

Bayesian Inference for Complex and Large-Scale Engineering Systems

by

Mohammad Khalil

A thesis submitted to the Faculty of Graduate and Postdoctoral
Affairs in partial fulfillment of the requirements
for the degree of

**Doctor of Philosophy
in
Civil Engineering**

**Carleton University
Ottawa, Ontario**

© 2013
Mohammad Khalil



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-94548-3

Our file Notre référence

ISBN: 978-0-494-94548-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

The process of blending observational data in numerical models can be formulated using the Bayesian estimation method. The Bayesian statistical framework strives to find the best state and/or parameter estimates using (a) the stochastic computational model and (b) the noisy observational data. In sequential Bayesian estimation techniques, the data assimilation is performed at each time instant as data arrives. A number of extensions to the linear Kalman filter (e.g. extended Kalman filter, unscented Kalman filter and ensemble Kalman filter) have become popular tools for sequential data assimilation problems applicable to nonlinear systems. In the Bayesian setting, the most general nonlinear filtering algorithm is the so-called particle filter which can handle general form of nonlinearities and non-Gaussian noises in the model and measurement operators. The problem of static (time-invariant) parameter estimation problem is formulated using the aforementioned filtering techniques in combination with the Markov Chain Monte Carlo sampling method. In this thesis, the usefulness of these methods is demonstrated through novel applications in structural dynamics and aeroelasticity.

For large-scale computational models, the parallel filtering algorithms aim to estimate the state of the system by exploiting high performance computing platforms. A significant research initiative has recently been focused on the parallelization of the ensemble Kalman filter due to its relatively low computational requirements. A major limitation of the ensemble Kalman filter to tackle the data assimilation problem for large-scale systems is the necessity of a large ensemble size which is often computationally impractical

or infeasible. By avoiding Monte Carlo sampling, a novel filter has been recently proposed in the literature to address this issue by using a functional series (i.e. polynomial chaos series) representation of the stochastic state vector. This representation is more efficient in capturing the model state statistics for accurate forecast and analysis of observations leading to the Polynomial Chaos-based Kalman filter. In this thesis, a parallel assimilation (update) algorithm is developed with a scalable implementation using the Polynomial Chaos-based Kalman filter that exploits a previously developed polynomial chaos based scalable domain decomposition solver. Numerical investigations involve the two-dimensional stationary stochastic diffusion and advection-diffusion problems. The scalability of the parallel Polynomial Chaos-based Kalman filter algorithm is investigated in a Linux cluster.

The following statements summarize the findings from the investigations undertaken in this thesis: (1) the proposed probabilistic flutter margin method applied to a two-degrees-of-freedom numerical model provides more accurate (unbiased) and precise (low variance) estimates for the flutter speed in comparison to the conventional method; (2) for the estimation of time-invariant parameters of a nonlinear system exhibiting noisy oscillation, Markov Chain Monte Carlo sampling, ensemble Kalman filter and particle filter provide parameter estimates which are comparable in accuracy for the case of dense observational data and weak measurement noise whereas for the cases of strong measurement noise and sparse observational data, the Markov Chain Monte Carlo sampling-based algorithm and particle filters outperform the ensemble Kalman filter; (3) For static parameter estimation problem of a non-linear aeroelastic system using wind tunnel data, numerical predictions of the limit cycle oscillation match the experimental observations from wind tunnel tests with reasonable accuracy; (4) the developed parallel data assimilation algorithm that integrates the polynomial chaos-based Kalman filter and

domain decomposition solver shows scalable performance for stochastic diffusion and advection-diffusion problems with spatially varying non-Gaussian random parameters;

(5) A new parallel solver for non-symmetric stochastic systems based on the preconditioned biconjugate gradient stabilized method is developed to tackle the steady-state stochastic advection-diffusion equation with spatially varying non-Gaussian diffusivity coefficient.

*Dedicated to my wife Samar,
my father Adnan,
and my mother Athra.*

Acknowledgments

During the course of my studies towards a Ph.D. degree, I have been privileged to have the support of the following people:

For their intellectual feedback, I would like to thank my professors and colleagues at Carleton University. In particular, I would like to thank my supervisor professor Abhijit Sarkar for providing an open, friendly and collaborative research atmosphere as well as his constant attention and feedback. Special thanks also go to Waad Subber for disseminating his knowledge on the method of Domain Decomposition and his expertise in high performance computing.

Furthermore, I would like to thank professor Sondipon Adhikari for his intellectual contribution in the field of system identification of vibrating structures. I would also like to thank Professor Dominique Poirel for his contribution in my understanding of the basics in aeroelasticity and providing me with wind-tunnel experimental data. His financial contribution is also well appreciated.

Many more people have helped me, directly or indirectly, knowingly or unknowingly, at different stages. I would like to thank Alex Oldford, Anton Matachniouk, Amin Fereidononi, Xiaoli Zhu, professor Neil Holtz, professor Heng Khoo, professor Burkan Isgor and the anonymous reviewers of my papers.

I thank the members of my thesis committee, Professor Chris Pettit, Professor Shirley Mills, Professor Ousmane Seidou, and Professor David Lau, for investing the time into reading my thesis and providing valuable feedback in improving the manuscript

Over the last four years, I was supported by an NSERC graduate research assistance-ship and a Department of National Defence internship for which I am grateful. I also would like to thank Carleton University for the internal scholarships I was provided with.

I would like to thank my father and mother for their endless support. Last but not least, I would like to thank my wife Samar who has been my guiding light over the course of my PhD studies. I am indebted to her for the endless support and encouragement.

Contents

1	Introduction and Motivation	1
1.1	Background	1
1.2	Brief Review of Filtering Algorithms for Sequential Data Assimilation .	3
1.3	Bayesian Inference for Estimation of Static Parameters	6
1.4	Parallel Data Assimilation Algorithms	7
1.5	Thesis Statement	10
1.6	Thesis Contributions	11
1.7	Thesis Overview	13
2	Bayesian framework for state and/or parameter estimation	15
2.1	Introduction	15
2.2	Bayesian Inference	16
2.2.1	Statistical Modeling	16
2.2.2	Bayesian Framework for Estimation	18
2.2.3	Estimators	20
2.2.4	Dynamical systems	20
2.3	Linear Filtering: The Kalman Filter	24
2.4	Extended Kalman filter	28
2.5	Unscented Kalman Filter	31

2.5.1	Unscented Transform	31
2.6	Ensemble Kalman filter	35
2.7	Particle Filters	37
2.7.1	Particle Representation of a Probability Distribution	37
2.7.2	Principle of Particle Filters	41
2.8	Application to Nonlinear Spring-Damper Oscillator	49
2.8.1	State Estimation	50
3	Markov Chain Monte Carlo	54
3.1	Monte Carlo Integration	55
3.2	Theory of Markov Chains	57
3.3	The Metropolis-Hastings Algorithm	61
3.3.1	Random-Walk Metropolis Algorithm	62
3.4	Application: Signal Identification	63
3.4.1	Maximum Likelihood Estimation	64
3.4.2	Bayesian inference	67
3.5	Application: Flutter Speed Estimation	70
3.5.1	Introduction	71
3.5.2	Formulation	73
3.5.3	Application to a Two-DOF Airfoil	77
4	Static Parameter Estimation	90
4.1	Introduction	90
4.2	Mathematical Formulation For Parameter Estimation	92
4.2.1	Nonlinear Filtering Approach - Joint State and Parameter Esti- mation	92

4.2.2	Markov Chain Monte Carlo Simulation Approach	94
4.3	Particle filter with EnKF Proposal - Extension to Combined State and Parameter Estimation	95
4.4	Noisy oscillation of a Duffing system	99
4.4.1	Duffing Oscillator Model	99
4.4.2	Noisy oscillation of the Duffing oscillator	102
4.4.3	Parameter Estimation	103
5	Static Parameter Estimation of a Non-linear Aeroelastic System using Wind Tunnel Data	117
5.1	Introduction	117
5.2	Theory	121
5.2.1	Mathematical Formulation For Parameter Estimation	122
5.3	Numerical Study - Self-Sustained Oscillations of an Airfoil	124
5.3.1	A Reduced Model	134
5.4	Wind Tunnel Experiments - Self-Sustained Oscillations of an Airfoil . .	139
5.4.1	Free-Decay Data	140
5.4.2	Limit Cycle Oscillation Data	144
5.4.3	A Reduced Model	151
6	Parallel Data Assimilation for Large-Scale Models of Stochastic PDEs	157
6.1	Polynomial Chaos Representation for Uncertainty Propagation	158
6.2	Domain Decomposition Methods for Spectral SFEM	160
6.3	Polynomial Chaos Based Ensemble Kalman Filter	166
6.3.1	Distributed implementation of the PCKF Analysis Step	171
6.4	Numerical Results	178

6.4.1	Stationary Stochastic Diffusion Problem	179
6.4.2	Stationary Stochastic Advection-Diffusion Problem	191
6.4.3	Parallel Scalability of the Parallel Data Assimilation Algorithm	202
7	Summary of Contributions and Future Research Directions	206
7.1	Summary of Contributions	206
7.2	Future Research Directions	208
	Appendices	211
A	Mathematical Preliminaries	211
A.1	Inverse and Determinant of Partitioned Symmetric Matrix	211
A.2	Marginal and Conditional Distributions of Multivariate Normal Distribution	216
A.3	Theorems for the Derivation of the Kalman Filter	221
A.4	Integral of a Product of Two Gaussian Functions	225
B	Newmark-Beta method for Time-marching of Coupled ODEs	226
C	Mathematical Representation of Random Processes	228
C.1	Karhunen-Loeve Expansion	228
C.1.1	Integral Eigenvalue Problem of Exponential Covariance Function	229
C.2	Polynomial Chaos Expansion	231
C.2.1	PCE Representation of Lognormal Stochastic Process	234
D	Parallel Iterative Solvers for the Extended Schur Complement System	237
D.1	Parallel One-Level Lumped Preconditioner	241
D.2	Parallel Matrix-Vector Product	243

D.3	Parallel Preconditioned Conjugate Gradient Method: Iterative Solver for the Extended Schur Complement System	244
D.3.1	Numerical Validation	244
D.4	Parallel Preconditioned Biconjugate Gradient Stabilized Algorithm: Iterative Solver for the Extended Schur Complement System	250
D.4.1	Numerical Validation	250
E	Equation of Motion for a 2D Airfoil with Quasi-Steady Aerodynamics	255
	References	258

List of Tables

- 3.1 Application of MCMC: Flutter Speed Estimation. True modal parameters and flutter-margin for 2-DOF airfoil. 77
- 3.2 Application of MCMC: Flutter Speed Estimation. Range of flutter speed estimates using 10 experiments. 89

- C.1 Polynomial chaoses for a random dimension of 2 of order up to and including 3 and their variances [1]. 233
- C.2 The expectation and variance of the PC basis. 236

List of Figures

2.1	Double-well potential.	50
2.2	True, measured and estimated displacement of the double-well system. .	52
2.3	Prior and Posterior displacement pdfs of the double-well system.	53
3.1	Application of MCMC: Signal Identification. Case 1: True signal and measurements.	66
3.2	Application of MCMC: Signal Identification. Case 1: Misfit function for a_3	66
3.3	Application of MCMC: Signal Identification. Case 1: Reconstructed signal using Maximum Likelihood Estimation.	66
3.4	Application of MCMC: Signal Identification. Case 2: True signal and measurements.	66
3.5	Application of MCMC: Signal Identification. Case 2: Misfit function for a_3	67
3.6	Application of MCMC: Signal Identification. Case 1: Marginal pdfs obtained using MCMC sampling of the posterior pdf.	68
3.7	Application of MCMC: Signal Identification. Case 1: Joint pdfs ob- tained using MCMC sampling of the posterior pdf.	68

3.8	Application of MCMC: Signal Identification. Case 2: Marginal pdfs obtained using MCMC sampling of the posterior pdf with flat prior.	69
3.9	Application of MCMC: Signal Identification. Case 2: Joint pdfs obtained using MCMC sampling of the posterior pdf with flat prior.	69
3.10	Application of MCMC: Signal Identification. Case 2: Marginal pdfs obtained using MCMC sampling of the posterior pdf with uniform prior.	70
3.11	Application of MCMC: Signal Identification. Case 2: Joint pdfs obtained using MCMC sampling of the posterior pdf with uniform prior.	70
3.12	Flow chart describing the probabilistic flutter margin method.	76
3.13	Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 1.95$ m/s - experiment 1.	78
3.14	Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 2.51$ m/s - experiment 1.	78
3.15	Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 3.06$ m/s - experiment 1.	79
3.16	Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 3.62$ m/s - experiment 1.	79
3.17	Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 4.18$ m/s - experiment 1.	79
3.18	Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 4.73$ m/s - experiment 1.	80
3.19	Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 1.95$ m/s - experiment 1.	80
3.20	Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 2.51$ m/s - experiment 1.	81

3.21	Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 3.06$ m/s - experiment 1.	81
3.22	Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 3.62$ m/s - experiment 1.	81
3.23	Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 4.18$ m/s - experiment 1.	81
3.24	Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 4.73$ m/s - experiment 1.	82
3.25	Application of MCMC: Flutter Speed Estimation. Flutter margin polynomial fit using conventional method - experiment 1.	83
3.26	Application of MCMC: Flutter Speed Estimation. Flutter margin polynomial fit using conventional method - all experiments.	83
3.27	Application of MCMC: Flutter Speed Estimation. Flutter margin linear fit - all experiments.	83
3.28	Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 1.95$ m/s - experiment 1.	85
3.29	Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 2.51$ m/s - experiment 1.	85
3.30	Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 3.06$ m/s - experiment 1.	85
3.31	Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 3.62$ m/s - experiment 1.	85
3.32	Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 4.18$ m/s - experiment 1.	86

3.33	Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 4.73$ m/s - experiment 1.	86
3.34	Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 1.95$ m/s - experiment 1.	86
3.35	Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 2.51$ m/s - experiment 1.	87
3.36	Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 3.06$ m/s - experiment 1.	87
3.37	Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 3.62$ m/s - experiment 1.	87
3.38	Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 4.18$ m/s - experiment 1.	88
3.39	Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 4.73$ m/s - experiment 1.	88
3.40	Application of MCMC: Flutter Speed Estimation. Flutter speed pdf obtained using Bayesian inference - experiment 1.	88
3.41	Application of MCMC: Flutter Speed Estimation. Flutter speed pdf obtained using Bayesian inference - all experiments.	89
4.1	Response of the Duffing oscillator under purely deterministic input. . .	102
4.2	Response of the Duffing oscillator under combined deterministic and random input.	103
4.3	Duffing oscillator c parameter estimates.	105
4.4	Duffing oscillator k_1 parameter estimates.	106
4.5	Duffing oscillator k_3 parameter estimates.	107
4.6	Posterior marginal pdfs of parameters for Duffing oscillator.	108

4.7	Duffing oscillator c parameter estimates with stronger measurement noise.	109
4.8	Duffing oscillator k_1 parameter estimates with stronger measurement noise.	110
4.9	Duffing oscillator k_3 parameter estimates with stronger measurement noise.	111
4.10	Posterior marginal pdfs of parameters for Duffing oscillator with stronger measurement noise.	112
4.11	Duffing oscillator c parameter estimates with sparse observational data.	113
4.12	Duffing oscillator k_1 parameter estimates with sparse observational data.	114
4.13	Duffing oscillator k_3 parameter estimates with sparse observational data.	115
4.14	Posterior marginal pdfs of parameters for Duffing oscillator with sparse observational data.	116
5.1	Flow chart describing algorithm used to sample the posterior parameter pdf for the fluttering aeroelastic system.	130
5.2	Fluttering aeroelastic system - Numerical study: (a) true response and (b) measured noisy response of the perturbed nonlinear oscillator. . . .	131
5.3	Fluttering aeroelastic system - Numerical study: posterior parameter pdfs obtained using MCMC simulations.	131
5.4	Fluttering aeroelastic system - Numerical study: posterior joint parameter pdfs obtained using MCMC simulations.	132
5.5	Fluttering aeroelastic system - Numerical study: three pitch realizations obtained using MAP parameter estimates superposing the measured data.	133
5.6	Fluttering aeroelastic system - Numerical study: reduced model posterior parameter pdfs obtained using MCMC simulations.	137

5.7	Fluttering aeroelastic system - Numerical study: reduced model posterior joint parameter pdfs obtained using MCMC simulations.	137
5.8	Fluttering aeroelastic system - Numerical study: three pitch realizations obtained using MAP parameter estimates of the reduced model superposing the measured data.	138
5.9	Fluttering aeroelastic system - Experimental setup of the NACA0012 airfoil in the wind tunnel.	140
5.10	Fluttering aeroelastic system - Wind tunnel experiments: filtered free decay response of the NACA0012 airfoil.	141
5.11	Fluttering aeroelastic system - Wind tunnel free-decay experiments for NACA0012 airfoil: posterior parameter pdfs obtained using MCMC simulations.	142
5.12	Fluttering aeroelastic system - Wind tunnel free-decay experiments for NACA0012 airfoil: posterior joint parameter pdfs obtained using MCMC simulations.	142
5.13	Fluttering aeroelastic system - Wind tunnel free-decay experiments for NACA0012 airfoil: three pitch realizations obtained using MAP parameter estimates superposing the measured data.	143
5.14	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: (a) filtered LCO time history and (b) phase plane.	145
5.15	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: three filtered LCOs for different initial conditions.	145
5.16	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: posterior parameter pdfs obtained using MCMC simulations. . .	148

5.17	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: posterior joint parameter pdfs obtained using MCMC simulations.	149
5.18	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: three pitch realizations obtained using MAP parameter estimates superposing the measured data.	150
5.19	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: reduced model posterior parameter pdfs obtained using MCMC simulations.	153
5.20	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: reduced model posterior joint parameter pdfs obtained using MCMC simulations.	154
5.21	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: three pitch realizations obtained using MAP parameter estimates of the reduced model superposing the measured data.	155
5.22	Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: phase space diagrams of observed LCO and predicted LCO using MAP parameter estimates.	155
5.23	Fluttering aeroelastic system - Estimated static aerodynamic moment coefficient.	156
6.1	Partitioned finite element mesh.	163
6.2	Eigenvalues of the exponential covariance function.	181
6.3	Selected eigenfunctions of the exponential covariance function.	181
6.4	Stationary stochastic diffusion problem: FEM mesh and decomposed domain.	183
6.5	Stationary stochastic diffusion problem: Prior PCE coefficients.	184

6.6	Stationary stochastic diffusion problem: Prior mean and standard deviation.	185
6.7	Data assimilation in stationary stochastic diffusion problem - Experiment 1: Sensor location.	185
6.8	Data assimilation in stationary stochastic diffusion problem - Experiment 1: Posterior PCE coefficients.	186
6.9	Data assimilation in stationary stochastic diffusion problem - Experiment 1: True response, and prior and posterior estimates.	187
6.10	Data assimilation in stationary stochastic diffusion problem - Experiment 1: Prior and posterior standard deviation.	187
6.11	Data assimilation in stationary stochastic diffusion problem - Experiment 2: Sensor locations.	188
6.12	Data assimilation in stationary stochastic diffusion problem - Experiment 2: Posterior PCE coefficients.	189
6.13	Data assimilation in stationary stochastic diffusion problem - Experiment 2: True response, and prior and posterior estimates.	190
6.14	Data assimilation in stationary stochastic diffusion problem - Experiment 2: Prior and posterior standard deviation.	190
6.15	Stationary stochastic advection-diffusion problem: FEM mesh and decomposed domain.	194
6.16	Stationary stochastic advection-diffusion problem: Prior PCE coefficients.	195
6.17	Stationary stochastic advection-diffusion problem: Prior mean and standard deviation.	196
6.18	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: Sensor location.	196

6.19	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: Posterior PCE coefficients.	197
6.20	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: True response, and prior and posterior estimates.	198
6.21	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: Prior and posterior standard deviation.	198
6.22	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: Sensor locations.	199
6.23	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: Posterior PCE coefficients.	200
6.24	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: True response, and prior and posterior estimates.	201
6.25	Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: Prior and posterior standard deviation.	201
6.26	Parallel scalability of the parallel data assimilation algorithm: fixed problem size and number of measurements.	204
6.27	Parallel scalability of the parallel data assimilation algorithm: fixed problem size per subdomain.	204
6.28	Parallel scalability of the parallel data assimilation algorithm: fixed number of measurements per subdomain.	205
D.1	Validation of parallel PCGM solver: Analytical response.	247
D.2	Validation of parallel PCGM solver: FEM mesh and decomposed domain.	248
D.3	Validation of parallel PCGM solver: Estimated mean.	249
D.4	Validation of parallel PCGM solver: Estimated standard deviation.	249
D.5	Validation of parallel BICGSTAB solver: Analytical response.	253

D.6	Validation of parallel BICGSTAB solver: Estimated mean.	254
D.7	Validation of parallel BICGSTAB solver: Estimated standard deviation.	254
E.1	Schematic of 2D airfoil with heave and pitch degrees of freedom.	256

List of Acronyms

- CPU** central processing unit
- DDM** domain decomposition method
- DOF** degree-of-freedom
- EKF** extended Kalman filter
- EnKF** ensemble Kalman filter
- FEM** finite element method
- GPU** graphics processing unit
- HPC** high-performance computing
- IID** independent and identically distributed
- KDE** kernel density estimation
- KF** Kalman filter
- LCO** limit cycle oscillation
- LHS** latin hypercube sampling
- MC** Monte Carlo
- MCMC** Markov Chain Monte Carlo
- MPI** message passing interface
- ODE** ordinary differential equation
- PBICGSTAB** preconditioned biconjugate gradient stabilized method
- PCGM** preconditioned conjugate gradient method
- PCKF** Polynomial Chaos-based Kalman filter
- PDE** partial differential equation

pdf probability density function

PF particle filter

SDE stochastic differential equation

SSFEM spectral stochastic finite element method

UKF unscented Kalman filter

UT unscented transform

Chapter 1

Introduction and Motivation

1.1 Background

Mathematical models of physical and engineering systems may be broadly classified as [2]: (a) statistics-based models (e.g. time-series models) fitted using available observational data; (b) physics-based models, generally described by partial differential equations derived by satisfying certain physical laws (e.g. equilibrium and conservation principles in mechanics); (c) reduced-order models which capture dominant spatio-temporal features of the physics-based models (e.g. modal representation of structural dynamic systems). However, no mathematical model is the “true model” that produces the observational data [3]. Hypothetically, even if one assumes that the true model is known, it may not be practically useful for inference (understanding and prediction) when fitted using finite data set as its large number of parameters are estimated with low precision [3]. In general, the measured state of the system contains information and noise. A model of the system strives to compactly represent the information in data by filtering out the noise with the primary objective to address “what inferences do the data

support about the population?” [3].

Given a data set, the efficacy of a model hinges on whether it is over-fitted or under-fitted, not by the criterion if it closely mimics a (unknown) true model [3]. This fact relates to the principle that involves the choice of models with minimum number of parameters by bias and variance trade-off [3]. The over-fitted and under-fitted models do not imply or assume the existence of a low-dimensional model representing “truth”. As the number of model parameters increases, the misfit of the model output with data (i.e. bias) decreases. On the other hand, the variance in the parameter estimates increases making inference difficult for such models. Some dominant features will be present in most of the samples. One can make (strong) inference about the dominant features using *a single sample* (i.e. single time series). Only weak inference may be made about the subdominant features exhibited by some (say 70%) of the samples. Only a few samples will display some rare features such as extreme events. Attempts to make inference about such rare events specific to a given sample leads to over-fitting as the resulting inference is not relevant to most samples. Such rare features (appearing in a few samples) are most suitably represented by a modeling error term [3]. When models fails to extract the dominant features in data (present in most samples), such models *under-fit* data. An optimal model strives to compromise between the error due to random fluctuations by bias and variance trade-off [3]. Of course, the order of the model can be increased as more data becomes available. While statistics-based models may provide more accurate predictions for a given set of tuning data (i.e. prior observations), the proponents of such models state that they do not capture the true physics [4]. Thus the approach adopted by the author is to combine both physics-based modeling with statistics-based techniques to obtain a “best” model which provides accurate predictions while satisfying the underlying physical laws.

It is worthwhile to point out the fact that recent availability of cost-effective high performance computing systems now enables numerical simulations with extremely high resolution computational models (with fewer computationally-imposed simplifying assumptions). Such simulations capture wide ranging spatial and temporal scales of a physical phenomenon. Furthermore, rapid development and deployment of modern sensing technologies permit gathering high-resolution (spatio-temporal) data, increasing in tandem with the resolution of numerical models [5]. This fact prompted the initiative to dynamically blend observational data into running computer simulations to reduce uncertainty and devise targeted data gathering ventures through symbiotic feedback from simulation and measurement processes [6]. To reap the benefit from the petascale computing platforms (and future exascale systems [7]), concomitant investment must be made for algorithm design of uncertainty quantification and data assimilation which scale up to a large number of processors/cores. In this thesis, some initiative in this direction is undertaken.

1.2 Brief Review of Filtering Algorithms for Sequential Data Assimilation

The process of blending observational data in numerical models is known as data assimilation [8–10]. Data assimilation aims to find the best state or state and parameter estimates using (a) the numerical model and (b) the observational data. There are two ways to pose this problem. Variational assimilation methods [8, 11], such as 3DVAR and 4DVAR, provide estimates that are close to the observations and prior guesses of the initial conditions and parameters while satisfying the model equations [10]. The

implicit assumption is that the model is corrupted by errors arising, for instance, due to neglected physics, linearization and discretization in space and time. The optimal model state obtained through the minimization process which fits best the observational data. The state estimate is then used as an initial condition for the numerical model to generate forecasts. Minimization methods require the repeated evaluation of the gradient of the cost function, which becomes computationally intensive for large-scale numerical models. Using the adjoint method, the gradient of the cost function can be computed at the expense of few function evaluations, enhancing the computational efficiency of the optimization process. The comprehensive theoretical and computational exposition of adjoint methods associated with linear and nonlinear models is available in [8, 12, 13].

A major disadvantage of variational methods is the necessity of obtaining a tangent linear adjoint operator of the models and forward observation operators, which may not be straightforward to obtain and implement for complex and large-scale problems [14, 15]. Another approach to data assimilation is sequential Bayesian estimation method. Operational forecasting is inherently sequential. Upon arrival of data, forecasts are issued continuously. In such dynamic environment, it is natural to blend data to models sequentially. Kalman filter (KF) has become a popular tool for sequential data assimilation problems involving linear systems [16, 17]. It optimally estimates the state of linear systems in the presence of additive Gaussian model and measurement noise when the observational data is linearly related to the state of the system. KF can be extended to weakly nonlinear and non-Gaussian systems by linearizing the model and measurement operators, leading to the popular (but no longer optimal) extended Kalman filter (EKF) [10, 18–20]. The major limitation of KF and EKF stems from the fact that the state evolution is assumed to be Gaussian or approximately Gaussian. Such approximation may lead to poor performance of EKF for strongly nonlinear dynamical

systems [21, 22]. To alleviate such problem, the unscented transform [23] is used to propagate the statistical moments of the state vector through the full nonlinear model operator leading to the so-called unscented Kalman filter (UKF) [24–27]. Using a set of carefully chosen sampling points (so-called sigma points), UKF estimates the first and second order moments of the state vector correct to the second order Taylor series expansion of the model operator.

A wide range of Monte Carlo based filtering algorithms have been developed to tackle more general classes of nonlinear systems perturbed by non-Gaussian noise. The ensemble Kalman filter (EnKF) is one of the most widely used filtering algorithms based on Monte Carlo sampling technique [10, 28–33]. The usefulness of EnKF has been successfully demonstrated for numerous applications of data assimilation problems (e.g. [10, 30–32]). EnKF is a computationally efficient technique as it adopts the linear update (assimilation) step of KF. In numerous applications, EnKF resolves the issue of poor error covariance evolution commonly encountered in EKF [34]. In the context of structural dynamics, EnKF has recently been introduced as a non-parametric identification tool by Ghanem and Ferro [31]. Although widely used, some publications have reported rather poor performance of EnKF in certain applications involving highly non-Gaussian posterior probability density functions [32, 33]. To alleviate such problems, an extension of EnKF has been proposed that utilizes a weighted sum of Gaussian probability density functions to represent the non-Gaussian system state [35, 36].

The most general nonlinear filtering algorithm is the so-called particle filter (PF) [9, 32, 37–44]. In the framework of Bayesian inference, PF can handle general form of nonlinearities and non-Gaussian noises in the model and measurement operators. In PF, the ensemble members (so-called particles) are properly weighted based on the observational data. In general, PF demands larger ensembles compared to EnKF leading

to higher computational costs, but more effectively tackles strongly non-Gaussian problems. The problem related to the requirement of larger ensembles is partly mitigated using the so-called resampling technique [9, 40, 41, 45]. In the context of nonlinear dynamics, PF was applied to identify a nonlinear stiffness parameter of a Duffing oscillator [41–43]. Kivman [32] compared the applicability of the EnKF and PF for parameter estimation in the Lorenz model. Ching, Beck and Porter [46] compared the performance of EKF and PF for the state and parameter estimation of various nonlinear systems. Recently, the author studied the performance of EKF, EnKF and PF for joint state and parameter estimation of a Duffing oscillator undergoing chaotic motion [47]. To improve the performance of PF, hybrid filtering algorithms which combine EKF, UKF or EnKF with PF have also been proposed leading to the so-called extended Kalman and unscented Kalman particle filters (e.g. [27]), the predictor-corrector filter [48] and the weighted ensemble Kalman filter (WEnKF) [49].

1.3 Bayesian Inference for Estimation of Static Parameters

A popular approach to solving the parameter estimation problem using sequential data assimilation involves augmenting the state vector with the unknown system parameters. The new augmented state vector is estimated using one of the above-mentioned filtering techniques. In this approach, the parameter space is only explored at the initialization of the algorithm [50]. In certain cases, the algorithm may become ineffective; after a few iterations the probability distribution of the parameters may degenerate to a single delta Dirac function [51–54] leading to incorrect estimates. To alleviate this problem, several

authors have proposed various techniques, albeit with limited success (e.g. [47, 55–57]).

In this thesis, another approach is proposed to handle the static parameter estimation problem which avoids augmenting the state vector with the unknown parameters. Instead, the conditional distribution of the state given the noisy measurements is first estimated using an appropriate nonlinear filtering technique. Markov Chain Monte Carlo (MCMC) sampling technique [58–60] is then used to generate samples from the posterior distribution of the unknown static parameters based on the conditional state estimates. MCMC approaches provide samples that are generated using the previous sample or set of samples with the aid of a Markov chain [58–60]. MCMC methods are rooted in the Metropolis algorithm [61] which was initially employed by physicists to compute complex integrals by rewriting them as expectations of some distributions, with MCMC providing samples from these distributions. It is worth mentioning that the framework of Probabilistic Graphical Models (PGMs) [62], although not used by the author, may be useful in formulating this problem as it provides mechanisms for analyzing complex interconnected systems.

1.4 Parallel Data Assimilation Algorithms

Parallel filtering algorithms with large-scale models are of increasing interest due to recent availability of cost-effective high performance computing systems [63]. The aim of these algorithms is to estimate the state of a large-scale system with a given numerical model and a set of observations. The algorithms also compute the error statistics in the state estimates. For large-scale systems, KF and EKF filtering techniques require (a) large computational resources to propagate the error covariance matrix and (b) large amount of memory to store the error covariance matrix. These requirements limit the

application of these filtering algorithms to high-dimensional state space models compared to EnKF which partly alleviates these problems. PF algorithms demand (a) large number of ensemble members (compared to EnKF) for good performance and (b) a resampling step of the posterior joint pdf which is again computationally expensive for high-dimensional models [10, 64, 65]. Although parallel algorithms for various filtering techniques (i.e. KF [66] and PF [67]) are reported in the literature, EnKF has received the most significant attention [68–76].

The development of EnKF paved the way for the application of sequential Bayesian inference technique for large-scale problems. However, filters based on EnKF still incurs significant computational expense due to (a) high computational cost needed to march the ensemble of states of large-scale models in time and (b) large memory requirement to store the ensemble matrix (containing the various realizations of the system state) and other matrices needed to update the ensemble members when data becomes available. This fact drives the motivation to parallelize the filtering algorithms to use the large memory and computational capability of distributed memory computing systems.

Parallelization interfaces, such as message passing interface (MPI) [77], provide the mechanism to distribute the ensemble matrix across the computing nodes, in order to use the entire pool of available distributed memory and central processing units (CPUs)/cores in the high-performance computing (HPC) system. EnKF has been parallelized either by (a) delegating the task of propagating and updating a certain number of ensemble members or (b) by partitioning the spatial domain into a number of subdomains and each processor would then carry out the propagation and analysis of the entire ensemble of the state vector pertaining to one subdomain. For simplicity, the two parallelization techniques will be denoted by *mode decomposition* and *domain decomposition* [71], respectively.

The most straightforward strategy to parallelize the forecast step is to use mode decomposition whereby the ensemble states are distributed over the available pool of computing nodes [68, 71, 73]. Each node would then propagate one or a few members of the ensemble states concurrently, taking advantage of the *embarrassing* (straightforward) parallelism [78] of the forecast step, as alluded to by Evensen [34]. When the available data are spatially or temporally sparse, the propagation of ensemble members in time is normally the most computationally intensive step of the EnKF filtering algorithm, especially for large-scale models [73]. Therefore parallelizing the forecast step will yield a highly scalable filtering algorithm according to Amdahl's law [79], reducing the time needed to solve a given data assimilation as more processors become available. This approach demands that each processor contains a few samples of the entire state vectors in its accessible memory. For the analysis (update) step, the matrix-vector operations incur significant inter-processor communications for collective operations. This fact seriously degrades the parallel scalability of the EnKF algorithm, especially when large amounts of observational data is assimilated [71]. Another bottleneck for this approach pertains to the memory requirement. The size of the model is restricted by the available memory as each processor must have access to at least one copy of the state vector.

Domain decomposition methods have received widespread attention to tackle large-scale (linear and nonlinear) systems arising from the spatial discretization of PDEs. In domain decomposition algorithms, such as iterative substructuring, exhibit scalable performance (for large number of processors) through data-parallelism. In the EnKF formalism, a common practice is to assign each processor a single subdomain [70, 71, 73]. Thus, each processor owns a copy of the entire ensemble of the state vector of the allocated subdomain. The domain decomposition approach circumvents the bottleneck due to the limited memory of the individual computing nodes arising in the aforementioned

mode decomposition based algorithm. For efficient parallel implementation of this algorithm, the analysis step adopts a simplification whereby spatial correlations among state elements and observations are ignored after some distance in space [69]. This assumption permits the efficient assimilation of measurements in the domain decomposition approach. In general, this approach is preferred if a domain decomposition based solver is already available for the forecast step [73].

1.5 Thesis Statement

For a low-resolution (small size) model that fits into the local memory of a compute node, Monte Carlo sampling method becomes *embarrassingly* parallel. In this scenario, each processor/core can handle a single Monte Carlo realization. For high-resolution models, a large number of processors now engage to tackle just a single Monte Carlo sample. In this case, Monte Carlo simulation loses the aforementioned parallelism. This fact imposes restriction on the number of Monte Carlo samples for high-resolution models.

For large-scale systems, a major limitation of EnKF is the requirement of a large ensemble size which is often computationally impractical or infeasible. For example, it has been shown [69] that for ensemble sizes less than or equal to 100, EnKF fails to obtain accurate statistics of the model state for a large-scale system. In order to avoid Monte Carlo sampling, Saad and Ghanem [80] have proposed a procedure to address this issue by using a polynomial chaos expansion (PCE) [1, 81] representation of the stochastic state vector. This representation is more efficient at capturing the model state statistics for accurate forecast and analysis of observations leading to Polynomial Chaos-based Kalman filter (PCKF). In this thesis, the author formulates a parallel assimilation

(update) algorithm and its scalable implementation using PCKF that makes use of a polynomial chaos based scalable domain decomposition solver [82–87].

1.6 Thesis Contributions

The novel contributions in this thesis can be summarized as follows:

- **Application of Bayesian Inference to the Flutter Margin Method:**

In Chapter 3, a probabilistic framework based on Bayesian inference that offers a rational treatment of the uncertainties in the vibration data obtained from flight tests is developed in order to provide reliable estimates of the flutter speed. For the numerical experiments conducted in this thesis, the method provides robust estimates of the flutter speed vis-a-vis the conventional method. A proof of concept has been published in which the flutter speed of a simple airfoil numerical model was estimated [88–90].

- **Estimation of Time-Invariant Parameters of Noisy Nonlinear Oscillatory Systems:**

The inverse problem of estimating time-invariant parameters of a nonlinear system exhibiting noisy oscillation is considered in Chapter 4. Firstly, a MCMC algorithm is developed for the time-invariant parameter estimation which uses a non-Gaussian filter, namely the EnKF for state estimation. Secondly, a recently proposed PF (that uses the EnKF to provide the proposal density), has been adapted for combined state and parameter estimation. This work has been submitted for publication in an international journal [91].

- **Static Parameter Estimation of a Non-linear Aeroelastic System using Wind Tunnel Data:**

In Chapter 5, the Bayesian inference is applied for parameter estimation of a self-sustaining aeroelastic oscillator. The system consists of an elastically mounted rigid wing on a rig fixed in a wind tunnel. For certain flow conditions in the transitional Reynolds number regime, the wing absorbs energy from the flow and settles in a stable limit cycle oscillation (LCO). The LCO originates from laminar boundary layer separation which leads to negative aerodynamic damping at small angles of attack. The proposed methodology uses EKF for the associated state estimation in order to estimate the aeroelastic parameters. To generate samples from the posterior joint probability density functions of the aeroelastic parameters, MCMC simulation technique is used. The findings of this investigation are published in the proceedings of ASME-FES [92] and IFASD [93] and in an international journal [94].

- **Data Assimilation for Large-Scale Computational Models:**

A parallel Bayesian estimation algorithm is developed in Chapter 6 which integrates PCKF [80] and a polynomial chaos based domain decomposition solver [82, 84–87, 95] for large-scale computational models. The domain decomposition solver is used to propagate the polynomial chaos (PC) coefficients of the state vector forward, providing an efficient forecast step. Whenever measurements are available, the analysis step updates the PC coefficients in parallel by taking advantage of the available spatial decomposition of the state vector. The distributed implementation of the solver and update (analysis) step are carried out using MPI. For numerical illustrations, the steady-state stochastic Poisson and advection-diffusion problems with non-Gaussian diffusivity are considered. Furthermore,

the parallel scalability of this algorithm is investigated on high performance computing platforms. Preliminary findings of this investigation are published in the proceedings of AIAA-SDM [96].

- **Parallel Solver for Non-Symmetric Stochastic Systems Based on Preconditioned Biconjugate Gradient Stabilized Method:**

In Appendix D.4, a parallel preconditioner and parallel algorithm for the matrix-vector product involving the Schur complement matrix are formulated for the distributed implementation of the preconditioned biconjugate gradient stabilized method for spectral stochastic finite element method. This solver is used in this thesis to tackle the steady-state stochastic advection-diffusion problem.

1.7 Thesis Overview

The thesis is organized in the following manner. Chapter 1 presents an introduction to the problem of Bayesian inference for engineering systems and provides a relevant literature review. Chapter 2 introduces the concepts of probability theory in the context of Bayesian inference. It also reviews some of the widely used nonlinear filtering techniques to tackle the sequential data assimilation problem. A description of the Markov Chain Monte Carlo sampling technique is provided in Chapter 3 with some illustrative applications including the problem of aeroelastic flutter speed estimation. The problem of static parameter estimation is formulated for dynamical systems in the context of Bayesian inference in Chapter 4. Chapter 5 describes the problem of nonlinear aeroelastic parameter estimation using wind tunnel data. Chapter 6 details a parallel filtering algorithm in order to tackle the data assimilation problem for large-scale models in com-

putational stochastic mechanics. The thesis concludes with summary of contributions and possible future research directions.

Chapter 2

Bayesian framework for state and/or parameter estimation

2.1 Introduction

This chapter introduces the underlying concepts for all probabilistic state and joint state and parameter estimation problems. The concept of conditional probability is central to those problems. A conditional probability describes the interdependence between different random events and allows one to use the statistical information provided by observations (or measurements) in making an inference regarding unobserved variables. In the case of state or state and parameter estimation of dynamical systems, the conditional probability links the information provided from the partial (sparse) and noisy measurements of the state vector to the entire state and parameter vectors of the system.

2.2 Bayesian Inference

2.2.1 Statistical Modeling

The state and parameters of a physical system are never known exactly. One can thus characterize these variables probabilistically in terms of random variables. Let us first recall some definitions and results regarding the characterization of random variables.

Definition 2.1 (Probability Density).

Denoting by $X \leq x$ the event of “the realization of the random variable X is less than or equal to x ”, the (cumulative) distribution function associated with the random variable X is denoted as

$$F_X(x) = P(X \leq x) \quad (2.1)$$

and the probability density (subject to existence) is defined by

$$p_X(x) = \frac{dF_X(x)}{dx} . \quad (2.2)$$

The probability density (or distribution) function characterizes completely the statistical knowledge of the random variable X .

Definition 2.2 (Joint Probability Density).

Let X and Y be two random variables. The joint probability density of X and Y by [97] is denoted as

$$p_{XY}(x, y) = \frac{\partial F_{XY}(x, y)}{\partial x \partial y} \quad (2.3)$$

where

$$F_{XY}(x, y) = P(\{X \leq x\} \cap \{Y \leq y\}) . \quad (2.4)$$

The concept of joint probability density of two random variables can be extended to

a probability density for a random vector $\mathbf{X} = [X_1, \dots, X_n]$. The pdf of \mathbf{X} is defined as the joint probability density of its elements denoted by $p(\mathbf{x}) = p(x_1, \dots, x_n)$. This is referred to as a multi-dimensional probability density [97].

Definition 2.3 (Marginal Probability Density).

Let X and Y be two random variables (or vectors) and $p_{XY}(x, y)$ their joint probability density. Then

$$p_X(x) = \int p_{XY}(x, y) dy \quad (2.5)$$

where $p_X(x)$ is the marginal probability density of X [97].

Definition 2.4 (Conditional Probability).

The probability of the occurrence of an event A conditional upon the occurrence of another event B is given by [97]

$$P(A | B) = \frac{P(A \cap B)}{P(B)}. \quad (2.6)$$

$P(A | B)$ characterizes the dependence of event A upon event B . If A and B are independent, $P(A | B) = P(A)$ [97].

Definition 2.5 (Conditional Probability Density).

The pdf of a random variable (or vector) X conditional upon another random variable (or vector) Y is given by [97]

$$p_X(x|y) = \frac{p_{XY}(x, y)}{p_Y(y)} = \frac{p_{XY}(x, y)}{\int p_{XY}(x, y) dx}. \quad (2.7)$$

The conditional probability density $p_X(x|y)$ represents the statistical knowledge of X given an observation of Y . It represents the basic tool of statistical estimation: if the

function $p_X(x|y)$ is known, an observation (or measurement) of the variable Y would provide the statistical knowledge of the variable X .

Definition 2.6 (Multivariate Normal Probability Density Function).

The pdf of a random vector of dimension n is said to be Gaussian (or Normal) with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ if it has the following form [97]

$$p_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \quad (2.8)$$

where $|\cdot|$ is the determinant of a matrix and superscript T denotes the transpose operator. A Normal distribution of a random vector with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ will be denoted by $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

2.2.2 Bayesian Framework for Estimation

Bayes' rule allows one to calculate the function $p_X(x|y)$ as follows

Theorem 2.1 (Bayes' Theorem).

Let X and Y be two random variables, then [10, 98–100]

$$p_X(x|y) = \frac{p_Y(y|x) p_X(x)}{p_Y(y)} = \frac{p_Y(y|x) p_X(x)}{\int p_Y(y|x) p_X(x) dx}. \quad (2.9)$$

In order to simplify the notations, $p(x)$ and $p(x|y)$ will denote $p_X(x)$ and $p_X(x|y)$ respectively.

Bayes' theorem provides the mathematical basis to the solution of the problem of estimation. Suppose a discretized system S can be described by the vector \mathbf{x} containing the unknown state or state and parameter variables. \mathbf{x} will be referred to as the state vector. Let us assume that a number of measurements of the system S are available,

being represented by a vector \mathbf{d} . Using Bayes' rule, one can write

$$p(\mathbf{x}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{d}|\mathbf{x})p(\mathbf{x})d\mathbf{x}} \quad (2.10)$$

where

- $p(\mathbf{x}|\mathbf{d})$ is the posterior probability density function of \mathbf{x} [10, 101]. It corresponds to the probability density of the parameter of interest, namely \mathbf{x} , given the measurements \mathbf{d} . $p(\mathbf{x}|\mathbf{d})$ therefore reflects all of the information provided by the measurements.
- $p(\mathbf{d}|\mathbf{x})$ is the likelihood function [10, 101]. It is derived directly from the measurement equation that relates the measurements to the state vector and the noise process that corrupts them. In most scenarios, this function can be evaluated without much conceptual or computational difficulty.
- $p(\mathbf{x})$ is the prior probability density function of \mathbf{x} [10, 101]. It represents the knowledge of the state vector prior to the assimilation of the measurements.

The main difficulty of implementing Bayes' rule in practice is the evaluation of the integral $\int p(\mathbf{d}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$. In the general case, this integral can not be obtained analytically and its numerical approximation becomes increasingly difficult to obtain as \mathbf{x} increases in dimension [102]. The case where \mathbf{d} depends linearly on \mathbf{x} and the measurement noise is additive and Gaussian in nature permits an analytical solution, in which case the posterior pdf is also Gaussian. For other (nonlinear and non-Gaussian) scenarios, the solution can only be approximated. The difficulty then lies in finding the best approximation technique to apply which is most appropriate for the problem at hand.

2.2.3 Estimators

The posterior pdf $p(\mathbf{x}|\mathbf{d})$ provides the complete statistical knowledge of \mathbf{x} . In a practical application, it is sometimes preferable to provide a value for \mathbf{x} that estimates its true value. The purpose of an estimator is to provide an estimate $\hat{\mathbf{x}}$ of \mathbf{x} . The two most common estimators obtained from the posterior pdf $p(\mathbf{x}|\mathbf{d})$ are [9]:

1. Minimum mean-square error estimator (MMSE) [102] or minimum variance estimator given by

$$\hat{\mathbf{x}}_{MMSE} = \int \mathbf{x} p(\mathbf{x}|\mathbf{d}) d\mathbf{x} \quad (2.11)$$

$\hat{\mathbf{x}}_{MMSE}$ is the posterior mean of \mathbf{x} . This estimator has the property of having the minimum mean square error of all estimators. Specifically, $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{MMSE}$ minimizes $E \left[(\hat{\mathbf{x}} - \mathbf{x})^T (\hat{\mathbf{x}} - \mathbf{x}) \right]$.

2. Maximum a posteriori estimator (MAP) [102] is given by

$$\hat{\mathbf{x}}_{MAP} = \arg \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{d}) \quad (2.12)$$

$\hat{\mathbf{x}}_{MAP}$ is the mode of the posterior distribution of \mathbf{x} . The maximum-likelihood (ML) estimate (which maximizes the likelihood function) can be considered as a MAP-estimate with a flat (uninformative) prior for \mathbf{x} .

2.2.4 Dynamical systems

The term dynamical system is used to denote a system whose state evolves with time. This thesis will deal with the discrete time representation of such systems. It will be assumed that the evolution of the state is random. Under this assumption, the problem

of state estimation bears resemblance with the problem addressed in the preceding subsections. Let us represent the ensemble of the states of the system on the time interval $[t_0, t_k]$ by \mathbf{X}_k :

$$\mathbf{X}_k = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k] \quad (2.13)$$

where \mathbf{x}_i is the state vector at time $t = t_i$ for $i = 0, 1, \dots, k$.

Similarly, one can group the measurement vectors on the same time interval:

$$\mathbf{D}_k = [\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_k]. \quad (2.14)$$

Bayes' rule can then be applied to derive the posterior pdf of \mathbf{X}_k :

$$p(\mathbf{X}_k | \mathbf{D}_k) = \frac{p(\mathbf{D}_k | \mathbf{X}_k) p(\mathbf{X}_k)}{\int p(\mathbf{D}_k | \mathbf{X}_k) p(\mathbf{X}_k) d\mathbf{X}_k}. \quad (2.15)$$

The following two assumptions permit the recursive evaluation of $p(\mathbf{X}_k | \mathbf{D}_k)$ in Eq. (2.15) in terms of $p(\mathbf{X}_{k-1} | \mathbf{D}_{k-1})$:

1. The successive states of the system form a Markov chain of order one. In other terms, the state at time t_k depends only on the state at the preceding instant:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k) \quad (2.16)$$

with \mathbf{v}_k being a vector white noise that leads to the random evolution of the state.

It is sometimes referred to as modeling (process) noise.

2. The measurement vectors are statistically independent at different time instants

are independent and their pdfs, conditional on the state vector, is given by

$$p(\mathbf{d}_k | \mathbf{x}_k, \mathbf{D}_{k-1}) = p(\mathbf{d}_k | \mathbf{x}_k) \quad (2.17)$$

Under these assumptions [49, 103]

$$\begin{aligned} p(\mathbf{X}_k | \mathbf{D}_k) &= p(\mathbf{x}_k, \mathbf{X}_{k-1} | \mathbf{d}_k, \mathbf{D}_{k-1}) \\ &= \frac{p(\mathbf{x}_k, \mathbf{X}_{k-1}, \mathbf{d}_k, \mathbf{D}_{k-1})}{p(\mathbf{d}_k, \mathbf{D}_{k-1})} \\ &= \frac{p(\mathbf{x}_k, \mathbf{d}_k | \mathbf{X}_{k-1}, \mathbf{D}_{k-1}) p(\mathbf{X}_{k-1}, \mathbf{D}_{k-1})}{p(\mathbf{d}_k | \mathbf{D}_{k-1}) p(\mathbf{D}_{k-1})} \\ &= \frac{p(\mathbf{d}_k | \mathbf{x}_k, \mathbf{X}_{k-1}, \mathbf{D}_{k-1}) p(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{D}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{D}_{k-1}) p(\mathbf{D}_{k-1})}{p(\mathbf{d}_k | \mathbf{D}_{k-1}) p(\mathbf{D}_{k-1})} \\ &= \frac{p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{X}_{k-2}, \mathbf{D}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{D}_{k-1}) \cancel{p(\mathbf{D}_{k-1})}}{p(\mathbf{d}_k | \mathbf{D}_{k-1}) \cancel{p(\mathbf{D}_{k-1})}} \\ &= \frac{p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{D}_{k-1})}{p(\mathbf{d}_k | \mathbf{D}_{k-1})} \\ &= p(\mathbf{X}_{k-1} | \mathbf{D}_{k-1}) \frac{p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{d}_k | \mathbf{D}_{k-1})}. \end{aligned} \quad (2.18)$$

The above equation provides a recursive relationship between the posterior pdf at time t_k and time t_{k-1} . In filtering, only the estimate of the current state \mathbf{x}_k is of interest. The marginal probability density of \mathbf{x}_k is calculated by:

$$p(\mathbf{x}_k | \mathbf{D}_k) = \int p(\mathbf{X}_k | \mathbf{D}_k) d\mathbf{X}_{k-1}. \quad (2.19)$$

Using the recursive relationship in Eq. (2.18), the above equation can be written as [49, 103]

$$p(\mathbf{x}_k | \mathbf{D}_k) = \frac{p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{D}_{k-1})}{p(\mathbf{d}_k | \mathbf{D}_{k-1})}. \quad (2.20)$$

This is an important relationship for any filtering (i.e. recursive estimation) problem

satisfying the aforementioned two assumptions. From a computational point of view, the two main steps in filtering can be summarized as follows:

1. Forecast Step (Chapman-Kolmogorov equation) [104]:

$$p(\mathbf{x}_k | \mathbf{D}_{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{D}_{k-1}) d\mathbf{x}_{k-1} \quad (2.21)$$

2. Analysis Step (Bayes' equation) [10, 98–100]:

$$p(\mathbf{x}_k | \mathbf{D}_k) = \frac{p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{D}_{k-1})}{p(\mathbf{d}_k | \mathbf{D}_{k-1})}, \quad (2.22)$$

where

$$p(\mathbf{d}_k | \mathbf{D}_{k-1}) = \int p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{D}_{k-1}) d\mathbf{x}_k. \quad (2.23)$$

In the general case, the above two integrals do not permit analytical solutions. This is the main difficulty encountered in the application of recursive estimation. Only for special cases (e.g. linear model and measurement equations with additive Gaussian noise), a closed-form expression for $p(\mathbf{x}_k | \mathbf{D}_k)$ is available using the Kalman filter algorithm [16, 17]. Otherwise, one takes recourse the following two strategies:

- Use an analytical approximation by (a) linearizing the model and measurement equations with respect to the state and noise variables, leading to the extended Kalman filter (EKF) [10, 18–20], or (b) assuming the state variables to be Gaussian and using deterministic sampling methods to propagate the state variables through the nonlinear model equation which result in the unscented Kalman filter (UKF) [24–27].
- Use sampling methods to approximate $p(\mathbf{x}_k | \mathbf{D}_k)$, resulting in either the Ensemble

Kalman Filter (EnKF) [10, 28–33] or Particle Filter (PF) [9, 32, 37–44].

2.3 Linear Filtering: The Kalman Filter

The Kalman filter [10, 19, 20] and its variants propagate a Gaussian probability density for the state. At each step of the filter, two quantities are provided which together characterize the Gaussian conditional distribution $p(\mathbf{x}_k | \mathbf{D}_k)$. These two quantities are (a) the mean vector denoted by $\hat{\mathbf{x}}_k$ and (b) the covariance matrix denoted by \mathbf{P}_k . The Kalman filter is only applicable to linear and Gaussian problems in which the following assumptions must be satisfied:

- The model and measurement equations are linear and the model and measurement noise processes are additive and Gaussian:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{q}_k , \quad (2.24)$$

$$\mathbf{d}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\epsilon}_k , \quad (2.25)$$

$$\mathbf{q}_k \sim \mathcal{N}(\mathbf{q}_k; \mathbf{0}, \mathbf{Q}_k) , \quad (2.26)$$

$$\boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{\epsilon}_k; \mathbf{0}, \boldsymbol{\Gamma}_k) , \quad (2.27)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, \mathbf{F} is the transition matrix of the dynamical model and $\mathbf{d} \in \mathbb{R}^m$ is the measurement vector which relates to the state vector by the measurement model matrix \mathbf{H} .

- The probability density of \mathbf{x}_0 is Gaussian:

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, \mathbf{P}_0) . \quad (2.28)$$

In a probabilistic framework, the model and measurement equations (Eqns. 2.24-2.25) translate to

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{F}_k \mathbf{x}_k, \mathbf{Q}_k) , \quad (2.29)$$

$$p(\mathbf{d}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{d}_k; \mathbf{H}_k \mathbf{x}_k, \mathbf{\Gamma}_k) . \quad (2.30)$$

Under the above assumptions, the probability density $p(\mathbf{x}_k|\mathbf{D}_{k-1})$ and $p(\mathbf{x}_k|\mathbf{D}_k)$ are Gaussian at any time instance t_k and the Kalman filter provides the exact solution to the problem of recursive Bayesian estimation.

In order to identify the Gaussian nature of the state vector, the following notation is used

$$p(\mathbf{x}_k|\mathbf{D}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{P}_k^f) , \quad (2.31)$$

$$p(\mathbf{x}_k|\mathbf{D}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^a, \mathbf{P}_k^a) , \quad (2.32)$$

to denote the prior and posterior probability density functions of the state at time step t_k , respectively. The prior Gaussian pdf is obtained by forecasting the first and second order statistics of the state vector from time step t_{k-1} to t_k using the dynamical model, hence the use of the superscript f to denote *forecast*. The superscript a denotes *analysis* as the posterior distribution of the state at time step t_k requires the analysis of the observation available at time step t_k .

Assuming that the posterior distribution of the state vector at time step t_{k-1} is Gaussian, i.e. $p(\mathbf{x}_{k-1}|\mathbf{D}_{k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}^a, \mathbf{P}_{k-1}^a)$, it can be shown that the prior distribution of the state at time step t_k is also Gaussian, i.e. $p(\mathbf{x}_k|\mathbf{D}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{P}_k^f)$.

Eq. (2.21) and Eq. (2.29) give

$$\begin{aligned}
p(\mathbf{x}_k | \mathbf{D}_{k-1}) &= \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{D}_{k-1}) d\mathbf{x}_{k-1} \\
&= \int \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1} \mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}^a, \mathbf{P}_{k-1}^a) d\mathbf{x}_{k-1} \\
&= \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^a, \mathbf{F}_{k-1}^T \mathbf{P}_{k-1}^a \mathbf{F}_{k-1} + \mathbf{Q}_{k-1}) \\
&= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{P}_k^f) .
\end{aligned} \tag{2.33}$$

The third equality in the above equation is obtained using Theorem A.6 of Appendix A.3.

From Eq. 2.33, one obtains the forecast mean and error covariance matrix which constitutes the forecast step of the Kalman filter.

- Forecast step:

$$\hat{\mathbf{x}}_k^f = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^a , \tag{2.34}$$

$$\mathbf{P}_k^f = \mathbf{F}_{k-1}^T \mathbf{P}_{k-1}^a \mathbf{F}_{k-1} + \mathbf{Q}_{k-1} . \tag{2.35}$$

One can also show that the conditional pdf of the state at time t_k given an observation at time t_k is also Gaussian, i.e. $p(\mathbf{x}_k | \mathbf{D}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^a, \mathbf{P}_k^a)$. According to Theorem A.5

of Appendix A.3, the joint distribution of \mathbf{x}_k and \mathbf{d}_k given \mathbf{D}_{k-1} is

$$p(\mathbf{x}_k, \mathbf{d}_k | \mathbf{D}_{k-1}) = p(\mathbf{d}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{D}_{k-1}) \quad (2.36)$$

$$= \mathcal{N}(\mathbf{d}_k; \mathbf{H}_k \mathbf{x}_k, \Gamma_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{P}_k^f) \quad (2.37)$$

$$= \mathcal{N} \left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{d}_k \end{pmatrix}; \begin{pmatrix} \hat{\mathbf{x}}_k^f \\ \mathbf{H}_k \hat{\mathbf{x}}_k^f \end{pmatrix}, \begin{pmatrix} \mathbf{P}_k^f & \mathbf{P}_k^f \mathbf{H}_k^T \\ \mathbf{H}_k \mathbf{P}_k^f & \mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \Gamma_k \end{pmatrix} \right). \quad (2.38)$$

By Theorem A.4, the conditional distribution of \mathbf{x}_k is:

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{d}_k, \mathbf{D}_{k-1}) &= p(\mathbf{x}_k | \mathbf{D}_k) \\ &= \mathcal{N} \left(\mathbf{x}_k; \hat{\mathbf{x}}_k^f + \mathbf{P}_k^f \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \Gamma_k \right]^{-1} (\mathbf{d}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^f), \right. \\ &\quad \left. \mathbf{P}_k^f - \mathbf{P}_k^f \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \Gamma_k \right]^{-1} \mathbf{H}_k \mathbf{P}_k^f \right) \\ &= \mathcal{N} \left(\mathbf{x}_k; \hat{\mathbf{x}}_k^f + \mathbf{K}_k (\mathbf{d}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^f), \mathbf{P}_k^f - \mathbf{K}_k \left[\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \Gamma_k \right]^{-1} \mathbf{K}_k^T \right) \\ &= \mathcal{N} \left(\mathbf{x}_k; \hat{\mathbf{x}}_k^f + \mathbf{K}_k (\mathbf{d}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^f), (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f \right) \\ &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^a, \mathbf{P}_k^a) \end{aligned} \quad (2.39)$$

where

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \Gamma_k \right]^{-1} \quad (2.40)$$

is the Kalman gain matrix.

Eq. 2.39 provide the analysis mean and error covariance matrix which constitutes the analysis step of the Kalman filter.

- Analysis step:

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{\Gamma}_k \right]^{-1}, \quad (2.41)$$

$$\hat{\mathbf{x}}_k^a = \hat{\mathbf{x}}_k^f + \mathbf{K}_k \left(\mathbf{d}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^f \right), \quad (2.42)$$

$$\mathbf{P}_k^a = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^f. \quad (2.43)$$

2.4 Extended Kalman filter

The extended Kalman filter [10, 19, 20] provides a solution to the sequential Bayesian inference problem in which the model or measurement equations are nonlinear. In general, the discrete state-space representation of a nonlinear system is given by

$$\mathbf{x}_{k+1} = \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k), \quad (2.44)$$

$$\mathbf{d}_k = \mathbf{h}_k(\mathbf{x}_k, \epsilon_k). \quad (2.45)$$

Here \mathbf{x}_k is the state vector, \mathbf{g}_k is the discrete nonlinear model operator, \mathbf{d}_k is the measurement vector which relates to the state by the nonlinear measurement operator \mathbf{h}_k . \mathbf{q}_k and ϵ_k are independent Gaussian random vectors with zero mean and covariance matrices \mathbf{Q}_k and $\mathbf{\Gamma}_k$, respectively.

For the extended Kalman filter, the model and measurement equations are linearized around the current state. The linearized problem can be solved using the Kalman filter described in the preceding section. Linearizing \mathbf{h}_k about $\mathbf{x}_k = \hat{\mathbf{x}}_k^f$ and $\epsilon_k = \mathbf{0}$, one

obtains

$$\begin{aligned} \mathbf{h}_k(\mathbf{x}_k, \epsilon_k) &\approx \mathbf{h}_k(\mathbf{x}_k^f, \epsilon_k = 0) + \left. \frac{\partial \mathbf{h}_k(\mathbf{x}_k, \epsilon_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \mathbf{x}_k^f, \epsilon_k = 0} (\mathbf{x}_k - \mathbf{x}_k^f) \\ &\quad + \left. \frac{\partial \mathbf{h}_k(\mathbf{x}_k, \epsilon_k)}{\partial \epsilon_k} \right|_{\mathbf{x}_k = \mathbf{x}_k^f, \epsilon_k = 0} \epsilon_k \end{aligned} \quad (2.46)$$

where $\frac{\partial \mathbf{h}_k(\mathbf{x}_k, \epsilon_k)}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{h}_k(\mathbf{x}_k, \epsilon_k)}{\partial \epsilon_k}$ describe the Jacobian matrices of $\mathbf{h}_k(\mathbf{x}_k, \epsilon_k)$ with respect to \mathbf{x}_k and ϵ_k respectively. It is further assumed that \mathbf{x} has a Gaussian prior probability density function (pdf) given by $\mathbf{x}_k \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{P}_k^f)$. In the analysis step, EKF estimates the conditional mean $\hat{\mathbf{x}}_k^a$ and covariance \mathbf{P}_k^a of \mathbf{x}_k given the measurement vector \mathbf{d}_k :

- Analysis step:

$$\mathbf{C}_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}_k, \epsilon_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \mathbf{x}_k^f, \epsilon_k = 0}, \quad (2.47)$$

$$\mathbf{D}_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}_k, \epsilon_k)}{\partial \epsilon_k} \right|_{\mathbf{x}_k = \mathbf{x}_k^f, \epsilon_k = 0}, \quad (2.48)$$

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{C}_k^T [\mathbf{D}_k \Gamma_k \mathbf{D}_k^T + \mathbf{C}_k \mathbf{P}_k^f \mathbf{C}_k^T]^{-1}, \quad (2.49)$$

$$\hat{\mathbf{x}}_k^a = \hat{\mathbf{x}}_k^f + \mathbf{K}_k (\mathbf{d}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^f, 0)), \quad (2.50)$$

$$\mathbf{P}_k^a = [\mathbf{I} - \mathbf{K}_k \mathbf{C}_k] \mathbf{P}_k^f. \quad (2.51)$$

Similarly, the model operator \mathbf{g}_k is linearized about $\mathbf{x}_k = \hat{\mathbf{x}}_k^a$ and $\mathbf{q}_k = 0$ to obtain

$$\begin{aligned} \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k) &\approx \mathbf{g}_k(\hat{\mathbf{x}}_k^a, 0) + \left. \frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k^a, \mathbf{q}_k = 0} (\mathbf{x}_k - \hat{\mathbf{x}}_k^a) \\ &\quad + \left. \frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)}{\partial \mathbf{q}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k^a, \mathbf{q}_k = 0} \mathbf{q}_k \end{aligned} \quad (2.52)$$

where $\frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)}{\partial \mathbf{q}_k}$ describe the tangent linear model operators, being the Jacobian matrices of $\mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)$ with respect to \mathbf{x}_k and \mathbf{q}_k respectively.

From Eqs. (2.44)-(2.52), one obtains the posterior mean and error covariance matrix which constitutes the forecast step.

- Forecast step:

$$\mathbf{A}_k = \left. \frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k^a, \mathbf{q}_k = \mathbf{0}}, \quad (2.53)$$

$$\mathbf{B}_k = \left. \frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{q}_k)}{\partial \mathbf{q}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k^a, \mathbf{q}_k = \mathbf{0}}, \quad (2.54)$$

$$\hat{\mathbf{x}}_{k+1}^f = \mathbf{g}_k(\hat{\mathbf{x}}_k^a, \mathbf{0}), \quad (2.55)$$

$$\mathbf{P}_{k+1}^f = \mathbf{A}_k \mathbf{P}_k^a \mathbf{A}_k^T + \mathbf{B}_k \mathbf{Q}_k \mathbf{B}_k^T. \quad (2.56)$$

The accuracy of the extended Kalman filter in tracking the state vector depends on the adequacy of the linear approximation of the model and measurement operators in the vicinity of the current state [105]. The major limitation of EKF is due to the third- and higher-order moments in the error covariance evolution equation being discarded, leading to poor error covariance evolution [10].

In order to alleviate the shortcomings of the EKF in the presence of strong nonlinearities, numerous variations have been proposed. Here are two such variations:

- Second order extended Kalman filter [20, 44, 106]: In this variation, the Kalman gain matrix \mathbf{K} is a function of not only the Jacobian of the model and measurement operators, but also the Hessian of these operators.
- Iterated extended Kalman filter [20, 44, 106]: The first step of the filter is identical to that of a classical extended Kalman filter. The estimated forecast and analysis

estimates of the state are then used as new points to linearize the model and measurement equations to provide another estimate. This process is repeated until convergence.

2.5 Unscented Kalman Filter

In order to address the limited first-order accuracy of means and covariances resulting from a first-order truncated Taylor-series approximation used in EKF, the unscented transform (UT) [23] has been proposed as a tool to propagate the statistical moments of the state vector through the full nonlinear model operator leading to the so-called unscented Kalman filter (UKF) [24–27]. Using a set of carefully chosen deterministic samples (so-called sigma points), UKF estimates the first and second order moments of the state vector correct to the second order Taylor series expansion of the model operator.

2.5.1 Unscented Transform

This method provides an approximate mean and covariance of the transformation of a random vector \mathbf{x} through a nonlinear function f without the need for a Taylor series expansion of f . The method can be summarized by the following steps [23, 107]:

1. Suppose that the random vector \mathbf{x} has dimension n , mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The first step involves the extraction of $(2n + 1)$ characteristic points called

sigma points. These points are determined as follows

$$\mathcal{X}_0 = \boldsymbol{\mu} \quad (2.57)$$

$$\mathcal{X}_i = \boldsymbol{\mu} + \left(\sqrt{(n + \kappa) \boldsymbol{\Sigma}} \right)_i \quad i = 1, \dots, n \quad (2.58)$$

$$\mathcal{X}_i = \boldsymbol{\mu} - \left(\sqrt{(n + \kappa) \boldsymbol{\Sigma}} \right)_{(i-n)} \quad i = n + 1, \dots, 2n \quad (2.59)$$

where κ is chosen such that $n + \kappa = 3$ whenever \mathbf{x} is assumed to be Gaussian, $\left(\sqrt{(n + \kappa) \boldsymbol{\Sigma}} \right)_i$ is the i th row (or column) of the matrix square root of $(n + \kappa) \boldsymbol{\Sigma}$.

2. The second step is the calculation of the image of each of the points onto the function f :

$$\mathcal{M}_i = f(\mathcal{X}_i) \quad i = 0, \dots, 2n \quad (2.60)$$

3. Finally, the mean and the covariance of $f(\mathbf{x})$ are approximated by a weighted average:

$$\hat{\mathbf{E}}[f(\mathbf{x})] = \sum_{i=0}^{2n} W_i \mathcal{M}_i \quad (2.61)$$

$$\hat{\mathbf{E}} \left[(f(\mathbf{x}) - \mathbf{E}[f(\mathbf{x})]) (f(\mathbf{x}) - \mathbf{E}[f(\mathbf{x})])^T \right] = \sum_{i=0}^{2n} W_i \left(\mathcal{M}_i - \hat{\mathbf{E}}[f(\mathbf{x})] \right) \left(\mathcal{M}_i - \hat{\mathbf{E}}[f(\mathbf{x})] \right)^T \quad (2.62)$$

where W_i is the weight given to the i th sigma point:

$$W_0 = \kappa / (n + \kappa) \quad (2.63)$$

$$W_i = 1/2 (n + \kappa) \quad i = 1, \dots, 2n \quad (2.64)$$

As already mentioned, in UKF, the unscented transform is used to propagate the state vector through the full nonlinear model and measurement operators. “These estimates of the mean and covariance are accurate to the second order (third order for true Gaussian priors) of the Taylor series expansion of any nonlinear function” [107].

- *Analysis step:* To perform the analysis step, the nonlinear measurement operator is used to map the sigma points of the state vector, as described by the following steps:

- The unscented transform is applied onto the extended forecast state vector

$\mathbb{X}_k^f = [\mathbf{x}_k^f, \boldsymbol{\epsilon}_k]$ with covariance:

$$\mathbb{P}_k^f = \begin{bmatrix} \mathbf{P}_k^f & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_k \end{bmatrix} \quad (2.65)$$

and the computed sigma points will be of the form $\mathcal{X}_i = [\mathcal{X}_i^{\mathbf{X}}, \mathcal{X}_i^{\boldsymbol{\epsilon}}]$.

- Transform the sigma points through the original nonlinear measurement operator:

$$\mathcal{M}_i = \mathbf{h}_k(\mathcal{X}_i^{\mathbf{X}}, \mathcal{X}_i^{\boldsymbol{\epsilon}}) \quad (2.66)$$

- Obtain the Kalman gain matrix:

$$\hat{\mathbf{d}}_k = \sum_{i=0}^{2n} W_i \mathcal{M}_i \quad (2.67)$$

$$\mathbf{P}_{xd} = \sum_{i=0}^{2n} W_i (\mathcal{X}_i^{\mathbf{X}} - \mathbf{x}_k^f) (\mathcal{M}_i - \hat{\mathbf{d}}_k)^T \quad (2.68)$$

$$\mathbf{P}_{dd} = \sum_{i=0}^{2n} W_i (\mathcal{M}_i - \hat{\mathbf{d}}_k) (\mathcal{M}_i - \hat{\mathbf{d}}_k)^T \quad (2.69)$$

$$\mathbf{K}_k = \mathbf{P}_{xd} \mathbf{P}_{dd}^{-1} \quad (2.70)$$

- Obtain the analysis mean and covariance matrix:

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k \left(\mathbf{d}_k - \hat{\mathbf{d}}_k \right), \quad (2.71)$$

$$\mathbf{P}_k^a = \mathbf{P}_k^f - \mathbf{K}_k \mathbf{P}_{dd}^{-1} \mathbf{K}_k^T. \quad (2.72)$$

- *Forecast step:* To propagate the state estimate through the nonlinear model operator using unscented transform (UT), the forecast step involves obtaining the so-called sigma points and applying the full nonlinear model operator onto each of these points. In order to forecast the state of the system, the following steps are performed:

- The unscented transform is applied onto the extended state vector $\mathbb{X}_k^a = [\mathbf{x}_k^a, \mathbf{q}_k]$ with covariance:

$$\mathbb{P}_k^a = \begin{bmatrix} \mathbf{P}_k^a & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \quad (2.73)$$

and the sigma points obtained will be of the form $\mathcal{X}_i = [\mathcal{X}_i^{\mathbf{x}}, \mathcal{X}_i^{\mathbf{q}}]$.

- Transform the sigma points through the original nonlinear model operator:

$$\mathcal{X}_i^f = \mathbf{g}_k \left(\mathcal{X}_i^{\mathbf{x}}, \mathcal{X}_i^{\mathbf{q}} \right) \quad (2.74)$$

- Obtain the forecast mean and covariance matrix:

$$\mathbf{x}_{k+1}^f = \sum_{i=0}^{2n} W_i \mathcal{X}_i^f \quad (2.75)$$

$$\mathbf{P}_{k+1}^f = \sum_{i=0}^{2n} W_i \left(\mathcal{X}_i^f - \mathbf{x}_{k+1}^f \right) \left(\mathcal{X}_i^f - \mathbf{x}_{k+1}^f \right)^T. \quad (2.76)$$

It is important to note that the UKF does not need the evaluation of the Jacobians and Hessians of the model and measurement operators, contrary to EKF. This is an advantage for UKF in cases when the model and measurement operators are non-differentiable or have complex mathematical forms. Furthermore, the $(2n + 1)$ sigma-points are propagated through the nonlinear operators. Compared with the EKFs first-order accuracy, the estimation accuracy of UKF is perfect for nonlinearities of third-order with Gaussian data and nonlinearities of second-order with non-Gaussian data [24, 27]. UKF has been shown to be more robust and accurate than EKF in numerous applications [24, 27, 107].

2.6 Ensemble Kalman filter

For strong nonlinearity, EKF may provide erroneous conditional mean and covariance estimates due to linearization of the model and measurement operators [9, 10]. UKF often encounters the same problem in practice, although the regularization trick and square-root UKF [108] can alleviate this problem. In EnKF, proposed by Evensen [34], a finite number of Monte Carlo samples of the state vector \mathbf{x}_k are propagated forward in time using the original model operator. One can estimate the pdf of \mathbf{x}_k using ensemble averaging. However, EnKF performs a linear analysis step as KF. The analysis step assumes a Gaussian state and measurement noise which offers computational efficiency but introduces errors in the estimated conditional pdf of \mathbf{x}_k .

Considering again the model and measurement equations as described by Eqs. (2.44)-(2.45), EnKF provides estimates of the conditional mean and covariance of the state vector as follows:

1. Create an initial ensemble $\{\mathbf{x}_{0,i}^f\}$ of size N with $i = 1, \dots, N$, using the prior pdf of \mathbf{x}_0 .

2. Analysis step:

$$\mathbf{d}_{k,i} = \mathbf{h}_k \left(\mathbf{x}_{k,i}^f, \boldsymbol{\epsilon}_{k,i} \right), \quad (2.77)$$

$$\bar{\mathbf{d}}_k = \frac{1}{N} \sum_{i=1}^N \mathbf{d}_{k,i}, \quad (2.78)$$

$$\bar{\mathbf{x}}_k^f = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{k,i}^f, \quad (2.79)$$

$$\mathbf{P}_{xd} = \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{x}_{k,i}^f - \bar{\mathbf{x}}_k^f \right) \left(\mathbf{d}_{k,i} - \bar{\mathbf{d}}_k \right)^T, \quad (2.80)$$

$$\mathbf{P}_{dd} = \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{d}_{k,i} - \bar{\mathbf{d}}_k \right) \left(\mathbf{d}_{k,i} - \bar{\mathbf{d}}_k \right)^T, \quad (2.81)$$

$$\mathbf{K}_k = \mathbf{P}_{xd} \mathbf{P}_{dd}^{-1}, \quad (2.82)$$

$$\mathbf{x}_{k,i}^a = \mathbf{x}_{k,i}^f + \mathbf{K}_k \left(\mathbf{d}_k - \mathbf{h}_k \left(\mathbf{x}_{k,i}^f, \boldsymbol{\epsilon}_{k,i} \right) \right). \quad (2.83)$$

3. Forecast step:

$$\mathbf{x}_{k+1,i}^f = \mathbf{g}_k \left(\mathbf{x}_{k,i}^a, \mathbf{q}_{k,i} \right), \quad (2.84)$$

$$\mathbf{x}_{k+1}^f = \frac{1}{N} \sum_{j=1}^N \mathbf{x}_{k+1,j}^f, \quad (2.85)$$

$$\mathbf{P}_{k+1}^f = \frac{1}{N-1} \sum_{j=1}^N \left(\mathbf{x}_{k+1,j}^f - \mathbf{x}_{k+1}^f \right) \left(\mathbf{x}_{k+1,j}^f - \mathbf{x}_{k+1}^f \right)^T. \quad (2.86)$$

For strongly nonlinear measurement operators, the linear analysis step in EnKF is the major limitation. For the case of *additive* measurement noise $\mathbf{h}_k(\mathbf{x}_k, \boldsymbol{\epsilon}_k) = \mathbf{h}_k(\mathbf{x}_k) + \boldsymbol{\epsilon}_k$, one can however augment the original state vector with $\mathbf{h}_k(\mathbf{x}_k)$. In that case, the measurement becomes a linear function of the new augmented state vector and thereby avoiding the linearization step.

The perturbation of measurements can be avoided using the square-root algorithm.

EnKF avoids the memory-intensive explicit storage of the conditional covariance matrix \mathbf{P}_k as required for EKF. In numerous applications, it has been demonstrated that EnKF resolves the issue of poor error covariance evolution commonly encountered in EKF and UKF [34].

2.7 Particle Filters

EnKF is a particularly appealing ensemble-based data assimilation method because it adopts the linear analysis step of KF. This fact greatly simplifies the analysis step. Although this approach has been widely used for some time, some publications reported rather poor performance by EnKF in certain applications such as those in [32, 33]. In these cases, highly non-Gaussian posterior pdfs lead to difficulties in EnKF. To alleviate such problem, an extension of EnKF is reported that utilizes a weighted sum of Gaussian pdfs (instead of a single Gaussian pdf) to represent the prior distribution [35, 36, 39]. Mandel *et. al.* [48] proposed another extension that combines EnKF with *nonparametric* density estimation techniques based on the distance in Sobolev spaces. A more general approach, namely PF, is devised in the framework of Bayesian inference [9, 40, 44, 53]. In this approach, the ensemble members (so-called *particles*) are properly weighted based on the measurements.

2.7.1 Particle Representation of a Probability Distribution

Introduction

In all cases of classical nonlinear filtering (e.g. EKF), the conditional probability density $p(\mathbf{x}_k | \mathbf{D}_k)$ is represented parametrically, for example by a Gaussian pdf. Each iteration of the filter step involves updating the parameters (e.g. mean and covariance) that fully

characterize the state pdf based on the available observation. In the case of particle filtering, the representation of $p(\mathbf{x}_k|\mathbf{D}_k)$ is different as the filter does not seek an approximate analytical representation of the probability density in parametric forms. It represents the distribution by a set of samples (called particles) distributed according to $p(\mathbf{x}_k|\mathbf{D}_k)$.

Approximating Moments of a pdf

In engineering applications, the conditional mean and variance of the state \mathbf{x}_k are often the two sought-out quantities for state estimation. The question then is how to estimate the first and second moments of $p(\mathbf{x}_k|\mathbf{D}_k)$:

$$E[g(\mathbf{x}_k)|\mathbf{D}_k] = \int g(\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{D}_k) d\mathbf{x}_k \quad (2.87)$$

where

$$\begin{aligned} g(\mathbf{x}_k) &= \mathbf{x}_k && \text{for the first moment} \\ g(\mathbf{x}_k) &= \mathbf{x}_k \mathbf{x}_k^T && \text{for the second moment} \end{aligned}$$

These quantities can be estimated by statistical averages when an ensemble of samples distributed according to $p(\mathbf{x}_k|\mathbf{D}_k)$ is available. Let us denote the ensemble by $\{\mathbf{x}_{k,i}\}$, $i = 1, \dots, N$. If the samples are assumed to be independent of each other, the statistical estimate of the quantity $g(\mathbf{x}_k)$, conditional on observations \mathbf{D}_k , is

$$\hat{E}[g(\mathbf{x}_k)|\mathbf{D}_k] = \frac{1}{N} \sum_{i=1}^N g(\mathbf{x}_{k,i}) . \quad (2.88)$$

The above estimate is a random variable with a mean $E[g(\mathbf{x}_k)|\mathbf{D}_k]$ and variance that tends towards zero as N tends to infinity. More specifically, the estimate of the first moment is

$$\hat{E}[\mathbf{x}_k|\mathbf{D}_k] = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{k,i} \quad (2.89)$$

with variance

$$\text{var} \left[\frac{1}{N} \sum_{i=1}^N \mathbf{x}_{k,i} \right] = \frac{1}{N} \text{var} [\mathbf{x}_k | \mathbf{D}_k] \quad (2.90)$$

which tends to zero as N tends to infinity.

Weighted Sampling of a Probability Distribution

In the previous subsection, it was assumed that it is possible to generate samples distributed according to $p(\mathbf{x}_k | \mathbf{D}_k)$. If this distribution does not have a simple analytical form (i.e. uniform, Gaussian or exponential), it is difficult to simulate its samples digitally using computers. Although there are methods for generating samples from an arbitrary distribution, the computational cost is generally high, especially when the probability density is multidimensional [58–60]. Thus, it is assumed that samples can be easily generate according to another pdf $q(\mathbf{x}_k | \mathbf{D}_k)$ (called the Proposal distribution, or importance function) and that one could evaluate $p(\mathbf{x}_k | \mathbf{D}_k)$ easily. It is then possible to approximate the moments of $p(\mathbf{x}_k | \mathbf{D}_k)$ using samples $\{\mathbf{x}_{k,i}\}$, $i = 1, \dots, N$ distributed according to $q(\mathbf{x}_k | \mathbf{D}_k)$. In that case, Eq. (2.87) can be rewritten as

$$\mathbb{E} [g(\mathbf{x}_k) | \mathbf{D}_k] = \int g(\mathbf{x}_k) p(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k \quad (2.91)$$

$$= \int g(\mathbf{x}_k) \frac{p(\mathbf{x}_k | \mathbf{D}_k)}{q(\mathbf{x}_k | \mathbf{D}_k)} q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k \quad (2.92)$$

$$= \int g(\mathbf{x}_k) \frac{p(\mathbf{D}_k | \mathbf{x}_k) p(\mathbf{x}_k)}{p(\mathbf{D}_k) q(\mathbf{x}_k | \mathbf{D}_k)} q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k \quad (2.93)$$

$$= \frac{1}{p(\mathbf{D}_k)} \int g(\mathbf{x}_k) \frac{p(\mathbf{D}_k | \mathbf{x}_k) p(\mathbf{x}_k)}{q(\mathbf{x}_k | \mathbf{D}_k)} q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k \quad (2.94)$$

$$= \frac{1}{p(\mathbf{D}_k)} \int g(\mathbf{x}_k) \tilde{w}_k(\mathbf{x}_k) q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k \quad (2.95)$$

$$= \frac{\int g(\mathbf{x}_k) \tilde{w}_k(\mathbf{x}_k) q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k}{\int p(\mathbf{D}_k | \mathbf{x}_k) p(\mathbf{x}_k) d\mathbf{x}_k} \quad (2.96)$$

$$= \frac{\int g(\mathbf{x}_k) \tilde{w}_k(\mathbf{x}_k) q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k}{\int \tilde{w}_k(\mathbf{x}_k) q(\mathbf{x}_k | \mathbf{D}_k) d\mathbf{x}_k} \quad (2.97)$$

where

$$\tilde{w}_k(\mathbf{x}_k) = \frac{p(\mathbf{D}_k|\mathbf{x}_k) p(\mathbf{x}_k)}{q(\mathbf{x}_k|\mathbf{D}_k)}. \quad (2.98)$$

Given independent samples $\{\mathbf{x}_{k,i}\}, i = 1, \dots, N$ distributed according to $q(\mathbf{x}_k|\mathbf{D}_k)$, the above expectation can be approximated by

$$\hat{\mathbb{E}}[g(\mathbf{x}_k) | \mathbf{D}_k] = \frac{\frac{1}{N} \sum_{i=1}^N g(\mathbf{x}_{k,i}) \tilde{w}_{k,i}}{\frac{1}{N} \sum_{j=1}^N \tilde{w}_{k,j}} \quad (2.99)$$

$$= \frac{\sum_{i=1}^N g(\mathbf{x}_{k,i}) \tilde{w}_{k,i}}{\sum_{j=1}^N \tilde{w}_{k,j}} \quad (2.100)$$

$$= \sum_{i=1}^N g(\mathbf{x}_{k,i}) \frac{\tilde{w}_{k,i}}{\sum_{j=1}^N \tilde{w}_{k,j}} \quad (2.101)$$

$$= \sum_{i=1}^N g(\mathbf{x}_{k,i}) w_{k,i} \quad (2.102)$$

where the weights are normalized according to:

$$w_{k,i} = \frac{\tilde{w}_{k,i}}{\sum_{i=1}^N \tilde{w}_{k,i}}.$$

Effective Ensemble Size

The effective ensemble size is an indicator used to assess the quality of the approximation, defined by

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_{k,i})^2} \quad (2.103)$$

which is between 0 and N . It is equal to N when all sample weights are equal, i.e. $w_{k,i} = 1/N$. The variance of the approximation in Eq. (2.102) is inversely proportional to the effective ensemble size [109].

2.7.2 Principle of Particle Filters

Introduction

The particle filter propagates a set of weighted samples (called particles) so that at each iteration k , the particles are representative of distribution $p(\mathbf{x}_k|\mathbf{D}_k)$. This is done recursively by changing the state and weight of each particle according to the model and measurement equations. One can make the analogy with the Kalman-based filters, in which the estimated state and covariance are updated recursively. However, two important differences are worth noting. First, the operation of forecasting involves propagating each sample in the ensemble forward by integrating the full nonlinear model operator. Furthermore, there is no unique particle filtering algorithm in the sense that the representation of a probability density by a set of weighted samples is not unique; it depends on the chosen proposal distribution (and ensemble of the particles). Therefore, the performance of the filter depend on this choice. This subsection introduces the principle of particle filter and the conventional choices for the proposal distribution.

Notion of a Particle

It has been assumed that a particle is a sample from a probability distribution, which is associated with a weight. For particle filters consisting of N particles, the ensemble of successive states of particle i from the initial state through the state at instant k will be denoted by $\mathbf{X}_{k,i} = \{\mathbf{x}_{0,i}, \dots, \mathbf{x}_{k,i}\}$. Its weight will be denoted by $w_{k,i}$.

Forecast and Analysis Steps

The proposal distribution used by the particle filter at time step k will be denoted by $q(\mathbf{x}_k|\mathbf{D}_k)$. It is from this distribution that particles are generated. This distribution, in

general, depends on the set of observations \mathbf{D}_k . Once the trajectories of particles are drawn according to the proposal distribution, the weights are evaluated so that the set of weighted particles approximate the probability density $p(\mathbf{x}_k|\mathbf{D}_k)$. The weight of each particle is therefore:

$$w_{k,i} = \frac{p(\mathbf{X}_{k,i}|\mathbf{D}_k)}{q(\mathbf{X}_{k,i}|\mathbf{D}_k)}. \quad (2.104)$$

In practice one does not know the analytical expression of $p(\mathbf{X}_k|\mathbf{D}_k)$ as it is precisely the density that one seeks to approximate. However, the calculation of the weights admits a recursive expression. Indeed, the solution of the Bayesian estimation problem has the following recursive form as in Eq. (2.18):

$$p(\mathbf{X}_k|\mathbf{D}_k) = p(\mathbf{X}_{k-1}|\mathbf{D}_{k-1}) \frac{p(\mathbf{d}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{d}_k|\mathbf{D}_{k-1})} \quad (2.105)$$

which in turn provides a recursive formula for the weights:

$$\begin{aligned} w_{k,i} &= \frac{p(\mathbf{X}_{k,i}|\mathbf{D}_k)}{q(\mathbf{X}_{k,i}|\mathbf{D}_k)} \\ &= p(\mathbf{X}_{k-1,i}|\mathbf{D}_{k-1}) \frac{p(\mathbf{d}_k|\mathbf{x}_{k,i}) p(\mathbf{x}_{k,i}|\mathbf{x}_{k-1,i})}{p(\mathbf{d}_k|\mathbf{D}_{k-1})} \frac{1}{q(\mathbf{X}_{k,i}|\mathbf{D}_{k-1})} \\ &= \frac{p(\mathbf{X}_{k-1,i}|\mathbf{D}_{k-1}) p(\mathbf{d}_k|\mathbf{x}_{k,i}) p(\mathbf{x}_{k,i}|\mathbf{x}_{k-1,i}) q(\mathbf{X}_{k-1,i}|\mathbf{D}_{k-1})}{q(\mathbf{X}_{k-1,i}|\mathbf{D}_{k-1}) p(\mathbf{d}_k|\mathbf{D}_{k-1}) q(\mathbf{X}_{k,i}|\mathbf{D}_{k-1})} \\ &= w_{k-1,i} \frac{p(\mathbf{d}_k|\mathbf{x}_{k,i}) p(\mathbf{x}_{k,i}|\mathbf{x}_{k-1,i}) q(\mathbf{X}_{k-1,i}|\mathbf{D}_{k-1})}{p(\mathbf{d}_k|\mathbf{D}_{k-1}) q(\mathbf{X}_{k,i}|\mathbf{D}_{k-1})} \\ &\propto w_{k-1,i} p(\mathbf{d}_k|\mathbf{x}_{k,i}) p(\mathbf{x}_{k,i}|\mathbf{x}_{k-1,i}) \frac{q(\mathbf{X}_{k-1,i}|\mathbf{D}_{k-1})}{q(\mathbf{X}_{k,i}|\mathbf{D}_{k-1})}. \end{aligned} \quad (2.106)$$

The densities $p(\mathbf{d}_k|\mathbf{x}_{k,i})$ and $p(\mathbf{x}_{k,i}|\mathbf{x}_{k-1,i})$ are obtained from the model and measurement equations. This is the principle of *sequential importance sampling* (SIS) filter. Eq. (2.106) can be simplified further if the proposal distribution permits the following

factorization:

$$q(\mathbf{X}_k | \mathbf{D}_k) = q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{D}_k) q(\mathbf{X}_{k-1} | \mathbf{D}_{k-1}) . \quad (2.107)$$

$q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{D}_k)$ is the proposal density used to sample the k th point of the trajectory, conditional on the previous states and measurements. In this case, the recursive expression of the weights becomes

$$w_{k,i} \propto w_{k-1,i} \frac{p(\mathbf{d}_k | \mathbf{x}_{k,i}) p(\mathbf{x}_{k,i} | \mathbf{x}_{k-1,i})}{q(\mathbf{x}_{k,i} | \mathbf{X}_{k-1,i}, \mathbf{D}_k)} . \quad (2.108)$$

The particle filtering algorithm consists of two steps at each iteration:

1. Forecast step: Obtain samples from a proposal distribution

$$\mathbf{x}_{k,i} \sim q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{D}_k) \quad i = 1, \dots, N \quad (2.109)$$

2. Analysis step: Update the weights of all particles based on the observation \mathbf{d}_k according to

$$\tilde{w}_{k,i} \propto w_{k-1,i} \frac{p(\mathbf{d}_k | \mathbf{x}_{k,i}) p(\mathbf{x}_{k,i} | \mathbf{x}_{k-1,i})}{q(\mathbf{x}_{k,i} | \mathbf{X}_{k-1,i}, \mathbf{D}_k)} \quad (2.110)$$

$$w_{k,i} = \frac{\tilde{w}_{k,i}}{\sum_{i=1}^N \tilde{w}_{k,i}} . \quad (2.111)$$

Necessity of a Resampling step

In the classical problem of recursive estimation, uncertainty of the state is initially high which gradually decreases with the sequential assimilation of observations. In most cases, the support of $p(\mathbf{x}_k | \mathbf{D}_k)$ is initially large and decreases over time. The particles are scattered across the state space according to some proposal distribution. Depending

on the choice of proposal distribution, in some cases one ends up with the majority of particles being dispersed in areas where $p(\mathbf{x}_k|\mathbf{D}_k)$ is very small in value. Therefore, these particles have a very low weight. The severity of this so-called degeneracy phenomenon depends on the problem and the choice of the proposal distribution.

To alleviate the degeneracy of the particles, a step will be undertaken to remove particles of negligible weight and duplicate the particles of high weight. There are different methods for resampling including multinomial sampling [45, 110–112]), residual resampling [52, 113] and stratified sampling [113]. Resampling can be performed at every step of the filter, or only performed if degeneracy is detected. The resulting filter is commonly known as a *Sampling Importance Resampling* (SIR) filter. The multinomial resampling is summarized as follows:

1. Draw N independent and identically-distributed (IID) samples from a uniform distribution on $[0, 1]$

$$u_i \sim \mathcal{U}(0, 1) \quad i = 1, \dots, N \quad (2.112)$$

2. Obtain the cumulative density function v of the weights:

$$v_j = \sum_{l=1}^j w_{k,l} \quad (2.113)$$

3. For $m = 1, \dots, N$, determine n such that $v_n \leq u_m < v_{n+1}$ and copy the n th particle $\mathbf{x}_{k,n}$ into the new ensemble
4. Reset the weights as $w_{k,i} = 1/N$, for $i = 1, \dots, N$.

As already mentioned, the resampling step can be performed at each iteration of the filter, or only when the degeneracy of the particles is detected. An indicator of the

presence of degeneracy the effective ensemble size defined in Subsection 2.7.1. N_{eff} is equal to the ensemble size N if all weights are equal and decreases as the extent of degeneracy increases. Thus, resampling can be performed when the effective ensemble size falls below a certain threshold value, denoted by N_{thr} [51–54]. A common threshold for resampling is $N_{eff} = N/3$ [53].

It is important to note that although the resampling step is essential to the effective functioning of the particle filter, it may have adverse effects on the operation of the filter and the quality of the particle representation of the posterior pdf $p(\mathbf{x}_k|\mathbf{D}_k)$. Indeed, particles with relatively high weight are duplicated numerous times leading to loss of diversity among the ensemble. If the model noise term is relatively weak, the ensemble members will be close to each other during the subsequent iterations and the space state is insufficiently explored. This is the motivation behind performing the resampling step when degeneracy is detected.

Choice of Proposal Distribution

The resampling step aims at reducing the effect of the degeneracy phenomenon whereby the majority of particles are clustered in areas where $p(\mathbf{x}_k|\mathbf{D}_k)$ is very low, due to a poor choice of proposal distribution. Thus the judicious selection of proposal distribution minimizes the occurrence rate of degeneracy and reduce the overall number of resampling steps. Next, some proposal distributions that are used in practice will be briefly reviewed.

Prior Proposal Distribution

The use of the prior pdf as the proposal distribution, i.e.

$$q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{D}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad (2.114)$$

leads to the simplest method, often referred to as the *bootstrap filter* [45]. It does not take into account the most recent observation \mathbf{d}_k . The particle trajectories are obtained by simply applying the stochastic model equation to update the state vector of each particle. The most recent observation is used in the weight-update equation which simplifies to

$$\tilde{w}_{k,i} \propto w_{k-1,i} p(\mathbf{d}_k | \mathbf{x}_{k,i}) . \quad (2.115)$$

This choice of proposal distribution ignores the most recent observation and thus is not optimal (it does not significantly minimize the occurrence rate of degeneracy).

Optimal Proposal Distribution

The optimal proposal distribution (or importance function) is the distribution that minimizes the variance of the particle weights (and thus also minimizes the occurrence of the degeneracy phenomenon) [53]. The optimal importance function is given by

$$q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{D}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1,i}, \mathbf{d}_k) \quad (2.116)$$

which would in turn give the following weight update equation [53]

$$\tilde{w}_{k,i} \propto w_{k-1,i} p(\mathbf{d}_k | \mathbf{x}_{k-1,i}) . \quad (2.117)$$

The optimal proposal suffers from two major drawbacks. It requires the ability to sample from $p(\mathbf{x}_k | \mathbf{x}_{k-1,i}, \mathbf{d}_k)$ and to evaluate $p(\mathbf{d}_k | \mathbf{x}_{k-1,i})$, which can not be achieved in general cases.

Proposal Distribution as Forecast pdf of EKF or UKF

The Gaussian proposal distribution is of the form

$$q(\mathbf{x}_k | \mathbf{X}_{k-1}, \mathbf{D}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) . \quad (2.118)$$

The sampling from Gaussian distribution can be performed conveniently. Furthermore, $p(\mathbf{d}_k | \mathbf{x}_{k-1,i})$ would be easy to evaluate. This proposal distribution is the posterior or forecast pdf obtained by EKF or UKF.

For instance, Van der Merwe and Wan [108] developed a particle filter with importance function given by a bank of unscented Kalman filters. A separate UKF is associated with each particle providing a unique proposal for each of the particles:

$$q(\mathbf{x}_{k,i} | \mathbf{X}_{k-1,i}, \mathbf{D}_k) = \mathcal{N}(\mathbf{x}_{k,i}; \hat{\mathbf{x}}_{k,i}, \mathbf{P}_{k,i}) . \quad (2.119)$$

Since the proposal distribution is not unique for all particles, this proposal results in a sub-optimal filter [104].

Proposal Distribution as Forecast of EnKF

For strongly nonlinear systems, the proposal based on EnKF is perhaps most appropriate as it maintains some non-Gaussian characteristics in the proposal. Unlike EKF and UKF based proposal, EnKF proposal does not differ from particle to particle. In other words, a single proposal is used to update the particle weights as explained next. The (non-Gaussian) EnKF based proposal is obtained by kernel density estimation (KDE) [114]. The forecast pdf of the particle filter is also estimated by kernel density estimation (KDE). This proposal is first proposed by Mandel and Beezley and termed as the predictor-corrector filter [48]. The algorithm of the predictor-corrector is summarized next:

1. Using the prior pdf of \mathbf{x}_0 , create an ensemble $\{\mathbf{x}_{0,i}^f\}$ of size N with $i = 1, \dots, N$.

2. Analysis step:

$$\mathbf{d}_{k,i} = \mathbf{h}_k \left(\mathbf{x}_{k,i}^f, \boldsymbol{\epsilon}_{k,i} \right), \quad (2.120)$$

$$\bar{\mathbf{d}}_k = \sum_{i=1}^N w_{k,i}^f \mathbf{d}_{k,i}, \quad (2.121)$$

$$\bar{\mathbf{x}}_k^f = \sum_{i=1}^N w_{k,i}^f \mathbf{x}_{k,i}^f, \quad (2.122)$$

$$\mathbf{P}_{xd} = \frac{1}{1 - \sum_{i=1}^N w_{k,i}^f} \frac{1}{2} \sum_{i=1}^N w_{k,i}^f \left(\mathbf{x}_{k,i}^f - \bar{\mathbf{x}}_k^f \right) \left(\mathbf{d}_{k,i} - \bar{\mathbf{d}}_k \right)^T, \quad (2.123)$$

$$\mathbf{P}_{dd} = \frac{1}{1 - \sum_{i=1}^N w_{k,i}^f} \frac{1}{2} \sum_{i=1}^N w_{k,i}^f \left(\mathbf{d}_{k,i} - \bar{\mathbf{d}}_k \right) \left(\mathbf{d}_{k,i} - \bar{\mathbf{d}}_k \right)^T, \quad (2.124)$$

$$\mathbf{K}_k = \mathbf{P}_{xd} \mathbf{P}_{dd}^{-1}, \quad (2.125)$$

$$\mathbf{x}_{k,i}^a = \mathbf{x}_{k,i}^f + \mathbf{K}_k \left(\mathbf{d}_k - \mathbf{h}_k \left(\mathbf{x}_{k,i}^f, \boldsymbol{\epsilon}_{j,i} \right) \right) \quad (2.126)$$

$$w_{k,i}^a = w_{k,i}^f \mathbb{P} \left(\mathbf{d}_k | \mathbf{x}_{k,i}^a \right) \frac{p_f \left(\mathbf{x}_{k,i}^a \right)}{p_\pi \left(\mathbf{x}_{k,i}^a \right)} \quad (2.127)$$

$$\approx w_{k,i}^f \mathbb{P} \left(\mathbf{d}_k | \mathbf{x}_{k,i}^a \right) \frac{\sum_{j=1}^N w_{k,j}^f \varphi \left(\frac{\|\mathbf{x}_{k,j}^f - \mathbf{x}_{k,i}^a\|_U}{h_j} \right)}{\sum_{l=1}^N \frac{1}{N} \varphi \left(\frac{\|\mathbf{x}_{k,l}^a - \mathbf{x}_{k,i}^a\|_U}{h_l} \right)} \quad (2.128)$$

$$\bar{\mathbf{x}}_k^a = \sum_{j=1}^N w_{k,j}^a \mathbf{x}_{k,j}^a, \quad (2.129)$$

$$\mathbf{P}_k^a = \frac{1}{1 - \sum_{i=1}^N w_{k,i}^a} \frac{1}{2} \sum_{j=1}^N w_{k,j}^a \left(\mathbf{x}_{k,j}^a - \bar{\mathbf{x}}_k^a \right) \left(\mathbf{x}_{k,j}^a - \bar{\mathbf{x}}_k^a \right)^T. \quad (2.130)$$

3. Forecast step:

$$w_{k+1,i}^f = w_{k,i}^a \quad (2.131)$$

$$\mathbf{x}_{k+1,i}^f = \mathbf{g}_k(\mathbf{x}_{k,i}^a, \mathbf{f}_k, \mathbf{q}_{k,i}), \quad (2.132)$$

$$\bar{\mathbf{x}}_{k+1}^f = \sum_{j=1}^N w_{k,j}^f \mathbf{x}_{k+1,j}^f, \quad (2.133)$$

$$\mathbf{P}_{k+1}^f = \frac{1}{1 - \sum_{i=1}^N w_{k,i}^f} \sum_{j=1}^N w_{k,j}^f \left(\mathbf{x}_{k,j}^f - \bar{\mathbf{x}}_k^f \right) \left(\mathbf{x}_{k,j}^f - \bar{\mathbf{x}}_k^f \right)^T. \quad (2.134)$$

φ is a chosen kernel function required for kernel density estimation, h is the bandwidth, and $\|\cdot\|_U$ is a norm on the state space. The Euclidean-norm is used herein.

2.8 Application to Nonlinear Spring-Damper Oscillator

A nonlinear spring-damper oscillator forced by random input is described by the following nonlinear differential equation in the state-space form (e.g. [18, 48, 115, 116])

$$\dot{x}(t) = 4x(t) - 4x^3(t) + \sigma\xi(t) \quad (2.135)$$

with $x(t)$ being the displacement, $\xi(t)$ is a Gaussian white noise describing the random input (additive modeling error) and σ denotes its strength. The oscillator contains two stable fixed points at $x = \pm 1$ and an unstable fixed point at $x = 0$. Its motion is analogous to the motion of a ball rolling between two wells (see Fig. 2.1). This system is called the double-well system [33] or the Bernoulli equation [117].

The above equation can be cast in Ito stochastic differential equation (SDE) [118] as

$$dx = [4x - 4x^3] dt + \sigma\xi(t) dt. \quad (2.136)$$

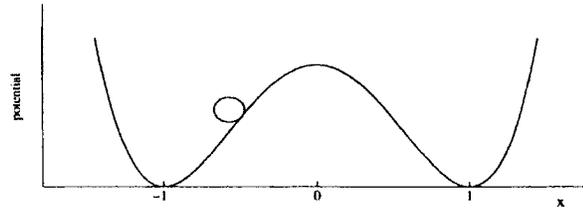


Figure 2.1: Double-well potential.

where $\xi(t) dt = dW = W(t_{k+1}) - W(t_k)$ is a Brownian increment.

The above Ito stochastic differential equation (SDE) is discretized using the Euler-Maruyama stochastic integration scheme [118–120] with time step Δt as

$$x_{k+1} = x_k + \Delta t [4x_k - 4x_k^3] + \sigma \sqrt{\Delta t} \varepsilon_k \quad (2.137)$$

where ε_k is a standard Gaussian random variables.

The evolution of the pdf for x is given by the following Fokker-Planck-Kolmogorov equation [33]:

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x} \{4x(x^2 - 1)p\} + \frac{1}{2} \sigma^2 \frac{\partial^2 p}{\partial x^2}. \quad (2.138)$$

2.8.1 State Estimation

Numerical experiments related to the state tracking of the nonlinear oscillator using the various nonlinear filters is conducted next. In Fig. 2.2, the true displacement of the oscillator and its noisy measurements d_k polluted by Gaussian noise $\varepsilon_k \sim \mathcal{N}(0, 0.08)$ are plotted. The modeling error has an amplitude of $\sigma = 0.5$. The observational data is acquired and analyzed at intervals of 1 time unit. For the ensemble Kalman filter, particle filter and the predictor-corrector filter, an ensemble size of $N = 150$ is used. To minimize sampling errors, latin hypercube sampling (LHS) [121, 122] is employed for the ensemble Kalman filter, the particle filter and the predictor-corrector filter.

The state estimation results using the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the ensemble Kalman filter (EnKF), the bootstrap filter and the predictor-corrector filter are shown in Fig. 2.2. For all filters, the MMSE estimates are used. The initial conditions for x is a Gaussian pdf with unity mean and variance of 0.01. The threshold effective ensemble size for the particle filter and the predictor-corrector filter for resampling purposes is taken to be 33% of the ensemble size (i.e. $N_{thr} = N/3$). The predictor-corrector filter tracks the jump of the state of the system from one potential well to another the fastest among all filters. The jump is also tracked by the particle filter and ensemble Kalman filter, albeit with longer delay. The extended and unscented Kalman filters do not track the jump during the time interval examined due to the non-Gaussian nature of the state vector.

Fig. 2.3 also presents the MMSE state estimate obtained by directly solving the Fokker-Planck-Kolmogorov equation using the finite difference method conditioned by the same data, denoted by the *optimal* solution. The non-Gaussian characteristics in the conditioned transition pdf are quite obvious as the oscillator jumps from one potential well to another, explaining the lack of tracking ability of the classical nonlinear filters (i.e. EKF and UKF). This transition (jump) takes place between 1 seconds and 5 seconds of the simulation. The non-Gaussian nature of the pdf manifests itself through bi-modality of the pdfs. The modes (peaks) of the pdfs are at the stable fixed points $x = \pm 1$.

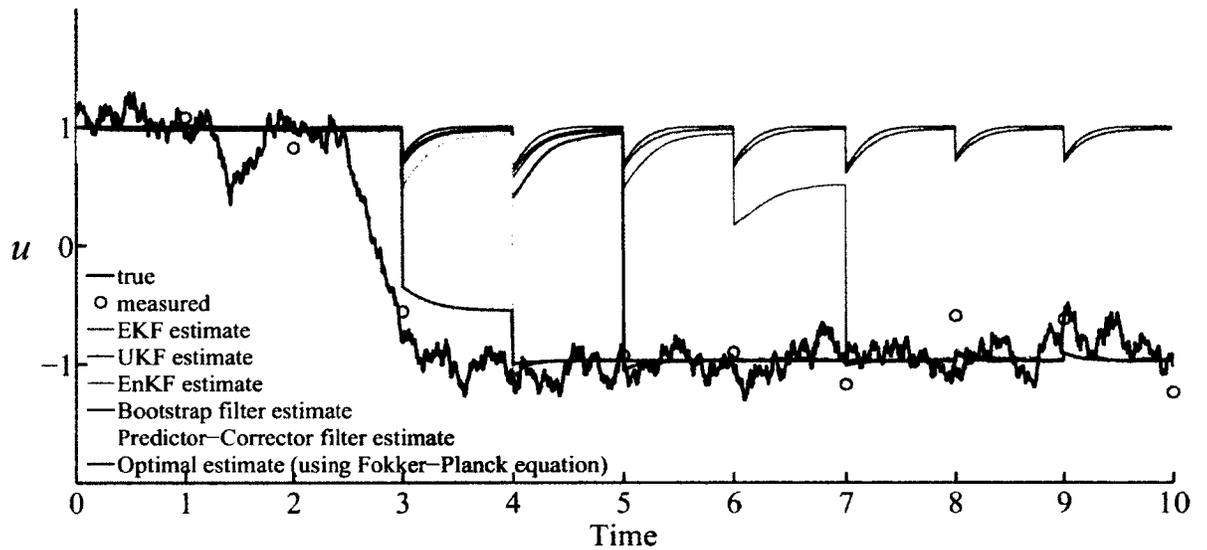


Figure 2.2: True, measured and estimated displacement of the double-well system. The true displacement is shown in black and the measurements as red circles. The MMSE displacement estimates are shown for EKF (red), UKF (brown), bootstrap filter (blue), EnKF (purple), predictor-corrector filter (orange) and the optimal estimate (green) obtained by solving the Fokker-Planck-Kolmogorov equation and conditioning using Bayes' theorem. The shaded region is the MMSE optimal estimate plus or minus 3 standard deviations representing the uncertainty in the estimate.

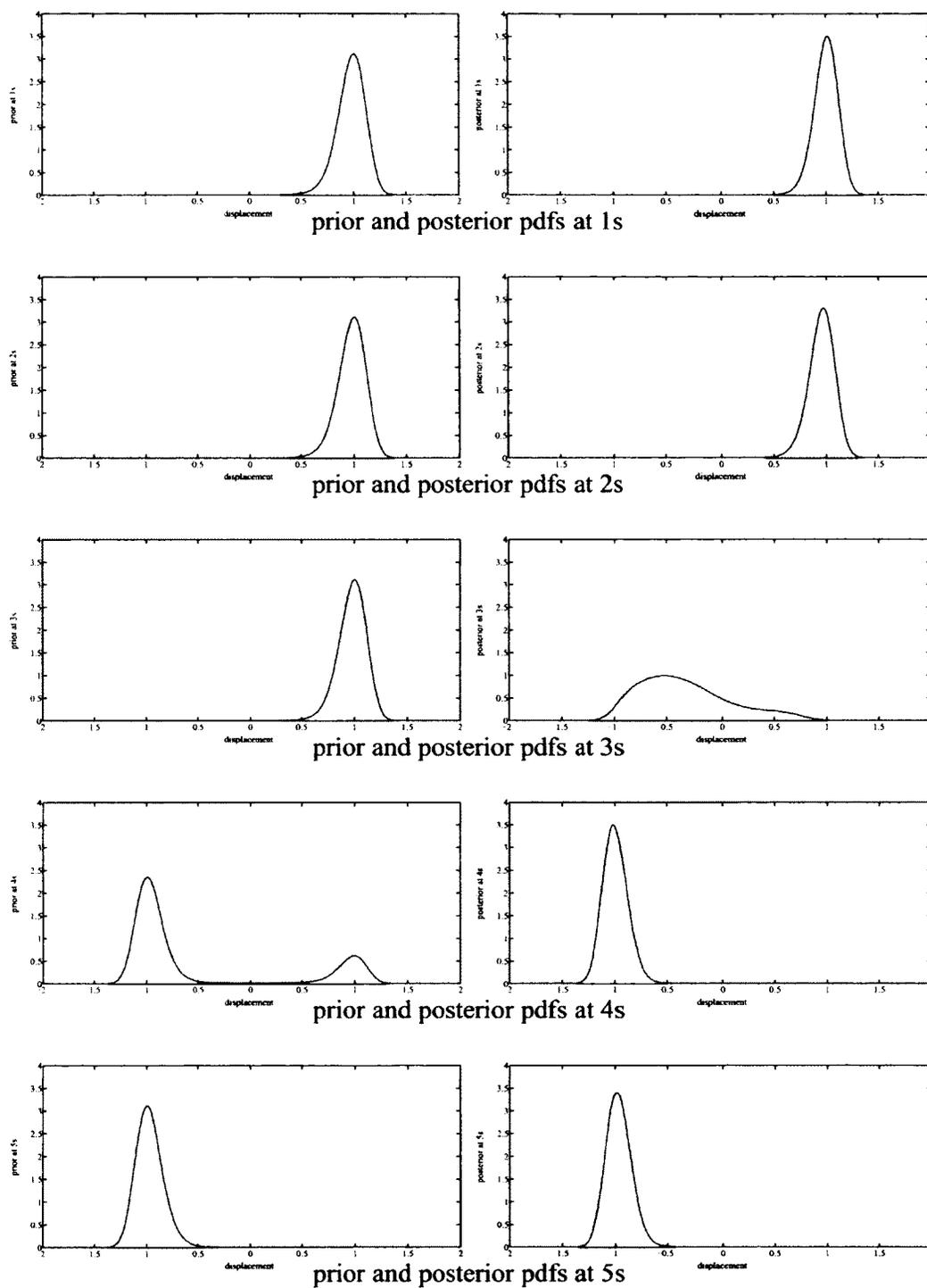


Figure 2.3: Prior and Posterior displacement pdfs of the double-well system obtained by solving the Fokker-Planck-Kolmogorov equation and conditioning the pdfs using Bayes' theorem.

Chapter 3

Markov Chain Monte Carlo

The application of Bayesian inference for tackling engineering problems has been limited due to the difficulty in integrating high-dimensional functions required to obtain the posterior distribution and statistical estimates of the unknown quantities of interest. Several methods that approximate these integrals have been proposed (e.g. [123, 124]). Markov Chain Monte Carlo (MCMC) methods provide samples from complex distributions while circumventing the need for high-dimensional integration. MCMC approaches provide new samples that are generated using the previous sample(s) by constructing a Markov chain. MCMC methods are rooted in the Metropolis algorithm [61] which was initially employed by physicists to compute complex integrals by rewriting them as expectations of some distributions, whereby MCMC provides samples from these distributions. This chapter provides an elementary exposition of the Monte Carlo and Markov chain Monte Carlo (MCMC) sampling techniques based on the textbooks [58–60]. First, the concept of Monte Carlo integration is provided. Next, the essential theory of Markov chains needed to develop MCMC is presented. The Metropolis-Hastings algorithm for MCMC is subsequently derived. The chapter ends with two engineering applications of MCMC.

3.1 Monte Carlo Integration

As explained in the previous chapter, Bayesian inference of a set of parameters involves combining the likelihood function with the prior density function of these parameters according to Bayes' theorem (Eq. (2.10)). The product of the likelihood and prior is proportional to the posterior probability. In order to use the the posterior distribution to make an inference regarding the parameters, one requires the estimation of the normalization factor (being the denominator in Eq. (2.10)), which in practice is analytically intractable in most cases. For high-dimensional problems arising in practice, the numerical approximation of the normalization factor is difficult to obtain. To alleviate this difficulty, one may resort to using Monte Carlo sampling techniques [55, 58–60, 125]. Monte Carlo methods provides samples of the parameter(s) of interest according to the posterior probability density. One can then statistically estimate the quantities of interest from these samples.

The inception of Monte Carlo methods is mainly credited to John von Neumann, Stanislaw M. Ulam, Enrico Fermi and Nicholas Metropolis while working on the so-called “Manhattan Project“ to design the atomic bomb [126]. While studying fissile material, they were faced with the difficulty of describing mathematically the process of chain reaction. They proposed to carry out a simulation using dice for tracking the probable paths of neutrons in a chain reaction [127]. At the suggestion of N. Metropolis and in honor of a gambling uncle of Ulam, the method is called Monte Carlo sampling in reference to the famous casinos of Monaco [125].

The principle of Monte Carlo sampling can be formally described as follows [55, 58–60, 125]. Let f be a function of a random variable (or vector) \mathbf{x} and π denotes the

(posterior) pdf of this variable. The expectation of $f(\mathbf{x})$ is defined as:

$$E[f(\mathbf{x})] = \int f(\mathbf{x}) \pi(\mathbf{x}) d\mathbf{x} . \quad (3.1)$$

In many cases, this integral can not be calculated analytically. The Monte Carlo approximation consists of producing an ensemble of random samples of the variable \mathbf{x} distributed according to the density π :

$$\{x_1, x_2, \dots, x_N\} \sim \pi . \quad (3.2)$$

This may be performed by one of many available techniques [55, 58–60, 125], including MCMC, and generally requires a large number of random draws. An estimator of the expected value of $f(\mathbf{x})$, denoted by $\tilde{f}_N(\mathbf{x})$, is based on the empirical mean of the function f applied to the samples x_i :

$$\tilde{f}_N(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N f(x_i) . \quad (3.3)$$

This estimator is a Monte Carlo estimator [55, 58–60, 125] that approximates the expected value of $f(\mathbf{x})$:

$$\tilde{f}_N(\mathbf{x}) \approx E[f(\mathbf{x})] . \quad (3.4)$$

This estimator is unbiased, i.e.

$$E[\tilde{f}_N(\mathbf{x})] = E[f(\mathbf{x})] . \quad (3.5)$$

When the sample size N is large enough, the law of large numbers guarantees the

convergence of the estimator to the expected value:

$$\lim_{N \rightarrow \infty} \tilde{f}_N(\mathbf{x}) = E[f(\mathbf{x})] . \quad (3.6)$$

Simple Monte Carlo sampling technique can generate *independent* random samples from the target distribution π from which one can extract relevant estimates (e.g. mean and mode). For a general non-Gaussian pdfs, independent samples are difficult to simulate in practice. One can however sample from a Markov chain whose *equilibrium* or *stationary* pdf matches π as described next. Although the samples are not independent, the statistical features of the pdf can be estimated with reasonable accuracy using a sufficiently large enough sample size. A brief description of Markov chains and the general algorithm of MCMC adopted in this thesis, namely the Metropolis algorithm is presented next. More details can be obtained in the books by Gilks *et al.* [58], Gamerman [59] and Liu [60].

3.2 Theory of Markov Chains

Before discussing the MCMC method, some fundamentals of Markov chains will be presented. For the sake of brevity, a theoretical overview of the properties needed to develop the MCMC technique will be given. A sequence of random vectors $\{\mathbf{x}_0, \mathbf{x}_1, \dots\}$ is called a first-order Markov chain if it satisfies the property [59, 60]

$$p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y}, \dots, \mathbf{x}_0 = \mathbf{a}) = p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y}) . \quad (3.7)$$

It states that \mathbf{x}_{i+1} statistically depends only on the previous random vector \mathbf{x}_i . When the transition pdf (kernel) $p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y})$ is time-invariant (i.e. independent of i), it

can be denoted by $A(\mathbf{y}, \mathbf{z})$. The Markov chain in this case is termed homogeneous. The transition kernel must satisfy, for all \mathbf{y} ,

$$A(\mathbf{y}, \mathbf{z}) \geq 0 \quad \forall \mathbf{y}, \mathbf{z} \in S \quad \text{and} \quad \int A(\mathbf{y}, \mathbf{z}) \, d\mathbf{z} = 1. \quad (3.8)$$

Some properties of Markov chains ensure the effectiveness of the method developed in the following section. This is particularly true for the property of irreducibility.

Definition 3.1 (Irreducibility).

Consider a sequence of states of a Markov chain

$$\mathbf{y} \rightarrow \mathbf{k}_1 \rightarrow \mathbf{k}_2 \rightarrow \cdots \rightarrow \mathbf{k}_n \rightarrow \mathbf{z} \quad (3.9)$$

such that the transition probabilities $A(\mathbf{y}, \mathbf{k}_1) \neq 0$, $A(\mathbf{k}_m, \mathbf{k}_{m+1}) \neq 0$, $A(\mathbf{k}_n, \mathbf{z}) \neq 0$, then there is a sequence of states from \mathbf{y} to \mathbf{z} with a non-zero probability of occurring. In that case it is said that \mathbf{y} and \mathbf{z} communicate. If \mathbf{z} and \mathbf{y} also communicate, one says that \mathbf{y} and \mathbf{z} intercommunicate. If all states in S intercommunicate, the Markov chain is irreducible under A . Otherwise, the chain is reducible under A . The concept of irreducibility can be stated as follows. Starting with any possible state in S , the chain can reach any other state in some number of iterations.

The next property sought in a Markov chain is the aperiodicity.

Definition 3.2 (Aperiodicity).

Let M be an irreducible Markov chain with transition kernel A and let \mathbf{y} be a fixed state. Define the set

$$T = \{k : p(\mathbf{x}_{i+k} = \mathbf{y} | \mathbf{x}_i = \mathbf{y}) > 0, k > 0\}. \quad (3.10)$$

These are the steps on which it is possible for a chain which starts in state \mathbf{y} to revisit \mathbf{y} . The greatest common divisor of the integers in T is called the period of the state \mathbf{y} [59, 60]. The chain is said to be periodic if the period of any of its states is greater than one. Otherwise, M is aperiodic. Stated differently, the chain is aperiodic if it does not periodically oscillate among different sets of states.

The last property of interest is reversibility.

Definition 3.3 (Reversibility).

A homogeneous Markov chain M is reversible if the transition kernel for the reversed chain M' coincides with that of M [59, 60], so that

$$p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y}) = p(\mathbf{x}_i = \mathbf{z} | \mathbf{x}_{i+1} = \mathbf{y}). \quad (3.11)$$

Theorem 3.1 (Detailed Balance Condition).

Let $A(\mathbf{y}, \mathbf{z}) = p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y})$ be the transition distribution or transition kernel for a given Markov chain. If, for some distribution π , $A(\mathbf{y}, \mathbf{z})$ satisfies detailed balance given by

$$A(\mathbf{y}, \mathbf{z}) \pi(\mathbf{y}) = A(\mathbf{z}, \mathbf{y}) \pi(\mathbf{z}), \quad (3.12)$$

the chain has a stationary distribution π and is said to be reversible with respect to π [128].

Proof. Let $B(\mathbf{y}, \mathbf{z})$ be the transition kernel of the reversed chain $M' = \{\mathbf{x}_n\}_{n=-\infty}^{\infty}$ where $\mathbf{x}_n = \mathbf{x}_{-n}$, i.e.,

$$B(\mathbf{y}, \mathbf{z}) = p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y}). \quad (3.13)$$

It must be shown that $B(\mathbf{y}, \mathbf{z}) = A(\mathbf{y}, \mathbf{z})$ if and only if detailed balance holds. By

definition of the process \mathbf{x} and Bayes' theorem,

$$\begin{aligned}
 B(\mathbf{y}, \mathbf{z}) &= p(\mathbf{x}_{i+1} = \mathbf{z} | \mathbf{x}_i = \mathbf{y}) \\
 &= p(\mathbf{x}_{-i-1} = \mathbf{z} | \mathbf{x}_{-i} = \mathbf{y}) \\
 &= \frac{p(\mathbf{x}_{-i} = \mathbf{y} | \mathbf{x}_{-i-1} = \mathbf{z}) p(\mathbf{x}_{-i-1} = \mathbf{z})}{p(\mathbf{x}_{-i} = \mathbf{y})}.
 \end{aligned} \tag{3.14}$$

Since each \mathbf{x}_n is distributed as π for all n , then

$$\begin{aligned}
 B(\mathbf{y}, \mathbf{z}) &= \frac{p(\mathbf{x}_{-i} = \mathbf{y} | \mathbf{x}_{-i-1} = \mathbf{z}) \pi(\mathbf{z})}{\pi(\mathbf{y})} \\
 &= \frac{A(\mathbf{z}, \mathbf{y}) \pi(\mathbf{z})}{\pi(\mathbf{y})}.
 \end{aligned} \tag{3.15}$$

Then $B(\mathbf{y}, \mathbf{z}) = A(\mathbf{y}, \mathbf{z})$ and the chain is reversible if and only if detailed balance holds. □

For an aperiodic, irreducible and reversible Markov chain (i.e. satisfying the above three conditions), the limiting distribution of successive iterates will reach the stationary (target) pdf, regardless of the starting value of the chain [58–60]. The stationary pdf π and the transition kernel $A(\mathbf{x}, \mathbf{y})$ satisfy the following eigenvalue problem [58–60]:

$$\int \pi(\mathbf{x}) A(\mathbf{x}, \mathbf{y}) d\mathbf{x} = \pi(\mathbf{y}). \tag{3.16}$$

The number of samples required to reach the stationary distribution from a starting sample \mathbf{x}_0 is called the *burn-in period* of the chain. After the burn-in period, the samples of the Markov chain approximately follow the target (stationary) pdf π [58, 60].

3.3 The Metropolis-Hastings Algorithm

Given a transition kernel $A(\mathbf{y}, \mathbf{z})$ for a 1st-order aperiodic, irreducible and positive-recurrent Markov chain, its stationary distribution π satisfies Eq. (3.16). In Bayesian inference problems, the target (posterior) distribution is available, but the corresponding transition kernel is not known *a priori*. The method first proposed by Metropolis *et al.* [61] and generalized by Hastings[129] is adopted in this thesis.

Starting from an arbitrary sample, a candidate point \mathbf{z} is generated from an arbitrarily chosen proposal pdf $q(\mathbf{y}, \mathbf{z})$ in Metropolis-Hastings algorithm. Then the candidate point \mathbf{z} is accepted with probability $\alpha(\mathbf{y}, \mathbf{z})$ where

$$\alpha(\mathbf{y}, \mathbf{z}) = \min \left(1, \frac{\pi(\mathbf{z})}{\pi(\mathbf{y})} \right). \quad (3.17)$$

For the Metropolis algorithm[61], the proposal pdf $q(\mathbf{y}, \mathbf{z})$ is chosen to be symmetric, i.e. $q(\mathbf{y}, \mathbf{z}) = q(\mathbf{z}, \mathbf{y})$. This constraint is however relaxed in the Metropolis-Hastings algorithm[129]. It can be shown how the above choice of acceptance probability $\alpha(\mathbf{y}, \mathbf{z})$ leads to the target pdf π . For the Metropolis-Hastings algorithm[129], the acceptance probability $\alpha(\mathbf{y}, \mathbf{z})$ is

$$\alpha(\mathbf{y}, \mathbf{z}) = \min \left(1, \frac{\pi(\mathbf{z}) q(\mathbf{z}, \mathbf{y})}{\pi(\mathbf{y}) q(\mathbf{y}, \mathbf{z})} \right) \quad (3.18)$$

For this choice of acceptance probability, the chain is positive recurrent with a stationary pdf π as described next. The transition kernel for the Metropolis-Hastings algorithm when $\mathbf{y} \neq \mathbf{z}$ is:

$$\begin{aligned} A(\mathbf{y}, \mathbf{z}) &= q(\mathbf{y}, \mathbf{z}) \alpha(\mathbf{y}, \mathbf{z}) \\ &= q(\mathbf{y}, \mathbf{z}) \min \left(1, \frac{\pi(\mathbf{z}) q(\mathbf{z}, \mathbf{y})}{\pi(\mathbf{y}) q(\mathbf{y}, \mathbf{z})} \right), \end{aligned} \quad (3.19)$$

When $\mathbf{y} \neq \mathbf{z}$, there are two possibilities, namely $\pi(\mathbf{z})q(\mathbf{z}, \mathbf{y}) \geq \pi(\mathbf{y})q(\mathbf{y}, \mathbf{z})$ and $\pi(\mathbf{z})q(\mathbf{z}, \mathbf{y}) < \pi(\mathbf{y})q(\mathbf{y}, \mathbf{z})$. For brevity, only the first case will be considered for the proof (second case can be proved similarly, but omitted here). When $\pi(\mathbf{z})q(\mathbf{z}, \mathbf{y}) \geq \pi(\mathbf{y})q(\mathbf{y}, \mathbf{z})$, one has [130]

$$\begin{aligned}
 \int \pi(\mathbf{y}) A(\mathbf{y}, \mathbf{z}) d\mathbf{y} &= \int \pi(\mathbf{y}) q(\mathbf{y}, \mathbf{z}) \min\left(1, \frac{\pi(\mathbf{z})q(\mathbf{z}, \mathbf{y})}{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{z})}\right) d\mathbf{y} \\
 &= \int \pi(\mathbf{y}) q(\mathbf{y}, \mathbf{z}) d\mathbf{y} \\
 &= \int \pi(\mathbf{z}) q(\mathbf{z}, \mathbf{y}) \min\left(1, \frac{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{z})}{\pi(\mathbf{z})q(\mathbf{z}, \mathbf{y})}\right) d\mathbf{y} \\
 &= \pi(\mathbf{z}) \int q(\mathbf{z}, \mathbf{y}) \min\left(1, \frac{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{z})}{\pi(\mathbf{z})q(\mathbf{z}, \mathbf{y})}\right) d\mathbf{y} \\
 &= \pi(\mathbf{z}) \int A(\mathbf{z}, \mathbf{y}) d\mathbf{y} \\
 &= \pi(\mathbf{z}).
 \end{aligned} \tag{3.20}$$

This fact proves that the acceptance probability $\alpha(\mathbf{y}, \mathbf{z})$ in Eq. (3.18) leads to the stationary pdf π which is the target pdf.

3.3.1 Random-Walk Metropolis Algorithm

The Metropolis-Hastings algorithm has the following attractive features: (1) the normalization constant $p(\mathbf{d})$ in Eq. (2.10) is not needed; (2) the candidate point \mathbf{z} can be generated from any proposal distribution $q(\mathbf{y}, \mathbf{z})$. For simplicity, the so-called random-walk Metropolis algorithm [60] is used. Its proposal distribution is widely used in practice and it is implemented as follows: the new candidate \mathbf{y} is obtained from the current point \mathbf{z} using $\mathbf{y} = \mathbf{z} + w\mathbf{r}$, where \mathbf{r} is drawn from an n -dimensional uniform distribution on $[-1, 1]^n$ and w is a positive constant. If w is selected to be too small, the chain will mix too slowly. On the other hand, a significant number of proposed samples would be

discarded for large w . In either case, the algorithm will have a small convergence rate, i.e. more samples would be needed to provide accurate estimates of characteristics (e.g. mean) of the target distribution. The convergence rate of an MCMC sampler can be maximized by choosing a value of w that minimizes the integrated autocorrelation time of the Markov chain, estimated from the samples generated by the MCMC algorithm [60]. Roberts *et al.* [131] proposed a general rule of thumb in choosing w such that an acceptance rate of approximately 25% is achieved for target distributions of dimension much larger than 2. An adaptive approach for choosing w is exercised: starting with a small value for w resulting in an acceptance ratio of almost unity, w is increased in steps until the desired acceptance ratio of 25% is achieved, at which point w is fixed and the previously generated samples are discarded.

3.4 Application: Signal Identification

In this example, the application of both maximum likelihood estimation and Bayesian inference will be compared and contrasted for identification of a sinusoidal signal. More specifically, the problem involves identifying the parameters a_1, a_2, a_3 for

$$y(t) = a_1 \cos(a_3 t) + a_2 \sin(a_3 t) . \quad (3.21)$$

The noisy measurements $y_k, i = 1, \dots, N$ are obtained as

$$y_k = a_1 \cos(a_3 t_k) + a_2 \sin(a_3 t_k) + n_k \quad (3.22)$$

$$= \hat{y}_k + n_k \quad (3.23)$$

where n_k is IID Gaussian noise with zero mean and variance equal to γ .

3.4.1 Maximum Likelihood Estimation

The likelihood function of the unknown parameters is

$$\begin{aligned} p(\mathbf{a}|\mathbf{y}) &= p(\mathbf{n} = \mathbf{y} - \hat{\mathbf{y}}_k(\mathbf{a})) \\ &= \frac{1}{(2\pi\gamma)^{N/2}} \exp \left[\sum_{k=1}^N -\frac{1}{2\gamma} (y_k - \hat{y}_k(a_1, a_2, a_3))^2 \right]. \end{aligned} \quad (3.24)$$

The corresponding misfit function (also known as the log-likelihood function) is given by

$$\varepsilon(\mathbf{a}; \mathbf{y}) = \sum_{i=1}^N \frac{1}{\gamma} (y_k - \hat{y}_k(a_1, a_2, a_3))^2 \quad (3.25)$$

which is simply a sum of squares of errors between the observed signal and its model. An estimate of the parameters can be obtained by maximizing the likelihood function or, equivalently, minimizing the misfit.

The parameters may be divided into two classes. In the first class one has a_1 and a_2 on which y depends linearly and in the second class one has a_3 on which y depends nonlinearly. The minimization can be performed over the linear parameters analytically, thus reducing the size of the search space for the nonlinear parameters.

The misfit function can be rewritten as [101]

$$\varepsilon(\mathbf{a}; \mathbf{y}) = \frac{1}{\gamma} (\mathbf{y} - \mathbf{C}(\mathbf{a}_{nl}) \mathbf{a}_l)^T (\mathbf{y} - \mathbf{C}(\mathbf{a}_{nl}) \mathbf{a}_l) \quad (3.26)$$

where

$$\mathbf{C}(\mathbf{a}_{nl}) = \begin{bmatrix} \cos(a_3 t_1) & \sin(a_3 t_1) \\ \vdots & \vdots \\ \cos(a_3 t_N) & \sin(a_3 t_N) \end{bmatrix} \quad (3.27)$$

and

$$\mathbf{a}_l = \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} \quad (3.28)$$

The least-square estimation of \mathbf{a}_l can be performed. Thus, the misfit is now represented as a function of $\mathbf{a}_{nl} = a_3$ only. \mathbf{a}_l can be replaced by its least-squares estimate $[\mathbf{C}(\mathbf{a}_{nl})^T \mathbf{C}(\mathbf{a}_{nl})]^{-1} \mathbf{C}(\mathbf{a}_{nl})^T \mathbf{y}$ in Eq. (3.26) to obtain [101]:

$$\varepsilon(\mathbf{a}; \mathbf{y}) = \frac{1}{\gamma} \left(\mathbf{y} - \mathbf{C}(\mathbf{a}_{nl}) [\mathbf{C}(\mathbf{a}_{nl})^T \mathbf{C}(\mathbf{a}_{nl})]^{-1} \mathbf{C}(\mathbf{a}_{nl})^T \mathbf{y} \right)^T \times \\ \left(\mathbf{y} - \mathbf{C}(\mathbf{a}_{nl}) [\mathbf{C}(\mathbf{a}_{nl})^T \mathbf{C}(\mathbf{a}_{nl})]^{-1} \mathbf{C}(\mathbf{a}_{nl})^T \mathbf{y} \right)$$

The misfit function, in this form, can be minimized easily to obtain an estimate of a_3 .

Case 1

For the first case examined, measurements of the signal are sampled at a frequency well above the Nyquist frequency. The true values of the parameters are $a_1 = 1.0$, $a_2 = 1.0$, $a_3 = 1.5$ and $\gamma = 0.05$. The true signal and measurements available are shown in Fig. 3.1. The misfit function for a_3 is shown in Fig. 3.2. The value of a_3 coinciding with the minimum of the misfit function is used as an estimate. Next, the least-square estimates a_1 and a_2 are obtained using the optimal value of a_3 . The resulting reconstructed signal is shown in Fig. 3.3.

Case 2

In this case, the available measurements are sparse in time such that the sampling frequency is just below the Nyquist frequency. The measurements are shown in Fig. 3.4. The misfit function for a_3 is shown in Fig. 3.5.

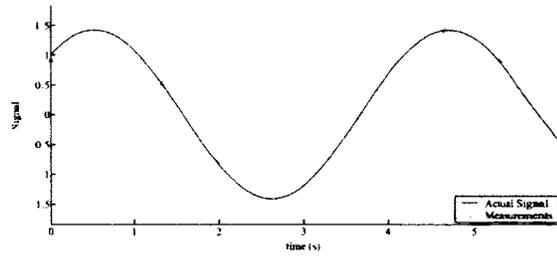


Figure 3.1: Application of MCMC: Signal Identification. Case 1: True signal and measurements.

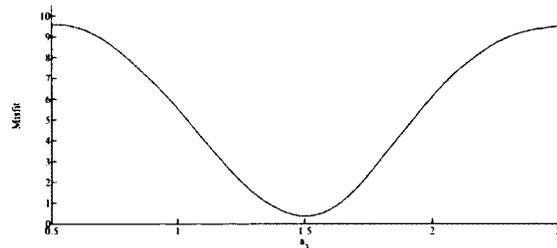


Figure 3.2: Application of MCMC: Signal Identification. Case 1: Misfit function for a_3 .

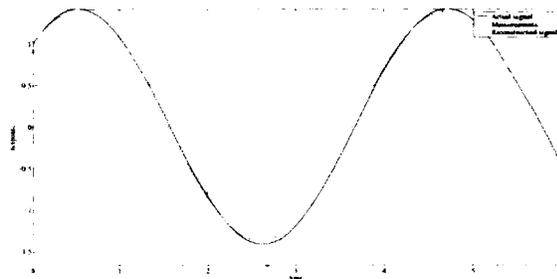


Figure 3.3: Application of MCMC: Signal Identification. Case 1: Reconstructed signal using Maximum Likelihood Estimation.

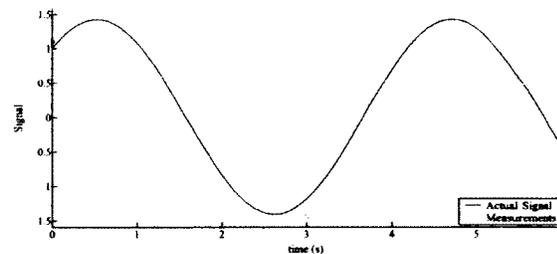


Figure 3.4: Application of MCMC: Signal Identification. Case 2: True signal and measurements.

Due to sparse measurements, maximum likelihood estimation no longer provides a unique estimate of a_3 . Thus there are many possible reconstructions of the signal, all

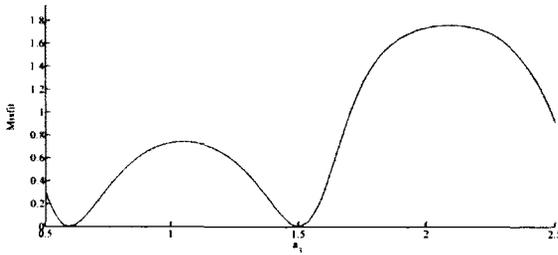


Figure 3.5: Application of MCMC: Signal Identification. Case 2: Misfit function for a_3 .

equally likely. In this case, the maximum likelihood estimation method fails to provide a unique solution.

3.4.2 Bayesian inference

According to Bayes' theorem, the posterior pdf of a_1, a_2 and a_3 is proportional to the product of likelihood function and prior pdf:

$$p(a_1, a_2, a_3 | y_1, \dots, y_N) \propto p(y_1, \dots, y_N | a_1, a_2, a_3) p(a_1, a_2, a_3)$$

where

$$p(y_1, \dots, y_N | a_1, a_2, a_3) \propto \prod_{i=1}^N p(y_i | a_1, a_2, a_3)$$

$$p(a_1, a_2, a_3) = \begin{cases} c, & \text{if } \text{abs}(a_1 - m_1) < w_1 \\ & \text{and } \text{abs}(a_2 - m_2) < w_2 \\ & \text{and } \text{abs}(a_3 - m_3) < w_3 \\ 0, & \text{otherwise} \end{cases}$$

If $w_1 = w_2 = w_3 = \infty$, a *flat prior* is used. A flat prior provides no information regarding the parameters to be estimated and thus the posterior pdf is equal to the

likelihood pdf. In certain cases, the likelihood function needs to be complemented by an informative prior to provide a meaningful estimate of the parameters. The above two cases will be revisited, utilizing Bayesian inference to provide posterior pdfs of the unknown parameters. The Metropolis-Hastings algorithm is used to provide MCMC samples of these parameters.

Case 1

For this case, as the data is dense in time, a flat prior is used, i.e. $w_1 = w_2 = w_3 = \infty$. The likelihood function provides sufficient information to provide a unique maximum-likelihood estimate. The MCMC samples of the posterior pdf are used to provide marginal and joint distributions of the parameters as shown in Figs. 3.6-3.7. One can observe the unimodality and slight non-Gaussian trends visible both the marginal and joint pdfs.

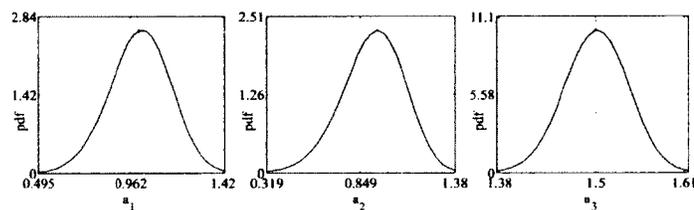


Figure 3.6: Application of MCMC: Signal Identification. Case 1: Marginal pdfs obtained using MCMC sampling of the posterior pdf.

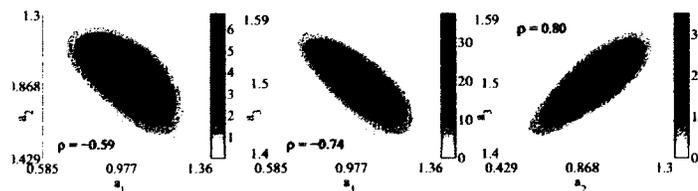


Figure 3.7: Application of MCMC: Signal Identification. Case 1: Joint pdfs obtained using MCMC sampling of the posterior pdf.

Case 2

A flat (non-informative) prior is first used for this case. The marginal and joint pdfs of the unknown parameters are shown in Figs. 3.8-3.9. One can see the unimodality and the slight non-Gaussian trends visible both the marginal and joint pdfs. One can see with sparse measurements, two disjoint sets of frequency values are possible given sparse data.

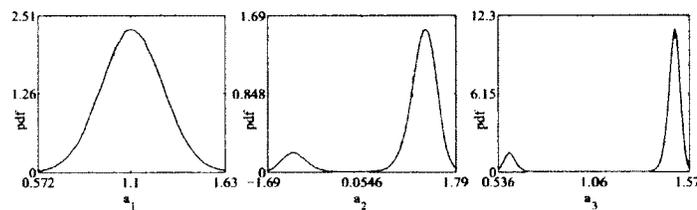


Figure 3.8: Application of MCMC: Signal Identification. Case 2: Marginal pdfs obtained using MCMC sampling of the posterior pdf with flat prior.

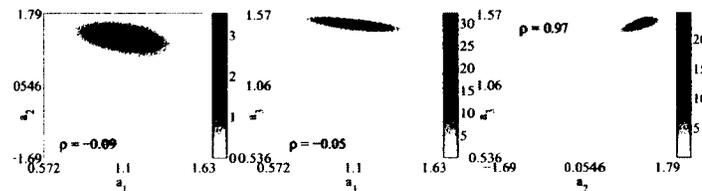


Figure 3.9: Application of MCMC: Signal Identification. Case 2: Joint pdfs obtained using MCMC sampling of the posterior pdf with flat prior.

If a uniform prior pdf (with finite support) is used, the results in Figs. 3.10-3.11 are obtained. An informative prior in this case provides a regularized solution. One can extract an estimate from this posterior distribution, such as the minimum variance estimate (being the mean) or the maximum a posteriori estimate (being the mode) (see Section 2.2.3 for definitions). This case demonstrates the power of Bayesian inference in certain cases in situations when the likelihood function does not provide adequate information for a unique estimate.

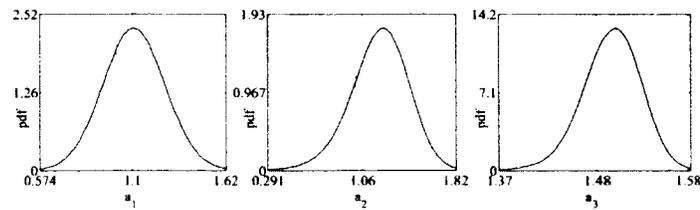


Figure 3.10: Application of MCMC: Signal Identification. Case 2: Marginal pdfs obtained using MCMC sampling of the posterior pdf with uniform prior.

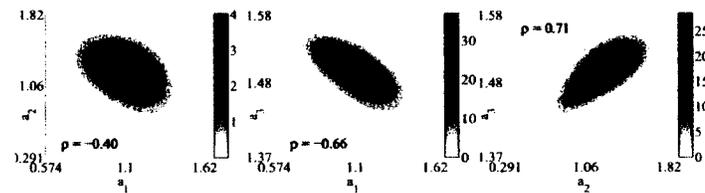


Figure 3.11: Application of MCMC: Signal Identification. Case 2: Joint pdfs obtained using MCMC sampling of the posterior pdf with uniform prior.

3.5 Application: Flutter Speed Estimation

Flutter is a potential catastrophic vibration phenomenon that can occur on aircraft past a given airspeed (known as flutter speed). The flutter margin method [132] is an experimental technique that provides a measure of stability of an aircraft based on its vibration data and in turn predicts flutter onset. In the following section, a probabilistic framework is provided that offers a rational treatment of the uncertainty in the vibration data obtained from pre-flutter flight tests. Some of the outcomes of the improved flutter flight testing technique developed herein are (a) a decrease in the number of flutter flight test points (thus saving costs), and (b) an increased confidence in the flutter speed prediction (thus decreasing the risk involved). The work contained in this section has already been published in peer-reviewed literature and presented at numerous conferences [88–90].

3.5.1 Introduction

The flutter margin, a quantity first utilized by Zimmerman and Weissenburger [132], is a measure of stability of an aeroelastic system. On the basis that the instability mechanism is dictated by the coalescence of the two modes of vibration that participate in flutter, the typical section model (two degrees-of-freedom (DOF), in pitch and heave) is used [133]. The flutter margin is equivalent to the third sub-determinant of the fourth-order characteristic equation of the typical section model [132]. Hence, according to the Routh's stability criteria [133], a necessary condition for stability is that the flutter margin must be positive. The flutter margin is given in terms of ω_1, ω_2 and β_1, β_2 being the modal frequencies and negative of decay rates, respectively, of the two modes that participate in flutter:

$$\begin{aligned}
 F = & \left[\left(\frac{\omega_2^2 - \omega_1^2}{2} \right) + \left(\frac{\beta_2^2 - \beta_1^2}{2} \right) \right]^2 \\
 & + 4\beta_1\beta_2 \left[\left(\frac{\omega_2^2 + \omega_1^2}{2} \right) + 2 \left(\frac{\beta_2 + \beta_1}{2} \right)^2 \right] \\
 & - \left[\left(\frac{\beta_2 - \beta_1}{\beta_2 + \beta_1} \right) \left(\frac{\omega_2^2 - \omega_1^2}{2} \right) + 2 \left(\frac{\beta_2 + \beta_1}{2} \right)^2 \right]^2. \quad (3.29)
 \end{aligned}$$

The flutter margin can be used to extrapolate the classical flutter speed from pre-flutter flight test data. Under modest restrictions, the flutter margin varies parabolically in dynamic pressure if structural damping is neglected. The inclusion of structural damping in the analysis leads to a sixth-order relationship between the flutter margin and airspeed. For larger airspeeds the relationship converges to a fourth-order polynomial, as per Eq. (3.30), since aerodynamic damping dominates structural damping.

$$F = b_1 U^4 + b_2 U^2 + b_3. \quad (3.30)$$

The coefficients b_1 through b_3 depend on configuration constants which describe the aeroelastic system (coupled structural and aerodynamic systems) [132]. However, in the absence of prior knowledge regarding the configurational parameters of the aeroelastic system, these coefficients can be inferred using conventional methodology from pre-flutter estimates of the flutter margin F using a least-square fitting procedure of Eq. (3.30) with at least three sub-critical airspeeds. The flutter speed, U_f , is the root of Eq. (3.30).

Since its establishment in 1964, the flutter margin method has been generalized to tackle various cases overlooked by the basic method. Price and Lee [134] extended the method to trinary flutter, for instance. Non-probabilistic robust control theory has lead widely used uncertainty quantification methods in aeroelasticity [135]. More specific to aeroelastic stability, Lind and Brenner have developed the application of non-probabilistic robust control methods to flight testing [136]. Recently, however, probabilistic methods have received more attention in the context of dynamic stability of aircraft. For example, the uncertainty in the modal parameters has been considered by Poirel *et al.* [137] to provide a probability density function (pdf) of the flutter margin and the flutter speed. The basic idea is to directly carry out the uncertainty analysis of the modal parameters to construct the flutter margin and flutter speed histograms to retrieve relevant statistical information. It was pointed out that, although the modal parameters were assumed to be Gaussian, the flutter speed pdf was not symmetric, but skewed. This methodology was further examined by Heeg [138]. A maximum likelihood parameter estimation method was used to construct a pdf of the flutter speed. Khalil *et al.* [88] provided a pdf of the flutter speed by estimating the joint distribution of the flutter margin

equation coefficients in Eq. (3.30). The flutter margin pdfs were obtained from modal parameter estimates which were artificially perturbed by Gaussian noise.

For this application, the joint distribution of the modal parameters is obtained from measured noisy free decay responses of the system. Bayesian inference provides a posterior pdf of these parameters and Markov Chain Monte Carlo (MCMC) sampling technique [58, 60] is used to sample from the posterior pdf. The flutter margin pdf is then constructed from the samples of the parameters. Finally, the joint distribution of the flutter margin equation coefficients in Eq. (3.30) and the flutter speed are obtained again using Bayesian inference. For numerical illustration, the proposed method is applied for a two-degrees-of-freedom numerical model, for which the true flutter speed is known.

3.5.2 Formulation

This section describes the application of Bayesian inference for flutter speed estimation problem. For given airspeeds $U_i, i = 1, \dots, n_u$, one can estimate the joint distribution of the modal parameters of the two modes involved in coalescence flutter from noisy measurements of free decay system response using Bayesian inference. Assuming the typical section model (two degree-of-freedom model describing the pitch and heave motion) is utilized, the problem involves identifying the modal parameters a_1, \dots, a_4 that describe the free decay response of the system given unknown initial conditions [139]:

$$u(t) = a_5 e^{-a_2 t} \cos(a_1 t + a_6) + a_7 e^{-a_4 t} \cos(a_3 t + a_8) \quad (3.31)$$

from noisy measurements

$$u_k = a_5 e^{-a_2 t_k} \cos(a_1 t_k + a_6) + a_7 e^{-a_4 t_k} \cos(a_3 t_k + a_8) + n_k \quad (3.32)$$

$$= \hat{u}_k(a_1, \dots, a_8) + n_k \quad (3.33)$$

where n_k is Gaussian noise with zero mean and variance equal to γ^2 .

The likelihood function of these unknown parameters is

$$p(a_1, \dots, a_8 | u_1, \dots, u_N) = \frac{1}{(2\pi\gamma)^{N/2}} \exp \left[\sum_{k=1}^N -\frac{1}{2\gamma^2} (u_k - \hat{u}_k(a_1, \dots, a_8))^2 \right] \quad (3.34)$$

$\hat{u}_k(a_1, \dots, a_8)$ is the predicted free decay system responses using the assumed values of a_1, \dots, a_8 . Samples of these parameters (and thus samples of modal parameters $\omega_1 = a_1$, $\omega_2 = a_3$ and $\beta_1 = a_2$, $\beta_2 = a_4$) are obtained from this joint pdf using MCMC sampling technique. The modal parameter samples are consequently transformed using Eq. (3.29) to obtain flutter margin samples from which a flutter margin pdf is constructed. The flutter margin pdf obtained at each experimental airspeed using the proposed method will be denoted by $p(F_i | \mathbf{b})$, being conditional on the unknown coefficients $\mathbf{b} = \{b_1, b_2, b_3\}$ from Eq. (3.30).

The joint distribution of the vector $\mathbf{b} = \{b_1, b_2, b_3\}$ of unknown coefficients is first estimated. This would in turn provide a pdf for the flutter speed being the root of Eq. (3.30). Let $\mathbf{F} = \{F_1, F_2, \dots, F_{n_u}\}$ represent the random vector of the flutter margin at airspeeds U_i , $i = 1, \dots, n_u$. By Bayes' theorem

$$p(b_1, b_2, b_3 | F_1, \dots, F_{n_u}) \propto p(F_1, \dots, F_{n_u} | b_1, b_2, b_3) p(b_1, b_2, b_3), \quad (3.35)$$

or more simply put

$$p(\mathbf{b}|\mathbf{F}) \propto p(\mathbf{F}|\mathbf{b}) p(\mathbf{b}) . \quad (3.36)$$

The marginal distributions of the flutter margin at various experimental airspeeds are obtained from experimental data. Experimental data does not provide information regarding the interdependence between the flutter margins at the different airspeeds. Thus, without further knowledge, the full joint distribution of the flutter margins required in Eq. (3.36) is not available. The maximum entropy theory is used to justify the assumption of independence between the flutter margins at differing airspeeds [140], i.e. $p(\mathbf{F}|\mathbf{b})$ can be expressed as the product of the marginal pdfs of F_i as in

$$p(\mathbf{F}|\mathbf{b}) = \prod_{i=1}^{n_u} p(F_i|\mathbf{b}) . \quad (3.37)$$

The prior pdf of \mathbf{b} is expressed as

$$p(\mathbf{b}) \propto \begin{cases} 1, & \text{if } b_2^2 - 4b_1b_3 > 0 \text{ (i.e. flutter will occur)} \\ 0, & \text{otherwise} \end{cases} . \quad (3.38)$$

The above form for $p(\mathbf{F}|\mathbf{b})$ is valid under the simplifying assumption that the flutter margin at one airspeed is independent from the flutter margin at other airspeeds. As for the prior pdf $p(\mathbf{b})$ in Eq. (3.38) of the unknown coefficients, it reflects the assumption that the polynomial in Eq. (3.30) has real roots.

As a result, the conditional pdf for \mathbf{b} in Eq. (3.35) has a complicated mathematical expression and the normalization constant is hard to obtain. Consequently, MCMC sampling technique is applied to obtain samples of \mathbf{b} from this complex conditional distribution $p(\mathbf{b}|\mathbf{F})$. These would in turn provide samples of the flutter speed U_f , being

the root of the flutter margin equation, Eq. (3.30), whose coefficients are \mathbf{b} . The specific MCMC sampling technique used is the Metropolis-Hastings algorithm which was reviewed previously. The flow chart in Fig. 3.12 outlines the steps required to obtain the posterior pdf of the flutter speed starting with vibration data.

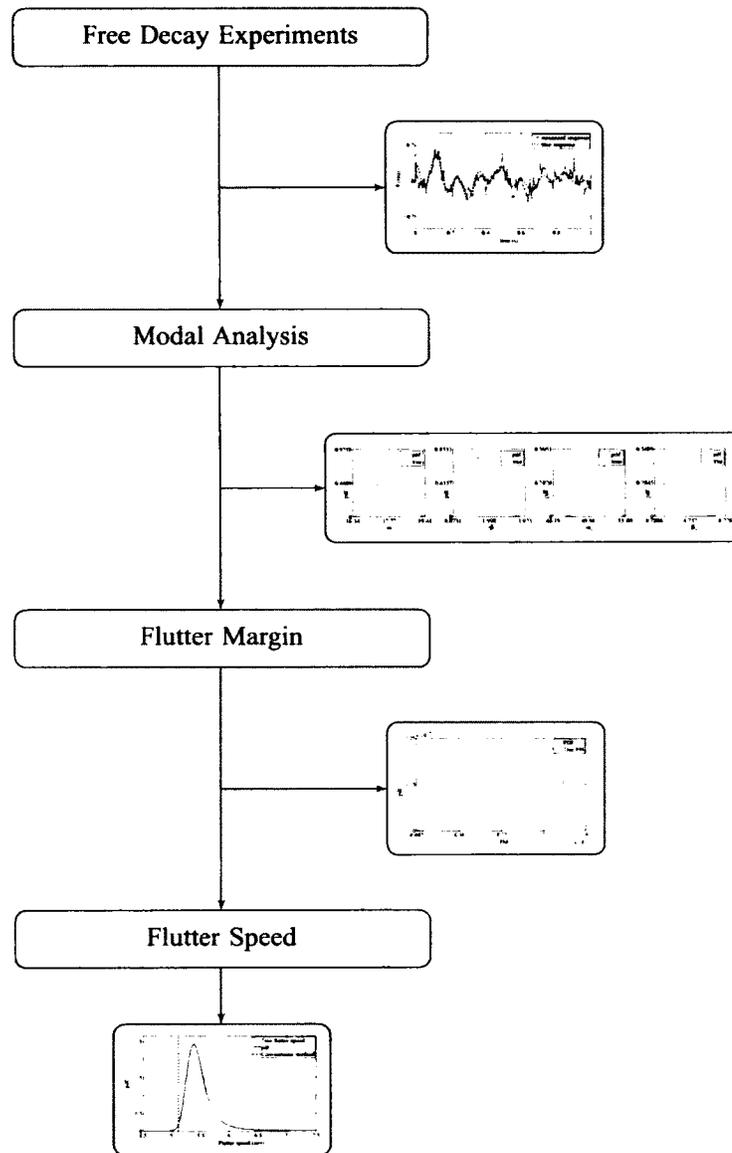


Figure 3.12: Flow chart describing the probabilistic flutter margin method.

Airspeed	$\omega_1(\text{rad/s})$	$\beta_1(1/\text{s})$	$\omega_2(\text{rad/s})$	$\beta_2(1/\text{s})$	F
1.95	17.3	-1.40	51.6	-1.88	1.39
2.51	17.2	-1.91	50.3	-2.31	1.26
3.06	17.1	-2.52	48.6	-2.63	1.10
3.62	17.0	-3.30	46.4	-2.79	0.91
4.18	16.7	-4.35	43.9	-2.68	0.69
4.73	16.1	-5.85	40.9	-2.11	0.44

Table 3.1: Application of MCMC: Flutter Speed Estimation. True modal parameters and flutter-margin for 2-DOF airfoil.

3.5.3 Application to a Two-DOF Airfoil

The proposed methodology is first applied to a 2-DOF airfoil model with quasi-steady aerodynamics model (see Appendix E for equation of motion) and compare these results with the conventional flutter margin method [88–90, 137]. The results are then compared with the known (deterministic) flutter speed, which is 5.57 m/s for the airfoil parameters chosen as determined by eigenvalue analysis. Note that quasi-steady aerodynamics is assumed and structural damping is neglected. Six equally spaced airspeeds were chosen for numerical experiments with the lowest airspeed being 35% of the flutter speed and the highest airspeed is at 85% of the flutter speed. The chosen airspeeds for the analysis and the respective modal parameter and flutter margin values are summarized in Tab. 3.1.

At each airspeed, noisy measurements of the free decay response in pitch are obtained over an interval of 1.0 seconds. The initial condition for the system is -0.1 m in heave and 0.1 rad in pitch with velocities equal to zero for both degrees of freedom. The standard deviation of measurement noise for the pitch measurements is chosen as $\gamma = 0.4$. The measurement noise intensity is chosen so that the resulting level of scatter in the modal parameters is similar to that observed in practice: coefficient of variation of 1% for frequency and 30% for damping [134]. The measurements were obtained with a sampling rate of 50 Hz. Figs. 3.13-3.18 display the true trajectories of the system and the

corresponding measured responses at all six airspeeds in Tab. 3.1. These experiments are repeated nine other times while changing the measurement noise realization to obtain a total of ten sets of free decay responses at each of the six airspeeds.

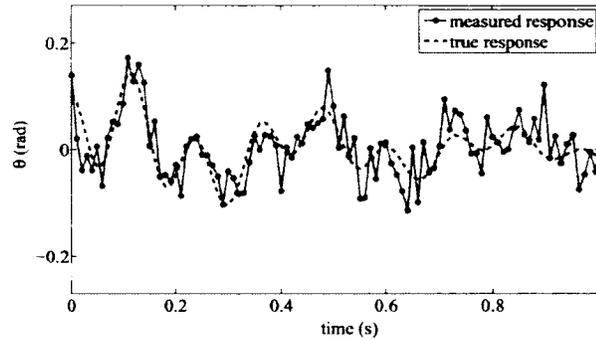


Figure 3.13: Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 1.95$ m/s - experiment 1.

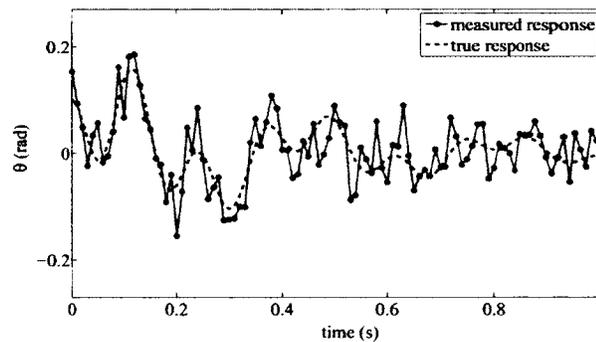


Figure 3.14: Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 2.51$ m/s - experiment 1.

MCMC is first applied to obtain jointly distributed samples for the 8 unknown parameters a_1, \dots, a_8 in Eq. (3.34) for each measured free decay response. The random-walk Metropolis algorithm (see Section 3.3) is used to generate these samples using an 8-dimensional uniform distribution on $[-w, w]^8$ as the proposal density where w is a positive constant. w is chosen adaptively such that an acceptance rate of approximately 25% is achieved [131].

At each airspeed, 2×10^6 MCMC samples of the modal parameters were obtained

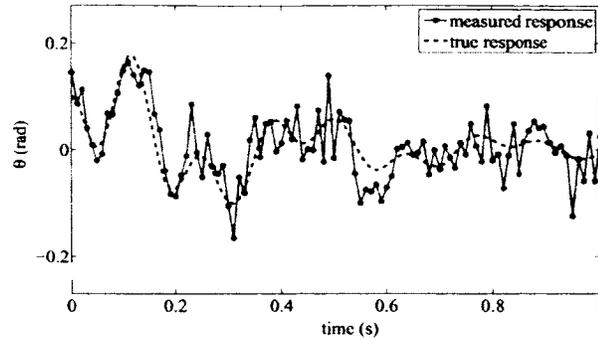


Figure 3.15: Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 3.06$ m/s - experiment 1.

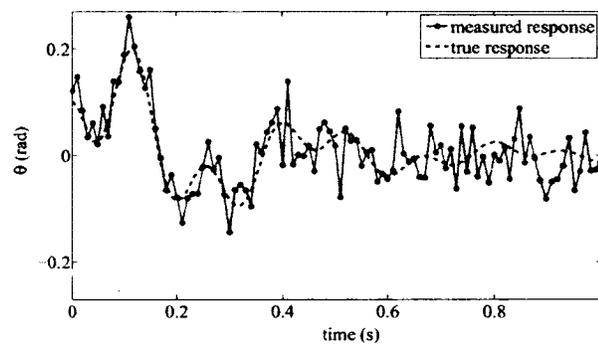


Figure 3.16: Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 3.62$ m/s - experiment 1.

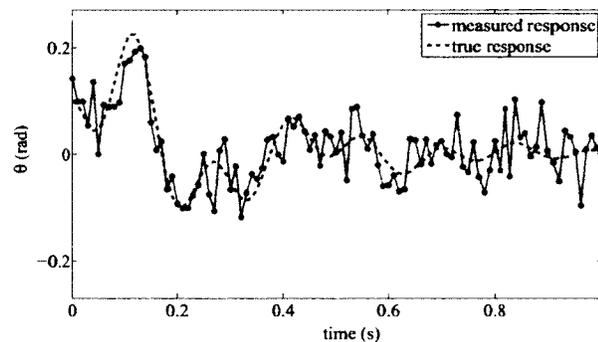


Figure 3.17: Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 4.18$ m/s - experiment 1.

with 170 MCMC chains running in parallel on a distributed-memory multiprocessor machine (HP Intel Xeon cluster with 176 processor cores) using message-passing interface (MPI) [77]. From these samples, one can extract the marginal pdfs of each modal pa-

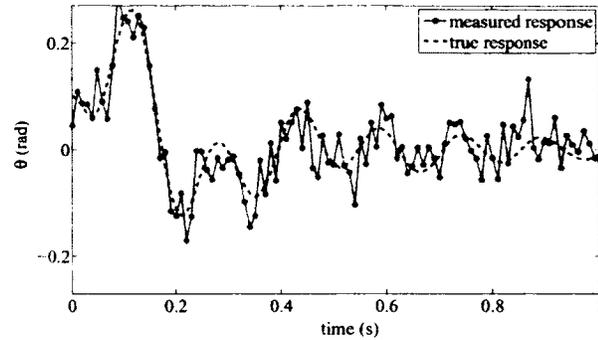


Figure 3.18: Application of MCMC: Flutter Speed Estimation. Pitch response of the 2-DOF airfoil at $U = 4.73$ m/s - experiment 1.

parameter at each air speed. For the first of ten experiments, these marginal pdfs are shown in Figs. 3.19-3.24. The dashed line indicates the true modal parameter value. One can see that the mode of the pdfs does not in general coincide with the true parameter value. Furthermore, it is also evident that there is some non-Gaussian trend in some of the marginal pdfs in the form of skewness. This is due to the limited information that the measured data contains relating to the modal parameters. One can also examine the joint distributions between the modal parameter estimates. These joint pdfs are presented in Figs. 3.28-3.33. It is also evident here that the joint distributions are non-Gaussian due to the limited amount of data.

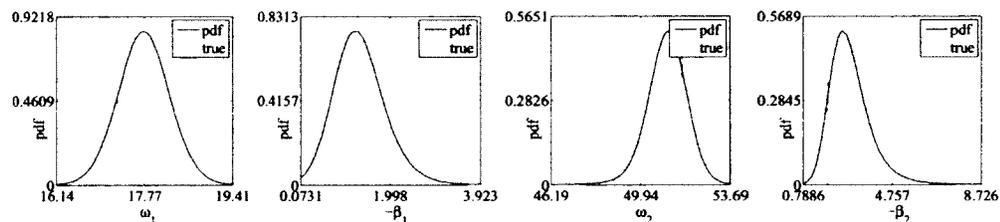


Figure 3.19: Application of MCMC: Flutter Speed Estimation. Marginal pdfs of modal parameters at $U = 1.95$ m/s - experiment 1.

One can apply the classical method in which flutter margin values at each airspeed are obtained using the most likely modal parameter values (corresponding to the peak of the joint distribution of the modal parameters). In this case one obtains six flutter margin

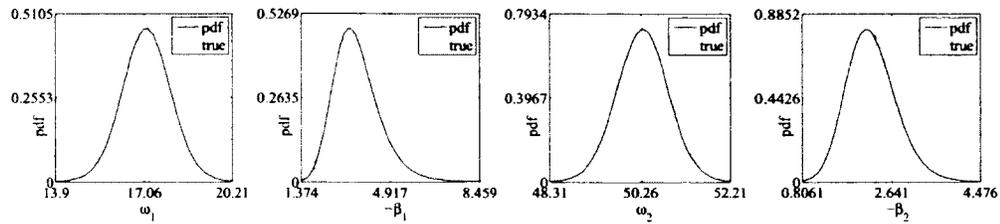


Figure 3.20: Application of MCMC: Flutter Speed Estimation.
Marginal pdfs of modal parameters at $U = 2.51$ m/s - experiment 1.

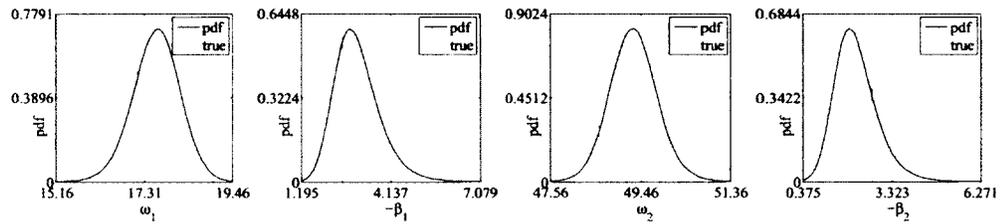


Figure 3.21: Application of MCMC: Flutter Speed Estimation.
Marginal pdfs of modal parameters at $U = 3.06$ m/s - experiment 1.

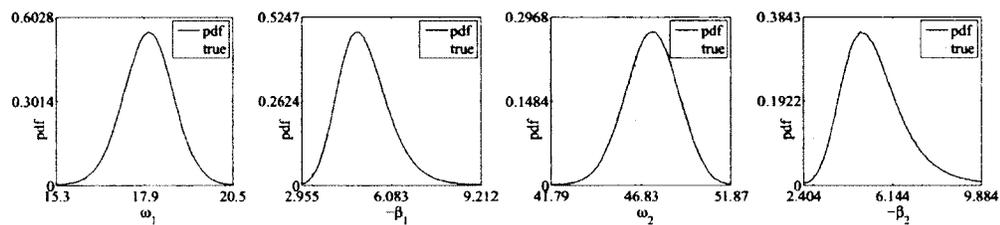


Figure 3.22: Application of MCMC: Flutter Speed Estimation.
Marginal pdfs of modal parameters at $U = 3.62$ m/s - experiment 1.

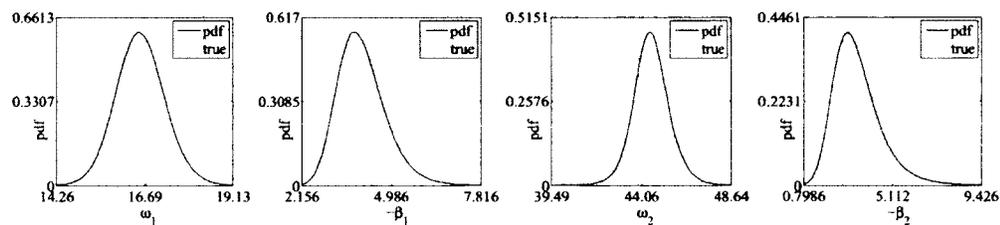


Figure 3.23: Application of MCMC: Flutter Speed Estimation.
Marginal pdfs of modal parameters at $U = 4.18$ m/s - experiment 1.

estimates at six airspeeds from which a quadratic polynomial in the square of the airspeed can be obtained using least-squares fitting. The error in the modal parameters is approximately 5% for frequency and 20% for damping for this experiment. For compar-

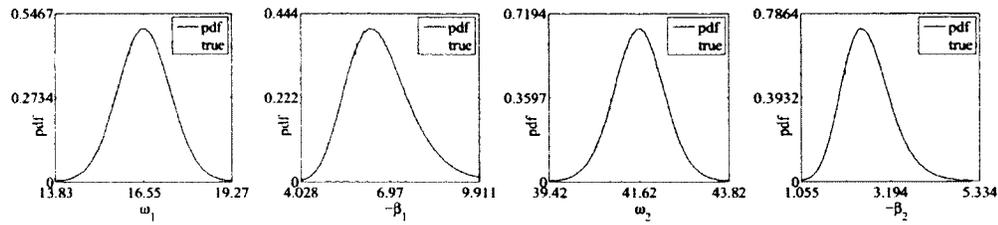


Figure 3.24: Application of MCMC: Flutter Speed Estimation.
Marginal pdfs of modal parameters at $U = 4.73$ m/s - experiment 1.

ison, the Eigensystem Realization Algorithm using Data Correlations (ERA/DC) [141] was also applied to the same data and a similar level of error was obtained for the modal parameter estimates.

Using the six estimated flutter margin values, a least-squares polynomial fit is obtained as shown in Fig. 3.25 for the first experiment. The extracted flutter speed using this method is 5.12 m/s and the true flutter speed is 5.57 m/s. Also shown is a linear fit for the flutter margin from which one can extrapolate an estimate for the flutter speed of 7.17 m/s. The analysis for nine other experiments is repeated (with the difference being the realization of measurement noise corrupting the true free decay response). The intent of these repeated experiments is to demonstrate the robustness of the estimates obtained from the proposed Bayesian method. The results using the polynomial fits and linear fits are shown in Figs. 3.26-3.27, respectively. In 2 of the 10 experiments, no flutter speed estimate was obtained using the conventional method (the fitted polynomial does not cross the zero flutter margin axis). The range of flutter speed estimates using this conventional approach for the eight experiments and the ensemble mean and coefficient of variation is given in Table 3.2. The range of flutter speed estimates is also provided using the linear fits for the ten experiments, the corresponding ensemble mean and coefficient of variation.

The joint pdfs of the modal parameters are presented in Figs. 3.28-3.33. One ob-

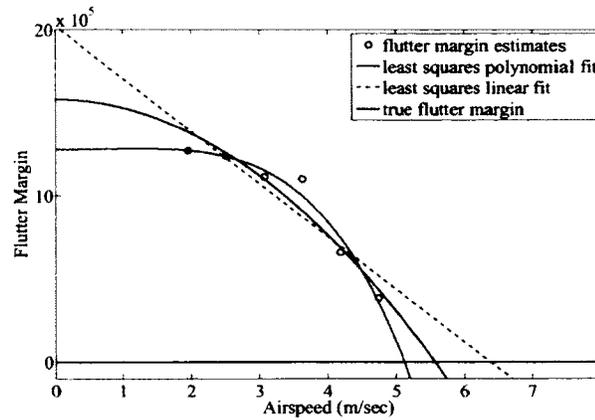


Figure 3.25: Application of MCMC: Flutter Speed Estimation. Flutter margin polynomial fit using conventional method - experiment 1.

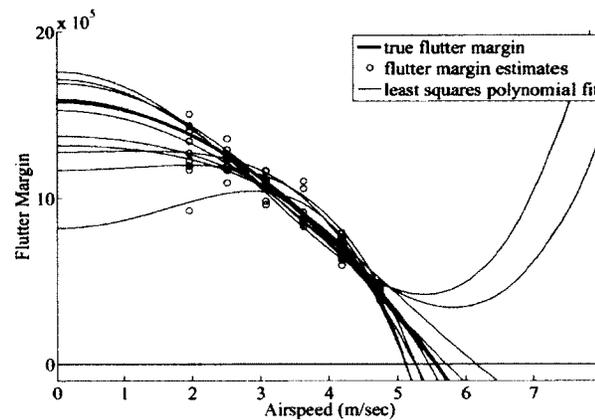


Figure 3.26: Application of MCMC: Flutter Speed Estimation. Flutter margin polynomial fit using conventional method - all experiments.

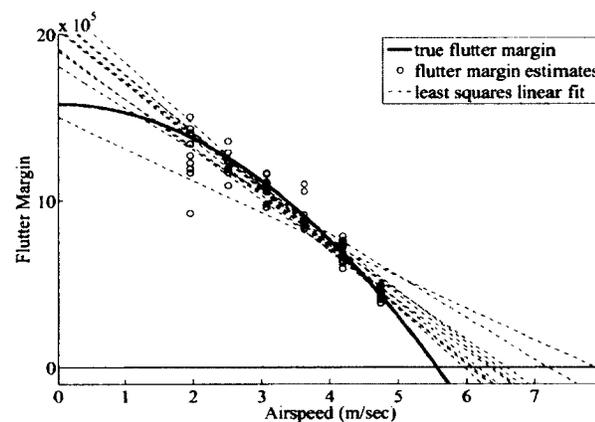


Figure 3.27: Application of MCMC: Flutter Speed Estimation. Flutter margin linear fit - all experiments.

serves a jointly Gaussian nature and small if any correlation between these parameters for different airspeeds. Using the MCMC samples obtained from the joint distribution between the four modal parameters, one obtains samples of the flutter margin and construct the flutter margin pdf at each airspeed. These pdfs are shown in Figs. 3.34-3.39. These flutter margin estimates at the five airspeeds represent $p(F_i|a)$ in Eq. (3.37). MCMC technique is used to obtain 1.7×10^9 samples of $\{a_1, a_2, a_3\}$ from the posterior pdf in Eq. (3.35). Each of these samples represent a possible flutter margin polynomial. In turn, 1.7×10^9 samples of the flutter speed are obtained, being the root of the flutter margin polynomial. The resulting histogram for the flutter speed pdf is shown in Fig. 3.40. One can see that both the true flutter speed as well as the estimate obtained using the conventional method fall within the support of the flutter speed pdf obtained using Bayesian inference. Furthermore, it is evident that if the mode (being the most likely value) at 5.37 m/s were to be used as an estimate for the flutter speed, Bayesian inference leads to a less conservative estimate, compared to the conventional method for this experiment (with an estimate of 5.12 m/s). The analysis is repeated for nine other experiments, obtaining a total of ten different flutter speed pdfs shown in Fig. 3.41. The range of flutter speed estimates (the mean, mode, median and 99% lower bound) for the ten experiments as well as the ensemble mean and coefficient of variation of each estimate are given in Table 3.2. The median provides on average the estimate with least error and scatter (measured using the coefficient of variation) in the median is of similar order of magnitude to those of the other estimates obtained from the pdfs.

Upon close examination of the data in Table 3.2, it can be concluded that the three estimates (namely the mode, mean and median) obtained using Bayesian inference are more precise (i.e. with less scatter) than those obtained using the conventional method. Furthermore, the three estimates obtained using the proposed method are more accu-

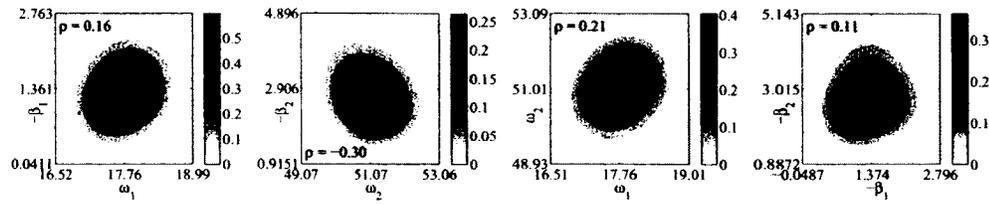


Figure 3.28: Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 1.95$ m/s - experiment 1.

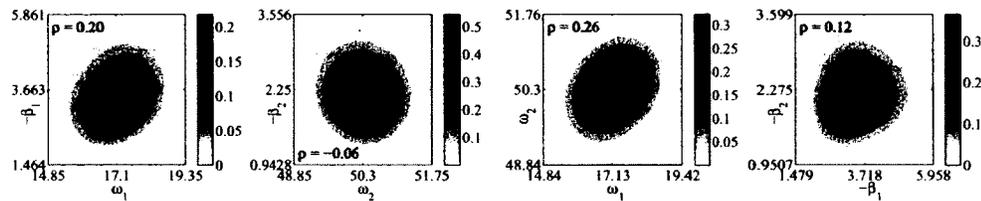


Figure 3.29: Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 2.51$ m/s - experiment 1.

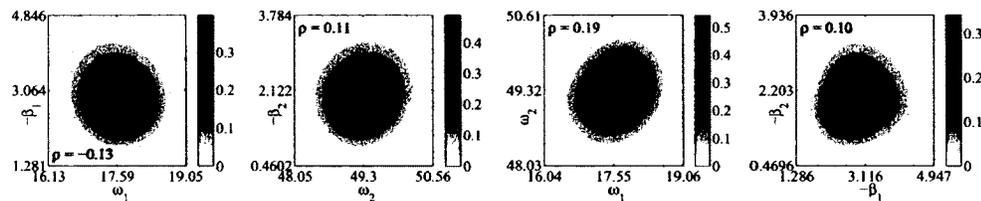


Figure 3.30: Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 3.06$ m/s - experiment 1.

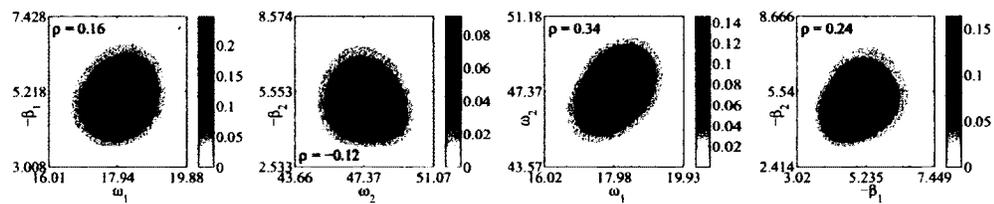


Figure 3.31: Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 3.62$ m/s - experiment 1.

rate (being closer to the true flutter speed on average), with the median being the most accurate based on the results obtained from the ten experiments.

In summary, a Bayesian estimation technique based on the Metropolis-Hastings MCMC algorithm was used to obtain the probability distribution function of the flutter speed

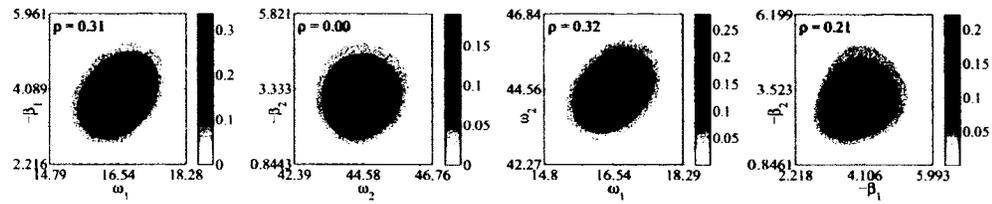


Figure 3.32: Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 4.18$ m/s - experiment 1.

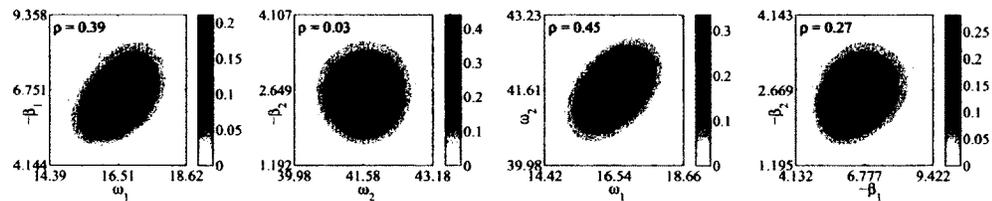


Figure 3.33: Application of MCMC: Flutter Speed Estimation. Joint pdfs of modal parameters at $U = 4.73$ m/s - experiment 1.

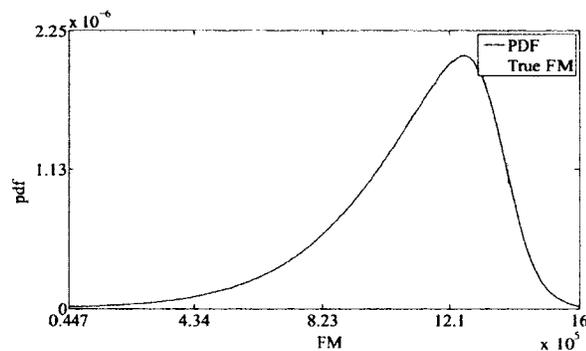


Figure 3.34: Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 1.95$ m/s - experiment 1.

from modal parameter estimates using the flutter margin method proposed by Zimmerman and Weissenburger [132]. A two-degrees-of-freedom numerical model was adopted for illustration. For the numerical experiments conducted, the method provided more accurate and more precise estimates for the flutter speed vis-a-vis the conventional method.

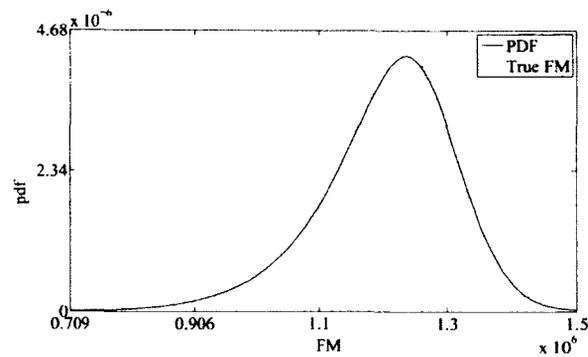


Figure 3.35: Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 2.51$ m/s - experiment 1.

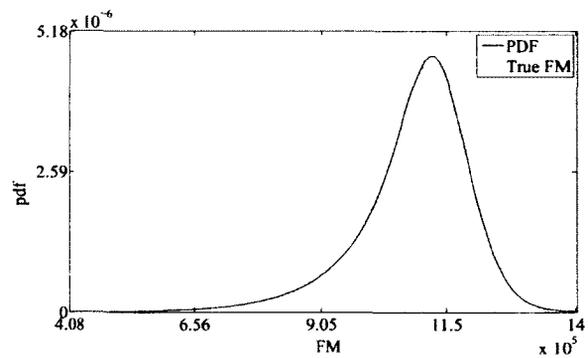


Figure 3.36: Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 3.06$ m/s - experiment 1.

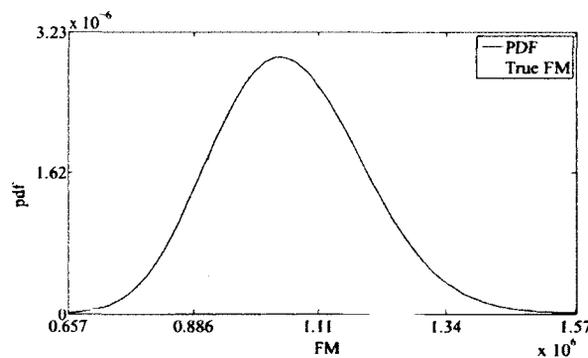


Figure 3.37: Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 3.62$ m/s - experiment 1.

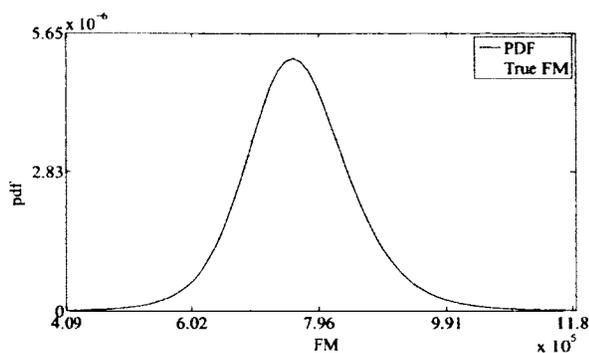


Figure 3.38: Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 4.18$ m/s - experiment 1.

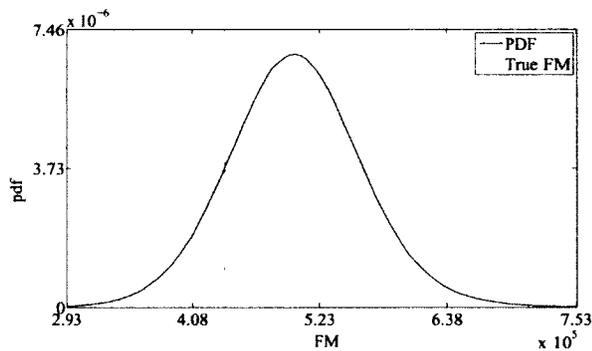


Figure 3.39: Application of MCMC: Flutter Speed Estimation. Flutter margin pdf at $U = 4.73$ m/s - experiment 1.

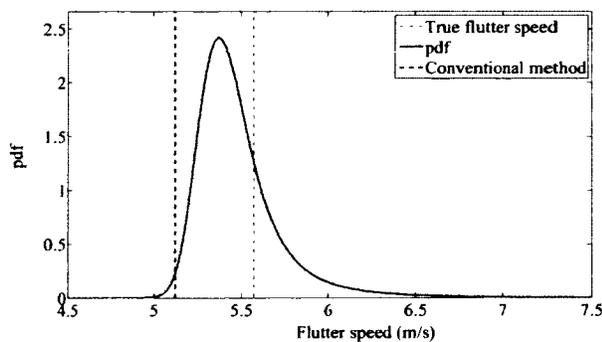


Figure 3.40: Application of MCMC: Flutter Speed Estimation. Flutter speed pdf obtained using Bayesian inference - experiment 1.

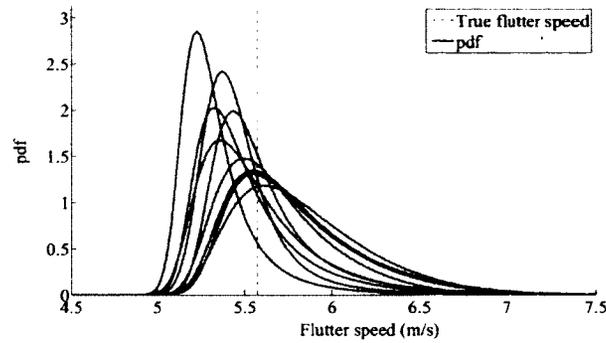


Figure 3.41: Application of MCMC: Flutter Speed Estimation. Flutter speed pdf obtained using Bayesian inference - all experiments.

Method and Estimate used	Range	Mean	Cov
Conventional, root	[5.11, 5.70]	5.32	0.041
Linear fit, root	[5.95, 7.83]	6.49	0.091
Bayesian inference, mean	[5.34, 5.82]	5.63	0.029
Bayesian inference, mode	[5.23, 5.62]	5.45	0.023
Bayesian inference, median	[5.29, 5.76]	5.56	0.028
Bayesian inference, 99% lower bound	[5.04, 5.23]	5.16	0.013

Table 3.2: Application of MCMC: Flutter Speed Estimation. Range of flutter speed estimates using 10 experiments.

Chapter 4

Static Parameter Estimation

4.1 Introduction

In Chapter 2, a review of various filtering techniques to tackle the problem of sequential data assimilation problem is provided. State estimates are updated as observations become available. The updated state vector is propagated in time between available measurements using the model operator. It was not explicitly discussed how to tackle the problem of parameter estimation using filtering. A popular approach to solving the parameter estimation problem using sequential data assimilation involves augmenting the state vector with the unknown system parameters. The new augmented state vector is estimated using one of the nonlinear filtering techniques presented in Chapter 2. In this approach, the parameter space is only explored at the initialization of the algorithm [50]. The judicious selection of the prior pdf of the unknown parameters is critical to the success of the filtering algorithms in estimating the unknown parameters. Unfortunately, it is difficult to select such prior for high-dimensional systems of practical interest to obtain accurate estimates of the unknown parameters. For poorly selected prior, the filtering algorithms fail (i.e. diverge to an incorrect estimate of the parameters).

To alleviate this problem, the time-invariant parameters are perturbed by artificial noise to achieve convergence of the filtering algorithm [47, 56]. To avoid such artificial dynamics, an approach proposed in [57] for PF algorithms consists of adding Markov Chain Monte Carlo (MCMC) steps to introduce some diversity in the parameter estimates. However, the algorithm may diverge due to the accumulation of errors over time [142]. For a restricted class of problems, Storvik [143] developed a methodology to marginalize the static (time-invariant) parameters from the posterior pdf. Storvik's approach [143] avoids the sample impoverishment in PF while estimating time-invariant parameters. For static parameter estimation, Jasper et al [144] used the Particle Markov Chain Monte Carlo method [145] which uses PF to design efficient proposal density for MCMC.

This chapter deals with the problem of static parameter estimation for dynamical systems using Bayesian inference. Firstly, a Markov Chain Monte Carlo (MCMC) algorithm is developed for the time-invariant parameter estimation which uses EnKF for state estimation. For this method, the state vector is not augmented with the unknown parameters as commonly adopted by some of the aforementioned methods. The results will be compared to those obtained from combined state and parameter estimation experiments using (a) EnKF, and (b) a newly developed Particle Filter (PF) with an EnKF proposal [48, 146] which has been specifically adapted for combined state and parameter estimation. Extensive numerical investigation is conducted using the Duffing oscillator model to study the effects of sparsity of observational data and strength of measurement noise on the parameter estimates. As the Duffing oscillator model displays a wide range of dynamical behaviour with small perturbations to the system parameters [115, 116], it is chosen for numerical investigation. The work contained in this chapter has been

submitted for publication in an international journal [91]¹.

4.2 Mathematical Formulation For Parameter Estimation

The model and measurement equations for a discrete state-space representation of a nonlinear system are given by [9, 10, 19, 20, 27, 40, 53, 147, 148]

$$\mathbf{u}_{k+1} = \mathbf{g}_k(\mathbf{u}_k, \boldsymbol{\phi}, \mathbf{f}_k, \mathbf{q}_k^u), \quad (4.1)$$

$$\mathbf{d}_j = \mathbf{h}_j(\mathbf{u}_{k(j)}, \boldsymbol{\phi}, \boldsymbol{\epsilon}_j). \quad (4.2)$$

Here $\mathbf{u} \in \mathbb{R}^n$ is the state vector, $\mathbf{g} \in \mathbb{R}^n$ is the discrete nonlinear model operator, $\boldsymbol{\phi} \in \mathbb{R}^{n_\phi}$ is the static (time-invariant) parameter vector, $\mathbf{f} \in \mathbb{R}^p$ is a deterministic input, $\mathbf{q}_k^u \in \mathbb{R}^s$ is a Gaussian random vector with the mean $\overline{\mathbf{q}_k^u} \in \mathbb{R}^s$ and covariance matrix $\mathbf{Q}^{uu} \in \mathbb{R}^{s \times s}$ respectively and $\mathbf{d} \in \mathbb{R}^m$, the measurement vector, maps the true state by the nonlinear measurement operator $\mathbf{h} \in \mathbb{R}^m$; $\boldsymbol{\epsilon}_k \in \mathbb{R}^r$ is a Gaussian random vector with the mean $\overline{\boldsymbol{\epsilon}_k} \in \mathbb{R}^r$ and covariance matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{r \times r}$ respectively. Here \mathbf{q}_k^u and $\boldsymbol{\epsilon}_k$ are assumed to be independent. Note that the indexes k and $k(j)$ denotes the time steps for the model integration and the arrival of measurement data (e.g. [10]).

4.2.1 Nonlinear Filtering Approach - Joint State and Parameter Estimation

In the framework of nonlinear filtering, the unknown time-invariant parameter vector $\boldsymbol{\phi}$ is generally treated as a time-varying quantity (e.g. [47, 148]). Additionally, the pa-

¹The coauthors permit the inclusion of this work in the dissertation

parameter vector is artificially perturbed randomly for filter convergence (e.g. [47, 148]). Therefore, the revised state space model is written as

$$\mathbf{u}_{k+1} = \mathbf{g}_k(\mathbf{u}_k, \boldsymbol{\phi}_k, \mathbf{f}_k, \mathbf{q}_k^u), \quad (4.3)$$

$$\boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k + \mathbf{q}_k^\phi, \quad (4.4)$$

$$\mathbf{d}_j = \mathbf{h}_j(\mathbf{u}_{k(j)}, \boldsymbol{\phi}_{k(j)}, \boldsymbol{\epsilon}_j), \quad (4.5)$$

where \mathbf{q}_k^ϕ is the artificial random noise vector which may require fine-tuning for optimal filter performance (e.g. [47, 148]). Next the new state vector is formed by appending the unknown parameter vector to the original state vector leading to the following augmented state space model

$$\mathbf{x}_{k+1} = \mathbf{g}_k(\mathbf{x}_k, \mathbf{f}_k, \mathbf{q}_k), \quad (4.6)$$

$$\mathbf{d}_j = \mathbf{h}_j(\mathbf{x}_{k(j)}, \boldsymbol{\epsilon}_j) \quad (4.7)$$

where

$$\mathbf{x}_k = \begin{Bmatrix} \mathbf{u}_k \\ \boldsymbol{\phi}_k \end{Bmatrix}; \quad \mathbf{q}_k = \begin{Bmatrix} \mathbf{q}_k^u \\ \mathbf{q}_k^\phi \end{Bmatrix}. \quad (4.8)$$

Consequently, any nonlinear filter can be used to tackle the combined state and parameter estimation from which the relevant parameter vector are easily extracted. In this investigation, the EnKF (detailed in Section 2.6) and PF (detailed in Section 2.7) see are used to obtain the joint state and parameter estimates. The Particle Filter (PF) with an EnKF proposal has been specifically adapted for combined state and parameter estimation, as detailed in the next section. For the subsequent numerical experiments, the artificial random noise vector \mathbf{q}_k^ϕ is set to zero, i.e. the parameters are modeled as

time-invariant quantities. As a result, the estimates obtained using joint state and parameter estimation experiments can be directly compared to those obtained using MCMC approach as described next.

4.2.2 Markov Chain Monte Carlo Simulation Approach

In the general Bayesian framework (e.g. [10]), the joint pdf of the state and parameter vector can be written as

$$\begin{aligned}
 p(\mathbf{u}_1, \dots, \mathbf{u}_{k(J)}, \dots, \mathbf{u}_K, \phi | \mathbf{d}_1, \dots, \mathbf{d}_J) &\propto p(\phi) \left[\prod_{k=1}^{k(1)} p(\mathbf{u}_k | \mathbf{u}_{k-1}, \phi) \right] p(d_1 | \mathbf{u}_{k(1)}, \phi) \\
 &\vdots \\
 &\left[\prod_{k=k(J-1)+1}^{k(J)} p(\mathbf{u}_k | \mathbf{u}_{k-1}, \phi) \right] p(d_J | \mathbf{u}_{k(J)}, \phi) \\
 &\left[\prod_{k=k(J)+1}^K p(\mathbf{u}_k | \mathbf{u}_{k-1}, \phi) \right]
 \end{aligned} \tag{4.9}$$

Consequently, the marginal posterior of the parameter vector can be obtained as

$$\begin{aligned}
 p(\phi | \mathbf{d}_1, \dots, \mathbf{d}_J) &\propto p(\phi) \left[\prod_{k=1}^{k(1)-1} \int_{-\infty}^{\infty} p(\mathbf{u}_k | \mathbf{u}_{k-1}, \phi) d\mathbf{u}_k \right] \int_{-\infty}^{\infty} p(\mathbf{u}_{k(1)} | \mathbf{u}_{k(1)-1}, \phi) p(d_1 | \mathbf{u}_{k(1)}, \phi) d\mathbf{u}_{k(1)} \\
 &\vdots \\
 &\left[\prod_{k=k(J-1)+1}^{k(J)-1} \int_{-\infty}^{\infty} p(\mathbf{u}_k | \mathbf{u}_{k-1}, \phi) d\mathbf{u}_k \right] \int_{-\infty}^{\infty} p(\mathbf{u}_{k(J)} | \mathbf{u}_{k(J)-1}, \phi) p(d_J | \mathbf{u}_{k(J)}, \phi) d\mathbf{u}_{k(J)} \\
 &\left[\prod_{k=k(J)+1}^K \int_{-\infty}^{\infty} p(\mathbf{u}_k | \mathbf{u}_{k-1}, \phi) d\mathbf{u}_k \right]
 \end{aligned} \tag{4.10}$$

The above equation can be concisely written as

$$p(\boldsymbol{\phi}|\mathbf{d}_1, \dots, \mathbf{d}_J) \propto p(\boldsymbol{\phi}) \prod_{j=1}^J \int_{-\infty}^{\infty} p(\mathbf{u}_{k(j)}|\mathbf{u}_{k(j)-1}, \boldsymbol{\phi}) p(d_j|\mathbf{u}_{k(j)}, \boldsymbol{\phi}) d\mathbf{u}_{k(j)} \quad (4.11)$$

where the computation of $p(\mathbf{u}_{k(j)}|\mathbf{u}_{k(j)-1}, \boldsymbol{\phi})$ involves a state estimation problem. In this investigation, EnKF is employed to tackle this state estimation problem.

MCMC method [58, 60] can be used to generate samples of the parameter vector from $p(\boldsymbol{\phi}|\mathbf{d}_1, \dots, \mathbf{d}_J)$ (known only up to a proportionality) as detailed in [94, 149, 150]. In this investigation, a new sample $\boldsymbol{\phi}'$ of the parameter vector is generated from the current sample $\boldsymbol{\phi}$ using a random-walk Metropolis-Hastings (MH) algorithm (see Section 3.3) as described in [94, 149, 150]. In this approach, the acceptance probability $\alpha(\boldsymbol{\phi}, \boldsymbol{\phi}')$ is

$$\alpha(\boldsymbol{\phi}, \boldsymbol{\phi}') = \min\left(1, \frac{p(\boldsymbol{\phi}'|\mathbf{d}_1, \dots, \mathbf{d}_J) q(\boldsymbol{\phi}', \boldsymbol{\phi})}{p(\boldsymbol{\phi}|\mathbf{d}_1, \dots, \mathbf{d}_J) q(\boldsymbol{\phi}, \boldsymbol{\phi}')}\right) \quad (4.12)$$

where the proposal density q is assumed to be uniformly distributed in $[-w, w]^{n_\phi}$ [94]. An appropriate proposal width w is critical to the efficiency of MH MCMC and chosen to have 25% acceptance ratio [94].

4.3 Particle filter with EnKF Proposal - Extension to Combined State and Parameter Estimation

In this section, the PF algorithm proposed by Mandel and Beezley [48, 146] (see Section 2.7 for details), originally used for the state estimation, is extended to handle combined state and parameter estimation using a general multivariate kernel density estimator. PF [9, 19, 44, 53] can handle the most general forms of nonlinearities in measurement and

model operators and non-Gaussian model and measurement noise. In contrast to EKF and EnKF, no Gaussian assumption on the state is necessary. Therefore, PF is superior to EnKF and EKF for strongly non-Gaussian models [32, 33].

For strongly nonlinear systems, the PF proposal based on EnKF is perhaps most appropriate as it maintains some non-Gaussian characteristics in the proposal. Unlike EKF and UKF based proposal, EnKF proposal does not differ from particle to particle. In other words, a single proposal is used to update the particle weights as explained next. The (non-Gaussian) EnKF based proposal $q(\mathbf{x}_{k(j),i}|\mathbf{x}_{k(j)-1,i}, \mathbf{D}_j)$ is obtained by the KDE [114]. The forecast pdf $p(\mathbf{x}_{k(j),i}|\mathbf{x}_{k(j)-1,i})$ of the particle filter is also estimated by KDE. This proposal is first proposed by Mandel and Beezley and termed as the predictor-corrector filter [48, 146], in which the following weight update equation was proposed

$$\begin{aligned} \tilde{w}_{k(j),i} &= w_{k(j)-1,i} \frac{p(\mathbf{d}_j|\mathbf{x}_{k(j),i}) p(\mathbf{x}_{k(j),i}|\mathbf{x}_{k(j)-1,i})}{q(\mathbf{x}_{k(j),i}|\mathbf{x}_{k(j)-1,i}, \mathbf{D}_j)} \\ &= w_{k(j)-1,i} p(\mathbf{d}_j|\mathbf{x}_{k(j),i}) \frac{\sum_{l=1}^N w_{k(j)-1,l} \mathbf{1}_{\|\mathbf{x}_{k(j),l}^f - \mathbf{x}_{k(j),i}^a\|_H \leq h(\mathbf{x}_{k(j),i}^a)}}{\sum_{l=1}^N \frac{1}{N} \mathbf{1}_{\|\mathbf{x}_{k(j),l}^a - \mathbf{x}_{k(j),i}^a\|_H \leq h(\mathbf{x}_{k(j),i}^a)}}, \quad (4.13) \end{aligned}$$

with $\mathbf{1}_{(\cdot)}$ being the indicator function and $h(\mathbf{x}_{k(j),i}^a)$ the bandwidth, chosen to be the distance to the \sqrt{N} -th nearest sample (data point) in the $\|\cdot\|_H$ norm. Mandel and Beezley were concerned with the state (and not parameter) estimation problem in which the vector \mathbf{x} consists of a discrete representation of smooth functions, for which the norm associated with the so-called Cameron-Martin space is suitable [151, 152].

In this section, this methodology is extended to tackle the problem of joint state and parameter estimation problem. The augmented state vector can no longer be considered a discrete representation of smooth functions, and thus a different approach to KDE

must be taken. The general multivariate kernel estimator [153] will be used resulting in a weight update equation given by

$$\begin{aligned}\tilde{w}_{k(j),i} &= w_{k(j)-1,i} \frac{p(\mathbf{d}_j | \mathbf{x}_{k(j),i}) p(\mathbf{x}_{k(j),i} | \mathbf{x}_{k(j)-1,i})}{q(\mathbf{x}_{k(j),i} | \mathbf{x}_{k(j)-1,i}, \mathbf{D}_j)} \\ &= w_{k(j)-1,i} p(\mathbf{d}_j | \mathbf{x}_{k(j),i}) \frac{\sum_{l=1}^N w_{k(j)-1,l} \frac{1}{|H_f|} K\left(H_f^{-1} \left(\mathbf{x}_{k(j),l}^f - \mathbf{x}_{k(j),i}^a\right)\right)}{\sum_{l=1}^N \frac{1}{N} \frac{1}{|H_a|} K\left(H_a^{-1} \left(\mathbf{x}_{k(j),l}^a - \mathbf{x}_{k(j),i}^a\right)\right)},\end{aligned}\quad (4.14)$$

with a Gaussian kernel $K(\mathbf{y}) = (2\pi)^{-n/2} \exp(-\frac{1}{2}\mathbf{y}^T \mathbf{y})$ and a bandwidth matrix H proportional to the square root of the ensemble covariance matrix, as suggested in [154]. The constant of proportionality is obtained by generalizing Scott's rule for the univariate case (see [114]) to the multivariate one, in which case one obtains

$$\begin{aligned}H_f &= N_{eff}^{-\frac{1}{n+4}} \Sigma_f^{1/2} \\ &= \left(\frac{1}{\sum_{i=1}^N (w_{k(j),i})^2} \right)^{-\frac{1}{n+4}} \times \\ &\quad \left[\left(\frac{1}{1 - \sum_{i=1}^N w_{k(j),i}^f} \right) \sum_{l=1}^N w_{k(j),l}^f \left(\mathbf{x}_{k(j),l}^f - \bar{\mathbf{x}}_{k(j)}^f \right) \left(\mathbf{x}_{k(j),l}^f - \bar{\mathbf{x}}_{k(j)}^f \right)^T \right]^{1/2}\end{aligned}\quad (4.15)$$

$$\begin{aligned}H_a &= N^{-\frac{1}{n+4}} \Sigma_a^{1/2} \\ &= N^{-\frac{1}{n+4}} \left[\left(\frac{1}{N-1} \right) \sum_{l=1}^N \left(\mathbf{x}_{k(j),l}^a - \bar{\mathbf{x}}_{k(j)}^a \right) \left(\mathbf{x}_{k(j),l}^a - \bar{\mathbf{x}}_{k(j)}^a \right)^T \right]^{1/2}\end{aligned}\quad (4.16)$$

The algorithm of the particle filter with EnKF proposal is summarized next:

1. Using the prior pdf of \mathbf{x}_0 , generate an ensemble $\{\mathbf{x}_{0,i}^f\}$ having size N with $i = 1, \dots, N$.

2. Analysis step:

$$\mathbf{d}_{j,i} = \mathbf{h}_j \left(\mathbf{x}_{k(j),i}^f, \epsilon_{j,i} \right), \quad (4.17)$$

$$\bar{\mathbf{d}}_j = \sum_{i=1}^N w_{k(j)-1,i} \mathbf{d}_{j,i}, \quad (4.18)$$

$$\bar{\mathbf{x}}_{k(j)}^f = \sum_{i=1}^N w_{k(j)-1,i} \mathbf{x}_{k(j),i}^f, \quad (4.19)$$

$$\mathbf{P}_{xd} = \frac{1}{1 - \sum_{i=1}^N w_{k(j)-1,i}^2} \sum_{i=1}^N w_{k(j)-1,i} \left(\mathbf{x}_{k(j),i}^f - \bar{\mathbf{x}}_{k(j)}^f \right) \left(\mathbf{d}_{j,i} - \bar{\mathbf{d}}_j \right)^T, \quad (4.20)$$

$$\mathbf{P}_{dd} = \frac{1}{1 - \sum_{i=1}^N w_{k(j)-1,i}^2} \sum_{i=1}^N w_{k(j)-1,i} \left(\mathbf{d}_{j,i} - \bar{\mathbf{d}}_j \right) \left(\mathbf{d}_{j,i} - \bar{\mathbf{d}}_j \right)^T, \quad (4.21)$$

$$\mathbf{K}_{k(j)} = \mathbf{P}_{xd} \mathbf{P}_{dd}^{-1}, \quad (4.22)$$

$$\mathbf{x}_{k(j),i}^a = \mathbf{x}_{k(j),i}^f + \mathbf{K}_{k(j)} \left(\bar{\mathbf{d}}_j - \mathbf{h}_k \left(\mathbf{x}_{k(j),i}^f, \epsilon_{j,i} \right) \right) \quad (4.23)$$

$$H_a = N^{-\frac{1}{n+4}} \left[\left(\frac{1}{N-1} \right) \sum_{l=1}^N \left(\mathbf{x}_{k(j),l}^a - \bar{\mathbf{x}}_{k(j)}^a \right) \left(\mathbf{x}_{k(j),l}^a - \bar{\mathbf{x}}_{k(j)}^a \right)^T \right]^{1/2} \quad (4.24)$$

$$H_f = \left(\frac{1}{\sum_{i=1}^N (w_{k(j)-1,i})^2} \right)^{-\frac{1}{n+4}} \times \left[\left(\frac{1}{1 - \sum_{i=1}^N w_{k(j)-1,i}^2} \right) \sum_{l=1}^N w_{k(j)-1,l} \left(\mathbf{x}_{k(j),l}^f - \bar{\mathbf{x}}_{k(j)}^f \right) \left(\mathbf{x}_{k(j),l}^f - \bar{\mathbf{x}}_{k(j)}^f \right)^T \right]^{1/2} \quad (4.25)$$

$$\tilde{w}_{k(j),i} = w_{k(j)-1,i} p \left(\bar{\mathbf{d}}_j | \mathbf{x}_{k(j),i} \right) \frac{\sum_{l=1}^N w_{k(j)-1,l} \frac{1}{|H_f|} K \left(H_f^{-1} \left(\mathbf{x}_{k(j),l}^f - \mathbf{x}_{k(j),i}^a \right) \right)}{\sum_{l=1}^N \frac{1}{N} \frac{1}{|H_a|} K \left(H_a^{-1} \left(\mathbf{x}_{k(j),l}^a - \mathbf{x}_{k(j),i}^a \right) \right)} \quad (4.26)$$

$$w_{k(j),i} = \frac{\tilde{w}_{k(j),i}}{\sum_{i=1}^N \tilde{w}_{k(j),i}}. \quad (4.27)$$

3. Forecast step:

$$\mathbf{x}_{k+1,i}^f = \mathbf{g}_k \left(\mathbf{x}_{k,i}^a, \mathbf{f}_k, \mathbf{q}_{k,i} \right). \quad (4.28)$$

4.4 Noisy oscillation of a Duffing system

In this section, several numerical experiments are performed for joint state and parameter estimation of a Duffing oscillator [115, 116] excited by combined harmonic and random input. For the parameter set chosen in the numerical experiments, the system exhibits a period-two subharmonic oscillation in the absence of random input [116]. EnKF, PF and MCMC sampling technique are then used to estimate the stiffness parameters in the presence of both model and measurement noise as described in details next.

4.4.1 Duffing Oscillator Model

This section illustrates the usefulness of the aforementioned theoretical formulations using a Duffing oscillator model. The Duffing system forced by combined harmonic and random inputs is described by the following nonlinear differential equation (e.g. [115, 116])

$$m\ddot{u}(t) + c\dot{u}(t) + k_1u(t) + k_3u^3(t) = \mathcal{T}\cos(\omega t) + \sigma\xi(t) \quad (4.29)$$

with $m[kg]$ being the mass, $c[N \cdot s \cdot m^{-1}]$ is the damping coefficient, $k_1[N \cdot m^{-1}]$ and $k_3[N \cdot m^{-3}]$ are the linear and cubic stiffness coefficients, $u(t) [m]$ is the displacement, $\mathcal{T}[N]$ and $\omega[rad \cdot s^{-1}]$ denote the amplitude and frequency of the harmonic excitation, $\xi(t)$ represents a Gaussian white noise random input (additive modeling error) and $\sigma[N]$ represents its strength.

The state-space model of Eq. (4.29) becomes

$$\dot{x}_1 = x_2 \quad (4.30)$$

$$\dot{x}_2 = -\frac{1}{m} [cx_2 + k_1x_1 + k_3x_1^3] + \frac{1}{m}\mathcal{T} \cos(\omega t) + \frac{1}{m}\sigma\xi(t) , \quad (4.31)$$

where $x_1 = u$ and $x_2 = \dot{u}$.

Of primary concern is the estimation of the stiffness coefficients k_1 and k_3 and the damping coefficient c . For joint state and parameter estimation using EnKF and PF, the original state vector $\mathbf{x} = \{x_1, x_2\}^T$ is augmented with the coefficients c , k_1 , and k_3 as three new state variables $x_3 = c$, $x_4 = k_1$ and $x_5 = k_3$ which satisfy the following equations

$$\dot{x}_3 = 0 , \quad (4.32)$$

$$\dot{x}_4 = 0 , \quad (4.33)$$

$$\dot{x}_5 = 0 , \quad (4.34)$$

i.e. they are modeled as time-invariant quantities. In a previous investigation [47], the unknown system parameters were modeled as Wiener processes resulting in an artificial inflation the variance of the estimates in order to avoid filter divergence (e.g. [155]). The need for this artificial perturbation was to provide a range of possible parameter values and select the best value(s) based on the partial observations of the system state. The use of a good proposal obtained using an EnKF analysis step circumvents the need to artificially inflate the variances of the unknown parameters for filter convergence and is thus avoided in this investigation.

The above set of equations have the following Ito SDEs representation

$$dx_1 = x_2 dt \quad (4.35)$$

$$dx_2 = -\frac{1}{m} [x_3 x_2 + x_4 x_1 + x_5 x_1^3 - \mathcal{T} \cos(\omega t)] dt + \frac{1}{m} \sigma \xi(t) dt. \quad (4.36)$$

$$dx_3 = 0 \quad (4.37)$$

$$dx_4 = 0 \quad (4.38)$$

$$dx_5 = 0. \quad (4.39)$$

where $\xi(t) dt = dW = W(t_{k+1}) - W(t_k)$ denotes a Brownian path increment.

The discretization of the above Ito SDEs using the Euler-Maruyama scheme [118–120] with time step Δt leads to

$$\{x_1\}_{k+1} = \{x_1\}_k + \Delta t \{x_2\}_k \quad (4.40)$$

$$\begin{aligned} \{x_2\}_{k+1} = \{x_2\}_k - \frac{1}{m} \Delta t [\{x_3\}_k \{x_2\}_k + \{x_4\}_k \{x_1\}_k + \{x_5\}_k \{x_1\}_k^3 - \mathcal{T} \cos(\omega t_k)] \\ + \frac{1}{m} \sigma \sqrt{\Delta t} \varepsilon_k \end{aligned} \quad (4.41)$$

$$\{x_3\}_{k+1} = \{x_3\}_k \quad (4.42)$$

$$\{x_4\}_{k+1} = \{x_4\}_k \quad (4.43)$$

$$\{x_5\}_{k+1} = \{x_5\}_k. \quad (4.44)$$

where ε_k denotes independent and identically distributed unit standard Gaussian random variables.

4.4.2 Noisy oscillation of the Duffing oscillator

The following numerical values are chosen [115, 116]: $m = 1\text{kg}$, $c = 0.3\text{N} \cdot \text{s} \cdot \text{m}^{-1}$, $k_1 = -1\text{N} \cdot \text{m}^{-1}$, $k_3 = 1\text{N} \cdot \text{m}^{-3}$, $\omega = 1.25\text{rad} \cdot \text{s}^{-1}$ and $\Delta t = 5 \times 10^{-3}\text{s}$. This choice of parameters produces limit cycle oscillation. The unforced system has three fixed points: an unstable fixed point at $u = 0\text{m}$ and two stable fixed points at $\pm 1\text{m}$. Fig. 4.1 shows two period-two subharmonic oscillation with period $4\pi/\omega$ under purely deterministic loading ($\sigma = 0\text{N}$) for $\mathcal{T} = 0.3\text{N}$. Each steady-state trajectory presented in the figure arises from two different initial conditions.

For modeling noise with $\sigma = 0.015\text{N}$, a sample of the steady-state noisy oscillation is shown in Fig. 4.2. Note that the quasi-periodic behaviour of the phase-space curve arises due to the modeling noise.

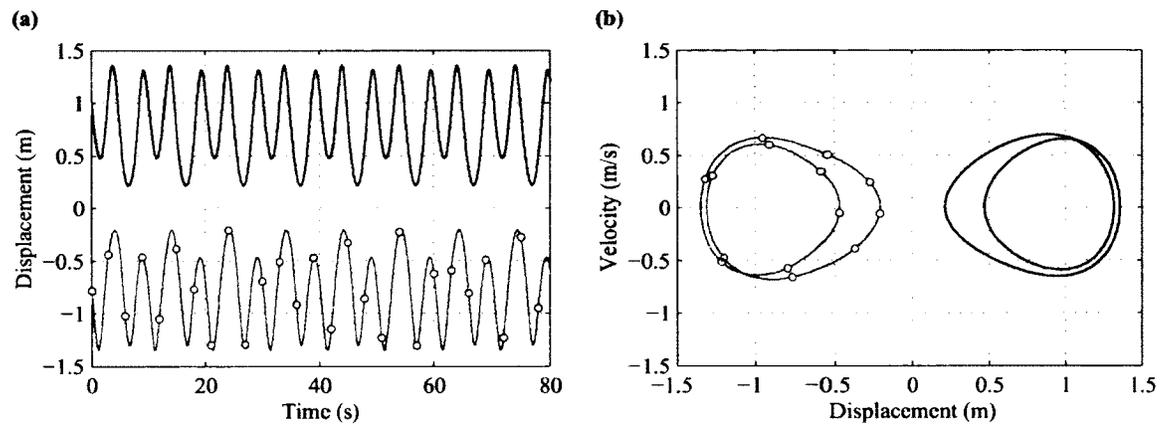


Figure 4.1: Response of the Duffing oscillator under purely deterministic input with $\mathcal{T} = 0.3\text{N}$, $\sigma = 0\text{N}$: (a) two steady-state trajectories, (b) the associated phase-space diagrams.

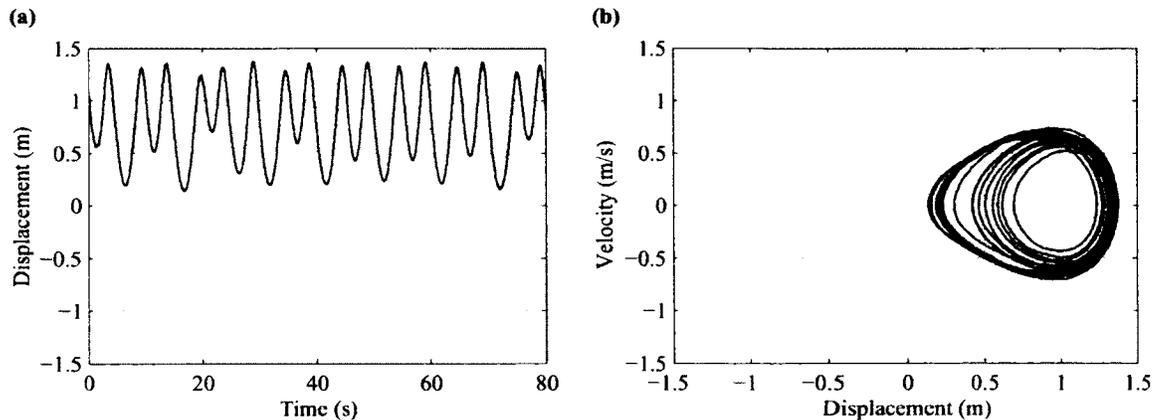


Figure 4.2: Response of the Duffing oscillator under combined deterministic and random input with $\mathcal{T} = 0.3N$, $\sigma_1 = 0.015N$: (a) steady-state trajectory, (b) the associated phase-space diagram.

4.4.3 Parameter Estimation

In this section, the combined state and parameter estimation problem of the Duffing system using EnKF and PF is addressed. MCMC sampling technique is also used to estimate the parameters with EnKF providing the state estimates. In particular, the estimation of the damping parameter c and stiffness parameters k_1 and k_3 is of interest. For simplicity, although the strength of model noise σ is assumed to be known in this investigation, it can also be estimated along with the system parameters as described in [94].

The observational data is modeled as

$$d_k = u_k + \epsilon_k . \quad (4.45)$$

For this experiment, the measurement error is assumed to be Gaussian: $\epsilon_k \sim \mathcal{N}(0, 1.3 \times 10^{-2})$. The measurement arrives at an interval of 0.2 time unit. The measurement error standard deviation is assumed to be 15% of the root-mean-square (RMS) value of

the true displacement of the oscillator. The modeling error amplitude is chosen to be $\sigma = 0.015N$. Fig. 4.3a shows the true displacement and the measurement d_k . For this parameter estimation experiment, the initial conditions are $u \sim \mathcal{N}(1, 0)$, $\dot{u} \sim \mathcal{N}(0, 0)$, $c \sim \mathcal{N}(0.39, 0.002025)$, $k_1 \sim \mathcal{N}(-1.3, 0.0225)$, $k_3 \sim \mathcal{N}(1.3, 0.0225)$. An ensemble size of $N = 25000$ is used for EnKF and PF. Furthermore, 450,000 MCMC samples from the joint posterior parameter pdf are obtained with 5000 EnKF samples used to represent the state vector for each MCMC parameter vector realization. The threshold effective ensemble size for the resampling in PF is taken to be 50% of the ensemble size (i.e. $N_{thr} = 0.5N$). From numerical investigations, it turns out that this threshold value yields the best performance for PF. To minimize sampling errors, an efficient sampling scheme based on LHS [121, 122] is adopted for EnKF and PF.

For the joint state and parameter estimation experiments, Figs. 4.3-4.5 show the estimates of the damping and stiffness coefficients c , k_1 , and k_3 using EnKF in subplot (b) and PF in subplot (c). The marginal pdfs of the parameters at the final time step of simulation are provided in Fig. 4.6. The marginal pdfs of the parameters using MCMC sampling are also provided in Fig. 4.6. All three methods yield parameter estimates of similar accuracy.

Effect of measurement noise

In this experiment, the effect of measurement noise intensity on the filter estimates is examined. The measurement noise is now taken to be Gaussian given by $\epsilon_k \sim \mathcal{N}(0, 7.1 \times 10^{-2})$. In contrast to the previous experiment, the standard deviation of the measurement error is now assumed to be 35% of the RMS value of the true displacement. Similar to the previous experiment, the time interval between the observational data is taken to be 0.2 time unit. Fig. 4.7a shows the true displacement of the system

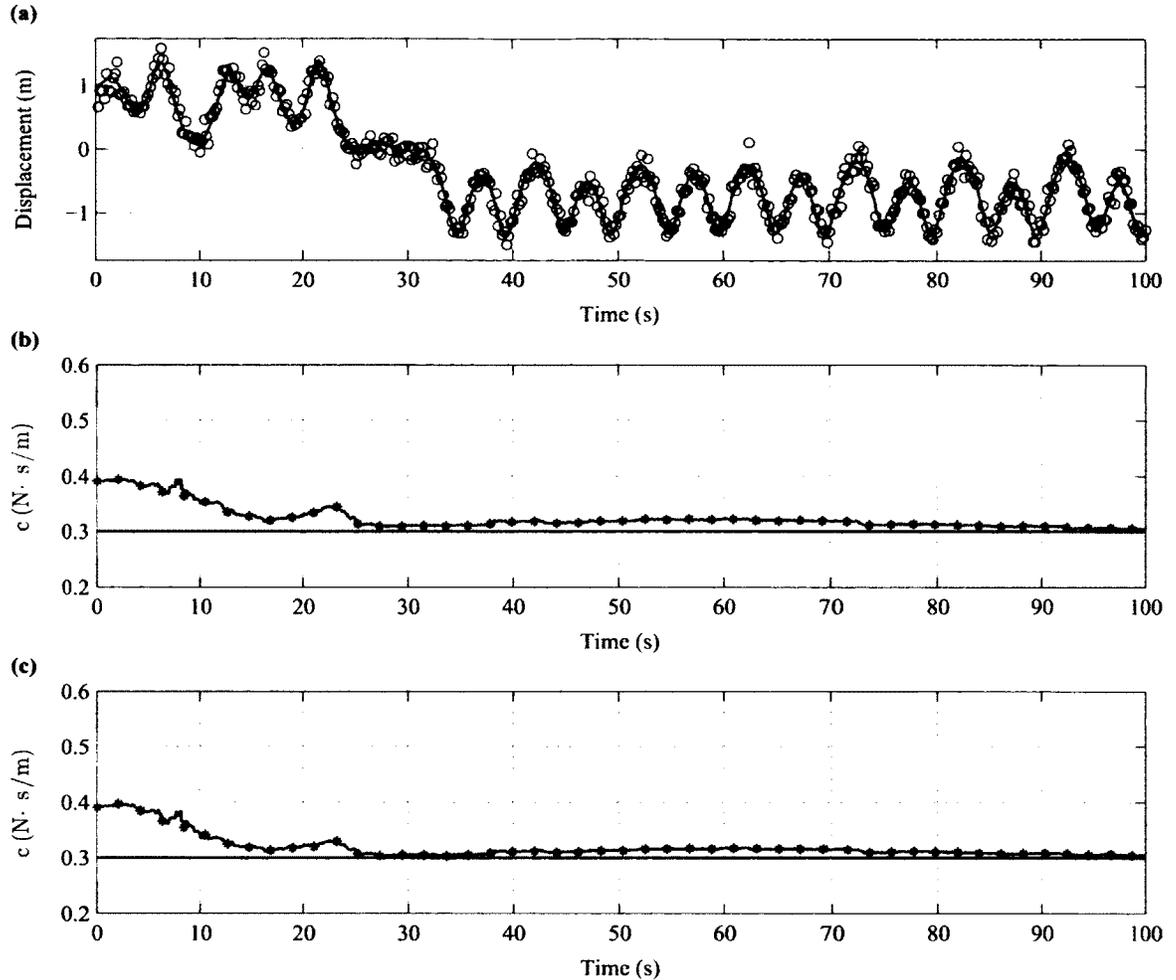


Figure 4.3: Duffing oscillator c parameter estimates: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

and the associated synthetic observational data d_k .

For this joint state and parameter estimation experiment, the same initial conditions, ensemble size (for PF and EnKF) and number of MCMC samples are used as in the previous experiment. Figs. 4.7-4.9 show the estimates of the damping and stiffness coefficients and Fig. 4.10 provides the posterior marginal pdfs once all the data has been assimilated. Evidently, the parameter estimates obtained using PF and MCMC are equivalent in accuracy. EnKF was only able to estimate the nonlinear stiffness co-

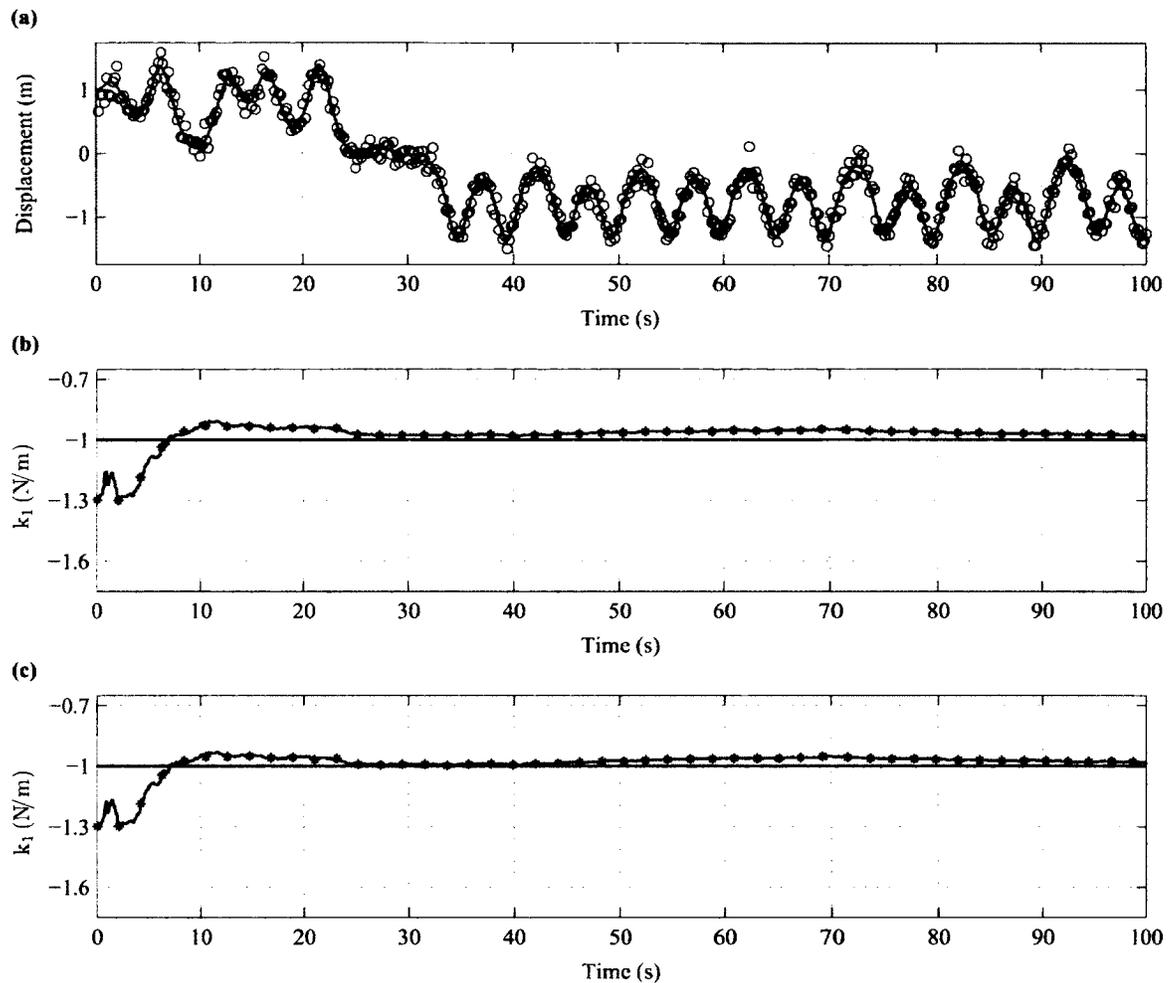


Figure 4.4: Duffing oscillator k_1 parameter estimates: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

efficient k_3 with reasonable accuracy. The superior performance of PF over EnKF is attributed to its ability to handle strongly non-Gaussian systems. Although MCMC sampling techniques relies on EnKF for state estimation, it still provides accurate estimates. In contrast to the extended state vector in joint state and parameter estimation, the state vector in pure state estimation carries less non-Gaussian characteristics. Thus EnKF is able to provide accurate state estimates for MCMC sampling, but encounters difficulties in joint state and parameter estimation. Of course, the error in the estimates

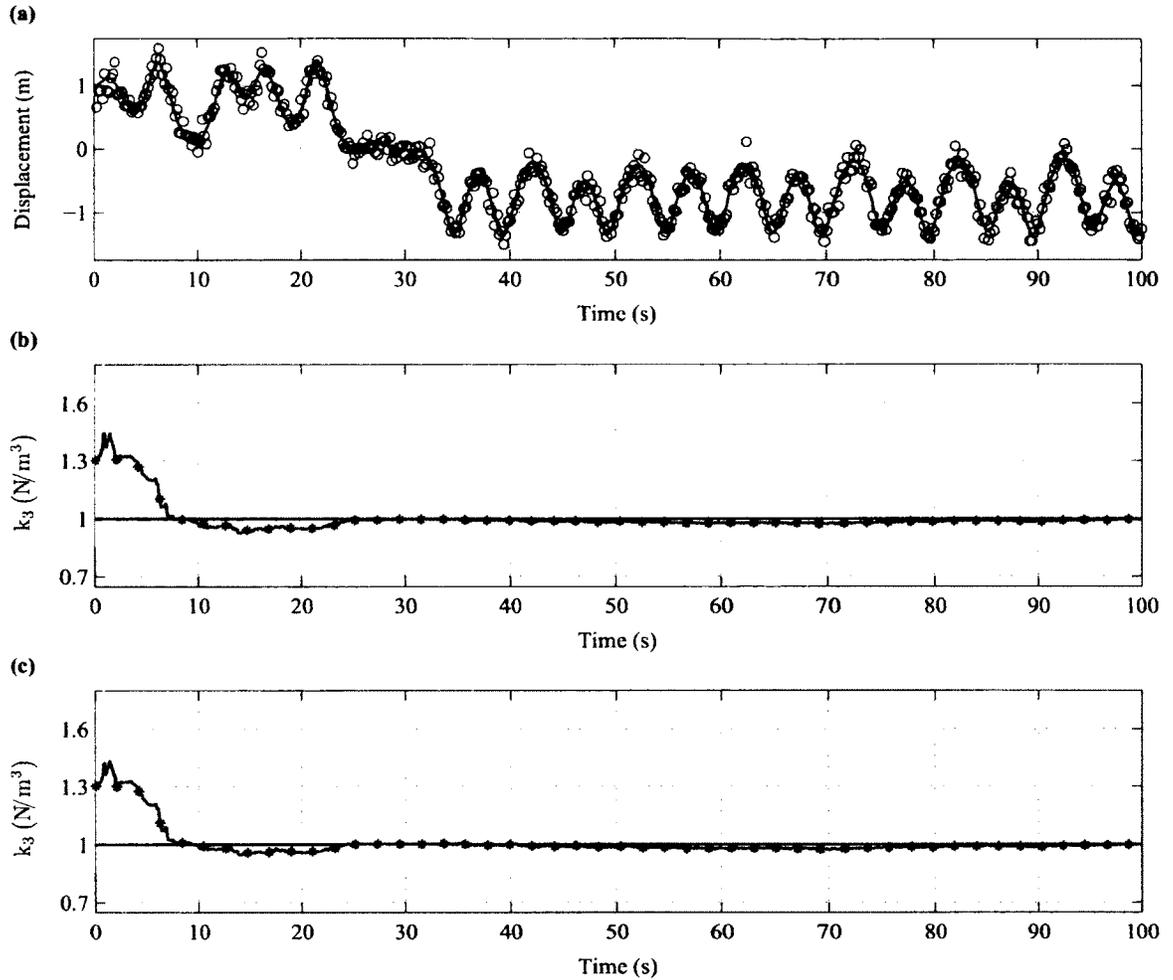


Figure 4.5: Duffing oscillator k_3 parameter estimates: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

is much larger for all three methods compared to the previous experiment due to the presence of larger measurement error.

Effect of observational data sparsity

In this last experiment, the effect of observational data sparsity is examined. The measurement noise is taken to be Gaussian given by $\epsilon_k \sim \mathcal{N}(0, 1.3 \times 10^{-2})$. The measurement time interval is taken to be 1.0 time unit, instead of 0.2 time unit used in

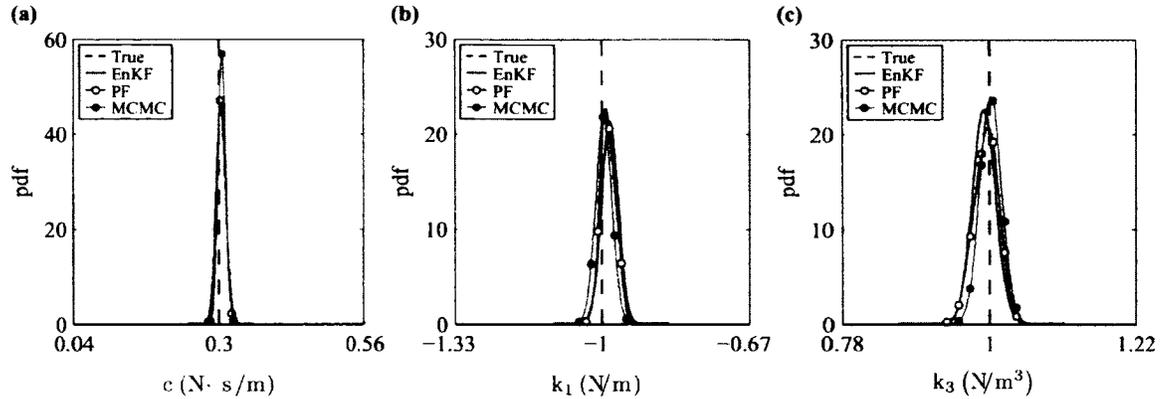


Figure 4.6: Posterior marginal pdfs of parameters for Duffing oscillator: (a) damping coefficient c , (b) linear stiffness coefficient k_1 , (c) nonlinear stiffness coefficient k_3 . Dashed vertical line indicates the true parameter value. Solid curve represents the EnKF estimate. Solid curve with hollow circular markers represents the PF estimate. Solid curve with solid circular markers represents the MCMC estimate.

the first experiment. Fig. 4.11a shows the true displacement of the oscillator and the observational data d_k . Apart from the smaller sampling rate for the observations, all experimental parameters are unchanged from the first experiment (subsection 4.4.3).

The estimates of the damping and stiffness coefficients are plotted in Figs. 4.11-4.13 and the marginal posterior pdfs in Fig. 4.14. Clearly, the performance of PF and MCMC is superior to EnKF which provided an accurate parameter estimate for the non-linear stiffness parameter only. The infrequent assimilation of data allows the augmented state vector to regain its non-Gaussian features due to the nonlinear dynamics, leading to a severe degradation in the performance of EnKF for combined state and parameter estimation.

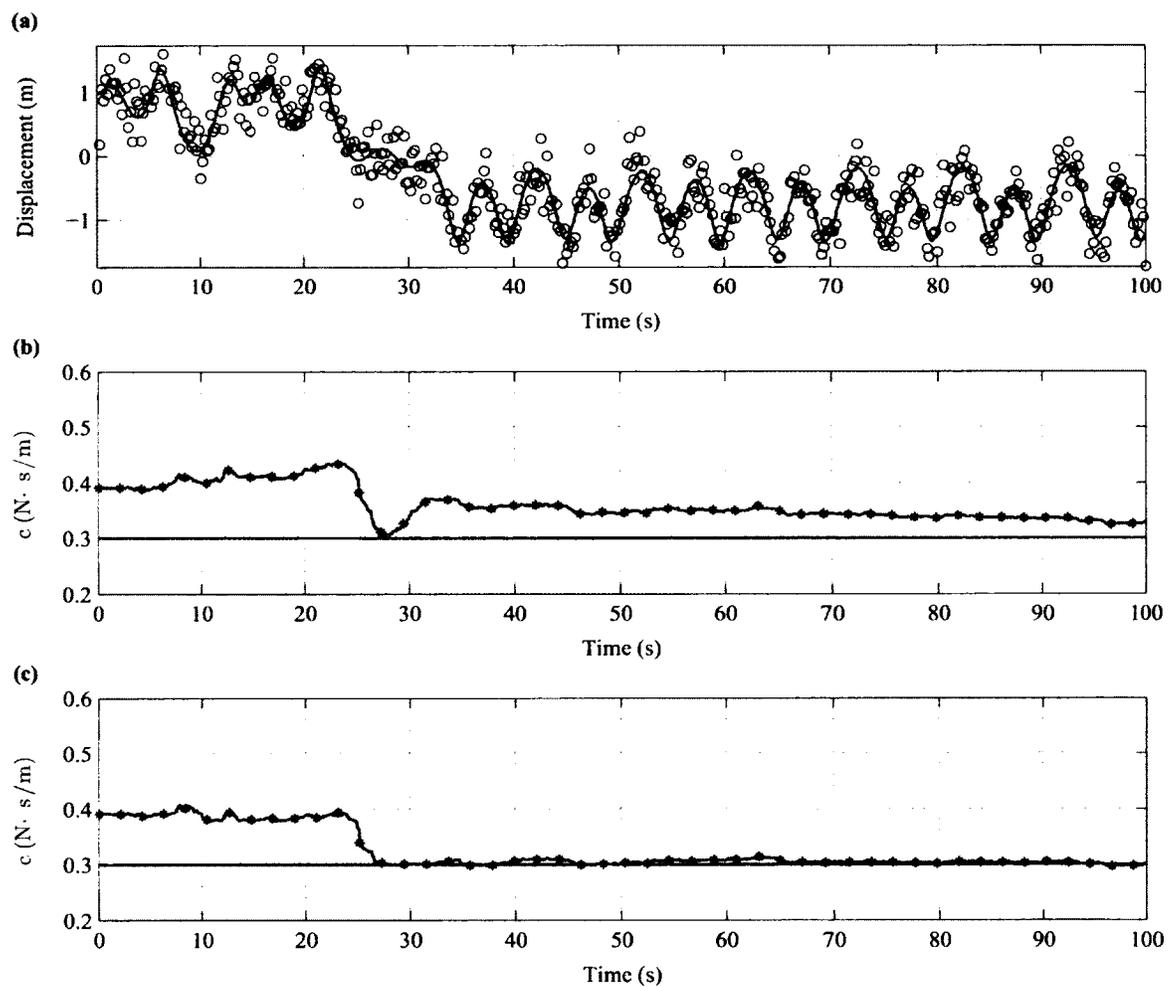


Figure 4.7: Duffing oscillator c parameter estimates with stronger measurement noise: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

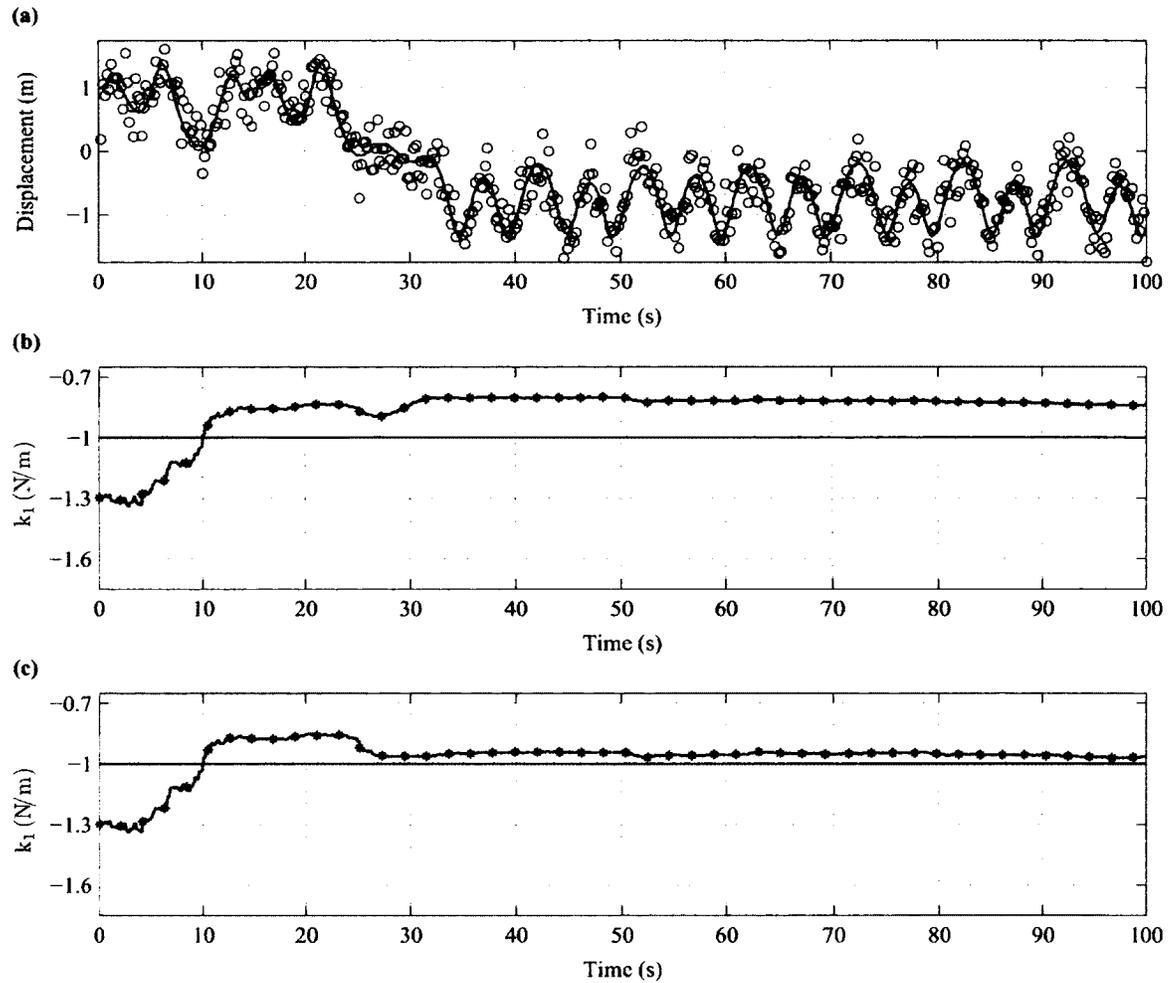


Figure 4.8: Duffing oscillator k_1 parameter estimates with stronger measurement noise: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

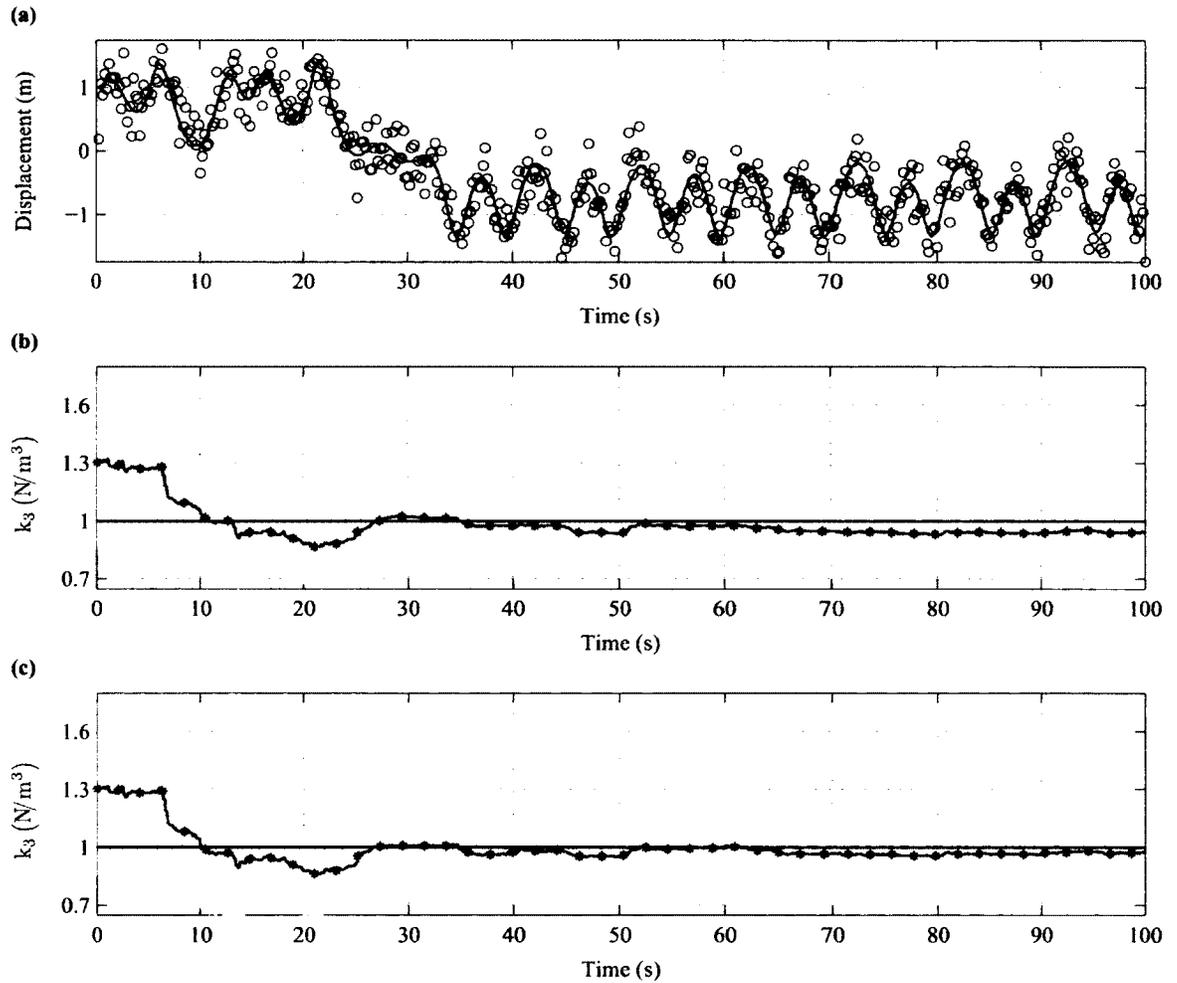


Figure 4.9: Duffing oscillator k_3 parameter estimates with stronger measurement noise: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

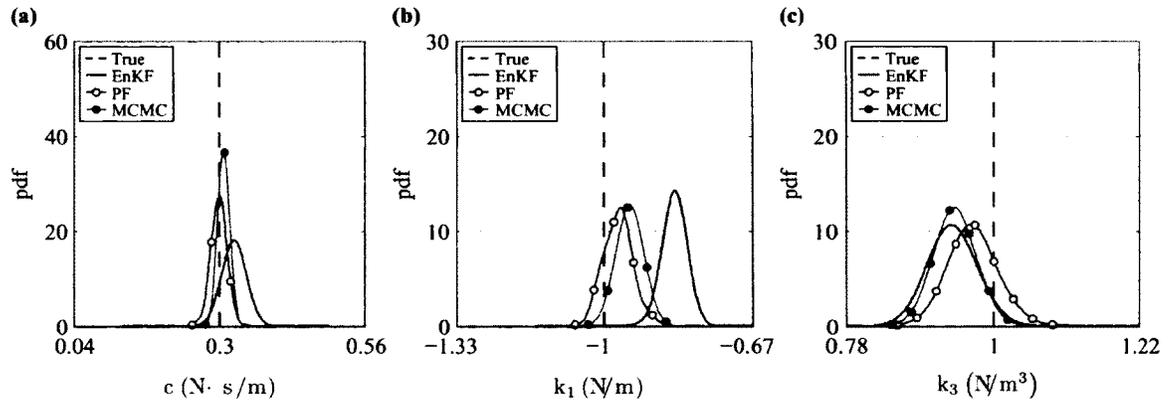


Figure 4.10: Posterior marginal pdfs of parameters for Duffing oscillator with stronger measurement noise: (a) damping coefficient c , (b) linear stiffness coefficient k_1 , (c) nonlinear stiffness coefficient k_3 . Dashed vertical line indicates the true parameter value. Solid curve represents the EnKF estimate. Solid curve with hollow circular markers represents the PF estimate. Solid curve with solid circular markers represents the MCMC estimate.

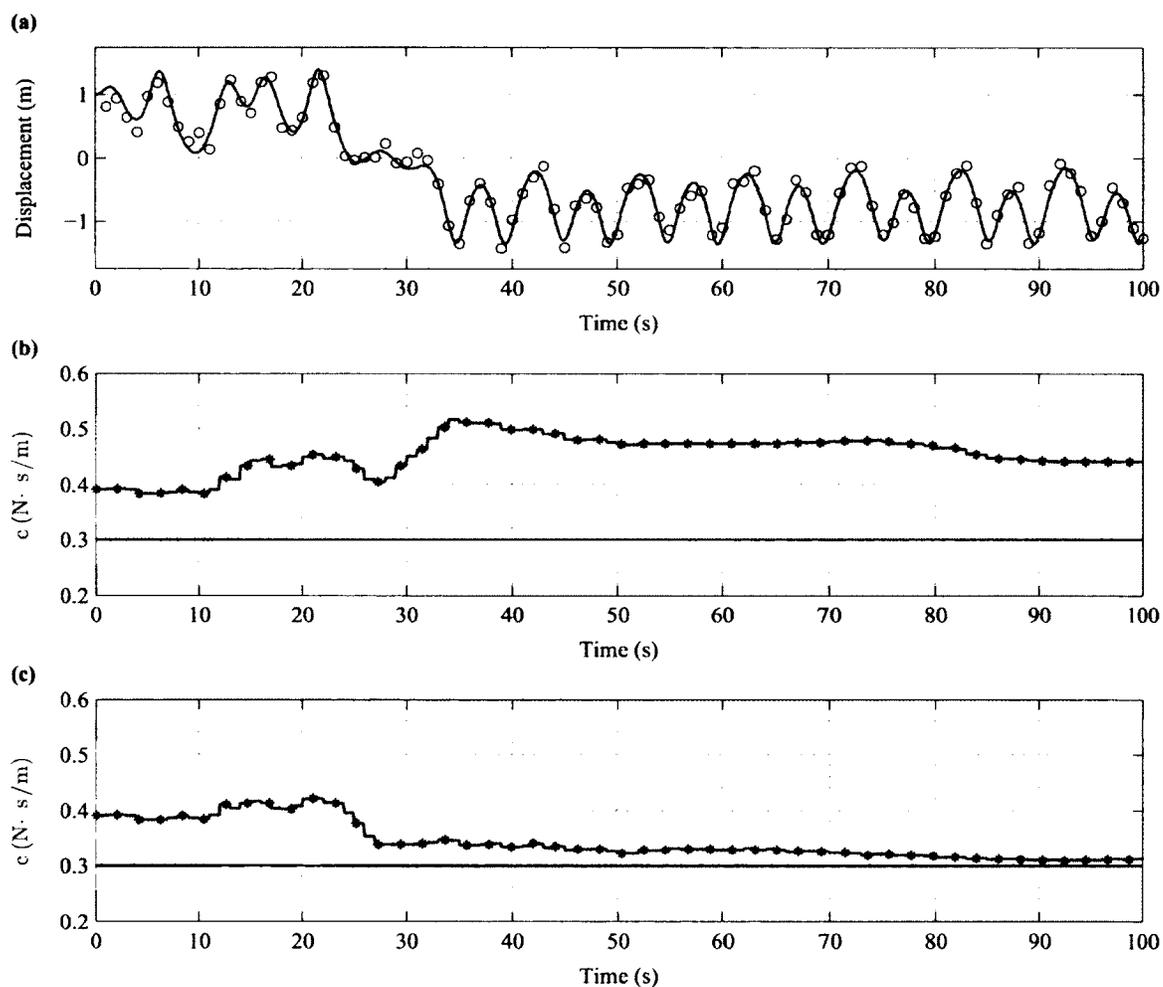


Figure 4.11: Duffing oscillator c parameter estimates with sparse observational data: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

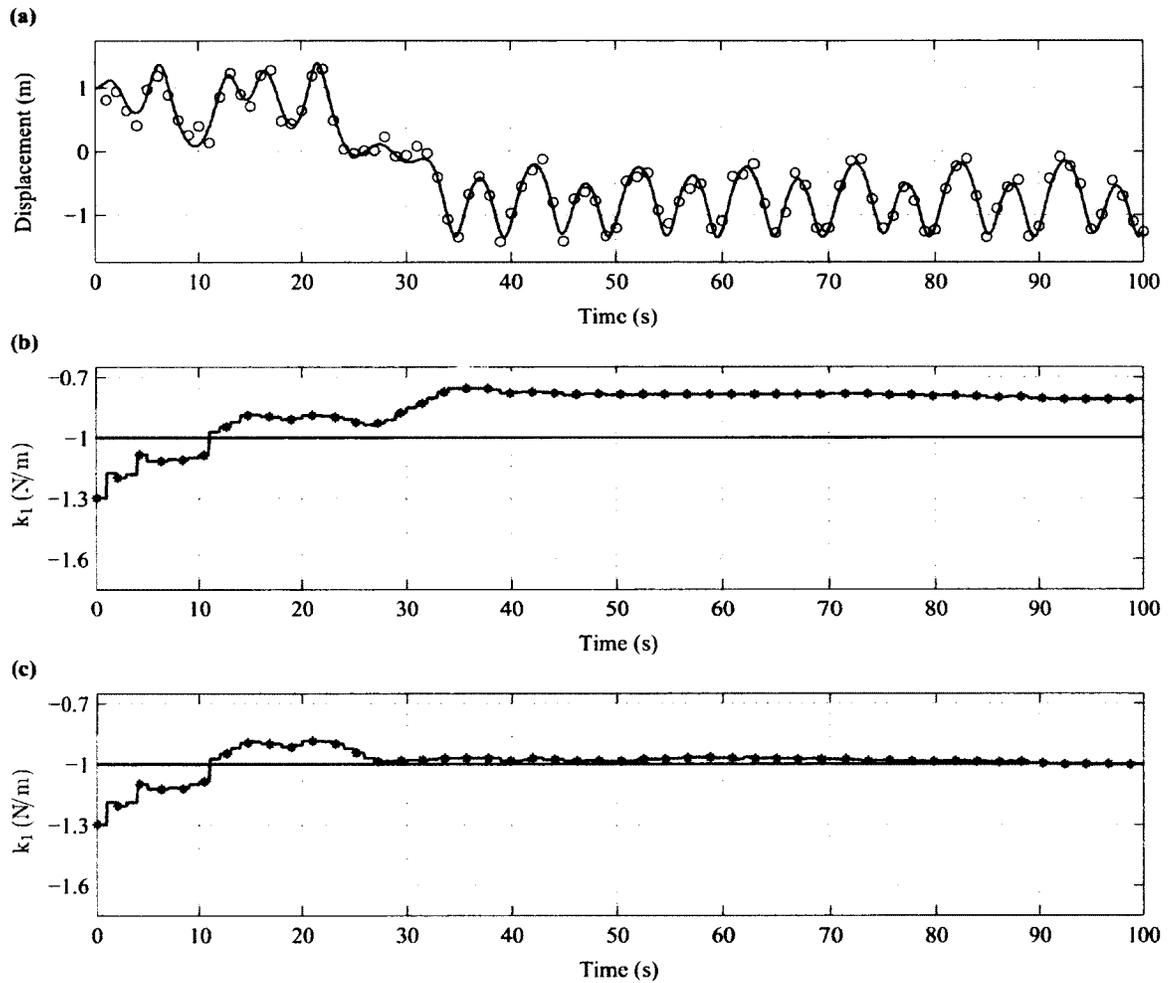


Figure 4.12: Duffing oscillator k_1 parameter estimates with sparse observational data: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

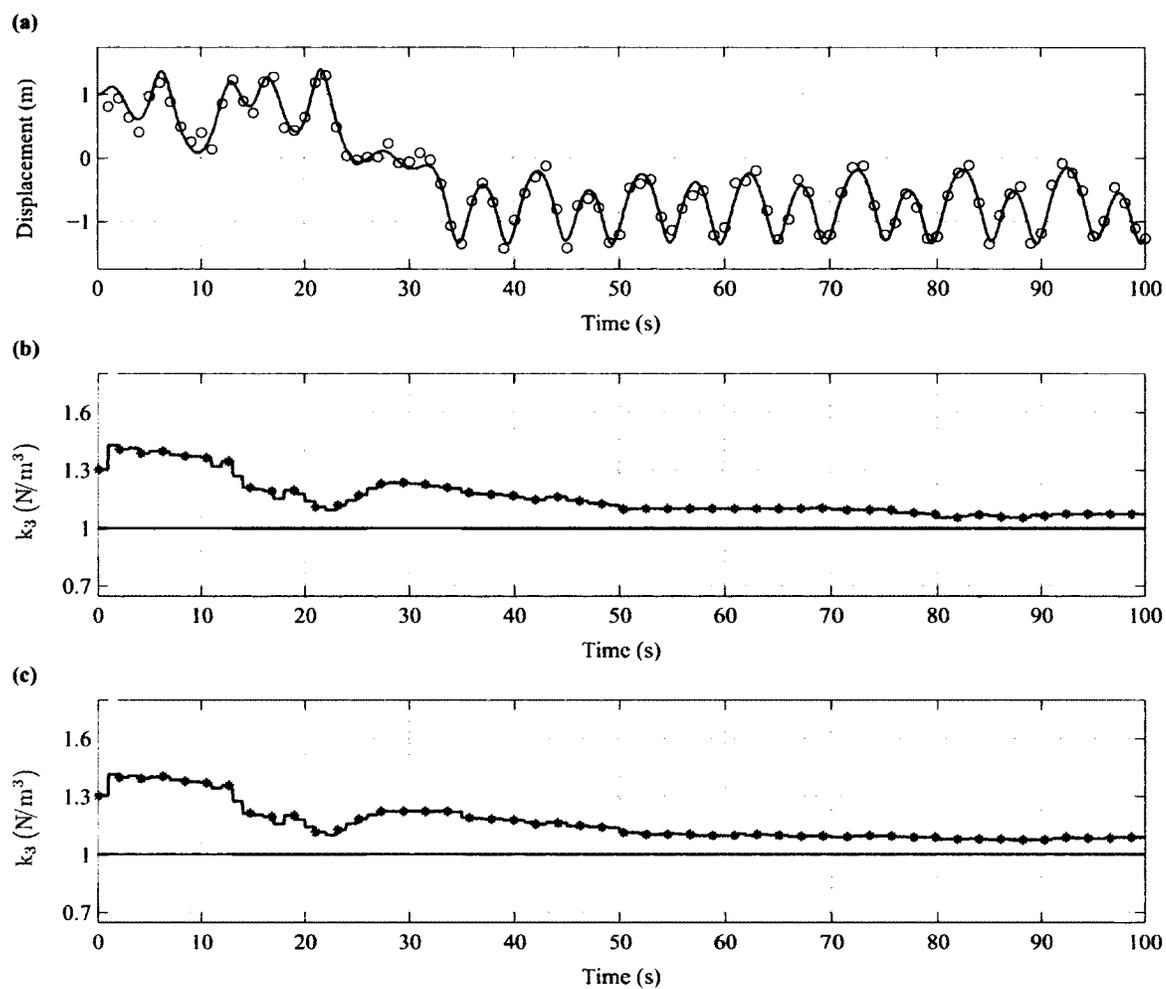


Figure 4.13: Duffing oscillator k_3 parameter estimates with sparse observational data: (a) true (solid line) and measured (circles) displacement, (b) EnKF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area), (c) PF estimate (line with stars) and estimated mean ± 3 error standard deviations (shaded area).

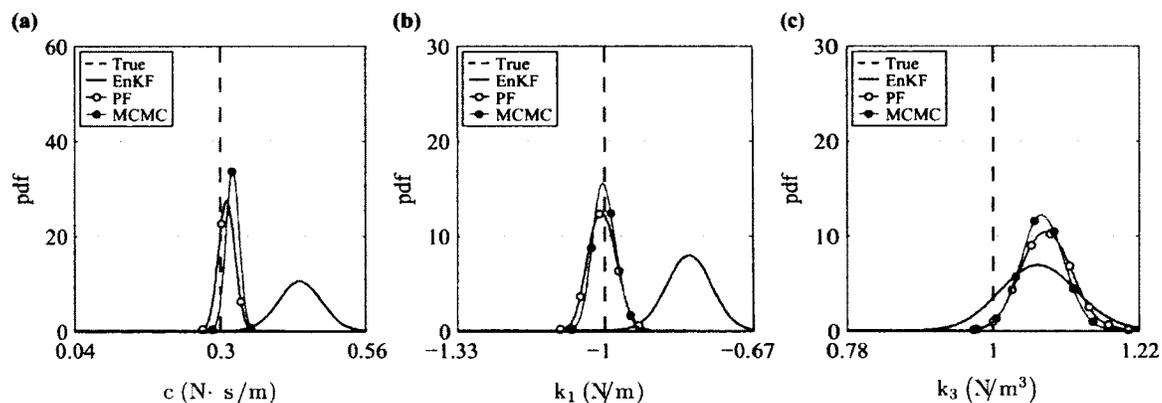


Figure 4.14: Posterior marginal pdfs of parameters for Duffing oscillator with sparse observational data: (a) damping coefficient c , (b) linear stiffness coefficient k_1 , (c) nonlinear stiffness coefficient k_3 . Dashed vertical line indicates the true parameter value. Solid curve represents the EnKF estimate. Solid curve with hollow circular markers represents the PF estimate. Solid curve with solid circular markers represents the MCMC estimate.

Chapter 5

Static Parameter Estimation of a Non-linear Aeroelastic System using Wind Tunnel Data

5.1 Introduction

In this chapter, the Bayesian parameter estimation problem of a self-sustaining aeroelastic oscillator is delineated. The system consists of an elastically mounted rigid wing on a rig fixed in a wind tunnel. For certain flow conditions in the transitional Reynolds number regime, i.e. $10,000 < Re < 1,000,000$, the wing absorbs energy from the flow and settles in a stable limit cycle oscillation (LCO). The LCO originates from laminar boundary layer separation which leads to negative aerodynamic damping at small angles of attack. An empirical model of the aeroelastic system is proposed in the form of a generalized Duffing-van der Pol oscillator. A statistical technique is developed to estimate the linear and nonlinear parameters of the aeroelastic oscillator. In addition,

the model noise term, in part accounting for the amplitude modulation of the LCO is also estimated. In particular, the Bayesian inference technique for parameter estimation involves a state estimation problem using the measurement data. The extended Kalman filter is used for the state estimation. To generate samples from the posterior joint probability density functions of the aeroelastic parameters, the Markov Chain Monte Carlo (MCMC) simulation technique is used. The methodology is first applied to surrogate data generated from a numerical model. Since the parameters are known *a priori* for the numerical model, this case verifies the Bayesian inference framework adopted in this investigation. The methodology is then applied to wind tunnel experimental data obtained from an elastically mounted rigid wing. The aerodynamic moment coefficient estimated from the wind tunnel experiment is then compared with the data available in the existing literature. The preliminary results of this investigation are published in the proceedings of ASME-FES [92] and IFASD [93]. The work contained in this chapter is published in an international journal [94] ¹.

In both civil and military sectors, there has been a significant increase in the use of flight vehicles at low Reynolds number. In the transitional Reynolds number regime specifically, i.e. $10^4 \leq Re \leq 10^6$, the separation of the laminar boundary layer may generate a separation bubble as the laminar shear layer transitions and re-attaches as a turbulent layer [156]. For a freely-rotating rigid NACA0012 airfoil in the transitional Reynolds number regime, experimental investigations identified the existence of self-sustained oscillations or LCOs of the airfoil [157]. The mechanism inducing LCOs is caused by laminar boundary layer separation leading to negative aerodynamic damping [158]. It is worth mentioning that McCroskey has reported on the sensitivity of the laminar separation dynamics for the NACA0012 airfoil to test conditions and model

¹Both the publisher and the coauthors permit the inclusion of this work in the dissertation

manufacturing precision [159]. However, the focus of McCroskey's report was on stall flutter occurring at higher angles of attack than the observed LCO under investigation. Previously, Huang *et al.* [160] obtained estimates of the aerodynamic loads on a stationary NACA0012 cantilevered wing in the same Reynolds number regime using wind tunnel tests. In this investigation, a generalized Duffing-van der Pol oscillator is used to model the system [158] and the aeroelastic coefficients are estimated using the experimental data of LCOs. Previously, the estimation process was performed using the ensemble Kalman filter which provides sequential joint state and parameter estimates [161]. However, this method involves modeling unknown static parameters as time-varying quantities to avoid filter divergence [51–54], even when these parameters are time-invariant (i.e. fixed or static).

The unknown parameters to be estimated relate to the aerodynamic loads acting on a NACA0012 airfoil undergoing limit cycle oscillation. In addition to the aerodynamic parameters, the modeling error term is also estimated. If the modeling error term is ignored in the dynamic model, the parameter estimates will be either highly biased or have high variances. The modeling error term also considers the effect of external excitation such as turbulence and unsteady aerodynamics. Regarding the latter, the Duffing-van der Pol oscillator implicitly assumes quasi-steady aerodynamics as it neglects the added mass effect and the time-dependent (memory effect) terms. A large mass ratio and relatively small reduced frequency justify the assumption of quasi-steady aerodynamics; refer to [158] for further details. Note that the error introduced by the quasi-steady assumption is compensated by the modeling error term.

To sample from the posterior of unknown parameters, the Metropolis-Hastings (MH) algorithm is used for MCMC (e.g. [58–60]). The marginal and joint probability density functions of the parameters are estimated using the MCMC samples. Using the noisy

measurements, the conditional probability density function (pdf) of the state, i.e. the pitch angle and pitch velocity, is required to estimate the posterior pdf of the parameters. The extended Kalman filter (EKF) is used for the state estimation which involves Gaussian approximation of the state [10, 19, 20]. Hence the additional contribution of the author lies in combining EKF and MCMC to sample from the posterior of the aeroelastic parameters for high-dimensional systems. Preliminary results are reported in [92, 93]. Mbalawata *et al.* recently combined the Hamiltonian Monte Carlo (HMC) algorithm of MCMC with EKF and applied the method to some canonical stochastic differential equations with synthetic data [162].

Despite strong nonlinearity in the model, it is observed that the assimilation of dense observational data leads to nearly Gaussian pdf of the state for both the numerical and physical experiments reported in this chapter. The nearly Gaussian nature of the conditional state vector was validated using Monte Carlo experiments but these results are not included for brevity. Thus, EKF is adequate for state estimation in this investigation. The strongly non-Gaussian state estimation problem can be handled by other nonlinear filters such as the unscented Kalman filter (UKF) [24, 163], ensemble Kalman filter [10, 34, 47, 92] and particle filter [40, 47, 92, 110], as discussed in the previous chapter. Furthermore, the parameter estimation method described later is very general. Only the state estimation algorithm therein would need to be changed to handle the strongly non-Gaussian cases. For example, EnKF is used for state estimation to tackle the parameter estimation problem of a Duffing oscillator (see Chapter 4 for details). Furthermore, Sandhu *et al.* [149] Bisailon *et al.* [150] applied this parameter estimation technique for Bayesian model selection purposes.

5.2 Theory

Bayesian estimation framework offers the ability to relate observational data with mathematical models [58, 60, 101]. In this approach, the so-called *forward* pdf or likelihood function defines the conditional pdf $p(\mathbf{d}|\phi)$ of the observation vector \mathbf{d} given the parameter vector ϕ . Using the observation \mathbf{d} , Bayesian inference provides estimates of the system parameter ϕ based on mathematical models. Consequently, the *inverse* or posterior pdf of parameter $p(\phi|\mathbf{d})$ of the system parameter vector ϕ is expressed as [10, 98–100].

$$p(\phi|\mathbf{d}) = \frac{p(\mathbf{d}|\phi)p(\phi)}{p(\mathbf{d})}. \quad (5.1)$$

The prior pdf $p(\phi)$ represents the available knowledge of the parameters before analyzing the observation vector \mathbf{d} [58, 60, 101]. The *posterior* pdf $p(\phi|\mathbf{d})$ contains the updated information of the system parameters using the available observations. In Eq. (5.1), the prior pdf is the product of the forward or *likelihood* pdf $p(\mathbf{d}|\phi)$ constructed using the mathematical model of the system. The factor $p(\mathbf{d}) = \int p(\mathbf{d}|\phi)p(\phi)d\phi$ is just a normalization constant.

In the following subsection, the posterior pdf $p(\phi|\mathbf{d})$ of the parameters will be derived using the Bayesian estimation method for discrete dynamical systems. The resulting expression for the posterior parameter pdf is the same as that in Eq. (4.11) which was derived using an alternative approach (see Subsection 4.2.2). The Markov Chain Monte Carlo method, complemented by the extended Kalman filter (see Section 2.4) for state estimation, is used to obtain samples of the parameter vector ϕ , as will be described in detail. The random-walk Metropolis algorithm [60] is used for MCMC sampling (see Section 3.3).

5.2.1 Mathematical Formulation For Parameter Estimation

The discrete state-space model of a general nonlinear system is represented by [9, 10, 19, 20, 27, 40, 53, 147, 148]

$$\mathbf{x}_{k+1} = \mathbf{g}_k(\mathbf{x}_k, \mathbf{f}_k, \mathbf{q}_k, \phi). \quad (5.2)$$

Here $\mathbf{x} \in \mathbb{R}^n$ represents the state vector, $\mathbf{g}_k \in \mathbb{R}^n$ defines the discrete nonlinear model operator, $\mathbf{f}_k \in \mathbb{R}^p$ defines the external deterministic input and $\mathbf{q} \in \mathbb{R}^s$ represents a Gaussian random model noise vector with mean $\overline{\mathbf{q}_k} \in \mathbb{R}^s$ and covariance matrix $\mathbf{Q}_k \in \mathbb{R}^{s \times s}$. The transitional density of the state vector \mathbf{x}_k is denoted by $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ [60].

It is assumed that the observation \mathbf{d}_k relating to the state \mathbf{x}_k can be described as

$$\mathbf{d}_k = \mathbf{h}_k(\mathbf{x}_k, \epsilon_k) \quad (5.3)$$

where \mathbf{h}_k defines the measurement operator and ϵ_k represents the measurement noise.

The conditional density $p(\mathbf{d}_k | \mathbf{x}_k)$ of the observation can be derived using Eq. (5.3).

The likelihood function of the unknown parameters is derived as follows [103]:

$$\begin{aligned} p(\mathbf{d}_{1:n} | \phi) &= p(\mathbf{d}_n, \mathbf{d}_{1:n-1} | \phi) \\ &= p(\mathbf{d}_n | \mathbf{d}_{1:n-1}, \phi) p(\mathbf{d}_{1:n-1} | \phi) \\ &= p(\mathbf{d}_{1:n-1} | \phi) \int p(\mathbf{d}_n, \mathbf{x}_n | \mathbf{d}_{1:n-1}, \phi) d\mathbf{x}_n \\ &= p(\mathbf{d}_{1:n-1} | \phi) \int p(\mathbf{d}_n | \mathbf{x}_n, \mathbf{d}_{1:n-1}, \phi) p(\mathbf{x}_n | \mathbf{d}_{1:n-1}, \phi) d\mathbf{x}_n \\ &= p(\mathbf{d}_{1:n-1} | \phi) \int p(\mathbf{d}_n | \mathbf{x}_n, \phi) p(\mathbf{x}_n | \mathbf{d}_{1:n-1}, \phi) d\mathbf{x}_n \\ &= \prod_{k=1}^n \int p(\mathbf{d}_k | \mathbf{x}_k, \phi) p(\mathbf{x}_k | \mathbf{d}_{1:k-1}, \phi) d\mathbf{x}_k, \end{aligned} \quad (5.4)$$

where $\mathbf{d}_{1:n} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n\}$.

Bayes' theorem provides the posterior pdf of the parameters as

$$\begin{aligned} p(\phi|\mathbf{d}_{1:n}) &\propto p(\phi) p(\mathbf{d}_{1:n}|\phi) \\ &\propto p(\phi) \prod_{k=1}^n \int p(\mathbf{d}_k|\mathbf{x}_k, \phi) p(\mathbf{x}_k|\mathbf{d}_{1:k-1}, \phi) d\mathbf{x}_k. \end{aligned} \quad (5.5)$$

The samples of the parameter vector ϕ are generated from the posterior pdf, $p(\phi|\mathbf{d}_{1:n})$, given in Eq. (5.5) using MCMC sampling technique. In Eq. (5.5), the likelihood $p(\mathbf{d}_k|\mathbf{x}_k, \phi)$ is the conditional pdf of the measurement vector \mathbf{d}_k given the current state vector \mathbf{x}_k and the parameter vector ϕ . The forecast pdf $p(\mathbf{x}_k|\mathbf{d}_{1:k-1}, \phi)$ denotes the conditional pdf of the state vector \mathbf{x}_k conditioned upon the observations $\mathbf{d}_{1:k-1}$ and the parameter vector ϕ .

In this investigation, it is assumed that the (partial) measurements of the state \mathbf{x}_k are corrupted by an additive Gaussian noise vector ϵ_k and thus $p(\mathbf{d}_k|\mathbf{x}_k, \phi)$ is a Gaussian distribution. Using EKF, the conditional pdf $p(\mathbf{x}_k|\mathbf{d}_{1:k-1}, \phi)$ of the state \mathbf{x}_k is estimated, being also Gaussian. The function being integrated in Eq. (5.5) is a product of two Gaussian functions of \mathbf{x}_k . The integral, in this case, has an analytical solution provided in Appendix A.4. The flow chart in Fig. 5.1 outlines the steps required to obtain samples of the posterior pdf, $p(\phi|\mathbf{d}_{1:n})$, in Eq. (5.5).

5.3 Numerical Study - Self-Sustained Oscillations of an Airfoil

The methodology is first verified using the surrogate data obtained from the following numerical model. The limit cycle oscillation (LCO) can be modeled by [92, 93]

$$I\ddot{\theta} + D\dot{\theta} + K\theta + D' \operatorname{sgn}(\dot{\theta}) = M + \sigma\xi(t). \quad (5.6)$$

The left-hand side represents the structural forces, including some dry friction with a coefficient D' . The right-hand side consists of two terms: 1) the modeling error term represented by a unit Gaussian white noise $\xi(t)$ having strength σ and 2) the aerodynamic moment M approximated by a 3rd order polynomial in θ and $\dot{\theta}$ as

$$M = a_1 + a_2\theta + a_3\dot{\theta} + a_4\theta^2 + a_5\theta\dot{\theta} + a_6\dot{\theta}^2 + a_7\theta^3 + a_8\theta^2\dot{\theta} + a_9\theta\dot{\theta}^2 + a_{10}\dot{\theta}^3. \quad (5.7)$$

When the structural and aerodynamic terms are lumped together, one can rewrite Eq. (5.6) as

$$\ddot{\theta} = b_1 + b_2\theta + b_3\dot{\theta} + b_4 \operatorname{sgn}(\dot{\theta}) + b_5\theta^2 + b_6\theta\dot{\theta} + b_7\dot{\theta}^2 + b_8\theta^3 + b_9\theta^2\dot{\theta} + b_{10}\theta\dot{\theta}^2 + b_{11}\dot{\theta}^3 + b_{12}\xi(t) \quad (5.8)$$

with θ as the pitch angle and $\xi(t)$ the unit Gaussian white noise term with b_{12} denoting its strength. $b_1 \dots, b_{12}$ are estimated using Bayesian inference from a set of noisy

observational data obtained at discrete times t_k :

$$d_k = \theta(t_k) + \epsilon_k \quad (5.9)$$

where ϵ_k is the white Gaussian measurement noise.

The state-space representation of Eq. (5.8) is

$$\dot{x}_1 = x_2, \quad (5.10)$$

$$\begin{aligned} \dot{x}_2 = & b_1 + b_2x_1 + b_3x_2 + b_4 \operatorname{sgn}(x_2) + b_5x_1^2 + b_6x_1x_2 + b_7x_2^2 \\ & + b_8x_1^3 + b_9x_1^2x_2 + b_{10}x_1x_2^2 + b_{11}x_2^3 + b_{12}\xi(t), \end{aligned} \quad (5.11)$$

where $x_1 = \theta$ and $x_2 = \dot{\theta}$.

The above set of equations can be cast as two coupled Ito stochastic differential equations:

$$dx_1 = x_2 dt, \quad (5.12)$$

$$\begin{aligned} dx_2 = & [b_1 + b_2x_1 + b_3x_2 + b_4 \operatorname{sgn}(x_2) + b_5x_1^2 + b_6x_1x_2 + b_7x_2^2 \\ & + b_8x_1^3 + b_9x_1^2x_2 + b_{10}x_1x_2^2 + b_{11}x_2^3] dt + b_{12}\xi(t) dt, \end{aligned} \quad (5.13)$$

where $\xi(t) dt = dW = W(t_{k+1}) - W(t_k)$ is a Brownian path increment.

The Ito stochastic differential equations are discretized using the Euler-Maruyama

stochastic integration scheme [118–120] with time step Δt to obtain

$$\{x_1\}_{k+1} = \{x_1\}_k + \Delta t \{x_2\}_k, \quad (5.14)$$

$$\begin{aligned} \{x_2\}_{k+1} = \{x_2\}_k + \Delta t [& b_1 + b_2 x_1 + b_3 x_2 + b_4 \operatorname{sgn}(x_2) + b_5 x_1^2 + b_6 x_1 x_2 + b_7 x_2^2 \\ & + b_8 x_1^3 + b_9 x_1^2 x_2 + b_{10} x_1 x_2^2 + b_{11} x_2^3] + b_{12} \sqrt{\Delta t} \varepsilon_k \end{aligned} \quad (5.15)$$

where ε_k are independent and identically distributed standard Gaussian random variables. The above state space model can be rewritten in the form of Eq. (5.2), in which $\mathbf{x} = \{x_1, x_2\}$.

For numerical illustrations, the following values of the aeroelastic parameters are considered: $b_1 = 0$, $b_2 = -540$, $b_3 = 4$, $b_4 = -0.05$, $b_5 = -180$, $b_6 = 0$, $b_7 = 0$, $b_8 = 2.4 \times 10^4$, $b_9 = -2.2 \times 10^3$, $b_{10} = 0$, $b_{11} = 0$ and $b_{12} = 0.07$. These parameter values lead to self-sustained oscillations with frequency and amplitude typical of experimental observations [157, 158]. In particular, note that the positive linear damping term b_3 is dynamically destabilizing. For larger amplitudes of oscillation, the negative non-linear damping term b_9 is stabilizing and becomes more dominant in comparison with the linear term b_3 and thereby limits the amplitude of oscillation. The time integration step is taken to be $\Delta t = 1 \times 10^{-4}$ and the variance of the measurement noise is chosen to be 5×10^{-6} , which closely mimics the noise observed in the wind tunnel experiments. Fig. 5.2 displays a sample pitch trajectory of the system (without measurement noise). The corresponding measured pitch (corrupted by measurement noise) is also plotted in Fig. 5.2 and this noisy data is used to estimate the model parameters. The 1 kHz sampling rate for data acquisition purposes resembles that of the wind tunnel experiments. The influence of model noise manifests itself in the LCO through both amplitude and frequency modulations. The uncorrelated small amplitude oscillation in the measured

data is introduced by the measurement noise.

The parameters b_1 through b_{12} will be estimated using Bayesian inference. A filtered version of the measured response using a low-pass filter with cut-off frequency 16 Hz is used to maintain consistency with the cases dealing with real wind tunnel experiments in the following section. The recorded wind tunnel observations are contaminated by noise with a dominant 30 Hz random signal superposing a relatively flat spectrum in other frequencies. The 30 Hz component is mainly attributed to the noise originating from the data acquisition hardware. If an unfiltered response is used, the white measurement noise assumption will be strongly violated resulting in biased parameter estimates. On the other hand, a filtered version of the signal will contain a measurement noise signal with a frequency spectrum that is flat up to 16 Hz and drops significantly beyond that. Although the filtered measurements contain a noise signal that is no longer theoretically white, it can be reasonably approximated strictly as a white noise as the frequency spectrum of the observed response also drops significantly beyond 16 Hz. Maybeck provides a justification behind this assumption in [164]. Thus a filtered measurement signal will be used for estimation purposes in order to maintain the white measurement noise assumption in EKF.

It is assumed for data assimilation purposes that the resulting filtered signal is corrupted by a measurement noise with an estimated variance equal to 4×10^{-7} . Eq. (5.5) describes the posterior pdf $p(\phi = \{b_1, \dots, b_{12}\} | \mathbf{d}_{1:n})$ of the parameters. EKF computes the forecast pdf $p(\mathbf{x}_k = x_k | \mathbf{d}_{1:k-1} = d_{1:k-1}, \phi = \{b_1, \dots, b_{12}\})$ in Eq. (5.5). The measured response is plotted in Fig. 5.2b. A *non-informative* prior (uniform pdf) satisfying positivity constraints whenever applicable is used. The estimated marginal pdfs of the parameters are plotted in Fig. 5.3. The model uncertainty is represented by the additive noise term $b_{12}\xi(t)$ in Eq. (5.6). Evidently, any informative prior of the model param-

eter vector ϕ (which includes the modeling noise strength) will *regularize* the inverse problem.

Fig. 5.3 displays the normalized histograms obtained from 5×10^6 MCMC samples with 160 MCMC chains running on a Linux cluster (176 core HP Intel Xeon cluster consisting of 22 nodes with 8 cores and 32 GB of memory per node) using the Message-Passing Interface (MPI). The true parameters are identified by dashed lines. As the measurement data contains only limited information regarding the system, the mode of the pdf fail to coincide with the actual parameter values. As expected, the model noise intensity parameter b_{12} exhibits non-Gaussian features as evident in Fig. 5.3. The other parameters have close to Gaussian pdfs. The shaded area corresponds to the probability of the parameter being the other side of zero of its MAP (maximum a posteriori) estimate, being the mode of the posterior distribution [102]. This will be used later to arrive at a reduced model.

For this case, most of the parameters pairs do not show significant correlation. For the parameter pairs having correlation coefficient greater than 0.5 in magnitude, the joint pdfs are plotted in Fig. 5.4. The justification of the high correlation among some parameter pairs can be provided using physical considerations. For instance, as the total stiffness of the system is bounded, the negative correlation between the linear stiffness coefficient b_2 and nonlinear stiffness coefficient b_8 is expected as they balance each other in terms of magnitude. The high correlation among other parameter pairs can also be explained in terms of redundancy. This issue would be elaborated in more detail in the next section. In Fig. 5.5, three sample LCO trajectories simulated using the MAP parameter estimates are plotted. All three trajectories show reasonable agreement with the filtered measured response. The slight discrepancies among the samples generated using MAP parameter estimates and measured output emerges appear in the transition

rate to LCO and the amplitude and frequency modulation of the LCO.

Further experiments have been carried out to examine the effect of increasing measurement noise intensity on the parameter estimates. Although the variance of the measurement noise was increased from 5×10^{-6} to 5×10^{-4} , the posterior pdfs were still observed to be unimodal, albeit with greater variance. Even when the measurement noise was increased to 5×10^{-2} , the posterior pdfs still maintained their unimodality. The results of these experiments were not shown here for brevity.

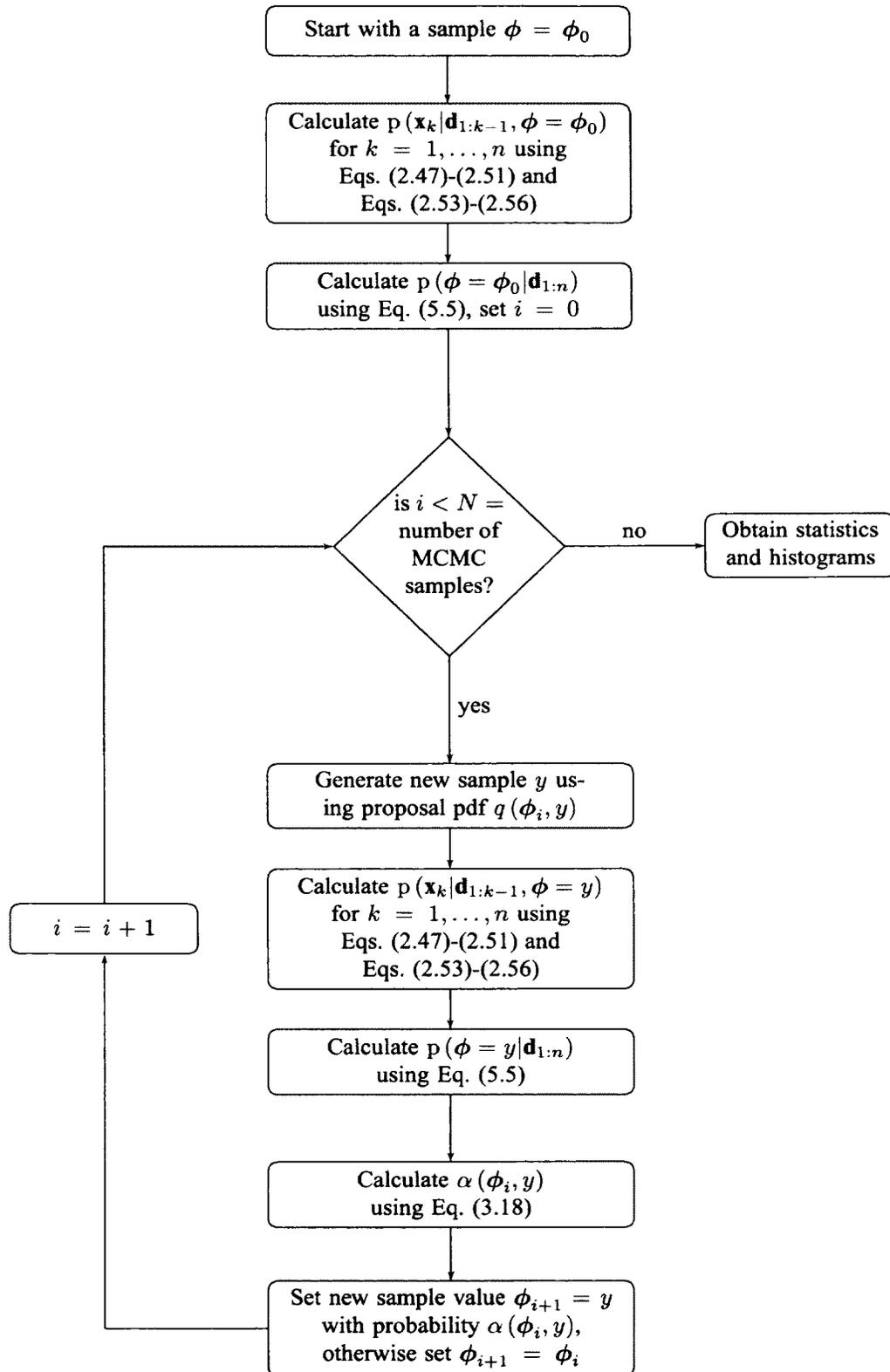


Figure 5.1: Flow chart describing algorithm used to sample the posterior parameter pdf for the fluttering aeroelastic system.

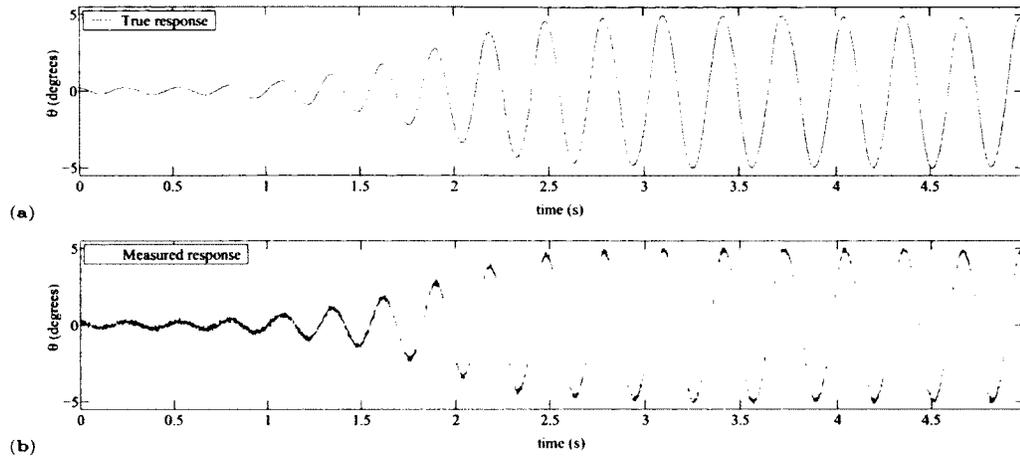


Figure 5.2: Fluttering aeroelastic system - Numerical study: (a) true response and (b) measured noisy response of the perturbed nonlinear oscillator.

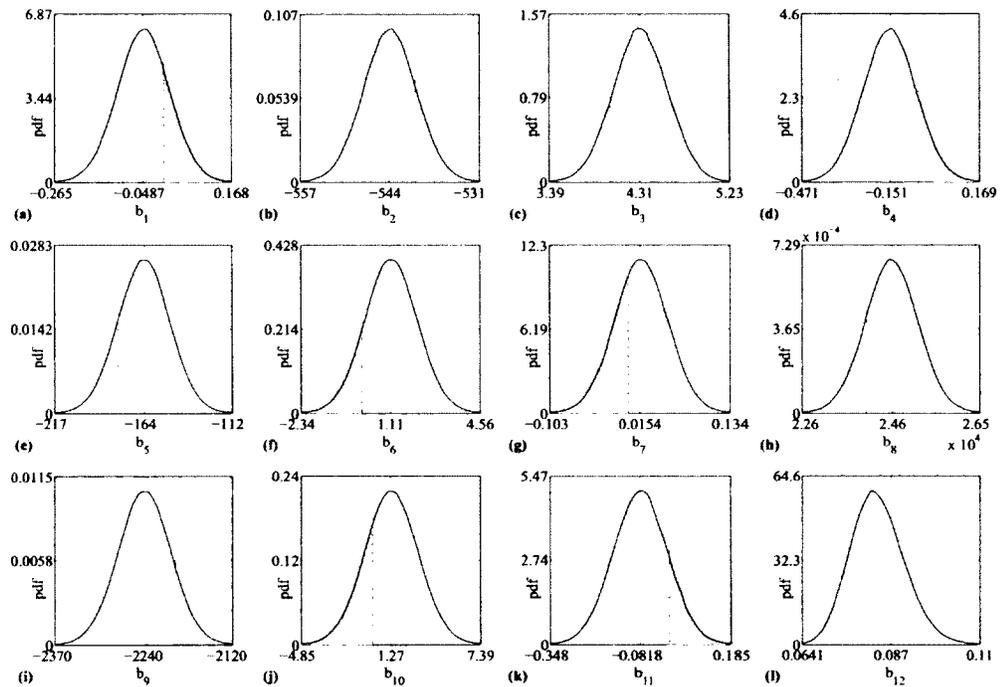


Figure 5.3: Fluttering aeroelastic system - Numerical study: posterior parameter pdfs obtained using MCMC simulations for parameter (a) b_1 , (b) b_2 , (c) b_3 , (d) b_4 , (e) b_5 , (f) b_6 , (g) b_7 , (h) b_8 , (i) b_9 , (j) b_{10} , (k) b_{11} and (l) b_{12} . The dashed lines indicate the true parameter values.

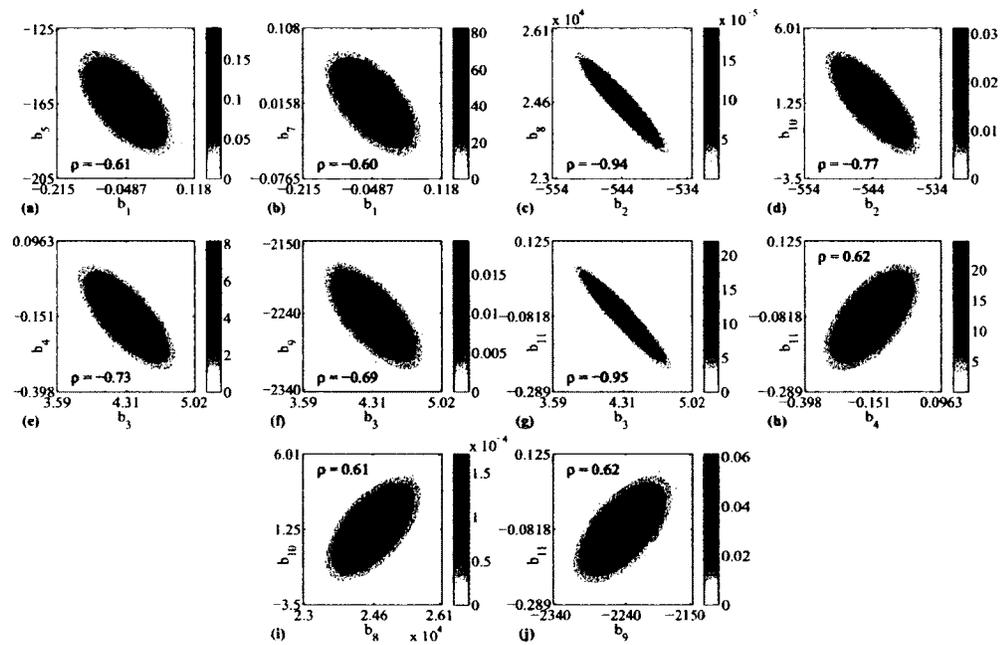


Figure 5.4: Fluttering aeroelastic system - Numerical study: posterior joint parameter pdfs obtained using MCMC simulations for parameters (a) b_1 and b_5 , (b) b_1 and b_7 , (c) b_2 and b_8 , (d) b_2 and b_{10} , (e) b_3 and b_4 , (f) b_3 and b_9 , (g) b_3 and b_{11} , (h) b_4 and b_{11} , (i) b_8 and b_{10} and (j) b_9 and b_{11} .

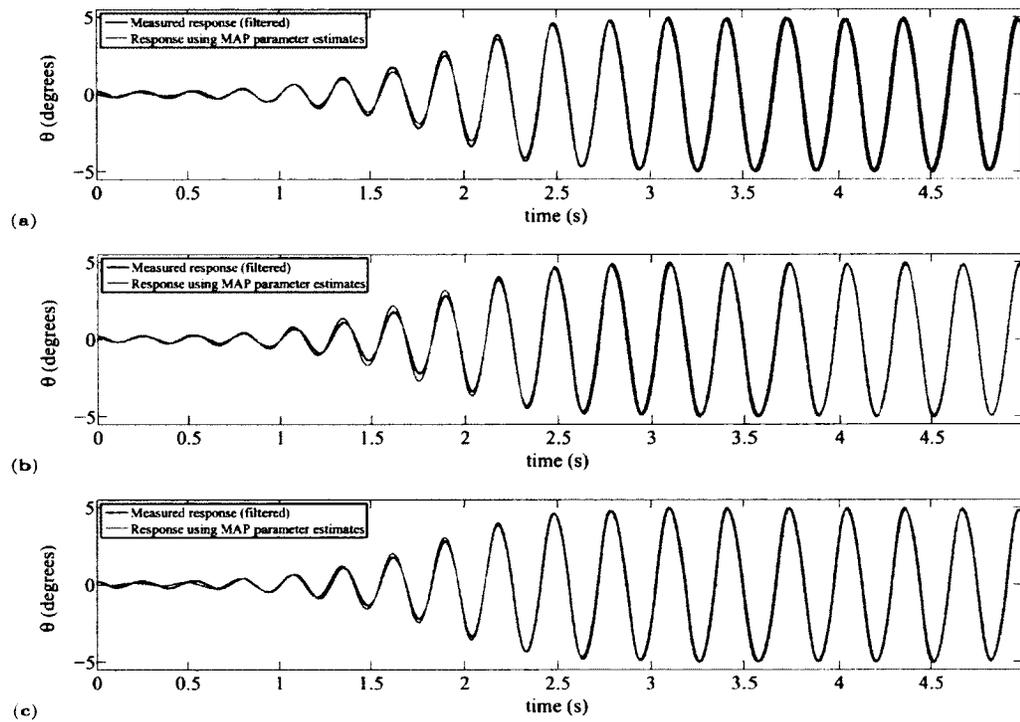


Figure 5.5: Fluttering aeroelastic system - Numerical study: three pitch realizations obtained using MAP parameter estimates superposing the measured data.

5.3.1 A Reduced Model

No mathematical model in engineering mechanics is the "true model" which produces the observational data (e.g. [3]). Hypothetically, even if one assumes that the *true* model is known, it may not be practically useful for inference (understanding and prediction) when fitted using finite data set as its large number of parameters are estimated with low precision [3]. Even if a very complex model represented by a large number of coupled nonlinear equations with numerous parameters can provide good predictive capability, it is of little use to *understand* the dynamics of the system without further approximation and simplification [3]. A reduced model with further simplification (albeit with inflated modeling error) may provide better understanding [3]. When the state of a system is measured, the associated data contains information and noise. A reduced model of the system strives to compactly represent the information in data by filtering out the noise with the primary objective to address: "*What inferences do the data support about the population?*" [3].

Given a data set, the efficacy of a reduced model hinges on whether it is over-fitted or under-fitted, not by the criterion if it closely mimics a (unknown) *true* model [3]. This fact relates to the principle of parsimony that involves choice of models with minimum number of parameters by bias and variance trade-off [3]. An elegant description of under-fitting and over-fitting is provided by Burnham and Anderson as follows [3]. The over-fitted and under-fitted models do not imply or assume the existence of a low-dimensional model representing "truth". As the number of model parameters increases, the misfit of model output with data (bias) decreases. On the other hand, the error or variance of the parameter estimates increases making inference difficult from such models. Some dominant features will be present in most of the samples. One can make (strong) inference about these dominant features using *a single sample*. Only weak inference

may be made about other sub-dominant features exhibited by some (say, 70%) of the samples. Only a few samples will display some rare features such as extreme events. Attempts to make inference about such rare events specific to a given sample leads to over-fitting as the resulting inferences is not relevant to the most samples. Burnham and Anderson further state that such rare features (appearing only in a few samples) are better handled by modeling error term. When models fail to extract the dominant features in data (present in most samples), such models *under-fit* data. An optimal reduced model seeks to compromise the error due to approximation in modeling and the error due to random fluctuations by bias and variance trade-off [3].

For the numerical model used to obtain synthetic LCO data, some terms were not influential since five parameters were chosen to be zero. Thus it is known for this case that a reduced model can and should capture the observed system response. In Fig. 5.4, the strong correlations among some parameter pairs are due to the presence of redundant terms. In Fig. 5.3, some parameters show high likelihood of being on the opposite side of zero of their MAP estimates (indicated by the shaded area). For an over-fitted model (having larger number of parameters), the variance of the parameter estimates increases, but the associated bias decreases. To obtain an optimal reduced model using the bias-variance trade-off principle, the parameter with the largest coefficient of variation (or largest normalized variance) is removed successively in order to reduce the degree of over-fitting. Since the posterior pdfs are unimodal, the coefficient of variation directly relates to the probability of a parameter being on the other side of zero. The term b_7 (multiplying $\dot{\theta}^2$) has the highest probability of taking values on the other side of zero of its MAP estimate. Therefore b_7 is eliminated and the remaining 11 parameters are estimated again using MCMC sampling. This process is repeated until the parameter estimates are either strictly positive or negative. This successive elimination process

reduces the model to 6 unknown parameters given by

$$\ddot{\theta} = b_2\theta + b_3\dot{\theta} + b_5\theta^2 + b_8\theta^3 + b_9\theta^2\dot{\theta} + b_{12}\xi(t). \quad (5.16)$$

Following the successive elimination process, 6 terms were removed. Out of these 6 terms, 5 were absent in the actual model. The remaining term is the dry friction term. It is conjectured that this term has negligible effect on the observed output. In Figs. 5.6-5.7, the marginal and joint pdfs of the parameters are plotted using 5×10^6 MCMC samples for the reduced model. The strong correlation between b_2 and b_8 stiffness parameters, as well as between b_3 and b_9 damping parameters, is due to their physical relationship and not due to redundancy in the model. In order to remove these correlations, the reduced model could be expressed using orthogonal polynomials. However, the parameters corresponding to these orthogonal polynomials lose physical significance. Fig 5.8 provides three realizations of the pitch response simulated using MAP estimates of the parameters showing good agreement with the observed output. This simple model reduction procedure is based on the *bias and variance trade-off* principle. Recently, Bayesian model selection [165] has been applied to the same wind-tunnel experimental data with the model reduction performed using the exact version of Bayesian Information Criterion [149]. The methodology has been extended to tackle the model selection problem of more general case of nonlinear systems [150].

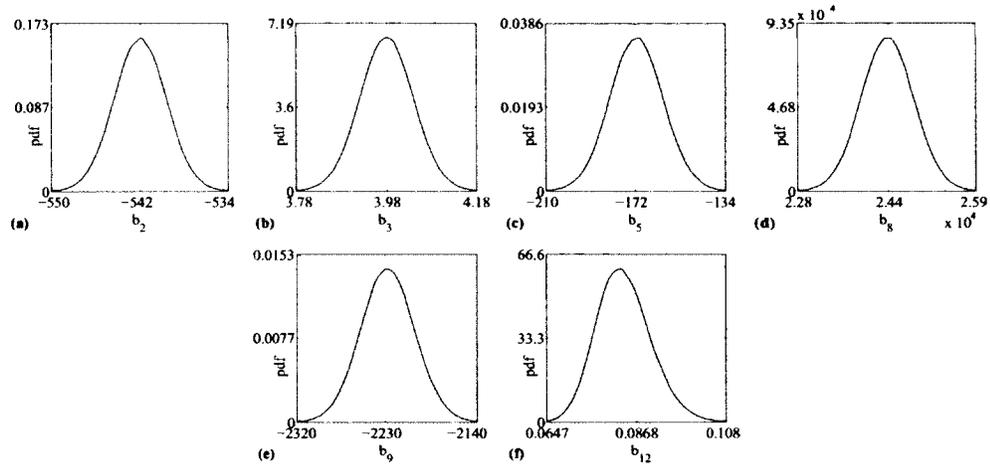


Figure 5.6: Fluttering aeroelastic system - Numerical study: reduced model posterior parameter pdfs obtained using MCMC simulations for parameter (a) b_2 , (b) b_3 , (c) b_5 , (d) b_8 , (e) b_9 and (f) b_{12} . The dashed lines indicate the true parameter values.

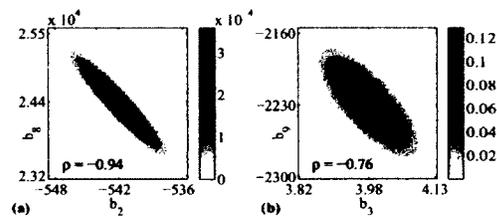


Figure 5.7: Fluttering aeroelastic system - Numerical study: reduced model posterior joint parameter pdfs obtained using MCMC simulations for parameters (a) b_2 and b_8 and (b) b_3 and b_9 .

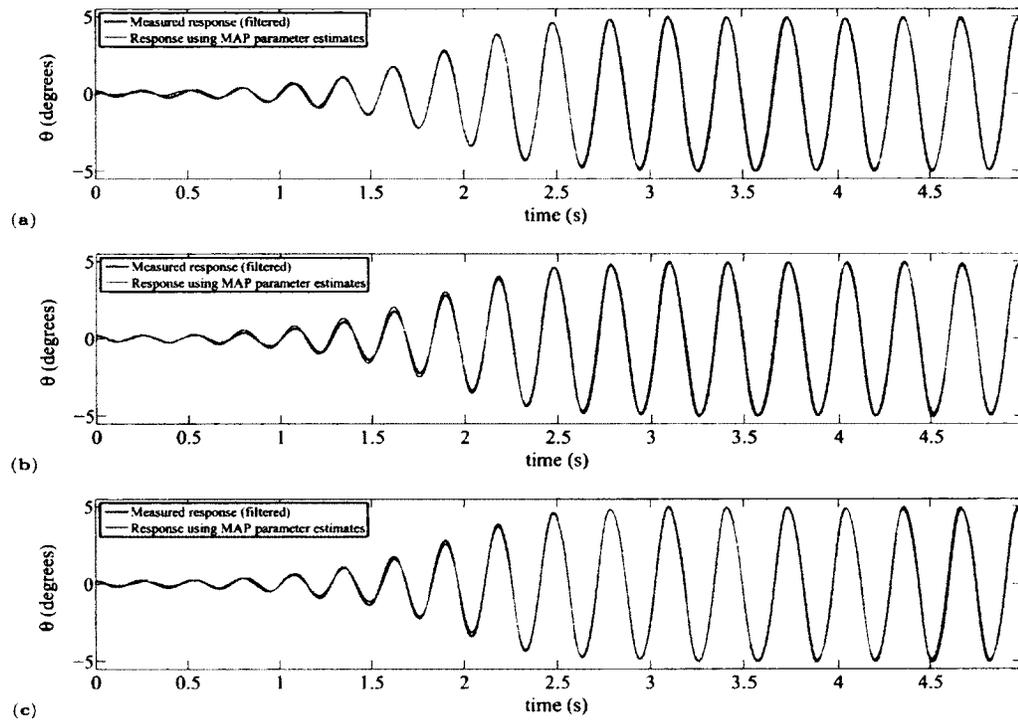


Figure 5.8: Fluttering aeroelastic system - Numerical study: three pitch realizations obtained using MAP parameter estimates of the reduced model superposing the measured data.

5.4 Wind Tunnel Experiments - Self-Sustained Oscillations of an Airfoil

The aerodynamic parameters of a NACA0012 airfoil undergoing LCO at transitional Reynolds number regime are estimated next. The wind tunnel experimental setup and data used herein are provided by Professor Dominique Poirel of the Royal Military College of Canada. The author carried out repeatability tests with the resulting observations shown in Fig. 5.15. A closed circuit wind tunnel having a test section of $0.76 \text{ m} \times 1.08$, capable of producing flow speeds ranging from 5 to 60 m/s having a maximum turbulence intensity level of 0.2% for the range of airspeeds considered. At the inlet, a pitot-static tube linked to a manometer measures the airspeed. In Fig. 5.9, the schematic diagram of a rigid NACA0012 airfoil is shown which undergoes pure rotation only. Its supporting structure contains two similar top and bottom translating sub-systems, on which the rotation mechanisms are mounted. The rotation mechanism consists of pulley-springs which introduce the elastic restoring force. The wing span and chord length are 0.61 m and 0.156 m, respectively leading to an aspect ratio of $AR = 3.9$. Furthermore, 3D effects are minimized using end plates. The distance between the wing tips and end plates is 7 mm (1% of the span), resulting in a solid blockage ratio of 5%. In its nominal configuration, the elastic axis, EA, otherwise known as axis of rotation, is located 0.01 m forward of the quarter-chord point (at 18.6% chord aft of the leading edge). For the rotating parts of the system, the mass ratio μ is approximately 55.6. Potentiometers are used to measure the response with a sampling rate of 1000 Hz (i.e. $\Delta t = 0.001 \text{ s}$). Using the free-decay response (no-flow case), the structural parameters are first estimated. For the flow velocity of 8.5 m/s, corresponding to $Re = 88,000$, the aerodynamic parameters of the model using LCO data are then estimated.

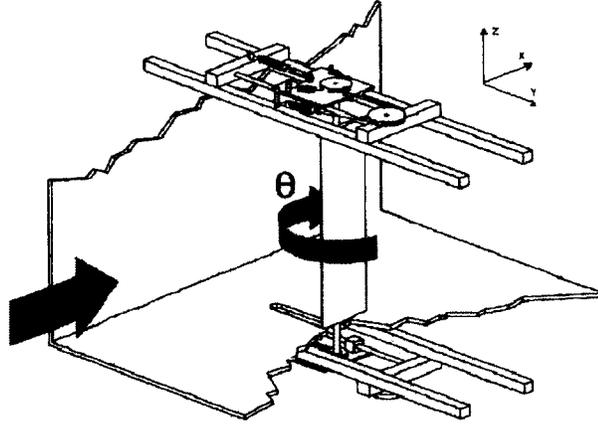


Figure 5.9: Fluttering aeroelastic system - Experimental setup of the NACA0012 airfoil in the wind tunnel.

5.4.1 Free-Decay Data

In the absence of airflow, the pitching motion of the airfoil can be modeled as

$$I\ddot{\theta} + D\dot{\theta} + K\theta + K'\theta^3 + D' \operatorname{sgn}(\dot{\theta}) = B + \sigma\xi(t) . \quad (5.17)$$

The above equation can also be expressed as

$$\ddot{\theta} = c_1 + c_2\theta + c_3\dot{\theta} + c_4\theta^3 + c_5 \operatorname{sgn}(\dot{\theta}) + c_6\xi(t) . \quad (5.18)$$

The cubic stiffness term accounts for the nonlinearity at higher amplitudes of oscillation. The dry friction at the support of the airfoil is modeled by the “sgn” term. The bias in the pitch output is captured by the c_1 or B term. Using Bayesian inference, $c_1 \dots, c_6$ are estimated from noisy pitch data sampled at discrete times t_k as

$$d_k = \theta(t_k) + \epsilon_k \quad (5.19)$$

where ϵ_k denotes a zero mean white Gaussian measurement noise having variance of

1×10^{-6} for the measurements filtered with an FIR filter having a cutoff frequency of 16 Hz is used. Note that this cutoff frequency is approximately seven times higher than the natural frequency. The filtered response is plotted in Fig. 5.10.

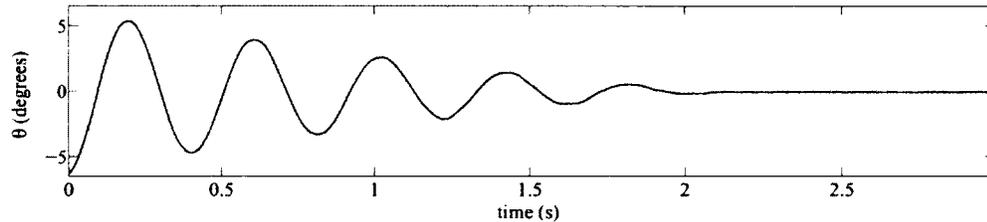


Figure 5.10: Fluttering aeroelastic system - Wind tunnel experiments: filtered free decay response of the NACA0012 airfoil.

Using these filtered measurements, the marginal pdfs of the parameters are estimated and plotted in Fig. 5.11. 5×10^6 MCMC samples were generated using 160 parallel MCMC chains to provide the estimated pdfs. Joint pdfs with correlation coefficients greater than 0.5 are plotted in Fig. 5.12. Using the MAP parameter estimates, three realizations of the pitch response are simulated and shown in Fig. 5.13. These samples differ due to the model noise term having different realizations, yet they show a good agreement with the measured response. As the model noise intensity is estimated to be significantly small, the model is capable of capturing the essential physics reasonably well.

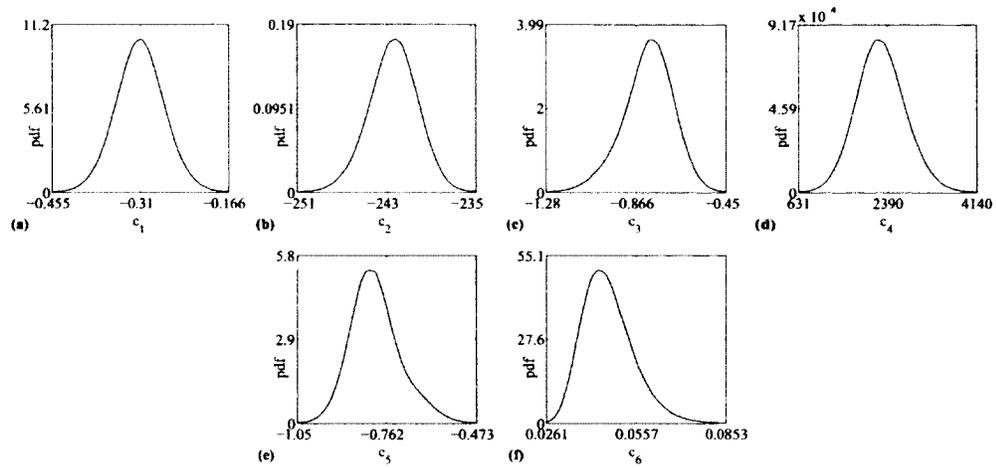


Figure 5.11: Fluttering aeroelastic system - Wind tunnel free-decay experiments for NACA0012 airfoil: posterior parameter pdfs obtained using MCMC simulations for parameter (a) c_1 , (b) c_2 , (c) c_3 , (d) c_4 , (e) c_5 and (f) c_6 .

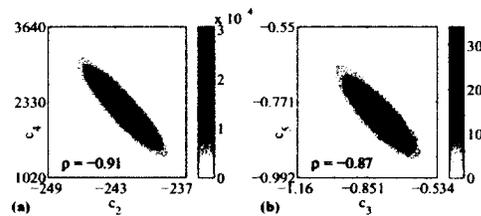


Figure 5.12: Fluttering aeroelastic system - Wind tunnel free-decay experiments for NACA0012 airfoil: posterior joint parameter pdfs obtained using MCMC simulations for parameters (a) c_2 and c_4 and (b) c_3 and c_5 .

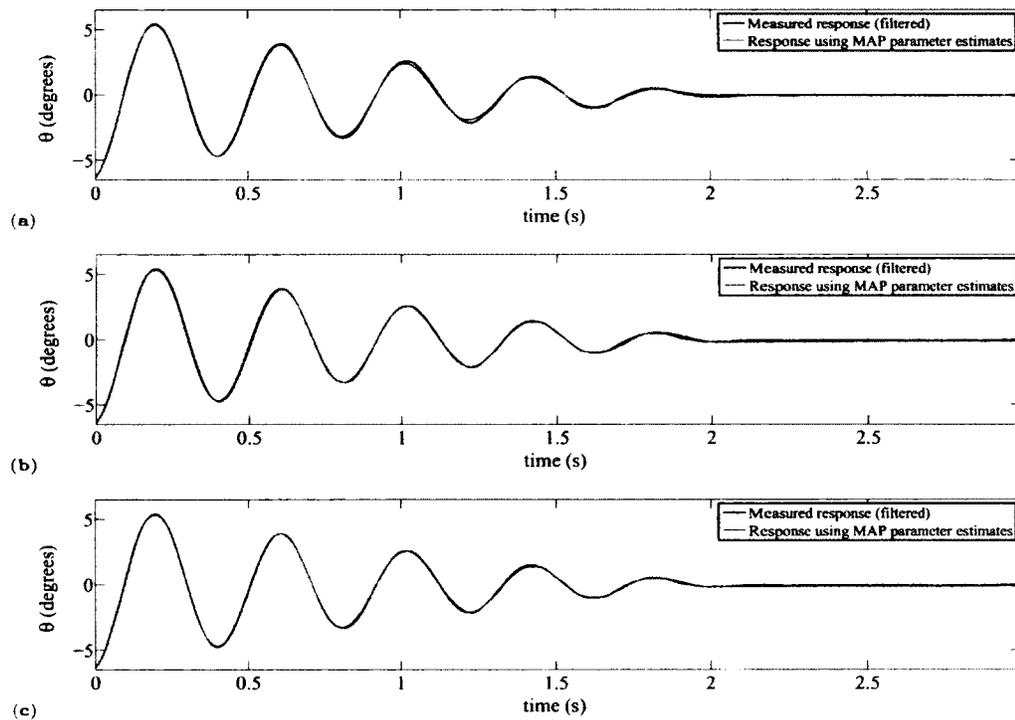


Figure 5.13: Fluttering aeroelastic system - Wind tunnel free-decay experiments for NACA0012 airfoil: three pitch realizations obtained using MAP parameter estimates superposing the measured data.

5.4.2 Limit Cycle Oscillation Data

As evident in Fig. 5.14, the equilibrium point of approximately zero pitch becomes unstable due to small perturbations and the system re-stabilizes on an LCO (having an amplitude of approximately 5 degrees) at an airspeed of 8.5 m/s (i.e. $Re = 88,000$). No periodicity is observed in the upstream flow, measured with hot-wire anemometry, to account for the observed LCO. Furthermore, when the wing is fixed, no periodicity is observed in the wake eliminating the possibility of vortex induced vibration [157]. It can thus be concluded that the oscillation is self-sustained or LCO with a single dominant frequency, at $f = 3.1$ Hz, and weak super-harmonics, at $2f$ and $3f$. Weaker higher order harmonics are also detected in the pitch response. In the power spectrum density of the pitch response, the magnitudes of these higher order harmonics are five orders lower than that of the fundamental harmonic and thus their effects are neglected for the subsequent analysis. To minimize the influence of measurement noise, an FIR filter is used with a cutoff frequency of 16 Hz. Clearly, the FIR filter does not affect the dominant frequency content of the LCO.

In order to ensure independence of the steady state LCO to initial conditions, different experiments were performed. Three sample pitch time responses are shown in Fig. 5.15. In the top two cases, the loss of stability of the fixed point was triggered by the natural turbulent excitation present in the wind tunnel, hence the different time at which the loss of stability occurs, whereas for the bottom case, a finite but small pitch displacement was given by the test operator at $t = 4.5$ s. Through the *bias and variance trade-off*, the optimal reduced model intends to capture the dominant features in the signal such as the transient regime (between LCO and onset of instability) and LCO characteristics (e.g. modulated amplitude and frequency). These features exist in any typical LCO signal. Inferences can be made from a *single sample* about these features. To arrive

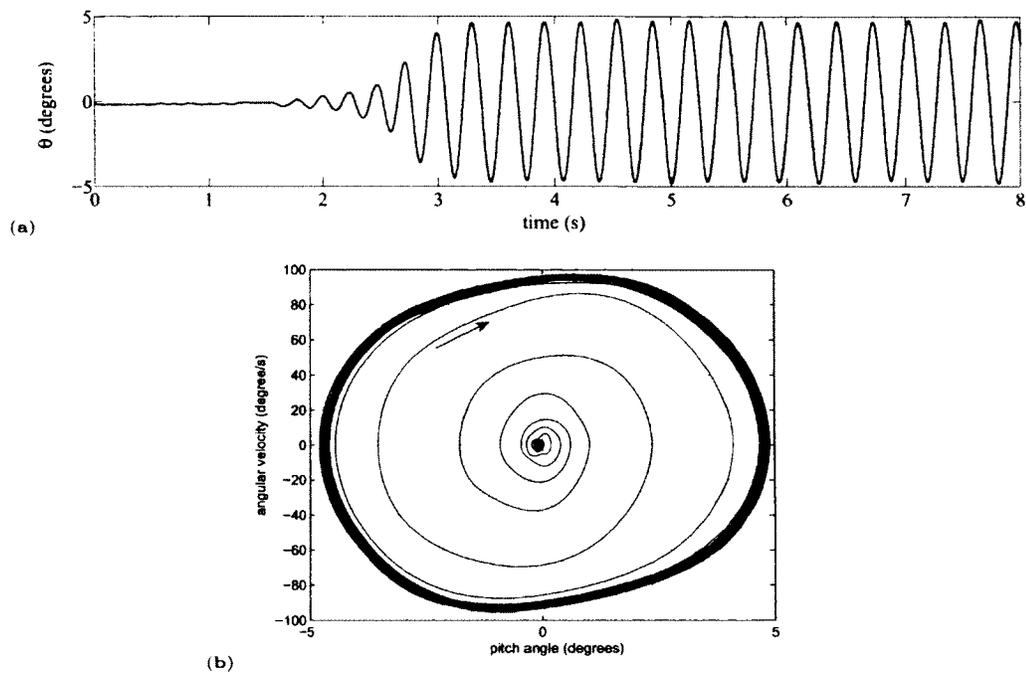


Figure 5.14: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: (a) filtered LCO time history and (b) phase plane.

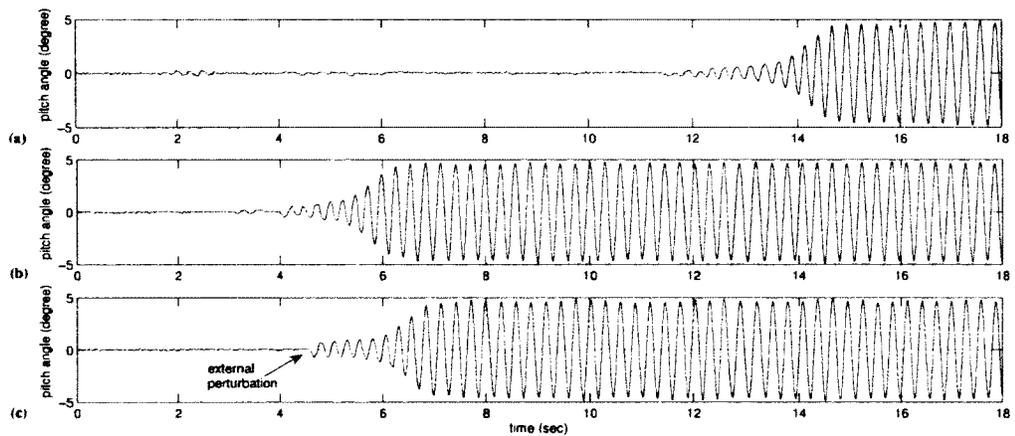


Figure 5.15: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: three filtered LCOs for different initial conditions.

at the optimal reduced model, the bias and variance trade-off must be achieved. This can be accomplished by performing numerical simulations of multiple models obtain by successively eliminating terms from the comprehensive (full) model. More rigorous information-theoretic framework for optimal model selection is described in [3].

The following equation is assumed to describe the LCO

$$I\ddot{\theta} + D\dot{\theta} + K\theta + K'\theta^3 + D' \operatorname{sgn}(\dot{\theta}) = M + B + \sigma\xi(t) \quad (5.20)$$

with a 5th order polynomial used to model the nonlinear aerodynamic moment as

$$\begin{aligned} M = & d_1 + d_2\theta + d_3\dot{\theta} + d_4\theta^2 + d_5\theta\dot{\theta} + d_6\dot{\theta}^2 \\ & + d_7\theta^3 + d_8\theta^2\dot{\theta} + d_9\theta\dot{\theta}^2 + d_{10}\dot{\theta}^3 \\ & + d_{11}\theta^4 + d_{12}\theta^3\dot{\theta} + d_{13}\theta^2\dot{\theta}^2 + d_{14}\theta\dot{\theta}^3 + d_{15}\dot{\theta}^4 \\ & + d_{16}\theta^5 + d_{17}\theta^4\dot{\theta} + d_{18}\theta^3\dot{\theta}^2 + d_{19}\theta^2\dot{\theta}^3 + d_{20}\theta\dot{\theta}^4 + d_{21}\dot{\theta}^5. \end{aligned} \quad (5.21)$$

Consequently, Eq. (5.20) can be written as

$$\begin{aligned} \ddot{\theta} = & c_5 \operatorname{sgn}(\dot{\theta}) + e_1 + e_2\theta + e_3\dot{\theta} \\ & + e_4\theta^2 + e_5\theta\dot{\theta} + e_6\dot{\theta}^2 \\ & + e_7\theta^3 + e_8\theta^2\dot{\theta} + e_9\theta\dot{\theta}^2 + e_{10}\dot{\theta}^3 \\ & + e_{11}\theta^4 + e_{12}\theta^3\dot{\theta} + e_{13}\theta^2\dot{\theta}^2 + e_{14}\theta\dot{\theta}^3 + e_{15}\dot{\theta}^4 \\ & + e_{16}\theta^5 + e_{17}\theta^4\dot{\theta} + e_{18}\theta^3\dot{\theta}^2 + e_{19}\theta^2\dot{\theta}^3 + e_{20}\theta\dot{\theta}^4 + e_{21}\dot{\theta}^5 + e_{22}\xi(t). \end{aligned} \quad (5.22)$$

where the value of c_5 is taken to be the MAP estimate as previously obtained from the free-decay experiments. Initially, the estimation process is first carried out using a 3rd

order polynomial representation of the aerodynamic moment, but the predicted output is unsatisfactory when compared with the experimental data in the transient regime. Therefore, a 5th order polynomial model of the nonlinear aerodynamic moment is adopted.

Using the measurements in Fig. 5.14, the estimated posterior marginal pdfs of the parameters are plotted in Fig. 5.16. The probability of a parameter being on the other side of zero of its MAP estimate is represented by the shaded area in Fig. 5.16. From 160 parallel MCMC chains, 20×10^6 MCMC samples are generated to estimate these pdfs. For the correlation coefficient greater than 0.5, some selected posterior joint pdfs are plotted in Fig. 5.17. Fig. 5.18 displays three sample trajectories of the system simulated using the MAP parameter estimates. These simulated sample pitch trajectories show reasonable agreement with the measured response from the wind tunnel experiments. Notice some difference among the simulated trajectories and the measured response in (a) the transition rate to LCO, (b) the amplitude and frequency modulation in LCO.

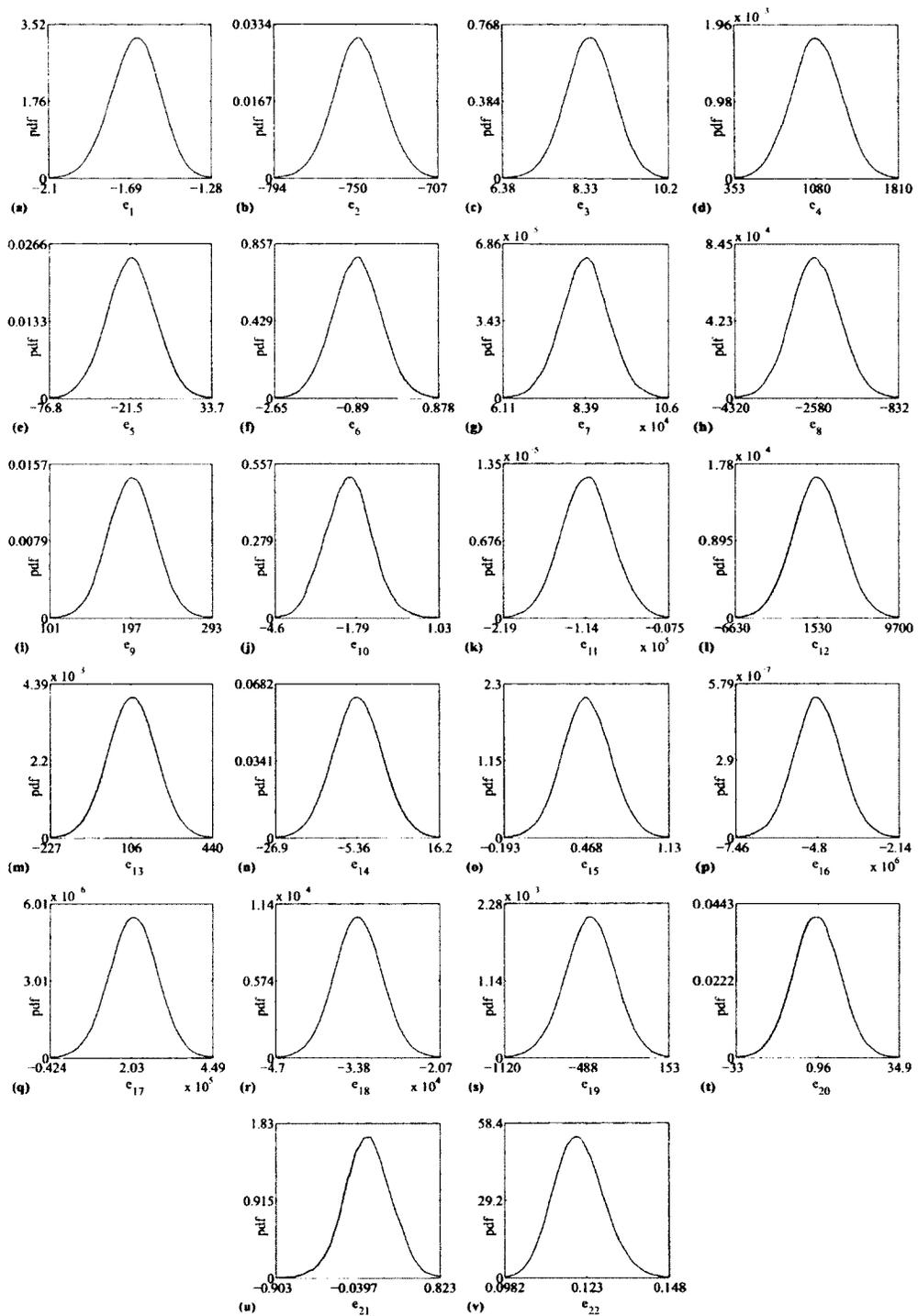


Figure 5.16: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: posterior parameter pdfs obtained using MCMC simulations for parameter (a) e_1 , (b) e_2 , (c) e_3 , (d) e_4 , (e) e_5 , (f) e_6 , (g) e_7 , (h) e_8 , (i) e_9 , (j) e_{10} , (k) e_{11} , (l) e_{12} , (m) e_{13} , (n) e_{14} , (o) e_{15} , (p) e_{16} , (q) e_{17} , (r) e_{18} , (s) e_{19} , (t) e_{20} , (u) e_{21} and (v) e_{22} .

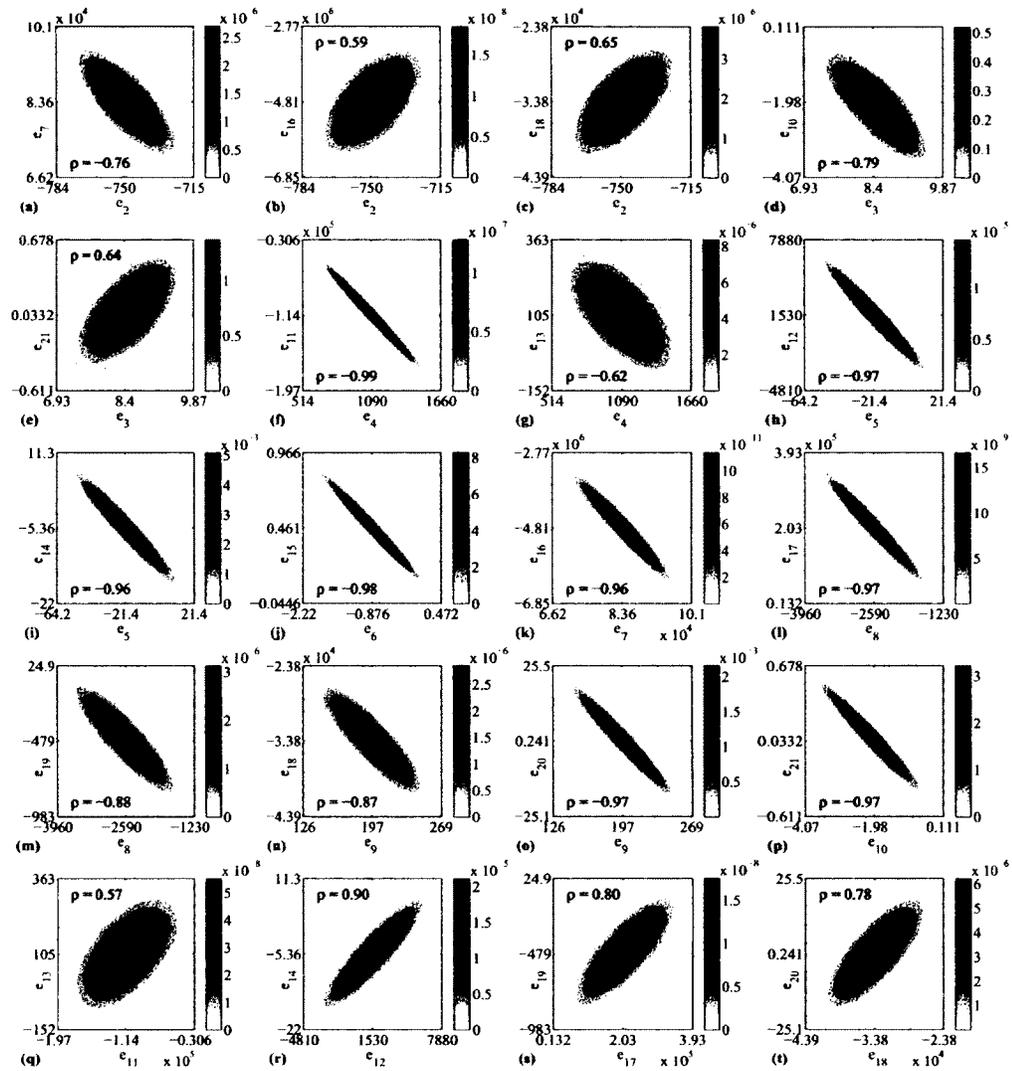


Figure 5.17: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: posterior joint parameter pdfs obtained using MCMC simulations for parameters (a) e_2 and e_7 , (b) e_2 and e_{16} , (c) e_2 and e_{18} , (d) e_3 and e_{10} , (e) e_3 and e_{21} , (f) e_4 and e_{11} , (g) e_4 and e_{13} , (h) e_5 and e_{12} , (i) e_5 and e_{14} , (j) e_6 and e_{15} , (k) e_7 and e_{16} , (l) e_8 and e_{17} , (m) e_8 and e_{19} , (n) e_9 and e_{18} , (o) e_9 and e_{20} , (p) e_{10} and e_{21} , (q) e_{11} and e_{13} , (r) e_{12} and e_{14} , (s) e_{17} and e_{19} , (t) e_{18} and e_{20} .

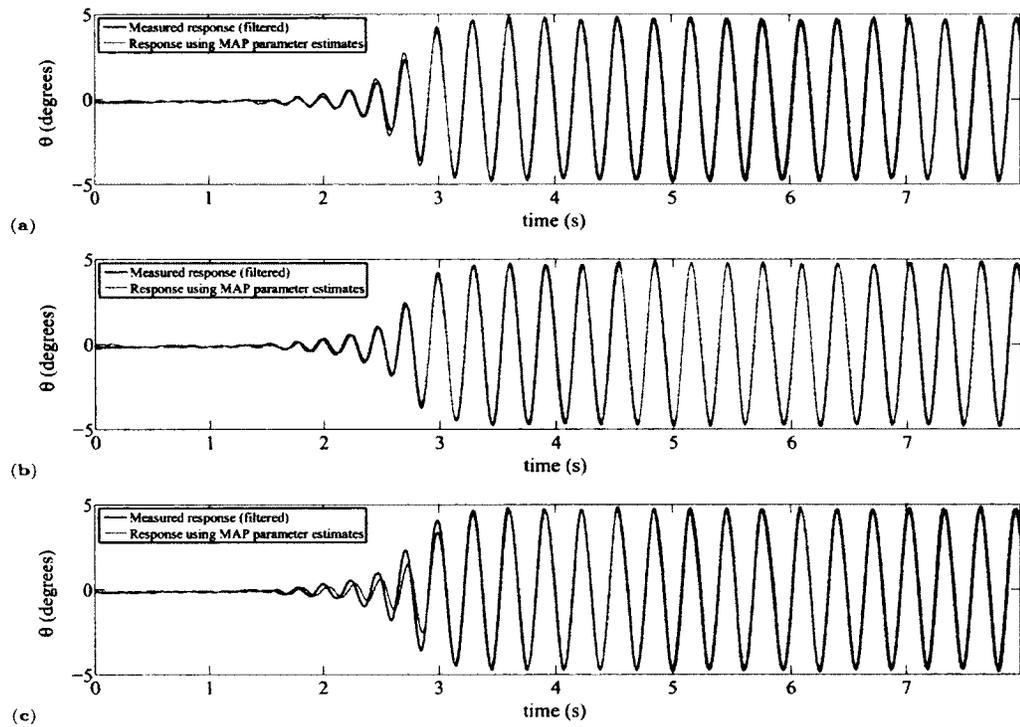


Figure 5.18: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: three pitch realizations obtained using MAP parameter estimates superposing the measured data.

5.4.3 A Reduced Model

The redundancy among some parameters may give rise to strong correlations as observed in Fig. 5.17. The term e_{20} multiplying $\theta\dot{\theta}^4$ has the highest probability of being on the other side of zero of its MAP estimate as indicated by the shaded area of its marginal pdf in Fig. 5.16. Following the similar logic outlined in Subsection 5.3.1, the term associated with this parameter is removed from the equation of motion and estimate the joint pdf of the remaining 21 parameters using MCMC sampling technique. The parameter elimination process is repeated until the remaining parameters of the model become positive or negative with probability almost 1. This successive elimination process leads to the reduced model with 14 parameters given by

$$\begin{aligned}
 \ddot{\theta} = & c_5 \operatorname{sgn}(\dot{\theta}) + e_1 + e_2\theta + e_3\dot{\theta} \\
 & + e_4\theta^2 + e_5\theta\dot{\theta} \\
 & + e_7\theta^3 + e_8\theta^2\dot{\theta} + e_9\theta\dot{\theta}^2 + e_{10}\dot{\theta}^3 \\
 & + e_{11}\theta^4 \\
 & + e_{16}\theta^5 + e_{18}\theta^3\dot{\theta}^2 + e_{19}\theta^2\dot{\theta}^3 + e_{22}\xi(t). \tag{5.23}
 \end{aligned}$$

To arrive at the optimal reduced model in Eq. (5.23), two odd and six even terms are eliminated from the original comprehensive model in Eq. (5.22). The presence of 6 redundant even terms may point out the negligible misalignment of the wing with the flow in terms of effect of pitch response. This is due to the fact that the aerodynamic moment for an uncambered airfoil is expected to be symmetric with respect to zero pitch angle.

For the reduced model in Eq. (5.23), the marginal and joint pdfs of the estimated

parameters are plotted in Figs. 5.19-5.20 using 20×10^6 MCMC samples. In Fig. 5.20, the strong correlation among some parameter pairs are due to the physical dependence among the corresponding terms, as observed for the nonlinear stiffness parameters e_7 and e_{16} , for example.

Using MAP parameter estimates of the reduced model, three sample pitch trajectories are plotted in Fig. 5.21 which again show good agreement with the wind tunnel measurements. The corresponding phase space diagram of the simulated and observed response are shown in Fig. 5.22. Again, these pitch trajectories show good agreement in the phase space. The effect of nonlinearity in the system is highlighted by the deviation from ellipticity of the orbit in the phase space diagram.

In Fig. 5.23, the static aerodynamic moment coefficient (i.e. for $\dot{\theta} = 0$) is plotted using the estimated parameters e_1 through e_{22} for the reduced model in Eq. (5.23) and the structural parameters c_1 through c_4 in Eq. (5.18). The moment coefficient is calculated using an estimated moment of inertia about the elastic axis of $0.0014 \text{ kg} \cdot \text{m}^2$ and is normalized by $(\rho U^2 / 2) \times c^2 \times s$ with $c = 0.156 \text{ m}$, $\rho = 1.19 \text{ kg/m}^3$, $s = 0.61 \text{ m}$ and $U = 8.5 \text{ m/s}$. Only the odd terms are considered when estimating the moment coefficient as the even terms in the fifth order moment model account for a bias arising from the misalignment of the airfoil with the flow in the wind tunnel.

For comparative and validation purposes, Fig. 5.23 shows the static moment (measured at quarter-chord) as experimentally observed by Huang *et al.* [160] for a cantilevered NACA0012 airfoil at $\text{Re} = 7.6 \times 10^4$. For our estimates, the effect of uncertainty is most significant in the pitch angle ranging from 2.5 to 4.3 degrees. Examining the phase portrait in Fig. 5.14, during one period of LCO less data is collected at these angular values than at higher or lower ones leading to higher uncertainty in the estimated moment coefficient for pitch angles ranging from 2.5 to 4.3 degrees. As the data is col-

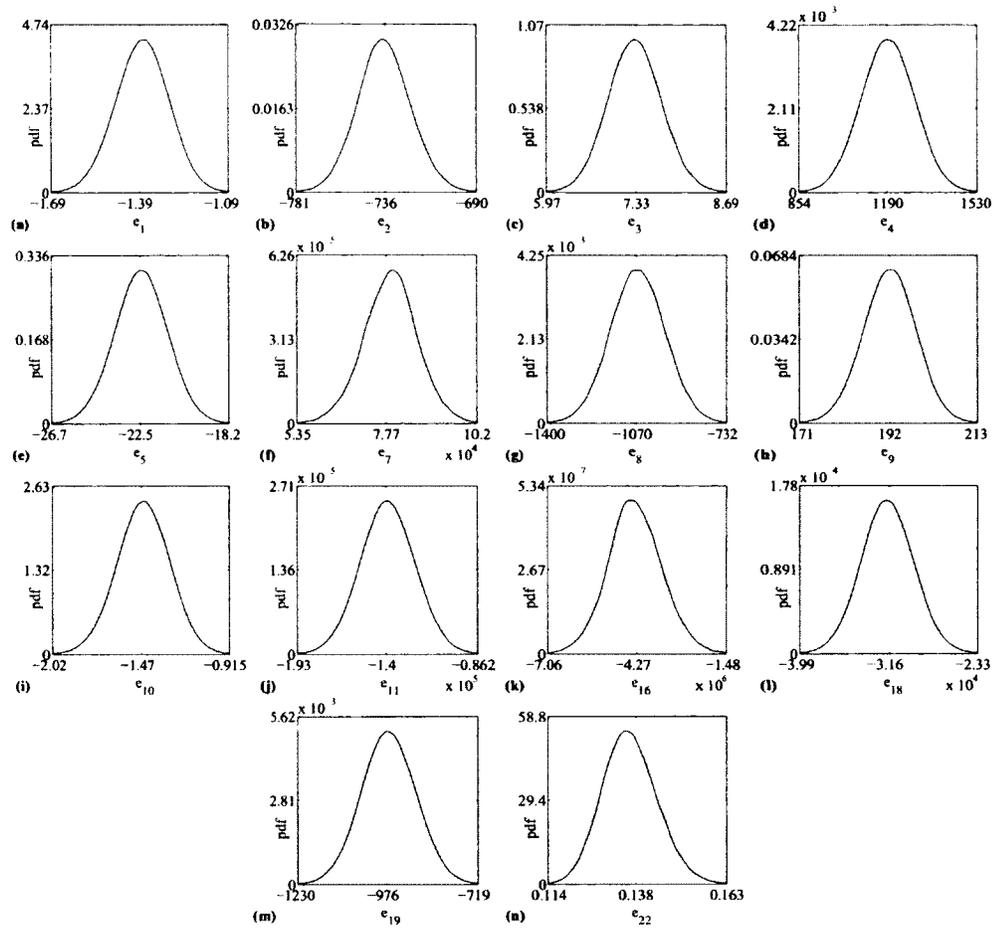


Figure 5.19: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: reduced model posterior parameter pdfs obtained using MCMC simulations for parameter (a) e_1 , (b) e_2 , (c) e_3 , (d) e_4 , (e) e_5 , (f) e_7 , (g) e_8 , (h) e_9 , (i) e_{10} , (j) e_{11} , (k) e_{16} , (l) e_{18} , (m) e_{19} and (n) e_{22} .

lected for pitch angles bounded by approximately 4.8 degrees (the amplitude of LCO), the increase in uncertainty associated with the moment coefficient for pitch angles beyond 4.8 degrees is due to the effect of extrapolation.

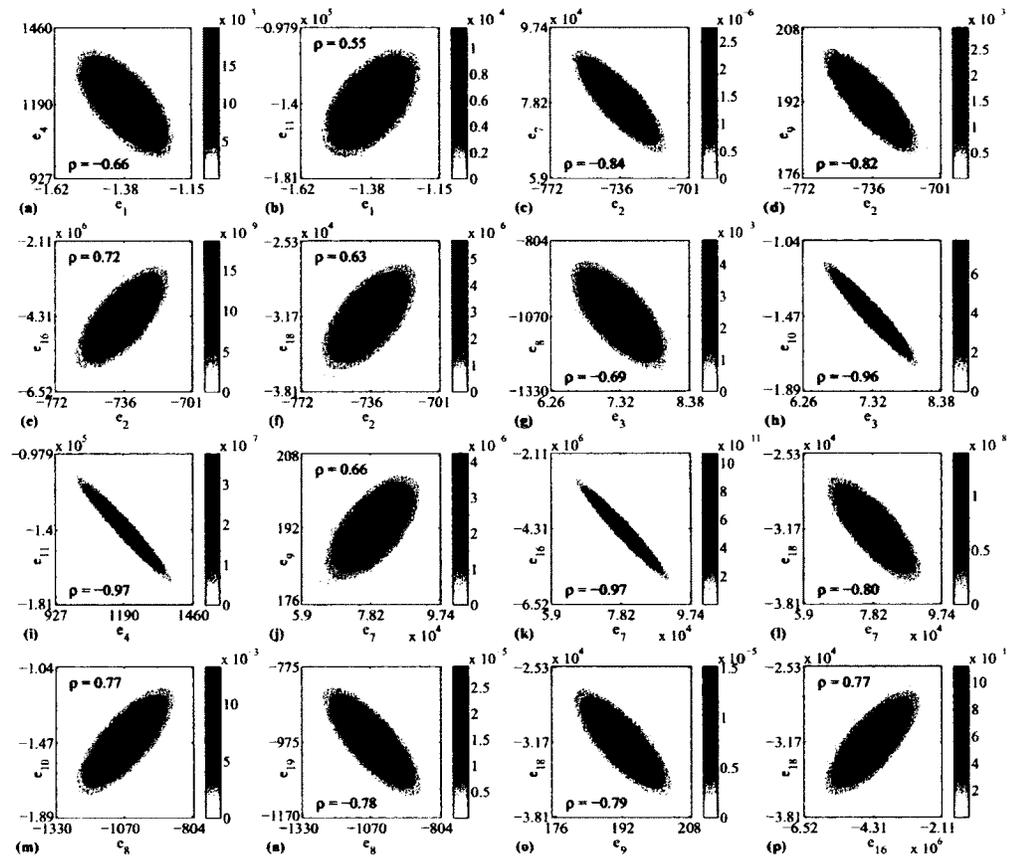


Figure 5.20: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: reduced model posterior joint parameter pdfs obtained using MCMC simulations for parameters (a) e_1 and e_4 , (b) e_1 and e_{11} , (c) e_2 and e_7 , (d) e_2 and e_9 , (e) e_2 and e_{16} , (f) e_2 and e_{18} , (g) e_3 and e_8 , (h) e_3 and e_{10} , (i) e_4 and e_{11} , (j) e_7 and e_9 , (k) e_7 and e_{16} , (l) e_7 and e_{18} , (m) e_8 and e_{10} , (n) e_8 and e_{19} , (o) e_9 and e_{18} and (p) e_{16} and e_{18} .

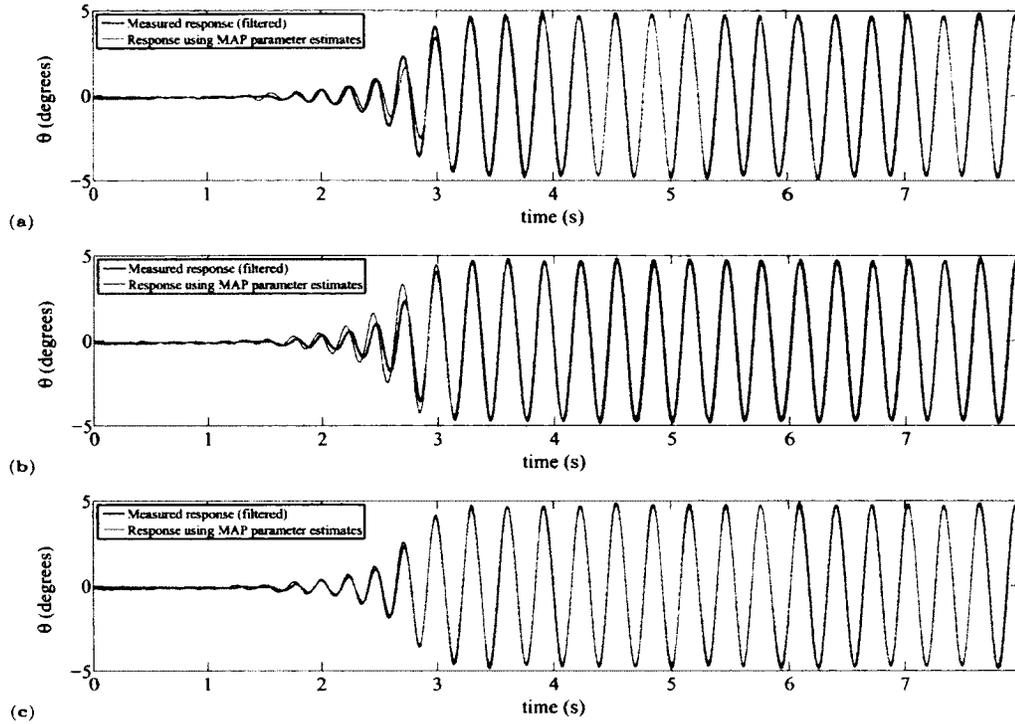


Figure 5.21: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: three pitch realizations obtained using MAP parameter estimates of the reduced model superposing the measured data.

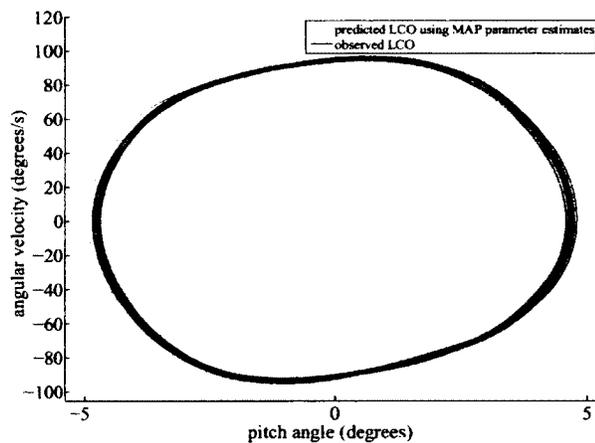


Figure 5.22: Fluttering aeroelastic system - Wind tunnel LCO experiments for NACA0012 airfoil: phase space diagrams of observed LCO and predicted LCO using MAP parameter estimates.

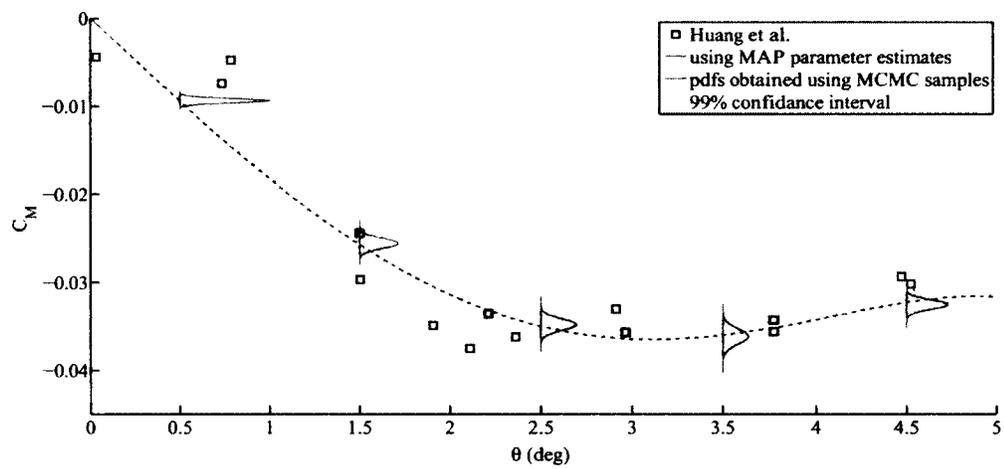


Figure 5.23: Fluttering aeroelastic system - Estimated static aerodynamic moment coefficient about the EA located at 18.6% chord for the NACA0012 airfoil at $Re = 88,000$. Also shown are experimental observations made by Huang *et al.* [160] for a cantilevered NACA0012 airfoil at $Re = 76,000$, with moment taken about quarter chord point.

Chapter 6

Parallel Data Assimilation for Large-Scale Models of Stochastic PDEs

Parallel data assimilation algorithms for filtering applications with large-scale models are of increasing interest due to the availability of cost-effective high-performance computing systems and large amounts of data sets gathered from sensor networks [63]. KF and EKF filtering techniques require (a) large computational resources to propagate the error covariance matrix and (b) large amount of memory to store the error covariance matrix. These requirements limit the application of these filtering algorithms to high-dimensional state space models compared to EnKF which partly alleviates these problems. For PF algorithms, (a) the number of particles grows exponentially with the dimension of the system for good performance [64, 65] and (b) a resampling step of the posterior joint pdf is needed to avoid degeneracy. Both of these requirements limit the applicability of PF for high-dimensional models [10]. Although some effort has been focused on parallelizing other filtering algorithms such as the Kalman filter [66] and particle filter [67], the parallelization of EnKF has received the most attention recently [68–76].

EnKF uses a low-rank representation of the state error covariance matrix by approximating it through ensemble averaging. EnKF inherits the analysis step of KF. A major limitation of applying EnKF to tackle a large-scale data assimilation problem is the requirement of a large ensemble size [69], often demanding excessive computational cost. To alleviate these issues, the recently reported Kalman filter algorithm [80] based on a polynomial chaos representation [1, 81] is adopted for parallelization to tackle the problem of sequential data assimilation for high-dimensional state space models governed by stochastic partial differential equations (PDEs) relevant to engineering mechanics. Preliminary findings of this work are published in the proceedings of AIAA-SDM [96]

6.1 Polynomial Chaos Representation for Uncertainty Propagation

The polynomial chaos representation can be used to represent non-stationary and non-Gaussian stochastic processes [1]. The polynomial chaoses $\{\Psi_j(\theta)\}_{i=0}^{\infty}$ constitute a generic basis of the space of all second-order random variables which can be used to represent a stochastic process $a(\mathbf{x}, \theta)$ using a truncated series as [1]

$$a(\mathbf{x}, \theta) = \sum_{j=0}^P a_j(\mathbf{x}) \Psi_j(\theta) , \quad (6.1)$$

where $a_j(\mathbf{x})$ are deterministic functions of the spatial coordinate \mathbf{x} and $\Psi_j(\theta)$ are polynomials of known random variables. See Appendix C for a brief overview of PCE.

To fully characterize the model output of a stochastic system using PCE, one simply needs to obtain the deterministic coefficients $a_j(\mathbf{x})$. In order to accomplish this

task, the PCE representation of the stochastic input and output processes are inserted into the partial differential equation representing the dynamics of the system. Using a Galerkin scheme, a coupled set of deterministic PDEs are obtained for the deterministic coefficients. The finite element technique can be used to tackle these coupled set of PDEs leading to the so-called spectral stochastic finite element method (SSFEM) [1]. This approach was first applied to problems of structural mechanics with random spatial parameters [1].

The above approximation (Eq. (6.1)) provides a discretization in the stochastic dimension. It is often combined with a finite element discretization [166, 167] (being a discretization in the spatial domain) and a discretization in time (e.g. the Newmark-Beta method [168]) to solve stochastic PDEs. “For large-scale engineering problems of practical interest, the application of SSFEM becomes computationally challenging as the size of the algebraic linear system grows rapidly with the spatial mesh resolution and the order of the stochastic dimension” [87]. The domain decomposition method (DDM) [82–87] may be used to efficiently solve the resulting system of equations using supercomputers. In the framework of non-linear filtering, the domain decomposition method for stochastic PDEs translates to an efficient forecast step for high-dimensional state space models that circumvent computationally intensive Monte Carlo simulations.

In the following subsections, a brief overview of SSFEM for stochastic PDEs based on PCE representation of random system parameters and response is provided. This is followed by a review of the Newmark-Beta method for time-marching. The domain decomposition method is introduced next to solve the resulting large-scale system of equations. The last subsection introduces the polynomial chaos based Kalman filter (PCKF) algorithm [80] as a data assimilation method.

6.2 Domain Decomposition Methods for Spectral SFEM

In this section, SSFEM is reviewed and the recently proposed DDM algorithm [82–87] is introduced as an efficient forecast step of filtering algorithms for high-dimensional state space models. In numerous problems of engineering mechanics, the realistic compute simulations should take into account the effect of random heterogeneity of the system parameters [1, 82–87]. In many cases, this fact leads to a governing stochastic PDEs (SPDEs) with stochastic operators (left hand side). In general, a stochastic PDE has the following form:

$$\mathcal{L}_t [u(\mathbf{x}, t, \theta)] + \mathcal{L}_s [u(\mathbf{x}, t, \theta)] = f(\mathbf{x}, t, \theta) \quad (6.2)$$

where $\mathbf{x} = \{x, y, z\} \in \Omega$ is a vector of physical coordinates, $t \in [0, T]$ is the variable denoting time and $\theta \in \Omega$ refers to the stochastic dimension and \mathcal{L}_t and \mathcal{L}_s represent the temporal and spatial differential operators. For example, the operators \mathcal{L}_t and \mathcal{L}_s may be represented as [166]

$$\mathcal{L}_t = -\rho(\theta) \frac{\partial^2}{\partial t^2} - \mu(\theta) \frac{\partial}{\partial t}, \quad (6.3)$$

$$\mathcal{L}_s = \frac{\partial}{\partial x} \left(k_x(\theta) \frac{\partial}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y(\theta) \frac{\partial}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_z(\theta) \frac{\partial}{\partial z} \right). \quad (6.4)$$

For elementary exposition, $k_x(\theta)$, $k_y(\theta)$ and $k_z(\theta)$ is assumed to be random functions of spatial co-ordinate \mathbf{x} (for more general cases, please refer to [169, 170]). Thus, $k_x(\theta)$, $k_y(\theta)$ and $k_z(\theta)$ can be represented by a general (non-stationary and non-Gaussian) stochastic process with a polynomial chaos expansion as $k_x(\theta) = \sum_{i=0}^P k_{x,i} \Psi_i(\theta)$, $k_y(\theta) = \sum_{i=0}^P k_{y,i} \Psi_i(\theta)$ and $k_z(\theta) = \sum_{i=0}^P k_{z,i} \Psi_i(\theta)$. Similarly, $\rho(\theta)$, $\mu(\theta)$ and $f(\mathbf{x}, t, \theta)$ can be represented using PCEs.

The finite element discretization (e.g. [166, 167]) of the response u given by

$$u(\mathbf{x}, t, \theta) = \sum_{i=1}^n N_i(\mathbf{x}) u_i(t, \theta) = [\mathbf{N}(\mathbf{x})] \mathbf{u}(t, \theta) . \quad (6.5)$$

Thus a discrete representation of the PDE in Eq. (6.2) takes the form of the following matrix differential equation [166, 167]

$$\mathbf{M}(\theta) \ddot{\mathbf{u}}(t, \theta) + \mathbf{C}(\theta) \dot{\mathbf{u}}(t, \theta) + \mathbf{K}(\theta) \mathbf{u}(t, \theta) = \mathbf{f}(t, \theta) . \quad (6.6)$$

The elements of the mass, damping and stiffness matrices are given by [166, 167]

$$[\mathbf{M}]_{ij}(\theta) = \int_{\Omega} N_i(\mathbf{x}) \rho(\theta) N_j(\mathbf{x}) d\mathbf{x} \quad (6.7)$$

$$[\mathbf{C}]_{ij}(\theta) = \int_{\Omega} N_i(\mathbf{x}) \mu(\theta) N_j(\mathbf{x}) d\mathbf{x} \quad (6.8)$$

$$[\mathbf{K}]_{ij}(\theta) = \int_{\Omega} [\nabla N_i(\mathbf{x})]^T \boldsymbol{\kappa}(\theta) [\nabla N_j(\mathbf{x})] d\mathbf{x} \quad (6.9)$$

and the forcing vector is given by [166, 167]

$$\mathbf{f} = \int_{\Omega} [\mathbf{N}(\mathbf{x})]^T f(\mathbf{x}, t, \theta) d\mathbf{x} + \text{contributions from boundary conditions} , \quad (6.10)$$

where

$$\boldsymbol{\kappa}(\theta) = \text{diag}(k_x(\theta), k_y(\theta), k_z(\theta)) , \quad (6.11)$$

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} . \quad (6.12)$$

The above dynamic equation can discretized in time (see Appendix B for details) to obtain the following linear system of equations that can be solved for \mathbf{x} at every discrete time step

$$\mathbf{A}(\theta) \mathbf{x}(\theta) = \mathbf{f}(\theta) \quad (6.13)$$

in which \mathbf{x} is the state vector given by

$$\mathbf{x} = \begin{Bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{Bmatrix}. \quad (6.14)$$

For large-scale systems, a class of domain decomposition algorithms are developed [82–87, 95] that exploit high performance computing platforms. The basic concepts of these domain decomposition algorithms are summarized next. For large-scale systems, the spatial computational domain is divided by a number of disjoint (non-overlapping) subdomains (see Fig. 6.1). The following system of equations is obtained for a typical subdomain s

$$\begin{bmatrix} \mathbf{A}_{II}^s(\theta) & \mathbf{A}_{IG}^s(\theta) \\ \mathbf{A}_{GI}^s(\theta) & \mathbf{A}_{GG}^s(\theta) \end{bmatrix} \begin{Bmatrix} \mathbf{x}_I^s(\theta) \\ \mathbf{x}_G^s(\theta) \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_I^s(\theta) \\ \mathbf{f}_G^s(\theta) \end{Bmatrix}, \quad (6.15)$$

where the output vector $\mathbf{x}^s(\theta)$ is partitioned into the interior unknowns $\mathbf{x}_I^s(\theta)$ and the interface unknowns \mathbf{x}_G^s . The local interface vector \mathbf{x}_G^s can be mapped to the unknown global interface vector \mathbf{x}_G by $\mathbf{x}_G^s = \mathbf{R}_s \mathbf{x}_G$ where \mathbf{R}_s is the subdomain specific restriction matrix.

The random system matrix, input and output for each subdomain are first represented

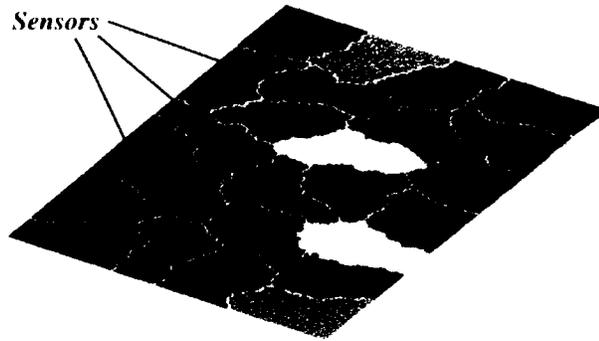


Figure 6.1: Partitioned finite element mesh.

using PCE resulting in the following system of equations for subdomain s

$$\left(\sum_{i=0}^M \begin{bmatrix} \mathbf{A}_{II,i}^s & \mathbf{A}_{IG,i}^s \\ \mathbf{A}_{GI,i}^s & \mathbf{A}_{GG,i}^s \end{bmatrix} \Psi_i(\theta) \right) \left(\sum_{j=0}^N \begin{Bmatrix} \mathbf{x}_{I,j}^s \\ \mathbf{x}_{G,j}^s \end{Bmatrix} \Psi_j(\theta) \right) \approx \sum_{l=0}^P \begin{Bmatrix} \mathbf{f}_{I,l}^s \\ \mathbf{f}_{G,l}^s \end{Bmatrix} \Psi_l(\theta) \quad (6.16)$$

Grouping all the subdomain equations and using the relationships $\mathbf{x}_I^s = \mathbf{R}_s \mathbf{x}_G$ and $\mathbf{f}_G^s = \mathbf{R}_s \mathbf{f}_G$, one obtains

$$\left(\sum_{i=0}^M \begin{bmatrix} \mathbf{A}_{II,i}^1 & \dots & 0 & \mathbf{A}_{IG,i}^1 \mathbf{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathbf{A}_{II,i}^{n_s} & \mathbf{A}_{IG,i}^{n_s} \mathbf{R}_{n_s} \\ \mathbf{R}_1^T \mathbf{A}_{GI,i}^1 & \dots & \mathbf{R}_{n_s}^T \mathbf{A}_{GI,i}^{n_s} & \sum_{s=1}^{n_s} \mathbf{R}_s^T \mathbf{A}_{GI,i}^s \mathbf{R}_s \end{bmatrix} \Psi_i(\theta) \right) \left(\sum_{j=0}^N \begin{Bmatrix} \mathbf{x}_{I,j}^1 \\ \vdots \\ \mathbf{x}_{I,j}^{n_s} \\ \mathbf{x}_{G,j} \end{Bmatrix} \Psi_j(\theta) \right) \approx \sum_{l=0}^P \begin{Bmatrix} \mathbf{f}_{I,l}^1 \\ \vdots \\ \mathbf{f}_{I,l}^{n_s} \\ \sum_{s=1}^{n_s} \mathbf{R}_s^T \mathbf{f}_{G,l}^s \end{Bmatrix} \Psi_l(\theta). \quad (6.17)$$

Performing a Galerkin projection of the above equation onto the PCE basis used to

represent the output process, the following linear system is obtained

$$\left\langle \left(\sum_{i=0}^M \begin{bmatrix} \mathbf{A}_{II,i}^1 & \dots & 0 & \mathbf{A}_{I\Gamma,i}^1 \mathbf{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathbf{A}_{II,i}^{n_s} & \mathbf{A}_{I\Gamma,i}^{n_s} \mathbf{R}_{n_s} \\ \mathbf{R}_1^T \mathbf{A}_{I\Gamma,i}^1 & \dots & \mathbf{R}_{n_s}^T \mathbf{A}_{I\Gamma,i}^{n_s} & \sum_{s=1}^{n_s} \mathbf{R}_s^T \mathbf{A}_{I\Gamma,i}^s \mathbf{R}_s \end{bmatrix} \Psi_i(\theta) \right) \left(\sum_{j=0}^N \begin{bmatrix} \mathbf{x}_{I,j}^1 \\ \vdots \\ \mathbf{x}_{I,j}^{n_s} \\ \mathbf{x}_{\Gamma,j} \end{bmatrix} \Psi_j(\theta) \right) \Psi_k(\theta) \right\rangle =$$

$$\left\langle \left(\sum_{l=0}^P \begin{bmatrix} \mathbf{f}_{I,l}^1 \\ \vdots \\ \mathbf{f}_{I,l}^{n_s} \\ \sum_{s=1}^{n_s} \mathbf{R}_s^T \mathbf{f}_{\Gamma,l}^s \end{bmatrix} \Psi_l(\theta) \right) \Psi_k(\theta) \right\rangle, \quad \text{for } k = 1, \dots, N \quad (6.18)$$

In concise notation, the above equation can be expressed as

$$\begin{bmatrix} \mathcal{A}_{II}^1 & \dots & 0 & \mathcal{A}_{I\Gamma}^1 \mathcal{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathcal{A}_{II}^{n_s} & \mathcal{A}_{I\Gamma}^{n_s} \mathcal{R}_{n_s} \\ \mathcal{R}_1^T \mathcal{A}_{I\Gamma}^1 & \dots & \mathcal{R}_{n_s}^T \mathcal{A}_{I\Gamma}^{n_s} & \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{A}_{I\Gamma}^s \mathcal{R}_s \end{bmatrix} \begin{bmatrix} \mathcal{X}_I^1 \\ \vdots \\ \mathcal{X}_I^{n_s} \\ \mathcal{X}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathcal{F}_I^1 \\ \vdots \\ \mathcal{F}_I^{n_s} \\ \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{F}_\Gamma^s \end{bmatrix}, \quad (6.19)$$

where

$$[\mathcal{A}_{II}^s]_{jk} = \sum_{i=0}^M \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_{II,i}^s, \quad [\mathcal{A}_{I\Gamma}^s]_{jk} = \sum_{i=0}^M \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_{I\Gamma,i}^s$$

$$[\mathcal{A}_{\Gamma I}^s]_{jk} = \sum_{i=0}^M \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_{\Gamma I,i}^s, \quad [\mathcal{A}_{\Gamma\Gamma}^s]_{jk} = \sum_{i=0}^M \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_{\Gamma\Gamma,i}^s$$

and

$$\begin{aligned}\mathcal{X}_I^s &= \{ \mathbf{x}_{I,0}^{s\top}, \dots, \mathbf{x}_{I,N}^{s\top} \}^\top, & \mathcal{F}_I^s &= \{ \mathbf{f}_{I,0}^{s\top} \langle \Psi_0^2 \rangle, \dots, \mathbf{f}_{I,N}^{s\top} \langle \Psi_N^2 \rangle \}^\top, \\ \mathcal{X}_\Gamma &= \{ \mathbf{x}_{\Gamma,0}^\top, \dots, \mathbf{x}_{\Gamma,N}^\top \}^\top, & \mathcal{F}_\Gamma^s &= \{ \mathbf{f}_{\Gamma,0}^{s\top} \langle \Psi_0^2 \rangle, \dots, \mathbf{f}_{\Gamma,N}^{s\top} \langle \Psi_N^2 \rangle \}^\top, \\ \mathcal{R}_s &= \mathbf{I}_{N+1} \otimes \mathbf{R}_s,\end{aligned}$$

with \otimes denoting the Kronecker matrix product [171] and \mathbf{I}_{N+1} the identity matrix of dimension $\{N + 1\}$.

Consequently, the following extended Schur complement [172] system for the interface variable \mathcal{X}_Γ can be obtained using Gaussian elimination as

$$\mathcal{S} \mathcal{X}_\Gamma = \mathcal{G}_\Gamma, \quad (6.20)$$

where \mathcal{X}_Γ is the vector containing the polynomial chaos coefficients of the interface nodes. The global extended Schur complement matrix \mathcal{S} is given by

$$\mathcal{S} = \sum_{s=1}^{n_s} \mathcal{R}_s^\top [\mathcal{A}_{\Gamma\Gamma}^s - \mathcal{A}_{\Gamma I}^s (\mathcal{A}_{II}^s)^{-1} \mathcal{A}_{I\Gamma}^s] \mathcal{R}_s, \quad (6.21)$$

and the corresponding right hand side vector \mathcal{G}_Γ is defined by

$$\mathcal{G}_\Gamma = \sum_{s=1}^{n_s} \mathcal{R}_s^\top [\mathcal{F}_\Gamma^s - \mathcal{A}_{\Gamma I}^s (\mathcal{A}_{II}^s)^{-1} \mathcal{F}_I^s]. \quad (6.22)$$

In the context of large-scale systems, iterative methods are generally utilized to solve the extended Schur complement system in Eq. (6.20) as direct solvers can no longer be used due to large memory requirements and poor scalability. The use of iterative methods to solve the extended Schur complement system resulting from domain decomposi-

tion is referred to as iterative substructuring [173]. A newly developed iterative solver based on preconditioned Conjugate Gradient Method (PCGM) [174] for the stochastic extended Schur complement system is used in this thesis for the case of symmetric system matrices [83–87, 95]. Its parallel implementation is outlined in Appendix D. For the cases involving non-symmetric system matrices, the author developed a new parallel solver for stochastic systems based on the preconditioned Biconjugate Gradient Stabilized (BICGSTAB) method [174] as described in Appendix D.4.

The major drawback of iterative solvers for the extended Schur complement system is the lack of robustness due to slow convergence [174]. The use of efficient preconditioners alleviates such issue by transforming the original system to a new one having better condition number as [172]

$$\mathcal{M}^{-1} \mathcal{S} \mathcal{X}_r = \mathcal{M}^{-1} \mathcal{G}_r, \quad (6.23)$$

where \mathcal{M}^{-1} is the preconditioning matrix. In this context, the preconditioning is performed from the left, but in general preconditioning could also be performed from the right or from both sides (see [174] for more details). In this thesis, the one-level lumped preconditioner is used, detailed in Appendix D.1.

6.3 Polynomial Chaos Based Ensemble Kalman Filter

Once (sparse and noisy) observations of the physical system become available at a specific time instant, the polynomial chaos coefficients of the system response can be updated to reflect the information available in observational data. This can simply be performed using a linear analysis step borrowed from the Kalman filter (see Section 2.3). The ensemble Kalman filter (see Section 2.6) propagates an ensemble of model states

forward in time until an observation becomes available. EnKF then assimilates the observation vector \mathbf{d} using a linear update equation of KF. In the polynomial chaos based Kalman filter (PCKF) [80], the same linear analysis step is used to update the polynomial chaos coefficients of the solution process.

The polynomial chaos representation of the forecast state vector of a dynamical system prior to assimilating the available data at a given time instant is given by

$$\mathbf{x}^f = \sum_{i=0}^N \mathbf{x}_i^f \Psi_i(\theta) . \quad (6.24)$$

The polynomial chaos coefficients can be grouped into a coefficient matrix as

$$\mathbb{A}^f = [\mathbf{x}_0^f, \mathbf{x}_1^f, \dots, \mathbf{x}_N^f] \quad (6.25)$$

where $\mathbb{A}^f \in \mathbb{R}^{n \times (N+1)}$, n is the dimension of the state vector and $(N + 1)$ is the number of polynomial chaos coefficients in the expansion. The state error covariance matrix

prior to assimilating the data is thus [80]

$$\begin{aligned}
\mathbf{P}^f &= \left\langle (\mathbf{x}^f - \bar{\mathbf{x}}^f) (\mathbf{x}^f - \bar{\mathbf{x}}^f)^T \right\rangle \\
&= \left\langle \left(\left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i(\theta) \right] - \mathbf{x}_0^f \right) \left(\left[\sum_{j=0}^N \mathbf{x}_j^f \Psi_j(\theta) \right] - \mathbf{x}_0^f \right)^T \right\rangle \\
&= \left\langle \left(\sum_{i=1}^N \mathbf{x}_i^f \Psi_i(\theta) \right) \left(\sum_{j=1}^N \mathbf{x}_j^f \Psi_j(\theta) \right)^T \right\rangle \\
&= \left\langle \left(\sum_{i=1}^N \mathbf{x}_i^f \Psi_i(\theta) \right) \left(\sum_{j=1}^N \mathbf{x}_j^{fT} \Psi_j(\theta) \right) \right\rangle \\
&= \left\langle \sum_{i=1}^N \sum_{j=1}^N \mathbf{x}_i^f \Psi_i(\theta) \mathbf{x}_j^{fT} \Psi_j(\theta) \right\rangle \\
&= \sum_{i=1}^N \sum_{j=1}^N \left\langle \mathbf{x}_i^f \Psi_i(\theta) \mathbf{x}_j^{fT} \Psi_j(\theta) \right\rangle \\
&= \sum_{i=1}^N \sum_{j=1}^N \mathbf{x}_i^f \mathbf{x}_j^{fT} \langle \Psi_i(\theta) \Psi_j(\theta) \rangle \\
&= \sum_{i=1}^N \sum_{j=1}^N \mathbf{x}_i^f \mathbf{x}_j^{fT} \delta_{ij} \langle \Psi_j^2(\theta) \rangle \\
&= \sum_{i=1}^N \mathbf{x}_i^f \mathbf{x}_i^{fT} \langle \Psi_i^2(\theta) \rangle .
\end{aligned} \tag{6.26}$$

The available observations are represented by a random vector of dimension m whose polynomial chaos expansion is given by

$$\mathbf{d} = \sum_{i=0}^N \mathbf{d}_i \Psi_i(\theta) \tag{6.27}$$

where the mean \mathbf{d}_0 contains the actual measurements and higher order terms $\{\mathbf{d}_i\}_{i=1}^N$ represent the measurement error. The coefficients can be stored into another coefficient matrix

$$\mathbb{D} = [\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_N] \tag{6.28}$$

where $\mathbb{D} \in \mathbb{R}^{m \times (N+1)}$. A similar derivation to Eq. (6.26) could be used to provide the following expression for the measurement error covariance matrix:

$$\Gamma = \sum_{i=1}^N \mathbf{d}_i \mathbf{d}_i^T \langle \Psi_i^2 \rangle. \quad (6.29)$$

The PC coefficients of the state vector is updated according to the linear analysis step of the Kalman filter as (see Eq. (2.42))

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{K} (\mathbf{d} - \mathbf{H} \mathbf{x}^f) \quad (6.30)$$

in which \mathbf{K} is the Kalman gain matrix and \mathbf{H} is measurement matrix that relates the measurement vector to the state vector (see Section 2.3).

The state and measurement vectors in Eq. (6.30) are replaced by their truncated polynomial chaos expansions, i.e. Eq. (6.24) and Eq. (6.27) respectively as [175]

$$\sum_{i=0}^N \mathbf{x}_i^a \Psi_i(\theta) = \sum_{i=0}^N \mathbf{x}_i^f \Psi_i(\theta) + \mathbf{K} \left(\sum_{i=0}^N \mathbf{d}_i \Psi_i(\theta) - \mathbf{H} \sum_{i=0}^N \mathbf{x}_i^f \Psi_i(\theta) \right). \quad (6.31)$$

Subsequently, a Galerkin projection is performed on both sides of the above equation onto the polynomial chaos $\Psi_j(\theta)$ used to represent the state and measurement vectors. For the following derivation, θ is omitted for brevity. Performing this projection onto

the left hand side first:

$$\begin{aligned}
\left\langle \left[\sum_{i=0}^N \mathbf{x}_i^a \Psi_i \right] \Psi_j \right\rangle &= \left\langle \sum_{i=0}^N \mathbf{x}_i^a \Psi_i \Psi_j \right\rangle \\
&= \sum_{i=0}^N \langle \mathbf{x}_i^a \Psi_i \Psi_j \rangle \\
&= \sum_{i=0}^N \mathbf{x}_i^a \langle \Psi_i \Psi_j \rangle \\
&= \sum_{i=0}^N \mathbf{x}_i^a \delta_{ij} \langle \Psi_j^2 \rangle \\
&= \mathbf{x}_j^a \langle \Psi_j^2 \rangle .
\end{aligned} \tag{6.32}$$

Projecting the right hand side of Eq. (6.31) onto Ψ_j leads to

$$\begin{aligned}
&\left\langle \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i + \mathbf{K} \left(\sum_{i=0}^N \mathbf{d}_i \Psi_i - \mathbf{H} \sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right) \right] \Psi_j \right\rangle \\
&= \left\langle \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right] \Psi_j + \mathbf{K} \left(\left[\sum_{i=0}^N \mathbf{d}_i \Psi_i \right] \Psi_j - \mathbf{H} \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right] \Psi_j \right) \right\rangle \\
&= \left\langle \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right] \Psi_j \right\rangle + \left\langle \mathbf{K} \left(\left[\sum_{i=0}^N \mathbf{d}_i \Psi_i \right] \Psi_j - \mathbf{H} \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right] \Psi_j \right) \right\rangle \\
&= \left\langle \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right] \Psi_j \right\rangle + \mathbf{K} \left(\left\langle \left[\sum_{i=0}^N \mathbf{d}_i \Psi_i \right] \Psi_j \right\rangle - \mathbf{H} \left\langle \left[\sum_{i=0}^N \mathbf{x}_i^f \Psi_i \right] \Psi_j \right\rangle \right) \\
&= \sum_{i=0}^N \langle \mathbf{x}_i^f \Psi_i \Psi_j \rangle + \mathbf{K} \left(\sum_{i=0}^N \langle \mathbf{d}_i \Psi_i \Psi_j \rangle - \mathbf{H} \sum_{i=0}^N \langle \mathbf{x}_i^f \Psi_i \Psi_j \rangle \right) \\
&= \sum_{i=0}^N \mathbf{x}_i^f \langle \Psi_i \Psi_j \rangle + \mathbf{K} \left(\sum_{i=0}^N \mathbf{d}_i \langle \Psi_i \Psi_j \rangle - \mathbf{H} \sum_{i=0}^N \mathbf{x}_i^f \langle \Psi_i \Psi_j \rangle \right) \\
&= \sum_{i=0}^N \mathbf{x}_i^f \delta_{ij} \langle \Psi_j^2 \rangle + \mathbf{K} \left(\sum_{i=0}^N \mathbf{d}_i \delta_{ij} \langle \Psi_j^2 \rangle - \mathbf{H} \sum_{i=0}^N \mathbf{x}_i^f \delta_{ij} \langle \Psi_j^2 \rangle \right) \\
&= \mathbf{x}_j^f \langle \Psi_j^2 \rangle + \mathbf{K} \left(\mathbf{d}_j \langle \Psi_j^2 \rangle - \mathbf{H} \mathbf{x}_j^f \langle \Psi_j^2 \rangle \right) \\
&= \left[\mathbf{x}_j^f + \mathbf{K} \left(\mathbf{d}_j - \mathbf{H} \mathbf{x}_j^f \right) \right] \langle \Psi_j^2 \rangle .
\end{aligned} \tag{6.33}$$

Combining Eq. (6.32) and Eq. (6.33) and eliminating $\langle \Psi_j^2 \rangle$ from both sides of the equality provides the analysis step involving the j th PCE coefficient of the state vector:

$$\mathbf{x}_j^a = \mathbf{x}_j^f + \mathbf{K} \left(\mathbf{d}_j - \mathbf{H} \mathbf{x}_j^f \right) . \quad (6.34)$$

The above result can be rewritten in matrix form as

$$\mathbb{A}^a = \mathbb{A}^f + \mathbf{K} \left(\mathbb{D} - \mathbf{H} \mathbb{A}^f \right) . \quad (6.35)$$

Finally, utilizing the expression for the Kalman gain matrix \mathbf{K} given in Eq. (2.41), the following assimilation step for PCKF is obtained [80, 175]

$$\mathbb{A}^a = \mathbb{A}^f + \mathbf{P}^f \mathbf{H}^T \left(\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \Gamma \right)^{-1} \left(\mathbb{D} - \mathbf{H} \mathbb{A}^f \right) . \quad (6.36)$$

The aforementioned polynomial chaos based domain-decomposition-based iterative solver developed in [83–87] is used to propagate the PC coefficients of the state vector in time, providing an efficient forecast step. The analysis step, whenever measurements are available, will make use of the available polynomial decomposition of the forecast state estimate to efficiently perform the update of the PC coefficients [80] using a parallel implementation as explained next.

6.3.1 Distributed implementation of the PCKF Analysis Step

The forecast step performed using the polynomial chaos based domain decomposition algorithm will naturally enhance the computational efficiency of the filtering algorithm compared to the sampling based filters (EnKF and PF). To increase the computational efficacy further, a parallel implementation of the analysis step of PCKF [80] is developed

utilizing the aforementioned polynomial chaos based domain decomposition solver [82–87]. In order to achieve this, the forecast error covariance matrix in Eq. (6.26) is first re-expressed in terms of the coefficient matrix \mathbb{A}^f as

$$\begin{aligned}
\mathbf{P}^f &= \sum_{i=1}^N \mathbf{x}_i^f \mathbf{x}_i^{f\top} \langle \Psi_i^2 \rangle \\
&= \begin{bmatrix} \mathbf{x}_1^f & \mathbf{x}_2^f & \dots & \mathbf{x}_N^f \end{bmatrix} \begin{bmatrix} \langle \Psi_1^2 \rangle & 0 & \dots & 0 \\ 0 & \langle \Psi_2^2 \rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle \Psi_N^2 \rangle \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^f & \mathbf{x}_2^f & \dots & \mathbf{x}_N^f \end{bmatrix}^\top \\
&= \begin{bmatrix} \mathbf{x}_0^f & \mathbf{x}_1^f & \dots & \mathbf{x}_N^f \end{bmatrix} \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \langle \Psi_1^2 \rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle \Psi_N^2 \rangle \end{bmatrix} \begin{bmatrix} \mathbf{x}_0^f & \mathbf{x}_1^f & \dots & \mathbf{x}_N^f \end{bmatrix}^\top \\
&= \mathbb{A}^f \mathbf{B} \mathbb{A}^{f\top} .
\end{aligned} \tag{6.37}$$

Similarly, the measurement error covariance matrix in Eq. (6.29) can be rewritten as

$$\mathbf{\Gamma} = \mathbb{D} \mathbf{B} \mathbb{D}^\top . \tag{6.38}$$

Using the above expressions in Eqs. (6.37)-(6.38), the analysis step of PCKF in Eq. (6.36) can be rewritten as

$$\begin{aligned}
\mathbb{A}^a &= \mathbb{A}^f + \mathbf{P}^f \mathbf{H}^\top (\mathbf{H} \mathbf{P}^f \mathbf{H}^\top + \mathbf{\Gamma})^{-1} (\mathbb{D} - \mathbf{H} \mathbb{A}^f) \\
&= \mathbb{A}^f + \mathbb{A}^f \mathbf{B} \mathbb{A}^{f\top} \mathbf{H}^\top (\mathbf{H} \mathbb{A}^f \mathbf{B} \mathbb{A}^{f\top} \mathbf{H}^\top + \mathbb{D} \mathbf{B} \mathbb{D}^\top)^{-1} (\mathbb{D} - \mathbf{H} \mathbb{A}^f) \\
&= \mathbb{A}^f + \mathbb{A}^f \mathbf{B} [\mathbf{H} \mathbb{A}^f]^\top \left([\mathbf{H} \mathbb{A}^f] \mathbf{B} [\mathbf{H} \mathbb{A}^f]^\top + \mathbb{D} \mathbf{B} \mathbb{D}^\top \right)^{-1} (\mathbb{D} - [\mathbf{H} \mathbb{A}^f]) .
\end{aligned} \tag{6.39}$$

The result of the matrix multiplication $\mathbf{H}\mathbf{A}^f$ represents the ensemble of PC coefficients of the *forecast*, or predicted observations. Denoting this by \mathbb{D}^f , one can express the analysis step of the filter as

$$\begin{aligned}\mathbf{A}^a &= \mathbf{A}^f + \mathbf{A}^f \mathbf{B} \mathbb{D}^f \mathbf{T} \left(\mathbb{D}^f \mathbf{B} \mathbb{D}^f \mathbf{T} + \mathbb{D} \mathbf{B} \mathbb{D} \mathbf{T} \right)^{-1} (\mathbb{D} - \mathbb{D}^f) \\ &= \mathbf{A}^f \left[\mathbf{I} + \mathbf{B} \mathbb{D}^f \mathbf{T} \left(\mathbb{D}^f \mathbf{B} \mathbb{D}^f \mathbf{T} + \mathbb{D} \mathbf{B} \mathbb{D} \mathbf{T} \right)^{-1} (\mathbb{D} - \mathbb{D}^f) \right] \\ &= \mathbf{A}^f \mathbf{C} .\end{aligned}\tag{6.40}$$

Eq. (6.40) indicates that the update step of PCKF involves a post-multiplication by a matrix \mathbf{C} . This matrix is a function of the observation matrix, \mathbb{D} , and the forecast observation matrix, $\mathbb{D}^f = \mathbf{H}\mathbf{A}^f$. \mathbf{H} is normally a sparse matrix that interpolates the response at the observation locations from the response at the finite element nodal points. For the case of a two-dimensional domain discretized using linear finite elements, the matrix \mathbf{H} has at most three non-zero elements per row i , obtained by evaluating the linear basis functions at the spatial coordinates for the i th observation. For non-overlapping domain decomposition methods [174], each observation lies exclusively in one of the n_s subdomains. In this case, the observation matrix can be partitioned into n_s local observation matrices, each acting on their respective local interior unknowns, and an interface observation matrix acting on the global interface unknowns:

$$\mathbf{H} = [\mathbf{H}_I^1, \dots, \mathbf{H}_I^2, \mathbf{H}_I^{n_s}, \mathbf{H}_I] .\tag{6.41}$$

One can also exploit the available spatial decomposition of the PC coefficients per-

taining to the forecast state vector in the form of

$$\mathbf{x}^f(\theta) = \sum_{i=0}^N \begin{bmatrix} \{\mathbf{x}^f\}_{I,i}^1 \\ \vdots \\ \{\mathbf{x}^f\}_{I,i}^{n_s} \\ \{\mathbf{x}^f\}_{\Gamma,i} \end{bmatrix} \Psi_i(\theta), \quad (6.42)$$

or in matrix form as

$$\mathbb{A}^f = \begin{bmatrix} \{\mathbf{x}^f\}_{I,0}^1 & \{\mathbf{x}^f\}_{I,1}^1 & \cdots & \{\mathbf{x}^f\}_{I,N}^1 \\ \{\mathbf{x}^f\}_{I,0}^2 & \{\mathbf{x}^f\}_{I,1}^2 & \cdots & \{\mathbf{x}^f\}_{I,N}^2 \\ \vdots & \vdots & & \vdots \\ \{\mathbf{x}^f\}_{I,0}^{n_s} & \{\mathbf{x}^f\}_{I,1}^{n_s} & \cdots & \{\mathbf{x}^f\}_{I,N}^{n_s} \\ \{\mathbf{x}^f\}_{\Gamma,0} & \{\mathbf{x}^f\}_{\Gamma,1} & \cdots & \{\mathbf{x}^f\}_{\Gamma,N} \end{bmatrix} = \begin{bmatrix} [\mathbb{A}^f]_I^1 \\ [\mathbb{A}^f]_I^2 \\ \vdots \\ [\mathbb{A}^f]_I^{n_s} \\ [\mathbb{A}^f]_\Gamma \end{bmatrix}, \quad (6.43)$$

where $[\mathbb{A}^f]_I^s$ denotes the ensemble of PC coefficients for the interior nodes belonging to the subdomain s and $[\mathbb{A}^f]_\Gamma$ is the ensemble of PC coefficients for the global interface nodes. Eqs. (6.41)-(6.43) can be used to provide a decomposition of the forecast

observation matrix $\mathbb{D}^f = \mathbf{H}\mathbf{A}^f$ in the form

$$\begin{aligned}
\mathbb{D}^f &= \mathbf{H}\mathbf{A}^f \\
&= [\mathbf{H}_I^1, \dots, \mathbf{H}_I^2, \mathbf{H}_I^{n_s}, \mathbf{H}_R] \begin{bmatrix} [\mathbf{A}^f]_I^1 \\ [\mathbf{A}^f]_I^2 \\ \vdots \\ [\mathbf{A}^f]_I^{n_s} \\ [\mathbf{A}^f]_R \end{bmatrix} \\
&= \mathbf{H}_I^1 [\mathbf{A}^f]_I^1 + \mathbf{H}_I^2 [\mathbf{A}^f]_I^2 + \dots + \mathbf{H}_I^{n_s} [\mathbf{A}^f]_I^{n_s} + \mathbf{H}_R [\mathbf{A}^f]_R \\
&= [\mathbb{D}^f]_I^1 + [\mathbb{D}^f]_I^2 + \dots + [\mathbb{D}^f]_I^{n_s} + [\mathbb{D}^f]_R \\
&= [\mathbb{D}^f]_R + \sum_{s=1}^{n_s} [\mathbb{D}^f]_I^s .
\end{aligned} \tag{6.44}$$

By construction, the head compute node has a copy of the global interface unknowns in its accessible memory design for the parallel iterative solvers discussed in Section 6.2. Thus, the contribution of the interface nodes to the forecast observation matrix, i.e. the term $[\mathbb{D}^f]_R$, can be obtained at the head compute node. As evident from Eq. (6.44), the contribution of each subdomain to the predicted observation matrix \mathbb{D}^f can be constructed locally. Subsequently, the summation can be performed using a gather operation providing \mathbb{D}^f at the head node. For large-scale systems, the number of measurements may be much less than the number of elements in the state vector for certain applications, i.e. $n_m \ll n$. Thus, this step requires little inter-processor communication between the head node and the compute nodes.

Since the measurements are likely to be read by the head processing node, the observation matrix \mathbb{D} can be easily constructed at the head node in its entirety without the need for interprocessor communication. The availability of \mathbb{D} and \mathbb{D}^f at the head com-

pute node allows for the evaluation of the matrix \mathbf{C} at the head node using Eq. (6.40) as

$$\mathbf{C} = \mathbf{I} + \mathbf{B} \mathbb{D}^f \mathbb{T} \left(\mathbb{D}^f \mathbf{B} \mathbb{D}^f \mathbb{T} + \mathbb{D} \mathbf{B} \mathbb{D} \mathbb{T} \right)^{-1} (\mathbb{D} - \mathbb{D}^f) . \quad (6.45)$$

Once matrix \mathbf{C} is obtained, it is made available in the local memory of all computing nodes using a broadcast operation. What remains to be computed is the product $\mathbb{A}^f \mathbf{C}$ resulting in the analysis ensemble \mathbb{A}^a . This step would dominate the time required for the analysis step for cases in which the dimension of the state vector dominates the number of measurements. The available spatial decomposition of the PC coefficients in Eq. (6.43) will also be a key ingredient in developing an effective parallel implementation of this step. The ensemble update step in Eq. (6.40) can now be performed in

parallel as

$$\begin{aligned}
 \mathbf{A}^a &= \mathbf{A}^f \mathbf{C} \\
 &= \begin{bmatrix} [\mathbf{A}^f]_I^1 \\ [\mathbf{A}^f]_I^2 \\ \vdots \\ [\mathbf{A}^f]_I^{n_s} \\ [\mathbf{A}^f]_R \end{bmatrix} \mathbf{C} \\
 &= \begin{bmatrix} [\mathbf{A}^f]_I^1 \mathbf{C} \\ [\mathbf{A}^f]_I^2 \mathbf{C} \\ \vdots \\ [\mathbf{A}^f]_I^{n_s} \mathbf{C} \\ [\mathbf{A}^f]_R \mathbf{C} \end{bmatrix} \\
 &= \begin{bmatrix} [\mathbf{A}^a]_I^1 \\ [\mathbf{A}^a]_I^2 \\ \vdots \\ [\mathbf{A}^a]_I^{n_s} \\ [\mathbf{A}^a]_R \end{bmatrix}, \tag{6.46}
 \end{aligned}$$

in which each compute node updates the local PC coefficients available in local memory, thus eliminating the need for inter-processor communication for this step.

The parallel algorithm for the distributed implementation of the polynomial chaos-based Kalman filter outlined above is provided in Algorithm 6.1. In the next section, the parallel scalability of this algorithm is investigated on high performance computing platforms.

<p>Input: Measurement matrix: \mathbb{D}</p> <p>Input: Distributed forecast ensemble: $[\mathbb{A}^f]_r$ and $[\mathbb{A}^f]_I^s$ for $s = 1, 2, \dots, n_s$</p> <p>Output: Distributed analysis ensemble: $[\mathbb{A}^a]_r$ and $[\mathbb{A}^a]_I^s$ for $s = 1, 2, \dots, n_s$</p> <p>1 for each processor $P_s, s = 1, 2, \dots, n_s$, do:</p> <p>2 Compute locally the forecast observation matrix: $[\mathbb{D}^f]_I^s = \mathbf{H}_I^s [\mathbb{A}^f]_I^s$</p> <p>3 Gather: $\mathbf{Y} = \sum_{s=1}^{n_s} [\mathbb{D}^f]_I^s$</p> <p>4 end do</p> <p>5 $\mathbf{Z} = \mathbf{H}_r [\mathbb{A}^f]_r$</p> <p>6 $\mathbb{D}^f = \mathbf{Y} + \mathbf{Z}$</p> <p>7 $\mathbf{C} = \mathbf{I} + \mathbf{B} \mathbb{D}^f \mathbf{T} \left(\mathbb{D}^f \mathbf{B} \mathbb{D}^f \mathbf{T} + \mathbb{D} \mathbf{B} \mathbb{D} \mathbf{T} \right)^{-1} (\mathbb{D} - \mathbb{D}^f)$</p> <p>8 Broadcast \mathbf{C}</p> <p>9 for each processor $P_s, s = 1, 2, \dots, n_s$, do:</p> <p>10 Update locally the PCE coefficients: $[\mathbb{A}^a]_I^s = [\mathbb{A}^f]_I^s \mathbf{C}$</p> <p>11 end do</p> <p>12 Update the interface PCE coefficients: $[\mathbb{A}^a]_r = [\mathbb{A}^f]_r \mathbf{C}$</p>

Algorithm 6.1: Parallel Polynomial Chaos based Kalman Filter.

6.4 Numerical Results

In this section, numerical results are presented for the two-dimensional stationary stochastic diffusion and advection-diffusion problems. The finite element discretization leads to a symmetric system of equations for the former problem and a non-symmetric one for the latter. The aforementioned parallel iterative solvers based on the preconditioned conjugate gradient method (see Appendix D) and the preconditioned biconjugate gradient stabilized method (see Appendix D.4) are used as forward solvers to obtain the forecast estimates and associated uncertainty. The parallel algorithm for the distributed implementation of the polynomial chaos-based Kalman filter as detailed in the previous section is used to assimilate the available data in order to obtain the posterior estimates. The absolute performance or parallel scalability of the newly developed PCKF parallel algorithm is examined. More precisely, the parallel scalability of the algorithm is investigated for three cases: (1) the global problem size (i.e. order of the state vector) and number of measurements (i.e. sensors) is fixed while the number of subdomains (i.e. central processing units or CPUs) is increased, (2) the global problem size and num-

ber of subdomains are fixed while number of measurements is increased, and (3) the number of subdomains and number of measurements are fixed while the global problem size is increased. Although the numerical investigation is restricted to two-dimensional problems, the proposed algorithm is general and can be applied to three-dimensional problems.

6.4.1 Stationary Stochastic Diffusion Problem

The stationary stochastic diffusion problem is defined by a linear elliptic stochastic partial differential equation (SPDE) as [176]

$$\begin{aligned} -\nabla \cdot (c(\mathbf{x}, \theta) \nabla u(\mathbf{x}, \theta)) &= f(\mathbf{x}) && \text{in } D \times \Omega, \\ u(\mathbf{x}, \theta) &= 0 && \text{on } \partial D \times \Omega, \end{aligned} \quad (6.47)$$

where D is the spatial domain for the boundary value problem, c is the random diffusion coefficient, f is the source or right-hand side term, u is the unknown species concentration, and Ω is a sample space belonging to the probability space (Ω, \mathcal{F}, P) in which \mathcal{F} is a σ -algebra and P is a probability measure (see [177] for details). The random diffusion coefficient is modeled as a stochastic process with given mean, variance, and covariance function. The source function f is assumed to be deterministic ($f = 1$) and thus the source of uncertainty arises only from the random diffusivity. The SSFEM discretization leads to a symmetric system matrix and thus the parallel iterative solver based on the preconditioned conjugate gradient method and domain decomposition method described in Appendix D will be used to obtain the forecast ensemble of the PCE coefficients for u . This solver is validated for a stationary stochastic diffusion problem in Appendix D.3.1 whereby an analytical solution is available.

Prior PCE of Stochastic Response

The diffusion coefficient $c(\mathbf{x}, \theta)$ is modeled as a lognormal stochastic process as

$$c(\mathbf{x}, \theta) = \exp(g(\mathbf{x}, \theta)) \quad (6.48)$$

with an underlying Gaussian process $g(\mathbf{x}, \theta)$ having the following exponential covariance function [1]

$$C(x_1, x_2; y_1, y_2) = \sigma^2 e^{-|x_1 - y_1|/b_1 - |x_2 - y_2|/b_2}, \quad (6.49)$$

where σ^2 is the variance and b_1 and b_2 are the correlations lengths along the x_1 and x_2 directions, respectively. The analytical expressions for the eigenfunctions $g_n(\mathbf{x})$ and eigenvalues λ_n of such covariance function are given in Appendix C.1.1. The numerical values used in the investigation are $\sigma = 0.3$ and $b_1 = b_2 = 50$, leading to the eigenvalue distribution given in Fig. 6.2. The first three eigenvalues account for approximately 99% of the energy of the Gaussian random process and their associated eigenfunctions are shown in Fig.(6.3). Thus, the underlying Gaussian process will be represented by the truncated KLE

$$g(\mathbf{x}, \theta) = g_0(\mathbf{x}) + \sum_{n=1}^3 \sqrt{\lambda_n} g_n(\mathbf{x}) \xi_n(\theta) \quad (6.50)$$

in which ξ_n are independent and identically distributed normal random variables and g_1 through g_3 are the eigenfunctions shown in Fig.(6.3). For this set of experiments, the mean of the underlying Gaussian process is $g_0(\mathbf{x}) = -4$. Thus the dimension of PCE is also $L = 3$ for the lognormal process. It is also determined through numerical experiments (results not shown for brevity) that a third-order PC expansion ($p = 3$) is required to sufficiently capture the non-Gaussian nature of the input. Thus the total

number of coefficients in the PC expansion is $N = 20$. The analytical expressions for PCE coefficients of the input lognormal diffusivity process in terms of the KLE eigenfunctions are provided in Appendix C.2.1.

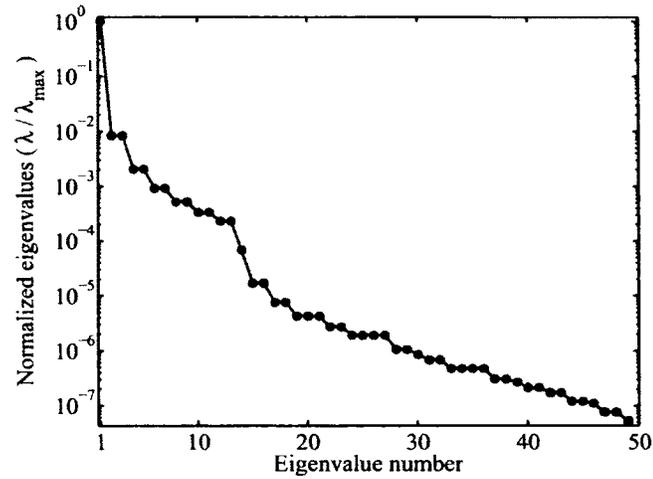


Figure 6.2: Eigenvalues of the exponential covariance function.

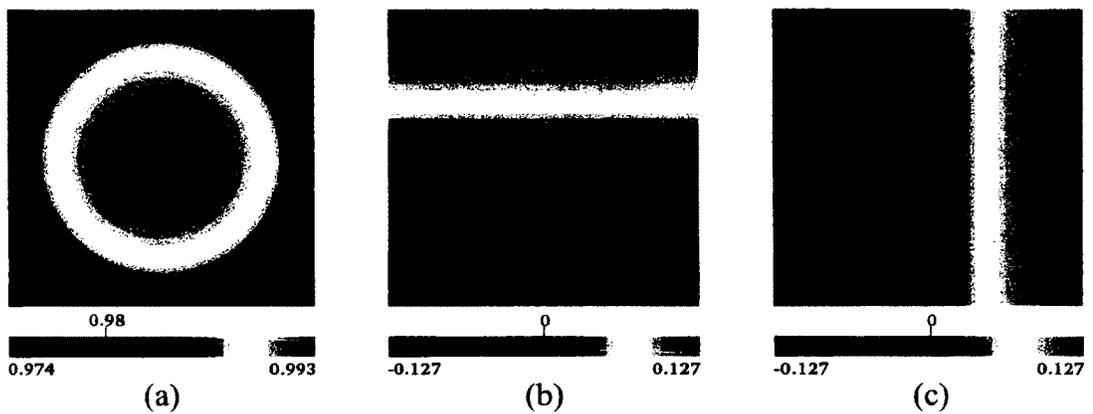


Figure 6.3: Selected eigenfunctions of the exponential covariance function: (a) eigenfunction of the largest eigenvalue, (b) eigenfunction of the second largest eigenvalue, and (c) eigenfunction of the third largest eigenvalue.

For numerical illustration, the spatial domain is discretized into a finite element mesh consisting of 100,241 linear triangular elements and 50,118 nodes as illustrated in

Fig. 6.4a. The spatial domain is decomposed into 176 subdomains as shown in Fig. 6.4b resulting in a total of 5,379 global interface nodes. For uncertainty representation, the response is represented using a three-dimensional third-order PCE (i.e. $N = 20$), resulting in a linear system of order 1,002,360. Using the parallel solver based on domain decomposition and preconditioned conjugate gradient method for the extended Schur complement system (see Appendix D for the parallel algorithm), the PC coefficients of the solution process are computed and illustrated in Fig. 6.5. The Krylov-subspace iterate is initialized to zero and the stopping criteria for the PCGM solver is

$$\frac{\| \mathcal{X}_{r_{i+1}} - \mathcal{X}_{r_i} \|}{\| \mathcal{X}_{r_i} \|} \leq \epsilon = 10^{-15} . \quad (6.51)$$

As these PCE coefficients represent the state of knowledge regarding the response prior to the assimilation of observations, these coefficients will be denoted by “prior” PCE coefficients of the response. The zeroth PCE coefficient representing the mean response is plotted again in Fig. 6.6a. The covariance matrix is a function of the PCE coefficients as in Eq. (6.26). The standard deviation of the response, computed as the square roots of the covariance matrix diagonal entries, is shown in Fig. 6.6b. At its peak mean response, the coefficient of variation (being the mean-normalized standard deviation) is approximately 30%. As evident, the mean and standard deviation of the response are zero whenever homogeneous Dirichlet boundary conditions are present. In fact, all of the PCE coefficients are zero at such boundaries as evident in Fig. 6.5.

Data Assimilation - Experiment 1

For an illustrative data assimilation experiment, a specific realization obtained from the prior PCE of the solution process is used as the “true” response from which noise-

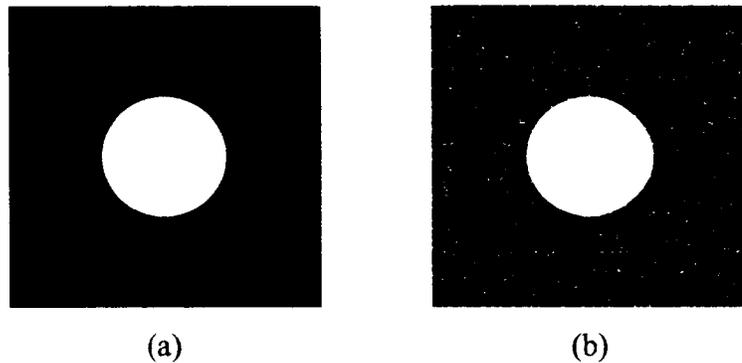


Figure 6.4: Stationary stochastic diffusion problem: (a) Finite element mesh and (b) the partitioned domain.

contaminated observations are obtained. The noise is assumed to be Gaussian in nature with zero mean and 0.1 standard deviation. For the first experiment, only one observation is made available at the location shown in Fig. 6.7. The proposed parallel data assimilation method based on domain decomposition and PCKF is used to update the prior PCE coefficients, shown in Fig. 6.5, providing the so-called posterior PCE coefficients. These posterior PCE coefficients are shown in Fig. 6.8. The number of coefficients has increased due to the assimilation of data. The sources of uncertainty in the observations, being sensor error for example, are independent from those in the forecast response (random diffusivity) and thus the dimension of the problem in the stochastic space increases when data is assimilated. Since only one measurement is available, the random dimension of the problem increases by one. Furthermore, since the error is Gaussian in nature, the PC order required to capture the uncertainty in that (observational) dimension is one. Thus, only one extra PCE coefficient is required in this case. In general, except for the mean coefficient, the PCE coefficients have been attenuated in the proximity of the sensor from which the observation data is gathered.

Fig. 6.9 shows the true response, and prior and posterior mean estimates. The posterior mean estimate, obtained using the available noisy observations, is closer to the

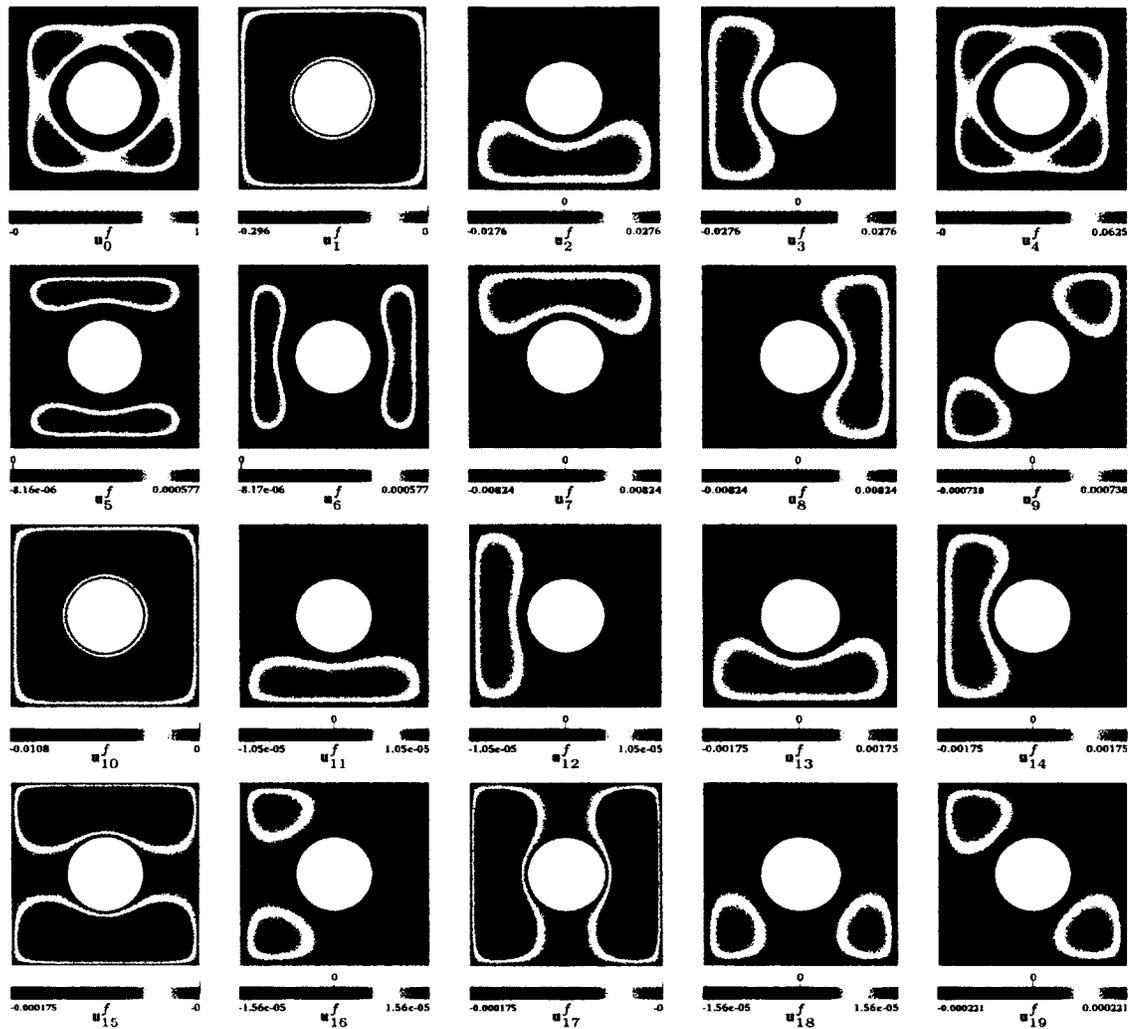


Figure 6.5: Stationary stochastic diffusion problem: Prior PCE coefficients.

true solution than the prior estimate. The assimilation of data reduces the error in the estimates as evident in Fig 6.10 in which the prior and posterior standard deviations are plotted. Not only does the standard deviation decrease globally, the spatial distribution of uncertainty changes as well: the amount of uncertainty is much lower in the proximity of the sensor, but it is higher further away from the sensor location. This is expected as the correlation decreases between the response at the sensor and locations that are further away from the sensor.

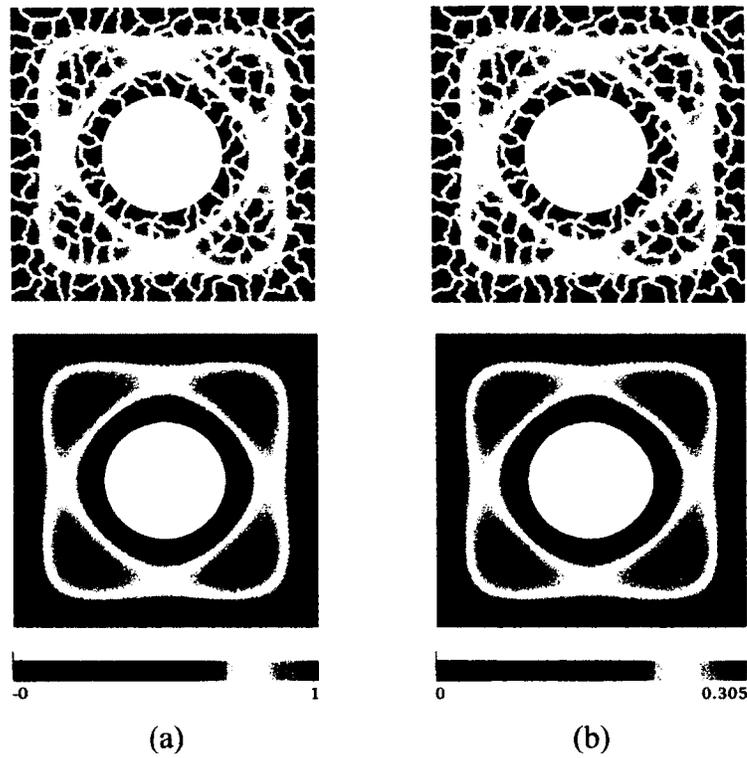


Figure 6.6: Stationary stochastic diffusion problem: (a) Prior mean response (b) standard deviation. The spatial decomposition of the mean response and standard deviation is shown in the top figures.

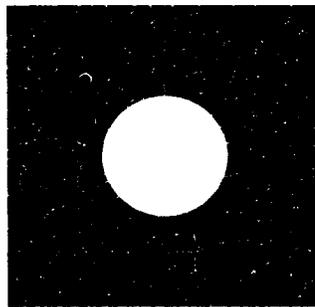


Figure 6.7: Data assimilation in stationary stochastic diffusion problem
- Experiment 1: Sensor location.

Data Assimilation - Experiment 2

In this data assimilation experiment, the number of observations available is increased from one in the previous case to 4 observations, as shown in Fig. 6.11. The same prior

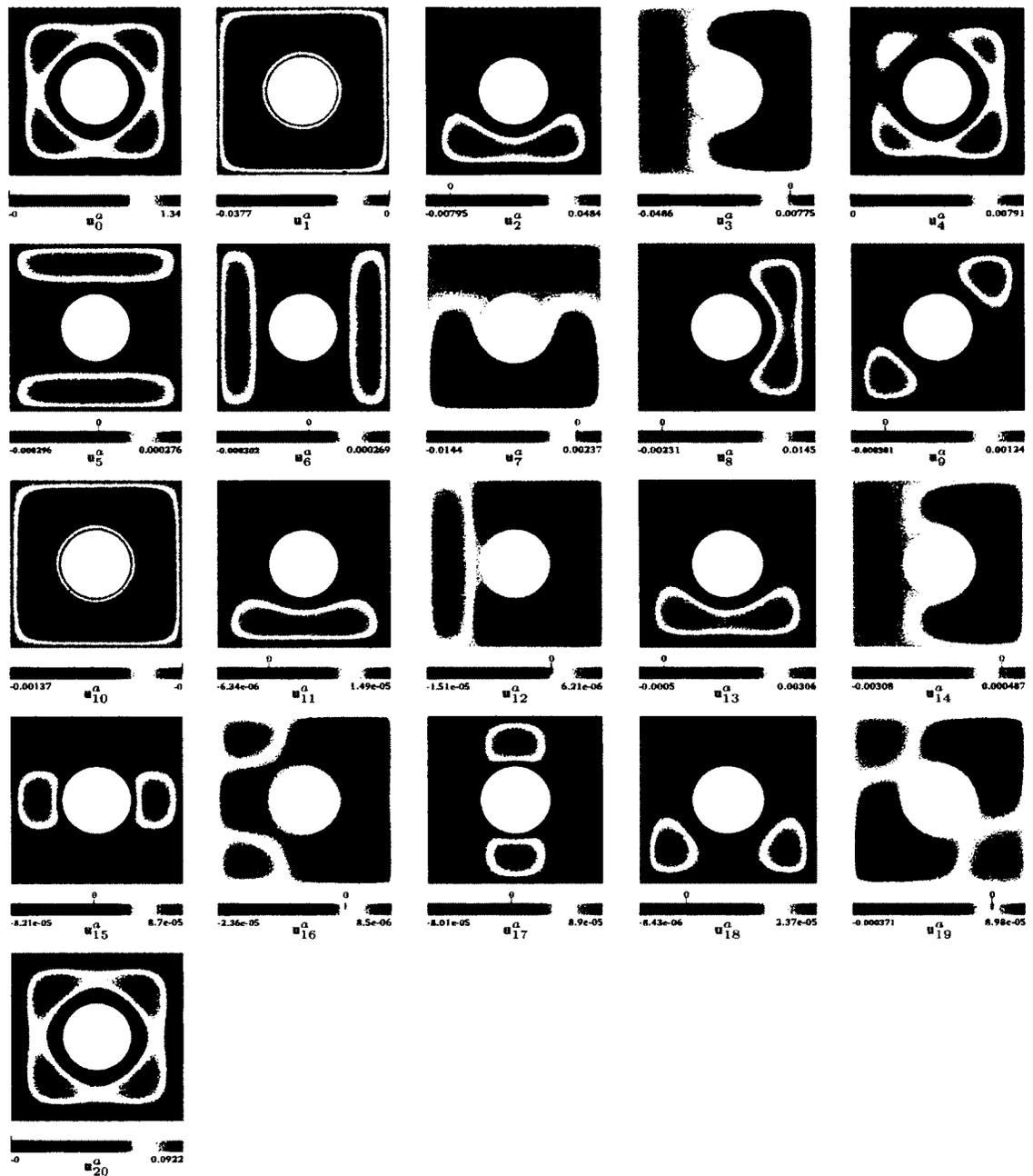


Figure 6.8: Data assimilation in stationary stochastic diffusion problem
- Experiment 1: Posterior PCE coefficients.

PCE coefficients are used as in the previous case. Furthermore, the same true response is used and the measurement noise is independent and identically distributed being Gaussian noise of zero mean and 0.1 standard deviation. The posterior PCE coefficients are

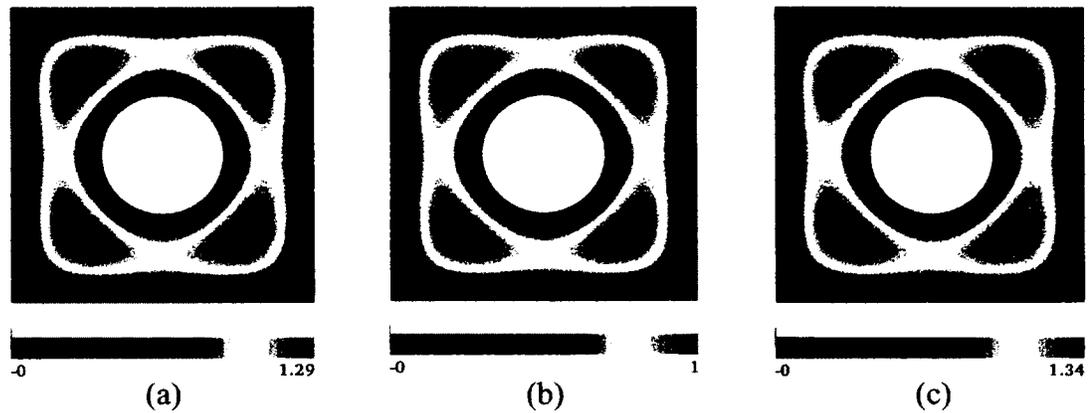


Figure 6.9: Data assimilation in stationary stochastic diffusion problem - Experiment 1: (a) True response, (b) prior mean estimate, and (c) posterior mean estimate.

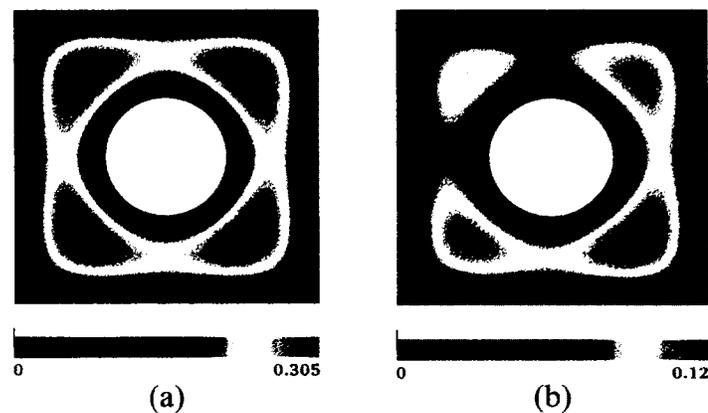


Figure 6.10: Data assimilation in stationary stochastic diffusion problem - Experiment 1: (a) prior standard deviation, and (b) posterior standard deviation.

shown in Fig. 6.12. The number of coefficients has increased due to the assimilation of data from 20 prior PCE coefficients to 24 posterior coefficients due to the assimilation of 4 observations. In general, with the exception of the mean coefficient, the PCE coefficients have been attenuated in the proximity of the sensors from which the observational data is gathered. Fig. 6.13 shows the true response, and prior and posterior mean estimates. Compared to the previous case, the assimilation of more data renders

the posterior mean estimate closer to the true solution. Furthermore, the posterior standard deviation, shown in Fig. 6.14, is reduced even more compared to the previous case of one observation. The presence of symmetry in the response and sensor distribution results in a posterior standard deviation of similar spatial features to that of the prior standard deviation.

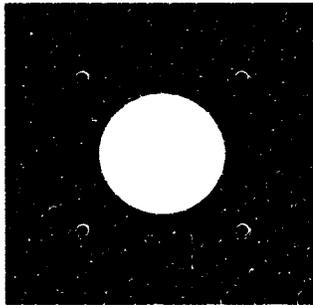


Figure 6.11: Data assimilation in stationary stochastic diffusion problem - Experiment 2: Sensor locations.

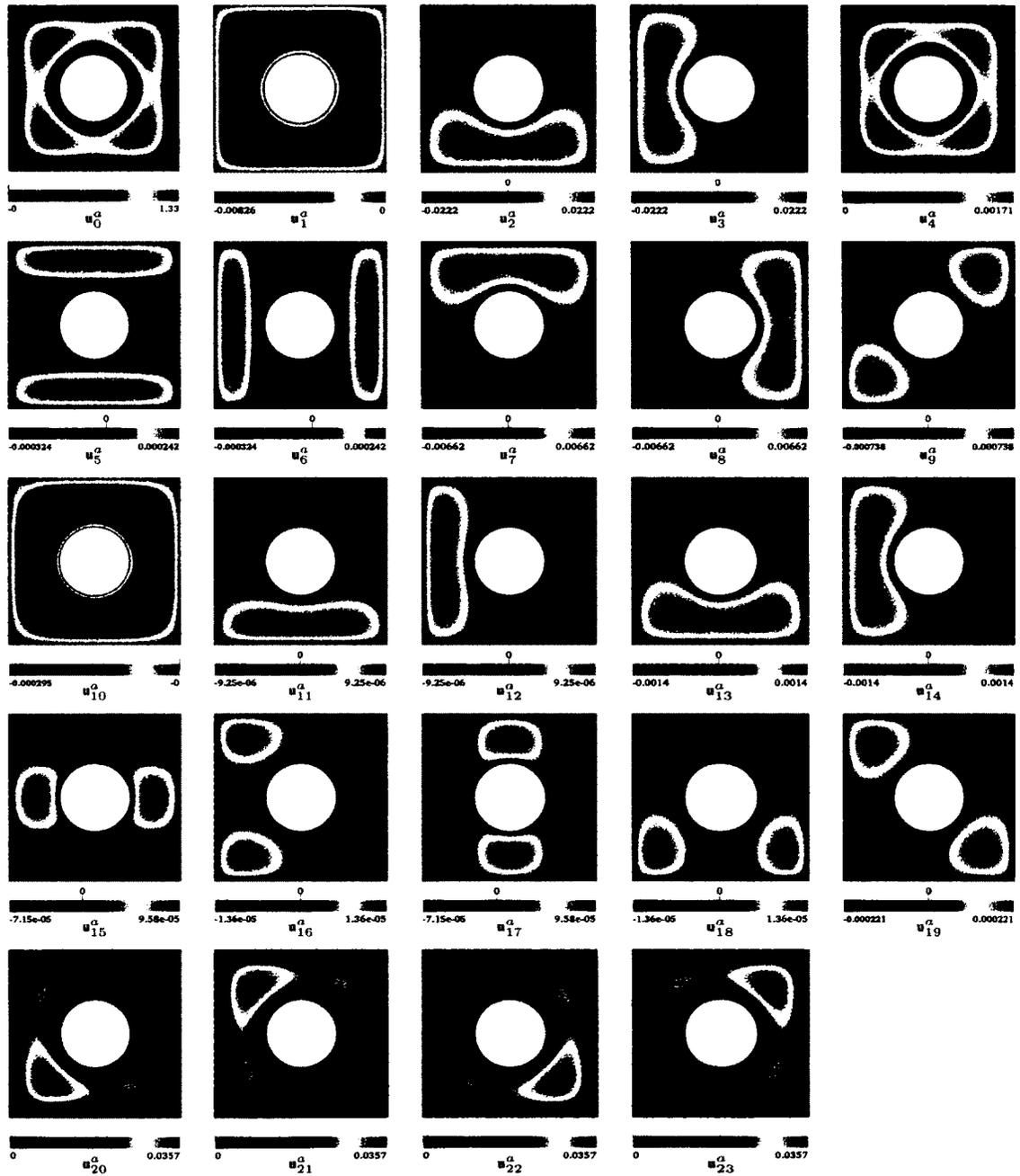


Figure 6.12: Data assimilation in stationary stochastic diffusion problem - Experiment 2: Posterior PCE coefficients.

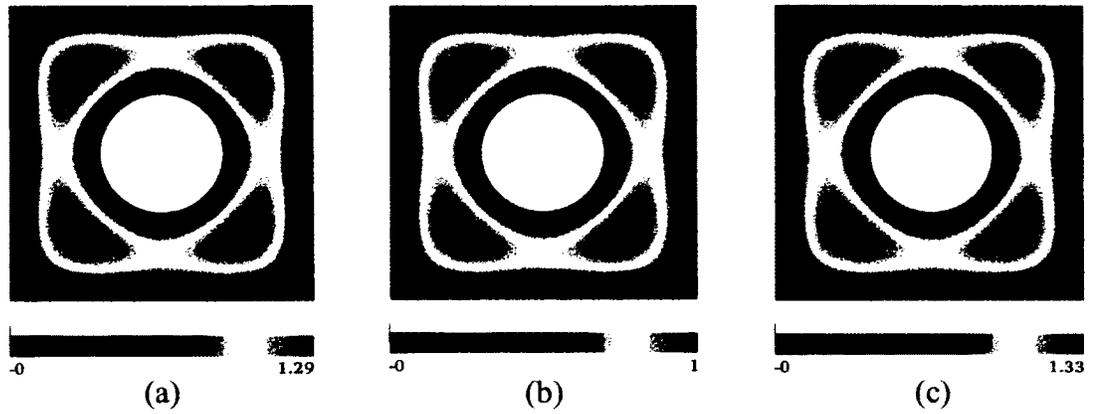


Figure 6.13: Data assimilation in stationary stochastic diffusion problem - Experiment 2: (a) True response, (b) prior mean estimate, and (c) posterior mean estimate.

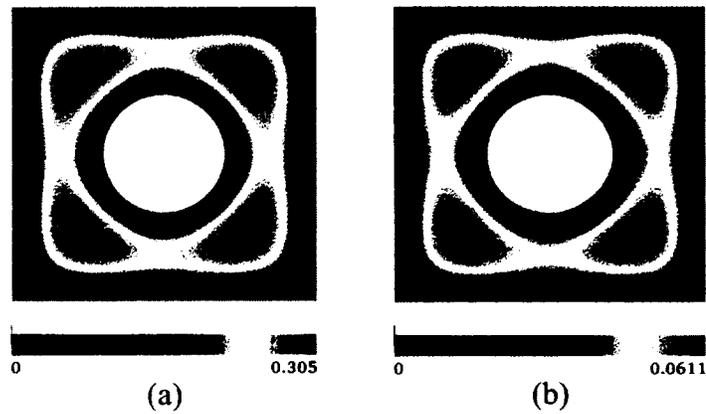


Figure 6.14: Data assimilation in stationary stochastic diffusion problem - Experiment 2: (a) prior standard deviation, and (b) posterior standard deviation.

6.4.2 Stationary Stochastic Advection-Diffusion Problem

The governing equations for the stationary stochastic advection-diffusion boundary value problem are [178]

$$\begin{aligned} -\nabla \cdot (c(\mathbf{x}, \theta) \nabla u(\mathbf{x}, \theta)) + \mathbf{b} \cdot \nabla u(\mathbf{x}, \theta) &= f(\mathbf{x}) && \text{in } D \times \Omega, \\ u(\mathbf{x}, \theta) &= u_0 && \text{on } \partial D \times \Omega, \end{aligned} \quad (6.52)$$

in which D is the spatial domain of interest, c is the random diffusion coefficient, f is the source or right-hand side term, \mathbf{b} is the advection velocity, u is the unknown concentration, and Ω is a sample space belonging to the probability space (Ω, \mathcal{F}, P) in which \mathcal{F} is a σ -algebra and P is a probability measure function [177]. The random diffusion coefficient is modeled as a lognormal stochastic process. The deterministic source function f is given by

$$f(x, y) = f_0 e^{-2000((x-x_0)^2+(y-y_0)^2)} \quad (6.53)$$

with $f_0 = 45.23$, $x_0 = 0.15$, and $y_0 = 0.5$. The case of Dirichlet boundary condition $u_0 = 0$ for the left boundary will be considered. The SSFEM discretization leads to a non-symmetric system matrix and thus a new parallel iterative solver based on the preconditioned biconjugate gradient stabilized method and domain decomposition is developed for stochastic systems in Appendix D.4. This solver will be used to forecast the PCE coefficients. This solver is validated for a stationary stochastic advection-diffusion problem in Appendix D.4.1.

Prior PCE of Stochastic Response

The diffusion coefficient $c(\mathbf{x}, \theta)$ is modeled as a lognormal stochastic process, i.e.

$$c(\mathbf{x}, \theta) = \exp (g(\mathbf{x}, \theta)) \quad (6.54)$$

with an underlying Gaussian process $g(\mathbf{x}, \theta)$ having the following exponential covariance function [1]

$$C(x_1, x_2; y_1, y_2) = \sigma^2 e^{-|x_1-y_1|/b_1 - |x_2-y_2|/b_2}, \quad (6.55)$$

in which σ^2 is the variance and b_1 and b_2 are the correlations lengths along the x_1 and x_2 directions, respectively. The analytical expressions for the eigenfunctions $g_n(\mathbf{x})$ and eigenvalues λ_n of such covariance function are given in Appendix C.1.1. The numerical values used in the investigation are $\sigma = 0.3$ and $b_1 = b_2 = 50$, leading to the eigenvalue distribution given in Fig. 6.2. The first three eigenvalues account for approximately 99% of the energy of the Gaussian random process and their associated eigenfunctions are shown in Fig.(6.3). Thus, the underlying Gaussian process will be represented by the truncated KLE

$$g(\mathbf{x}, \theta) = g_0(\mathbf{x}) + \sum_{n=1}^3 \sqrt{\lambda_n} g_n(\mathbf{x}) \xi_n(\theta) \quad (6.56)$$

in which ξ_n are independent and identically distributed normal random variables and g_1 through g_3 are the eigenfunctions shown in Fig. 6.3. For this set of experiments, the mean of the underlying Gaussian process is $g_0(\mathbf{x}) = -4.75$. Thus a three-dimensional PCE ($L = 3$) of the lognormal process will be required to capture the input uncertainty. It is also determined through numerical investigation (results not shown for brevity) that a third-order PC expansion ($p = 3$) is required to sufficiently capture the non-Gaussian

nature of the input. A third-dimensional PCE of order 3 will be used to represent the input diffusivity resulting in a total of $N = 20$ PCE coefficients whose analytical expressions are provided in Appendix C.2.1.

For numerical illustration, the spatial domain is discretized into a finite element mesh consisting of 100,514 linear triangular elements and 50,256 nodes as illustrated in Fig. 6.15a. The spatial domain is decomposed into 176 subdomains as shown in Fig. 6.15b resulting in a total of 5,560 global interface nodes. For uncertainty representation, the response is represented using a three-dimensional third-order PCE (i.e. $N = 20$), resulting in a linear system of order 1,005,120. Using the parallel solver based on domain decomposition and preconditioned biconjugate gradient stabilized method for the extended Schur complement system (see Appendix D.4 for the parallel algorithm), the coefficients of the solution process are computed and illustrated in Fig. 6.16. The Krylov-subspace iterate is initialized to zero and the stopping criteria for the preconditioned BICGSTAB solver is

$$\frac{\| \mathcal{X}_{r_{i+1}} - \mathcal{X}_{r_i} \|}{\| \mathcal{X}_{r_i} \|} \leq \epsilon = 10^{-15} . \quad (6.57)$$

These PCE coefficients represent the state vector prior to the assimilation of observations. The zeroth PCE coefficient representing the mean response is replotted in Fig. 6.17a. The standard deviation of the response, computed as the square roots of the covariance matrix diagonal entries, is shown in Fig. 6.17b. At its peak mean response, the coefficient of variation (being the mean-normalized standard deviation) is approximately 10%.

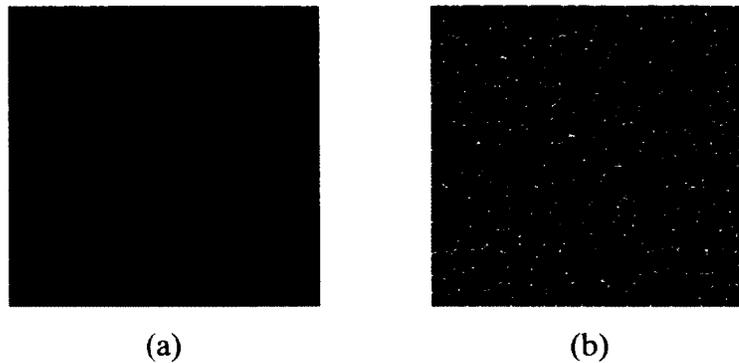


Figure 6.15: Stationary stochastic advection-diffusion problem: (a) Finite element mesh and (b) the partitioned domain.

Data Assimilation - Experiment 1

A realization obtained from the prior PCE of the solution process is used to represent the “true” response from which noise-contaminated observations are obtained. The noise is assumed to be Gaussian in nature with zero mean and 0.05 standard deviation. For the first experiment, a single observation at the location shown in Fig. 6.18 is used to update the PCE coefficients. The updated (or posterior) PCE coefficients are shown in Fig. 6.19. The number of coefficients has increased by one due to the assimilation of a single observation. It is obvious that some PCE coefficients have been attenuated by the assimilation of observational data, except for the mean coefficient. The true response, and prior and posterior mean estimates are provided in Fig. 6.20. The posterior mean estimate, obtained using the available noisy observation, is closer to the true solution than the prior estimate. The assimilation of data reduces the error in the estimates as evident in Fig 6.21. It is important to note negative posterior mean estimates in some regions of the computational domain. In certain cases such as this, the assimilation of data using PCKF results in physically inconsistent estimates due to the limitation of the linear analysis step common to all Kalman filter based algorithms including EnKF. This

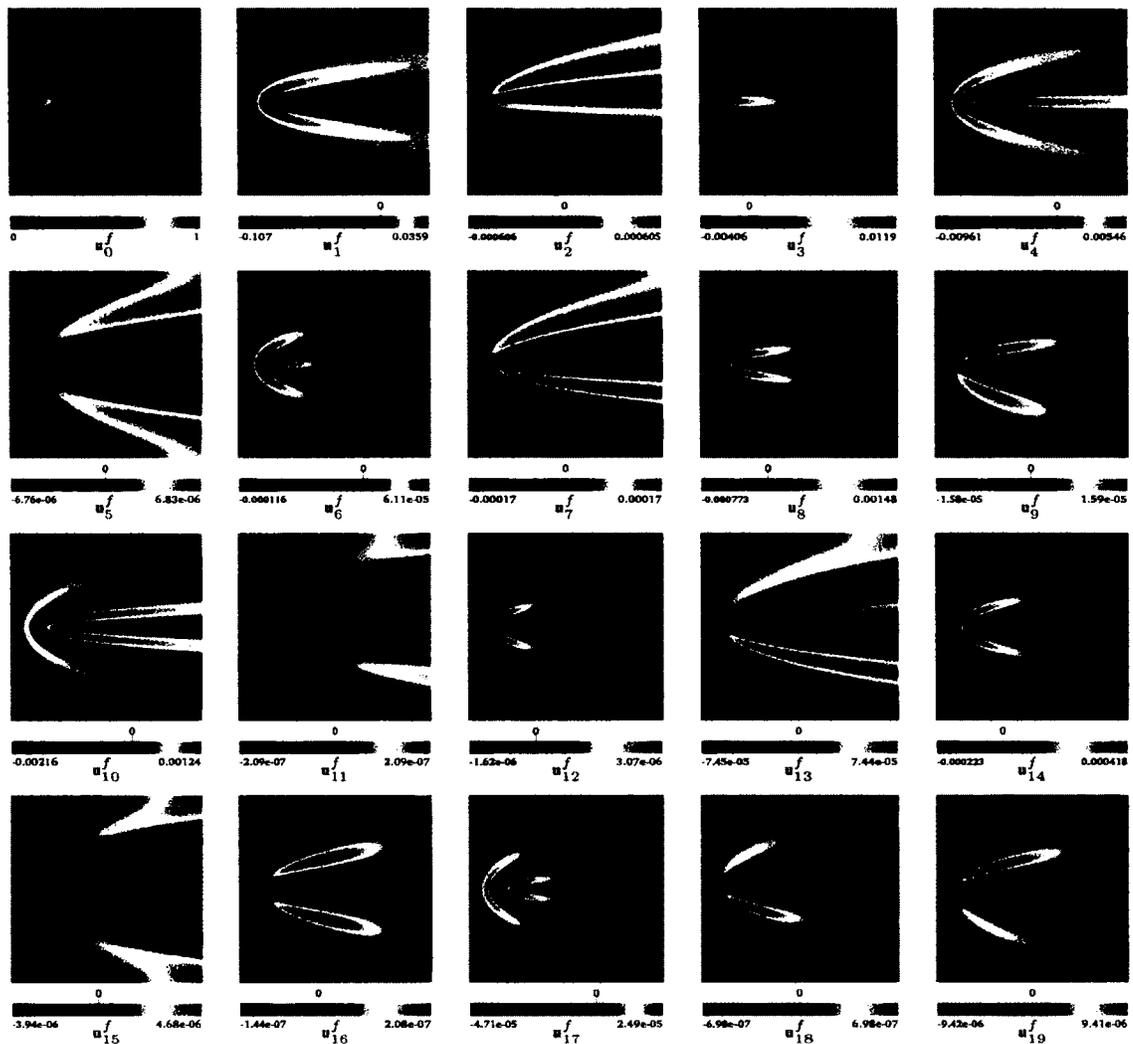


Figure 6.16: Stationary stochastic advection-diffusion problem: Prior PCE coefficients.

analysis step is derived for Gaussian prior and measurement errors and will result in unwanted artifacts (such as an estimated negative concentration) for cases of non-Gaussian priors, as is the case in this experiment. The relative magnitude of these negative estimates is small due to the weak non-Gaussian characteristics in the prior pdf of the state vector. For strongly non-Gaussian prior pdfs, the particle filter might be a more suitable alternative albeit with a greater computational requirement.

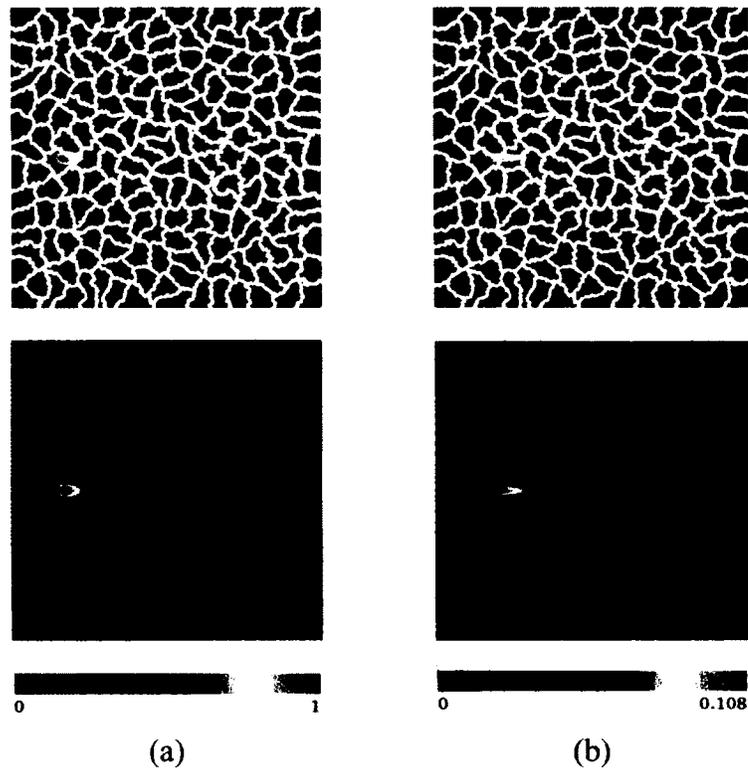


Figure 6.17: Stationary stochastic advection-diffusion problem: (a) Prior mean response (b) standard deviation. The spatial decomposition of the mean response and standard deviation is shown in the top figures.

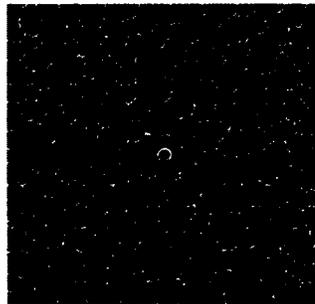


Figure 6.18: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: Sensor location.

Data Assimilation - Experiment 2

In this data assimilation experiment, the number of observations available for data assimilation is four as opposed to one for the previous experiment. The locations of the

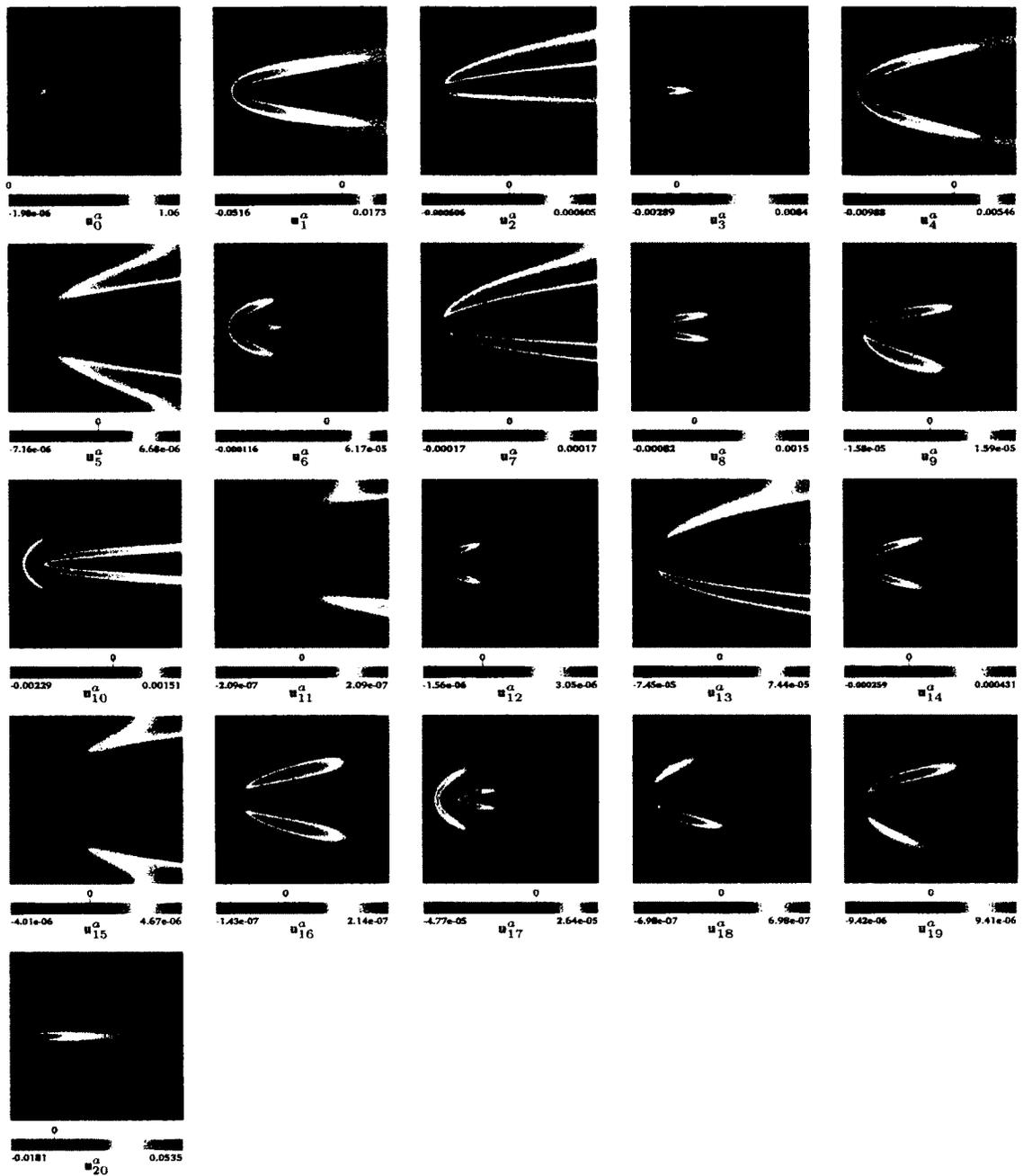


Figure 6.19: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: Posterior PCE coefficients.

sensors from which data will be gathered are illustrated in Fig. 6.22. The same prior PCE coefficients are used in this experiment. Furthermore, the same true response is used with independent and identically distributed Gaussian measurement noise of zero

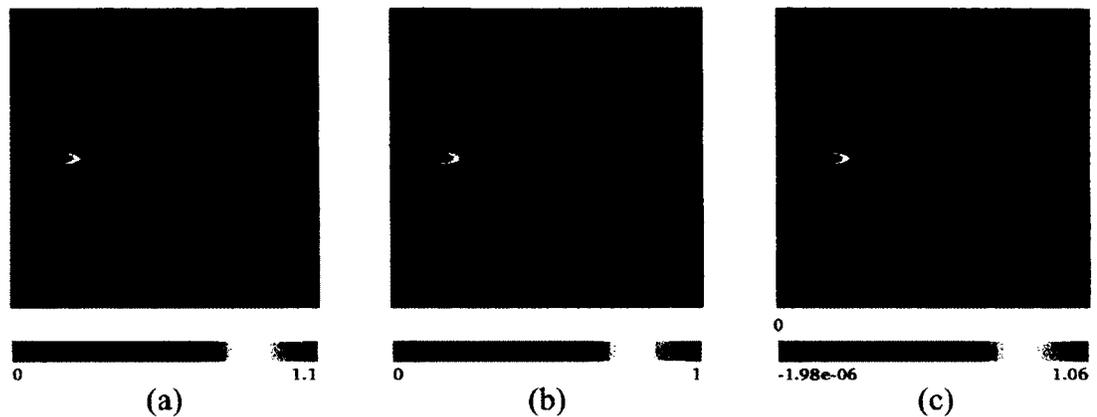


Figure 6.20: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: (a) True response, (b) prior mean estimate, and (c) posterior mean estimate.

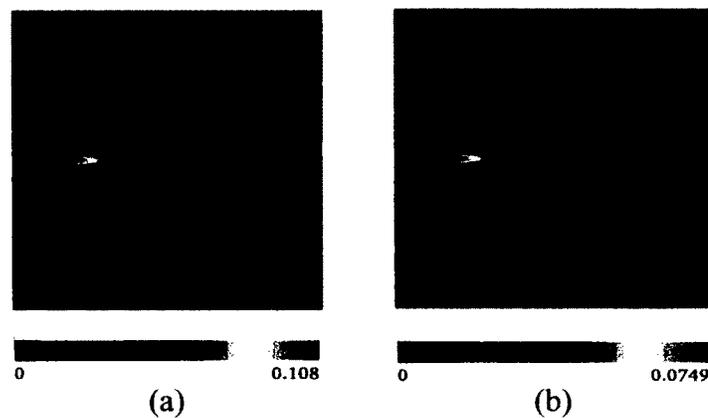


Figure 6.21: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 1: (a) prior standard deviation, and (b) posterior standard deviation.

mean and 0.05 standard deviation. The posterior PCE coefficients are shown in Fig. 6.23 in which number of coefficients has increased due to the assimilation of four data points. In general, with the exception of the mean coefficient, the PCE coefficients have been further attenuated by the assimilation of an additional three observations, in comparison to the previous experiment. Fig. 6.24 illustrates the true response, and prior and posterior mean estimates. Compared to the previous case, the assimilation of more data

renders the posterior mean estimate closer to the true solution. Furthermore, the posterior standard deviation, shown in Fig. 6.25, is reduced further compared to the previous experiment involving only one observation.

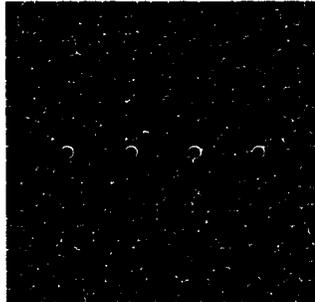


Figure 6.22: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: Sensor locations.

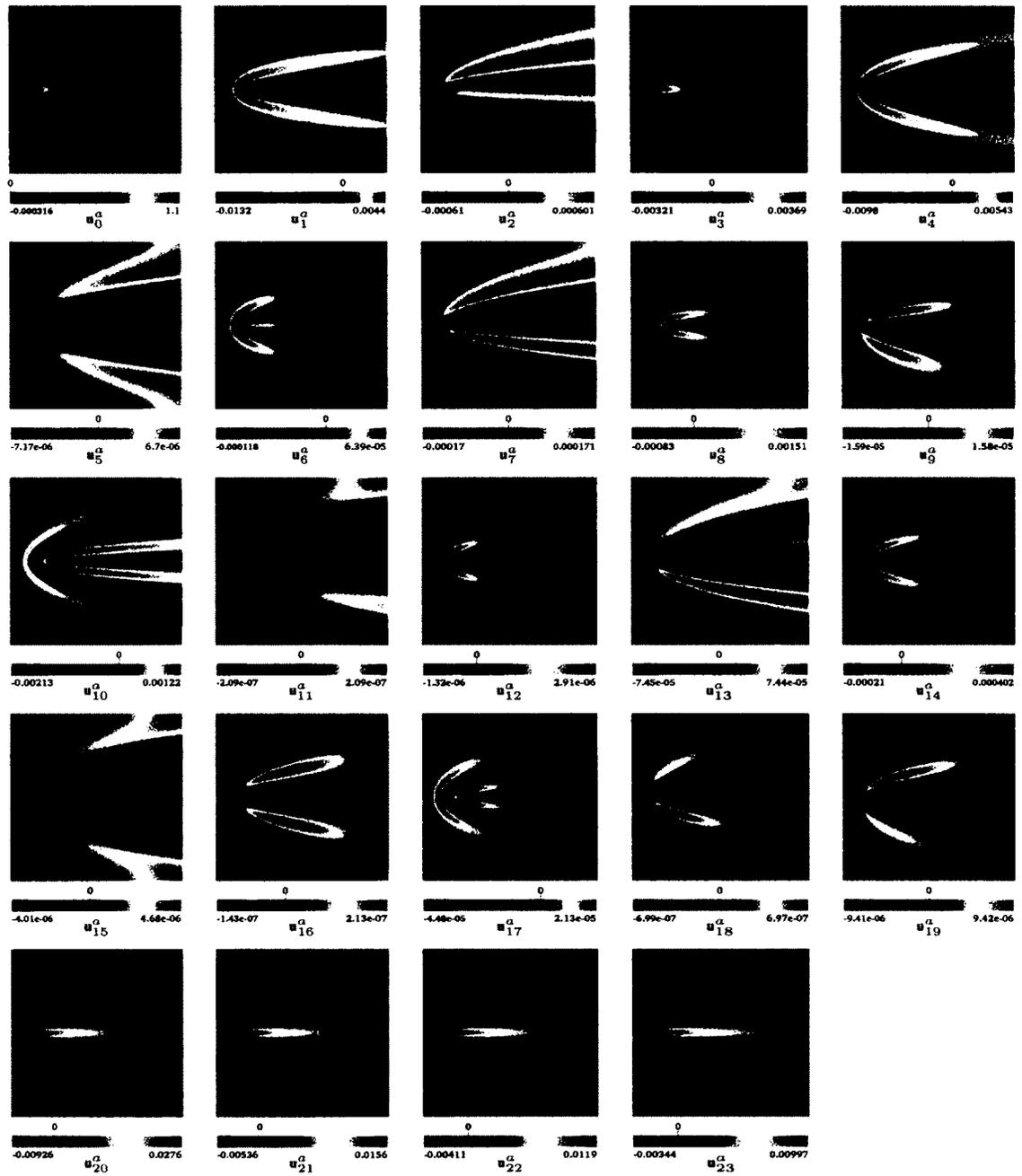


Figure 6.23: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: Posterior PCE coefficients.

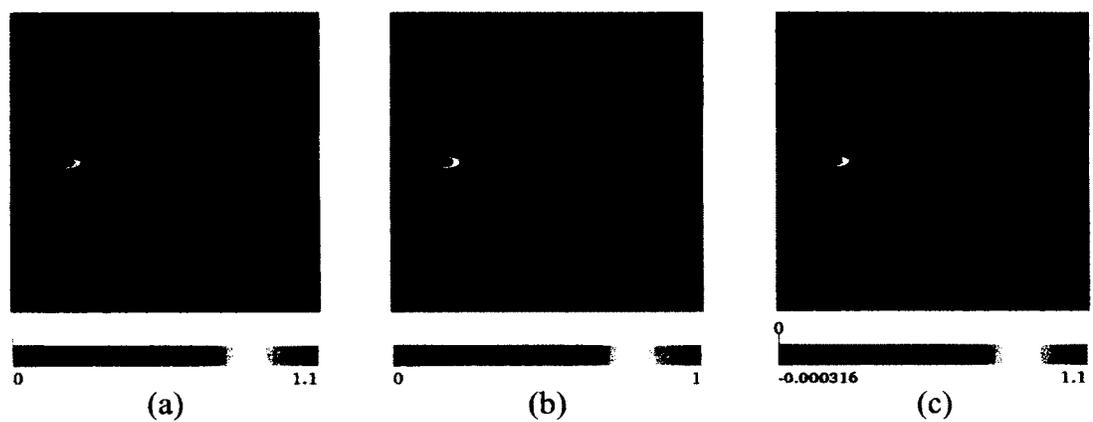


Figure 6.24: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: (a) True response, (b) prior mean estimate, and (c) posterior mean estimate.

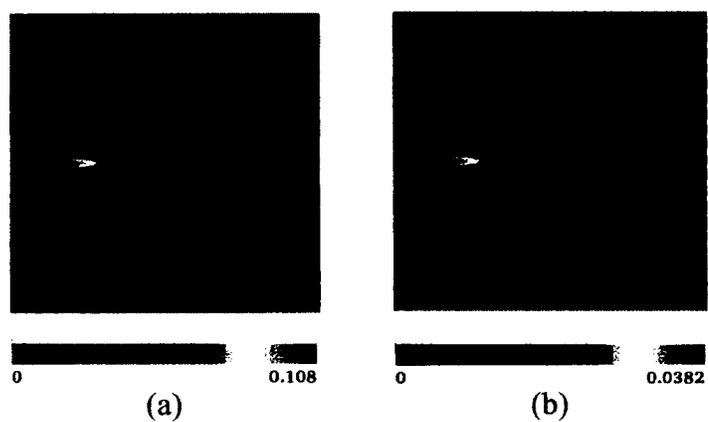


Figure 6.25: Data assimilation in stationary stochastic advection-diffusion problem - Experiment 2: (a) prior standard deviation, and (b) posterior standard deviation.

6.4.3 Parallel Scalability of the Parallel Data Assimilation Algorithm

The parallel scalability of the proposed parallel assimilation algorithm is examined next. The scalability of the algorithm is examined under three aspects: (1) the global problem size (i.e. order of the state vector elements) and number of measurements (i.e. sensors) is fixed while the number of subdomains (i.e. CPUs) is increased, (2) the problem size per subdomain is fixed, the number of measurements fixed, while the number of subdomains is increased, and (3) the number of measurements per subdomain is fixed, the global problem size is fixed, while the number of subdomains is increased. Although more strategic scalability studies can be formulated based on statistical design of experiments [179], this approach is not followed in this thesis. The algorithm is implemented using Fortran programming language (Fortran 90) and parallelized using Message Passing Interface (MPI) libraries [77]. Timing statistics of the parallel code are obtained using the MPI timer `MPI_WTIME()` for 10 runs and the average execution time is reported. The code is executed on a Linux cluster consisting of 176 core HP Intel Xeon cluster consisting of 22 nodes with 8 cores and 32 GB of memory per node with InfiniBand interconnect.

Parallel scalability with fixed problem size and number of measurements

In this scalability study, a constant global problem size is maintained while the number of CPUs to tackle the problem is increased in order to reduce the total execution time. The parallel scalability of the proposed parallel data assimilation algorithm (Algorithm 6.1) with respect to the number of subdomains is shown in Fig. 6.26. Three plots are provided, each for a different number of measurements. Within each plot, the

results for three global problem sizes (total number of unknowns) are shown. For the first case of 32 measurements, i.e. Fig. 6.26a, an initial increase in number of processors used to tackle the problem greatly reduces the total execution time. For this specific case, the optimal number of CPUs is 50. Increasing the number of CPUs further results in a communication overhead that dominates the total execution time, resulting in longer time of execution. As the number of measurements assimilated increases from 32 to 128, the optimal value of CPUs needed to solve the problem increases (beyond 50) for the case of 3.5 million total unknowns as evident by the black curve shown in Fig. 6.26b. A further increase of number of measurements to 512 pushes the optimal number of required CPUs beyond fifty even for the case of 1 million unknowns (see blue curve in Fig. 6.26c). These results indicate a trend of increasing optimal number of CPUs required for this parallel data assimilation algorithm as both the number of measurements and global problem size increases.

Parallel scalability with fixed problem size per subdomain

Next, the parallel scalability of the proposed parallel data assimilation algorithm is examined for fixed problem size per subdomain. The goal of this scalability study is to tackle an n -times larger system with n -times more processors or cores while ideally maintaining a nearly constant execution time. Fig.6.27 shows that for the fixed problem size per processor, the total execution time grows steadily with the number of processors used to tackle the problem. This is a trend observed for the three different number of assimilated measurements. The deviation from the ideal constant execution time is largely attributed to the increase in inter-node communication as the number of processors used increases.

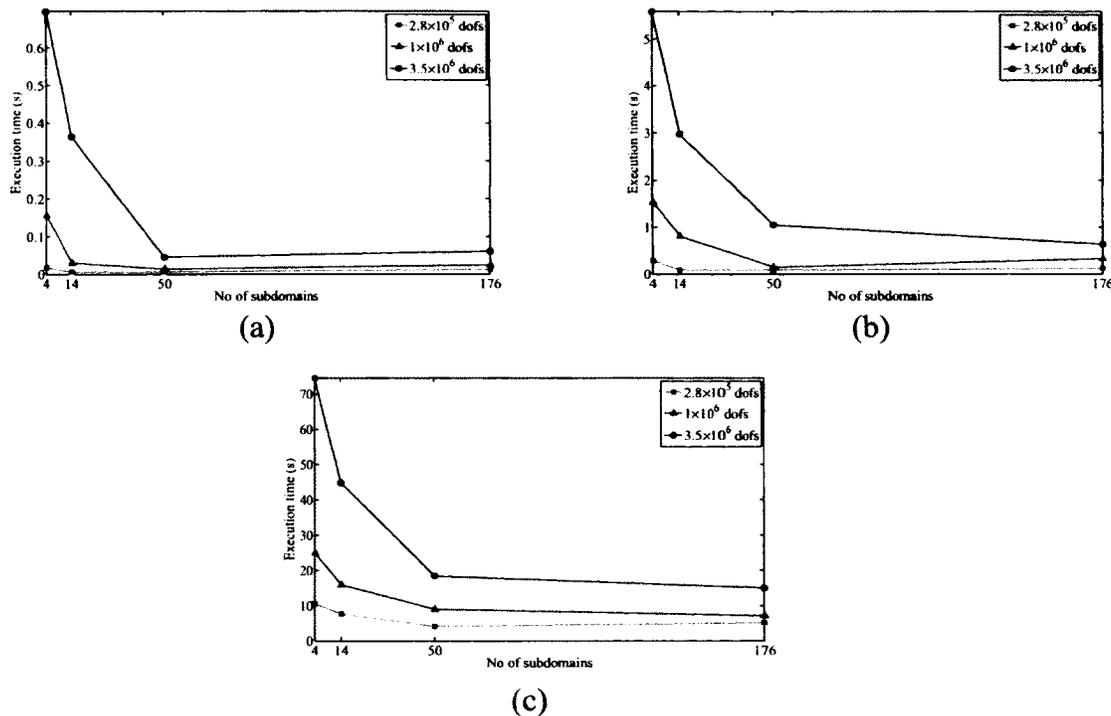


Figure 6.26: Parallel scalability of the parallel data assimilation algorithm: fixed problem size and number of measurements for (a) 32 measurements, (b) 128 measurements, and (c) 512 measurements.

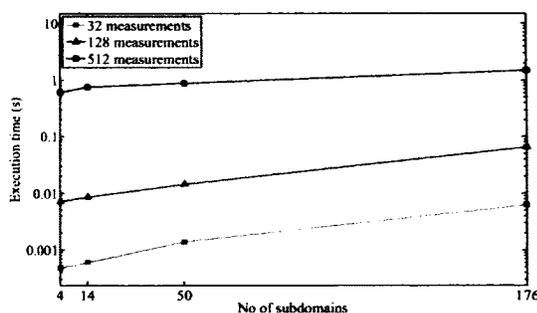


Figure 6.27: Parallel scalability of the parallel data assimilation algorithm: fixed problem size per subdomain.

Parallel scalability with fixed number of measurements per subdomain

For a final scalability study of the proposed parallel data assimilation algorithm, the parallel scalability is examined for fixed number of assimilated measurements per sub-

domain. The goal of this scalability study is to assimilate n -times more measurements with n -times more processors or cores for a given global problem size while ideally maintaining a nearly constant execution time. Fig.6.28 shows that for the case of fixed number of measurements per processor, the total execution time grows steadily with the number of processors used to tackle the problem. Furthermore, there is strong evidence that all three curves for varying global problem size will converge to a single curve if the number of domains is increased further. Thus it seems that the total execution time for the parallel data assimilation algorithm is not heavily dependent on the global problem size when the number of measurements per subdomain remains fixed and more subdomains are used.

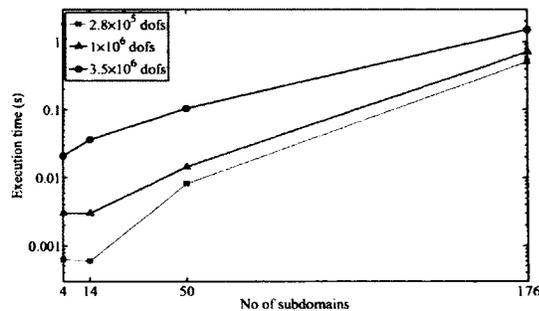


Figure 6.28: Parallel scalability of the parallel data assimilation algorithm: fixed number of measurements per subdomain.

Chapter 7

Summary of Contributions and Future Research Directions

7.1 Summary of Contributions

In this thesis, Bayesian inference is used to tackle the problem of state or state and parameter estimation of nonlinear stochastic ODEs and large-scale models arising from finite element discretization of stochastic PDEs. The contributions of the thesis lie in both novel applications and algorithmic development. For nonlinear oscillators, the problem of static parameter estimation is tackled using an MCMC algorithm complemented by nonlinear filters for state estimation. The methodology is validated using both numerical experiments and wind tunnel data obtained from an aeroelastic system undergoing limit cycle oscillation. For aeroelastic flutter, a Bayesian inference scheme is formulated to propagate the uncertainties in the vibration data providing robust estimates of the flutter speed.

For large-scale systems, a parallel data assimilation algorithm is developed which is based on polynomial chaos expansion for uncertainty representation and the Kalman

filter linear analysis step to update the polynomial chaos coefficients of the state vector. This algorithm takes advantage of the polynomial chaos based domain decomposition solvers for large-scale computational models. The parallel scalability of the proposed algorithm is investigated on a Linux cluster consisting of 22 nodes with 8 cores and 32 GB of memory per node having InfiniBand interconnect. For stochastic systems with non-symmetric matrices, a parallel iterative solver is developed based on domain decomposition method and preconditioned biconjugate gradient stabilized Krylov-subspace solver. This solver is used to tackle the steady-state stochastic advection-diffusion problem.

The following salient features emerged from the investigations undertaken in this thesis: (1) the proposed probabilistic flutter margin method in Chapter 3 applied to a two-degrees-of-freedom numerical model provided more accurate and more precise estimates for the flutter speed in comparison to the conventional method; (2) for the inverse problem of estimating time-invariant parameters of a nonlinear system exhibiting noisy oscillation considered in Chapter 4, Markov Chain Monte Carlo sampling, ensemble Kalman filter and particle filter provided parameter estimates which are comparable in accuracy for the case of dense observational data and weak measurement noise whereas for the cases of strong measurement noise and sparse observational data, the parameter estimates from ensemble Kalman filter (for combined state and parameter estimation) became unwieldy. When the ensemble Kalman filter was used for just the state estimation, the Markov Chain Monte Carlo sampling-based algorithm and particle filter both provided accurate estimates for the parameters; (3) Regarding the static parameter estimation problem of a non-linear aeroelastic system using wind tunnel data, as presented in chapter 5, the numerical predictions obtained using the reduced model showed good agreement with the experimental observations from wind tunnel tests. The estimated static moment coefficient compared fairly well with the data published in liter-

ature, with anticipated differences due to variations in the experimental set-up and conditions; (4) Regarding the parallel data assimilation algorithm developed in Chapter 6 that integrates the polynomial chaos-based Kalman filter and a polynomial chaos-based domain decomposition solver, a scalable performance was observed for stochastic diffusion and advection-diffusion problems with spatially varying non-Gaussian stochastic parameters; (5) The developed parallel solver for non-symmetric stochastic systems based on the preconditioned biconjugate gradient stabilized method (see Appendix D.4) was validated for the steady-state stochastic advection-diffusion problem with random diffusivity modeled as a spatially varying non-Gaussian stochastic process.

7.2 Future Research Directions

Based on the investigations carried out in this thesis, the following research directions are envisioned:

- **Application of Bayesian Inference to the Flutter Margin Method:**

- The proposed Bayesian flutter margin method needs to be generalized to consider more complex aeroelastic models involving, for instance, (a) unsteady aerodynamic model and (b) turbulence. These issues must be addressed for wider applicability of the method.
- As a blind validation experiment, the method should be applied to available wind tunnel data. In certain cases, the presence of nonlinearity in the observed wind tunnel data must be modeled explicitly.
- The method should be applied to more realistic problems including flutter speed estimation of F-18 using flight test data. For in-flight tests, note that

the airspeed is not known precisely and thus one must account for uncertainty in airspeed values.

- **Static Parameter Estimation of a Non-linear Aeroelastic System using Wind Tunnel Data:**

- The aerodynamic behaviour used to model the observed limit cycle oscillation is based on quasi-steady aerodynamics. The effect of unsteady aerodynamics should be considered for future investigations.
- Further investigations are initiated [149, 180] in order to (a) apply the proposed methodology to other transitional Reynolds numbers which identifies the effect of Reynolds number on aerodynamic parameters, (b) apply the model selection methods for model reduction based on the exact Bayesian Information Criterion, and (c) study the effect of unsteady aerodynamics on the observed limit cycle oscillation.

- **Data Assimilation for Large-scale Computational Models:**

- The proposed parallel algorithm for data assimilation demonstrates good parallel scalability on available high performance computing platform. The rapid advancement of high performance computing systems and rapid development and deployment of modern high-resolution sensing technologies warrants further improvement of the aforementioned data assimilation algorithm towards its scalability to extreme-scale systems.
- The application of the parallel forward solver and parallel data assimilation methods examined in this thesis was limited to stationary stochastic PDEs

as the first step in algorithmic development. To examine the performance of these methods for more realistic systems, the applications should consider time-dependent problems.

- **Parallel Solver for Non-Symmetric Systems Based on Preconditioned Biconjugate Gradient Stabilized Method:**

- It is envisioned that the proposed parallel solver based on preconditioned biconjugate gradient stabilized method will have difficulty in scaling to very large number of processors/cores due to the use of a one-level preconditioner. Two-level parallel preconditioners [181] should be used to enhance the scalability of the algorithm to very large number of processors/cores in order to tackle large-scale filtering problems.

Appendix A

Mathematical Preliminaries

A.1 Inverse and Determinant of Partitioned Symmetric Matrix

The book by Levine [182] is followed for this section.

Theorem A.1.

For a $r \times r$ matrix \mathbf{A} , $r \times q$ matrix \mathbf{C} , $q \times q$ matrix \mathbf{B} , and $q \times r$ matrix \mathbf{D} , one has

$$(\mathbf{A} + \mathbf{CBD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{C}(\mathbf{B}^{-1} + \mathbf{DA}^{-1}\mathbf{C})^{-1}\mathbf{DA}^{-1}. \quad (\text{A.1})$$

Proof.

$$\begin{aligned}
& (\mathbf{A} + \mathbf{CBD}) \left[\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{C} (\mathbf{B}^{-1} + \mathbf{DA}^{-1}\mathbf{C})^{-1} \mathbf{DA}^{-1} \right] \\
&= (\mathbf{A} + \mathbf{CBD}) \mathbf{A}^{-1} - (\mathbf{A} + \mathbf{CBD}) \mathbf{A}^{-1}\mathbf{C} (\mathbf{B}^{-1} + \mathbf{DA}^{-1}\mathbf{C})^{-1} \mathbf{DA}^{-1} \\
&= \mathbf{I} + \mathbf{CBDA}^{-1} - (\mathbf{C} + \mathbf{CBDA}^{-1}\mathbf{C}) (\mathbf{B}^{-1} + \mathbf{DA}^{-1}\mathbf{C})^{-1} \mathbf{DA}^{-1} \\
&= \mathbf{I} + \mathbf{CBDA}^{-1} - \mathbf{CB} (\mathbf{B}^{-1} + \mathbf{DA}^{-1}\mathbf{C}) (\mathbf{B}^{-1} + \mathbf{DA}^{-1}\mathbf{C})^{-1} \mathbf{DA}^{-1} \\
&= \mathbf{I} + \mathbf{CBDA}^{-1} - \mathbf{CBDA}^{-1} = \mathbf{I}
\end{aligned}$$

□

Theorem A.2 (Inverse of a Partitioned Symmetric Matrix).

Given a symmetric 2×2 block matrix \mathbf{A} of dimension $n \times n$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \quad (\text{A.2})$$

and its inverse $\mathbf{B} = \mathbf{A}^{-1}$ which can also be divided into four blocks as

$$\mathbf{B} = \mathbf{A}^{-1} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} \end{bmatrix} \quad (\text{A.3})$$

with the following dimensions

- \mathbf{A}_{11} and \mathbf{B}_{11} are $r \times r$,
- \mathbf{A}_{22} and \mathbf{B}_{22} are $q \times q$,
- $\mathbf{A}_{12} = \mathbf{A}_{21}^T$ and $\mathbf{B}_{12} = \mathbf{B}_{21}^T$ are $r \times q$,
- $r + q = n$,

one has

$$\mathbf{B}_{11} = (\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T)^{-1} = \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}(\mathbf{A}_{22} - \mathbf{A}_{12}^T\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1}\mathbf{A}_{12}^T\mathbf{A}_{11}^{-1} \quad (\text{A.4})$$

$$\mathbf{B}_{22} = (\mathbf{A}_{22} - \mathbf{A}_{12}^T\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1} = \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T)^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \quad (\text{A.5})$$

$$\mathbf{B}_{12}^T = -\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T)^{-1} \quad (\text{A.6})$$

$$\mathbf{B}_{12}^T = -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}(\mathbf{A}_{22} - \mathbf{A}_{12}^T\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1} \quad (\text{A.7})$$

Proof. By definition of an inverse, one must have

$$\begin{aligned} \mathbf{I}_n &= \mathbf{A}\mathbf{A}^{-1} = \mathbf{A}\mathbf{B} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{12}^T & \mathbf{B}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{12}^T & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} \\ \mathbf{A}_{12}^T\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{12}^T & \mathbf{A}_{12}^T\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_q \end{bmatrix}, \end{aligned}$$

i.e.,

$$\mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{12}^T = \mathbf{I}_r \quad \text{or} \quad \mathbf{B}_{11} = \mathbf{A}_{11}^{-1} - \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{B}_{12}^T$$

$$\mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} = \mathbf{0} \quad \text{or} \quad \mathbf{B}_{12} = -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{B}_{22}$$

$$\mathbf{A}_{12}^T\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{12}^T = \mathbf{0} \quad \text{or} \quad \mathbf{B}_{12}^T = -\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T\mathbf{B}_{11}$$

$$\mathbf{A}_{12}^T\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} = \mathbf{I}_q \quad \text{or} \quad \mathbf{B}_{22} = \mathbf{A}_{22}^{-1} - \mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T\mathbf{B}_{12}$$

Replace \mathbf{B}_{12}^T by the expression above into \mathbf{B}_{11} to get

$$\mathbf{B}_{11} = \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}^T\mathbf{B}_{11}$$

or

$$(\mathbf{I} - \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T) \mathbf{B}_{11} = \mathbf{A}_{11}^{-1} \quad \text{or} \quad (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T) \mathbf{B}_{11} = \mathbf{I}_r$$

or

$$\mathbf{B}_{11} = (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T)^{-1} .$$

Applying Theorem A.1 to this expression, the other expression in the theorem may also be obtained, i.e.

$$\mathbf{B}_{11} = \mathbf{A}_{11}^{-1} - \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \mathbf{B}_{12}^T .$$

Similarly,

$$\mathbf{B}_{22} = (\mathbf{A}_{22} - \mathbf{A}_{12}^T \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} ,$$

$$\mathbf{B}_{12}^T = -\mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T)^{-1} ,$$

and

$$\mathbf{B}_{12} = -\mathbf{A}_{11}^{-1} \mathbf{A}_{12} (\mathbf{A}_{22} - \mathbf{A}_{12}^T \mathbf{A}_{11}^{-1} \mathbf{A}_{12})^{-1} .$$

□

Theorem A.3 (Determinant of a Partitioned Symmetric Matrix).

Given a symmetric 2×2 block matrix \mathbf{A} of dimension $n \times n$,

$$|\mathbf{A}| = \left| \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \right| = |\mathbf{A}_{22}| |\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T| = |\mathbf{A}_{11}| |\mathbf{A}_{22} - \mathbf{A}_{12}^T \mathbf{A}_{11}^{-1} \mathbf{A}_{12}| . \quad (\text{A.8})$$

Proof. Given the decomposition of the matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{12}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} - \mathbf{A}_{12}^T \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \end{bmatrix} \\ = \begin{bmatrix} \mathbf{I} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T & \mathbf{0} \\ \mathbf{A}_{22}^{-1} \mathbf{A}_{21} & \mathbf{I} \end{bmatrix}.$$

In general,

$$|\mathbf{AB}| = |\mathbf{A}| |\mathbf{B}|$$

and

$$\begin{vmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{D} \end{vmatrix} = \begin{vmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{D} \end{vmatrix} = |\mathbf{B}| |\mathbf{D}|.$$

Finally,

$$|\mathbf{A}| = \left| \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{12}^T & \mathbf{I} \end{bmatrix} \right| \left| \begin{bmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} - \mathbf{A}_{12}^T \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \end{bmatrix} \right| \\ = \left| \begin{bmatrix} \mathbf{I} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} \right| \left| \begin{bmatrix} \mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T & \mathbf{0} \\ \mathbf{A}_{22}^{-1} \mathbf{A}_{21} & \mathbf{I} \end{bmatrix} \right| \\ = |\mathbf{A}_{11}| |\mathbf{A}_{22} - \mathbf{A}_{12}^T \mathbf{A}_{11}^{-1} \mathbf{A}_{12}| \\ = |\mathbf{A}_{22}| |\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{12}^T|$$

□

A.2 Marginal and Conditional Distributions of Multivariate Normal Distribution

Theorem A.4. Assume an n -dimensional random vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

has a normal distribution $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

where \mathbf{x}_1 and \mathbf{x}_2 are two sub-vectors of respective dimensions r and q with $r + q = n$.

Note that $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T$, and $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^T$.

The marginal distributions of \mathbf{x}_1 and \mathbf{x}_2 are also normal with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_{ii}$ ($i = 1, 2$), respectively. Furthermore, The conditional distribution of \mathbf{x}_i given \mathbf{x}_j is also normal with mean vector

$$\boldsymbol{\mu}_{i|j} = \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_{ij} \boldsymbol{\Sigma}_{jj}^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_j) \quad (\text{A.9})$$

and covariance matrix

$$\boldsymbol{\Sigma}_{i|j} = \boldsymbol{\Sigma}_{ii} - \boldsymbol{\Sigma}_{ij}^T \boldsymbol{\Sigma}_{jj}^{-1} \boldsymbol{\Sigma}_{ij} \quad (\text{A.10})$$

Proof. The joint density of \mathbf{x} is:

$$\begin{aligned} p(\mathbf{x}) &= p(\mathbf{x}_1, \mathbf{x}_2) \\ &= \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right] \\ &= \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left[-\frac{1}{2} f(\mathbf{x}_1, \mathbf{x}_2) \right] \end{aligned}$$

where f is defined as

$$\begin{aligned} f(\mathbf{x}_1, \mathbf{x}_2) &= (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= [(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T, (\mathbf{x}_2 - \boldsymbol{\mu}_2)^T] \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 - \boldsymbol{\mu}_1 \\ \mathbf{x}_2 - \boldsymbol{\mu}_2 \end{bmatrix} \\ &= (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \Sigma^{11} (\mathbf{x}_1 - \boldsymbol{\mu}_1) + 2(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \Sigma^{12} (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &\quad + (\mathbf{x}_2 - \boldsymbol{\mu}_2)^T \Sigma^{22} (\mathbf{x}_2 - \boldsymbol{\mu}_2) \end{aligned}$$

in which

$$\Sigma^{-1} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix}.$$

According to Theorem A.2,

$$\Sigma^{11} = (\Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{12}^T)^{-1} = \Sigma_{11}^{-1} + \Sigma_{11}^{-1} \Sigma_{12} (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \Sigma_{12}^T \Sigma_{11}^{-1},$$

$$\Sigma^{22} = (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} = \Sigma_{22}^{-1} + \Sigma_{22}^{-1} \Sigma_{12}^T (\Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{12}^T)^{-1} \Sigma_{12} \Sigma_{22}^{-1},$$

and

$$\Sigma^{12} = -\Sigma_{11}^{-1} \Sigma_{12} (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} = (\Sigma^{21})^T.$$

Substituting the expressions for Σ^{11} , Σ^{22} and Σ^{12} into $f(\mathbf{x}_1, \mathbf{x}_2)$, one obtains

$$\begin{aligned}
f(\mathbf{x}_1, \mathbf{x}_2) &= (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \left[\Sigma_{11}^{-1} + \Sigma_{11}^{-1} \Sigma_{12} (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \Sigma_{12}^T \Sigma_{11}^{-1} \right] (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
&\quad - 2 (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \left[\Sigma_{11}^{-1} \Sigma_{12} (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \right] (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\
&\quad + (\mathbf{x}_2 - \boldsymbol{\mu}_2)^T \left[(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \right] (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\
&= (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \Sigma_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
&\quad + (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \left[\Sigma_{11}^{-1} \Sigma_{12} (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \Sigma_{12}^T \Sigma_{11}^{-1} \right] (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
&\quad - 2 (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \left[\Sigma_{11}^{-1} \Sigma_{12} (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \right] (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\
&\quad + (\mathbf{x}_2 - \boldsymbol{\mu}_2)^T \left[(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \right] (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\
&= (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \Sigma_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\
&\quad + [(\mathbf{x}_2 - \boldsymbol{\mu}_2) - \Sigma_{12}^T \Sigma_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1)]^T (\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})^{-1} \\
&\quad \times [(\mathbf{x}_2 - \boldsymbol{\mu}_2) - \Sigma_{12}^T \Sigma_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1)] .
\end{aligned}$$

The last equality is due to the following equations for any vectors \mathbf{u} and \mathbf{v} and a symmetric matrix $\mathbf{A} = \mathbf{A}^T$:

$$\begin{aligned}
&\mathbf{u}^T \mathbf{A} \mathbf{u} - 2 \mathbf{u}^T \mathbf{A} \mathbf{v} + \mathbf{v}^T \mathbf{A} \mathbf{v} = \mathbf{u}^T \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{A} \mathbf{v} - \mathbf{u}^T \mathbf{A} \mathbf{v} + \mathbf{v}^T \mathbf{A} \mathbf{v} \\
&= \mathbf{u}^T \mathbf{A} (\mathbf{u} - \mathbf{v}) - (\mathbf{u} - \mathbf{v})^T \mathbf{A} \mathbf{v} = \mathbf{u}^T \mathbf{A} (\mathbf{u} - \mathbf{v}) - \mathbf{v}^T \mathbf{A} (\mathbf{u} - \mathbf{v}) \\
&= (\mathbf{u} - \mathbf{v})^T \mathbf{A} (\mathbf{u} - \mathbf{v}) = (\mathbf{v} - \mathbf{u})^T \mathbf{A} (\mathbf{v} - \mathbf{u}) .
\end{aligned}$$

Let us define

$$\mathbf{b} \triangleq \boldsymbol{\mu}_2 + \Sigma_{12}^T \Sigma_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) ,$$

$$\mathbf{A} \triangleq \Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} ,$$

and

$$\begin{aligned}
 f_1(\mathbf{x}_1) &\triangleq (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) , \\
 f_2(\mathbf{x}_1, \mathbf{x}_2) &\triangleq [(\mathbf{x}_2 - \boldsymbol{\mu}_2) - \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1)]^T (\boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12})^{-1} \\
 &\quad [(\mathbf{x}_2 - \boldsymbol{\mu}_2) - \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1)] \\
 &= (\mathbf{x}_2 - \mathbf{b})^T \mathbf{A}^{-1} (\mathbf{x}_2 - \mathbf{b}) .
 \end{aligned}$$

Therefore,

$$f(\mathbf{x}_1, \mathbf{x}_2) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_1, \mathbf{x}_2) .$$

Now, the joint distribution can be written as:

$$\begin{aligned}
 p(\mathbf{x}) &= p(\mathbf{x}_1, \mathbf{x}_2) \\
 &= \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left[-\frac{1}{2} f(\mathbf{x}_1, \mathbf{x}_2) \right] \\
 &= \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_{11}|^{1/2} |\boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}|^{1/2}} \exp \left[-\frac{1}{2} f(\mathbf{x}_1, \mathbf{x}_2) \right] \\
 &= \frac{1}{(2\pi)^{r/2} |\boldsymbol{\Sigma}_{11}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \right] \\
 &\quad \times \frac{1}{(2\pi)^{q/2} |\mathbf{A}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_2 - \mathbf{b})^T \mathbf{A}^{-1} (\mathbf{x}_2 - \mathbf{b}) \right] \\
 &= \mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \mathcal{N}(\mathbf{x}_2; \mathbf{b}, \mathbf{A}) .
 \end{aligned}$$

The third equality is due to Theorem A.3:

$$|\boldsymbol{\Sigma}| = |\boldsymbol{\Sigma}_{11}| |\boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}| .$$

The marginal distribution of \mathbf{x}_1 is

$$p(\mathbf{x}_1) = \int p(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 = \frac{1}{(2\pi)^{r/2} |\boldsymbol{\Sigma}_{11}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \right].$$

Similarly, it can be shown that the marginal distribution of \mathbf{x}_2 is

$$p(\mathbf{x}_2) = \int p(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1 = \frac{1}{(2\pi)^{q/2} |\boldsymbol{\Sigma}_{22}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_2 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_{22}^{-1} (\mathbf{x}_2 - \boldsymbol{\mu}_2) \right].$$

The conditional distribution of \mathbf{x}_2 given \mathbf{x}_1 is

$$p(\mathbf{x}_2|\mathbf{x}_1) = \frac{p(\mathbf{x}_1, \mathbf{x}_2)}{p(\mathbf{x}_1)} = \frac{1}{(2\pi)^{q/2} |\mathbf{A}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_2 - \mathbf{b})^T \mathbf{A}^{-1} (\mathbf{x}_2 - \mathbf{b}) \right]$$

with

$$\mathbf{b} = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1)$$

$$\mathbf{A} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}$$

Similarly, the conditional distribution of \mathbf{x}_1 given \mathbf{x}_2 is

$$p(\mathbf{x}_1|\mathbf{x}_2) = \frac{p(\mathbf{x}_2, \mathbf{x}_1)}{p(\mathbf{x}_2)} = \frac{1}{(2\pi)^{r/2} |\mathbf{B}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_1 - \mathbf{c})^T \mathbf{B}^{-1} (\mathbf{x}_1 - \mathbf{c}) \right]$$

with

$$\mathbf{c} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{21}^T \boldsymbol{\Sigma}_{22}^{-1} (\mathbf{x}_2 - \boldsymbol{\mu}_2)$$

$$\mathbf{B} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{21}^T \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21}$$

□

A.3 Theorems for the Derivation of the Kalman Filter

Theorem A.5.

$$\mathcal{N}(\mathbf{a}; \mathbf{F}\mathbf{b}, \mathbf{C}) \mathcal{N}(\mathbf{b}; \mathbf{m}, \mathbf{D}) = \mathcal{N}\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}; \begin{pmatrix} \mathbf{F}\mathbf{m} \\ \mathbf{m} \end{pmatrix}, \begin{pmatrix} \mathbf{F}\mathbf{D}\mathbf{F}^T + \mathbf{C} & \mathbf{F}\mathbf{D} \\ \mathbf{D}\mathbf{F}^T & \mathbf{D} \end{pmatrix}\right) \quad (\text{A.11})$$

Proof. The following definitions will be used

$$\mathbf{x} = \mathbf{b} - \mathbf{m},$$

$$\Sigma_{xx} = \mathbf{D},$$

$$\mathbf{y} = \mathbf{a} - \mathbf{F}\mathbf{m},$$

$$\Sigma_{yy} = \mathbf{C},$$

$$\mathbf{A} = \mathbf{F}.$$

Thus, the theorem may be rewritten as

$$\mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \Sigma_{yy}) \mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma_{xx}) = \mathcal{N}\left(\begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}; \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{A}\Sigma_{xx}\mathbf{A}^T + \Sigma_{yy} & \mathbf{A}\Sigma_{xx} \\ \Sigma_{xx}\mathbf{A}^T & \Sigma_{xx} \end{pmatrix}\right).$$

It is assumed that the dimensions of \mathbf{y} and \mathbf{x} are $g \times 1$ and $k \times 1$, respectively, and \mathbf{A}

is a $g \times k$ matrix. Expanding the product of the normal distributions:

$$\begin{aligned}
& \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \boldsymbol{\Sigma}_{yy}) \mathcal{N}(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_{xx}) \\
&= (2\pi)^{-g/2} |\boldsymbol{\Sigma}_{yy}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{A}\mathbf{x})^T \boldsymbol{\Sigma}_{yy}^{-1} (\mathbf{y} - \mathbf{A}\mathbf{x})\right) \\
&\quad \times (2\pi)^{-k/2} |\boldsymbol{\Sigma}_{xx}|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}^T \boldsymbol{\Sigma}_{xx}^{-1} \mathbf{x}\right) \\
&= (2\pi)^{-(g+k)/2} |\boldsymbol{\Sigma}_{yy}|^{-1/2} |\boldsymbol{\Sigma}_{xx}|^{-1/2} \\
&\quad \times \exp\left(-\frac{1}{2} \begin{pmatrix} \mathbf{y} - \mathbf{A}\mathbf{x} \\ \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\Sigma}_{yy} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{xx} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} - \mathbf{A}\mathbf{x} \\ \mathbf{x} \end{pmatrix}\right) \\
&= (2\pi)^{-(g+k)/2} |\boldsymbol{\Sigma}_{yy}|^{-1/2} |\boldsymbol{\Sigma}_{xx}|^{-1/2} \\
&\quad \times \exp\left(-\frac{1}{2} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \mathbf{I}_g & -\mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\Sigma}_{yy} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{xx} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I}_g & -\mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}\right) \\
&= (2\pi)^{-(g+k)/2} |\boldsymbol{\Sigma}_{yy}|^{-1/2} |\boldsymbol{\Sigma}_{xx}|^{-1/2} \\
&\quad \times \exp\left(-\frac{1}{2} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \mathbf{I}_g & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\Sigma}_{yy} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{xx} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I}_g & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}\right) \\
&= (2\pi)^{-(g+k)/2} |\boldsymbol{\Sigma}_{yy}|^{-1/2} |\boldsymbol{\Sigma}_{xx}|^{-1/2} \\
&\quad \times \exp\left(-\frac{1}{2} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}^T \left(\begin{pmatrix} \mathbf{I}_g & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_{yy} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{xx} \end{pmatrix} \begin{pmatrix} \mathbf{I}_g & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}^T \right)^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}\right) \\
&= (2\pi)^{-(g+k)/2} |\boldsymbol{\Sigma}_{yy}|^{-1/2} |\boldsymbol{\Sigma}_{xx}|^{-1/2} \\
&\quad \times \exp\left(-\frac{1}{2} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \mathbf{A}\boldsymbol{\Sigma}_{xx}\mathbf{A}^T + \boldsymbol{\Sigma}_{yy} & \mathbf{A}\boldsymbol{\Sigma}_{xx} \\ \boldsymbol{\Sigma}_{xx}\mathbf{A}^T & \boldsymbol{\Sigma}_{xx} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}\right).
\end{aligned}$$

For the above derivation, the following identity was used:

$$\begin{pmatrix} \mathbf{I}_g & -\mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{I}_g & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}.$$

Using Theorem A.3,

$$\left| \begin{pmatrix} \mathbf{A}\Sigma_{xx}\mathbf{A}^T + \Sigma_{yy} & \mathbf{A}\Sigma_{xx} \\ \Sigma_{xx}\mathbf{A}^T & \Sigma_{xx} \end{pmatrix} \right| = |\Sigma_{xx}| |\Sigma_{yy}|$$

which leads to

$$\begin{aligned} & \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \Sigma_{yy}) \mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma_{xx}) \\ &= (2\pi)^{-(g+k)/2} \left| \begin{pmatrix} \mathbf{A}\Sigma_{xx}\mathbf{A}^T + \Sigma_{yy} & \mathbf{A}\Sigma_{xx} \\ \Sigma_{xx}\mathbf{A}^T & \Sigma_{xx} \end{pmatrix} \right|^{-1/2} \\ & \quad \times \exp \left(-\frac{1}{2} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \mathbf{A}\Sigma_{xx}\mathbf{A}^T + \Sigma_{yy} & \mathbf{A}\Sigma_{xx} \\ \Sigma_{xx}\mathbf{A}^T & \Sigma_{xx} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix} \right) \\ &= \mathcal{N} \left(\begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix}; \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{A}\Sigma_{xx}\mathbf{A}^T + \Sigma_{yy} & \mathbf{A}\Sigma_{xx} \\ \Sigma_{xx}\mathbf{A}^T & \Sigma_{xx} \end{pmatrix} \right) \end{aligned}$$

□

Theorem A.6.

$$\begin{aligned} & \int \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}^a, \mathbf{P}_{k-1}^a) d\mathbf{x}_{k-1} \\ &= \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}^a, \mathbf{F}_{k-1}^T \mathbf{P}_{k-1}^a \mathbf{F}_{k-1} + \mathbf{Q}_{k-1}) \end{aligned} \quad (\text{A.12})$$

Proof. According to Theorem A.5,

$$\begin{aligned} & \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}^a, \mathbf{P}_{k-1}^a) \\ &= \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{pmatrix}; \begin{pmatrix} \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}^a \\ \hat{\mathbf{x}}_{k-1}^a \end{pmatrix}, \begin{pmatrix} \mathbf{F}_{k-1}\mathbf{P}_{k-1}^a\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} & \mathbf{F}_{k-1}\mathbf{P}_{k-1}^a \\ \mathbf{P}_{k-1}^a\mathbf{F}_{k-1}^T & \mathbf{P}_{k-1}^a \end{pmatrix}\right) \end{aligned}$$

Theorem A.4 provides the marginal distribution of \mathbf{x}_k :

$$p(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}^a, \mathbf{F}_{k-1}^T \mathbf{P}_{k-1}^a \mathbf{F}_{k-1} + \mathbf{Q}_{k-1})$$

which means

$$\begin{aligned} & \int \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}^a, \mathbf{P}_{k-1}^a) d\mathbf{x}_{k-1} \\ &= \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}^a, \mathbf{F}_{k-1}^T \mathbf{P}_{k-1}^a \mathbf{F}_{k-1} + \mathbf{Q}_{k-1}) \end{aligned}$$

□

A.4 Integral of a Product of Two Gaussian Functions

The multiplication of two Gaussian functions of a vector \mathbf{x} of dimension d given by $\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})$ and $\mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{B})$ is a Gaussian function [183]. In particular

$$\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) \mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{B}) = z \mathcal{N}(\mathbf{x}; \mathbf{c}, \mathbf{C}) \quad (\text{A.13})$$

where

$$\mathbf{C} = [\mathbf{A}^{-1} + \mathbf{B}^{-1}]^{-1}, \quad (\text{A.14})$$

$$\mathbf{c} = \mathbf{C} [\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}], \quad (\text{A.15})$$

$$z = (2\pi)^{-d/2} |\mathbf{C}|^{1/2} |\mathbf{A}|^{-1/2} |\mathbf{B}|^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{a}'\mathbf{A}^{-1}\mathbf{a} + \mathbf{b}'\mathbf{B}^{-1}\mathbf{b} - \mathbf{c}'\mathbf{C}^{-1}\mathbf{c}) \right]. \quad (\text{A.16})$$

Thus the integral of a product of two Gaussian functions of random vector \mathbf{x} of dimension d has the following closed form solution

$$\int \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A}) \mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{B}) \mathbf{x} = (2\pi)^{-d/2} |\mathbf{C}|^{1/2} |\mathbf{A}|^{-1/2} |\mathbf{B}|^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{a}'\mathbf{A}^{-1}\mathbf{a} + \mathbf{b}'\mathbf{B}^{-1}\mathbf{b} - \mathbf{c}'\mathbf{C}^{-1}\mathbf{c}) \right] \quad (\text{A.17})$$

Appendix B

Newmark-Beta method for Time-marching of Coupled ODEs

The book by Zienkiewicz [166] is closely followed for the derivation of this method. The coupled set of ordinary differential equations (ODEs) given by

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + \mathbf{C}\dot{\mathbf{u}}_{n+1} + \mathbf{K}\mathbf{u}_{n+1} = \mathbf{f}_{n+1}, \quad (\text{B.1})$$

need to be integrated in time comprising the forecast step of many filtering techniques. The Newmark-Beta scheme is widely used to numerically integrate such system of ODEs. The Newmark-Beta method [168] states that the function \mathbf{u} and its first-order time derivative $\dot{\mathbf{u}}$ are given by

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_n + \frac{1}{2} (1 - \beta_2) \Delta t^2 \ddot{\mathbf{u}}_n + \frac{1}{2} \beta_2 \Delta t^2 \ddot{\mathbf{u}}_{n+1}, \quad (\text{B.2})$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + (1 - \beta_1) \Delta t \ddot{\mathbf{u}}_n + \beta_1 \Delta t \ddot{\mathbf{u}}_{n+1}. \quad (\text{B.3})$$

where β_1 and β_2 are some given parameters [166].

The above equations can be rewritten as

$$\ddot{\mathbf{u}}_{n+1} = \ddot{\hat{\mathbf{u}}}_{n+1} + \frac{2}{\beta_2 \Delta t^2} \mathbf{u}_{n+1}, \quad (\text{B.4})$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\hat{\mathbf{u}}}_{n+1} + \frac{2\beta_1}{\beta_2 \Delta t} \mathbf{u}_{n+1}, \quad (\text{B.5})$$

where

$$\ddot{\hat{\mathbf{u}}}_{n+1} = -\frac{2}{\beta_2 \Delta t^2} \mathbf{u}_n - \frac{2}{\beta_2 \Delta t} \dot{\mathbf{u}}_n - \frac{1 - \beta_2}{\beta_2} \ddot{\mathbf{u}}_n, \quad (\text{B.6})$$

$$\dot{\hat{\mathbf{u}}}_{n+1} = -\frac{2\beta_1}{\beta_2 \Delta t} \mathbf{u}_n + \left(1 - \frac{2\beta_1}{\beta_2}\right) \dot{\mathbf{u}}_n + \left(1 - \frac{\beta_1}{\beta_2}\right) \Delta t \ddot{\mathbf{u}}_n. \quad (\text{B.7})$$

Combining Eqs. B.6-B.7 with the dynamic equation Eq. (B.1), one can solve for \mathbf{u} at time instant $(n + 1)$ as

$$\mathbf{A} \mathbf{u}_{n+1} = \mathbf{f}_{n+1} \quad (\text{B.8})$$

where

$$\mathbf{A} = \frac{2}{\beta_2 \Delta t^2} \mathbf{M} + \frac{2\beta_1}{\beta_2 \Delta t} \mathbf{C} + \mathbf{K}, \quad (\text{B.9})$$

$$\mathbf{f}_{n+1} = \mathbf{f}_{n+1} - \mathbf{C} \dot{\hat{\mathbf{u}}}_{n+1} - \mathbf{M} \ddot{\hat{\mathbf{u}}}_{n+1}. \quad (\text{B.10})$$

Appendix C

Mathematical Representation of Random Processes

The following follows the book by Ghanem and Spanos [1], the paper by Sarkar and Ghanem [184], the paper by Sarkar *et al.* [82] and the papers by Subber and Sarkar [83–87].

C.1 Karhunen-Loeve Expansion

The Karhunen-Loeve expansion (KLE) of an arbitrary stochastic process $\alpha(\mathbf{x}, \theta)$ is based on the spectral expansion of its covariance function $R_{\alpha\alpha}(\mathbf{x}, \mathbf{y})$ where \mathbf{x} and \mathbf{y} represent geometric and temporal coordinates [185]. The KLE representation of the stochastic process $\alpha(\mathbf{x}, \theta)$ is given by [185]

$$\alpha(\mathbf{x}, \theta) = \bar{\alpha}(\mathbf{x}) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i(\theta) \phi_i(\mathbf{x}), \quad (\text{C.1})$$

where $\bar{\alpha}(\mathbf{x})$ is the mean of $\alpha(\mathbf{x}, \theta)$ and $\{\xi_i(\theta)\}$ denotes a set of orthogonal (but not generally independent) random variables. Furthermore, $\{\lambda_i\}$ and $\{\phi_i(\mathbf{x})\}$ are the eigenvalues and eigenvectors of the covariance function, described by the following integral eigenvalue problem

$$\int_{\mathcal{D}} R_{\alpha\alpha}(\mathbf{x}, \mathbf{y}) \phi_i(\mathbf{y}) d\mathbf{y} = \lambda_i \phi_i(\mathbf{x}), \quad (\text{C.2})$$

where \mathcal{D} denotes the domain over which the process $\alpha(\mathbf{x}, \theta)$ is defined.

Using KLE, the random process $\alpha(\mathbf{x}, \theta)$ is approximated by a combination of deterministic shape functions $\phi_i(\mathbf{x})$ having random amplitudes $\sqrt{\lambda_i} \xi_i(\theta)$. For broad-band processes (i.e. processes closer to white noise), a significant number of scales (frequencies) are present in the stochastic process, thereby increasing the number of terms in its KLE representation. For numerous engineering and natural systems, the system parameters may exhibit smooth variations whereby only a few dominant scales of fluctuations effectively contributes to the process $\alpha(\mathbf{x}, \theta)$. For such narrow-banded processes, KLE contains only a few terms to capture the randomness effectively.

C.1.1 Integral Eigenvalue Problem of Exponential Covariance Function

The following derivation of the eigenfunctions and eigenvalues of the exponential covariance function follows closely Ref. [1]. The Karhunen-Loève Expansion (KLE) of a two-dimensional stochastic process is of the form

$$\alpha(x_1, x_2, \theta) = \alpha_0(x_1, x_2) + \sum_{n=1}^{\infty} \sqrt{\lambda_n} f_n(x_1, x_2) \xi_n(\theta), \quad (\text{C.3})$$

in which $\alpha_0(x_1, x_2)$ represents the mean process, λ_n and $f_n(x_1, x_2)$ are the eigenpair of the covariance function, and ξ_n could be obtained using

$$\xi_n(\theta) = \frac{1}{\sqrt{\lambda_n}} \int \alpha(x_1, x_2, \theta) f_n(x_1, x_2) dx_1 dx_2. \quad (\text{C.4})$$

In this thesis, the exponential covariance function is used having the form

$$C(x_1, x_2; y_1, y_2) = \sigma^2 e^{-|x_1 - y_1|/b_1 - |x_2 - y_2|/b_2}, \quad (\text{C.5})$$

in which σ^2 is the variance and b_1 and b_2 are the correlations lengths along the x_1 and x_2 directions, respectively. It is assumed that the process is defined over the two-dimensional domain $(-a_1, a_1) \times (-a_2, a_2)$.

The eigenvalues and eigenfunctions of the covariance kernel form the solution pair to the following integral eigenvalue problem

$$\lambda_n f_n(x_1, x_2) = \int C(x_1, x_2; y_1, y_2) f_n(y_1, y_2) dy_1 dy_2. \quad (\text{C.6})$$

The following form of the eigenfunctions and eigenvalues are assumed:

$$f_n(x_1, x_2) = f_i^{(1)}(x_1) f_j^{(2)}(x_2) \quad (\text{C.7})$$

and

$$\lambda_n = \lambda_i^{(1)} \lambda_j^{(2)}. \quad (\text{C.8})$$

These assumptions lead to the following analytical solutions for $f_i^{(1)}(x_1)$ and $\lambda_i^{(1)}$ [1]

$$\lambda_i^{(1)} = \frac{2b_1}{1 + b_1^2 \omega_i^2}, \quad (\text{C.9})$$

and

$$f_i^{(1)}(x_1) = \frac{\cos(\omega_i x_1)}{\sqrt{a_1 + \frac{\sin(2\omega_i a_1)}{2\omega_i}}} \quad \text{for } i \text{ odd ,} \quad (\text{C.10})$$

$$f_i^{(1)}(x_1) = \frac{\cos(\omega_i x_1)}{\sqrt{a_1 - \frac{\sin(2\omega_i a_1)}{2\omega_i}}} \quad \text{for } i \text{ even ,} \quad (\text{C.11})$$

with ω_i satisfying the following equations

$$\frac{1}{b_1} - \omega_i \tan(\omega_i a_1) = 0 \quad \text{for } i \text{ odd ,} \quad (\text{C.12})$$

$$\omega_i + \frac{1}{b_1} \tan(\omega_i a_1) = 0 \quad \text{for } i \text{ even .} \quad (\text{C.13})$$

$f_j^{(2)}(x_2)$ and $\lambda_j^{(2)}$ have the same analytical expressions, with b_1 , a_1 and x_1 replaced by b_2 , a_2 and x_2 , respectively. Within this thesis, it will be assumed that $b_1 = b_2$, in which case there will be two functions of the form Eq. (C.7) with the same eigenvalue.

C.2 Polynomial Chaos Expansion

The Karhunen-Loeve expansion can not be used to represent a stochastic process, such as the random system output, when its covariance is not known. In such cases, the output process can be expressed as the nonlinear functional of the input random variables $\{\xi_i(\theta)\}$ using the polynomial chaos expansion (PCE) [1, 81] as

$$u(\theta) = a_0 \Gamma_0 + \sum_{i_1=1}^{\infty} a_{i_1} \Gamma_1(\xi_{i_1}(\theta)) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} \Gamma_2(\xi_{i_1}(\theta), \xi_{i_2}(\theta)) + \dots, \quad (\text{C.14})$$

where the symbol $\Gamma_n(\xi_{i_1}, \dots, \xi_{i_n})$ are called the Polynomial Chaoes [1, 81] of order n in the variables $(\xi_{i_1}, \dots, \xi_{i_n})$. The polynomial chaoes of order up to and including

three are [80]

$$\Gamma_0 = 1 , \quad (\text{C.15})$$

$$\Gamma_1 (\xi_{i_1}) = \xi_{i_1} , \quad (\text{C.16})$$

$$\Gamma_2 (\xi_{i_1}, \xi_{i_2}) = \xi_{i_1} \xi_{i_2} - \delta_{i_1 i_2} , \quad (\text{C.17})$$

$$\Gamma_3 (\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) = \xi_{i_1} \xi_{i_2} \xi_{i_3} - \xi_{i_1} \delta_{i_2 i_3} - \xi_{i_2} \delta_{i_1 i_3} - \xi_{i_3} \delta_{i_1 i_2} , \quad (\text{C.18})$$

where $\delta_{i_1 i_2}$ denotes the Kronecker delta defined as

$$\delta_{i_1 i_2} = \begin{cases} 1 & \text{if } i_1 = i_2 \\ 0 & \text{if } i_1 \neq i_2 \end{cases} . \quad (\text{C.19})$$

Concisely, the polynomial chaos expansion, truncated after the N th term, is represented as

$$u (\theta) = \sum_{i=0}^{N-1} u_i \Psi_i (\theta) . \quad (\text{C.20})$$

For example, the third order polynomial chaos expansion of a stochastic process u using three random variables $\{\xi_1, \xi_2, \xi_3\}$ is [1]

$$\begin{aligned} u = & u_0 + u_1 \xi_1 + u_2 \xi_2 + u_3 \xi_3 + \\ & u_4 (\xi_1^2 - 1) + u_5 (\xi_2^2 - 1) + u_6 (\xi_3^2 - 1) + \\ & u_7 \xi_1 \xi_2 + u_8 \xi_1 \xi_3 + u_9 \xi_2 \xi_3 + \\ & u_{10} (\xi_1^3 - 3\xi_1) + u_{11} (\xi_2^3 - 3\xi_2) + u_{12} (\xi_3^3 - 3\xi_3) + \\ & u_{13} (\xi_1^2 \xi_2 - \xi_2) + u_{14} (\xi_1^2 \xi_3 - \xi_3) + u_{15} (\xi_2^2 \xi_1 - \xi_1) + \\ & u_{16} (\xi_2^2 \xi_3 - \xi_3) + u_{17} (\xi_3^2 \xi_1 - \xi_1) + u_{18} (\xi_3^2 \xi_2 - \xi_2) + \\ & u_{19} \xi_1 \xi_2 \xi_3 . \end{aligned} \quad (\text{C.21})$$

i^{th} Polynomial Chaos	Order of the Homogeneous Chaos	Ψ_i	$\langle \Psi_i^2 \rangle$
0	0	1	1
1	1	ξ_1	1
2		ξ_2	1
3		ξ_3	1
4	2	$\xi_1^2 - 1$	2
5		$\xi_2^2 - 1$	2
6		$\xi_3^2 - 1$	2
7		$\xi_1 \xi_2$	1
8		$\xi_1 \xi_3$	1
9		$\xi_2 \xi_3$	1
10	3	$\xi_1^3 - 3\xi_1$	6
11		$\xi_2^3 - 3\xi_2$	6
12		$\xi_3^3 - 3\xi_3$	6
13		$\xi_1^2 \xi_2 - \xi_2$	2
14		$\xi_1^2 \xi_3 - \xi_3$	2
15		$\xi_2^2 \xi_1 - \xi_1$	2
16		$\xi_2^2 \xi_3 - \xi_3$	2
17		$\xi_3^2 \xi_1 - \xi_1$	2
18		$\xi_3^2 \xi_2 - \xi_2$	2
19		$\xi_1 \xi_2 \xi_3$	1

Table C.1: Polynomial chaoses for a random dimension of 2 of order up to and including 3 and their variances [1].

Table (C.1) shows the explicit expressions for the polynomials used in the above expansion of u . These polynomials are statistically orthogonal, i.e. $\langle \Psi_i(\theta) \Psi_j(\theta) \rangle = 0$ for $i \neq j$. $\{\Psi_i\}_{i=0}^{\infty}$ constitutes a complete set in the space of second order random variables [1]. The total number of terms (N) in a polynomial chaos expansion in Eq. (C.20) with order p and random dimension L is given by [80]

$$N = \frac{(L+p)!}{L! p!}. \quad (\text{C.22})$$

C.2.1 PCE Representation of Lognormal Stochastic Process

The derivation of the PCE coefficients for a lognormal process follows closely Ref. [186, 187]. Denoting the underlying Gaussian process by $g(\mathbf{x}, \theta)$ having a covariance function $C(x_1, x_2)$ with a variance of σ^2 , the KLE representation of this process is given by

$$g(\mathbf{x}, \theta) = g_0(\mathbf{x}) + \sum_{n=1}^3 \sqrt{\lambda_n} f_n(\mathbf{x}) \xi_n(\theta) . \quad (\text{C.23})$$

The above equation can be compactly represented as

$$g(\mathbf{x}, \theta) = g_0(\mathbf{x}) + \sum_{n=1}^L g_n(\mathbf{x}) \xi_n, \quad (\text{C.24})$$

where $g_n(\mathbf{x}) = \sqrt{\lambda_i} f_n(\mathbf{x})$.

The lognormal stochastic process $l(\mathbf{x}, \theta) = \exp [g(\mathbf{x}, \theta)]$ can be rewritten using the KLE of the Gaussian process in Eq. (C.24) as

$$l(\mathbf{x}, \theta) = \exp \left[g_0(\mathbf{x}) + \sum_{i=1}^L g_i(\mathbf{x}) \xi_i \right] . \quad (\text{C.25})$$

The PCE of the lognormal process $l(\mathbf{x}, \theta)$ has the following form

$$l(\mathbf{x}, \theta) = \sum_{i=0}^P l_i(\mathbf{x}) \Psi_i(\boldsymbol{\xi}) , \quad (\text{C.26})$$

in which the coefficients $l_i(\mathbf{x})$ are obtained via Galerkin projection as

$$l_i(\mathbf{x}) = \frac{\langle l(\mathbf{x}, \theta) \Psi_i(\boldsymbol{\xi}) \rangle}{\langle \Psi_i^2(\boldsymbol{\xi}) \rangle} . \quad (\text{C.27})$$

The denominator in Eq.(C.27) can be evaluated analytically (see Appendix C.2) and the numerator can be expressed as [187]

$$\langle l(\mathbf{x}, \theta) \Psi_i(\boldsymbol{\xi}) \rangle = \int_{-\infty}^{+\infty} \exp \left[g_0(\mathbf{x}) + \sum_{j=1}^L g_j(\mathbf{x}) \xi_j \right] \Psi_i(\boldsymbol{\xi}) \exp \left[-\frac{1}{2} \sum_{j=1}^L \xi_j^2 \right] d\boldsymbol{\xi}, \quad (\text{C.28})$$

which can be simplified to

$$\langle l(\mathbf{x}, \theta) \Psi_i(\boldsymbol{\xi}) \rangle = \exp \left[g_0(\mathbf{x}) + \frac{1}{2} \sum_{j=1}^L g_j^2(\mathbf{x}) \right] \langle \Psi_j(\boldsymbol{\eta}) \rangle, \quad (\text{C.29})$$

where

$$\eta_j = \xi_j - g_j(\mathbf{x}). \quad (\text{C.30})$$

Eq.(C.29) can be expressed concisely as

$$\langle l(\mathbf{x}, \theta) \Psi_i(\boldsymbol{\xi}) \rangle = l_0(\mathbf{x}) \langle \Psi_j(\boldsymbol{\eta}) \rangle, \quad (\text{C.31})$$

where $l_0(\mathbf{x})$ is the mean of the lognormal stochastic process given by

$$l_0(\mathbf{x}) = \exp \left[g_0(\mathbf{x}) + \frac{1}{2} \sum_{i=1}^L g_i^2(\mathbf{x}) \right]. \quad (\text{C.32})$$

Note that in the limit as $L \rightarrow \infty$, the mean lognormal process converges to

$$l_0(\mathbf{x}) = \exp \left[g_0(\mathbf{x}) + \frac{1}{2} \sigma^2 \right]. \quad (\text{C.33})$$

The quantity $\langle \Psi_j(\boldsymbol{\eta}) \rangle$ is the expectation of the PC basis around the Gaussian coefficients $g_i(\mathbf{x})$. This expectation can be evaluated analytically and the results are given in Table C.2 for $L = 3$.

$\Psi_i(\xi)$	$\langle \Psi_i^2(\xi) \rangle$	$\langle \Psi_i(\eta) \rangle$
ξ_1	1	$g_1(\mathbf{x})$
ξ_2	1	$g_2(\mathbf{x})$
ξ_3	1	$g_3(\mathbf{x})$
$\xi_1^2 - 1$	2	$g_1^2(\mathbf{x})$
$\xi_2^2 - 1$	2	$g_2^2(\mathbf{x})$
$\xi_3^2 - 1$	2	$g_3^2(\mathbf{x})$
$\xi_1 \xi_2$	1	$g_1(\mathbf{x}) g_2(\mathbf{x})$
$\xi_1 \xi_3$	1	$g_1(\mathbf{x}) g_3(\mathbf{x})$
$\xi_2 \xi_3$	1	$g_2(\mathbf{x}) g_3(\mathbf{x})$
$\xi_1^3 - 3\xi_1$	6	$g_1^3(\mathbf{x})$
$\xi_2^3 - 3\xi_2$	6	$g_2^3(\mathbf{x})$
$\xi_3^3 - 3\xi_3$	6	$g_3^3(\mathbf{x})$
$\xi_1^2 \xi_2 - \xi_2$	2	$g_1^2(\mathbf{x}) g_2(\mathbf{x})$
$\xi_1^2 \xi_3 - \xi_3$	2	$g_1^2(\mathbf{x}) g_3(\mathbf{x})$
$\xi_2^2 \xi_1 - \xi_1$	2	$g_2^2(\mathbf{x}) g_1(\mathbf{x})$
$\xi_2^2 \xi_3 - \xi_3$	2	$g_2^2(\mathbf{x}) g_3(\mathbf{x})$
$\xi_3^2 \xi_1 - \xi_1$	2	$g_3^2(\mathbf{x}) g_1(\mathbf{x})$
$\xi_3^2 \xi_2 - \xi_2$	2	$g_3^2(\mathbf{x}) g_2(\mathbf{x})$
$\xi_1 \xi_2 \xi_3$	1	$g_1(\mathbf{x}) g_2(\mathbf{x}) g_3(\mathbf{x})$

Table C.2: The expectation and variance of the PC basis.

Thus, the PCE of the lognormal stochastic process can be written as

$$l(\mathbf{x}, \theta) = l_0(\mathbf{x}) \sum_{i=0}^L \frac{\langle \Psi_i(\eta) \rangle}{\langle \Psi_i^2(\xi) \rangle} \Psi_i(\xi). \quad (\text{C.34})$$

Appendix D

Parallel Iterative Solvers for the Extended Schur Complement System

In the context of numerical solutions of partial differential equations, domain decomposition methods tackle a boundary value problem over a domain of interest by solving a set of smaller problems on a number of subdomains [173, 188–190]. In the context of non-overlapping domain decomposition methods, the resulting Schur complement system describes the behaviour of the collection of unknowns forming the interfaces between these subdomains [173, 188–190]. The solution of the Schur complement system is obtained using iterative methods [174]. During each iteration, the problem on each subdomain is tackled independently, leading to data parallelism. Typically, preconditioned Krylov subspace methods are utilized to solve the Schur complement system iteratively [174]. From this perspective, domain decomposition methods provide parallel preconditioners for the Krylov subspace iterative methods to solve the Schur complement system [173, 188–190].

Krylov subspace methods can be categorized into two groups: one applicable to symmetric positive-definite systems and another to non-symmetric systems. Within the

scope of this thesis, the conjugate gradient method (CGM) [174, 191, 192] will be used for the first class of systems. For non-symmetric systems, CGM can not guarantee orthogonality of the residual vectors [193] and thus can not be applied directly. The Generalized Minimum Residual Method (GMRES) [174, 192, 194] guarantees orthogonality of the residual vectors while requiring the storage of all previously computed residual vectors. Thus GMRES becomes memory intensive especially for large scale systems. The Biconjugate Gradient Method (BICG) provides a non-optimal approximation of the solution vector in the same Krylov subspace as GMRES, but with less computational work per iteration [195]. Furthermore, BICG does not need the storage of all previous residuals in contrast to GMRES, and thus memory requirement does not grow with the number of iterations. BICG achieves this by replacing the orthogonal sequence of residual vectors in GMRES by two mutually orthogonal sequences, one based on the original system matrix and the other on its transpose [192]. The applicability of BICG to non-symmetric systems is limited due to its irregular convergence behaviour [192]. The Conjugate Gradient Squared algorithm (CGS) [196] improves upon BICG by (a) avoiding the need for matrix-vector product steps involving the transpose of the system matrix, and (b) providing better convergence characteristics with similar computational complexity as BICG. The Biconjugate Gradient Stabilized method (BICGSTAB) [174, 192, 197] was developed to remedy some of the numerical problems associated with CGS, resulting in a method with a more regular convergence property. Due to its attractive features, BICGSTAB will be parallelized with the aid of DDM to provide a solver for large scale non-symmetric systems investigated within this thesis. For both CGM and BICGSTAB, the Schur complement matrix will not be constructed explicitly. Instead, the effect of the Schur complement matrix on a vector (i.e. a matrix-vector product) is computed in parallel.

The performance of Krylov subspace methods depends heavily on the spectral properties of the system matrix [174], i.e. the distribution of eigenvalues. A transformation of the original system, known as preconditioning, ensues more favorable spectral characteristics [174]. Thus, preconditioning is a commonly used technique to improve the convergence attributes of Krylov subspace methods. Preconditioning is performed with the aid of a preconditioning matrix, or preconditioner, \mathcal{M} . When the preconditioner is applied from the left, the preconditioned extended Schur complement system in Eq. (6.20) has the following form:

$$\mathcal{M}^{-1}\mathcal{S} \mathcal{X}_r = \mathcal{M}^{-1}\mathcal{G}_r . \quad (\text{D.1})$$

The algorithms for preconditioned CGM and preconditioned BICGSTAB to solve the Schur complement system of equations are given in Algorithms D.1 and D.2, respectively [174, 192].

<p>Input: Initial iterate \mathcal{X}_{r_0} Output: Interface solution \mathcal{X}_r</p> <pre> 1 $r_0 = \mathcal{G}_r - \mathcal{S} \mathcal{X}_{r_0}$ 2 for $i = 0, 1, \dots$, do: 3 $z_i = \mathcal{M}^{-1} r_i$ 4 $\rho_i = (r_i, z_i)$ 5 if $i = 0$ then 6 $p_{i+1} = z_i$ 7 else 8 $\beta_i = \rho_i / \rho_{i-1}$ 9 $p_{i+1} = z_i + \beta_i p_i$ 10 end if 11 $q_{i+1} = \mathcal{S} p_{i+1}$ 12 $\alpha_{i+1} = \rho_i / (p_{i+1}, q_{i+1})$ 13 $\mathcal{X}_{r_{i+1}} = \mathcal{X}_{r_i} + \alpha_{i+1} p_{i+1}$ 14 $r_{i+1} = r_i - \alpha_{i+1} q_{i+1}$ 15 if iterate has converged, exit 16 end do </pre>

Algorithm D.1: Preconditioned Conjugate Gradient Method.

A parallel preconditioner is critical component of an efficient solver for large scale

```

Input: Initial iterate  $\mathcal{X}_{r_0}$ 
Output: Interface solution  $\mathcal{X}_r$ 
1  $\mathbf{r}_0 = \mathcal{G}_r - \mathcal{S} \mathcal{X}_{r_0}$ 
2 Choose  $\tilde{\mathbf{r}}$  such that  $(\tilde{\mathbf{r}}, \mathbf{r}_0) \neq 0$ , for example  $\tilde{\mathbf{r}} = \mathbf{r}_0$ 
3 for  $i = 0, 1, \dots$ , do:
4      $\rho_i = (\tilde{\mathbf{r}}, \mathbf{r}_i)$ 
5     if  $\rho_i = 0$  method fails
6     if  $i = 0$  then
7          $\mathbf{p}_{i+1} = \mathbf{r}_i$ 
8     else
9          $\beta_i = (\rho_i / \rho_{i-1}) (\alpha_i / \omega_i)$ 
10         $\mathbf{p}_{i+1} = \mathbf{r}_i + \beta_i (\mathbf{p}_i - \omega_i \mathbf{v}_i)$ 
11    end if
12     $\hat{\mathbf{p}} = \mathcal{M}^{-1} \mathbf{p}_{i+1}$ 
13     $\mathbf{v}_{i+1} = \mathcal{S} \hat{\mathbf{p}}$ 
14     $\alpha_{i+1} = \rho_i / (\tilde{\mathbf{r}}, \mathbf{v}_{i+1})$ 
15     $\mathbf{s} = \mathbf{r}_i - \alpha_{i+1} \mathbf{v}_{i+1}$ 
16    if norm of  $\mathbf{s}$  is small, set  $\mathcal{X}_{r_{i+1}} = \mathcal{X}_{r_i} + \alpha_{i+1} \hat{\mathbf{p}}$  and exit
17     $\hat{\mathbf{s}} = \mathcal{M}^{-1} \mathbf{s}$ 
18     $\mathbf{t} = \mathcal{S} \hat{\mathbf{s}}$ 
19     $\omega_{i+1} = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$ 
20     $\mathcal{X}_{r_{i+1}} = \mathcal{X}_{r_i} + \alpha_{i+1} \hat{\mathbf{p}} + \omega_{i+1} \hat{\mathbf{s}}$ 
21     $\mathbf{r}_{i+1} = \mathbf{s} - \omega_{i+1} \mathbf{t}$ 
22    if iterate has converged, exit
23 end do

```

Algorithm D.2: Preconditioned Biconjugate Gradient Stabilized Method.

systems. A class of parallel preconditioners requires only distributed matrix-vector products [174]. For optimal convergence behavior of the iterative solver, these preconditioners should mimic the extended Schur complement matrix as close as possible, i.e. $\mathcal{M} \approx \mathcal{S}$. As a result, the eigenvalues of the preconditioned system $\mathcal{M}^{-1}\mathcal{S}$ would be distributed around unity, resulting in a smaller condition number and faster convergence of the iterative solver [174]. The one-level lumped preconditioner used in this thesis is one such preconditioner, and its parallel implementation will be discussed next.

D.1 Parallel One-Level Lumped Preconditioner

Starting with the extended Schur complement matrix defined in Eq. (6.21):

$$\mathcal{S} = \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{A}_{\Gamma\Gamma}^s - \mathcal{A}_{\Gamma I}^s (\mathcal{A}_{II}^s)^{-1} \mathcal{A}_{I\Gamma}^s] \mathcal{R}_s, \quad (\text{D.2})$$

a one-level preconditioner can be constructed using only the local interface coupling matrices $\mathcal{A}_{\Gamma\Gamma}^s$ in the following form [181]:

$$\mathcal{M}^{-1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} \mathcal{R}_s. \quad (\text{D.3})$$

Once the inverse of the local matrices $\mathcal{A}_{\Gamma\Gamma}^s$ is computed, this preconditioning step involves an MPI gather operation on local (subdomain level) matrix-vector products. Generally, the preconditioning step of Krylov subspace methods involves solving the following system of equations

$$\mathcal{M} \mathbf{z} = \mathbf{r} \quad (\text{D.4})$$

with \mathbf{r} denoting the so-called residual vector and \mathbf{z} the preconditioned residual. Equiva-

lently, the result of the following matrix-vector product is needed during each iteration:

$$\begin{aligned} \mathbf{z} &= \mathcal{M}^{-1} \mathbf{r} \\ &= \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} \mathcal{R}_s \mathbf{r}. \end{aligned} \quad (\text{D.5})$$

as outlined in Algorithm D.3 [84, 87]. Step 2 of Algorithm D.3 can be performed in parallel as it does not depend on data belonging to other subdomains. \mathbf{r}^s and \mathbf{z}^s are vectors that are defined locally at the subdomain level whereas \mathbf{r} and \mathbf{z} are globally defined vectors.

Input: Residual \mathbf{r}
Output: Preconditioned residual \mathbf{z}
1 **for** each processor $P_s, s = 1, 2, \dots, n_s$, **do**:
2 Scatter the residual: $\mathbf{r}^s = \mathcal{R}_s \mathbf{r}$
3 Compute $\mathbf{z}^s = [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} \mathbf{r}^s$
4 Gather the preconditioned residual: $\mathbf{z} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathbf{z}^s$
5 **end do**

Algorithm D.3: Parallel Preconditioning of Extended Schur Complement System.

The computationally intensive steps of the preconditioned CGM and preconditioned BICGSTAB methods, as outlined in Algorithms D.1 and D.2, are (a) obtaining the result of the matrix-vector product involving the Schur complement matrix in the form of $\mathbf{q} = \mathcal{S} \mathbf{p}$, and (b) computing the preconditioned residual vector, i.e. $\mathbf{z} = \mathcal{M}^{-1} \mathbf{r}$. The above parallel preconditioner makes use of iterative substructuring techniques to obtain the preconditioning effect at the subdomain (or local) level with the aid of MPI scatter and gather operations. Similarly, the matrix-vector product involving the Schur complement matrix can be parallelized as explained in the following section.

D.2 Parallel Matrix-Vector Product

One of the most computationally demanding operations in the preconditioned CGM and BICGSTAB methods for solving the Schur complement system of equations involves a matrix-vector product of the form

$$\begin{aligned} \mathbf{q} &= \mathcal{S} \mathbf{p} \\ &= \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{A}_{\Gamma\Gamma}^s - \mathcal{A}_{\Gamma I}^s (\mathcal{A}_{II}^s)^{-1} \mathcal{A}_{I\Gamma}^s] \mathcal{R}_s \mathbf{p}, \end{aligned} \quad (\text{D.6})$$

in which \mathcal{S} denotes the Schur complement matrix. For preconditioned CGM, this is performed in step 11 of Algorithm D.1. For preconditioned BICGSTAB, this operation is performed twice per iteration, in steps 13 and 18. Iterative substructuring methods [173, 181, 188–190] permit a parallelization of this operation without requiring the explicit construction of the Schur complement matrix as detailed in Algorithm D.4 [84, 87]. It is clear that the global matrix-vector product can be computed by solving local Dirichlet problems (step 4) in addition to three local matrix-vector products (steps 3, 5, and 6) followed by a gather operation (step 8) [181].

Input: Search direction vector \mathbf{p}
Output: Matrix-vector product \mathbf{q}

- 1 **for** each processor $P_s, s = 1, 2, \dots, n_s$, **do**:
- 2 Scatter the search direction vector: $\mathbf{p}^s = \mathcal{R}_s \mathbf{p}$
- 3 Compute $\mathbf{u}_1 = \mathcal{A}_{I\Gamma}^s \mathbf{p}^s$
- 4 Solve $\mathcal{A}_{II}^s \mathbf{u}_2 = \mathbf{u}_1$
- 5 Compute $\mathbf{u}_3 = \mathcal{A}_{\Gamma I}^s \mathbf{u}_2$
- 6 Compute $\mathbf{u}_4 = \mathcal{A}_{\Gamma\Gamma}^s \mathbf{p}^s$
- 7 Compute $\mathbf{q}^s = \mathbf{u}_4 - \mathbf{u}_3$
- 8 Gather the preconditioned residual: $\mathbf{q} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathbf{q}^s$
- 9 **end do**

Algorithm D.4: Parallel Matrix-Vector Product for the Extended Schur Complement System.

The parallelization of the matrix-vector product, along with the parallel precondi-

tioning algorithm detailed in the previous section, permits an efficient distributed implementation of Krylov subspace solvers to tackle the Schur complement system. Subber and Sarkar [84, 87, 181] developed a parallel preconditioned CGM solver for symmetric systems based on the preconditioner and matrix-vector product algorithms presented above as detailed in the next section. Similarly, a parallel preconditioned BICGSTAB solver is also developed to tackle non-symmetric systems as presented later.

D.3 Parallel Preconditioned Conjugate Gradient Method: Iterative Solver for the Extended Schur Complement System

Algorithms D.3 and D.4 are employed by Subber and Sarkar to develop a parallel version of the preconditioned CGM algorithm (Appendix D.1) as detailed in Algorithm D.5 [84, 87]. There are four global communication operations, two gathers (steps 4 and 16) and two scatters (steps 6 and 22). The operations performed locally, as indicated in the algorithm, are executed in parallel and the result is gathered at the head node.

D.3.1 Numerical Validation

For numerical validation of the parallel iterative solver based on the preconditioned conjugate gradient method, the stochastic diffusion problem is examined described as [176]:

$$\begin{aligned} -\nabla \cdot (c(\mathbf{x}, \theta) \nabla u(\mathbf{x}, \theta)) &= f(\mathbf{x}) && \text{in } D \times \Omega, \\ u(\mathbf{x}, \theta) &= 0 && \text{on } \partial D \times \Omega, \end{aligned} \tag{D.7}$$

```

Input: Initial iterate  $\mathcal{X}_{\Gamma_0}$ 
Output: Interface solution  $\mathcal{X}_{\Gamma}$ 
1  $\mathbf{r}_0 = \mathcal{G}_{\Gamma} - \mathcal{S} \mathcal{X}_{\Gamma_0}$ 
2 for  $i = 0, 1, \dots$ , do:
3   for each processor  $P_s, s = 1, 2, \dots, n_s$ , do:
4     Scatter:  $\mathbf{r}^s = \mathcal{R}_s \mathbf{r}_i$ 
5     Compute locally:  $\mathbf{z}^s = [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} \mathbf{r}^s$ 
6     Gather:  $\mathbf{z}_i = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathbf{z}^s$ 
7   end do
8    $\rho_i = (\mathbf{r}_i, \mathbf{z}_i)$ 
9   if  $i = 0$  then
10     $\mathbf{p}_{i+1} = \mathbf{z}_i$ 
11  else
12     $\beta_i = \rho_i / \rho_{i-1}$ 
13     $\mathbf{p}_{i+1} = \mathbf{z}_i + \beta_i \mathbf{p}_i$ 
14  end if
15  for each processor  $P_s, s = 1, 2, \dots, n_s$ , do:
16    Scatter:  $\mathbf{p}^s = \mathcal{R}_s \mathbf{p}_{i+1}$ 
17    Compute locally:  $\mathbf{u}_1 = \mathcal{A}_{\Gamma\Gamma}^s \mathbf{p}^s$ 
18    Solve locally:  $\mathcal{A}_{\Gamma I}^s \mathbf{u}_2 = \mathbf{u}_1$ 
19    Compute locally:  $\mathbf{u}_3 = \mathcal{A}_{\Gamma I}^s \mathbf{u}_2$ 
20    Compute locally:  $\mathbf{u}_4 = \mathcal{A}_{\Gamma\Gamma}^s \mathbf{p}^s$ 
21    Compute locally:  $\mathbf{q}^s = \mathbf{u}_4 - \mathbf{u}_3$ 
22    Gather:  $\mathbf{q}_{i+1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathbf{q}^s$ 
23  end do
24   $\alpha_{i+1} = \rho_i / (\mathbf{p}_{i+1}, \mathbf{q}_{i+1})$ 
25   $\mathcal{X}_{\Gamma_{i+1}} = \mathcal{X}_{\Gamma_i} + \alpha_{i+1} \mathbf{p}_{i+1}$ 
26   $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_{i+1} \mathbf{q}_{i+1}$ 
27  if iterate has converged, exit
28 end do

```

Algorithm D.5: Parallel Preconditioned Conjugate Gradient Method.

in which the square $D = (0, 1)^2$ represents the spatial domain for the boundary value problem. The random diffusivity c is modeled as a lognormal random variable, i.e. $c \sim \text{LogN}(\mu, \sigma^2)$, whose third-order truncated polynomial chaos expansion given by

$$c \approx \exp\left(\mu + \frac{\sigma^2}{2}\right) \left[1 + \sigma\xi + \sigma^2(\xi^2 - 1)/2 + \sigma^3(\xi^3 - 3\xi)/6\right]. \quad (\text{D.8})$$

Here, ξ is a Gaussian random variable having zero mean and unit variance. In this special case of space-invariant diffusion coefficient, the governing equation Eq. (D.7) can be rewritten as

$$-c(\theta)\nabla^2 u(\mathbf{x}, \theta) = f(\mathbf{x}). \quad (\text{D.9})$$

The deterministic source function is chosen as

$$f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y), \quad (\text{D.10})$$

leading to the following analytical solution for the response

$$u(x, y, \theta) = \frac{\sin(\pi x) \sin(\pi y)}{c(\theta)}. \quad (\text{D.11})$$

This solution satisfies both the governing equation and the homogeneous Dirichlet boundary value conditions. For $\mu = 1$ and $\sigma = 0.3$, 1000 samples of ξ are generated using Latin Hypercube Sampling (LHS) technique [121, 122] providing 1000 realizations of the diffusion coefficient c using Eq. (D.8). Finally, 1000 realizations of the solution process could be obtained using Eq. (D.11), from which the ensemble mean and standard deviation are obtained as shown in Figs. D.1.

The estimates in Figs. D.1 are used as approximations to the true mean and standard deviation for validation purposes. The spatial domain is discretized into a fi-

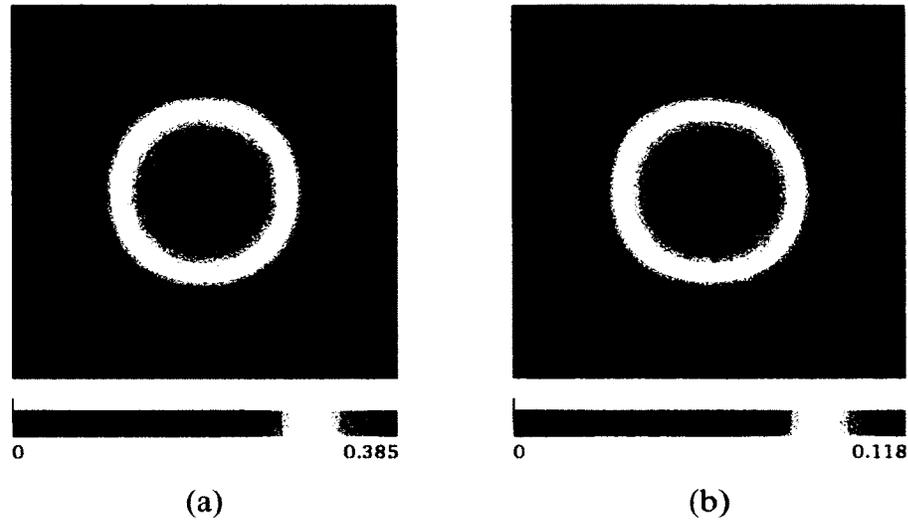


Figure D.1: Validation of parallel PCGM solver: (a) MC ensemble mean of analytical response (b) MC ensemble standard deviation of analytical response.

nite element mesh consisting of 19,694 linear triangular elements and 10,020 nodes as illustrated in Fig. D.2a. The spatial domain is decomposed into 20 subdomains as shown in Fig. D.2b resulting in a total of 344 global interface nodes. Using the parallel solver based on domain decomposition and preconditioned conjugate gradient method for SPDEs detailed in this section, the results in Fig. D.1 will be reproduced. A third order PCE expansion for the solution process is used and the coefficients of the solution process are obtained using the domain decomposition method. The Krylov-subspace iterate is initialized to zero. The stopping criteria for the PCGM solver is

$$\frac{\| \mathcal{X}_{r_{i+1}} - \mathcal{X}_{r_i} \|}{\| \mathcal{X}_{r_i} \|} \leq \epsilon = 10^{-10}. \quad (\text{D.12})$$

The zeroth order PCE coefficient representing the mean response is shown in Fig. D.3a with the error plotted in Fig. D.3b. One can see that the error is approximately three orders of magnitude smaller than the actual mean response. The main contributors to the

total error are (a) the finite element discretization error, (b) the MC sampling error in obtaining the “true” estimate, (c) the truncation error associated with a third order expansion of the solution process, and (d) the error associated with the termination of the PCGM iterations once the stopping criteria is met. The standard deviation in the solution process is next examined as a measure of uncertainty in the response. The covariance matrix is a function of the first through third order PCE coefficients as in Eq. (6.26). The standard deviation of the response, computed as the square roots of the covariance matrix diagonal entries, along with the error are shown in Fig. D.4. Again, the error is approximately three orders of magnitude smaller than the actual standard deviation.

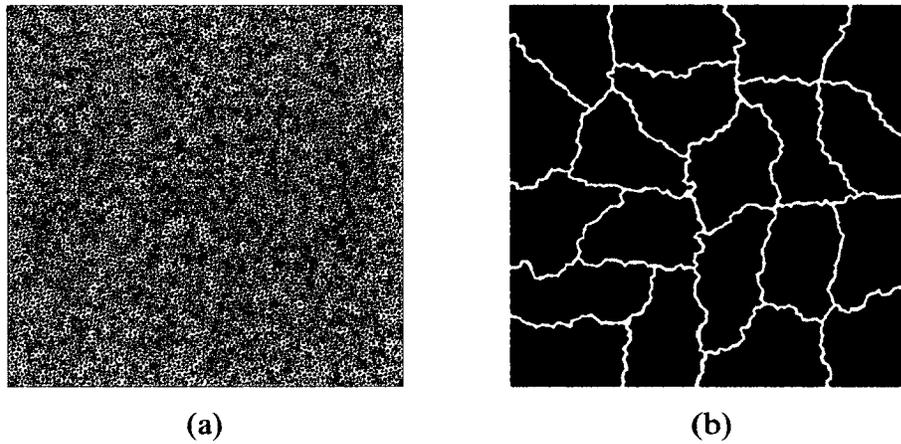


Figure D.2: Validation of parallel PCGM solver: (a) Finite element mesh and (b) the partitioned domain.

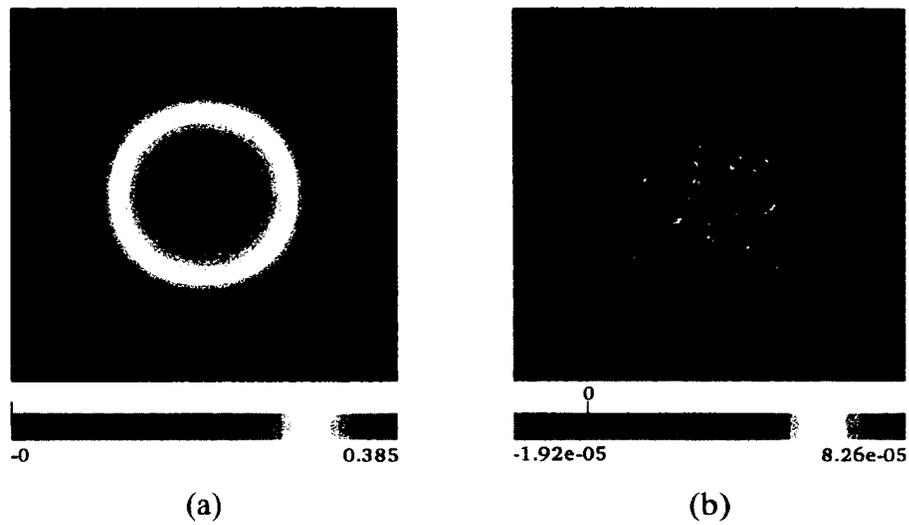


Figure D.3: Validation of parallel PCGM solver: (a) Mean of the solution process using the parallel solver based on domain decomposition and preconditioned conjugate gradient method and (b) associated error in comparison to MC ensemble mean of analytical response.

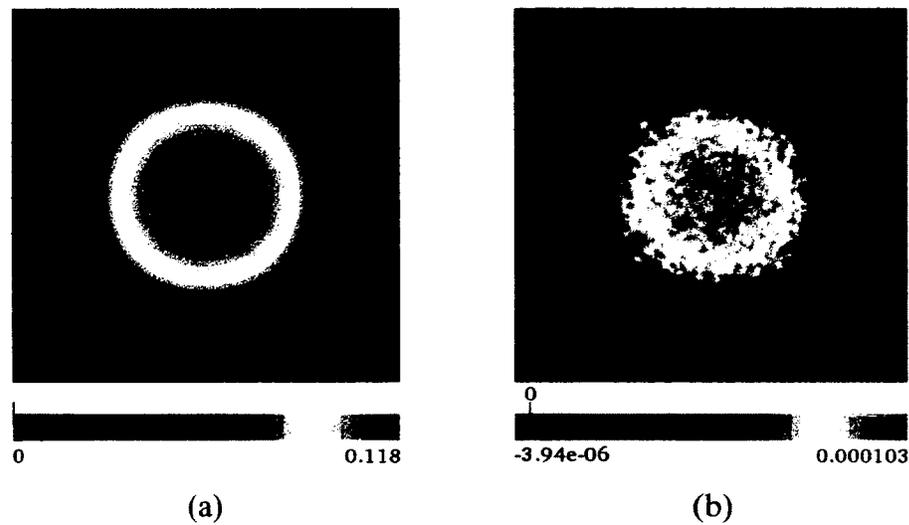


Figure D.4: Validation of parallel PCGM solver: (a) Standard deviation of the solution process using the parallel solver based on domain decomposition and preconditioned conjugate gradient method and (b) associated error in comparison to MC ensemble standard deviation of analytical response.

D.4 Parallel Preconditioned Biconjugate Gradient Stabilized Algorithm: Iterative Solver for the Extended Schur Complement System

As illustrated in the previous section, a parallel preconditioner and a parallel distributed algorithm for the matrix-vector product involving the Schur complement matrix can be exploited to obtain a parallel version of the preconditioned CGM solver. This solver will be used in this thesis for systems with symmetric matrices. In a similar fashion, the parallel preconditioning and matrix-vector product algorithms can be used to parallelize any Krylov subspace method. As discussed in Subsection D, BICGSTAB will be adapted to tackle large-scale non-symmetric systems. The following parallel version of preconditioned BICGSTAB method is used as outlined in Appendix D.2 exploiting the available parallel Algorithms D.3 and D.4. The parallel preconditioned BICGSTAB method is detailed in Algorithm D.6.

D.4.1 Numerical Validation

In this section the parallel iterative solver based on the preconditioned biconjugate gradient stabilized method and domain decomposition method outlined above will be validated using the stochastic advection-diffusion equation given by [178]

$$-\nabla \cdot (c(\mathbf{x}, \theta) \nabla u(\mathbf{x}, \theta)) + \mathbf{b}(\mathbf{x}) \cdot \nabla u(\mathbf{x}, \theta) = f(\mathbf{x}) \quad \text{in } D \times \Omega. \quad (\text{D.13})$$

The spatial domain D used for this validation exercise is the unit square $(0, 1)^2$. The

```

Input: Initial iterate  $\mathcal{X}_{\Gamma_0}$ 
Output: Interface solution  $\mathcal{X}_{\Gamma}$ 
1  $r_0 = \mathcal{G}_{\Gamma} - \mathcal{S} \mathcal{X}_{\Gamma_0}$ 
2 Choose  $\tilde{r}$  such that  $(\tilde{r}, r_0) \neq 0$ , for example  $\tilde{r} = r_0$ 
3 for  $i = 0, 1, \dots$ , do:
4    $\rho_i = (\tilde{r}, r_i)$ 
5   if  $\rho_i = 0$  method fails
6   if  $i = 0$  then
7      $p_{i+1} = r_i$ 
8   else
9      $\beta_i = (\rho_i / \rho_{i-1}) (\alpha_i / \omega_i)$ 
10     $p_{i+1} = r_i + \beta_i (p_i - \omega_i v_i)$ 
11  end if
12  for each processor  $P_s, s = 1, 2, \dots, n_s$ , do:
13    Scatter:  $p^s = \mathcal{R}_s p_{i+1}$ 
14    Compute locally:  $\hat{p}^s = [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} p^s$ 
15    Gather:  $\hat{p} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \hat{p}^s$ 
16  end do
17  for each processor  $P_s, s = 1, 2, \dots, n_s$ , do:
18    Scatter:  $\hat{p}^s = \mathcal{R}_s \hat{p}$ 
19    Compute locally:  $u_1 = \mathcal{A}_{\Gamma\Gamma}^s \hat{p}^s$ 
20    Solve locally:  $\mathcal{A}_{\Gamma\Gamma}^s u_2 = u_1$ 
21    Compute locally:  $u_3 = \mathcal{A}_{\Gamma\Gamma}^s u_2$ 
22    Compute locally:  $u_4 = \mathcal{A}_{\Gamma\Gamma}^s p^s$ 
23    Compute locally:  $v^s = u_4 - u_3$ 
24    Gather:  $v_{i+1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T v^s$ 
25  end do
26   $\alpha_{i+1} = \rho_i / (\tilde{r}, v_{i+1})$ 
27   $s = r_i - \alpha_{i+1} v_{i+1}$ 
28  if norm of  $s$  is small, set  $\mathcal{X}_{\Gamma_{i+1}} = \mathcal{X}_{\Gamma_i} + \alpha_{i+1} \hat{p}$  and exit
29  for each processor  $P_s, s = 1, 2, \dots, n_s$ , do:
30    Scatter:  $s^s = \mathcal{R}_s s$ 
31    Compute locally:  $\hat{s}^s = [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} s^s$ 
32    Gather:  $\hat{s} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \hat{s}^s$ 
33  end do
34  for each processor  $P_s, s = 1, 2, \dots, n_s$ , do:
35    Scatter:  $\hat{s}^s = \mathcal{R}_s \hat{s}$ 
36    Compute locally:  $u_1 = \mathcal{A}_{\Gamma\Gamma}^s \hat{s}^s$ 
37    Solve locally:  $\mathcal{A}_{\Gamma\Gamma}^s u_2 = u_1$ 
38    Compute locally:  $u_3 = \mathcal{A}_{\Gamma\Gamma}^s u_2$ 
39    Compute locally:  $u_4 = \mathcal{A}_{\Gamma\Gamma}^s p^s$ 
40    Compute locally:  $t^s = u_4 - u_3$ 
41    Gather:  $t = \sum_{s=1}^{n_s} \mathcal{R}_s^T t^s$ 
42  end do
43   $\omega_{i+1} = (t, s) / (t, t)$ 
44   $\mathcal{X}_{\Gamma_{i+1}} = \mathcal{X}_{\Gamma_i} + \alpha_{i+1} \hat{p} + \omega_{i+1} \hat{s}$ 
45   $r_{i+1} = s - \omega_{i+1} t$ 
46  if iterate has converged, exit
47 end do

```

Algorithm D.6: Parallel Preconditioned Biconjugate Gradient Stabilized Method.

diffusion coefficient c is modeled as a lognormal random variable, i.e. $c \sim \text{LogN}(\mu, \sigma^2)$, with a third-order truncated polynomial chaos expansion in Eq. (D.8). In this special case the governing equation Eq. (D.13) can be written as

$$-c(\theta)\nabla^2 u(\mathbf{x}, \theta) + \mathbf{b} \cdot \nabla u(\mathbf{x}, \theta) = f(\mathbf{x}) . \quad (\text{D.14})$$

The case of zero right-hand side (i.e. $f = 0$) and Dirichlet boundary value conditions of $u = 0$ for the left boundary and $u = 1$ for the right boundary is considered. Under these conditions and an advection velocity of $\mathbf{b}(\mathbf{x}) = (1, 0)$, the following analytical solution satisfies both the governing equation and the imposed Dirichlet boundary conditions [178]

$$u(x, y, \theta) = \frac{e^{\frac{x}{c(\theta)}} - 1}{e^{\frac{1}{c(\theta)}} - 1} . \quad (\text{D.15})$$

Similar to the validation experiments outlined in Section D.3.1, 1000 samples of c are generated using its truncated PCE representation and Latin Hypercube Sampling (LHS) technique [121, 122] using an underlying Gaussian random variable having mean $\mu = 1$ and standard deviation $\sigma = 0.3$. These samples provide 1000 realizations of the solution process using Eq. (D.15) resulting in an ensemble mean and standard deviation shown in Figs. D.5.

The mean and standard deviation estimates in Fig. D.5 will be used as the “true” mean and standard deviation for validation of the parallel solver. The FEM discretization and domain decomposition of the spatial domain are illustrated in Fig. D.2 resulting in 19,694 linear triangular elements and 10,020 nodes for the FEM mesh decomposed into 20 subdomains. Using the parallel solver based on domain decomposition and preconditioned biconjugate gradient stabilized method, the PCE coefficients of a third order PCE expansion for the solution process are obtained. A zero initial value for the

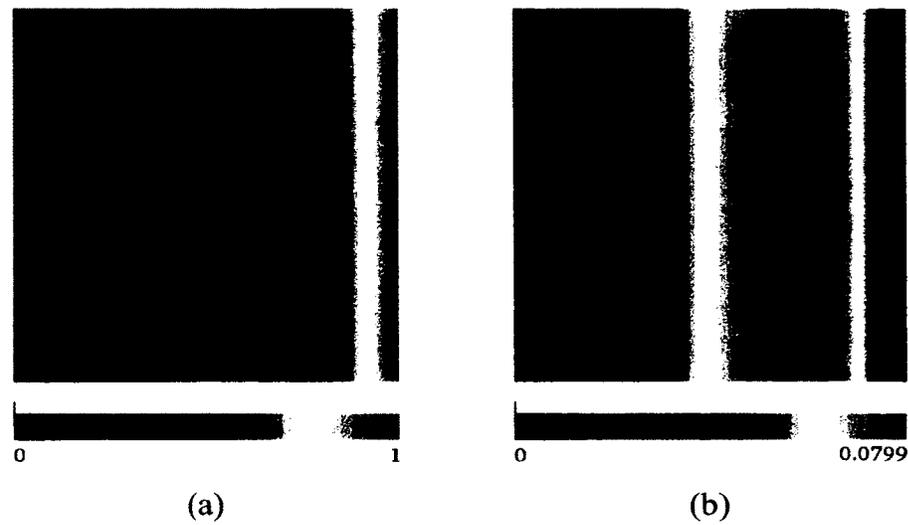


Figure D.5: Validation of parallel BICGSTAB solver: (a) MC ensemble mean of analytical response (b) MC ensemble standard deviation of analytical response.

Krylov-subspace iterate is used along with the stopping criteria given by Eq. (D.12) with $\epsilon = 10^{-15}$. The mean response, being the zeroth order coefficient, is shown in Fig. D.6a. The error between that estimate and the true mean response is given in Fig. D.6b, being roughly four orders of magnitude smaller than the true mean response. The covariance matrix is computed using Eq. (6.26) and the standard deviation of the response is subsequently obtained, along with the associated error, and plotted in Fig. D.7. The error in the standard deviation is relatively greater than the error in the mean, an observation that is expected of estimates obtained using finite MC sampling and truncated PCE representations.

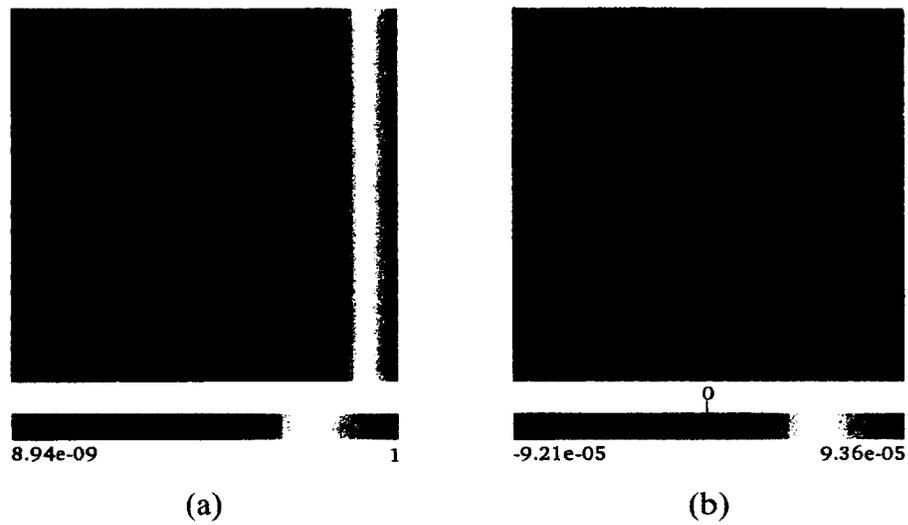


Figure D.6: Validation of parallel BICGSTAB solver: (a) Mean of the solution process using the parallel solver based on domain decomposition and preconditioned biconjugate gradient stabilized method and (b) associated error in comparison to MC ensemble mean of analytical response.

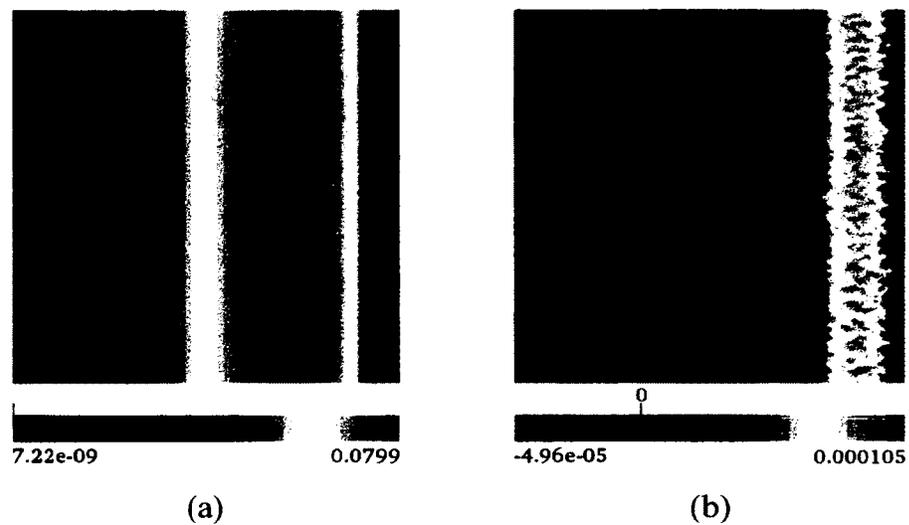


Figure D.7: Validation of parallel BICGSTAB solver: (a) Standard deviation of the solution process using the parallel solver based on domain decomposition and preconditioned biconjugate gradient stabilized method and (b) associated error in comparison to MC ensemble standard deviation of analytical response.

Appendix E

Equation of Motion for a 2D Airfoil with Quasi-Steady Aerodynamics

Based on thin airfoil theory and using a quasi-steady aerodynamic model, the equation of motion for a 2D airfoil undergoing heave, denoted by h , and pitch, denoted by θ , motion (see Fig E.1) is given by [198]

$$m\ddot{h} + \frac{mcx_\theta}{2}\ddot{\theta} + d_h\dot{h} + k_h h = -L \quad (\text{E.1})$$

$$I_{EA}\ddot{\theta} + \frac{mcx_\theta}{2}\ddot{h} + d_\theta\dot{\theta} + k_\theta\theta = M_{EA} \quad (\text{E.2})$$

with the aerodynamic loads being

$$L = \frac{\rho U^2}{2} cs 2\pi \left(\theta + \frac{\dot{h}}{U} + \frac{\dot{\theta}c(1/2 - a_h)}{2U} \right) \quad (\text{E.3})$$

$$M_{EA} = \frac{\rho U^2}{2} cs 2\pi \left(\theta \frac{c(1/2 + a_h)}{2} + \dot{h} \frac{c(1/2 + a_h)}{2U} + \dot{\theta} \left(\frac{c^2(1/2 - a_h)(1/2 + a_h)}{4U} - \frac{c^2}{16U} \right) \right) \quad (\text{E.4})$$

where m represents the structural mass, I_{EA} the mass moment of inertia, k_h the spring stiffnesses in heave, k_θ the spring stiffnesses in pitch, c the chord length of the airfoil, x_θ the mass offset aft of the elastic axis normalized by the half-chord length, L the

aerodynamic lift, M_{EA} the aerodynamic moment about the elastic axis, U the free flow airspeed, ρ the air density, s the span of the airfoil, and a_h the elastic axis offset aft of mid-chord normalized by the half-chord length. In Fig.E.1, AC denotes the aerodynamic center being the point on the airfoil at which the aerodynamic moment remains constant with respect to changing angle of attack (in low angle range and high Reynolds Number) and EA denotes the elastic axis.

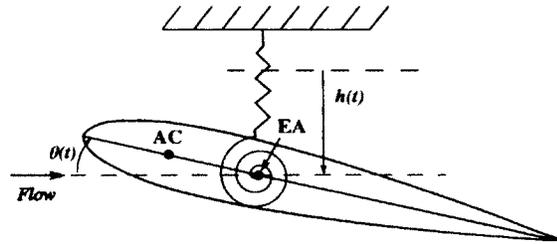


Figure E.1: Schematic of 2D airfoil with heave and pitch degrees of freedom.

One can combine the structural and aerodynamic terms together and rewrite the above system of equations in matrix form as

$$[M] \begin{Bmatrix} \ddot{h} \\ \ddot{\theta} \end{Bmatrix} + [C] \begin{Bmatrix} \dot{h} \\ \dot{\theta} \end{Bmatrix} + [K] \begin{Bmatrix} h \\ \theta \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}, \quad (\text{E.5})$$

where

$$[M] = \begin{bmatrix} m & \frac{mcx_{\theta}}{2} \\ \frac{mcx_{\theta}}{2} & I_{EA} \end{bmatrix}, \quad (\text{E.6})$$

$$[C] = \begin{bmatrix} d_h + \rho U c s \pi & \rho U c^2 s \pi \frac{1/2 - a_h}{2} \\ -\rho U c^2 s \pi \frac{1/2 + a_h}{2} & d_{\theta} + \rho U c^3 s \pi \left(\frac{1}{16} - \frac{(1/2 - a_h)(1/2 + a_h)}{4} \right) \end{bmatrix}, \quad (\text{E.7})$$

$$[K] = \begin{bmatrix} k_h & \rho U^2 c s \pi \\ 0 & k_{\theta} - \rho U^2 c^2 s \pi \frac{1/2 + a_h}{2} \end{bmatrix}. \quad (\text{E.8})$$

Note that the two degrees of freedom h and θ are coupled by the off-diagonal el-

elements of the system matrices. There is coupling in the mass, damping and stiffness matrices. It is interesting to note that for both the mass and damping matrices, there is a two-way coupling. There is, however, a one-way coupling in stiffness. Physically, it means that a non-zero pitch angle would create a lift force for the heave degree of freedom for a static problem. However, a non-zero heave would not create a moment for the pitch degree of freedom.

Bibliography

- [1] R. G. Ghanem and P. Spanos. *Stochastic Finite Elements : A Spectral Approach*. Courier Dover Publications, 1991.
- [2] P. C. Young. Data-based mechanistic modelling and validation of rainfall-flow processes. In M. G. Anderson and P. D. Bates, editors, *Model Validation: Perspectives in Hydrological Science*. John Wiley & Sons, 2001.
- [3] K. P. Burnham and D. R. Anderson. *Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, 2002.
- [4] G. Reikard and W. E. Rogers. Forecasting ocean waves: Comparing a physics-based model with statistical models. *Coastal Engineering*, 58(5):409–416, 2011.
- [5] R. G. Ghanem. Uncertainty quantification in computational and prediction science. *International Journal for Numerical Methods in Engineering*, 80:671–672, 2009.
- [6] *Dynamic Data-Driven Application Systems*. <http://www.dddas.org/>.
- [7] The opportunities and challenges of Exascale computing. In *Summary Report of the Advanced Scientific Computing Advisory Committee*, U.S. Department of Energy Office of Science, 2010.
- [8] A. F. Bennett. *Inverse Modeling of the Ocean and the Atmosphere*. Cambridge University Press, 2002.
- [9] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [10] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, second edition, 2009.
- [11] F. X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations. *Tellus Series A: Dynamic Meteorology and Oceanography*, 38A:97–110, 1986.
- [12] G. I. Marchuk. *Adjoint Equations and Analysis of Complex Systems*. Kluwer Academic Publishers, 1995.

- [13] G. I. Marchuk, I. V. Agoshkov, and P. V. Shutyaev. *Adjoint Equations and Perturbation Algorithms in Nonlinear Problems*. CRC Press, 1996.
- [14] R. Geiring, T. Kaminski, and T. Slawig. Generating efficient derivative code with TAF adjoint and tangent linear Euler flow around an airfoil. *Future Generation Computer Systems*, 21:1345–1355, 2005.
- [15] R. G. Hanea and G. J. M. Velders. A hybrid Kalman filter algorithm for large-scale atmospheric chemistry data assimilation. *Monthly Weather Review*, 135(1):140–151, 2007.
- [16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.
- [17] R. E. Kalman and R. C. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83:95–108, 1961.
- [18] R. N. Miller, M. Ghill, and F. Gauthiez. Advanced data assimilation in strongly nonlinear dynamical systems. *Journal Of The Atmospheric Sciences*, 51(8):1037–1056, 1994.
- [19] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2005.
- [20] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [21] G. Evensen. Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. *Journal Of Geophysical Research - Oceans*, 97(C11):17905–17924, 1992.
- [22] P. Gauthier, P. Courtier, and P. Moll. Assimilation of simulated wind lidar data with a Kalman filter. *Monthly Weather Review*, 121(6):1803–1820, 1993.
- [23] J. K. Uhlmann. *Dynamic map building and localization: New theoretical foundations*. PhD thesis, University of Oxford, Oxford, U.K., 1995.
- [24] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*, Orlando, Florida, USA, 1997.
- [25] A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss. Estimation of parameters and unobserved components for nonlinear systems from noisy time series. *Physical Review E*, 66:016210, 2002.
- [26] S. Mariani and A. Ghisi. Unscented Kalman filtering for nonlinear structural dynamics. *Nonlinear Dynamics*, 49:131–150, 2007.

- [27] E. A. Wan and R. van der Merwe. Parallelization of a large scale Kalman filter: comparison between mode and domain decomposition. In S. Haykin, editor, *Kalman Filtering and Neural Networks*. John Wiley & Sons, 2001.
- [28] G. Burgers, P. J. van Leeuwen, and G. A. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.
- [29] P. J. van Leeuwen. Comment on data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 127(6):1374–1377, 1999.
- [30] J. D. Annan. Parameter estimation using chaotic time series. *Tellus Series A: Dynamic Meteorology and Oceanography*, 57(5):709–714, 2005.
- [31] R. G. Ghanem and G. Ferro. Health monitoring for strongly non-linear systems using the ensemble Kalman filter. *Structural Control and Health Monitoring*, 13(1):245–259, 2006.
- [32] G. A. Kivman. Sequential parameter estimation for stochastic systems. *Nonlinear Processes In Geophysics*, 10(3):253–259, 2003.
- [33] R. N. Miller, E. F. Carter, and S. T. Blue. Data assimilation into nonlinear stochastic models. *Tellus Series A: Dynamic Meteorology and Oceanography*, 51(2):167–194, 1999.
- [34] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143–10162, 1994.
- [35] J. L. Anderson and S. L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999.
- [36] T. Bengtsson, C. Snyder, and D. Nychka. Toward a nonlinear ensemble filter for high-dimensional systems. *Journal Of Geophysical Research-Atmospheres*, 108(D24):8775, 2003.
- [37] D. T. Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, 129(5):1194–1207, 2001.
- [38] P. J. van Leeuwen. A variance-minimizing filter for large-scale applications. *Monthly Weather Review*, 131(9):2071–2084, 1998.
- [39] S. Kim, G. L. Eyink, J. M. Restrepo, F. J. Alexander, and G. Johnson. Ensemble filtering for nonlinear dynamics. *Monthly Weather Review*, 131(11):2586–2594, 2003.
- [40] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

- [41] C. S. Manohar and D. Roy. Monte Carlo filters for identification of nonlinear structural dynamical systems. *Sadhana - Academy Proceedings In Engineering Sciences*, 31:399–427, Part 4, 2006.
- [42] V. Namdeo and C. S. Manohar. Nonlinear structural dynamical system identification using adaptive particle filters. *Journal of Sound and Vibration*, 306:524–563, 2007.
- [43] S. J. Ghosh, C. S. Manohar, and D. Roy. A sequential importance sampling filter with a new proposal distribution for state and parameter estimation of nonlinear dynamical systems. *Proceedings of the Royal Society of London: Series A*, 464:25–47, 2007.
- [44] H. Tanizaki. *Nonlinear Filters: Estimation and Applications*. Springer, second edition, 1996.
- [45] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear non-Gaussian Bayesian state estimation. *IEE Proceedings - F Radar And Signal Processing*, 140(2):107–113, 1993.
- [46] J. Ching, J. L. Beck, and K. A. Porter. Bayesian state and parameter estimation of uncertain dynamical systems. *Probabilistic Engineering Mechanics*, 21:81–96, 2006.
- [47] M. Khalil, A. Sarkar, and S. Adhikari. Nonlinear filters for chaotic oscillatory systems. *Journal of Nonlinear Dynamics*, 55(1-2):113–137, 2009.
- [48] J. Mandel and J. D. Beezley. Predictor-corrector and morphing ensemble filters for the assimilation of sparse data into high-dimensional nonlinear systems. In *Proceedings of the IOAS-AOLS, 11th Symposium on Integrated Observing and Assimilation Systems for the Atmosphere, Oceans, and Land Surface*, page 4.12, San Antonio, Texas, USA, 2007.
- [49] N. Papadakis, E. Memin, A. Cuzol, and N. Gengembre. Data assimilation with the weighted ensemble Kalman filter. *Tellus Series A: Dynamic Meteorology and Oceanography*, 62(5):673–697, 2010.
- [50] C. Andrieu and A. Doucet. Online expectation-maximization type algorithms for parameter estimation in general state space models. In *Proceedings of ICASSP '03, the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, 2003.
- [51] A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.
- [52] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

- [53] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
- [54] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference Conference on Uncertainty in Artificial Intelligence*, Stanford, California, USA, 2000.
- [55] J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [56] G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, 1998.
- [57] W. R. Gilks and C. Berzuini. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(1):127–146, 2001.
- [58] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.
- [59] D. Gamerman. *Markov Chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman & Hall, 1997.
- [60] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, 2001.
- [61] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, , and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [62] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
- [63] W. Zwiefelhofer and G. Mozdzynski, editors. *Use of High Performance Computing in Meteorology*. World Scientific, 2005.
- [64] T. Bengtsson, P. Bickel, and B. Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. 2:316–334, 2008.
- [65] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.
- [66] P. M. Lyster, S. E. Cohn, R. Mnard, L. P. Chang, S. J. Lin, and R. G. Olsen. Parallel implementation of a Kalman filter for constituent data assimilation. *Monthly Weather Review*, 125(7):1674, 1997.

- [67] S. Hong, S. Chin, P. M. Djurić, and M. Bolić. Design and implementation of flexible resampling mechanism for high-speed parallel particle filters. *Journal of VLSI Signal Processing*, 44:47–62, 2006.
- [68] C. L. Keppenne. Data assimilation into a primitive-equation model with a parallel ensemble Kalman filter. *Monthly Weather Review*, 128(6):1971–1981, 2000.
- [69] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.
- [70] C. L. Keppenne and M. M. Rienecker. Initial testing of a massively parallel ensemble Kalman filter with the Poseidon isopycnal ocean general circulation model. *Monthly Weather Review*, 130(12):2951–2965, 2002.
- [71] A. J. Segers and A. W. Heemink. Parallelization of a large scale Kalman filter: comparison between mode and domain decomposition. In P. Wilders, A. Ecer, J. Periaux, N. Satofuka, and P. Fox, editors, *Parallel Computational Fluid Dynamics: Practice and Theory, Proceedings of the Parallel CFD 2001 Conference*. Elsevier, 2002.
- [72] C. L. Keppenne and M. M. Rienecker. Assimilation of temperature into an isopycnal ocean general circulation model using a parallel ensemble Kalman filter. *Journal of Marine Systems*, 40-41(1-4):363–380, 2003.
- [73] L. Nerger, W. Hiller, and J. Schroeter. PDAF - the parallel data assimilation framework: experiences with Kalman filtering. *Use of High Performance Computing in Meteorology, Proceedings of the 11th ECMWF Workshop*, pages 63–83, 2005.
- [74] T. Miyoshi. *Ensemble Kalman filter experiments with a primitive-equation global model*. PhD thesis, University of Maryland, Maryland, USA, 2005.
- [75] J. Mandel. Efficient implementation of the ensemble Kalman filter. In *CCM Report 231*, University of Colorado Denver, Denver, Colorado, USA, 2006.
- [76] R. Tavakoli. Multi-level parallelization of ensemble Kalman filter (enkf) for automatic reservoir history matching. *Institute for Computational Engineering and Sciences, Center for Subsurface Modeling (CSM) Annual Industrial Affiliates meeting, Austin, Texas, USA*, 2010.
- [77] *Message Passing Interface*. <http://www.mpi-forum.org/>.
- [78] I. Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. AddisonWesley, 1995.
- [79] P.S. Pacheco. *Parallel programming with MPI*. Morgan Kaufmann, 1997.
- [80] G. Saad and R. G. Ghanem. Characterization of reservoir simulation models using a polynomial chaos-based ensemble Kalman filter. *Water Resources Research*, 45:article number W04417, 2009.

- [81] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.
- [82] A. Sarkar, N. Benabbou, and R. G. Ghanem. Domain decomposition of stochastic PDEs: Theoretical formulations. *International Journal for Numerical Methods in Engineering*, 77:689–701, 2009.
- [83] W. Subber, H. Monajemi, M. Khalil, and A. Sarkar. A scalable parallel uncertainty analysis and data assimilation framework applied to some hydraulic problems. In *Proceedings of the International Symposium on Uncertainties in Hydrologic and Hydraulic Modeling*, Montreal, Quebec, Canada, 2008.
- [84] W. Subber and A. Sarkar. Domain decomposition of stochastic PDEs: A novel preconditioner and its parallel performance. In D. J. K. Mewhort, N. M. Cann, G. W. Slater, and T. J. Naughton, editors, *High Performance Computing Symposium*, volume 5976 of *Lecture Notes in Computer Science*, pages 251–268. Springer, 2010.
- [85] W. Subber and A. Sarkar. Primal and dual-primal iterative substructuring methods of stochastic PDEs. *Journal of Physics: Conference Series*, 256(1), 2010.
- [86] W. Subber and A. Sarkar. Domain decomposition methods of stochastic PDEs. In *Proceedings of the Twentieth International Conference on Domain Decomposition Methods*, San Diego, California, USA, 2011.
- [87] W. Subber and A. Sarkar. A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner. *Journal of Computational Physics*, under review.
- [88] M. Khalil, D. Poirel, A. Sarkar, and S. Dunn. Application of Bayesian inference to the flutter margin method. In *Proceedings of IFASD 2009, the International Forum on Aeroelasticity and Structural Dynamics*, Seattle, Washington, USA, 2009.
- [89] M. Khalil, D. Poirel, and A. Sarkar. Application of Bayesian inference to the flutter margin method - new developments. In *Proceedings of the ASME 3rd US-European Fluids Engineering Summer Meeting*, paper FEDSM-ICNMM2010-30041, Montreal, Quebec, Canada, 2010.
- [90] M. Khalil, D. Poirel, and A. Sarkar. Application of Bayesian inference to the flutter margin method - new developments. In *Proceedings of IFASD 2011, the International Forum on Aeroelasticity and Structural Dynamics*, Paris, France, 2011.
- [91] M. Khalil, A. Sarkar, S. Adhikari, and D. Poirel. The estimation of time-invariant parameters of noisy nonlinear oscillatory systems. *Journal of Sound and Vibration*, under review.

- [92] M. Khalil, D. Poirel, and A. Sarkar. Parameter estimation of a fluttering aeroelastic system in the transitional Reynolds number regime. In *Proceedings of the ASME 3rd US-European Fluids Engineering Summer Meeting*, paper FEDSM-ICNMM2010-30047, Montreal, Quebec, Canada, 2010.
- [93] M. Khalil, D. Poirel, and A. Sarkar. Parameter estimation of a fluttering aeroelastic system in the transitional Reynolds number regime. In *Proceedings of IFASD 2011, the International Forum on Aeroelasticity and Structural Dynamics*, Paris, France, 2011.
- [94] M. Khalil, D. Poirel, and A. Sarkar. Parameter estimation of a fluttering aeroelastic system in the transitional Reynolds number regime. *Journal of Sound and Vibration*, DOI 10.1016/j.jsv.2013.02.012 (In press), 2013.
- [95] W. Subber and A. Sarkar. Domain decomposition methods of stochastic PDEs: A two-level scalable preconditioner. *Journal of Physics: Conference Series*, 341(1), 2012.
- [96] M. Khalil, W. Subber, and A. Sarkar. Data assimilation for large-scale computational models. In *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA paper AIAA-2013-1468, Boston, Massachusetts, USA, 2013.
- [97] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.
- [98] D. R. Cox and D. V. Hinkley. *Theoretical Statistics*. Chapman and Hall, 1974.
- [99] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. John Wiley & Sons, second edition, 1998.
- [100] S. J. Press. *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications*. John Wiley & Sons, second edition, 2003.
- [101] S. M. Tan, C. Fox, and G. Nicholls. *Physics 707 (Inverse Problems Course) Lecture Notes*. <http://home.comcast.net/~szemengtan/>, 2009.
- [102] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
- [103] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski. Overview of sequential Monte Carlo methods for parameter estimation on general state space models. In *Proceedings of the 15th IFAC Symposium on System Identification (SYSID)*, Saint-Malo, France, 2009.
- [104] M. Bocquet, C. A. Pires, and L. Wu. Beyond Gaussian statistical modeling in geophysical data assimilation. *Monthly Weather Review*, 138:2997–3023, 2010.

- [105] J. Guckenheimer and P. Holmes. *Kalman Filtering : Theory and Applications*. IEEE Press, 1985.
- [106] D. Simon. *Optimal State Estimation: Kalman, H-[infinity], and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [107] R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health & Science University, Portland, Oregon, USA, 2004.
- [108] R. van der Merwe and E. Wan. The square-root unscented Kalman filter for state and parameter estimation. In *Proceedings of ICASSP '01, the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, USA, 2001.
- [109] P. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. PhD thesis, Linköping University, Linköping, Sweden, 2002.
- [110] A. Doucet. On sequential simulation-based methods for Bayesian filtering. In *Technical Report CUED/FINFENG/TR 310*, Department of Engineering, Cambridge University, 1998.
- [111] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599s, 1999.
- [112] R. Douc, O. Cappe, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of ISPA 2005, the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69, Zagreb, Croatia, 2005.
- [113] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.
- [114] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman & Hall, 1986.
- [115] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Field*. Springer-Verlag, 1983.
- [116] S. Lynch. *Dynamical Systems with Applications using MATLAB*. Birkhäuser, 2004.
- [117] A. S. Stordal, H. A. Karlsen, G. Nævdal, H. J. Skaug, and B. Vallès. Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter. *Computational Geosciences*, 15(2):293–305, 2010.
- [118] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1999.

- [119] G. Maruyama. Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4:48–90, 1955.
- [120] D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review, Education Section*, 43:525–546, 2001.
- [121] M. McKay, R. Beckman, and W. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.
- [122] A. Olsson and G. Sandberg. Latin hypercube sampling for stochastic finite element analysis. *Journal of Engineering Mechanics*, 128:121–125, 2002.
- [123] A. F. M. Smith. Bayesian computational methods. *Philosophical Transactions, Physical Sciences and Engineering*, 337(1647):369–386, 1991.
- [124] M. Evans and T. Swartz. Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statistical Science*, 10:254–272, 1995.
- [125] N. Metropolis. The beginning of the Monte Carlo method. In *Los Alamos Science Special Issue*, volume 15, pages 125–130. Cambridge University Press, 1989.
- [126] R. Turner. *Perspectives on Projects*. Taylor & Francis, 2009.
- [127] S. B. McGrayne. *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy*. Yale University Press, 2011.
- [128] S. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
- [129] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [130] D. Draper. *Engineering 206 (Bayesian Statistics Course) Lecture Notes*. <http://www.soe.ucsc.edu/classes/ams206/Winter03/>, 2003.
- [131] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7:110–120, 1997.
- [132] N. H. Zimmerman and J. T. Weissenburger. Prediction of flutter onset speed based on flight testing at subcritical speeds. *Journal of Aircraft*, 1(4):190–202, 1964.
- [133] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. *Aeroelasticity*. Courier Dover Publications, 1996.
- [134] S. J. Price and B. H. K. Lee. Evaluation and extension of the flutter-margin method for flight flutter prediction. *Journal of Aircraft*, 30(3):395–401, 1993.

- [135] C. L. Pettit. Uncertainty quantification in aeroelasticity: Recent results and research challenges. *Journal of Aircraft*, 41(5):1217–1229, 2004.
- [136] R. Lind and M. Brenner. Flutterometer: An on-line tool to predict robust flutter margins. *Journal of Aircraft*, 37(6):1105–1112, 2000.
- [137] D. Poirel, S. Dunn, and J. Porter. Flutter-margin method accounting for modal parameter uncertainties. *Journal of Aircraft*, 42(5):1236–1243, 2005.
- [138] J. Heeg. Stochastic characterization of flutter using historical wind tunnel data. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA paper 2007-1769, Honolulu, Hawaii, USA, 2007.
- [139] C. Hoen. An engineering interpretation of the complex eigensolution of linear dynamic systems. In *Proceedings of IMAC XXIII, the 23rd Conference and Exposition on Structural Dynamics*, Orlando, Florida, USA, 2005.
- [140] P. P. da Silva, V. Kreinovich, and C. Servin. Maximum entropy in support of semantically annotated datasets. In F. Bobillo, P. C. G. da Costa, C. d’Amato, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, T. P. Martin, M. Nickles, M. Pool, and P. Smrz, editors, *URSW*, volume 423 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [141] J. N. Juang, J. E. Cooper, and J. R. Wright. An eigensystem realization algorithm using data correlations (ERA/DC) for modal parameter identification. *Control-Theory and Advanced Technology*, 4(1):5–14, 1988.
- [142] C. Andrieu, N. de Freitas, and A. Doucet. Sequential MCMC for Bayesian model selection. In *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*, Caesarea, Israel, 1999.
- [143] G. Storvik. Particle filters in state space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289, 2002.
- [144] J. A. Vrugt, C. J. F. ter Braak, C. G. H. Diks, and G. Schoups. Hydrologic data assimilation using particle Markov Chain Monte Carlo simulation: Theory, concepts and applications. *Advances in Water Resources*, 51:457–478, 2013.
- [145] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [146] J. Mandel and J. D. Beezley. An ensemble Kalman-particle predictor-corrector filter for non-gaussian data assimilation. In *ICCS 2009, Lecture Notes in Computer Science 5545*, Springer, pages 470–478, 2009.

- [147] S. Haykin, editor. *Kalman Filtering and Neural Networks*. John Wiley & Sons, 2001.
- [148] C. K. Chui and G. Chen. *Kalman Filtering with Real-time Applications*. Springer, 1999.
- [149] R. Sandhu, M. Khalil, D. Poirel, and A. Sarkar. Model selection methods for nonlinear aeroelastic systems using wind-tunnel data. In *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA paper AIAA-2013-1938, Boston, Massachusetts, USA, 2013.
- [150] P. Bisailon, R. Sandhu, M. Khalil, D. Poirel, and A. Sarkar. Model selection for strongly nonlinear systems. In *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA paper AIAA-2013-1939, Boston, Massachusetts, USA, 2013.
- [151] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *The Annals of Mathematics*, 48(2):385–392, 1947.
- [152] H. H. Kuo. Gaussian measures in Banach spaces. In A. Dold and B. Eckmann, editors, *Lecture Notes in Mathematics*, volume 463, pages 1–109. Springer-Verlag, 1975.
- [153] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 1992.
- [154] W. Häardie and M. Müller. Multivariate and semiparametric kernel regression. In M. G. Schimek, editor, *Smoothing and Regression: Approaches, Computation, and Application*, pages 357–391. John Wiley & Sons, 2000.
- [155] E. M. Constantinescu, A. Sandu, T. Chaib, and G. R. Carmichael. Ensemble-based chemical data assimilation. I: General approach. *Quarterly Journal Of The Royal Meteorological Society*, 133(626):1229–1243, 2007.
- [156] M. Gad el Hak. Control of low-speed airfoil aerodynamics. *AIAA Journal*, 28(9):1537–1552, 1990.
- [157] D. Poirel, Y. Harris, and A. Benaissa. Self-sustained aeroelastic oscillations of a NACA0012 airfoil at low-to-moderate Reynolds numbers. *Journal of Fluids and Structures*, 24(5):700–719, 2008.
- [158] D. Poirel and W. Yuan. Aerodynamics of laminar separation flutter at a transitional Reynolds number. *Journal of Fluids and Structures*, 26(7-8):1174–1194, 2010.
- [159] W. J. McCroskey. A critical assessment of wind tunnel results for the naca0012 airfoil. Technical Memorandum NASA/TM-100019, NASA, October 1987.

- [160] R. F. Huang, W. W. Shy, S. W. Lin, and F. B. Hsiao. Influence of surface flow on aerodynamic loads of a cantilever wing. *AIAA Journal*, 34(3):527–532, 1996.
- [161] M. Khalil, D. Poirel, and A. Sarkar. Parameter estimation of stochastic aeroelastic systems from wind tunnel data: A sequential data assimilation approach. Presented at *the 10th US National Congress of Computational Mechanics*. Columbus, Ohio, USA, 2009.
- [162] I. Mbalawata, S. Särkkä, and H. Haario. Parameter estimation in stochastic differential equations with Markov Chain Monte Carlo and non-linear Kalman filtering. *Computational Statistics*, pages 1–29, 2012.
- [163] C. Popescu and Y.S. Wong. A nonlinear statistical approach for aeroelastic response prediction. *AIAA Journal of Guidance, Control and Dynamics*, 26(4):565–572, 2003.
- [164] P. S. Maybeck, editor. *Stochastic Models, Estimation, and Control*. Academic Press, San Diego, 1979.
- [165] S. Konishi and G. Kitagawa. *Information Criteria and Statistical Modeling (Springer Series in Statistics)*. Springer, 2007.
- [166] O. C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, third edition, 1977.
- [167] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.
- [168] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division, ASCE*, 85:67–94, 1959.
- [169] R. G. Ghanem and S. Dham. Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transport in Porous Media*, 32(3):239–262, 1998.
- [170] O. Le Maitre, M. Reagan, H. Najm, R. G. Ghanem, , and O. Knio. A stochastic projection method for fluid flow. ii: Random process. *Journal of Computational Physics*, 181:9–44, 2002.
- [171] A. Graham. *Kronecker Products and Matrix Calculus: With Applications*. Ellis Horwood, 1981.
- [172] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, third edition, 1996.
- [173] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

- [174] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, second edition, 2003.
- [175] G. Saad. *Stochastic Data Assimilation with Application to Multi-Phase Flow and Health Monitoring Problems*. PhD thesis, University of Southern California, Los Angeles, California, USA, 2007.
- [176] D. G. Furnival. *Iterative Methods for the Stochastic Diffusion Problem*. PhD thesis, University of Maryland, College Park, Maryland, USA, 2008.
- [177] P. Billingsley. *Probability and Measure*. John Wiley & Sons, 1979.
- [178] A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences*. Springer, 2004.
- [179] W. Mendenhall. *Introduction to probability and statistics*. Duxbury Press, 1979.
- [180] R. Sandhu. Bayesian model selection and parameter estimation for a nonlinear fluid-structure interaction problem. Master's thesis, Carleton University, Ottawa, Ontario, Canada, 2013.
- [181] W. Subber. *Domain Decomposition Methods for Uncertainty Quantification*. PhD thesis, Carleton University, Ottawa, Ontario, Canada, 2012.
- [182] W. S. Levine. *Control System Fundamentals*. CRC Press, 2000.
- [183] M. J. F. Gales and S. S. Airey. Product of Gaussians for Speech Recognition. Technical report, Computer Speech & Language, 2003.
- [184] A. Sarkar and R. G. Ghanem. A substructure approach for the mid-frequency vibration of stochastic systems. *Journal of the Acoustical Society of America*, 113(4):1922–1934, 2003.
- [185] M. Loeve. *Probability Theory*. Springer, fourth edition, 1977.
- [186] R. G. Ghanem. Stochastic finite elements with multiple random non-Gaussian properties. *Journal of Engineering Mechanics*, 125:16–41, 1998.
- [187] R. G. Ghanem. The nonlinear Gaussian spectrum of log-normal stochastic processes and variables. *Journal of Applied Mechanics*, 66(4):964–973, 1999.
- [188] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Numerical Mathematics and Scientific Computation. Oxford University Press, 1999.
- [189] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.

- [190] T. Mathew. *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*. Number 61 in Lecture Notes in Computational Science and Engineering. Springer, 2008.
- [191] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.
- [192] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, second edition, 1994.
- [193] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM Journal on Numerical Analysis*, 21(2):352–362, 1984.
- [194] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [195] R. Fletcher. Conjugate gradient methods for indefinite systems. In G. A. Watson, editor, *Numerical Analysis*, volume 506 of *Lecture Notes in Mathematics*, pages 73–89. Springer Berlin Heidelberg, 1976.
- [196] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 10(1):36–52, 1989.
- [197] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.
- [198] Y. Fung. *An Introduction to the Theory of Aeroelasticity*. John Wiley & Sons, 1955.