/POSEIDON orbit is accurate to 3 cm [90, 4]<sup>3</sup>.

The sea level signal represents the height of the ocean surface relative to the geoid, i.e., the gravitational equi-potential surface at sea level. As the sea level signal is measured with respect to the geoid, it contains errors in the geoid. Systematic orbit errors as well as geoid errors have been treated simply as white measurement noise since the dynamic range of sea surface height is on the order of 100 cm [33]. In our present application, oceanographers are interested in the anomaly signal, i.e., the variation of the surface level with respect to its temporal mean. The sea level anomaly signal, obtained by removing a four-year temporal mean of the sea level signal at every point is essentially free of any geoid errors, except for those due to lateral orbit errors<sup>4</sup> over steep topography. The major sources of low wavenumber uncertainty are the satellite orbit errors [71] and, in the Mediterranean, the atmospheric loading, i.e., distortion of the ocean surface due to pressure systems, not removed by the inverse

---

<sup>2</sup>An illustration of the TOPEX/POSEIDON and ERS-1 satellite tracks, gridded to the resolution of the sea level map, is shown in Figure 6-8

<sup>3</sup>The ERS-1 data used here contain corrections of the ERS-1 orbit errors based on a global minimization of TOPEX/POSEIDON-ERS-1 crossover differences [87], so its accuracy is similar to that of the TOPEX/POSEIDON. More in Section 6.3.

<sup>4</sup>Lateral orbit error refers to the following: The satellite does not repeat its orbit exactly on every pass but wanders laterally from its nominal orbit by a kilometer or so. Over steep geoid topography, this wandering translates into a sea surface elevation error because the sea surface reproduces geoid gradients.

barometer correction [12, 88].

While the sea level anomaly signal is free of geoid errors, it has a much smaller dynamic range, resulting in a poor signal-to-noise ratio [64]: the sea level anomaly signal is at most 10 cm RMS in the energetic regions of the Mediterranean while the residual altimeter noise, including orbit and atmospheric load errors, is of order 5 cm RMS or more. The sea surface maps generated without correcting for the systematic orbit and atmospheric load errors, show obvious inconsistencies (c.f., Figure 6-4).

Orbit and atmospheric loading errors are removed in practice, along with some of the oceanic signal, by fitting and subtracting low order polynomials or cubic splines from each altimeter track so as to minimize track-to-track crossover differences [87, 64, 10, 91, 79, 85, 83], i.e., minimizing the difference of the surface height measurements from two tracks at the location at which the tracks cross. Measurements corrected for the low wavenumber errors are then used in some type of interpolation algorithm to produce a map of the sea surface. The problem with these methods is that they do not make use of any explicit statistical model for the low wavenumber errors. The crossover difference minimization algorithms are ad hoc. The estimation of sea-surface topography and of orbit/atmospheric load corrections are treated as two separate steps, i.e., the uncertainties in the orbit correction step are not taken into account when estimating the sea surface topography.

Joint estimation of sea level anomaly and the low wavenumber measurement errors would allow the uncertainties in the errors to be naturally reflected in the estimates and vice versa. However, conventional estimation algorithms are not feasible, as the specification of a statistical model of the sea anomaly in the form of a correlations matrix would result in an estimation problem of impossibly high dimension. On the other hand, multiscale  $1/f$  models have previously been used for mapping altimeter data in the North Pacific Ocean by [33] with great success, although substantial modifications are needed here to allow for the joint estimation of sea level anomaly and orbit/atmospheric loading errors. The approach developed here provides explicit statistical models for the sea level anomaly and for the orbit and atmospheric load errors, and produces estimation error variances for all estimates at multiple scales. The

major advantage of the multiresolution scheme is that its numerical implementation is extremely efficient, and therefore the estimation can be repeated a large number of times in order to test different statistical models and to obtain consistent results. It is flexible enough to accommodate two datasets — TOPEX/POSEIDON and ERS-1 — that have different characteristics.

## 6.2 Multiscale Modeling of the Anomaly Signals

### 6.2.1 Statistical Models of Altimeter Data

An abstract statistical description of the TOPEX/POSEIDON or ERS-1 altimeter data may be written as follows:

$$\text{Measurement equation:} \quad h(t) = \zeta(x, y, t) + v(t) \quad (6.1)$$

$$\text{Sea surface prior covariance:} \quad P(x, \bar{x}, y, \bar{y}, t, \bar{t}) = E [\zeta(x, y, t) \cdot \zeta(\bar{x}, \bar{y}, \bar{t})] \quad (6.2)$$

$$\text{Measurement error covariance:} \quad R(t, \bar{t}) = E [v(t) \cdot v(\bar{t})], \quad (6.3)$$

where  $\zeta(x, y, t)$  represents the oceanographic sea level anomaly signal taken at time  $t$  and location  $(x, y)$ , and  $h(t)$  represents the altimetric measurement of  $\zeta(x, y, t)$  subject to noise  $v(t)$  which includes residual tidal, atmospheric, and orbit errors. The sea-surface height  $\zeta$  and the measurement noise  $v$  are modeled to be zero mean (i.e., systematic signal components are assumed to have been removed), so that  $E[v(t)] = 0$ ,  $E[\zeta(x, y, t)] = 0$ .  $R(t, \bar{t})$  and  $P(x, \bar{x}, y, \bar{y}, t, \bar{t})$  represent the second-order prior statistics of  $v(t)$  and  $\zeta(x, y, t)$ , respectively. Solving for the time-varying process  $\zeta(x, y, t)$  based on the prior statistics and measurements in (6.1)–(6.3) is extremely difficult, predominantly due to the complicated ocean surface dynamics captured by  $P(x, \bar{x}, y, \bar{y}, t, \bar{t})$ . The lack of precise knowledge of  $P(x, \bar{x}, y, \bar{y}, t, \bar{t})$  compounds the problem. Instead, as a first step towards ultimately solving the dynamic estimation problem, we consider the ocean surface to be temporally static (a reasonable assumption for the large-scale, baroclinic modes of variability over periods up to ten days in

length) and statistically spatially stationary:

$$E[\zeta(x, y, t) \cdot \zeta(\bar{x}, \bar{y}, \bar{t})] = P(x - \bar{x}, y - \bar{y}). \quad (6.4)$$

Motivated by spectral analyses of the altimetric data and earlier work [93, 33, 45, 36] we model the surface  $\zeta(x, y, t)$  as a  $1/k^\mu$  process, where  $k$  is the horizontal wavenumber.

Studies [44, 90, 87] of the measurement noise suggest that the TOPEX/POSEIDON or ERS-1 noise process  $v(t)$  can be modeled as the sum of two separate random processes:

$$v(t) = v_{WGN}(t) + v_{CORR}(t). \quad (6.5)$$

On the one hand, the random measurement error of each altimeter has a very short correlation period and is here modeled by a Gaussian white noise process  $v_{WGN}(t)$ . On the other hand, the uncertainties in the orbital position of the satellite and in the atmospheric load correction are a correlated process of low wavenumber, represented here by  $v_{CORR}(t)$ .

For relatively small geographic areas such as the Mediterranean Sea, the low-wavenumber error may be modeled as a low-order polynomial, leading to the following noise model:

$$v(i, t) = v_{WGN}(t) + \sum_{j=0}^J \alpha_j(i)t^j, \quad (6.6)$$

where  $J$  represents the order of the polynomial correction, and  $\alpha_j(i)$  represents the unknown  $j$ th-order polynomial coefficient for satellite track  $i$ . We model the individual coefficients  $\alpha_j(i)$  as being uncorrelated with one another and with  $v_{WGN}$ ; that is, each track is modeled as having independent errors. Although the estimator is capable of polynomial corrections to arbitrary degree, only a first-order ( $J = 1$ ) correction is applied, i.e., to estimate bias and tilt corrections only. This is consistent with the low-order corrections applied by other authors ( $J = 0$  used in [64];  $J = 1$  used in [91]) so as to minimize the attenuation of the low-wavenumber oceanographic signal.

Earlier altimetric optimal interpolation efforts have either ignored  $v_{CORR}$  [33] or

tried to remove it via a preprocessing step [64], i.e., choosing the polynomial correction coefficients that minimize differences near satellite track cross-overs. This latter approach requires ad-hoc assumptions and fails to take into account the posterior error statistics of the preprocessing step when estimating  $\zeta(x, y, t)$ . In contrast, the multiscale models, to be described next, is flexible enough not only to model the ocean surface but also to specify the noise model (6.5) explicitly.

## 6.2.2 Multiscale Models

In principle, the optimal interpolation (minimum variance) solution for the above estimation problem could be obtained by matrix inversion within the conventional linear least squares estimation framework. However, the fields computed in Section 6.3 are  $100 \times 225$  pixels in size and directly estimating such fields would require the inversion of a  $22,500 \times 22,500$  matrix. In practice, such an approach is computationally infeasible. Multiscale external models, as described in Section 2.3.5 have been applied successfully in a number of large estimation problems including mapping the sea surface height in the Pacific Ocean [33], with a modest amount of computational effort, while still computing estimation error statistics. The significant challenge here is to develop multiscale models which capture the desired statistics of not only the sea level anomaly signal but also the systematic portion of the measurement errors. The correlated noise term in (6.5) and (6.6), which was ignored in the model of [33], required a nontrivial effort to model. The resulting model will be capable of *jointly* computing the sea surface and the orbit/atmospheric load errors.

The ocean surface of interest is modeled as the finest scale of a multiscale stochastic process  $\mathbf{z}(s)$  defined on a quad-tree, i.e., a tree on which each node has four descendents (except those nodes on the finest scale). The process  $\mathbf{z}(s)$  and its associated measurement process  $\mathbf{h}(s)$  are modeled as

$$\mathbf{z}(s) = \mathbf{A}(s)\mathbf{z}(s\bar{\gamma}) + \mathbf{B}(s)\mathbf{w}(s), \quad (6.7)$$

$$\mathbf{h}(s) = \mathbf{C}(s)\mathbf{z}(s) + \mathbf{v}(s), \quad (6.8)$$

where  $E[\mathbf{z}(0)] = 0$ ,  $\mathbf{P}(0) = E[\mathbf{z}(0)\mathbf{z}^T(0)]$ ,  $E[\mathbf{v}(s)] = 0$ ,  $\mathbf{R}(s) = E[\mathbf{v}(s)\mathbf{v}^T(s)]$ , and  $\mathbf{w}(s)$  is a unit-variance white noise process. For our altimetric problem, measurements exist only at the finest scale of the tree. The multiscale state  $\mathbf{z}(s)$  is made up of bias and tilt parameters for each track, along with a multiresolution process  $\zeta(s)$  which represents the ocean surface  $\zeta(x, y, t)$ :

$$\mathbf{z}(s) = [\zeta(s) \ \alpha_0(1) \ \alpha_1(1) \ \dots \ \alpha_0(M) \ \alpha_1(M)]^T, \quad (6.9)$$

where  $M$  counts the number of satellite tracks present in the measurements and where we have assumed that  $J = 1$ , i.e., a first-order polynomial correction for the low-wavenumber errors. As in [33, 36],  $\zeta(s)$  is a multiresolution description of the sea surface height: on fine scales of the tree  $\zeta(s)$  captures the local sea surface height and on coarse scales  $\zeta(s)$  captures broad averages. The actual software implementation uses a reduced order  $\mathbf{z}(s)$ . In particular, only those  $\alpha_j(i)$  corresponding to tracks which are measured by children of  $s$  need to be kept in  $\mathbf{z}(s)$ . However, for clarity we shall retain the state definition of (6.9) throughout this chapter.

Given this definition of the state, (6.9), three quantities must be specified to complete the multiscale model: (i) a prior covariance,  $\mathbf{P}(0)$ , at the root node of the tree consistent with  $E[\alpha_j(i)^2]$ ; (ii) a multiscale state model,  $\mathbf{A}(s)$  and  $\mathbf{B}(s)$ , consistent with (6.9) and the  $1/k^\mu$  prior on  $\zeta(s)$ ; and (iii) a multiscale measurement model,  $\mathbf{C}(s)$  and  $\mathbf{R}(s)$ , consistent with (6.1), (6.6), and  $E[v_{wGN}(t)^2]$ .

The prior model  $\mathbf{P}(0)$  can be written

$$\mathbf{P}(0) = \text{diag} \left( E[\zeta(0)^2] \ E[\alpha_0(1)^2] \ E[\alpha_1(1)^2] \ \dots \ E[\alpha_1(M)^2] \right), \quad (6.10)$$

where  $E[\zeta(0)^2]$  describes the prior sea level anomaly variance at the coarsest scale of the multiresolution tree.

For the sea surface  $\zeta(s)$  we use the  $1/k^\mu$  multiscale model of [33]: the scale-to-scale variations in  $\zeta(s)$  are random with exponentially decreasing variance. The ocean surface has been observed to have  $1/k^\mu$ -like power spectrum [45]. As explained in [33, 36], phenomena with  $1/k^\mu$ -like spectra display self-similar scaling properties

such that the variability of such a phenomena scales geometrically with the spatial resolution. Such scaling rules are captured in the external models through a scaling relationship in the gain  $B(s)$  for the multiscale process noise. That is,

$$\zeta(s) = \zeta(s\bar{\gamma}) + B_o 2^{(1-\mu)m(s)/2} w(s), \quad (6.11)$$

where  $m(s)$  denotes the scale of node  $s$  on the multiscale tree. The particular values of  $B_o$ ,  $\mu$ , and  $E[\zeta(0)^2] = 2B_o^2$  are determined through a set of multiscale likelihood tests (described in Section 2.2.3). The numbers are shown in Section 6.3. The model for  $\mathbf{z}(s)$  due to the  $\alpha_j(i)$  is simple: the coefficients  $\alpha_j(i)$  are just copied unchanged from parent to child. The resulting multiscale model with the combined state of  $\zeta(s)$  and the  $\alpha_j(i)$  is then

$$\mathbf{A}(s) = \mathbf{I}, \quad (6.12)$$

$$\mathbf{B}(s) = \text{diag} \left( B_o 2^{(1-\mu)m(s)/2} \ 0 \ 0 \ \dots \ 0 \right). \quad (6.13)$$

While the prior models for the sea level anomaly and for the correlated measurement noise are specified separately as uncorrelated random processes. The coupling between the two, i.e., joint estimation, comes through the measurement equations. The multiscale measurements at the finest scale of the tree, consistent with (6.1) and (6.6), are defined:

$$\mathbf{C}(s)\mathbf{z}(s) = \begin{bmatrix} \zeta(x, y, t_1) + \alpha_0(i_1) + \alpha_1(i_1) \cdot (l(s) - 37^\circ) \\ \vdots \\ \zeta(x, y, t_N) + \alpha_0(i_N) + \alpha_1(i_N) \cdot (l(s) - 37^\circ) \end{bmatrix}, \quad (6.14)$$

where  $N$  is the total number of tracks,  $i_1, \dots, i_N$ , which pass through the finest-scale pixel corresponding to node  $s$ . At a finest-scale pixel where measurements exist, each measurement is one row in (6.14). The  $N$  measurements are taken at times  $t_1, \dots, t_N$ , and  $l(s)$  represents the latitude in degrees at which these measurements were taken. Since the satellite measurements are taken sequentially in time and in space, the

latitude,  $l(s)$ , relative to  $37^\circ\text{N}$  is used as a convenient surrogate for time,  $t$ , in (6.6). This replacement is possible because along each track the measurement latitude is a nearly linear function of the measurement time. The correlated measurement noise in (6.5), treated as a low-order polynomial, is part of the multiscale state  $\mathbf{z}(s)$ , while the remaining measurement noise is  $\mathbf{v}(s)$ . Thus,

$$\mathbf{R}(s) = \text{diag}(E[v_{WGN}(t_1)^2] \ \dots \ E[v_{WGN}(t_N)^2]), \quad (6.15)$$

where the value assumed by each  $E[v_{WGN}(t)^2]$  along the diagonal of  $\mathbf{R}(s)$  depends upon the satellite which took the measurement.

At this point the description of the multiscale model structure is complete, and it is this model which will be used to compute the experimental results in the next section.

### 6.3 Experimental Results

To illustrate the application of the multiresolution estimator described above, we have mapped sea level anomaly TOPEX/POSEIDON and ERS-1 data in the Mediterranean. TOPEX/POSEIDON data were processed and interpolated onto a fixed 6 km equidistant alongtrack grid by [63]: they include the standard environmental, inverse barometer, and orbit error corrections. Tidal corrections are from the CSR3.0 [29] model. ERS-1 data are from the files distributed by AVISO [86]. In addition to the standard environmental corrections, the AVISO data include corrections of the ERS-1 orbit errors based on a global minimization of TOPEX/POSEIDON–ERS-1 crossover differences [87]. (This ad hoc orbit correction is not necessary since the multiresolution estimation framework can accommodate different bias and tilt priors,  $E[\alpha_j(i)^2]$ , for each altimeter track. Nevertheless, sufficient orbit/atmospheric load errors remain to demonstrate the power of the multiresolution estimator.) A four-year (one-year for the ERS-1 altimeter) mean sea level is computed and subtracted from the data in order to obtain sea level anomaly.

The following subsections compare and contrast sea level anomaly maps of the Mediterranean Sea produced using several different methods: a) simple interpolation, b) multiscale estimation without polynomial corrections as in [33], c) multiscale estimation with first-order polynomial corrections as discussed in Section 6.2, and d) a suboptimal space-time optimal interpolation algorithm. Each map is computed from the measurements of one 10-day TOPEX/POSEIDON repeat cycle; in particular, we used TOPEX data from repeat cycle 24 (May 9–19, 1993) which exhibited significant correlated measurement errors and visually demonstrates the removal of residual orbit/atmospheric load errors. Finally, we show results from a joint TOPEX/ERS-1 estimation.

In each case, the anomaly map is a grid of  $100 \times 225$  estimates. The multiscale estimates are computed on a tree having 9 scales, which corresponds to a finest scale of  $256 \times 256$  pixels in size; each pixel is square, 23km on a side. Our algorithm requires that the measurements be defined on the finest scale of the multiscale tree and therefore the data need to be repositioned by up to  $\pm 1/2$  pixel width. For this reason the pixel size must be chosen sufficiently fine for this repositioning to be trivial. The associated map is then just a particular  $100 \times 225$  pixel subset of the finest scale. Prior statistics were obtained empirically from TOPEX/POSEIDON repeat cycles 3 to 161 (October 1992 to February 1997) and from ERS-1 repeat cycles 6 to 18 (October 1992 to January 1994).

### 6.3.1 Prior Statistics

As mentioned earlier, the efficiency of the multiresolution estimator permits the testing of a large number of hypothesized statistical prior models. We conducted a series of tests to determine the most likely prior model parameters [68, 35]. (In all, more than 16,000 tests, one test per hypothesized parameter setting, of the Mediterranean sea level anomaly were created to obtain the results summarized below; each test required about one tenth the computational effort of the estimates which follow.) Figure 6-1 displays the optimal value for the white measurement error variance,  $E[v_{wGN}(t)^2]$ , for individual repeat cycles of each of the three altimeters. These values

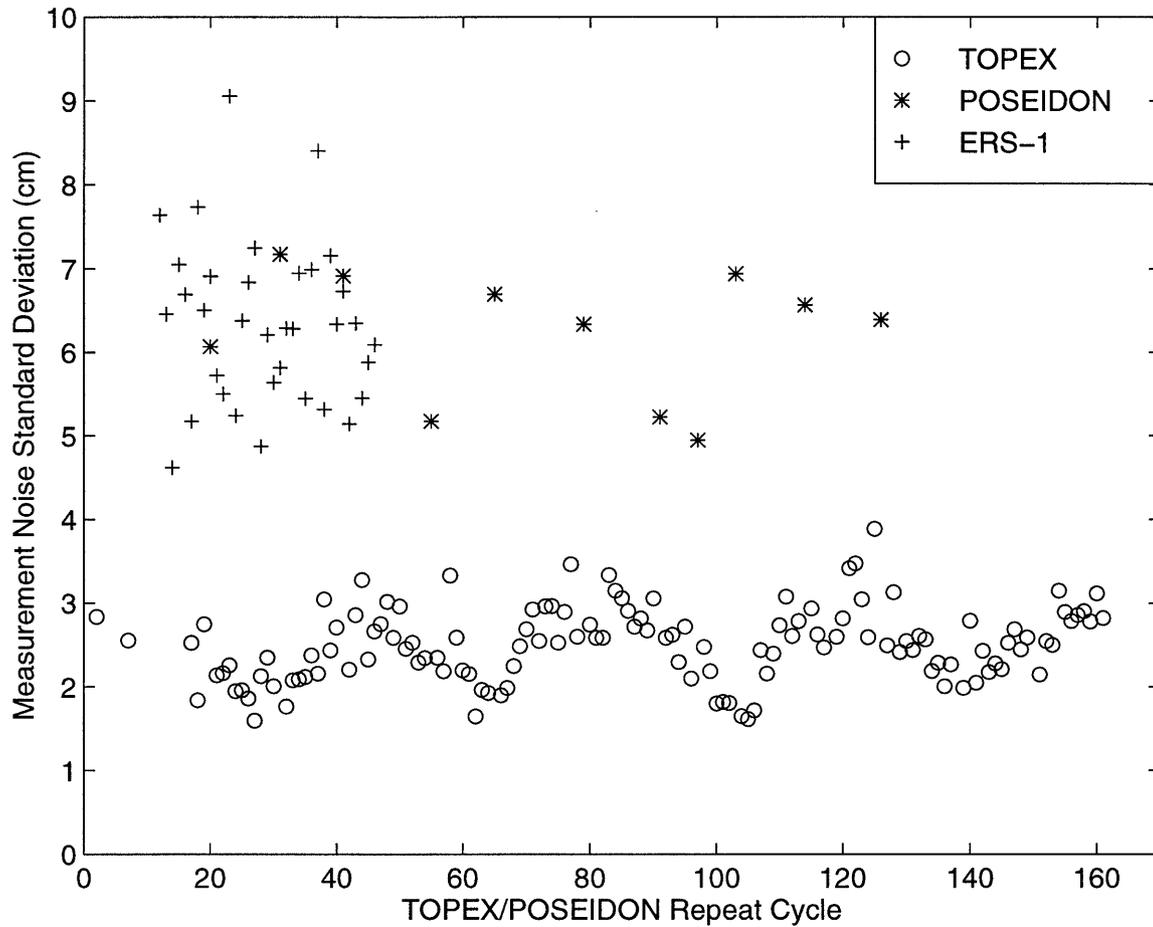


Figure 6-1: Optimal standard deviation for the measurement noise of the TOPEX, POSEIDON, and ERS-1 altimeters.

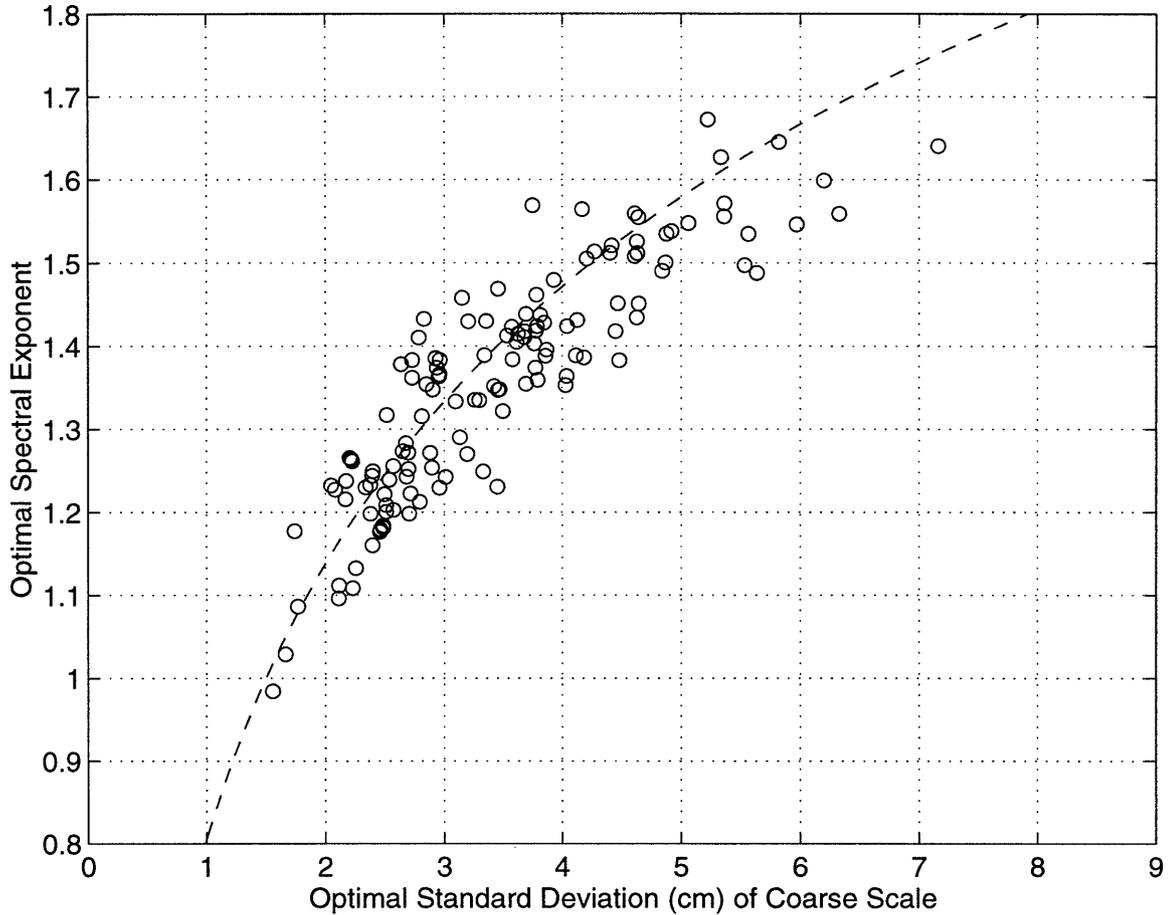


Figure 6-2: Optimal oceanographic model parameters of the measured sea level anomaly signal for TOPEX data in repeat cycles 18 through 161. The horizontal axis plots  $B_o$ , the standard deviation of  $\zeta(s)$  at the coarsest scale, and the vertical axis plots the spectral exponent,  $\mu$  in (6.13). The dashed line plots  $B_o 2^{(1-\mu) \cdot 3} = \text{constant}$ ; the distribution of points *along* and *across* the curve reflect coarse-scale and fine-scale variability, respectively.

are determined by computing the statistical likelihood of the satellite data and the multiscale prior model for a variety of choices of  $E[v_{wGN}(t)^2]$ ; the optimal choice is the one corresponding to the greatest likelihood. The results are unambiguous: they suggest measurement noise standard deviation values of 3 cm, 6 cm, and 6.5 cm for the TOPEX, POSEIDON, and ERS-1 altimeters, respectively.

Figure 6-2 displays the results of jointly optimizing the model parameters  $B_o$  and  $\mu$  in (6.13), based on the measurement noise values established for TOPEX in the previous paragraph. (The data are distributed along the exponential  $B_o 2^{(1-\mu) \cdot 3} =$

Map	RMS of Estimate (cm)	RMS of Std. Error (cm)
Map 1: $B_o = 2.5$ , $\mu = 1.24$	4.771	3.024
Map 2: $B_o = 3.5$ , $\mu = 1.40$	4.932	3.142
Map 3: $B_o = 4.5$ , $\mu = 1.53$	5.014	3.183
(Map 3 – Map 1)	0.344	0.277

Table 6.1: Sensitivity of the multiscale estimates and error statistics to the multiscale model parameters for TOPEX repeat cycle 24. These figures imply an estimate sensitivity of 3.5% and a standard error sensitivity of 4.4%, per unit of  $B_o$ .

constant, shown as a dashed line in Figure 6-2. The distribution of points *away* from the dashed line reflects fine-scale statistical variability while the distribution *along* the line reflects coarse-scale variability. Essentially, the multiscale decomposition allows fine-scale statistics to be determined in great detail, whereas coarse-scale statistics are subject to larger uncertainty.) Despite this spread, Figure 6-2 suggests that  $B_o = 3.5$  cm and  $\mu = 1.4$  constitute a reasonable choice as prior model parameters. Table 6.1 shows the results of a sensitivity test for this multiscale model: the estimates and error statistics are not particularly sensitive to the choice of model, varying by 3.5% and 4.4%, respectively, per 1 cm change in  $B_o$  and corresponding change in  $\mu$  (about 0.15) along the dashed line plotted in Figure 6-2.

Figure 6-3 displays the optimal bias and tilt coefficient standard deviations, with the prior model and measurement noise parameters set as determined in the above two paragraphs. The most striking feature is the seasonal modulation of the optimized values. During the summer months, the bias is on the order of 2.5 cm, but it increases to 4–8 cm during the winter. This seasonal modulation is consistent with the hypothesis that the low-wavenumber error budget in the Mediterranean is dominated by relatively small orbit errors during the summer months and by larger atmospheric load and other environmental correction errors during the winter months when more and stronger storm systems move through the region. The tilt standard deviations demonstrate a similar, although less pronounced, variation, going from

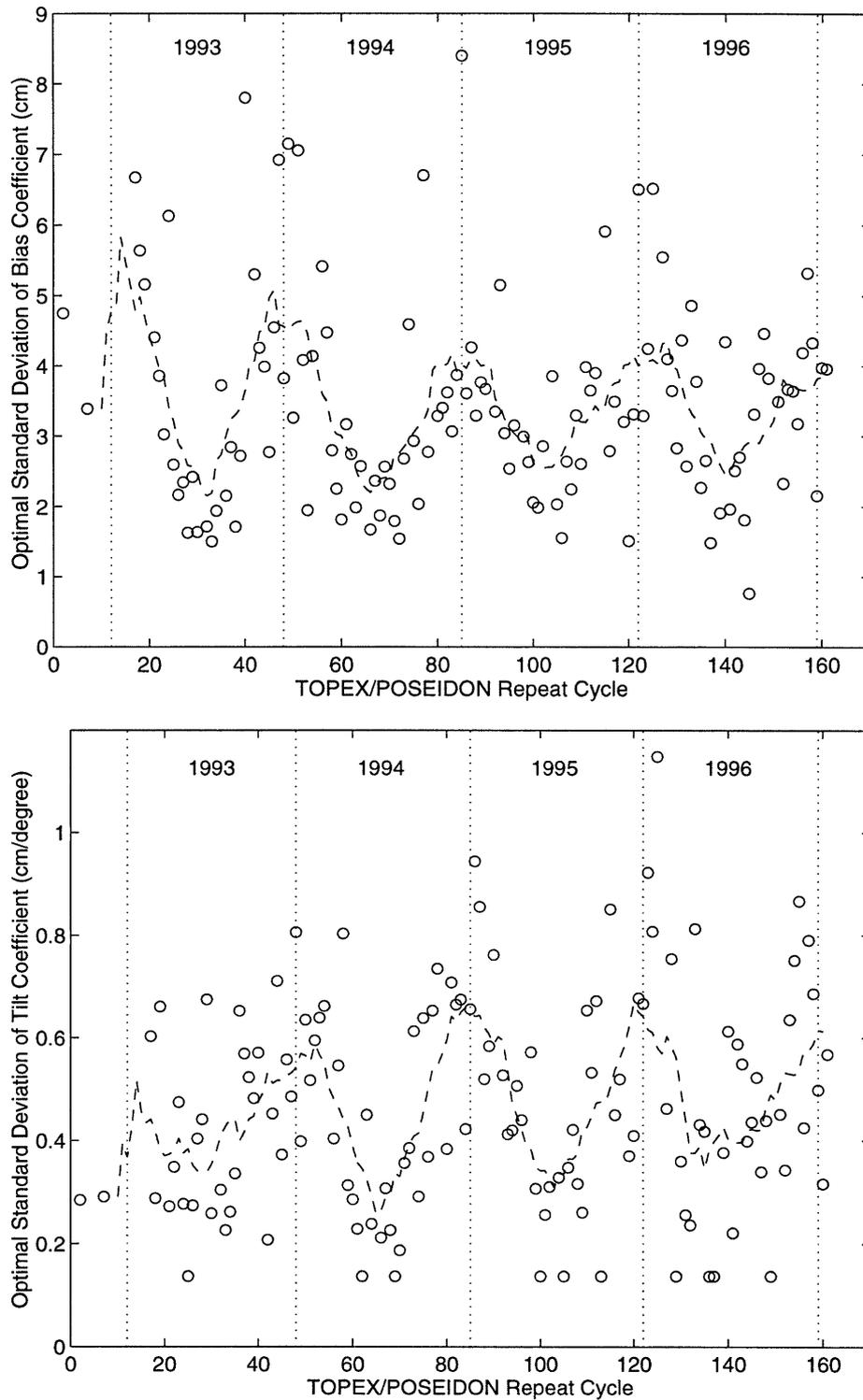


Figure 6-3: Optimal standard deviation of the bias and tilt coefficients,  $(\alpha_0(i))$  and  $(\alpha_1(i))$  in (6.6), for each TOPEX repeat cycle. The dashed lines show four-month averages.

TOPEX measurement noise variance, $E [v_{WGN}(t)^2]$ :	$(3 \text{ cm})^2$
ERS-1 measurement noise variance, $E [v_{WGN}(t)^2]$ :	$(6.5 \text{ cm})^2$
Bias variance, $E [\alpha_0(i)^2]$ :	$(4 \text{ cm})^2$
Tilt variance, $E [\alpha_1(i)^2]$ :	$(0.4 \text{ cm}/^\circ \text{ latitude})^2$
Spectral slope, $\mu$ , in $1/k^\mu$ model:	1.4
Coarse scale standard deviation, $B_o$ :	3.5 cm
Mean sea level anomaly variance, $E [\zeta(0)^2]$ :	$25 \text{ cm}^2$

Table 6.2: Prior model parameters for multiresolution estimates of TOPEX repeat cycle 24.

about 0.3 (cm/ $^\circ$  latitude) in the summer months to about 0.7 (cm/ $^\circ$  latitude) in the winter months. The actual bias and tilt statistics chosen will clearly depend on the repeat cycle being studied.

Table 6.2 summarizes the prior statistics which we used in the multiscale processing of data from repeat cycle 24; the various results are described and contrasted in the following subsections.

### 6.3.2 Basic Interpolation

Figure 6-4 displays the coverage of the TOPEX altimeter and a surface height anomaly map obtained using a simple interpolation scheme [9, Section 12.3] that iteratively applies an FFT to fill the gaps between the TOPEX altimeter data locations. The points in the gaps are first approximated with the average, i.e., the D.C. component, of the available data. An FFT is applied, followed by an inverse fast Fourier transform (IFFT) on the first two Fourier coefficients. The results, which contains only the lowest two frequencies, are used to fill the data gaps. The procedure is repeated with one more Fourier coefficient kept for the IFFT at each iteration until all frequencies in the FFT are added to the data gap. This approach does not attempt to remove correlated measurement errors, nor does it produce any sort of estimation error statistics.

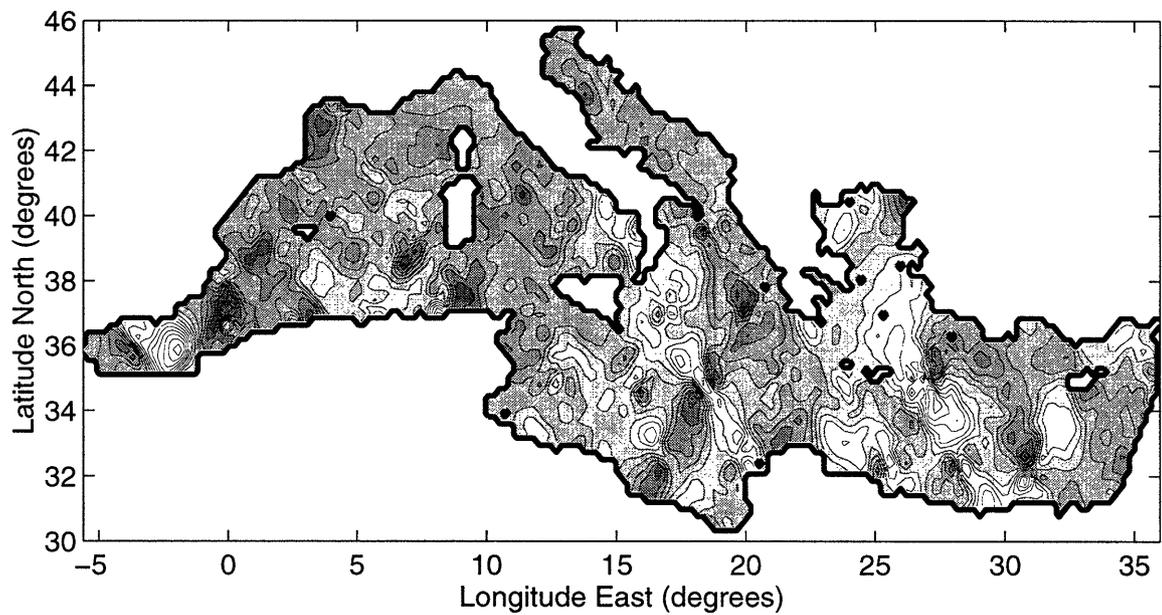
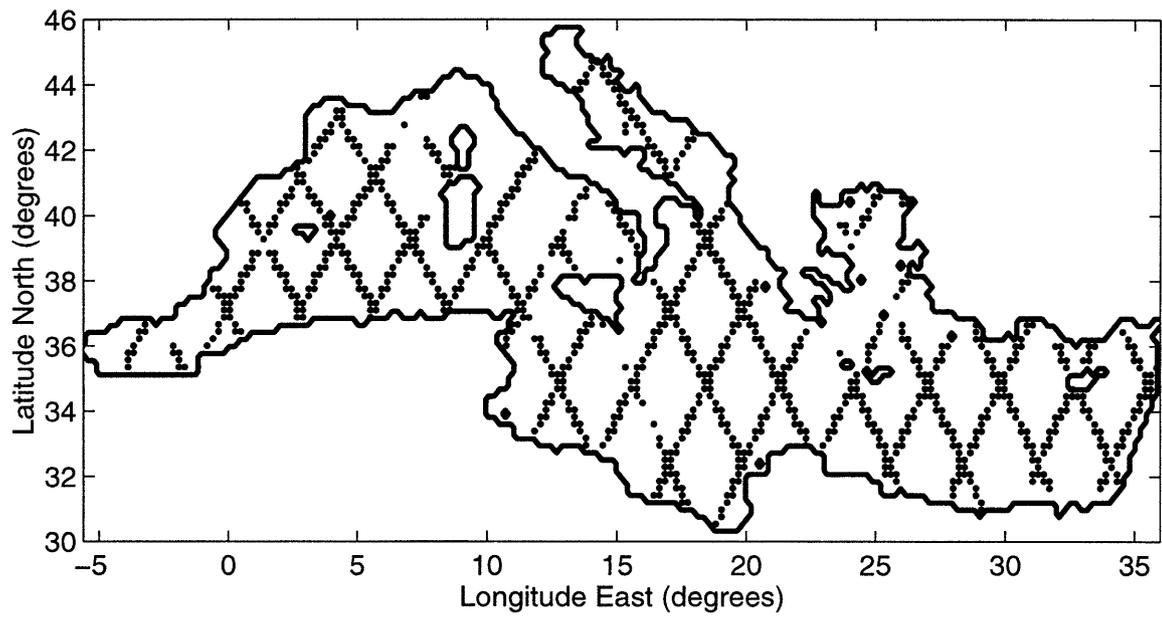


Figure 6-4: Top: Mediterranean coverage of the TOPEX altimeter for the period May 9–19, 1993 (repeat cycle 24), with each data point gridded to the finest scale of the multiscale tree. Bottom: sea level anomaly map for this period created using a simple interpolation scheme. Contour intervals are 3 cm.

### 6.3.3 Multiscale — No Correlated Error Model

Figure 6-5 shows the surface height estimates and associated uncertainties based on a multiscale model as in [33]: the multiscale model is identical to that described in Section 6.2 but does not take into account correlated measurement errors, i.e.,  $v(i, t) = v_{wGM}(t)$  in (6.6). The maps in Figure 6-5 are based on the same TOPEX data as in Figure 6-4. The results are computed in 50 seconds on a Sun SPARC-10.

On scales in excess of 500 km, the multiscale estimates and those of the interpolated map in Figure 6-4 are similar. However, noticeable differences appear on shorter scales. In particular, the interpolated map suffers from a very large variability at fine scales; this variability is considerably ameliorated in the multiscale results of Figure 6-5 because the prior model constrains the ocean surface estimates. However, both maps suffer from obvious flaws in the vicinity of altimeter tracks due to low-wavenumber orbit and atmospheric load errors. In addition to the improved quality in the estimates, the point that distinguishes this multiscale model from the simple interpolation scheme is the ability to compute estimation error statistics. Qualitatively, the estimated uncertainty map in the bottom half of Figure 6-4 is not surprising: the uncertainty is lowest near the satellite measurements and increases as one moves away.

### 6.3.4 Multiscale — Correlated Error Model

Figure 6-6 shows the surface height estimates and associated uncertainties based on the multiscale model of Section 6.2 which models the correlated component of the satellite measurement error as a first-order polynomial. Again, the maps are produced based on the same TOPEX data as in Figure 6-4. The increase in the multiscale state dimension, i.e., the dimension of  $\mathbf{z}(s)$  in (6.9), increases the computational requirements — 4 minutes on a Sun SPARC-10.

The difference between the correlated error model and its primitive cousin, presented in the previous subsection, becomes clear when comparing the estimates of Figures 6-5 and 6-6. The obvious inconsistencies present in the estimates of Figure 6-5 along the altimeter tracks have largely disappeared in Figure 6-6. Indeed,

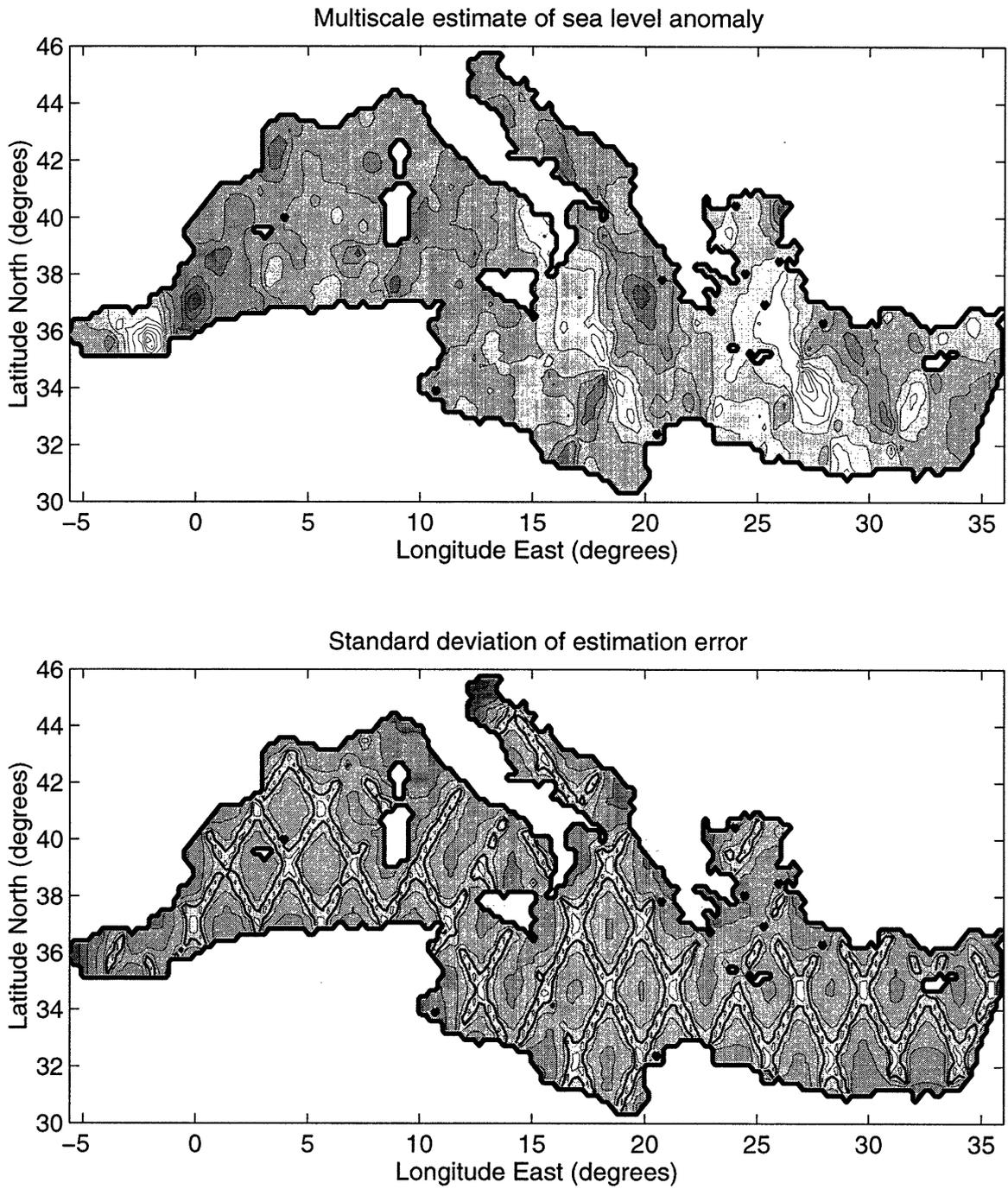


Figure 6-5: Multiscale estimates (top) and standard deviation of uncertainty (bottom) of sea level anomaly. The maps are based on the same 10-day TOPEX data as in Figure 6-4, and they do not take correlated errors into account. Contour intervals of the estimates are 3 cm. Contour intervals of the error estimates are 0.5 cm with the 2 cm contour marked in bold.

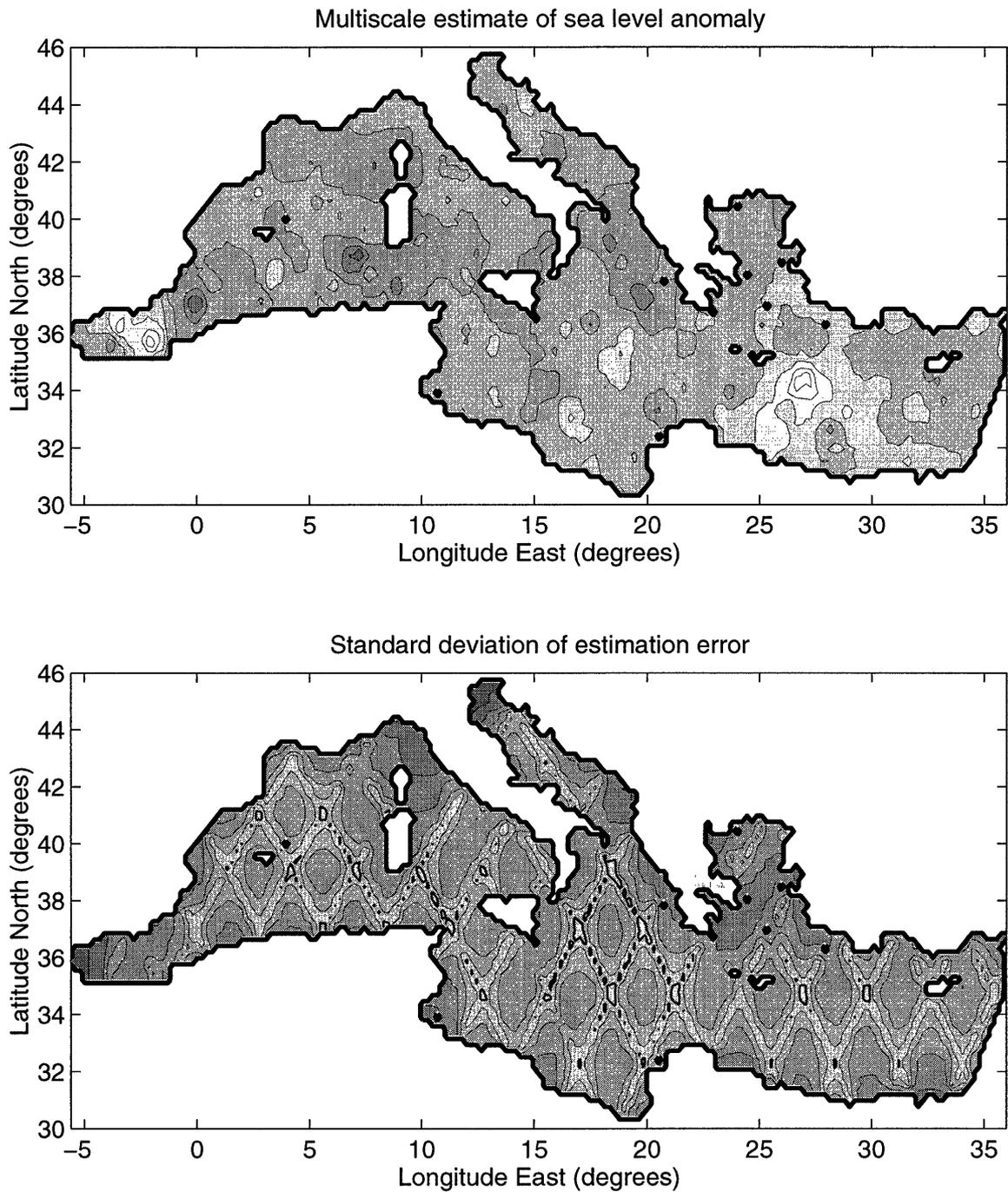


Figure 6-6: Multiscale estimates (top) and standard deviation of uncertainty (bottom) of sea level anomaly. The maps are based on the same 10-day TOPEX data as in Figures 6-4 and 6-5, but this time a first-order polynomial model has been used to remove the correlated component of measurement error along each satellite track. Contour intervals are as in Figure 6-5.

whereas the dominant features in the estimates of Figure 6-5 (and also of Figure 6-4) are light and dark lines, each due to the correlated error of a single track, the estimates of Figure 6-6 most prominently reveal broad regions of higher or lower sea surface elevation, with the satellite tracks appearing only as secondary features resulting from the higher accuracy (smaller uncertainty) of the estimates in the vicinity of the measurements.

In addition to the error statistics of the surface, shown in Figure 6-6, it should be pointed out that the error statistics of the correlated measurement error are also computed, i.e., the error statistics of the coefficients  $\alpha_j(i)$  in (6.6).

### 6.3.5 Suboptimal Space-Time Interpolation

The map and uncertainty estimates of Figure 6-6 can also be compared and contrasted to those of Figure 6-7 which were produced using suboptimal space-time interpolation [4], and which are an example of maps that have been used to study Mediterranean circulation [4]. The interpolation algorithm starts from the usual LLSE formulation

$$\mathbf{h} = \mathbf{F}\boldsymbol{\zeta} + \mathbf{v}, \quad (6.16)$$

where  $\boldsymbol{\zeta}$  is the ocean surface level at all spatial grid points over all time,  $\mathbf{h}$  is measurements over all time,  $\mathbf{F}$  is a selection matrix picking out those element of  $\boldsymbol{\zeta}$  where a measurement is made, and  $\mathbf{v}$  is the measurement noise. The linear least squares estimates and the associated estimation error covariance are

$$\hat{\boldsymbol{\zeta}} = \mathbf{P}_{\zeta,h}\mathbf{P}_h^{-1}\mathbf{h}, \quad (6.17)$$

$$\mathbf{P}_e = E[\tilde{\boldsymbol{\zeta}}\tilde{\boldsymbol{\zeta}}^T] = \mathbf{P}_\zeta - \mathbf{P}_{\zeta,h}\mathbf{P}_h^{-1}\mathbf{P}_{\zeta,h}^T, \quad (6.18)$$

where  $\mathbf{P}_{\zeta,h} = E[\boldsymbol{\zeta}\mathbf{h}^T]$ ,  $\mathbf{P}_\zeta = E[\boldsymbol{\zeta}\boldsymbol{\zeta}^T]$ ,  $\mathbf{P}_h = E[\mathbf{h}\mathbf{h}^T]$ , and  $\tilde{\boldsymbol{\zeta}} = \boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}}$ . Each entry of  $\mathbf{P}_\zeta$  is derived from an analytical expression written as the product of a polynomial spatial correlation term and a Gaussian temporal correlation term, both of which are empirically obtained [4]. Clearly, (6.17) and (6.18) are computationally infeasible

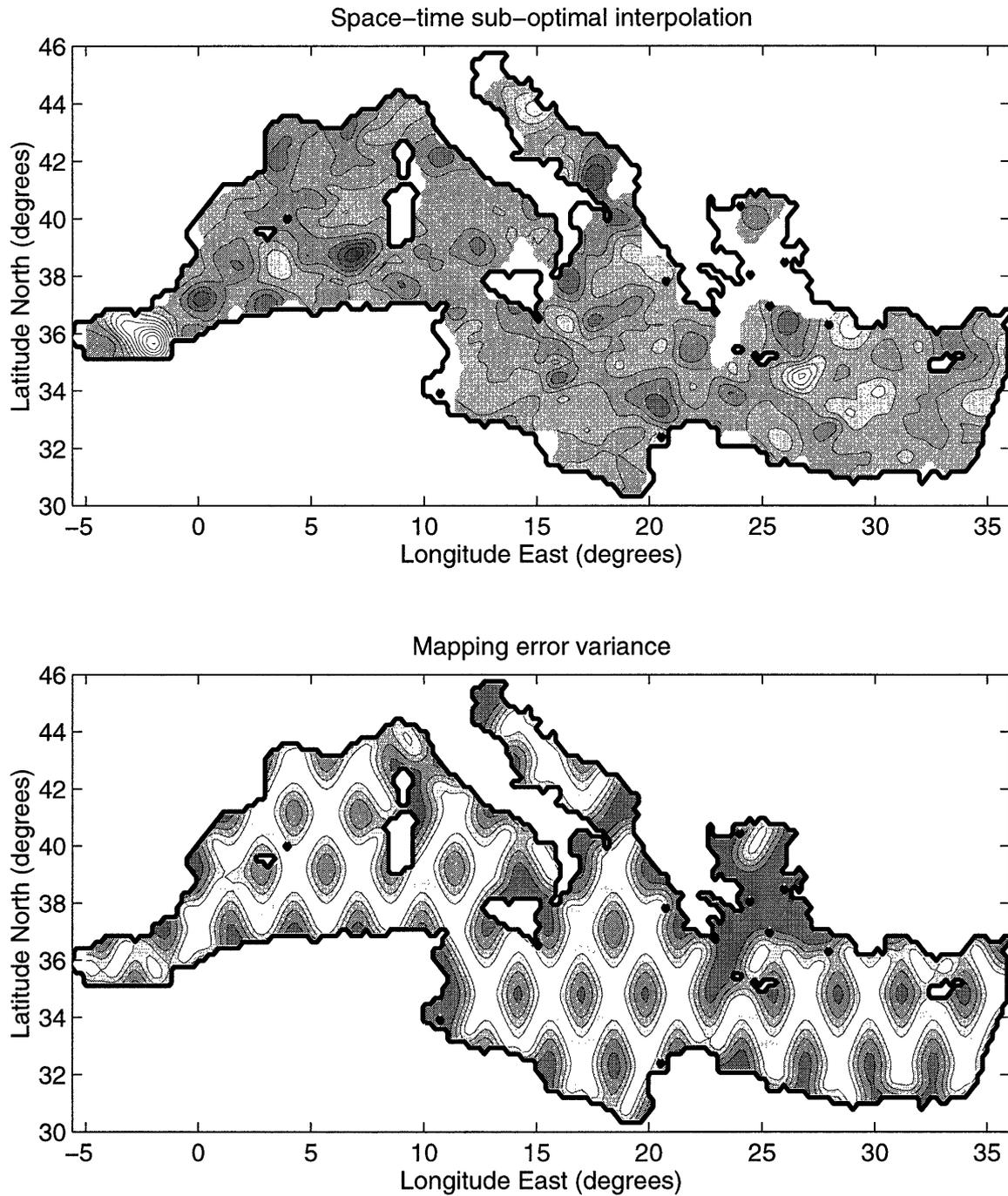


Figure 6-7: Maps of sea-level anomaly (top) and normalized mapping error variance as a percentage of the total signal variance (bottom) for a sub-optimal space-time optimal interpolation scheme. The maps are based on the same 10-day TOPEX data as in Figures 6-4, 6-5, and 6-6. Contour intervals are 3 cm for the estimates and 20% for the uncertainty. The white areas in the sea-level anomaly map (top) are values outside of the range of the contour intervals, which is the same as those in Figure 6-5. (Courtesy of Nadia Ayoub and Pierre De May.)

due the high dimensionality of both  $\zeta$  and of  $\mathbf{h}$ . The first approximation made in the interpolation algorithm avoids the matrix inversion  $\mathbf{P}_h^{-1}$  by replacing it with  $\mathbf{Q}_h$ , whose  $(i, j)$ -th element is the reciprocal of the  $(i, j)$ -th element of  $\mathbf{P}_h$  [89]. Further reduction in computation is attained by considering only those measurements that fall into a “sphere of influence” [4]. That is, each element of  $\hat{\zeta}$ ,  $\zeta(x, y, t)$ , is computed using a different subset of  $\mathbf{h}$  within a spatial decorrelation scale of 150 km and temporal decorrelation of 15 days. No correlation is assumed outside of this sphere and thus no measurements outside are used for estimation of  $\zeta(x, y, t)$ .

The qualitative difference between the multiscale and the space-time interpolation maps results primarily from the use of different statistical priors. For the multiscale maps, the ocean is modeled as the  $1/k^\mu$  process described earlier which makes allowance for correlations at multiple scales. The space-time interpolation maps were obtained assuming a spatial decorrelation scale and a temporal decorrelation scale beyond which no correlation is assumed. We have compared a series of maps for TOPEX/POSEIDON repeat cycles 2–75 created using the multiscale interpolation scheme with maps created using the space-time interpolation scheme. The conclusion is that both sets of maps are in general consistent to within the uncertainty of the multiscale estimates. The major difference between the two interpolation schemes lies in the estimates of uncertainty which are a very sensitive function of prior statistical assumptions. On the one hand, the decorrelation scales for the space-time interpolation scheme were chosen somewhat arbitrarily to produce reasonable-looking results. On the other hand, the computational efficiency of the multiscale estimates permitted a systematic search for a set of statistical parameters which are consistent with all of the available data. Realistic estimates of uncertainty are of primary importance for quantitative studies of the circulation and for blending the altimeter data with other observations or with model output.

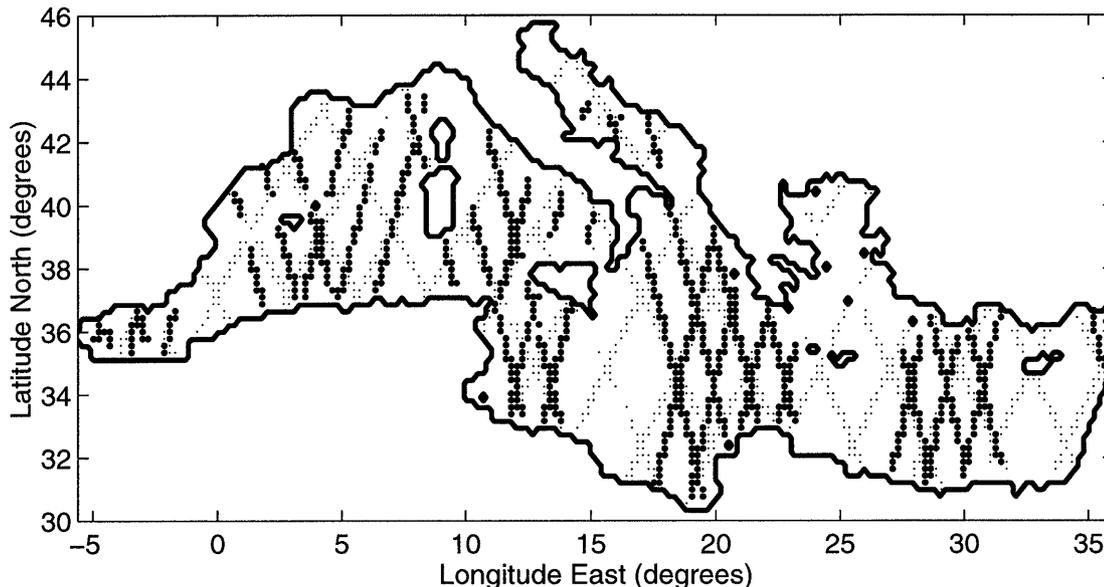


Figure 6-8: Additional coverage provided by the ERS-1 altimeter for the period May 9–19, 1993 (repeat cycle 24), concurrent with that of Figures 6-4, 6-5, 6-6, and 6-7. The small dots indicate tracks of the TOPEX altimeter while the larger dots indicate ERS-1 tracks; the dots are shown gridded to the finest scale of the multiscale tree.

### 6.3.6 Multiscale — Joint TOPEX/POSEIDON–ERS-1 Estimation

One final application will demonstrate the ability of our model to fuse TOPEX/POSEIDON and ERS-1 data. Since the ERS-1 repeat cycle spans 35 days, only the subset of ERS-1 measurements falling within the 10 day period of TOPEX/POSEIDON repeat 24 were used. The additional coverage provided by the ERS-1 altimeter during that period is illustrated in Figure 6-8. Figure 6-9 shows the resulting estimates and error statistics, computed using the correlated measurement-error model of Section 6.2. The inclusion of the ERS-1 data introduces additional tracks, hence additional  $\alpha_j(i)$  coefficients, hence a higher multiscale state dimension. Therefore estimates and error statistics are computed in about 5 minutes on a Sun SPARC-10, as compared to 4 minutes without the ERS-1 measurements. The addition of the ERS-1 data does not substantially modify the estimate of sea level anomaly.

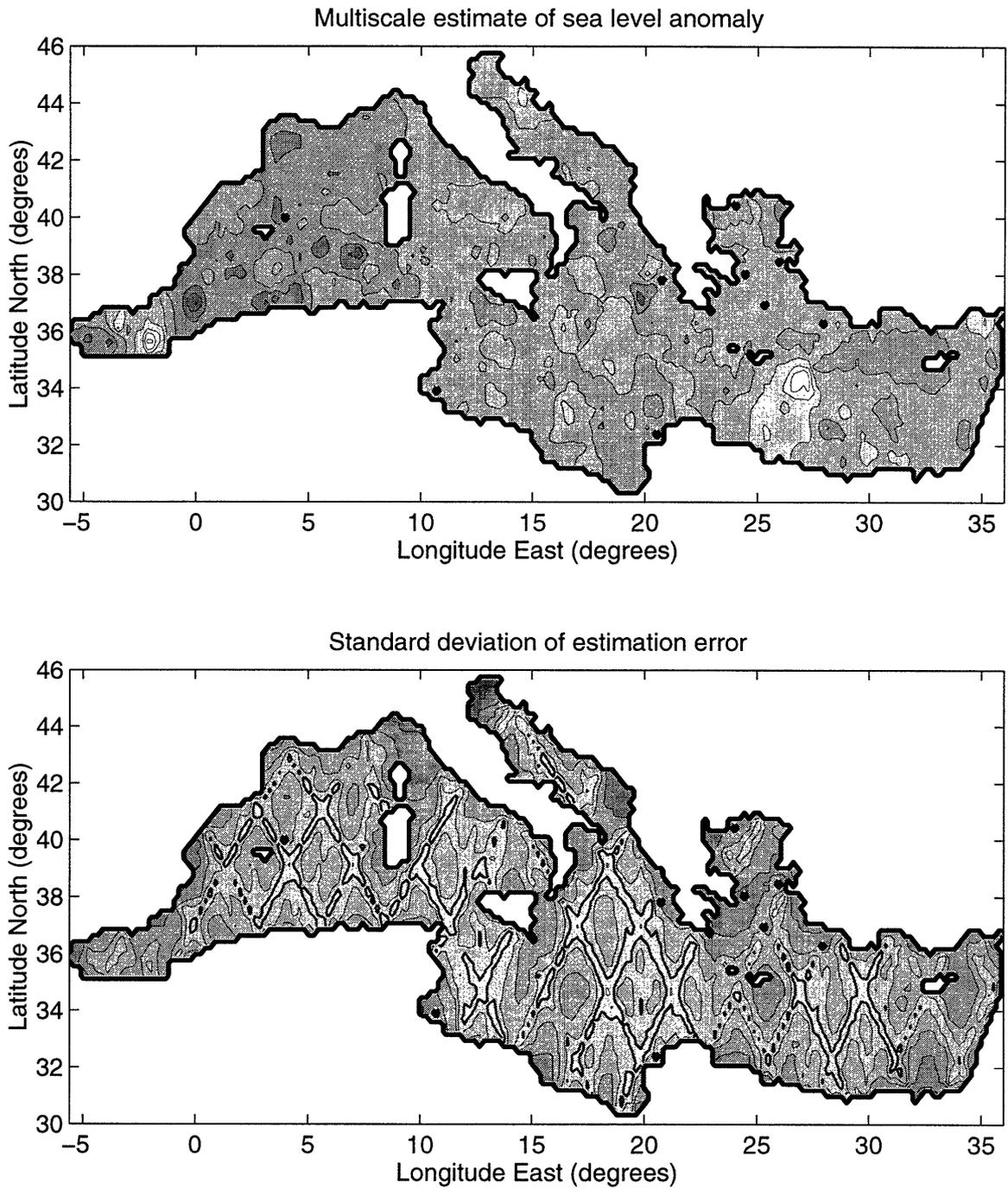


Figure 6-9: Multiscale estimates (top) and standard deviation of uncertainty (bottom) of sea level anomaly. The maps are computed from TOPEX and ERS-1 data using the same multiscale model as that used to create Figure 6-6. Contour intervals are as in Figures 6-5 and 6-6.

However, the reduction in the uncertainty is significant over most of the domain. Since the ERS-1 measurements are of poorer quality, error variation reduction is less dramatic around the ERS-1 tracks than that around the TOPEX/POSEIDON tracks.

## 6.4 Concluding Remarks

Most of the past efforts [18, 66, 33] with respect to the multiscale framework have concentrated on developing models and on understanding the statistical properties of the framework. We now believe that our understanding has improved to the point where our framework can make real scientific contributions and be used by other researchers as a tool for solving related estimation problems [73, 72].

The particular mapping example discussed in this chapter assumes a model on a square grid which is stationary,  $1/k$ -like, and static in time. The assumption of a stationary model on a square grid admittedly restricts this software to regional investigations, since at ocean basin scales the curvature of the earth is inconsistent with such a model. The use of a  $1/f$ -like prior model leads to a simple model in our multiscale framework. Other prior models, such as the subsampled internal models used in [73], are possible, but require more complicated multiscale implementations. Finally, the assumption that the ocean is static limits the time window of measurements to be used in producing a single map. However, three aspects of the multiscale method lend themselves to processing dynamic data:

1. The computational efficiency of the framework makes it practical to compute large numbers of maps, successive in time, which could then be interpolated.
2. The present framework's tolerance for nonstationarities allows the definition of a time-varying measurement noise variance. For example, (6.15) can be modified such that the diagonal elements of  $\mathbf{R}(s)$  take the form

$$E [v_{wGN}(t_i)^2] + \phi(|t_i - t_o|) \quad (6.19)$$

where  $\phi(|t_i - t_o|)$  is a positive, monotonically increasing function, and  $t_o$  is a

given time origin. Such a definition of  $\mathbf{R}(s)$  would have the effect of down-weighting measurements which lie away from the time origin  $t_o$  [36].

3. In principle an oct-tree could be constructed — multiscale in two space dimensions and one in time. This would be a very elegant solution to the nonstationarity issue. However, the development of efficient models on such trees presents formidable challenges and is the subject of ongoing research.

It must be pointed out, however, that the regional and time-stationary limitations are inherent in our particular chosen *model*, not in the multiscale framework itself. The development of nonstationary [33, 36] or temporally dynamic multiscale models is also the subject of ongoing research. The primitive development of multiscale dynamic estimation algorithms described in the previous chapters unfortunately have not yet reached a point of sophistication and maturity to effectively deal with an estimation problem as complex and as large as the ocean.



# Chapter 7

## Conclusion

### 7.1 Summary of Contributions

In this thesis we have developed a new approach to suboptimal recursive estimation of space-time dynamic systems. The point of departure for our work is 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 (MRFs), we have chosen to use multiscale models, which can lead to extremely fast algorithms for estimation. Propagating the error field model through the update step is exactly the same as deriving the *a posteriori* error model in static estimation, and is known. The major challenge in applying the multiscale recursive estimation 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.

We know from multiscale internal realization methods that constructing a multiscale internal model, i.e., computing the model parameters, requires a certain set of covariances and cross-covariances among states at nodes in the tree model. Such required statistics in the predicted estimation error model can be derived from a larger set of covariances and cross-covariances among nodes in the updated estimation error model. The mixing due to temporal dynamics determines what set of statistics in the updated error model must be computed. The cost of computing this set of statistics is also determined by the choice of linear functionals that define the states in the error models. The linear functionals must satisfy the dual objectives of not only accurately modeling the estimation error statistics for a given dynamic process and measurement configuration but also minimizing the computational cost of propagating the error model through the prediction and update steps.

We examined 1-D and 2-D diffusion processes in detail. We found that the internal models originally designed for modeling MRFs are able to accurately model the steady-state estimation errors as well as the steady-state process. However, the internal models do not facilitate a computationally efficient multiscale prediction step; that is, the required set of statistics on the updated error model is costly to compute. This difficulty leads to the development of the class of non-redundant models, in which the linear functionals are better distributed to allow more efficient calculations of the required set of statistics. The estimation results obtained indicate that near-optimal performance can be achieved using this multiscale recursive estimation procedure. The quality of the suboptimal multiscale estimators hinges mostly on the accuracy of the models for the estimation errors. Where the models are highly accurate, e.g., models for 1-D diffusion and full-order models for 2-D diffusion, the multiscale estimators are nearly optimal. The challenge lies in building lower-dimensional models that further reduce computational cost yet still accurately model the error field. The reduced-order non-redundant models with sampled boundary-point linear functionals introduced for 2-D diffusion represents one method of trading off estimator performance for computational efficiency.

While the multiscale recursive algorithm and the non-redundant models have been applied successfully to 1-D and 2-D diffusion processes, with satisfactory estimation results for the relatively small examples we have shown, the real benefits of this methodology are to be seen in truly large-scale estimation problems. One such problem is the estimation of ocean surface height in the Mediterranean Sea using satellite altimetric measurements. Following the strategy of similar problems examined before for the Pacific Ocean, we use the highly efficient and low-dimensional  $1/f$  model as a prior model for the Mediterranean sea level anomaly signal. The particular problem we addressed here is the treatment of the correlated component in the measurement noise, which is augmented as a part of the multiscale states and jointly estimated with the sea level anomaly signal. Sea level maps generated from models that take the correlated measurement noise into account are free of the obvious artifacts present in maps generated from models that do not. Temporal dynamics are taken into account by adjusting the measurement quality, i.e., measurements further from the point in time of interest are assumed to be of higher uncertainty. The use of  $1/f$  for the modeling of the sea level signal and the complexity of the ocean dynamics makes the direct application of the multiscale recursive estimation premature at this point. A number of issues regarding multiscale modeling and prediction, which we discuss next, must be addressed before such application is possible.

## 7.2 Open Problems and Future Directions

### 7.2.1 Multiscale Prediction

The multiscale recursive estimation algorithm developed in Chapter 3 takes advantage of the fact that the states in internal models can be expressed as linear functions of the fine-scale variables where the field of interest is placed. States in non-redundant models are themselves elements of the field of interest. Temporal dynamics, specified for the field of interest, can then be used to express the temporal relationship between the states in the predicted error model and states in the updated error model. The

ability to write this temporal relationship for the state variables in the error models allows us to deduce the necessary statistics for the predicted error model from statistics for the updated error model. On the other hand, in many of the oceanographic applications,  $1/f$  external multiscale models are used. One wide-open area of research that this thesis has not addressed is multiscale prediction for external models where states at the coarser nodes cannot be written as linear functionals of the field of the interest. The temporal evolution of the coarser states must be specified via other means, and must be consistent with the dynamics specified for the field of interest, usually modeled at the finest scale.

The multiscale recursive estimation algorithm assumes a fixed set of linear functionals. In the iterative implementation of the algorithm for constructing a steady-state estimator, this set of functionals is assumed to be the “correct” functionals for the steady-state prediction errors. The justification is that even though the functionals may not be the best set for the initial conditions, they are the “correct” set for the steady-state model, to which the iterative procedure will eventually converge. For the recursive implementation of the algorithm, the justification is that the same set of functionals may be robust enough to capture the time-varying error statistics or that the error statistics do not change very much over time, both of which are true for the 1-D and 2-D diffusion examples we encountered. A more sophisticated recursive algorithm can make use of time-varying linear functionals and fine-tune the accuracy of the error model for the time-varying changing error field. For example, for reduced-order internal models or non-redundant models, we may choose to sample linear functionals more densely near measurements, where more details of the statistical structure may need to be captured. If the measurement locations change over time, then the locations around which the functionals are sampled more densely should also change over time. For another example, refer to the CCR generated functionals for the steady-state 2-D diffusion process in Figures 5-1 and 5-2 and those for the steady-state prediction errors in Figure 5-5. If the recursive estimator is initialized from a steady-state process  $\mathbf{P}_z$  and converges to  $\tilde{\mathbf{P}}_p$ , it is conceivable that the linear functionals we use to model the error field in a recursive procedure should

evolve from the ones in Figures 5-1 and 5-2 to the ones in Figure 5-5.

The use of non-local linear functionals for multiscale prediction is appealing because, as we saw in Section 5.3.1, for a given limit on state dimensions, i.e., when reduced-order models are desired, properly chosen non-local functionals, e.g., by CCR, are better at capturing the conditional correlations of the field of interest and yield better estimators. Whether multiscale prediction is feasible for non-local functionals centers on if and how (3.27) can be written succinctly for non-local functionals; that is, whether the non-local functionals are almost the left eigenvectors of the dynamics matrix  $\mathbf{A}_d$ . We mentioned in Section 3.3.3 that non-local functionals of simple averages and end-point functionals are almost left eigenvectors for 1-D diffusion. It is not known, though, how multiscale prediction can be done for more general functionals, such as those computed by CCR.

In Section 3.6.1 we touched upon the issue of convergence of the multiscale recursive algorithm. We noted that there is no compelling reason to believe that the multiscale model realized from the exact steady-state predicted error covariance  $\tilde{\mathbf{P}}_p$  is the same as the steady-state model to which the iterative procedure converges. We have not experienced convergence problems for the multiscale recursive algorithm with the approximations used in the examples in this thesis. Whether the iterative procedure converges under more general conditions, under what conditions it converges, and to what steady state it converges are all important theoretical questions that need to be addressed.

## 7.2.2 Multiscale Modeling

Multiscale modeling issues and multiscale prediction issues are closely related. The invention of non-redundant models was aimed primarily at satisfying the requirements of multiscale prediction. We saw that the different ways of placing the same type of point linear functionals in internal models and in non-redundant models resulted in different computational complexity for the multiscale prediction step. The linear functionals must satisfy a host of other objectives, most of which were discussed and listed in Section 3.3.3. The challenge in multiscale modeling is one of balancing the

different and often conflicting objectives. For the diffusion processes we considered in this thesis, we have fortuitously found a happy compromise that satisfies the two most important of these objectives. The point linear functionals on non-redundant models adequately capture the statistics of estimation errors of diffusion processes and allow computationally efficient prediction. On the other hand, internal or non-redundant models, having only point linear functionals, cannot accommodate non-local measurements. For the applications we considered this is a mild restriction and thus less important to us. In Section 7.2.3 we show an example of optical flow on a 1-D contour in some detail and describe a new type of hybrid model that is neither internal nor non-redundant but has desirable features of both. Such hybrid models satisfy the three objectives of accurate statistical realization, efficient prediction, and placement of non-local measurements. In other applications, such as data fusion of disparate types of measurements with physical models, the ability to model non-local, as well as local, measurements is far more critical. State augmentation on internal models in order to accommodate non-local measurements has been studied before [25, 23], but the incorporation of non-local measurements, e.g., tomographic data that represent line integrals of spatial variables, on either  $1/f$  models that we have used for modeling the ocean or on non-redundant models, has not been examined.

There are a number of issues concerning non-redundant models that have not been fully explored or resolved. The most obvious difference between 2-D reduced-order non-redundant models and internal models is that the field of interest is not fully represented on the non-redundant models. Since the field is always placed at the finest scale of an internal model, estimates and estimation error variances are always available there. The state variables in an internal model collectively form an overcomplete basis for the field of interest. On the other hand, the state variables in a reduced-order non-redundant model forms an incomplete basis. In Section 5.3.3, we have used interpolation schemes to address this problem with some success. It is an open problem what other computationally efficient and statistically accurate methods can be devised.

We also saw in Section 5.3.2 that for internal models given a limit on state di-

dimension, the non-local functionals produced by CCR more accurately model the estimation errors than the spatially sampled boundary-point linear functionals of the same state dimensions. We may be motivated, therefore, to keep non-local linear functionals on non-redundant models too. For instance, denote by  $\zeta(s)$  the dense set of boundary points that would ordinarily be kept at node  $s$  on a full-order non-redundant model. Denote by  $\mathbf{x}(s)$  the state variables. In an ordinary full-order non-redundant model,  $\mathbf{x}(s) = \zeta(s)$ . However, we may keep non-local linear functionals as state variables such that  $\mathbf{x}(s) = \mathbf{M}(s)\zeta(s)$ , where  $\mathbf{M}(s)$  may be dense, but square and invertible. Then,  $\mathbf{x}(s)$  and  $\zeta(s)$  are equivalent via a basis transform. The set of non-local functionals,  $\mathbf{x}(s)$ , e.g., calculated and ordered by CCR according to their ability of conditional decorrelation, provides a better way of defining reduced-order states. Rather than keep a sampled subset of the boundary points, we may keep the most significant elements of  $\mathbf{x}(s)$ , in which case the reduced-order  $\mathbf{M}(s)$  is non-square, non-invertible, and dense. For such non-redundant models, the state variables  $\mathbf{x}(s)$  are no longer even selected elements of the field of interest, but some linear functions of the field of interest. Thus, no estimates and error variances of the individual elements are available, but only estimates and error variances of certain linear functions of the elements of the field of interest. One way of deducing  $\zeta(s)$  from  $\mathbf{x}(s)$  may be to compute a pseudo-inverse  $\mathbf{M}^\#(s)$ . The proposal here illustrates again the competing requirements we place on the functionals. The set of functionals that best realize the statistics of the field of interest are not the most convenient for modeling the quantities, i.e., the individual elements of the field of interest, that we either want to estimate or for which we have measurements.

We have examined only diffusion [74] and advection-diffusion problems [95, 94] in this thesis. It is of considerable interest and practical value to understand how to adapt our methodology to dynamics that allow wave-like behavior for applications in geophysical estimation problems. It has been demonstrated that although dynamic models for the ocean used by oceanographers are generally very complex, relatively

simple PDEs, such as the barotropic Rossby wave equation,

$$\frac{\partial}{\partial t} \left[ \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} \right] + \beta \frac{\partial \eta}{\partial x} = 0, \quad (7.1)$$

where  $x$  and  $y$  are the two spatial dimensions and  $\eta$  is the ocean surface height, are adequate in explaining part of the ocean surface variability [45]. There are two main issues in extending our multiscale recursive estimation methodology to the wave equation. First, we need to understand the dynamic mixing of (7.1) at each step of the multiscale recursive algorithm. For such models we would expect that the propagation of error models over time need to account for the modes of wave propagation. Second, and more importantly, is the selection of appropriate linear functionals for modeling the estimation error field, especially since there is no statistical steady state for the wave equation. The feasibility of our multiscale method for diffusion depended in large part on the ability of the linear functionals to model the error fields accurately. It remains for the future to determine whether similar functionals can be used to model the error field of the wave-like dynamics, and if not, what functionals can.

### 7.2.3 Optical Flow and Hybrid Models

Optical flow computation in computer vision applications has been referred to in several places in the thesis. The estimation theoretic interpretation of optical flow deserves more detailed mention here. More importantly, the latter part of this subsection documents some results regarding the multiscale modeling of a simple 1-D example of optical flow computation on a contour [47]. We introduce a hybrid model that is neither non-redundant nor internal, but designed to accommodate non-local measurements.

Optical flow refers to the apparent velocity vector field corresponding to the observed motion of brightness patterns in successive image frames. The starting point of optical flow computation is a *brightness constraint* [53], which assumes that the

changes in image brightness are due only to motions in the image frame:

$$0 = \frac{d}{dt}E(x, y, t) = \frac{\partial}{\partial t}E(x, y, t) + \nabla E(x, y, t) \cdot \mathbf{f}(x, y, t), \quad (7.2)$$

where  $E(x, y, t)$  is the image intensity as a function of time  $t$  and space  $(x, y)$  and  $\mathbf{f}(x, y, t)$  is the optical flow vector field, and

$$\mathbf{f} = \begin{bmatrix} \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{bmatrix}^T = [f_x \ f_y]^T \quad (7.3)$$

$$\nabla E = \begin{bmatrix} \frac{\partial E}{\partial x} & \frac{\partial E}{\partial y} \end{bmatrix}. \quad (7.4)$$

The brightness constraint of (7.2) provides only one constraint for two unknowns at each point, so it does not completely specify the flow field  $\mathbf{f}(x, y, t)$ ; that is, the problem is ill-posed. In [53] a *smoothness constraint* is introduced as a regularization term so that solving for the optical flow field is formulated as an optimization problem:

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{f}} \int \int R^{-1} \left\| \frac{\partial E}{\partial t} + \frac{\partial E}{\partial x} f_x + \frac{\partial E}{\partial y} f_y \right\|^2 + \left\| \frac{\partial}{\partial x} f_x + \frac{\partial}{\partial y} f_y \right\|^2 dx dy, \quad (7.5)$$

where the first term of the integrand corresponds to the brightness constraint and the second term is the smoothness constraint, which penalizes large gradients in the optical flow field and favors a solution with the least amount of variation. The constant  $R$ , allows a trade-off between the relative importance of the brightness constraint and the smoothness constraint.

The optimization problem of (7.5) can be interpreted as an estimation problem [66, 15, 32]. Let us rewrite (7.2) and (7.5) in vector form since in practice images are available on a discrete space-time grid. Let  $\mathbf{z}_x$  be the vector of the  $x$ -components of the optical flow vectors  $\mathbf{f}(x, y, t)$  at all  $N$  grid points,  $\mathbf{z}_y$  be the vector of the  $y$ -components, and  $\mathbf{z} = [\mathbf{z}_x^T \ \mathbf{z}_y^T]^T$ . Let  $\mathbf{g}$  be the vector of temporal gradients  $-\frac{\partial}{\partial t}E(x, y, t)$ . Let  $\mathbf{H}_x$  be the matrix whose diagonal contains the  $x$ -components of the spatial gradient terms, i.e.,  $\partial E/\partial x$ ,  $\mathbf{H}_y$  be the matrix containing the  $y$ -components,

and  $\mathbf{H} = [\mathbf{H}_x^T \ \mathbf{H}_y^T]^T$ , such that the brightness constraint of (7.2) can be written as

$$\mathbf{g} = \mathbf{H}\mathbf{z}. \quad (7.6)$$

Let  $\mathbf{M}_x$  be a discrete approximation of the gradient operator  $\partial/\partial x$  in (7.5),  $\mathbf{M}_y$  be an approximation of  $\partial/\partial y$ , and

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_y \end{bmatrix}, \quad (7.7)$$

such that (7.5) can be written as

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z}} \|\mathbf{g} - \mathbf{H}\mathbf{z}\|_{\mathbf{R}}^2 + \|\mathbf{M}\mathbf{z}\|_{\mathbf{I}}^2 \quad (7.8)$$

$$= \arg \min_{\mathbf{z}} (\mathbf{g} - \mathbf{H}\mathbf{z})^T \mathbf{R}^{-1} (\mathbf{g} - \mathbf{H}\mathbf{z}) + \mathbf{z}^T \mathbf{M}^T \mathbf{M} \mathbf{z}, \quad (7.9)$$

where  $\mathbf{R} = \mathbf{R}\mathbf{I}$ . The maximum likelihood estimate of  $\mathbf{z}$  based on measurements

$$\mathbf{g} = \mathbf{H}\mathbf{z} + \mathbf{v} \quad (7.10)$$

$$\mathbf{0} = \mathbf{M}\mathbf{z} + \mathbf{w}, \quad (7.11)$$

where  $\mathbf{v} \sim N(0, \mathbf{R})$  and  $\mathbf{w} \sim N(0, \mathbf{I})$  are uncorrelated Gaussian noise, is exactly the same as the solution of (7.9) [15, 66]. In this maximum likelihood estimation interpretation, the two sets of measurements in (7.10) and (7.11) correspond to the brightness and smoothness constraint terms in (7.9), respectively. Notice that (7.11) can be rewritten as  $-\mathbf{M}\mathbf{z} = \mathbf{w}$ , which resembles (2.27) from Section 2.3.2. The Bayesian interpretation views the smoothness constraint as a prior model for  $\mathbf{z}$ . If  $\mathbf{M}$  is invertible, it follows directly from (7.11) that  $\mathbf{z} \sim N(0, \mathbf{P}_z^{-1})$ , where  $\mathbf{P}_z = (\mathbf{M}^T \mathbf{M})^{-1}$ . The LLSE estimate of  $\mathbf{z}$ , satisfying the following normal equation (similar to (3.17) in Section 3.2.1)

$$(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{M}^T \mathbf{M}) \hat{\mathbf{z}} = \mathbf{H}^T \mathbf{R}^{-1} \mathbf{g}, \quad (7.12)$$

is again the same as the solution of (7.9).

There are a number of possible prior models that can play the role of the regularization term:

- (i) The smoothness constraint is interpreted as a statistical prior for  $\mathbf{z}$ . Since  $\mathbf{M}$  is an approximation to the gradient operator, it is non-invertible [66]. However, we can specify a prior covariance of  $\mathbf{z}$  by choosing  $\mathbf{P}_z = (\mathbf{M}^T \mathbf{M} + \epsilon \mathbf{I})^{-1}$ , where  $\epsilon$  is small.
- (ii) The smoothness constraint is interpreted similar to a MRF prior [15]. Since  $\mathbf{M}$  is an approximation to the gradient operator, it is sparse. The inverse of the prior covariance for  $\mathbf{z}$  in the Bayesian formulation,  $\mathbf{P}_z^{-1} = \mathbf{M}^T \mathbf{M}$ , is then also sparse, indicating a Markov random field structure. Although  $\mathbf{M}^T \mathbf{M}$  is singular, the issue is sidestepped because (i)  $\mathbf{P}_z^{-1}$  rather than  $\mathbf{P}_z$  is explicitly solved and (ii) with measurements the *a posteriori* estimation error covariance  $\mathbf{P}_e = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{M}^T \mathbf{M})^{-1}$  is invertible. Furthermore,  $\mathbf{H}$  from the brightness constraint is sparse, so  $\mathbf{P}_e^{-1}$  is also sparse. This is the reason why the Markov structure is preserved in the *a posteriori* estimation error, as we mentioned in Section 3.2.1.
- (iii) The smoothness constraint is interpreted as a fractal prior [66] and is specified with a multiscale  $1/f$  model. The multiscale model realizes a covariance for the optical flow field that approximates  $(\mathbf{M}^T \mathbf{M} + \epsilon \mathbf{I})^{-1}$ .

Analogous to the smoothness constraint in space in the case of single-frame optical flow computation, a smoothness constraint in time, i.e., *temporal coherence*, is imposed in the multi-frame case [15, 14]. The optimization problem of (7.5) now has an additional term  $\rho \left\| \frac{\partial}{\partial t} \mathbf{f} \right\|^2$ . Analogous to the interpretation of the smoothness constraint as (7.11), the estimation-theoretic interpretation of the temporal coherence constraint is a temporal dynamic equation  $\frac{\partial}{\partial t} \mathbf{f}(x, y, t) = \mathbf{q}(t)$ ,  $\mathbf{q}(t) \sim (0, \rho^{-1} \mathbf{I})$ , whose first-order approximation is

$$\mathbf{z}(t) = \mathbf{z}(t-1) + \mathbf{u}(t), \quad (7.13)$$

where  $\mathbf{u}(t) \sim (0, \rho^{-1}\mathbf{I})$  is of length  $2N$ . Higher-order approximations to the continuous-time temporal coherence constraint result in the much more complicated dynamics necessitating the use of descriptor systems formulation [14, 13]. However, the first-order approximation has been shown to be adequate [15] and has the particular simple dynamics, i.e.,  $\mathbf{A}_d = \mathbf{I}$  as we mentioned in Section 3.4.1.

The MRF-based recursive estimation algorithm for multi-frame optical flow computation in [15] (c.f., Section 3.2.1), imposes a MRF neighborhood structure on the estimation errors in estimating  $\mathbf{z}(t)$ . Both the brightness constraint and the smoothness constraint are interpreted as measurements at each time step. As we have seen in Section 2.3.2, MRFs can be exactly modeled on multiscale models with linear functionals consisting of end points of subintervals in 1-D or boundary points of subregions in 2-D. Thus, instead of imposing a MRF structure on the estimation errors, we may alternatively impose a multiscale model with a fixed set of linear functionals known to be suitable for realizing MRFs, as we saw in Chapter 3. The remaining question is whether the measurements corresponding to the brightness and the smoothness constraints at each time step can be easily accommodated within such multiscale models.

Consider the problem of computing optical flow on a contour [47] as a 1-D example. It is 1-D in the sense that the flow vectors are computed on a 1-D grid, but at each grid point there are still an  $x$ - and a  $y$ -component to the flow vector. We model the optical flow field along the 1-D contour on an internal model shown in Figure 7-1. The  $x$ - and  $y$ -components of the optical flow field,  $\mathbf{z}_x$  and  $\mathbf{z}_y$  are mapped to the finest scale. Each pair of closely spaced circles indicate the two components at the same spatial grid point, i.e.,  $z_{x,i}$  and  $z_{y,i}$  at location  $i$ . The state dimension at each multiscale tree node is 8.

The measurements corresponding to the brightness constraints involve only the two components of the flow vector at the same spatial location. That is, the  $i$ -th row of  $\mathbf{g}$  in (7.10) has two non-zero elements,  $z_{x,i}$  and  $z_{y,i}$ . These measurements are local and readily placeable on the multiscale tree, since the  $z_{x,i}$ ,  $z_{y,i}$  pair always appear at the same node. If we used  $1/f$  models to model the flow field in the single-frame

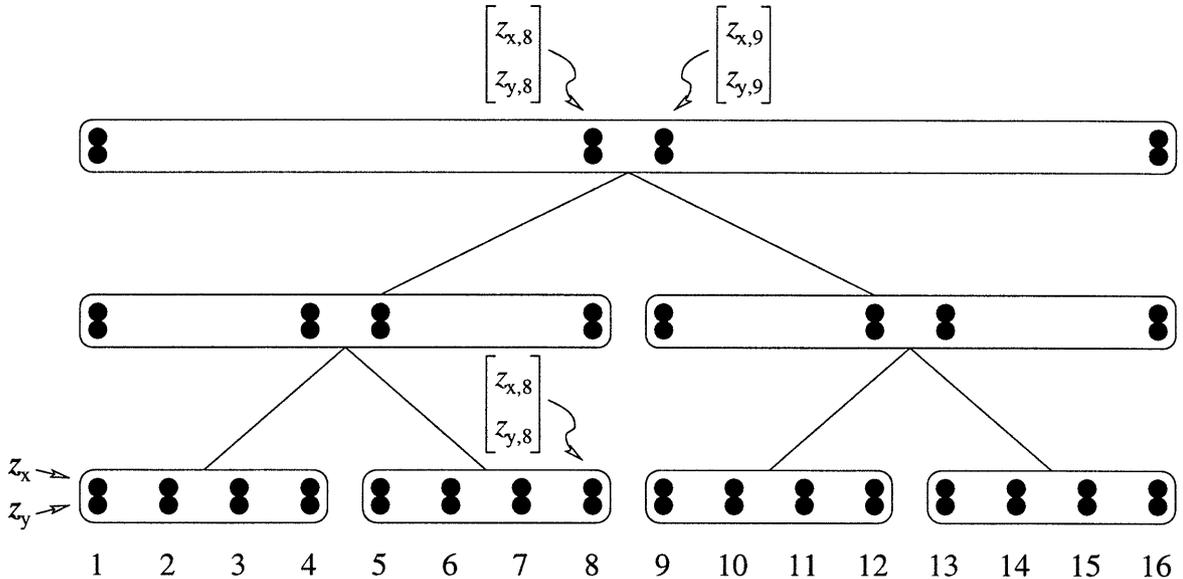


Figure 7-1: A multiscale internal model for optical flow field along a 1-D contour. The  $x$ - and  $y$ -components of the optical flow field,  $z_x$  and  $z_y$  are mapped to the finest scale. Each pair of closely spaced circles indicate the two components at the same spatial grid point. The state dimension at each multiscale tree node is 8.

case [66], we will have no trouble placing the measurements corresponding to the brightness constraint, since the state variables at the finest scale consist of exactly  $z_{x,i}$  and  $z_{y,i}$ .

The measurements corresponding to the smoothness constraints, on the other hand, involve elements of  $z_x$  at two spatially adjacent points or of  $z_y$  at two spatially adjacent points. That is, a row of (7.11) has two non-zero element,  $z_{x,i}$  and  $z_{x,i+1}$ , or  $z_{y,i}$  and  $z_{y,i+1}$ , for some  $i$ . Such non-local measurements cannot be placed on the  $1/f$  models, since the coarse-scale variables cannot be written as linear functionals of the fine-scale variables, and the non-local measurements always span two adjacent nodes at the finest scale. This problem did not exist in the single-frame case where the smoothness constraint itself was interpreted as the prior model [66], so that the issue of placing non-local measurements on the tree was avoided. Fortunately, for internal models with end-point linear functionals, a variation of which is the one in Figure 7-1, these measurements can always be placed at some coarser-scale node. For instance, the measurement  $0 = z_{x,9} - z_{x,8} + w_8$  can be placed at the root node of the tree.

Since for prediction with a tridiagonal  $\mathbf{A}_d$ , non-redundant models are computationally more efficient (as explained in Section 3.4), we may desire the use of non-redundant models. However, the non-local measurement corresponding to the smoothness constraint cannot be placed on non-redundant models. In the case of optical flow with a first-order approximation to the temporal coherence constraint, i.e.,  $\mathbf{A}_d = \mathbf{I}$ , the computational complexity of multiscale prediction is not an issue. Prediction would be an issue if a higher-order approximation is used such that  $\mathbf{A}_d$  is no longer a diagonal matrix. This simple example of 1-D optical flow illustrates that the different objectives of easy multiscale prediction and easy modeling of measurements may lead us to conflicting choices of linear functionals and multiscale models.

Fortunately, given a tridiagonal dynamics matrix  $\mathbf{A}_d$ , non-local measurements over adjacent spatial grid points, and approximate MRF correlation structure of the error process, in the case of the 1-D optical flow example here, a *hybrid* model with functionals shown in Figure 7-2 can satisfy all three objectives of easy prediction, easy measurement placement, and accurate statistical realization. In this model, elements of the process at adjacent grid points always appear at the same node, permitting easy placement of measurements. The model is capable of exactly realizing second-order 1-D MRFs, although the need for the higher order multiscale model is due to non-local measurements. Similar in spirit to the non-redundant models shown in Figures 3-5, 3-8, and 3-9, the state variables at a child node uses linear functionals that are “offset” from the functionals at its parent, such that the additional linear functionals needed for multiscale prediction are readily available at nearby nodes as illustrated in Figure 7-2. The hybrid model shown here is not non-redundant as some of the elements of the process of interest appear twice on the tree. However, it is not strictly speaking an internal model either as the coarse-scale states are not linear functionals of the finest scale, although they are of the overall process of interest.





# Appendix A

## Multiscale Smoothing Algorithms

The essential equations of the multiscale smoothing algorithm are listed here. More detailed development of these equations can be found in [18, 69].

The multiscale smoothing algorithm has been loosely referred to as the multiscale estimation algorithm at various places before. If we treat the multiscale process as a whole, the algorithm presented here produces the best estimate of the process given all available measurements. In this sense the algorithm is an estimation algorithm. On the other hand, the algorithm is scale recursive. Considering the multiscale process scale by scale, the algorithm consists of an estimation procedure upward in scale and a smoothing procedure downward in scale. The algorithm parallels the Rauch-Tung-Striebel smoothing algorithm [75], but scale now takes the role of time. In this sense, the algorithm is a multiscale smoothing algorithm or a multiscale smoother.

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

$$\mathbf{x}(s) = \mathbf{A}(s)\mathbf{x}(s\bar{\gamma}) + \mathbf{B}(s)\mathbf{w}(s) \quad (\text{A.1})$$

$$\mathbf{y}(s) = \mathbf{C}(s)\mathbf{x}(s) + \mathbf{v}(s) \quad (\text{A.2})$$

where  $\mathbf{w}(s)$  is a zero-mean unit-variance white noise process and  $\mathbf{v}(s)$  is a zero-mean white noise process with covariance  $\mathbf{R}(s)$ . We are also given the statistics of the states at the root node: zero mean with covariance  $\mathbf{P}(0)$ . First, the prior covariances

of all states at individual nodes on the tree are computed via a Lyapunov equation

$$\mathbf{P}(s) = \mathbf{A}(s)\mathbf{P}(s\bar{\gamma})\mathbf{A}^T(s) + \mathbf{B}(s)\mathbf{B}^T(s) \quad (\text{A.3})$$

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

- $\mathbf{Y}_s = \{\mathbf{y}(\sigma) | \sigma \text{ is a descendant of } s\}$  is the collection of measurements at all nodes below  $s$  but not including  $s$ .
- $\hat{\mathbf{x}}(\sigma|s) = E[\mathbf{x}(\sigma) | \sigma \in \mathbf{Y}_s \cup \mathbf{y}(s)]$  is the best estimate of  $x(\sigma)$  given measurement at node  $s$  and all nodes below  $s$ .
- $\hat{\mathbf{x}}(\sigma|s+) = E[\mathbf{x}(\sigma) | \sigma \in \mathbf{Y}_s]$  is the best estimate of  $x(\sigma)$  given measurement at all nodes below  $s$ .
- $\tilde{\mathbf{P}}(\sigma|s) = \text{Cov}[\mathbf{x}(\sigma) - \hat{\mathbf{x}}(\sigma|s)]$
- $\tilde{\mathbf{P}}(\sigma|s+) = \text{Cov}[\mathbf{x}(\sigma) - \hat{\mathbf{x}}(\sigma|s+)]$

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

$$\hat{\mathbf{x}}(s|s+) = 0 \quad (\text{A.4})$$

$$\mathbf{P}(s|s+) = \mathbf{P}(s) \quad (\text{A.5})$$

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

$$\mathbf{x}(s\bar{\gamma}) = \mathbf{F}(s)\mathbf{x}(s) + \bar{\mathbf{w}}(s) \quad (\text{A.6})$$

$$\mathbf{y}(s) = \mathbf{C}(s)\mathbf{x}(s) + \mathbf{v}(s) \quad (\text{A.7})$$

where

$$\mathbf{F}(s) = \mathbf{P}(s\bar{\gamma})\mathbf{A}^T(s)\mathbf{P}(s)^{-1} \quad (\text{A.8})$$

$$E[\bar{\mathbf{w}}(s)\bar{\mathbf{w}}(s)^T] = \mathbf{P}(s\bar{\gamma}) - \mathbf{P}(s\bar{\gamma})\mathbf{A}^T(s)\mathbf{P}(s)^{-1}\mathbf{A}(s)\mathbf{P}(s\bar{\gamma}) = \mathbf{Q}(s) \quad (\text{A.9})$$

The upward sweep computes the best estimate of the state at a node given all measurements at or below that node. It consists of three steps at each scale:

(a) Update step:

$$\hat{\mathbf{x}}(s|s) = \hat{\mathbf{x}}(s|s+) + \mathbf{K}(s) [\mathbf{y}(s) - \mathbf{C}(s)\hat{\mathbf{x}}(s|s+)] \quad (\text{A.10})$$

$$\mathbf{P}(s|s) = [\mathbf{I} - \mathbf{K}(s)\mathbf{C}(s)] \mathbf{P}(s|s+) \quad (\text{A.11})$$

$$\mathbf{K}(s) = \mathbf{P}(s|s+)\mathbf{C}^T(s) [\mathbf{C}(s)\mathbf{P}(s|s+)\mathbf{C}^T(s) + \mathbf{R}(s)]^{-1} \quad (\text{A.12})$$

The updated estimate is the best estimate of  $\mathbf{x}(s)$  given all measurements at or below  $s$ .

(b) Prediction step:

$$\hat{\mathbf{x}}(s|s\alpha_i) = \mathbf{F}(s\alpha_i)\hat{\mathbf{x}}(s\alpha_i|s\alpha_i) \quad (\text{A.13})$$

$$\mathbf{P}(s|s\alpha_i) = \mathbf{F}(s\alpha_i)\mathbf{P}(s\alpha_i|s\alpha_i)\mathbf{F}^T(s\alpha_i) + \mathbf{Q}(s\alpha_i) \quad (\text{A.14})$$

The predicted estimate is the best estimate of  $\mathbf{x}(s)$  given all measurements at node  $s\alpha_i$  ( $i = 1 \dots q$ ) or below.

(c) Merge step:

$$\hat{\mathbf{x}}(s|s+) = \mathbf{P}(s|s+) \sum_{i=1}^q \mathbf{P}^{-1}(s|s\alpha_i) \hat{\mathbf{x}}(s|s\alpha_i) \quad (\text{A.15})$$

$$\mathbf{P}(s|s+) = \left[ (1-q)\mathbf{P}(s)^{-1} + \sum_{i=1}^q \mathbf{P}^{-1}(s|s\alpha_i) \right]^{-1} \quad (\text{A.16})$$

The merge step combines the predicted estimate of  $\mathbf{x}(s)$  given each of its child subtrees. The merged estimate is the best estimate of  $\mathbf{x}(s)$  given all measurements below node  $s$ .

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

$$\hat{\mathbf{x}}(s|0) = \hat{\mathbf{x}}(s|s) + \mathbf{J}(s) [\hat{\mathbf{x}}(s\bar{\gamma}|0) - \hat{\mathbf{x}}(s\bar{\gamma}|s)] \quad (\text{A.17})$$

$$\mathbf{P}(s|0) = \mathbf{P}(s|s) + \mathbf{J}(s) [\mathbf{P}(s\bar{\gamma}|0) - \mathbf{P}(s\bar{\gamma}|s)] \mathbf{J}^T(s) \quad (\text{A.18})$$

$$\mathbf{J}(s) = \mathbf{P}(s|s) \mathbf{F}^T(s) \mathbf{P}^{-1}(s\bar{\gamma}|s) \quad (\text{A.19})$$

The smoothing error is itself a multiscale process and can be modeled as

$$\tilde{\mathbf{x}}(s|0) = \mathbf{J}(s) \tilde{\mathbf{x}}(s\bar{\gamma}|0) + \check{\mathbf{w}}(s) \quad (\text{A.20})$$

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

$$E [\check{\mathbf{w}}(s) \check{\mathbf{w}}(s)^T] = \mathbf{P}(s|s) - \mathbf{P}(s|s) \mathbf{F}^T(s) \mathbf{P}^{-1}(s\bar{\gamma}|s) \mathbf{F}(s) \mathbf{P}(s|s) \quad (\text{A.21})$$

Note that the state covariances at individual nodes of the smoothing error model have already been computed in (A.19) and (A.21).

# Appendix B

## MATLAB Programs

### B.1 Data Structure

The majority of the code for the implementation of the multiscale algorithms are written in MATLAB. Version 5 of MATLAB provides two new data types — structures and cell arrays. Different types of information that are very naturally associated with a node, such as its scale, the indices of its parent and of its children, the model parameters, etc., can now be gracefully grouped together in a user defined “node” structure. All the nodes can be placed in a cell array, which in turn is put in a user defined “tree” structure along with other pieces of information pertinent to the whole tree. Function calls are greatly simplified by passing a tree structure as a unit, whereas in older versions of MATLAB the different types of data must be stored in separate matrices and passed as individual parameters of function calls.

A “node” structure may contain the following elements:

<code>node.scale</code>	: scale $m(s)$
<code>node.parent_pt</code>	: location index of node $s\bar{\gamma}$
<code>node.child_pt</code>	: location indices of node(s) $s\alpha_i$
<code>node.A</code>	: $\mathbf{A}(s)$
<code>node.BB</code>	: $\mathbf{BB}^T(s)$
<code>node.L</code>	: linear functionals $\mathbf{L}(s)$

`node.P` : state covariance  $P(s)$   
`node.y` : measurements  $\mathbf{y}(s)$   
`node.C` :  $\mathbf{C}(s)$   
`node.R` : measurement noise covariance  $\mathbf{R}(s)$

Additional elements are easily appended to the above list. For example, the multiscale smoother may add the estimates  $\hat{\mathbf{x}}(s|0)$  as well as the estimation error covariances  $\mathbf{P}(s|0)$  onto the above list.

A “tree” structure may contain a cell array of nodes, the index of the root node, and the number of scales on the tree:

`tree.nodes{}` : cell array of nodes  
`tree.root_pt` : location index of the root node  
`tree.scales` : total number of scales

Additional elements related to the properties of a multiscale tree can be added as well as housekeeping information such as a field that lists the type of data available in the nodes.

MATLAB codes written for the multiscale recursive estimator are available at <http://ssg.mit.edu/group/hot/code/>.

## B.2 Sea Level Anomaly Interpolation Code

The implementation of the multiscale estimation algorithm in its full generality is a rather complicated undertaking. A complete set of computer code for the multiscale smoothing algorithm is available publicly. The front-end visible to the user is written in MATLAB; the multiscale computational engine is written in C by Paul Fieguth and optimized for speed. No programming experience is needed to try the software, although a significant understanding of MATLAB scripts would be required to customize our program for a different application.

Anyone interested in compiling and running our code will require MATLAB (4.x or higher) software and an ANSI-compatible C compiler (precompiled versions of

the code, not requiring any compilation, are available for Sun-SPARC and SGI platforms). Most workstations should have ample computational power. However, large multiscale trees require a large amount of memory, particularly for high state dimensions.

The programs may be obtained via anonymous FTP to `ocho.uwaterloo.ca` from directory `pub/Software/Mediterranean`. The file `README` describes the purpose of each program and how to get started.



# Bibliography

- [1] Shin Ichi Aihara. “On Adaptive Boundary Control for Stochastic Parabolic Systems”. *IEEE Transactions on Automatic Control*, 42(3):350–363, March 1997.
- [2] H. Akaike. “Markovian Representation of Stochastic Processes by Canonical Variables”. *SIAM Journal of Control*, 13(1), January 1975.
- [3] Michael Athans. “Toward a Practical Theory for Distributed Parameter Systems”. *IEEE Transactions on Automatic Control*, pages 245–247, April 1970.
- [4] N. Ayoub, P.-Y. Le Traon, and P. De Mey. “A Description of the Mediterranean Surface Variable Circulation from Combined ERS-1 and TOPEX/POSEIDON Altimeter Data”. *Journal of Maritime Systems*, 1997. In press.
- [5] H. T. Banks and K. Kunisch. *Estimation Techniques for Distributed Parameter Systems*. Birkhäuser, Boston, 1989.
- [6] Michele Basseville, Albert Benveniste, Kenneth Chou, Stuart Golden, Ramine Nikoukhah, and Alan Willsky. “Modeling and Estimation of Multiresolution Stochastic Processes”. *IEEE Transactions on Information Theory*, 38(2):766–784, March 1992.
- [7] Michele Basseville, Albert Benveniste, and Alan Willsky. “Multiscale Autoregressive Processes, Part I: Schur-Levinson Parametrizations”. *IEEE Transactions on Signal Processing*, 40(8):1915–1934, August 1992.

- [8] Michele Basseville, Albert Benveniste, and Alan Willsky. “Multiscale Autoregressive Processes, Part II: Lattice Structures for Whitening and Modeling”. *IEEE Transactions on Signal Processing*, 40(8):1935–1954, August 1992.
- [9] T. Beer. *Environmental Oceanography: An Introduction to the Behavior of Coastal Waters*. Pergamon, New York, 1983.
- [10] P. Bonnefond, P. Exertier, P. Schaeffer, S. Bruinsma, and F. Barlier. “Satellite Altimetry from a Short-Arc Orbit Technique: Application to the Mediterranean”. *Journal of Geophysical Research*, 100(C12):25365–25382, 1995.
- [11] William L. Briggs. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, Philadelphia, 1987.
- [12] J. Candela and C. J. Lozano. “Barotropic Response of the Western Mediterranean to Observed Atmospheric Pressure Forcing”. In P. la Violette, editor, *The Seasonal and Interannual Variability of the Western Mediterranean Sea*, pages 325–359, Washington, D.C., 1994. American Geophysical Union.
- [13] Toshio Michael Chin. *Dynamic Estimation in Computational Vision*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, October 1991.
- [14] Toshio Michael Chin, William Clem Karl, and Alan S. Willsky. “Sequential Filtering for Multi-Frame Visual Reconstruction”. *Signal Processing*, 28:311–333, August 1992.
- [15] Toshio Michael Chin, William Clem Karl, and Alan S. Willsky. “Probabilistic and Sequential Computation of Optical Flow Using Temporal Coherence”. *IEEE Transactions on Image Processing*, 3(6):773–788, November 1994.
- [16] Toshio Michael Chin, William Clem Karl, and Alan S. Willsky. “A Distributed and Iterative Method for Square Root Filtering in Space-Time Estimation”. *Automatica*, 31(1):67–82, 1995.

- [17] Toshio Michael Chin and Arthur J. Mariano. “Wavelet-based Compression of Covariances in Kalman Filtering of Geophysical Flows”. In *Proceedings of SPIE*, volume 2242, pages 842–850, April 1994.
- [18] Kenneth Chou, Alan Willsky, and Albert Benveniste. “Multiscale Recursive Estimation, Data Fusion, and Regularization”. *IEEE Transactions on Automatic Control*, 39(3):464–478, March 1994.
- [19] Kenneth Chou, Alan Willsky, and Ramine Nikoukhah. “Multiscale Systems, Kalman Filters, and Riccati Equations”. *IEEE Transactions on Automatic Control*, 39(3):479–492, March 1994.
- [20] Kenneth C. Chou. *A Stochastic Modeling Approach to Multiscale Signal Processing*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1991.
- [21] Jerome M. Coleman. *Gaussian Spacetime Models: Markov Field Properties*. PhD thesis, University of California at Davis, Davis, CA, August 1995.
- [22] Arlindo da Silva and Jing Guo. “Documentation of the Physical-space Statistical Analysis System (PSAS) Part I: The Conjugate Gradient Solver Version PSAS-1.00”. DAO Office Notes 96-02, Data Assimilation Office, Goddard Laboratory for Atmospheres, NASA, February 1996.
- [23] Michael Daniel. *Multiresolution Statistical Modeling with Application to Modeling Groundwater Flow*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, February 1997.
- [24] Michael Daniel and Alan Willsky. “Modeling and Estimation of Fractional Brownian Motion Using Multiresolution Stochastic Processes”. In J. L. Vehel, E. Lutton, and C. Tricot, editors, *Fractals in Engineering*, pages 124–137. Springer, 1997.

- [25] Mike Daniel and Alan Willsky. “A Multiresolution Methodology for Signal-Level Fusion and Data Assimilation with Applications to Remote Sensing”. *Proceedings of the IEEE*, 85(1):164–183, January 1997.
- [26] Ingrid Daubechies. “Orthonormal Bases of Compactly Supported Wavelets”. *Communications of Pure and Applied Mathematics*, 41:909–996, November 1988.
- [27] A.P. Dempster. “Covariance Selection”. *Biometrics*, 28:157–175, March 1972.
- [28] H. Derin and P. Kelly. “Discrete-Index Markov-Type Random Processes”. *Proceedings of IEEE*, 77(10):1485–1510, 1989.
- [29] C. K. Shum et. al. “Accuracy Assessment of Recent Ocean Tide Models”. *Journal of Geophysical Research*, 1996. In press.
- [30] Paul Fieguth. “Application of Multiscale Estimation to Large Scale Multidimensional Time Varying Problems”. Ph.D. Thesis Proposal, MIT, Cambridge, MA, November 1993.
- [31] Paul Fieguth, M. R. Allen, and M. J. Murray. “Hierarchical Methods for Global-Scale Estimation Problems”. In *Proceedings of Canadian Conference on Electrical Engineering*, pages 161–164, Waterloo, Canada, 1998.
- [32] Paul Fieguth, William Karl, and Alan Willsky. “Efficient Multiresolution Counterparts to Variational Methods in Surface Reconstruction”. *Computer Vision and Image Understanding*, 70(2):157–176, 1998.
- [33] Paul Fieguth, William Karl, Alan Willsky, and Carl Wunsch. “Multiresolution Optimal Interpolation and Statistical Analysis of TOPEX/POSEIDON Satellite Altimetry”. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2):280–292, March 1995.
- [34] Paul Fieguth, Dimitris Menemenlis, Terrence Ho, Alan Willsky, and Carl Wunsch. “Mapping Mediterranean Altimeter Data with a Multiresolution Optimal

- Interpolation Algorithm”. *Journal of Atmospheric and Oceanic Technology*, 15:535–546, April 1998.
- [35] Paul Fieguth and Alan Willsky. “Fractal Estimation Using Models on Multiscale Trees”. *IEEE Transactions on Signal Processing*, 44(5):1297–1300, 1996.
- [36] Paul Werner Fieguth. *Application of Multiscale Estimation to Large Scale Multi-dimensional Imaging and Remote Sensing Problems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1995.
- [37] Patrick Flandrin. “On the Spectrum of Fractional Brownian Motions”. *IEEE Transactions on Information Theory*, 35(1):197–199, January 1989.
- [38] Patrick Flandrin. “Wavelet Analysis and Synthesis of Fractional Brownian Motion”. *IEEE Transactions on Information Theory*, 38(2):910–917, March 1992.
- [39] Charlie Fosgate, Hamid Krim, William Irving, and Alan Willsky. “Multiscale Segmentation and Anomaly Enhancement of SAR Imagery”. *IEEE Transactions on Image Processing*, 6(1):7–20, January 1997.
- [40] Austin Frakt and Alan Willsky. “Efficient Multiscale Stochastic Realization”. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Seattle, Washington, May 1998.
- [41] Austin B. Frakt. “A Realization Theory for Multiscale Auto-Regressive Stochastic Models”. Ph.D. Thesis Proposal, MIT, Cambridge, MA, September 1996.
- [42] Austin B. Frakt. “Multiscale Hypothesis Testing with Application to Anomaly Characterization from Tomographic Projections”. Master’s thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1996.
- [43] Austin B. Frakt, W. Clem Karl, and Alan S. Willsy. “A Multiscale Hypothesis Testing Approach to Anomaly Detection and Localization from Noisy Tomographic Data”. *IEEE Transactions on Image Processing*, 7(6):825–837, June 1998.

- [44] L.-L. Fu, E. J. Christensen, C. A. Yamarone Jr., M. Lefebvre, Y. Ménard, M. Dorner, and P. Escudier. “TOPEX/POSEIDON Mission Overview”. *Journal of Geophysical Research*, 99(C12):24369–24381, 1994.
- [45] Philippe Gaspar and Carl Wunsch. “Estimates from Altimeter Data of Barotropic Rossby Waves in the Northwestern Atlantic Ocean”. *Journal of Physical Oceanography*, 19(12):1821–1844, 1989.
- [46] Alan George. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.
- [47] Ellen Hildreth. “Computations Underlying the Measurement of Visual Motion”. *Artificial Intelligence*, 23:309–354, 1984.
- [48] Terrence Ho, Paul Fieguth, and Alan Willsky. “Computationally Efficient Steady-State Multiscale Estimation for 1-D Diffusion Processes”. *Automatica*. Submitted.
- [49] Terrence Ho, Paul Fieguth, and Alan Willsky. “Multiresolution Stochastic Models for the Efficient Solution of Large-Scale Space-Time Estimation Problems”. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3098–4001, Atlanta, Georgia, May 1996.
- [50] Terrence Ho, Paul Fieguth, and Alan Willsky. “Computationally Efficient Multiscale Estimation of Large-Scale Dynamic Systems”. In *Proceedings of International Conference on Image Processing*, Chicago, Illinois, 1998. To be presented.
- [51] Terrence Ho, Austin Frakt, and Alan Willsky. “Multiscale Realization and Estimation for Space and Space-Time Problems”. In *Proceedings of IEEE International Symposium on Information Theory*, Cambridge, Massachusetts, 1998. To be presented.
- [52] Terrence T. Ho. “Large-Scale Multiscale Estimation of Dynamic Systems”. Ph.D. Thesis Proposal, MIT, Cambridge, MA, November 1996.

- [53] B. K. P. Horn and B. G. Schunck. “Determining Optical Flow”. *Artificial Intelligence*, 17:185–203, 1981.
- [54] Leonard R. Ingersoll, Otto J. Zobel, and Alfred C. Ingersoll. *Heat Conduction with Engineering, Geological, and Other Applications*. The University of Wisconsin Press, Madison, 1954.
- [55] William W. Irving. *Theory for Multiscale Stochastic Realization and Identification*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1995.
- [56] William W. Irving. “A Canonical Correlations Approach to Multiscale Stochastic Realization”. *IEEE Transactions on Automatic Control*, 1998. In press.
- [57] William W. Irving, Paul W. Fieguth, and Alan S. Willsky. “An Overlapping Tree Approach to Multiscale Stochastic Modeling and Estimation”. *IEEE Transactions on Image Processing*, 6(11):1517–1529, November 1997.
- [58] William W. Irving, Leslie M. Novak, and Alan S. Willsky. “A Multiresolution Approach to Discrimination in SAR Imagery”. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1157–1169, October 1997.
- [59] Thomas Kailath. “The Divergence and Bhattacharyya Distance Measures in Signal Selection”. *IEEE Transactions on Communication Technology*, Comm-15(1), February 1967.
- [60] J. Kaufhold, M. Schneider, W. Karl, and A. Willsky. “A Statistical Approach to Segmentation of MRI Imagery”. *Journal of Pattern Recognition and Artificial Intelligence*. To appear.
- [61] M. S. Keshner. “1/f Noise”. *Proceedings of IEEE*, 70:212–218, 1982.
- [62] Andrew J. Kim and Hamid Krim. “Hierarchical Stochastic Modeling for Segmentation and Compression of SAR Imagery”. *IEEE Transactions on Signal Processing*. To appear.

- [63] C. King, D. Stammer, and C. Wunsch. The cmpo/mit topex/poseidon altimetric data set. Technical Report 30, Massachusetts Institute of Technology, Cambridge, 1994.
- [64] G. Larnicol, P.-Y. Le Traon, N. Ayoub, and P. De Mey. “Mean Sea Level and Surface Circulation Variability of the Mediterranean Sea from Two Years of TOPEX/POSEIDON Altimetry”. *Journal of Geophysical Research*, 100(C12):25163–25177, 1995.
- [65] Hanoch Lev-Ari, Sydney R. Parker, and Thomas Kailath. “Multidimensional Maximum-Entropy Covariance Extension”. *IEEE Transactions on Information Theory*, 35(3):497–508, May 1989.
- [66] Mark Luetzgen, William Karl, and Alan Willsky. “Efficient Multiscale Regularization with Applications to the Computation of Optical Flow”. *IEEE Transactions on Image Processing*, 3(1):41–64, January 1994.
- [67] Mark Luetzgen, William Karl, Alan Willsky, and Robert Tenney. “Multiscale Representations of Markov Random Fields”. *IEEE Transactions on Signal Processing*, 41(12):3377–3395, December 1993.
- [68] Mark Luetzgen and Alan Willsky. “Likelihood Calculation for a Class of Multiscale Stochastic Models, with Application to Texture Discrimination”. *IEEE Transactions on Image Processing*, 4(2):194–207, February 1995.
- [69] Mark Luetzgen and Alan Willsky. “Multiscale Smoothing Error Models”. *IEEE Transactions on Automatic Control*, 40(1):173–175, January 1995.
- [70] Mark R. Luetzgen. *Image Processing with Multiscale Stochastic Models*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1993.
- [71] J. A. Marshall, N. P. Zelensky, S. M. Klosko, D. S. Chinn, S. B. Luthcke, K. E. Rachlin, and R. G. Williamson. “The Temporal and Spatial Characteristics of TOPEX/POSEIDON Radial Orbit Error”. *Journal of Geophysical Research*, 100(C12):25331–25352, 1995.

- [72] D. Menemenlis, T. Webb, C. Wunsch, U. Send, and C. Hill. “Basin-Scale Ocean Circulation from Combined Altimetric, Tomographic and Model Data”. *Nature*, 385:618–621, 1997.
- [73] Dimitris Menemenlis, Paul Fieguth, Carl Wunsch, and Alan Willsky. “Adaptation of a Fast Optimal Interpolation Algorithm to the Mapping of Oceanographic Data”. *Journal of Geophysical Research*, 102(C5):10573–10584, 1997.
- [74] S. Omatu, J. H. Seifeld, T. Soeda, and Y. Sawaragi. “Estimation of Nitrogen Dioxide Concentration in the Vicinity of a Roadway by Optimal Filtering Theory”. *Automatica*, 24(1):19–29, January 1988.
- [75] H. E. Rauch, F. Tung, and C. T. Striebel. “Maximum Likelihood Estimates of Linear Dynamic Systems”. *AIAA Journal*, 3(8):1445–1450, 1965.
- [76] Michael Schneider. “Multiscale Methods for the Segmentation of Images”. Master’s thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1996.
- [77] Michael Schneider. “Krylov Subspace Algorithms for Estimation”. Ph.D. Thesis Proposal, MIT, Cambridge, MA, June 1998.
- [78] Michael Schneider, Paul Fieguth, Clem Karl, and Alan Willsky. “Multiscale Methods for the Segmentation of Images”. *IEEE Transactions on Image Processing*. To appear.
- [79] Ziv Sirkes and Carl Wunsch. “Note on Apparent Systematic and Periodic Errors in Geosat Orbits”. *Geophysical Research Letters*, 17(9):1307–1310, Aug. 1990.
- [80] T. P. Speed and H. T. Kiiveri. “Gaussian Markov Distributions Over Finite Graphs”. *The Annals of Statistics*, 14(1):138–150, 1986.
- [81] Gilbert Strang and Truong Ngyeun. “Wavelets and Filter Banks”. MIT 18.325 Class Notes, to be published by Wellsley-Cambridge Press, 1995.
- [82] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Chapman and Hall, New York, 1989.

- [83] Chang-Kou Tai. “Geosat Crossover Analysis in the Tropical Pacific, Part 1, Constrained Sinusoidal Crossover Adjustment”. *Journal of Geophysical Research*, 93(C9):10621–10629, Sept. 1988.
- [84] A. Tewfik and M. Kim. “Correlation Structure of the Discrete Wavelet Coefficients of Fractional Brownian Motion”. *IEEE Transactions on Information Theory*, 38(2):904–909, March 1992.
- [85] R. T. Tokmakian and P. G. Challenor. “Observations in the Canary Basin and the Azores Frontal Region Using Geosat Data”. *Journal of Geophysical Research*, 98(C3):4761–4773, Mar. 1993.
- [86] P.-Y. Le Traon, F. Blanc, J. Dorandeu, P. Gaspar, P. Sicard, J. Stum, and P. Vincent. “AVISO User Handbook: Sea Level Anomaly Files”. Technical Report AVI-NT-011-312-CN, Centre National d’Etudes Spatiales, Toulouse Cedex, France, 1995.
- [87] P.-Y. Le Traon, P. Gaspar, F. Bouyssel, and H. Makhmara. “Using TOPEX/POSEIDON Data to Enhance ERS-1 Data”. *Journal of Atmospheric and Oceanic Technology*, 12:161–170, 1995.
- [88] P.-Y. Le Traon and P. Gauzelin. “Response of the Mediterranean Mean Sea Level to Atmospheric Pressure Forcing”. *Journal of Geophysical Research*, 102(C1):973–984, 1997.
- [89] P.-Y. Le Traon, F. Nadal, and N. Ducet. “An Improved Mapping Method of Multi-Satellite Altimeter Data”. *Journal of Atmospheric and Oceanic Technology*, 1997. Submitted.
- [90] P.-Y. Le Traon, J. Stum, J. Dorandeu, and P. Gaspar. “Global Statistical Analysis of TOPEX and POSEIDON Data”. *Journal of Geophysical Research*, 99(C12):25619–25631, 1994.

- [91] J. Vasquez-Cuervo, J. Font, and J. J. Martinez-Benjamin. “Observations on the Circulation in the Alboran Sea Using ERS1 Altimetry and Sea Surface Temperature Data”. *Journal of Geophysical Research*, 1996. In press.
- [92] Gregory W. Wornell. “A Karhunen-Loève-like Expansion for 1/f Processes via Wavelets”. *IEEE Transactions on Information Theory*, 36(4):859–861, July 1990.
- [93] C. Wunsch and D. Stammer. “The Global Frequency-Wavenumber Spectrum of Oceanic Variability Estimated from TOPEX/POSEIDON Altimetric Measurements”. *Journal of Geophysical Research*, 100(C12):24895–24910, 1995.
- [94] Carl Wunsch. “Using Transient Tracers: the Regularization Problem”. *Tellus*, 39B:477–492, 1987.
- [95] Carl Wunsch. “Transient Tracers as a Problem in Control Theory”. *Journal of Geophysical Research*, 93(C7):8099–8110, July 1988.

