LLNL-TR-760558

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Efficient Stochastic Inversion Using Adjoint Models and Kernel-PCA

X. Chen, C. Thimmisetty, C. Tong, J. White, C. Morency, C. Huang, M. Korkali, L. Min

October 26, 2018

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Efficient Stochastic Inversion Using Adjoint Models and Kernel-PCA

Xiao Chen[1], Charanraj A. Thimmisetty[1], Charles H. Tong[1],
Joshua A. White[2], Christina Morency[2], Can Huang[3],
Mert Korkali[3], Liang Min[3]

[1] Center for Applied Scientific Computing,
Lawrence Livermore National Laboratory, Livermore, California, USA
[2] Atmospheric, Earth and Energy Division,
Lawrence Livermore National Laboratory, Livermore, California, USA
[3] Engineering,
Lawrence Livermore National Laboratory, Livermore, California, USA

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

October 31, 2018

Abstract

Performing stochastic inversion on a computationally expensive forward simulation model with a high-dimensional uncertain parameter space (e.g. a spatial random field) is computationally prohibitive even with gradient information provided. Moreover, the 'nonlinear' mapping from parameters to observables generally gives rise to non-Gaussian posteriors even with Gaussian priors, thus hampering the use of efficient inversion algorithms designed for models with Gaussian assumptions. In this work, we propose a novel Bayesian stochastic inversion methodology, characterized by a tight coupling between a gradient-based Langevin Markov Chain Monte Carlo (LMCMC) method and a kernel principal component analysis (KPCA). This approach addresses the 'curse-of-dimensionality' via KPCA to identify a low-dimensional feature space within the high-dimensional and nonlinearly correlated spatial random field. Moreover, non-Gaussian full posterior probability distribution functions are estimated via an efficient LMCMC method on both the projected low-dimensional feature space and the recovered high-dimensional parameter space. We demonstrate this computational framework by integrating and adapting recent developments such as data-driven statistics-on-manifolds constructions and reduction-through-projection techniques to solve inverse problems in linear elasticity.

Chapter 1

Introduction

Computational science and engineering have enabled researchers to model complex physical processes in many disciplines—e.g. mechanical behavior [1], climate projection [2], subsurface flow and reactive transport [3], seismic wave propagation [4, 5], and power grid planning [6]. However, uncertainty in the model parameters makes the underlying problems essentially stochastic in nature. Applying uncertainty quantification (UQ) to improve model predictability usually requires solving an inverse problem (inverse UQ) by 'fusing' prior knowledge, simulations, and experimental observations. Deterministic approaches to solve inverse problems, such as regularized weighted nonlinear least squares methods, are capable of providing an optimal statistical estimator with associated error bars for the inverse solutions. However, these approaches, by their deterministic nature, cannot produce solutions with a full description of the posterior probability density functions (pdf). Unlike deterministic inversion, stochastic inversion aims to provide this fuller description. A pdf representation is critical for prediction of system performance, so that appropriate decisions can be made according to the probability and risk associated with specific events.

Bayesian inference provides a systematic framework for integrating prior knowledge and measurement uncertainties to compute detailed posteriors [7]. However, it can be computationally intractable [8] to compute the full pdf for parameters assigned to each grid point of a discretized parametric random field—i.e., the curse of dimensionality [8]. Moreover, unreasonable choices of prior knowledge due to ignorance of the information embedded in the underlying dataset for model parameters can have major effects on inferring the posterior pdf. In addition, the nonlinear mapping between the observables and parameters leads to non-Gaussian posteriors even with additive noise and Gaussian prior assumptions [8]. In general, it is mathematically challenging to sample directly from a non-Gaussian and multi-modal posteriors especially in a very high-dimensional random space. MCMC methods are relevant techniques for sampling non-standard posteriors. Despite the computational intensity encountered in MCMC, these methods have grown in rigor and sophistication with recent technical developments such as delayed rejection (DR) [9, 10], adaptive Metropolis (AM) [11, 12, 13], delayed rejection adaptive Metropolis (DRAM) [14], stochastic Newton [8], Langevin [15] and transport map accelerated MCMC [16].

The gradient-free MCMC methods, e.g., random walk MCMC, DR, AM, and DRAM, become computationally intractable as the size of the parameter space increases just moderately. Even though the gradient-enhanced MCMC algorithms such as Langevin [15] and stochastic Newton methods [8] have decreased the computational complexity of classical MCMC to $O(n^{1/3})$, expensive high-fidelity forward models, mesh-defined high-dimensional parameter spaces, and multi-modal non-Gaussianity cause significant computational challenges in practice, rendering these algorithms unsuitable for large-scale, real-world problems.

One way to address the computational complexity of MCMC is through a construction of low-fidelity surrogate models using design of experiments (DOE) with the help of machine learning techniques, e.g., global polynomials [17, 18, 19], radial basis functions [20, 21], Gaussian processes [22], neural networks [23, 24], and/or proper orthogonal decomposition (POD) based reduced modeling. The use of low-fidelity models, based on surrogate and/or reduced-order modeling, greatly reduces the computational cost of the stochastic inversion. Low-fidelity model-based stochastic inversion, however, tends to produce entirely different inverse solutions or sub-optimal solutions compared to the true posterior obtained by the corresponding high-fidelity model-based stochastic inversion.

Instead of performing forward model reduction, another way to reduce MCMC complexity is through control reduction, by performing Bayesian inference in a low-dimensional subspace embedded in the high-dimensional parameter space, while still using the high-fidelity forward model constrained onto this low-dimensional space. Karhunen-Loéve or principal component analysis (PCA) is a well-known choice for such parametric control dimension reduction. Traditionally, PCA is designed for the representation of linear correlation of the underlying data. Many realistic parametric random fields, however, exhibit non-linear correlations in the underlying data. The subspace spanned by PCA might not even cover the solution domain. Furthermore, one has to perform an exhaustive search to reach to the true posterior due to the widely scattered reduced space represented by the linear PCA-extracted subspace.

The method proposed here uses unsupervised learning techniques to obtain relevant subspaces. Recent advances in unsupervised machine learning algorithms have provided ways to explore non-linear datasets using manifold learning techniques. Specifically, kernel PCA [25] (KPCA) has been demonstrated to perform better clustering than linear PCA on complex non-linear data. Recently, Sarma [26] and Ma [27] demonstrated the efficiency and benefits of KPCA for deterministic forward and inverse uncertainty propagation.

Here, we propose a novel framework for efficient stochastic inversion using adjoint partial differential equations (PDEs), automatic differentiation (AD), and KPCA. We demonstrate our approach on a stochastic linear elasticity inversion problem. For this application, a full statistical analysis in the high-dimensional "ambient" space spanned by grid-defined model parameters is computationally prohibitive. In addition, the model output is a high-dimensional vector space defining the solution variables over the whole spatial discretization. Thus, we have the challenge of an ambient space where each measurement is a high-dimensional vector obtained as an expensive model evaluation. The solution, however, is constrained: it does not occupy the whole ambient space, but merely a low-dimensional manifold within it. Because only a low-dimensional probability space needs to be explored, we can design novel algorithms to accelerate the convergence of MCMC algorithms.

We use the following sequence to reduce the computational burden of solving large-scale stochastic inverse problems in elasticity. The methods studied here are general, however, and can be extended to many other application areas.

- The linear elastic model is described by a system of self-adjoint PDEs that facilitate computation of the cost functional gradient with respect to the high dimensional, grid-defined model parameters. At any configuration, the gradient of the cost functional with respect to the model parameters may be computed using two simulations (a forward and adjoint simulation).

- Using geostatistical methods—specifically the single normal equation simulation (SNESIM) algorithm [28]—we generate statistical realizations of a complex property model used as the basis for prior knowledge. Then, a low-dimensional feature space is obtained by performing KPCA on the generated geostatistical realizations.

- The feature random variables obtained from the KPCA are uncorrelated but not Gaussian. In general, Bayesian frameworks requires frequent sampling on these feature random variables. To improve sampling efficiency, we sample them using a polynomial chaos expansion (PCE) coupled with an inverse cumulative distribution function (ICDF) transformation.

- We then construct an automatic differentiation-based discretized adjoint model of the KPCA-based and PCE-based ICDF transformation, and couple the discretized adjoint model with the high-fidelity adjoint PDE model. This approach provides gradients of the cost functional with respect to the low-dimensional feature random variables.

- Bayesian inference is then performed on the low-dimensional feature space using an efficient LMCMC scheme. The convergence rate of this KPCA and gradient-based stochastic inversion through MCMC is greatly improved, thanks to the nonlinear control reduction with good classification and clustering properties.

- Unlike traditional machine learning problems, this process in each MCMC iteration step requires the projection of the low-dimensional feature space back to the high-dimensional parameter space, since the high-fidelity forward models are functions of grid-defined model parameters. The projection is obtained by exploring both local fixed-point iteration and non-iterative algebra approaches.

- This projection from the feature space back to parameter space gives us access to posterior pdf of the grid-defined high-dimensional model parameters.

The remainder of this paper is organized as the following. Section 2 provides the mathematical framework of our procedure, providing a detailed derivation of each step in the proposed method. To help guide the reader through these developments, Table 1.1 provides a summary of the proposed workflow and the key challenges each step seeks to address; Figures 1.1 and

1.2 provide the flowchart of the mapping from the parameter space to Gaussian space and posterior sample generation with proposed approach, respectively; and Algorithms 1 and 2 provide a concise summary of the steps necessary to implement the methodology. In Section 3, we apply this methodology to identify elastic properties of a geologically complex system. Section 3.5 gives some insights on the advantages of KPCA and the implementation of the proposed method for stochastic inversion. Finally, conclusions are given in Section 3.5.1, with an outline of future work.

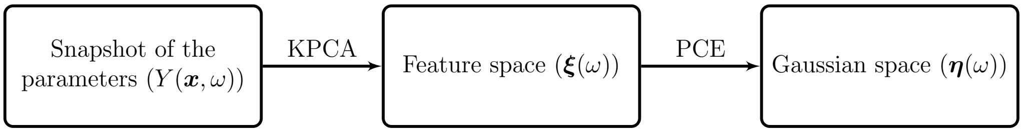

Snapshot of the parameters $(Y(\boldsymbol{x},\omega))$
KPCA
Feature space $(\boldsymbol{\xi}(\omega))$
PCE
Gaussian space $(\boldsymbol{\eta}(\omega))$

Fig. 1.1. Mapping from the parameter space to Gaussian space

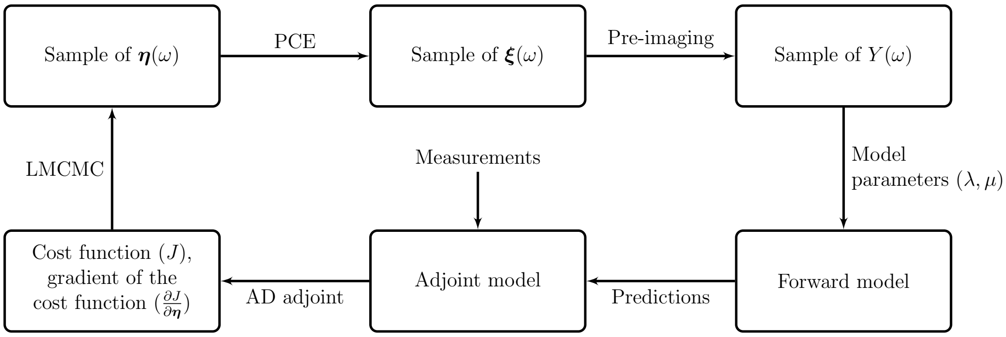

Sample of $\boldsymbol{\eta}(\omega)$
PCE
Sample of $\boldsymbol{\xi}(\omega)$
Pre-imaging
Sample of $Y(\omega)$
LMCMC
Measurements
Model parameters (λ,μ)
Cost function (J), gradient of the cost function $(\frac{\partial J}{\partial \boldsymbol{\eta}})$
AD adjoint
Adjoint model
Predictions
Forward model

Fig. 1.2. Posterior sample generation with proposed approach

Table 1.1: Summary of the proposed approach

Section	Challenge	Approach	Explanation
2.1	High-fidelity gradient computation	Adjoint gradient	Numerical gradient computation using finite difference methods requires many forward model runs. Here, the self-adjoint PDE allows us to compute gradients in the parameter space with two model runs (a forward and adjoint simulation).
2.2	High dimensionality of the parameters	KPCA	KPCA is used to find a low-dimensional feature space where the solution is not an outlier in the prior probability space.
2.3	Sampling non-Gaussian feature random variables	PCE	KPCA feature random variables are uncorrelated but dependent on non-Gaussian random variables. An ICDF transformation is used to build the PCE of the feature random variables to facilitate efficient sampling.
2.4	Ill-posedness of the inverse problem	Bayesian inference	Sparse and noisy measurements and high-dimensionality of the parameter space make the inverse problem ill-posed. Bayesian inference provides a systematic way to address these problems and provides a probabilistic inverse solution.
2.5–2.6	Computational intractability of the MCMC	LMCMC and automatic differentiation	Gradient free MCMC ($O(n)$) quickly runs into computational intractability as the problem size of the parameter space increases. Gradient based LMCMC ($O(n^{1/3})$) is used to make the solution tractable by performing inversion in the lower-dimensional feature space and leveraging the derivative information of feature random variables obtained by automatic differentiation.

Chapter 2

Mathematical Formulation

2.1 Elasticity model

This section introduces a model problem to test the proposed inversion approach: the deformation behavior of a linear elastic body under mechanical loads. The governing PDE is a linear momentum balance equation involving two elastic coefficients, the Lamé parameters of the material [**?**]. The goal is to estimate these material properties based on sparse measurement data and prior knowledge.

Let the physical domain $\mathcal{D} \subset \mathbb{R}^d$, $d = 2, 3$, be a bounded, connected, open and Lipschitz continuous domain with a boundary $\Gamma = \partial\mathcal{D}$. Assume Γ_D and Γ_N are two subsets of Γ such that $\Gamma_D \cap \Gamma_N = \emptyset$ and $\Gamma_D \cup \Gamma_N = \Gamma$. Let Dirichlet and Neumann boundary conditions (prescribed displacements and prescribed tractions) be specified along Γ_D and Γ_N, respectively. For an integer $m \geq 0$, we follow the classical notation of a standard Sobolev space $H^m(\mathcal{D})$ with norm $||\cdot||_m$ in accordance with Adams et al. [29].

To express the governing PDE in variational form—suitable for finite element discretization—let

$$\mathcal{U} = \{\mathbf{u} : \mathcal{D} \to \mathbb{R}^d \mid \mathbf{u} \in H^1, \mathbf{u} = \overline{\mathbf{u}} \text{ on } \Gamma_D\}, \tag{2.1}$$

$$\mathcal{V} = \{\mathbf{v} : \mathcal{D} \to \mathbb{R}^d \mid \mathbf{v} \in H^1, \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\} \tag{2.2}$$

be spaces of trial displacement fields $\mathbf{u}(\mathbf{x})$ and weighting functions $\mathbf{v}(\mathbf{x})$. Prescribed displacement boundary conditions $\overline{\mathbf{u}}$ are assigned on Γ_D. The weak problem is then to find $\mathbf{u} \in \mathcal{U}$ such that, for all $\mathbf{v} \in \mathcal{V}$, the following linear momentum balance equation is satisfied,

$$a(\mathbf{v},\mathbf{u}) = (\mathbf{v},\mathbf{f}) + (\mathbf{v},\mathbf{t})_{\Gamma_N}, \tag{2.3}$$

where the respective bilinear forms are

$$a(\mathbf{v},\mathbf{u}) = \int_{\mathcal{D}} \lambda(\nabla\cdot\mathbf{v})(\nabla\cdot\mathbf{u})\,d\mathcal{D} + \int_{\mathcal{D}} 2\mu(\nabla^s\mathbf{v} : \nabla^s\mathbf{u})\,d\mathcal{D}, \tag{2.4}$$

$$(\mathbf{v},\mathbf{f}) = \int_{\mathcal{D}} \mathbf{v}\cdot\mathbf{f}\,d\mathcal{D}, \tag{2.5}$$

$$(\mathbf{v},\mathbf{t})_{\Gamma_N} = \int_{\Gamma_N} \mathbf{v}\cdot\mathbf{t}\,d\Gamma. \tag{2.6}$$

Here, $\nabla^s = (\nabla + \nabla^T)/2$ is the symmetric gradient operator, $\mathbf{f}$ is a body force due to self-weight, and $\mathbf{t}$ is an externally applied traction on Γ_N. The two material coefficients $\lambda(\mathbf{x})$ and $\mu(\mathbf{x})$ are the Lamé parameters describing the elastic properties of the body.

For brevity, we omit most of the details of the finite element discretization, as they are standard [**?**]. We introduce a partition of $\mathcal{D}$ into non-overlapping elements $\mathcal{D}^e$. On this mesh, both vector and scalar fields are discretized using bilinear or trilinear basis functions $\{\phi^a\}$ as

$$\mathbf{u}^h(\mathbf{x}) = \sum_{a=1}^{n_{\text{nodes}}} \mathbf{u}^a\phi^a(\mathbf{x}), \tag{2.7}$$

$$\mathbf{v}^h(\mathbf{x}) = \sum_{a=1}^{n_{\text{nodes}}} \mathbf{v}^a\phi^a(\mathbf{x}), \tag{2.8}$$

$$\lambda^h(\mathbf{x}) = \sum_{a=1}^{n_{\text{nodes}}} \lambda^a\phi^a(\mathbf{x}), \tag{2.9}$$

$$\mu^h(\mathbf{x}) = \sum_{a=1}^{n_{\text{nodes}}} \mu^a\phi^a(\mathbf{x}), \tag{2.10}$$

where the coefficients represent the nodal values of each field. Introducing these discrete fields into the variation form (2.3), the problem can be recast as a discrete linear system

$$Au = b \tag{2.11}$$

whose solution u is an algebraic vector of unknown displacement components at the mesh nodes. We will refer to the solution of this linear system as the *forward simulation*.

The matrix A depends on the material properties $\lambda^h(\mathbf{x})$ and $\mu^h(\mathbf{x})$. These material properties are assigned at each node of the mesh. Let p denote an algebraic vector containing the property coefficients $\{\lambda^a, \mu^a\}$. The vector p has dimension $2 \times n_{\text{nodes}}$. The vector space of possible p configurations is therefore extremely large for highly-refined meshes. Attempting to solve an inverse problem for p in this space is challenging. It will be even more challenging to provide the uncertainty information in this space.

Assuming discrete observations u^{obs} are available in certain locations, a simple cost functional can be defined as

$$J(p) = \frac{1}{2} e^T D e \qquad \text{with} \qquad e_i = u_i - u_i^{\text{obs}}, \tag{2.12}$$

where D is a diagonal matrix containing weighting coefficients for each observation. For a displacement component u_i where no observational data is available, the corresponding diagonal entry D_{ii} is zero. Note that additional terms can be added to the cost functional to include regularization terms and other types of observational data beyond displacements.

The minimization of the cost functional is an optimization problem that can benefit from the calculation of gradient information. In particular, the gradient vector g has components

$$g_i = \frac{\partial J}{\partial p_i} = \frac{\partial J}{\partial u_j} \frac{\partial u_j}{\partial p_i} = e_k D_{kj} \frac{\partial u_j}{\partial p_i}. \tag{2.13}$$

Here, summation over repeated indices is implied. By differentiating equation (2.11) with respect to p, we find [30, **?**]

$$\frac{\partial A_{mn}}{\partial p_i} u_n + A_{mj} \frac{\partial u_j}{\partial p_i} = 0. \tag{2.14}$$

and therefore,

$$\frac{\partial u_j}{\partial p_i} = -A^{-1}_{jm} \frac{\partial A_{mn}}{\partial p_i} u_n. \tag{2.15}$$

Inserting this expression into the gradient formula and using the symmetry properties of A, the gradient can be expressed as

$$g_i = -w_m \frac{\partial A_{mn}}{\partial p_i} u_n, \tag{2.16}$$

where the vector w is the solution of the linear system,

$$Aw = De. \tag{2.17}$$

Note that this system is similar to equation (2.11) due to the self-adjoint nature of the underlying PDE. We will refer to the solution of this system as the *adjoint simulation*. Once the fields $\boldsymbol{u}(\mathbf{x})$ and $\boldsymbol{w}(\mathbf{x})$ are computed by solving the forward and adjoint systems, equation (2.16) allows individual components of the gradient vector to be computed explicitly as

$$g_i^{\mu} = \frac{\partial J}{\partial \mu^i} = \int_{\mathcal{D}} 2\phi^i (\nabla^s \mathbf{w}^h : \nabla^s \mathbf{u}^h) d\mathcal{D}, \tag{2.18}$$

$$g_i^{\lambda} = \frac{\partial J}{\partial \lambda^i} = \int_{\mathcal{D}} \phi^i (\nabla \cdot \mathbf{w}^h)(\nabla \cdot \mathbf{u}^h) d\mathcal{D}. \tag{2.19}$$

2.2 Discretization of the random field and kernel principal component analysis

The high dimensionality of the discretized parameter space can lead to intractability of the stochastic inversion problem. This section introduces a KPCA method to find a low-dimensional but relevant feature space.

To describe the stochastic nature of the PDE, let Ω be a sample space associated with probability triplet $(\Omega, \mathcal{F}, \mathbb{P})$ where $\mathcal{F} \subset 2^{\Omega}$ is a σ-algebra of the events in Ω and $\mathbb{P}$ is the probability measure $\mathbb{P} : \mathcal{F} \to [0, 1]$. We assume the two material coefficients $\mu(\boldsymbol{x}, \omega) : \mathcal{D} \times \Omega \to \mathbb{R}$ and $\lambda(\boldsymbol{x}, \omega) : \mathcal{D} \times \Omega \to \mathbb{R}$—the elastic Lamé parameters—are now random fields belonging to an infinite-dimensional probability space.

Let $Y(\boldsymbol{x}, \omega) := \ln(\mu(\boldsymbol{x}, \omega))$ be a random field. The covariance function can be defined as $C_Y(\boldsymbol{x}, \boldsymbol{y}) =< \tilde{Y}(\boldsymbol{x}, \omega)\tilde{Y}(\boldsymbol{y}, \omega) >_{\omega}$, where $\tilde{Y}(\boldsymbol{x}, \omega) := Y(\boldsymbol{x}, \omega) - < Y(\boldsymbol{x}, \omega) >_{\omega}$ and $< . >_{\omega}$ is an expectation operator. Assuming C_Y is bounded, symmetric and positive definite, it can be represented as [31]

$$C_Y(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{\infty} \gamma_i e_i(\boldsymbol{x}) e_i(\boldsymbol{y}), \tag{2.20}$$

where $\gamma_1 \geq \gamma_2 \geq \cdots > 0$ are the eigenvalues, and $e_i(\boldsymbol{x})$ and $e_j(\boldsymbol{y})$ are deterministic and mutually orthogonal functions,

$$\int_{\mathcal{D}} e_i(\boldsymbol{x}) e_j(\boldsymbol{x}) \, d\boldsymbol{x} = \delta_{ij}, \quad i, j \geq 1. \tag{2.21}$$

Using Karhunen-Loève (KL) expansion, the random process $\overline{Y}(\boldsymbol{x}, \omega)$ can be expressed in terms of $e_i(\boldsymbol{x})$ as

$$\overline{Y}(\boldsymbol{x}, \omega) = \sum_{i=1}^{\infty} \xi_i(\omega)\sqrt{\gamma_i} e_i(\boldsymbol{x}), \tag{2.22}$$

where $\{\xi_i(\omega)\}$ are zero-mean and uncorrelated random variables, i.e.,$< \xi_i(\omega) >_\omega = 0$ and $< \xi_i(\omega)\xi_j(\omega) >_\omega = \delta_{ij}$. The eigenvalues $\{\gamma_i\}$ and the eigenfunctions $f_i(\boldsymbol{x})$ are obtained by solving the following integral equation either analytically or numerically,

$$\int_{\mathcal{D}} C_Y(\boldsymbol{x}, \boldsymbol{y}) f_i(\boldsymbol{x})\, d\boldsymbol{x} = \gamma_i e_i(\boldsymbol{y}),\ i = 1, 2, \ldots. \tag{2.23}$$

The attenuation of the eigenvalues $\{\gamma_i\}$ allows truncation of the infinite sum in Equation (2.22) up to N_R terms,

$$\overline{Y}(\boldsymbol{x}, \omega) \approx \sum_{i=1}^{N_R} \xi_i(\omega)\sqrt{\gamma_i} e_i(\boldsymbol{x}), \tag{2.24}$$

where N_R is the stochastic dimension. The KL expansion is optimal [17] in the sense that it minimizes the mean-square error out of all possible orthonormal bases in $L^2(\mathcal{D} \times \Omega)$.

In practice, a closed form expression for the C_Y is rarely available. Instead, a numerical approximation to the $C_Y(\boldsymbol{x}, \boldsymbol{y})$ is obtained using realizations of $Y(\boldsymbol{x}, \omega)$ as:

$$C_Y(\boldsymbol{x}, \boldsymbol{y}) \approx \frac{1}{M} \sum_{i=1}^{M} (Y(\boldsymbol{x}, \omega_i) - < Y(\boldsymbol{x}, \omega_i) >_\omega)(Y(\boldsymbol{y}, \omega_i) - < Y(\boldsymbol{y}, \omega) >_\omega)^T, \tag{2.25}$$

where M is the number of realizations extracted from the random field $Y(\boldsymbol{x}, \omega)$. Given C_Y, approximation to Equation (2.23) can be obtained using the Nystrom algorithm [32] as

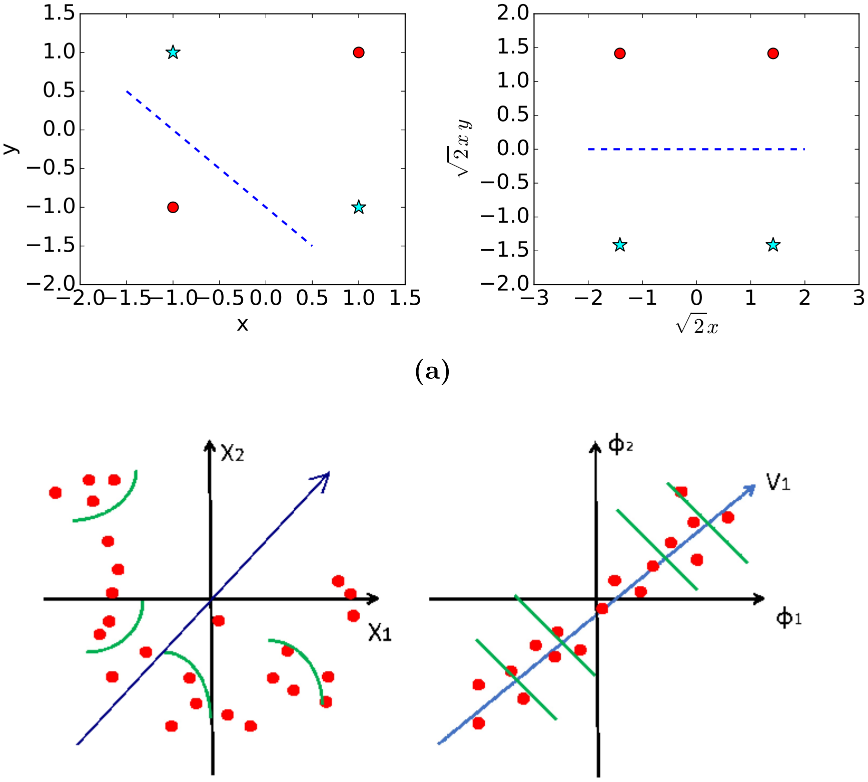
$$\sum_{i=1}^{M} w_i C_Y(\boldsymbol{x}_i, \boldsymbol{y}) e(\boldsymbol{x}_i) = \gamma e(\boldsymbol{y}). \tag{2.26}$$

Here, M is the number of sample points where realizations $\boldsymbol{x}_i$'s are provided, and w_i's are weights of the quadrature rule. Assuming we have enough sample points and equal weights $w_i = \frac{1}{M}$, equation (2.26) can be solved by simple eigen-decomposition of $C_Y(\boldsymbol{x}_i, \boldsymbol{y})$, for which principal component analysis (PCA) [33] can be used to reduce the dimension.

The current data assimilation framework has the ability to infuse various sources of information into the Bayesian framework. For instance, the application considered in this paper is the elastic deformation of subsurface geologic formations under mechanical loads. Along with displacement measurements (model solutions), we often have access to elasticity parameter measurements (hard data) at a few sparse locations obtained from wells. In addition, geophysical parameters can be obtained with 3D seismic observations (soft data). The soft and hard data are generally used to generate geostatistical realizations of model parameters. For instance, a simple geostatistical spatial random process for the prior parameter field can be obtained with two point statistical methods such as Kriging [34, 35, 36]. A more general category of data-driven methods that build on soft and hard data measurements includes multi-point statistics (MPS) [28], soft computing methods such as neural network, fuzzy logic, support vector machines [**?**, **?**, **?**, **?**, **?**] and Gaussian process on manifolds [37]. In the numerical examples, we will use MPS to generate elastic property models describing complex channelized structures frequently encountered in the subsurface.

The stochastic dimension of the prior model obtained using MPS is proportional to the number of finite element grid points in the simulation model. Equation (2.26), which is equivalent of performing PCA of the covariance matrix, can be used to reduce this dimension size. However, in general, PCA can only obtain efficient embeddings for linearly correlated data points. Recently, Sarma [26] and Ma [27] have shown that KPCA is an appealing alternative for dealing with complex prior models.

We use two simple examples to demonstrate the desirable properties of KPCA. Figures 2.1 (a) and (b) depict a classification problem where the objective is to classify a XOR dataset [33]. KPCA with a second-order polynomial kernel can classify data perfectly, while PCA has a lower accuracy. Figures 2.1 (c) and (d) show another example [33], the goal of which is to reduce the dimensionality of a non-linear dataset that lies across a curve. It indicates that a KPCA-based one-dimensional (1D) subspace is closer to true data than a PCA-based 1D subspace. In the following, we take advantage of both dimension reduction and improved feature representation properties of KPCA to increase the efficiency of stochastic inversion. Specifically, KPCA is used to find a low-dimensional and relevant feature space where the solution is not an outlier in the prior probability space.



(a)

(b)

Fig. 2.1. KPCA motivating examples: (a) data classification with PCA (left) and KPCA (right) (b) non-linear dimension reduction of a non-linear dataset with PCA (left) and KPCA (right)

For the sake of completeness, we include a brief matrix derivation of KPCA below. More comprehensive derivations can be found in Schölkopf [38, 39] and Sarma [26]. Let N_R be a positive integer representing the dimension of the random field (in this case it is equal to the number of mesh grid points), and M be the number of observations of the random field. Given a set of discrete realizations $\{\mathbf{y}_l\}_{l=1}^{M}$ of the random field where each component (or *snapshot*) is $\mathbf{y}_l = [y_{1,l}, \ldots, y_{N,l}]^T \in \mathbb{R}^N$, $l = 1, 2, \ldots, M$, we define a linear or nonlinear mapping Φ as:

$$\Phi : \mathbb{R}^{N_R} \rightarrow \mathbb{R}^{N_F},\ y_l \rightarrow \Phi(y_l) \in \mathbb{R}^{N_F},\ l = 1, 2, \ldots, M, \tag{2.27}$$

where $\mathbb{R}^{N_F}$ is the new induced feature space. Here, $N_F \gg N_R$, and the feature space $\mathbb{R}^{N_F}$ in general contains much more information (that is, higher dimension) than the original space $\mathbb{R}^{N_R}$. For convenience, we introduce matrix notations $\mathbf{Y} := [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M]$ and $\mathbf{\Phi} := [\mathit{\Phi}(\mathbf{y}_1), \mathit{\Phi}(\mathbf{y}_2), \dots, \mathit{\Phi}(\mathbf{y}_M)]$. In addition, let $\mathbf{1}_M := \frac{1}{M}\mathbf{1}_{N_R \times M}$ be a matrix with all its elements equal to $\frac{1}{M}$; and let $\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{Y}\mathbf{1}_M$ and $\tilde{\mathbf{\Phi}} := \mathbf{\Phi} - \mathbf{\Phi}\mathbf{1}_M$ be the centered matrix of $\mathbf{Y}$ and $\mathbf{\Phi}$, respectively.

In classical PCA, a discrete covariance matrix [40] is obtained as

$$\mathbf{C}_o := \frac{1}{M}\sum_{l=1}^{M} \tilde{\mathbf{y}}_l \tilde{\mathbf{y}}_l^T = \frac{1}{M}\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T. \tag{2.28}$$

Here, the set $\{\tilde{\mathbf{y}}_l\}_{l=1}^M$ is a centered measurement vector given by $\tilde{\mathbf{y}}_l = \mathbf{y}_l - \bar{\mathbf{y}}$, where $\bar{\mathbf{y}} = \frac{1}{M}\sum_{l=1}^{M}\mathbf{y}_l$. Similar to the continuous version of the KL expansion with given mean and covariance kernel function, the KL expansion of the random fields for the discrete case can be characterized with following equation based on Mercer's theorem:

$$\mathbf{y} = D_o \Lambda_o^{1/2} \boldsymbol{\xi} + \mathbf{Y}\mathbf{1}_1, \tag{2.29}$$

where D_o is a matrix of eigenvectors associated with $\mathbf{C}_o$; Λ_o is a diagonal matrix of the eigenvalues of $\mathbf{C}_o$; $\boldsymbol{\xi} = [\xi_1, \xi_2, \cdots, \xi_{N^R}]^T \in \mathbb{R}^{N_R}$ is a column random vector with statistical properties $< \xi_i \xi_j >_\omega = \delta_{i,j}$ and $< \xi_i >_\omega = 0$. A nonlinear choice for the Φ such as radial basis functions leads to the nonlinear form of PCA. Next, we compute the centralized form of the feature vectors $\{\tilde{\Phi}, (\mathbf{y}_l)\}_{l=1}^M$ where $\tilde{\Phi}(\mathbf{y}_l) = \Phi(\mathbf{y}_l) - \bar{\Phi}$, $\bar{\Phi} = \frac{1}{M}\sum_{l=1}^{M}\Phi(\mathbf{y}_l)$. Similar to PCA, we have the following discrete covariance after the nonlinear mapping

$$\mathbf{C}_f = \frac{1}{M}\sum_{l=1}^{M} \tilde{\Phi}(\mathbf{y}_l)\tilde{\Phi}(\mathbf{y}_l)^T = \frac{1}{M}\tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^T. \tag{2.30}$$

Since N_F is usually much larger than N_R, it is infeasible in practice to perform PCA on the feature space due to the very high dimensionality of the covariance matrix. For instance, for the polynomial kernel $(\mathbf{x} \cdot \mathbf{y})^d$ of order d, the dimension of the feature space will be [38]

$$N_F = \frac{(N_R + d - 1)!}{d!(N_R - 1)!}. \tag{2.31}$$

Alternatively, the nonlinear mapping can be seen as a kernel map, thus allowing us to handle the high dimensionality by using a technique called a "kernel trick." A kernel trick introduces a virtual mapping Φ, from beginning to the end, where the mapping Φ only acts as an intermediate functional, resulting in smaller dimensional equivalent system compared to $\mathbf{C}$. The eigen-problem of the covariance matrix $\mathbf{C}_f$ in the feature space is now given as:

$$\mathbf{C}_f \mathbf{V}_f = \mathbf{V}_f \Lambda_f. \tag{2.32}$$

Here, $\mathbf{V}_f$ is the matrix of eigenvectors and Λ_f is a diagonal eigenvalue matrix. The relationship between the eigenvectors $\{\mathbf{v}_l\}$ of $\mathbf{V}_f$ and the data set of $\{\tilde{\Phi}(\mathbf{y}_l)\}$, can be written as

$$\mathbf{C}_f \mathbf{v}_l = \frac{1}{M} \sum_{j=1}^{M} \tilde{\Phi}(\mathbf{y}_j) \tilde{\Phi}(\mathbf{y}_i)^T, \quad \mathbf{v}_l = \frac{1}{N_R} \sum_{j=1}^{M} (\tilde{\Phi}(\mathbf{y}_i)^T \mathbf{v}_l) \tilde{\Phi}(\mathbf{y}_j) = \gamma_l \mathbf{v}_l, \tag{2.33}$$

which shows that the eigenvectors $\{\mathbf{v}_l\}$ are elements in the space spanned by $\tilde{\Phi}(\mathbf{y}_l)$, $l = 1, \ldots, M$.

Let $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M]$ with $\boldsymbol{\alpha}_l = [\alpha_{l,1}, \alpha_{l,2}, \ldots, \alpha_{l,N_R}]^T$, and eigenmatrix $\mathbf{V}_f = \tilde{\boldsymbol{\Phi}} \boldsymbol{\alpha}$ where each component of the eigenvector $\mathbf{v}_l = \sum_{j=1}^{N_R} \alpha_{l,j} \tilde{\Phi}(y_i) = \tilde{\boldsymbol{\Phi}} \boldsymbol{\alpha}_l$. Substituting this into Equation (2.33) leads to

$$\mathbf{C}_f \tilde{\Phi} \boldsymbol{\alpha} = \tilde{\Phi} \boldsymbol{\alpha} \Lambda_f. \tag{2.34}$$

Using the definition of $\mathbf{C}_f$ from Equation (2.30) and multiplying both sides by $\tilde{\Phi}^T$, and further setting $K_c = \tilde{\Phi}^T \tilde{\Phi}$, we have

$$\frac{1}{M} K_c^2 \mathbf{V} = K_c \boldsymbol{\alpha} \Lambda_f. \tag{2.35}$$

Assuming K_c is a nonsingular matrix, the equation above is equivalent to the following kernel eigenvalue problem

$$\frac{1}{M} K_c \mathbf{V} = \boldsymbol{\alpha} \Lambda_f, \tag{2.36}$$

where K_c is a matrix of $M \times M$. This kernel trick allows us to perform KPCA in the high dimensional feature space, with similar computational expense as PCA. We just need to perform an eigen-decomposition on a relatively small space $\mathbb{R}^M$, which is independent of the selection of the nonlinear mapping and the feature space.

Solving Equation (2.36) leads to the eigenvector matrix $\mathbf{V}$, and the corresponding $\mathbf{V}_f$ in Equation (2.32) can be retrieved using,

$$\mathbf{V}_f = \tilde{\mathbf{\Phi}}\mathbf{V}. \tag{2.37}$$

Here, $\mathbf{V}_f$ has the property that

$$\mathbf{V}_f^T\mathbf{V}_f = \mathbf{V}^T\tilde{\Phi}^T\tilde{\Phi}\mathbf{V} = \mathbf{V}^T K_c \mathbf{V} = M\Lambda_f. \tag{2.38}$$

Using the same notation of $\mathbf{V}_f$, we have the orthonormal eigenvector matrix

$$\mathbf{V}_f = \frac{1}{\sqrt{M}}\tilde{\Phi}\mathbf{V}\Lambda_f^{-1/2}. \tag{2.39}$$

Assuming $K = \Phi^T\Phi$, the centered K_c can be easily obtained using

$$\begin{aligned} K_c &= (\Phi - \bar{\Phi})^T(\Phi - \bar{\Phi}) = (\Phi - \Phi 1_{N_R})^T(\Phi - \Phi 1_{N_R}) \\ &= \Phi^T\Phi - \Phi^T\Phi\mathbf{1}_{N_R} - \mathbf{1}_{N_R}^T\Phi^T\Phi + \mathbf{1}_{N_R}\Phi^T\Phi\mathbf{1}_{N_R} \\ &= K - K\mathbf{1} - \mathbf{1}K + \mathbf{1}K\mathbf{1} \end{aligned}$$

Thus, we have the KL expansion in the feature space as

$$\mathbf{Y}_f = \mathbf{V}\Lambda^{1/2}\boldsymbol{\xi} + \bar{\Phi} = \frac{1}{\sqrt{M}}\tilde{\Phi}\mathbf{V}\Lambda_F^{-1/2}\Lambda^{1/2}\boldsymbol{\xi} + \bar{\Phi} = \frac{1}{\sqrt{M}}\tilde{\Phi}\mathbf{V}\boldsymbol{\xi} + \bar{\Phi}, \tag{2.40}$$

where $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_{N_R}]^T$ is a random vector with properties $\mathbb{E}[\xi_i] = 0, \mathbb{E}[\xi_i\xi_j] = \delta_{i,j}$. The polynomial kernel and Gaussian kernel defined below are frequently used in practice, which are given by

$$k(\mathbf{x}, \mathbf{y}) = c + (\mathbf{x} \cdot \mathbf{y})^d, \ d \geq 1, \tag{2.41}$$

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\tfrac{||\mathbf{x}-\mathbf{y}||^2}{\sigma}), \ \sigma > 0, \tag{2.42}$$

respectively. Kernel functions directly calculate the dot product in the space of $\mathbb{R}^F$ using elements in the input space $\mathbb{R}^{N_R}$. Since there is no actual mapping of $\Phi(y)$, kernels play the role of the intermediate functional.

Although stochastic inversion is performed in the feature space, our interest is to obtain the snapshots from the posterior in the original space $\mathbb{R}^{N_R}$. In order to achieve this, a pre-imaging problem is solved to project snapshots from the feature space back to the original space. In general, due to the non-linearity of the mapping Φ, neither existence nor uniqueness of the pre-image is guaranteed. One method to perform pre-imaging involves solving the following optimization problem [38],

$$\min_{\mathbf{y}} \rho(\mathbf{y}) = ||\Phi(\mathbf{y}) - \boldsymbol{Y}||^2\,, \tag{2.43}$$

where the $\mathbf{y} \in \mathbb{R}^{N_F}$ and $\boldsymbol{Y} \in \mathbb{R}^{N_R}$ are points in the feature space and original space, respectively, and $\|\cdot\|$ is the Euclidean norm. The above minimization problem can be reduced to the following iterative fixed point problem [26, 38]

$$\mathbf{y}^{k+1} = \frac{\sum_{l=1}^{N_R} \beta_i \sum_{j=1}^{d} j(\mathbf{y}_i \cdot y^k)^{j-1}\mathbf{y}_i}{\sum_{l=1}^{N_R} \beta_i \sum_{j=1}^{d} j(\mathbf{y}_i \cdot y^k)^{j-1}}. \tag{2.44}$$

Note here that non-iterative pre-imaging techniques based on reproducing kernel Hilbert space (RKHS) also developed by several researchers [41, **?**, **?**, **?**] and a comprehensive comparison these methods can be found in [**?**].

The resulting KPCA method allows us to find a low-dimensional, relevant feature space and obtain a pre-image. The next section introduces a procedure to efficiently sample the KPCA-feature random variables.

2.3 Mapping non-Gaussian feature random variables to Gaussian random variables

KPCA feature random variables are uncorrelated but dependent non-Gaussian random variables. This section introduces a ICDF-transformation-based PCE construction to sample from the feature random variables.

Let $\boldsymbol{\xi}^d$ be the discrete observations of $\boldsymbol{\xi}$ obtained from the measurements of the snapshots $\{\mathbf{y}_l\}_{l=1}^M$. Letting $\mathbf{Y}_f = \boldsymbol{\Phi}$ and multiplying both sides of Equation (2.40) by $\boldsymbol{\Phi}^T$, we obtain

$$\tilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi} - \boldsymbol{\Phi}\mathbf{1}_M = \frac{1}{\sqrt{M}}\tilde{\Phi}\mathbf{V}\boldsymbol{\xi}^d \Rightarrow K_c = \frac{1}{\sqrt{M}}K_c\mathbf{V}\boldsymbol{\xi}^d. \tag{2.45}$$

Assuming K_c is nonsingular, we have

$$\mathbf{V}\boldsymbol{\xi}^d = \sqrt{M}\mathbf{1}_M \tag{2.46}$$

which can be solved using a least-squares method or singular value decomposition (SVD).

Random variables $\boldsymbol{\xi}^d$ computed from Equation (2.46) act as a prior distribution for the Bayesian inversion framework. In general, $\boldsymbol{\xi}^d$ are non-Gaussian, uncorrelated and dependent random variables, which may complicate the Bayesian inversion procedures (e.g. by requiring more frequent sampling from their distributions).

Determination of a unique map from the dependent $\boldsymbol{\xi}^d$ to a standard independent random variable space $\boldsymbol{\eta}$ is an active research area. One way to achieve a non-unique mapping is using iso-probabilistic mappings such as the generalized Nataf transformation [42] and Rosenblatt transformation [43]. However, these transformations require information such as conditional distributions, which are hard to construct from limited observations. Therefore, we assume $\{\xi_l^d\}_{l=1}^M$ are independent similar to [44, 45], and to facilitate the sampling we construct a polynomial chaos expansion (PCE) for each ξ_l^d.

PCE, originally introduced by Wiener [46, 17], represents any random variable with finite variance as a summation of a series of polynomials over the centered normalized Gaussian variables. We can represent each component of $\{\xi_l^d\}_{l=1}^M$ obtained from Equation (2.46) using PCE as

$$\xi_l^d = \sum_{n=0}^{\infty} c_{n,l} \Psi_n(\eta_l(\omega)), l = 1, 2, ..., \tag{2.47}$$

where η_l are i.i.d. standard Gaussian random variables, $\Psi_n(\eta_l(\omega))$ are Hermite polynomials, and $c_{n,l}$ are real valued deterministic coefficients. The associated orthogonal system $\{\Psi_n(\eta)\}_{n\in\mathbb{N}}$ forms the homogeneous polynomial chaos basis. The coefficients in the equation above can be computed using Bayesian inference [47] or using a non-intrusive projection method [48]. We use a projection method [49] to find a continuous parameterized representation similar to Equation (2.47) based on the discrete $\boldsymbol{\xi}^d$. Let $\{\eta_l\}$ be a standard Gaussian random variable, then by matching the cumulative density function (cdf) of ξ_l^d and η_l, each component of ξ_l can be expressed in terms of random variables η_l by following non-linear mapping:

$$\xi_l^d = F_{\xi_l^d}^{-1} \circ F_{\eta_l}(\eta_l), \tag{2.48}$$

where $F_{\xi_l^d}$ and F_{η_l} denote the cdfs of ξ_l^d and η_l respectively. The coefficients of the PCE are then computed using the projection of $F_{\xi_i^d}^{-1} \circ F_{\eta_l}$ on the orthonormal chaos basis system,

$$c_{n,l} =< \xi_l^d, \Psi_n >= \int_\Omega F_{\xi_l^d}^{-1} \circ F_{\eta_l} \Psi_n d\mathbb{P}_\eta(\omega), \tag{2.49}$$

However, the cdf $F_{\xi_l^d}$ is not known and needs to be estimated using the empirical cdf [50] based on the discrete observations of $\boldsymbol{\xi}^d$. The empirical cdf ($\tilde{F}_{\xi_l^d}$) of ξ_l^d can be estimated from sampling using,

$$\tilde{F}_{\xi_i^d}(x) = \frac{1}{M}\sum_{k=1}^{M} I(\xi_l^{d(k)} \leq x), \tag{2.50}$$

where $I(A)$ is the indicator function of event A. We then introduce the following approximation

$$F^{-1}_{\xi_i^d} \sim \tilde{F}^{-1}_{\xi_i^d}, \text{ where } \tilde{F}^{-1}_{\xi_i^d} : [0,1] \to \mathbb{R} \tag{2.51}$$

which is uniquely defined as

$$\tilde{F}^{-1}_{\xi_i^d}(y) = \min\{x \in \{\xi^{(k)}_{l^d}\}_{k=1}^{M}; \tilde{F}_{\xi_i^d}(x) \geq y\}. \tag{2.52}$$

Then the coefficients of the polynomial chaos expansion can be computed using a numerical integration. Instead of using the indicator functions, we use kernel density estimation [51] to construct the empirical cdf,

$$\tilde{f(\xi)} = \frac{1}{M}\sum_{l=1}^{M} K_h(\xi - \xi_l), \tag{2.53}$$

where $K_h(\cdot)$ is the kernel function.

$$c_{n,l} =< \xi_l, \Psi_n >= \int_\Omega F^{-1}_{\xi_l^d} \circ F_{\eta_l^d} \Psi_n d\mathbb{P}_\eta(\omega), = \int_\Omega F^{-1}_{\xi_l^d} \circ F_{\eta_l^d} \Psi_n \frac{e^{-\eta^2/2}}{\sqrt{2\pi}} dx \tag{2.54}$$

The coefficients $c_{n,l}$ can be efficiently calculated using the Gauss-Hermite quadrature rules.

The above procedure allows us to sample from the feature random variables within the Bayesian inference framework.

2.4 Bayesian inference

Bayesian inference provides a systematic framework for integrating prior knowledge and measurement uncertainties and computes a probabilistic solution to the inverse problem. It treats the parameters $\mu(\boldsymbol{x})$, $\lambda(\boldsymbol{x})$ of the forward model (2.3) as a random process. Instead of performing Bayesian inference with respect to these parameters directly, we perform the inference in the extracted feature space of $\boldsymbol{\eta}$. We denote the stochastic elasticity forward model (2.3) as $\mathbf{u} = f(\boldsymbol{\eta})$, which describes the relationship between the observed output state $\mathbf{u}_{obs}$ and the uncertain model parameters $\boldsymbol{\eta}$. As such, the posterior distribution from the Bayesian inference can be expressed as

$$\pi_{posterior}(\boldsymbol{\eta}) := \pi(\boldsymbol{\eta}|\mathbf{u}_{obs}) \propto \pi_{prior}(\boldsymbol{\eta})\pi_{likelihood}(\mathbf{u}_{obs}|\boldsymbol{\eta}). \tag{2.55}$$

This approach allows us to fuse simulations and measurements into the inversion framework. Unlike deterministic inversion, the expression (2.55) provides a probabilistic characterization of the solution [8] for the stochastic inverse problem. In this context, the likelihood function $\pi_{likehood}(\mathbf{u}_{obs}|\boldsymbol{\eta})$ is a conditional probability of the model outputs with given model parameters $\boldsymbol{\eta}$. Also, the prior probability density function (pdf) $\pi_{prior}(\boldsymbol{\eta})$ allows us to inject prior knowledge into the model. In our case, the prior density function π_{prior} is a multivariate Gaussian of the form:

$$\pi_{prior}(\boldsymbol{\eta}) \propto \exp(-\frac{1}{2}\|\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}\|^2_{\Gamma^{-1}_{prior}}). \tag{2.56}$$

The simplification above is possible due to the independence of the $\boldsymbol{\eta}$ vector. Specifically, the covariance matrix Γ_{prior} is an identity matrix and $\bar{\boldsymbol{\eta}}$ is a zero vector. The representation of likelihood function is core to the characterization of the posterior density function $\pi_{posterior}$. In the limiting case where the measurement and the model are exactly unbiased, the Bayesian model can easily be reduced to

$$\pi_{posterior}(\boldsymbol{\eta}) := \pi(\boldsymbol{\eta}|\mathbf{u}_{obs}) \propto \pi_{prior}(\boldsymbol{\eta}). \tag{2.57}$$

To further simplify the discussion, here we assume that the error between the measurement and the model is unbiased and additive, and the noise follows a Gaussian distribution. This leads to following expression for the likelihood function

$$\pi_{likelihood}(\mathbf{u}_{obs}|\boldsymbol{\eta}) \propto \exp(-\frac{1}{2}\|f(\boldsymbol{\eta}) - \mathbf{u}_{obs}\|^2_{\Gamma^{-1}_{noise}}). \tag{2.58}$$

We note that our procedure is still valid for other choices of likelihood functions. Our particular choice for likelihood is due to limited information on measurement and modeling errors. The choice of the likelihood function of the form Equation (2.58) leads to following log-likelihood function,

$$-\log(\pi(\mathbf{u}_{obs}|\boldsymbol{\eta})) = \frac{1}{2}\|f(\boldsymbol{\eta}) - \mathbf{u}_{obs}\|^2_{\Gamma^{-1}_{noise}}, \tag{2.59}$$

and the corresponding posterior density can be derived as

$$\pi_{posterior}(\boldsymbol{\eta}) \propto \exp(J(\boldsymbol{\eta})), \tag{2.60}$$

where $J(\boldsymbol{\eta})$ is given by

$$J(\boldsymbol{\eta}) := \frac{1}{2}\|f(\boldsymbol{\eta}) - \mathbf{u}_{obs}\|^2_{\Gamma^{-1}_{noise}} + \frac{1}{2}\|\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}\|^2_{\Gamma^{-1}_{prior}}. \tag{2.61}$$

Due to the non-linear relation between the parameters $\boldsymbol{\eta}$ and the measurements, direct sampling from the posterior is not possible even with the chosen likelihood function [8]. MCMC methods provide a systematic way to sample from the corresponding posteriors.

2.5 Gradient-based adjoint MCMC

The nonlinear mapping between the observables and parameters leads to non-Gaussian posteriors even with additive noise and a Gaussian prior assumption. MCMC methods are relevant techniques for sampling non-standard posteriors. They require many simulations of the forward models, however, leading to computational intractability when the forward models are expensive to evaluate. Here, we employ LMCMC to reduce the computational complexity, using gradient information computed in the feature space based on the adjoint PDE and automatic differentiation in the feature space. Theoretically, LMCMC has a computational complexity of $O(n^{1/3})$, while Metropolis Hastings MCMC (MHMCMC) based on random walk has the complexity of $O(n)$ where n is the dimension of the inference parameters. LMCMC considers the following overdamped Langevin-Ito diffusion process,

$$dX = \nabla \log \pi_{posterior}(X)dt + \sqrt{2}dW. \tag{2.62}$$

The probability distribution $\rho(t)$ of $X(t)$ approaches a stationary distribution, which is invariant under diffusion, and $\rho(t)$ approaches the true posterior ($\rho_\infty = \pi_{poster}$) asymptotically. Approximate sample paths of the Langevin diffusion can be generated by many discrete-time methods. Using a fixed time step $\tau > 0$, the above equation can be written as,

$$X_{k+1} = X_k + \tau\nabla \log \pi(X_k) + \sqrt{2\tau}\xi_k \tag{2.63}$$

where each ξ_k is an independent draw from a multivariate normal distribution on $\mathbb{R}^{N_F}$ with mean 0 and identity covariance matrix.

This proposal is accepted or rejected similar to the Metropolis-Hasting algorithm using α,

$$\alpha = \min\{1, \frac{\pi(X_{k+1})q(X_k|X_{k+1})}{\pi(X_k)q(X_{k+1}|X_k)}\} \tag{2.64}$$

where

$$q(x'|x) \propto \exp(-\frac{1}{4\tau}\|x' - x - \tau\nabla\log\pi(x)\|_2^2) \tag{2.65}$$

2.6 Adjoint Information of the posterior density function

In this section, we introduce a technique to compute the gradient information of the negative logarithm of the posterior function with respect to the random parameters $\boldsymbol{\eta}$,

$$\begin{aligned} J(\boldsymbol{\eta}) \quad &:= \tfrac{1}{2}\|f(\boldsymbol{\eta}) - \mathbf{u}_{obs}\|^2_{\Gamma^{-1}_{noise}} + \tfrac{1}{2}\|\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}\|^2_{\Gamma^{-1}_{prior}} && (2.66)\\ &= J_1(\boldsymbol{\eta}) + J_2(\boldsymbol{\eta}), && (2.67)\end{aligned}$$

where $J_1(\boldsymbol{\eta}) = \frac{1}{2}\|f(\boldsymbol{\eta}) - \mathbf{u}_{obs}\|^2_{\Gamma^{-1}_{noise}}$ and $J_2(\boldsymbol{\eta}) = \frac{1}{2}\|\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}\|^2_{\Gamma^{-1}_{prior}}$. It is nontrivial to obtain the functional derivative of $J(\boldsymbol{\eta})$. Here we use the adjoint model and automatic differentiation to compute the gradients. Using the mathematical derivations in the preceding sections, the relationship between the variables $\boldsymbol{\eta}, \boldsymbol{\xi}, \mathbf{y}, \mu, \lambda, \mathbf{u}$ can be summarized as,

$$\boldsymbol{\eta} \xrightarrow{\text{PCE}} \boldsymbol{\xi} \xrightarrow{\text{Pre-image}} \mathbf{y} \xrightarrow{\exp} \mu, \lambda \xrightarrow{\text{forward model}} \mathbf{u}. \tag{2.68}$$

The objective functional J can be expressed in terms of $\boldsymbol{\eta}$ by

$$\boldsymbol{\eta} \to \frac{1}{2}(f(\boldsymbol{\eta}) - \mathbf{u}_{obs}, \Gamma^{-1}_{noise}(f(\boldsymbol{\eta}) - \mathbf{u}_{obs})) + \frac{1}{2}(\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}, \Gamma^{-1}_{prior}(\boldsymbol{\eta} - \bar{\boldsymbol{\eta}})) \tag{2.69}$$

The second part of $J(\boldsymbol{\eta})$ is a quadratic form in the parameters $\boldsymbol{\eta}$. The expression for the gradient of $J_2(\boldsymbol{\eta})$ can directly be obtained as

$$\nabla_{\boldsymbol{\eta}} J_2(\boldsymbol{\eta}) = \Gamma^{-1}_{prior}(\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}) \tag{2.70}$$

To derive the gradient of J_1, we follow the procedure similar to Giering *et al.* [52]. Consider the Taylor expansion J_1 with respect to the control variables at a given point $\boldsymbol{\eta}_0$

$$J_1(\boldsymbol{\eta}) = J_1(\boldsymbol{\eta}_0) + (\nabla_{\boldsymbol{\eta}} J_1(\boldsymbol{\eta}_0), \boldsymbol{\eta} - \boldsymbol{\eta}_0) + O(|\boldsymbol{\eta} - \boldsymbol{\eta}_0|), \tag{2.71}$$

or in shorthand,

$$\delta J_1 = (\nabla_{\boldsymbol{\eta}} J_1(\boldsymbol{\eta}_0), \delta\boldsymbol{\eta}). \tag{2.72}$$

We use the shorthand notation whenever linear approximations are involved. Suppose J_1 is sufficiently regular, then for each parameter vector $\boldsymbol{\eta}_0$, and using symmetry property of the inner product and applying the product rule of differentiation yields

$$\delta J_1 = (\Gamma_{noise}^{-1}(f(\boldsymbol{\eta}) - \mathbf{u}_{obs}), \nabla_{\boldsymbol{\eta}} f(\boldsymbol{\eta}_0)\delta\boldsymbol{\eta}). \tag{2.73}$$

Using the definition of the adjoint operator we obtain

$$\delta J_1 = ((\nabla_{\boldsymbol{\eta}} f(\boldsymbol{\eta}_0))^T \Gamma_{noise}^{-1}(f(\boldsymbol{\eta}) - \mathbf{u}_{obs}), \delta\boldsymbol{\eta}). \tag{2.74}$$

Therefore, according to the definition of gradient, the gradient of the J_1 with respect to $\boldsymbol{\eta}$ is

$$\nabla_{\boldsymbol{\eta}} J_1(\boldsymbol{\eta}_0) = (\nabla_{\boldsymbol{\eta}} f(\boldsymbol{\eta}_0))^T \Gamma_{noise}^{-1}(f(\boldsymbol{\eta}) - \mathbf{u}_{obs}), \tag{2.75}$$

Since the function $f := f_1 \circ f_2 \circ f_3 \circ f_4$, applying the chain rule yields

$$\begin{aligned} f' &:= f_1' \circ f_2' \circ f_3' \circ f_4' & (2.76)\\ &= \nabla_{\lambda,\mu}\mathbf{u}\nabla_{\mathbf{y}}\lambda\nabla_{\boldsymbol{\xi}}\mathbf{y}\nabla_{\boldsymbol{\eta}}\boldsymbol{\xi}. & (2.77)\end{aligned}$$

The gradient information can be rewritten as

$$\nabla_{\boldsymbol{\eta}} J_1(\boldsymbol{\eta}_0) = (\nabla_{\boldsymbol{\eta}}\boldsymbol{\xi})^T(\nabla_{\boldsymbol{\xi}}\mathbf{y})^T(\nabla_{\mathbf{y}}\lambda)^T(\nabla_{\lambda,\mu}\mathbf{u})^T\Gamma_{noise}^{-1}(f(\boldsymbol{\eta}) - \mathbf{u}_{obs}), \tag{2.78}$$

The linear operator $\nabla_{\lambda,\mu}\mathbf{u}$ represents the tangent linear model of the forward problem and its adjoint operator is $(\nabla_{\lambda,\mu}\mathbf{u})^T$. Both operators depend on the point $\boldsymbol{\eta}_0$ at which the model is linearized. The linear operator $(\nabla_{\boldsymbol{\eta}}\boldsymbol{\xi})^T$ represents the adjoint model of the PCE, and $(\nabla_{\boldsymbol{\xi}}\mathbf{y})^T$ represents the adjoint model of the pre-image iteration mapping.

The adjoint model $(\nabla_{\lambda,\mu}\mathbf{u})^T$ can easily be obtained with the procedure detailed in §2.1. The PCE mapping in Equation (2.47) and the pre-image mapping methods are continuous smooth mappings. The adjoint models for these mappings are obtained with automatic differentiation [53].

2.7 Algorithms

In this section, we summarize the above derivations into two simple algorithms to facilitate the implementation of the proposed methodology.

Algorithm 1 Computation of posterior density function and gradients

```
    Read the snapshots $\{\boldsymbol{y}_l\}_{l=1}^{M}$ of the parameters $\mu, \lambda$
    Compute KPCA reduced model using Equation (2.40)
    Parameterize the random variables $\boldsymbol{\xi}$ with PCE using Equation (2.47)
    Compute prior density function $\pi_{prior}$ as defined by Equation (2.56)
    Compute likelihood function $\pi_{likelihood}$ as defined by Equation (2.58)
     Compute the posterior density function using Equation (2.55)
    Compute the gradient of the cost functional with respect to parameters $\lambda$ and $\mu$ using adjoint model
    Compute the gradient of the cost functional in the feature space using automatic differentiation
```

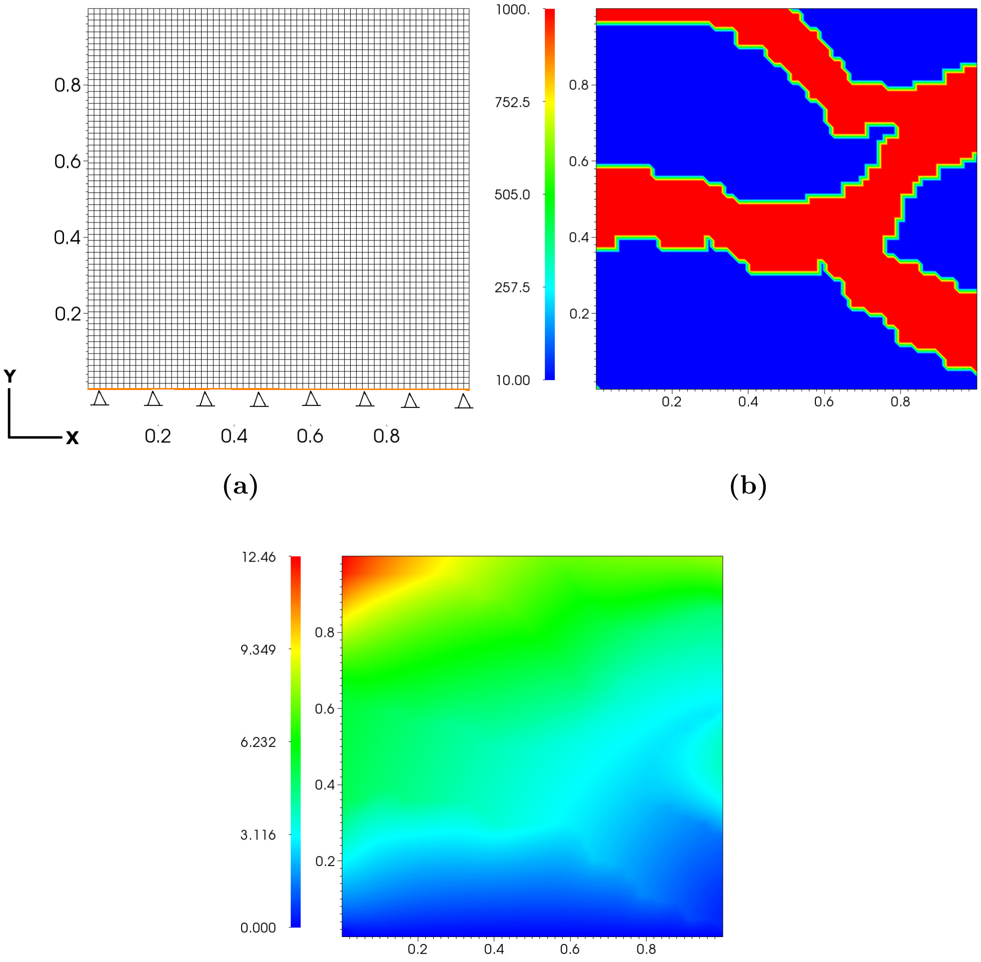
Algorithm 2 Posterior sampling using Langevin MCMC framework

```
    Choose initial parameters $\boldsymbol{\eta}_0$
    Compute $\pi_{posterior}(\boldsymbol{\eta}_0)$ using Algorithm 1
    for $l$=1 to N do
        Draw sample $y$ from the proposal density function
        Compute $\pi_{posterior}(y)$ using algorithm 1
        Compute $\alpha(\boldsymbol{\eta}_l, y) = \min\{1, \frac{\pi_{posterior}(y)q(y|\boldsymbol{\eta}_l)}{\pi_{posterior}(\boldsymbol{\eta}_l)q(\boldsymbol{\eta}_l|y)}\}$, where $q(y|\boldsymbol{\eta}_l)$ and $q(\boldsymbol{\eta}_l|y)$ are computed using Equation 2.65
        Draw $u \sim U([0, 1])$
        if  $u < \alpha(\boldsymbol{\eta}_l, y)$ then
            Accept : Set $\boldsymbol{\eta}_{l+1} = y$
        else
            Reject : Set $\boldsymbol{\eta}_{l+1} = \boldsymbol{\eta}_k$
        end if
    end for
```

Chapter 3

Stochastic Elasticity Inversion

In this section, we demonstrate the computational efficiency of the proposed method for the stochastic inversion of a 2D-linear elasticity model through a numerical example. The objective is to recover elasticity parameters of a geologically-complex rock characterized by sinuous channels of one material embedded in another. Figure 3.1 (a) shows the mesh and boundary conditions of the numerical example. The bottom boundary is supported by a pinned connection to curtail vertical and horizontal motion and other boundaries are free to expand. The square shaped domain is allowed to deform under self-weight due to gravity. Measurements of the displacements are assumed to be available at the top, left and right boundaries. For the sake of simplicity, we assume the Poisson ratio of rock is fixed at a typical value of $\nu = 0.25$. This implies $\lambda = \mu$, and therefore we need only invert for one elastic parameter field instead of two. Figure 3.1 (b) depicts a discrete realization λ_1. Blue and red color domains here correspond to two distinct rock types with considerable differences in their elastic properties. Homogeneous elasticity models tend to over simplify the system and can lead to sub-optimal solutions. Figure 3.1 (c) shows a contour plot of the displacement magnitude with elasticity parameters λ_1. A forward and an adjoint simulations are performed in any LMCMC sampling step to compute the gradient of the cost functional with respect to the model parameters. Figure 3.2 demonstrates an example of the gradient computation, here, Fig. 3.2 (a) shows a realization of the elasticity parameter λ_2, used to evaluate the adjoint solution based on the measurements obtained with parameters λ_1. Figure 3.2 (b) depicts forward displacement magnitude of the model with parameters λ_2 due to self weight and Fig 3.2 (c) shows the corresponding adjoint displacement magnitude contour computed with the adjoint PDE. Figure 3.2 (d) shows the gradient of the cost function with respect to λ_2 evaluated with self-adjoint PDE formulation.

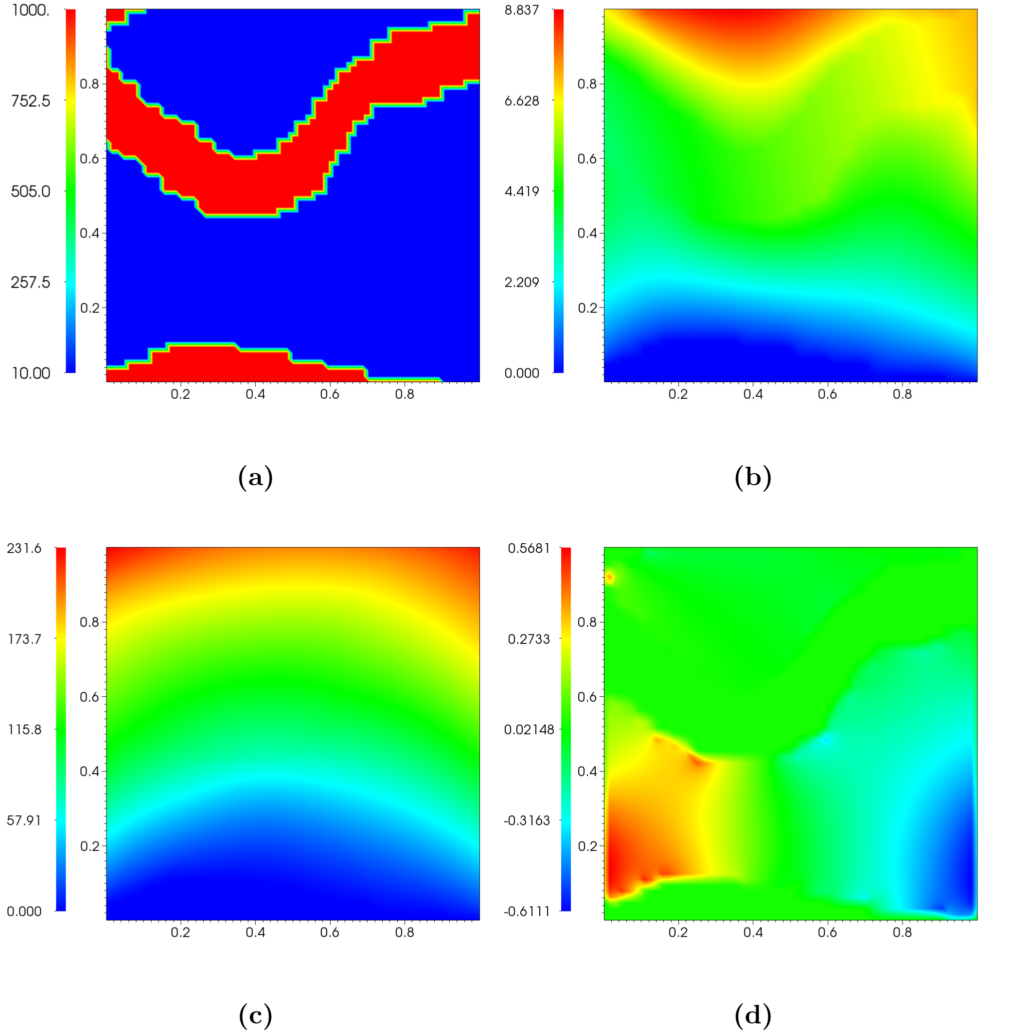


(a)

(b)

(c)

Fig. 3.1. a) Physical setup of the numerical example used for the demonstration b) a realization λ_1 of the elasticity parameters c) corresponding displacement magnitude due to self weight.



(a) **(b)**

(c) **(d)**

Fig. 3.2. (a) A realization of the elasticity parameter λ_2 (b) forward displacement magnitude due to self weight (c) adjoint displacement magnitude (d) gradient of the cost function with respect to λ_2 based on the measurements obtained with elasticity parameters λ_1

3.1 Snapshot generation

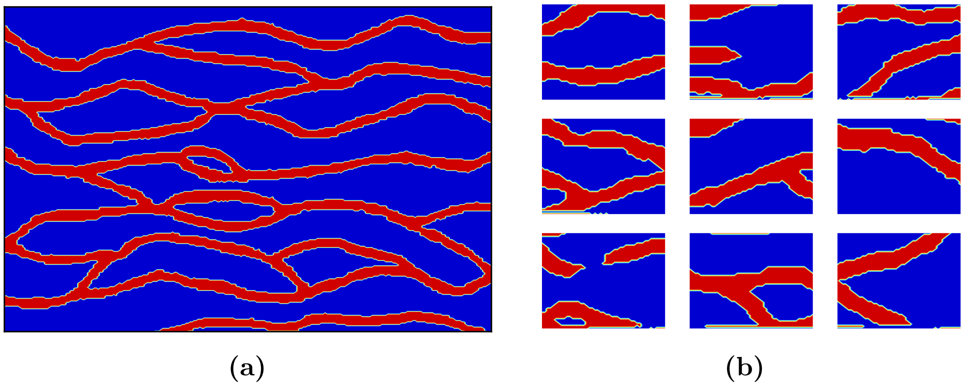


Fig. 3.3. (a) Training image and (b) a few snapshots generated with the SNESIM algorithm

For natural materials like rock, elasticity parameters often exhibit multi-scale spatial fluctuations due to inherent heterogeneity [54]. In our numerical experiments, we rely on the single normal equation simulation (SNESIM) algorithm [28] based on a training image, as shown in Fig. 3.3 (a), similar to Ma et al. and Sarma et al. [27, 26]. We generate 1000 realizations of a "channelized" rock. Figure 3.3 (right) depicts a few snapshots generated using the SNESIM algorithm. Here, λ for the channel material (red) and host material (blue) are assumed to be 10 and 1000 MPa, respectively. In order to guarantee positive values for the elasticity parameters, the inversion procedure is carried on $\ln(\lambda)$.

3.2 Efficiency of the kernel PCA and the pre-image

In contrast to linear PCA, KPCA is performed in the feature space instead of the original space. For the polynomial kernel $(\mathbf{x} \cdot \mathbf{y})^d$, an input space of realization in $\mathbb{R}^{N_R}$ is mapped to a feature space of dimension N_F given by (2.31). Compared to the dimension of the original space $\mathbb{R}^{N_R}$, N_F is very large with higher order polynomial kernels. For instance, in our channelized model, we have $N_R = 10^3$ and for $d = 5$ this leads to $N_F \approx 10^{15}$, a very high-dimensional space which allows kernel PCA to explore and capture distinctive properties of the nonlinear data. Note here that the KPCA-feature space is still obtained by a low-dimensional eigendecomposition similar to PCA with the kernel trick.

Since our interest is to find inverse solutions in the original space, an additional pre-imaging step is required to transform the feature snapshots back into the original snapshots. Unlike linear PCA, the solution to the pre-imaging is not unique and also suffers from instability. In order to choose the best kernel for our procedure, we test Gaussian, linear, quadratic, cubic, 4th and 5th order polynomial kernels for their pre-imaging efficiency using a few selected snapshots. Figure 3.5 depicts the results from this procedure for a pre-selected snapshot. It shows that higher order (d) polynomial kernels lead to more efficient mapping. Also, we observed the computation of the pre-image became unstable for polynomial kernels order greater than five. Figure 3.6 shows the eigenvalue decay of the covariance matrix for Gaussian and polynomial kernels, showing that linear PCA and KPCA have similar eigen spectrums. Figure 3.4 displays a few snapshots generated using mean perturbation in KPCA space with Gaussian, linear, quadratic, cubic, fourth, and fifth order kernels. This demonstrates that as the order of the polynomial kernel increases, the mean perturbed data looks more like a channelized structure—i.e., higher order kernels are able to represent data more effectively. Based on Figs. 3.5, 3.6 and 3.4, we select a polynomial kernel with order 5 and dimension 20 (about 75% contribution).

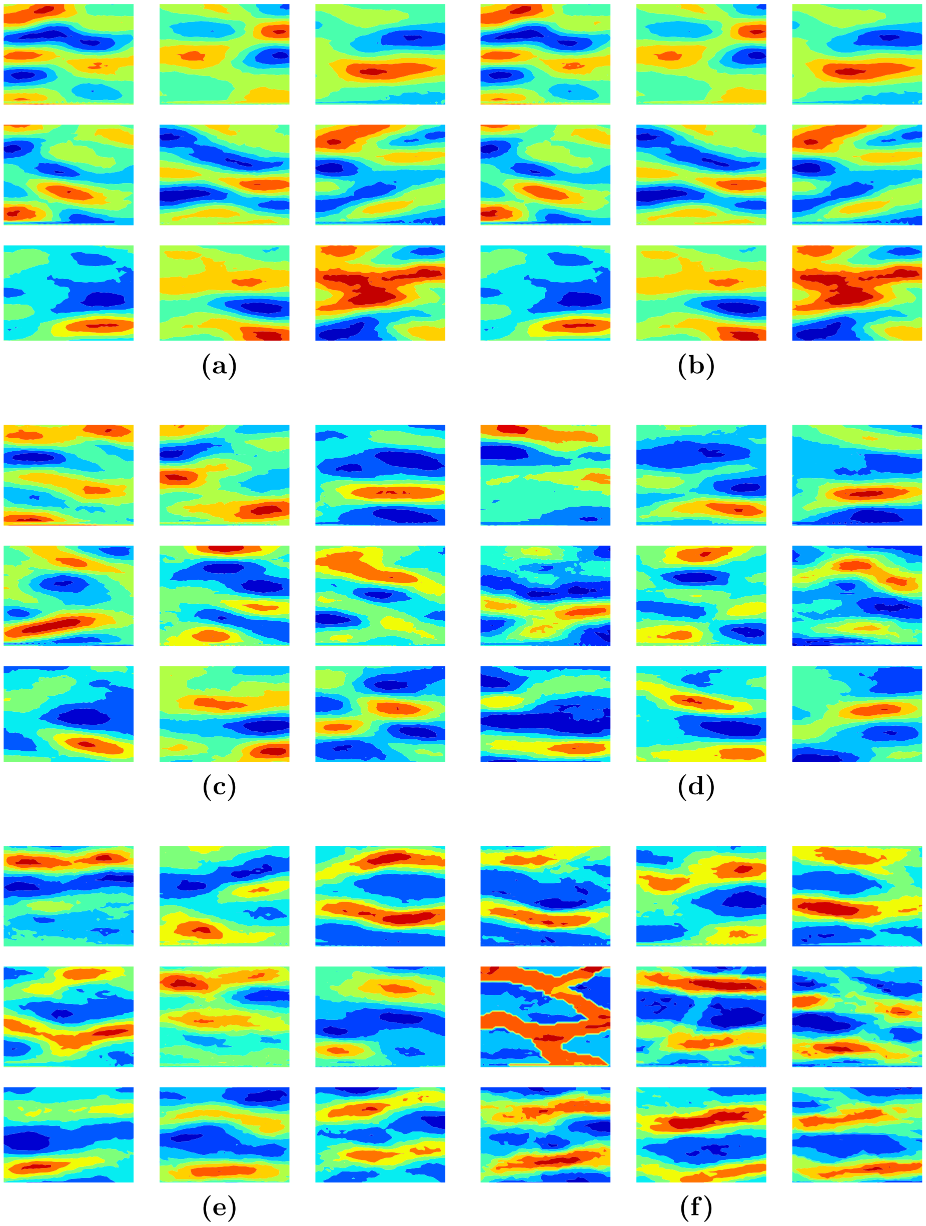


Fig. 3.4. A few snapshots generated using mean perturbation in KPCA space with a)Gaussian b)linear c)quadratic d)cubic e)fourth order and f)fifth order kernels

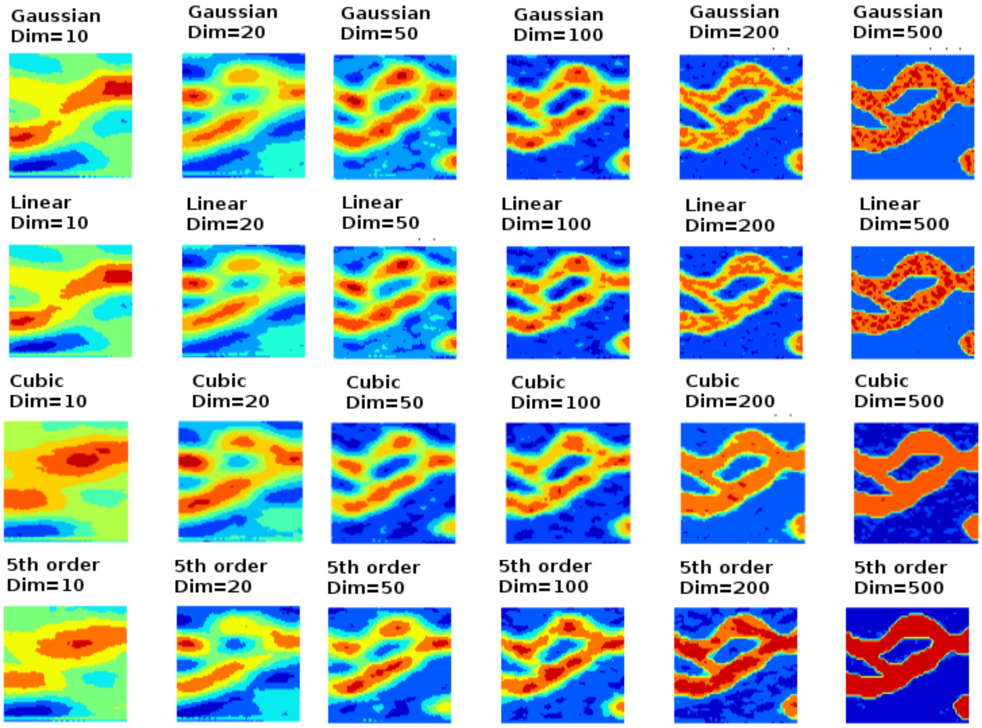


Fig. 3.5. KPCA with Gaussian, linear, cubic and fifth order kernels in 10, 20, 50, 100, 200 and 500 dimensions

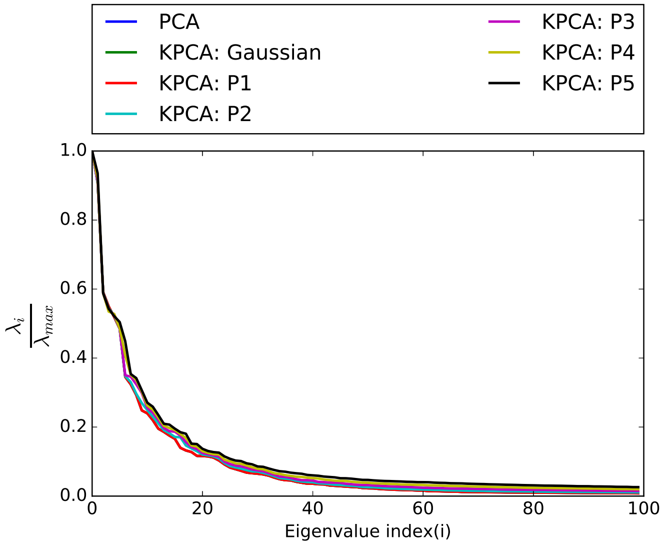


Fig. 3.6. Eigenvalue decay of the snapshots for different kernels

3.3 Efficiency of the PCE

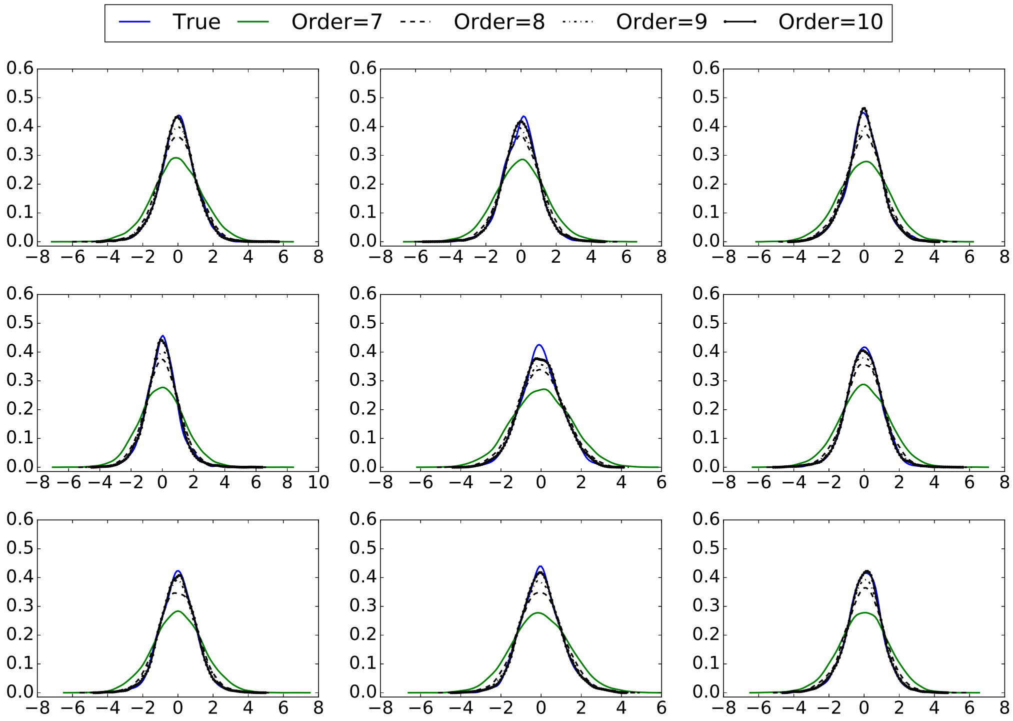


Fig. 3.7. Probability density function of a few ξ^d obtained using true samples and from the samples of PCE with different orders

Nonlinear mapping of the parameter space $\Phi : \mathbb{R}^{N_R} \rightarrow \mathbb{R}^{N_F}$, $N_F \gg N_R$ and solving (2.46) lead to 1000 discrete realizations of the $\boldsymbol{\xi}^d$. In general, $\boldsymbol{\xi}^d$ are non-Gaussian, uncorrelated and dependent random variables. To generate these realizations in a computationally efficient way during the inversion procedure, assuming $\boldsymbol{\xi}^d$ are independent similar to [44, 45], we construct multiple PCEs for $\boldsymbol{\xi}^d$ using ICDF mapping. Figure 3.7 depicts the probability density functions of a few selected ξ^d constructed from the 1000 discrete realizations (true) and also samples obtained from the PCE with different orders. This figure demonstrates that, as the order of the PCE increases, PCE is able to capture the true distribution of the ξ^d. Based on this plot, the PCE with order 10 is used to map ξ^d to the standard Gaussian variable η.

3.3.1 Numerical test for the gradient

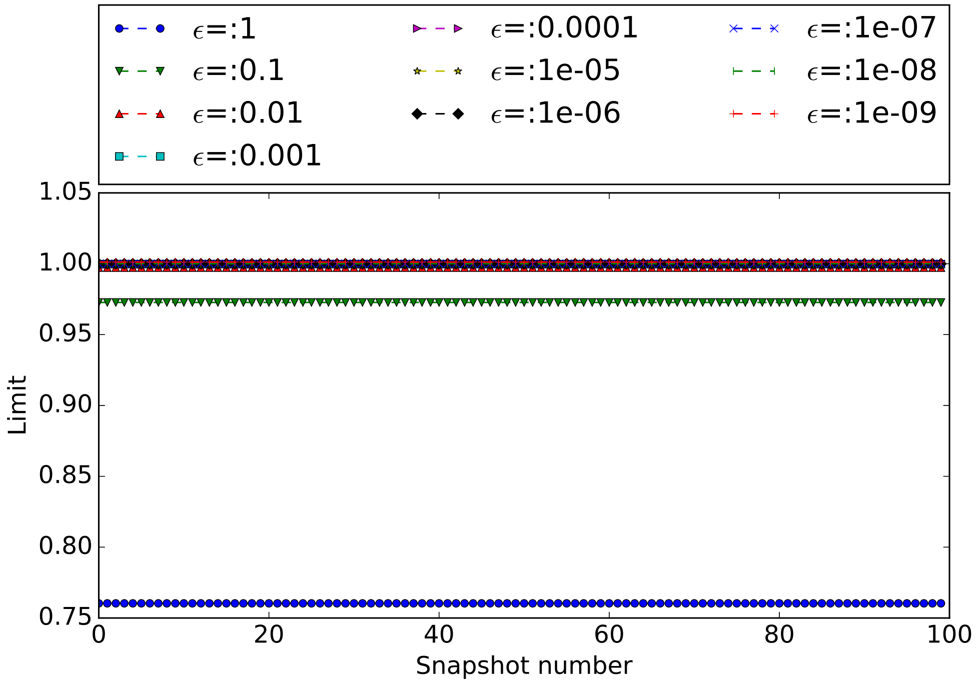


Fig. 3.8. Gradient limit for different choices of ϵ for 100 snapshots.

The gradients in the feature space are computed using adjoint PDE and TAPENADE [53], an automatic differentiation toolkit. In order to test the accuracy of computed gradients, we make use of the properties Gateaux differential $d_h f$. A Gateaux differential is defined as,

$$d_h J = \lim_{\epsilon \to 0} \frac{J(\eta + \epsilon h) - J(\eta)}{\epsilon}. \tag{3.1}$$

A property of the Gateaux derivative is that, if $h = \frac{\nabla_\eta J}{||\nabla_\eta J||^2}$, then $\epsilon \to 0$, $d_h J \to 1$. We use this property to test the accuracy of the derivative of the cost functional with respect to η. Fig. 3.8 shows the gradient limit for different choices of ϵ for 100 snapshots. This figure shows that for sufficiently small ϵ (> 0.1) the limit in Equation 3.1 goes to 1, thus the accuracy of the gradient computation is verified.

3.4 Stochastic inversion using MHMCMC and Langevin MCMC

The goal of our numerical demonstration is to recover the elastic parameters of the complex geological elasticity parameter field shown in Fig. 3.9 (a). The "ground truth" observations of displacements at the top, left and right boundary grid points are synthesized by running a forward simulator with aforementioned elasticity parameters. Due to sparsity of the measurements and low-dimensionality of the feature space we foresee that the posterior solution will converge to the lower dimensional version (Fig. 3.9 (b)) of the original snapshot.

The samples of the posterior distribution are obtained with LMCMC and random walk MHMCMC algorithms. Since the posterior exploration is carried out in $\boldsymbol{\eta}$ space, a multi-dimensional standard normal distribution servers as a prior distribution. For MHMCMC, the proposal or sampling distribution is assumed to be Gaussian centered at current accepted sample with standard deviation of 0.1. The Langevin parameter τ is chosen as 0.08 based on trail and error and likelihood is scaled by 1000 to avoid floating point underflow errors. Figures 3.9 (c) and (d) show the posterior mean and standard deviation snapshots obtained using MHMCMC. Similarly, Fig. 3.9 (e) and (f) show the posterior mean and standard deviation snapshots obtained using LMCMC. As envisioned before both MCMC and LHMCMC are able to recover the low-dimensional version of the original parameter field. Figure 3.10 depicts the posterior distribution of the η for the random walk MHMCMC and LMCMC. The detailed analysis of the posterior distribution is carried out in the next section. Three MCMC chains with initial guess for η as -2, 0 and 2 are used to check the global convergence of the MCMC algorithms. Figure 3.11 shows the convergence of the MCMC chains for random walk MHMCMC and LMCMC. Chains start converging around the 100th and 500th sample for LMCMC and MHMCMC, respectively, i.e., gradient information assisted in substantially faster convergence.

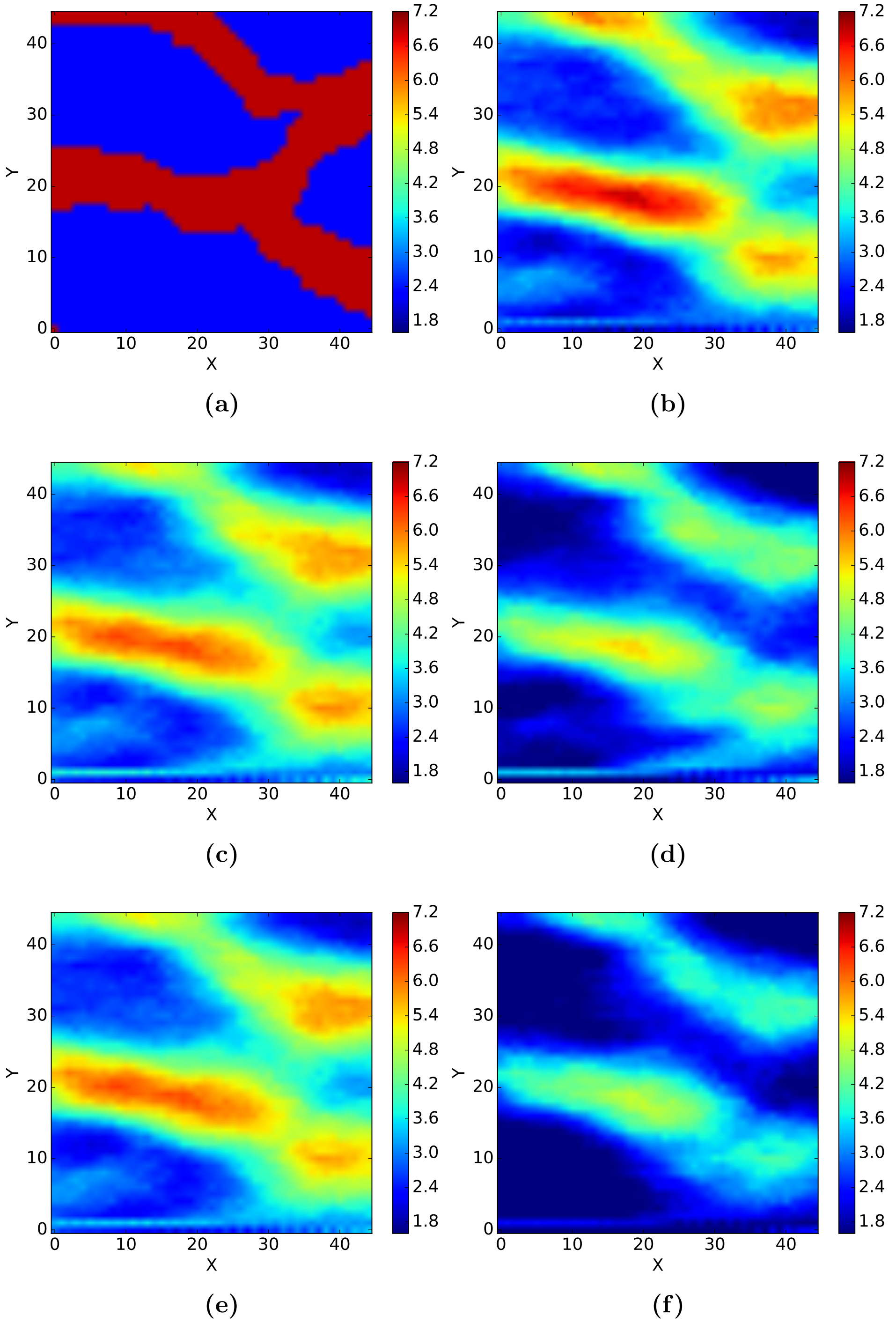
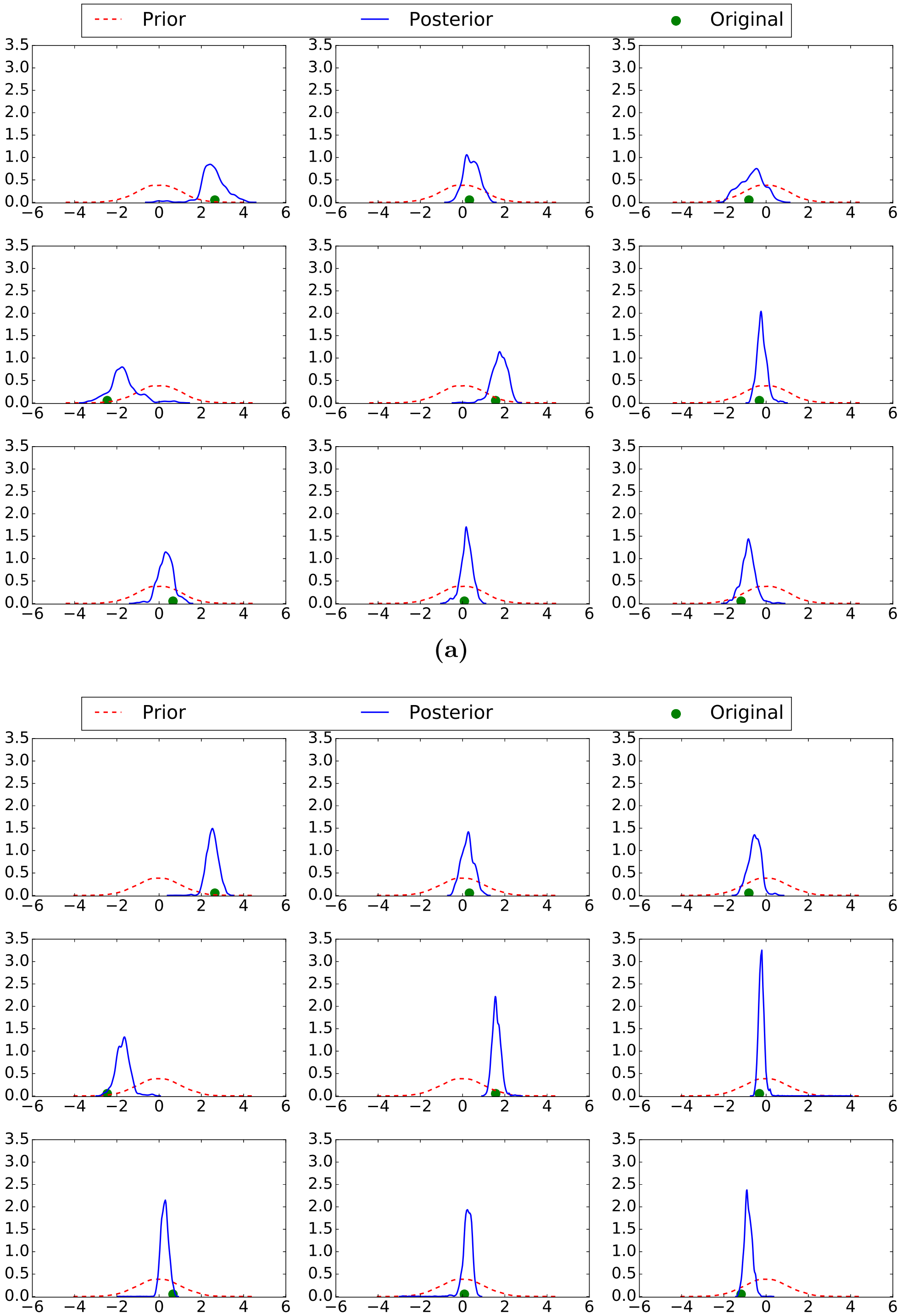


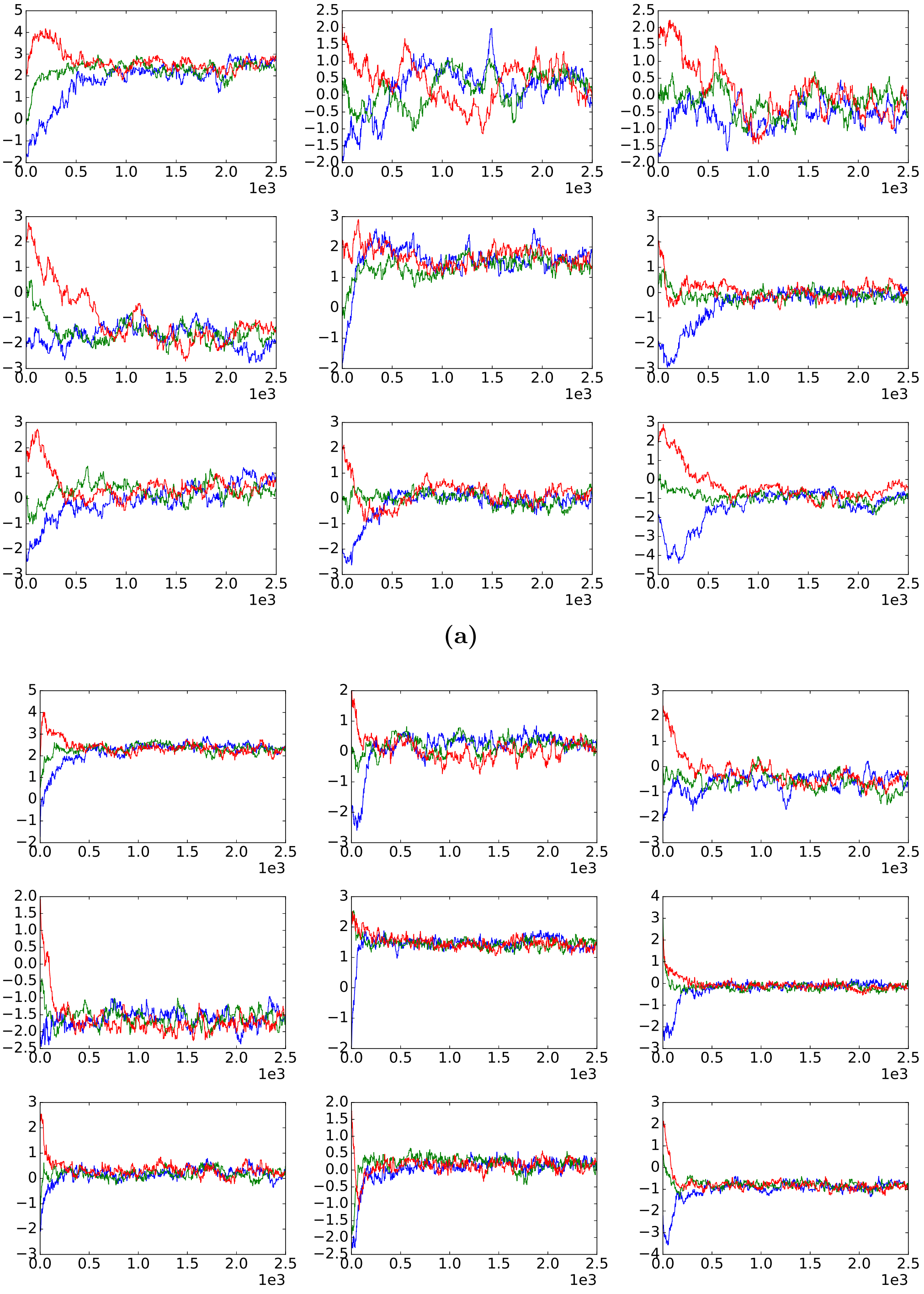
Fig. 3.9. a) Original b) KPCA projected c) MHMCMC posterior mean d) MHMCMC posterior standard deviation e) Langevin MCMC posterior mean and f) Langevin MCMC posterior standard deviation snapshots



(a)

(b)

Fig. 3.10. Prior and posterior probability density functions for a few η's with original value for a) MHMCMC b) Langevin MCMC



(a)

(b)

Fig. 3.11. Posterior MCMC chains for a few η's with staring at -2 (red), 0 (green) and 2(blue) for a)MCMC b)Langevin MCMC

3.5 Discussion

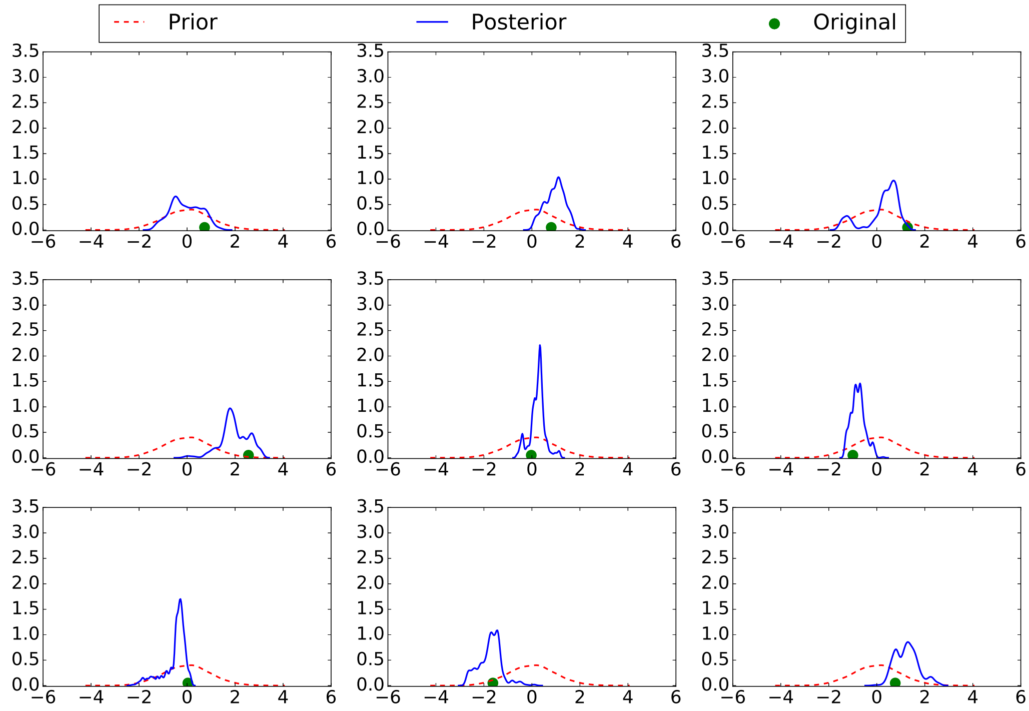


Fig. 3.12. Prior and posterior probability density functions for a few η's with original value for obtained using PCA-based Langevin MCMC

As shown in Figure 3.6, dimension reduction via linear PCA and KPCA generally gives similar reduced orders based on existing data points, but the reduced-order space they represent can be very different. Since the proposed method is based on LMCMC, which has computational complexity of $O(n^{1/3})$ compared to MHMCMC complexity of $O(n)$, its computational cost scales better. To see the effect of KPCA on posterior sampling, we run a PCA-based LMCMC. The KPCA-based LMCMC and PCA-based LMCMC have 33.66% and 10.10% acceptance rate, respectively.

As to why KPCA is more efficient than PCA, we propose the following explanation: the posterior probability density functions (PDFs) inverted by PCA-based MCMC (as seen in Figure 3.12) are generally non-Gaussian and possibly multi-modal. In contrast, those inverted using KPCA-based MCMC have near-normal distributions and are generally unimodal as a result of the embedded nonlinear mapping from the feature space to the parameter space. Since it is generally more expensive (requires more iterations) to achieve convergence for non-standard PDFs with many peaks, the gradient-based MCMC, which approximates the posteriors by a local Gaussian, is expected to be more efficient. A second reason is that, compared to linear PCA, the embedded manifold identified by the data-driven KPCA contains a more 'concentrated' distribution of the underlying parameters that need to be inverted. Even though finding an optimal point (deterministic inversion) in the detected manifold may not be very distinguishable from stochastic inversion, the latter (stochastic inversion) performed in such a clustered manifold will be critical for achieving high performance and accuracy. Specifically, the neighborhood identified by linear PCA for any given channelized material parameter point may contain very few channelized structures, which can cause great difficulties for a high-dimensional random field inversions especially when considering stochastic inversions. Hence, the KPCA-based MCMC will demonstrate improved efficiency even without gradient information, thus making it useful even for the applications where the adjoint model cannot be derived easily.

As pointed out before, a relevant feature space identification for the problem considered here is analogous to a typical binary classification (channel vs no-channel) problem encountered in machine learning community. The discriminant function or boundary between two classes is linear in PCA—i.e., PCA detects a linear manifold in the original space. The KPCA or other kernel based methods such as diffusion maps, transforms data to a non-linear space with the kernel trick and detect a linear manifold in that space. Since the discriminant function deduced here is a linear function in terms of the weights, they detect a linear manifold in the non-linear space. In the future work, we will pursue feature space identification in the kernel space with non-linear or 'curved' manifold learning using so-called deep autoencoders. Note here that KPCA and PCA can be described with autoencoder with a particular choice of activation function and decoding part of the deep network allows us to construct pre-imaging with a simple matrix-vector multiplication.

3.5.1 Numerical test for the gradient

We have presented an efficient stochastic inversion method in the framework of Bayesian inference based on an adjoint model, automatic differentiation, and Kernel PCA. The complexity of the MCMC is reduced through control reduction and efficient gradient computation. We demonstrate a practical way to characterize a full pdf assigned to each grid point of a discretized parametric random field based on prior knowledge or estimation of the random field and observational information of measurements data. To ensure the efficiency of the stochastic inversion, the control reduction is obtained by performing Bayesian inference in a low-dimensional feature space captured via KPCA. Different kernels such as Gaussian, first, second, third, fourth and fifth-order polynomials were tested and the kernel of KPCA is chosen based on snapshots obtained from the pre-imaging and mean perturbation. A PCE is devised for economic sampling from the feature space. The proposed method uses a high-fidelity forward model and thus can avoid sub-optimal solutions computed using surrogate-based methods. A gradient based LMCMC method is adopted for posterior sampling using cheaply computed gradients with an adjoint model and automatic differentiation. The efficiency of the proposed method is demonstrated through a synthetic numerical example with the objective of recovering the subsurface elastic parameters of the complex geological channelized field. Gradient-free MCMC and LMCMC were able to sample from the true posterior after 500 and 100 forward model runs, respectively. The KPCA-based MCMC results show a higher acceptance rate compared to the PCA-based MCMC, since the neighborhood identified by KPCA for any given channelized material parameter point contains more channelized structures. The method proposed has a generic nature and it can be adapted to other types of physics. For example, in future work we will consider the application of the proposed framework to a large-scale seismic inversion problem. It should be pointed out that the KPCA is a linear manifold statistical learning on the kernel space constructed by the nonlinear transformation from the original space. In future work, we will pursue feature reduction for optimal control and stochastic inversion with a broader choice of unsupervised learning approaches—e.g. non-linear manifold statistical learning techniques such as diffusion maps and deep-learning based autoencoders.

Chapter 4

Power Grid Stochastic State and Parameter Estimation

4.1 Grid data testing and analysis

Background An electric power grid is an interconnected network for delivering electricity from generators to loads by transmission and distribution systems. It is also a network overlaid with sensing and measurement, communication, and monitoring and control components that maintain grid reliability, security, and efficiency. Today's grid is evolving into the smart grid to provide more reliable, more efficient, and more sustainable electricity to customers [55]. To achieve those attributes, a variety of smart grid technologies are demanded.

In sensor and measurement fields, the synchrophasor is one of the most important smart grid technologies [55, 56, 57, 58, 59, 60]. A synchrophasor system primarily consists of phasor measurement units (PMUs), phasor data concentrators (PDCs), and communication networks as shown in Fig. 1. It typically uses PMUs to produce synchrophasor measurements from current and voltage signals (e.g., the ones from current and voltage transducers) and a standard time signal (e.g., the one from a global positioning system (GPS)), and then utilizes PDCs to transfer synchrophasor data from PMUs/PDCs to a control center and/or various applications [56, 57, 58, 55]. In the past decade, an increasing number of synchrophasor systems have been installed around the world and a series of synchrophasor applications have been implemented in grids [59, 60]. The synchrophasor is expected to perform high-precision low-latency and time-synchronized measurement and provide significant insight into grid planning and operation.

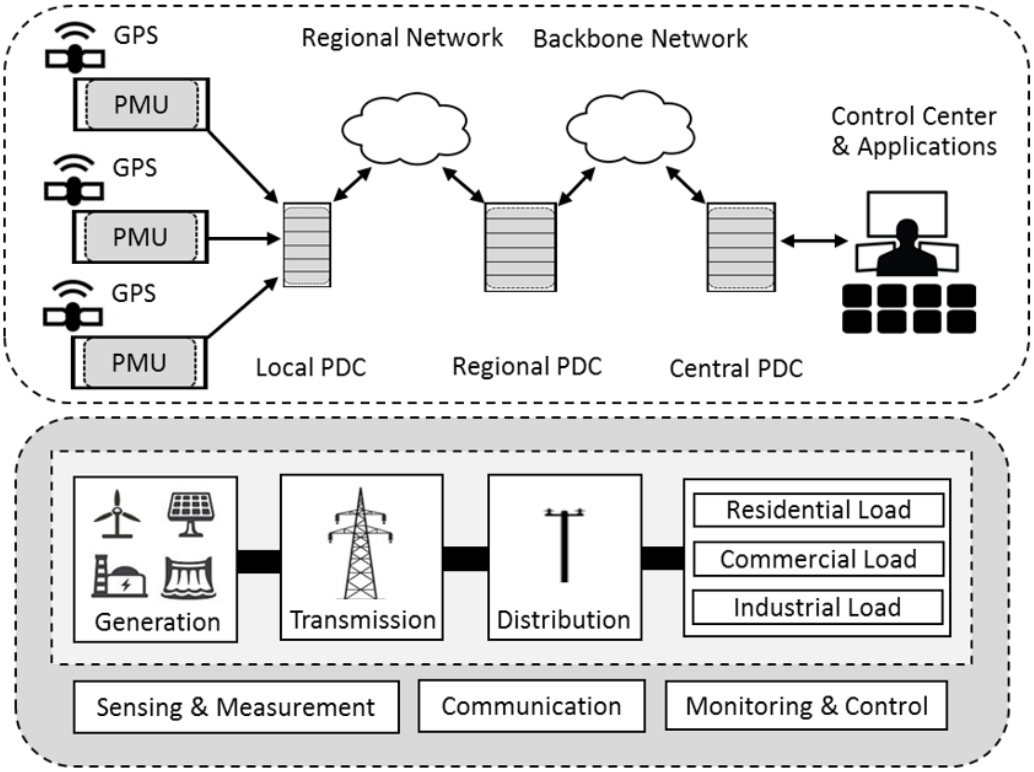


Fig. 4.1. A synchrophaor system over an electric power grid.

In practice, the synchrophasor inevitable involves measurement errors, which may affect or even disable certain synchrophasor applications [59, 60, 61, 62, 63]. It is a demanding task to analyze and model the synchrophasor measurement error. Traditionally, the synchrophasor is designed for transmission systems and the synchrophasor measurement error is assumed as a Gaussian noise in most synchrophasor applications. Several studies point out that this assumption is violated in reality, and the results are misleading or even damaging to certain applications (e.g., PMU-based state estimation) [61, 62, 63]. For example, Wang *et al.* analyze real PMU measurements and reveal that the PMU measurement errors do not follow a Gaussian distribution [61]; and Mili *et al.* assess the sensitivity of different state estimators to Gaussian/non-Gaussian noises and develop a robust state estimation method to cope with non-Gaussian measurement errors [62].

Now the synchrophasor is being deployed in distribution systems, such as the deployment of micro-PMUs (μPMUs) and FNET/GridEye [64, 65, 66, 67]. Compared with transmission systems, distribution systems are invested with fewer sensors and measurements, and distribution networks are often plagued by large measurement uncertainties due to their highly distributed and diverse infrastructure. At present we pay increasingly more attention to distribution systems, especially with the rapid development of distributed energy resources (DER) and distribution management systems (DMS). However, we have very limited knowledge about the nature of distribution system measurements and measurement errors. Accordingly, NASPI established Distribution Task Team (NASPI DisTT) to promote the distribution-level synchrophasor development. In NASPI DisTT 2017 winter report, one of the most urgent needs is to investigate the nature of the synchrophasor measurement error in real distribution systems [64].

This work analyzes the distribution-level synchrophasor measurement error with online and offline tests, and adopts graphical and numerical methods to mathematically and systematically identify the actual distribution of the measurement error. To the best knowledge of the authors, this is the first work to perform this kind of studies.

Definition In this work, the measurement error refers to the difference between the measured value and the true value of a selected quantity. It mainly consists of two components: a systematic error and a random error, which are often represented by a consistent bias and a random noise, respectively. In theory, the random error plays a decisive role in the distribution of the measurement error, which is typically introduced by the unknown and unpredictable changes occurring in measurement devices (e.g., the electronic noise and circuit aging) and/or in the environment (e.g., the wind, temperature, and communication). To investigate the distribution of the distribution-level synchrophasor measurement error thoroughly, tests are performed in various measurement devices including PMU, μPMU, and FDR, and in different environments, covering both primary and secondary distribution networks. The testing and analysis methods are explained below.

Testing It is difficult to directly obtain the real-time measurement error between the measurement value and the true value. Here, the distribution of the measurement error is identified in an indirect way via multiple synchronized measurements (MSMs), where multiple identical and independent synchrophasor measurement devices are deployed to simultaneously and independently meter the quantity x with the measurement ($z_i = x+e_i$, $z_j = x+e_j$, i and $j \in N$, N is the number of MSMs) and the measurement error (e_i, e_j). Subsequently, the Gaussian/non-Gaussian distribution of the measurement error can be determined by constructing the probabilistic distribution of the difference of the MSM errors or MSMs, i.e. $\Delta e = e_i - e_j$, $\Delta z = z_i - z_j$, and $\Delta e = \Delta z$ [68]. The general principle is that: in the event that A and B are independent random variables, if A and B are both normally distributed, then A ± B is normally distributed (**Proposition**), and if A ± B is not normally distributed, then either A or B (or both A and B) is not normally distributed (**Contraposition**). In other words, if $\Delta e = e_i - e_j$ is non-Gaussian, then e_i or/and e_j is non-Gaussian. Note that the measurements are synchronously taken from the identical devices in the same environment, where the measurement errors are supposed to follow the same distribution. Thus, it can be argued that if Δe is non-Gaussian, then both e_i and e_j are non-Gaussian.

In addition to the online test above (i.e. field experiment), an offline test is performed on the distribution monitoring platform FNET/GridEye (i.e. laboratory experiment), where a distribution signal is generated by a power system simulator and measured by a high-precision FDR, and the resulting measurement error is calculated by a calibrator straightforwardly [67].

Analysis There are two common ways to check normality/Gaussianity: graphical methods and numerical methods [69]. Here, the measurement $z(t, \omega)$, $t \in T, \omega \in \Omega$ is viewed as a stochastic process defined on the probability space $(\Omega, \mathcal{F}, P)$ over time T, where Ω and $\mathcal{F}$ are the sample and event spaces and P is the function that maps the events to the provabilities.

The graphical method describes the deviation from Gaussianity via the plot of the standard Gaussian distribution and the scaled error $\Delta e_s(\omega)$ defined as

$$\Delta e_s(\omega) = \frac{\Delta e(\omega) - \mu_{\Delta e}}{\sigma_{\Delta e}} \tag{4.1}$$

where $\mu_{\Delta e}$ and $\sigma_{\Delta e}$ are the mean and the standard deviation of the error $\Delta e(\omega)$, respectively.

The numerical method is implemented using Shapiro-Wilk (SW) and

Kolmogorov-Smirnov (KS) tests, which reject or accept the null hypothesis of Gaussianity with test and % confidence index α [69].

Results In power engineering, distribution systems are monitored by various measurement devices with multiple time-scales, like μPMU with reporting rate 120fps, PMU with reporting rate 10/30/60fps, and supervisory control and data acquisition (SCADA) updating a sample in every few seconds. In statistics, a statistical test has little power when the sample size is extremely small or large (e.g., < 30 or > 6000) [**?**, 70]. Thus, in the tests described below, the distributions under different time windows and sample sizes are considered collectively.

In Test 1, a voltage phasor in a primary distribution system is measured by two μPMUs, and in Test 2, a voltage phasor in a secondary distribution system is metered with two PMUs, manufactured by the same vendor. Figs. 4.2(a), 4.2(b), 4.3(a), and 4.3(b) depict the probability density function (pdf) of the scaled error Δe_s along with the standard Gaussian distribution within different time windows, and show that the distributions in various cases are all non-Gaussian with multiple peaks and non-symmetric nature. Figs. 2(c), 2(d), 3(c), and 3(d) describe the varying time window versus skewness (third-order moment) and kurtosis (fourth-order moment) of Δe, and show that skewness deviates from zero and kurtosis departs from Gaussian-kurtosis, indicating the non-symmetric and long/short-tail nature of the pdf of Δe even for the large time windows. Moreover, Table I and II give the percentage of non-Gaussian random variables (rvs) out of total random samples taken with different time windows and different sample sizes (note that SW test is generally more powerful than KS test and the power of both KS and SW tests is low for small samples, e.g, sample size < 60 [69, 70]). It is observed that most rvs are non-Gaussian based on SW and KS tests and their joint distribution is non-Gaussian. Specifically, the skewness and kurtosis in Figs. 3(c) and 3(d) are much higher than the ones in Figs. 2(c) and 2(d), partially because the measurement error in a secondary distribution system is more vulnerable to environmental changes than in a primary distribution system.

In Test 3, the laboratory experiment is carried out on the FNET platform using the FDR with reporting rate 10fps. It is found from Fig. 4.4 and Table III that even though there are few environmental fluctuations in the laboratory environment, the observed measurement errors still follow a non-Gaussian distribution. Through the online and offline tests and the graphical

Table 4.1: Summary of the Gaussianity Tests on μPMU Data

Measurement	Time window (s)	Sample size	% of non-Gaussian distributions			
			Shapiro-Wilk		Kolmogorov-Smirnov	
			α=5%	α=10%	α=5%	α=10%
Voltage angle	1	120	89.5	93.5	40	49.5
	5	600	99	99	75	81.5
	10	1200	100	100	76.5	84
	30	3600	99.5	100	81	85
Voltage magnitude	1	120	74.5	82	27	37.5
	5	600	98.5	98.5	65.5	72
	10	1200	99.5	99.5	78	84
	30	3600	100	100	95.5	97.5

and numerical analysis, extensive results reveal that the real measurement error potentially follows a non-Gaussian distribution.

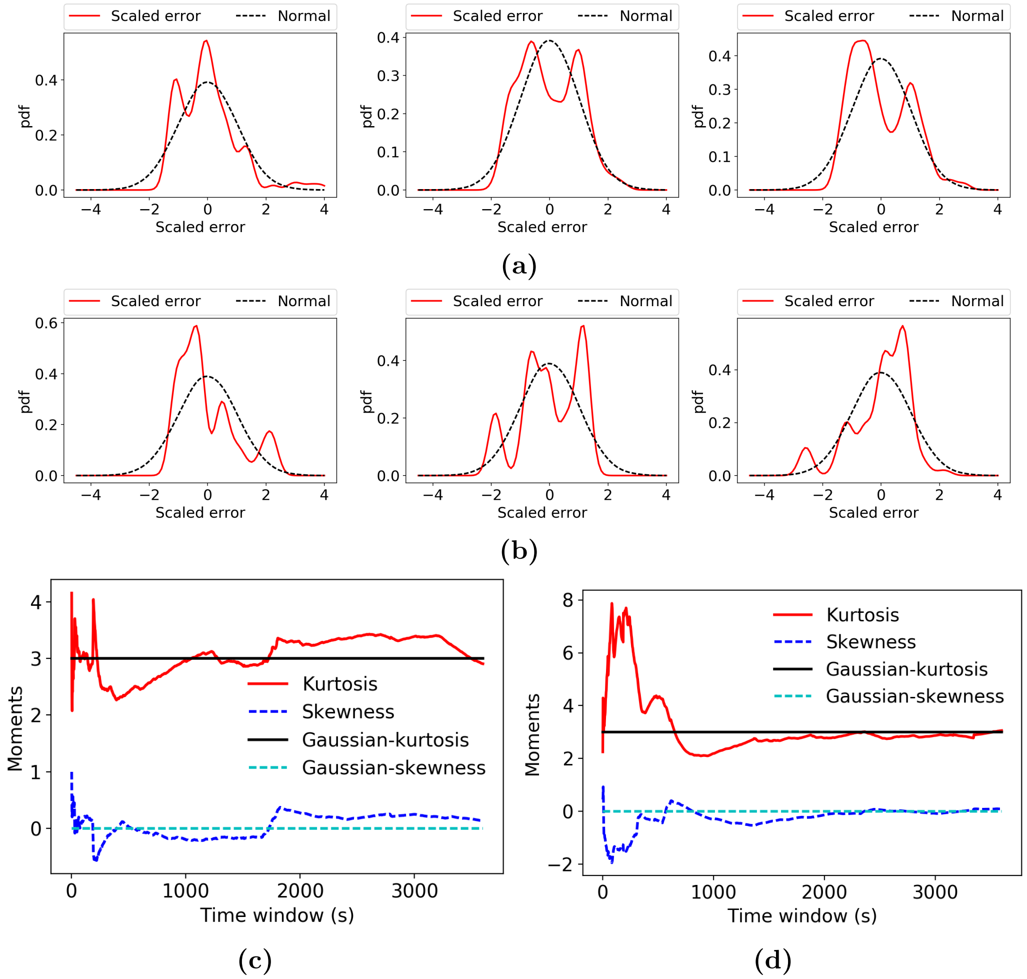
To sum up, today's power distribution system is being transformed from a passive system into an active and intelligent network. It is advantageous to understand the distribution system measurement characteristics, which is critical to distribution system applications (e.g., state estimation). This work studies the distribution-level synchrophasor measurement error and shows that, based on various tests, the measurement error follows a non-Gaussian distribution, instead of the traditionally assumed Gaussian distribution. It suggests the use of the non-Gaussian model, such as Gaussian mixture model (GMM), for representing the measurement error, which is more accurate and realistic than the traditional Gaussian model. The GMM parameters can be deduced through fitting GMM models to the observations or can be treated as unknowns with some prior distributions and updated via Bayesian inference. The obtained results are helpful for the understanding of distribution-level measurement characteristics, and for the modeling and simulation of distribution system applications.

Table 4.2: Summary of the Gaussianity Tests on PMU Data

Measurement	Time window (s)	Sample size	% of non-Gaussian distributions			
			Shapiro-Wilk		Kolmogorov-Smirnov	
			α=5%	α=10%	α=5%	α=10%
Voltage angle	1	60	44	48	24	28
	5	300	96	100	88	88
	10	600	100	100	100	100
	60	3600	100	100	100	100
Voltage magnitude	1	60	4	8	0	0
	5	300	84	88	36	40
	10	600	92	92	72	72
	60	3600	100	100	100	100

Table 4.3: Summary of the Gaussianity Tests on FDR Data

Measurement	Time window (s)	Sample size	% of non-Gaussian distributions			
			Shapiro-Wilk		Kolmogorov-Smirnov	
			α=5%	α=10%	α=5%	α=10%
Voltage angle	5	50	32	46	4	10
	30	300	100	100	100	100
	60	600	100	100	100	100
	120	1200	100	100	100	100
Voltage magnitude	5	50	100	100	100	100
	30	300	100	100	100	100
	60	600	100	100	100	100
	120	1200	100	100	100	100



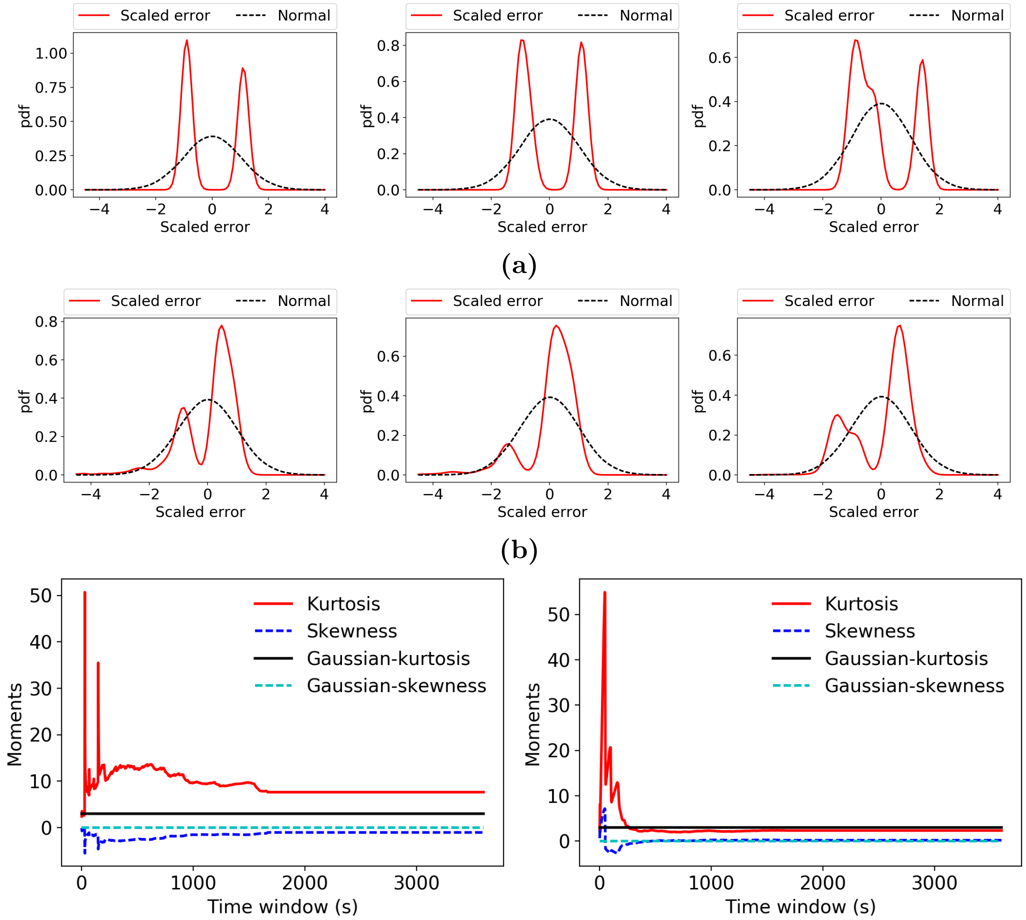
(a)

(b)

(c)

(d)

Fig. 4.2. Test 1 results: (a) pdf of the voltage angle error differences Δe_s with standard Gaussian for time windows of 1s, 5s, 10s; (b) pdf of the voltage magnitude error differences Δe_s with standard Gaussian for time windows of 1s, 5s, and 10s; (c) varying time windows versus skewness and kurtosis of voltage angle Δe; and (d) varying update period versus skewness and kurtosis of voltage magnitude Δe.

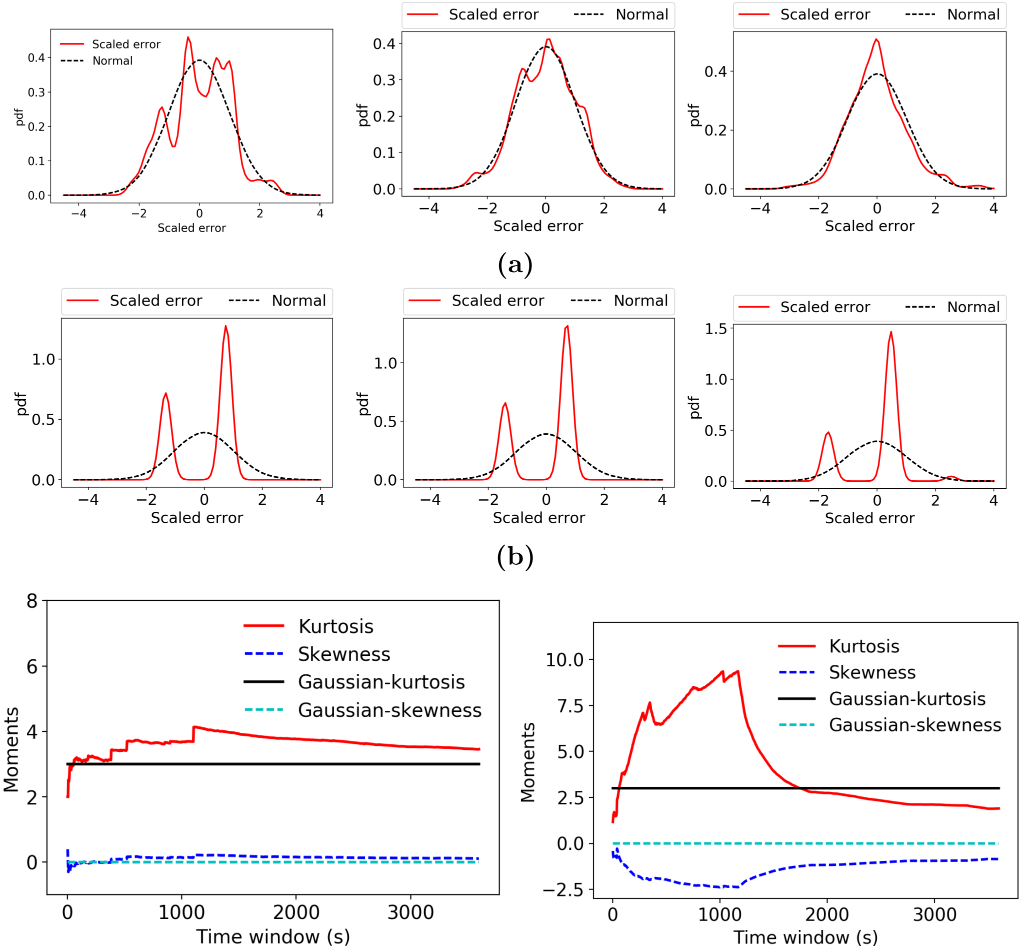


(a)

(b)

(c) **(d)**

Fig. 4.3. Test 2 results: (a) pdf of the voltage angle error differences Δe_s with standard Gaussian for time windows of 1s, 5s, 10s; (b) pdf of the voltage magnitude error differences Δe_s with standard Gaussian for time windows of 1s, 5s, and 10s; (c) varying time windows versus skewness and kurtosis of voltage angle Δe; and (d) varying update period versus skewness and kurtosis of voltage magnitude Δe.



(a)

(b)

(c)

(d)

Fig. 4.4. Test 3 results: (a) pdf of the voltage angle errors with standard Gaussian for time windows of 5s, 30s, 60s; (b) pdf of the voltage magnitude errors with standard Gaussian for time windows of 5s, 30s, and 60s; (c) varying time windows vs skewness and kurtosis of voltage angle Δe; and (d) varying time windows vs skewness and kurtosis of voltage magnitude Δe.

4.2 State estimation

Motivation State estimation (SE) is one of the most significant components of power grid energy management systems (EMS). It typically finds accurate and reliable state estimates through a set of redundant measurements and further enables EMS to perform various planning and operation tasks. In the past, SE was designed for transmission systems and rarely deployed for distribution systems due to the subordinate nature of distribution networks. Currently, distribution systems are becoming increasingly complex and uncertain with high penetration of distributed energy sources (DER), and also more visibility with the development of micro-PMU (PMU) and advanced metering infrastructure (AMI). Consequently, the increasing network complexity along with the increasing measurement availability make SE as an essential part of distribution management systems (DMS), providing DMS real-time system states and greatly improving the system observability and controllability.

The current distribution system SE (DSSE) algorithms mainly derive from the transmission system SE, e.g., weighted least squares (WLS) based SE and Kalman filters (KF) based SE, and inherit the Gaussian/linearity assumptions used in the WLS/KF SE. However, these assumptions are violated especially in the uncertain and diverse distribution network with high penetration of DER. Furthermore, the existing SE/DSSE algorithms view the power grid as an over determined system, which ignore the measurement possibilities and model uncertainties and only work out a group of deterministic estimates. From a statistical point of view, the deterministic state estimates ignore some valuable information for the distribution systems with dynamic and diverse nature. To overcome the limitations above, an advanced state estimation method is proposed for power grid DMS.

4.2.1 Problem formulation

In the distribution systems, the relation between measurements $\boldsymbol{z}$ and state variables $\boldsymbol{x}$ is typically formulated as:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e}, \tag{4.2}$$

where z is the M-dimensional vector of measurements typically obtained from the μPMU or FNET/GridEye [64, 71, 67]; $\boldsymbol{x}$ is the N-dimensional vector of state variables such as voltage magnitude and phase angels; $\boldsymbol{h}$ is the non-linear function that relates $\boldsymbol{x}$ and $\boldsymbol{z}$; and e is the vector of measurement errors. The goal of DSSE is to enhance the knowledge of the states given measurements. Given limited number of measurements and non-linearities in the measurement model defined in the Equation (4.2), the corresponding problem becomes mathematically ill-posed. However, these methods produce Gaussian posterior states, given the nonlinear relation between the states and measurements will lead to non-Gaussian posterior states even with simple additive noise and Gaussian priors. Also, as pointed out by the previous literature, the measurement errors follow non-Gaussian distributions leading to non-Gaussian posterior states.

We consider Bayesian inference, a systematic framework for integrating prior knowledge and measurement uncertainties. It treats the states $\boldsymbol{x}$ as a random variables and computes the posterior distribution via

$$\pi_{post}(\boldsymbol{x}) := \pi(\boldsymbol{x}|\boldsymbol{z}) \propto \pi_{prior}(\boldsymbol{x})\pi_{lik}(\boldsymbol{z}|\boldsymbol{x}). \tag{4.3}$$

In this context, $\pi_{prior}(\boldsymbol{x})$ is a prior state model that allows us to infuse expert knowledge into the measurement model; $\pi_{prior}(\boldsymbol{x})$ is the likelihood function, a conditional probability of the measurements given the states. Here, we assume a Gaussian mixture (GM) model for the measurement error, that the probability density function (pdf) of error $\boldsymbol{e}$ as weighted sum of the Gaussians as shown below,

$$p(\boldsymbol{e}|\boldsymbol{\gamma}) = \sum_{i=1}^{N_c} w_i p(\boldsymbol{e}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{4.4}$$

Here, N_c is the number of Gaussian mixtures and w_i a positive quantity with $\sum w_i = 1$; $\boldsymbol{\gamma}$ are the set of parameters such as mean $\boldsymbol{\mu}_i$, covariance $\boldsymbol{\Sigma}_i$, w_i that describe each component of the GM; and $p(\boldsymbol{e}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is M-dimensional Gaussian density function of each mixture component given by

$$p(\boldsymbol{e}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{e^{-\frac{1}{2}(\boldsymbol{e}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{e}-\boldsymbol{\mu}_i)}}{(2\pi)^{M/2}\det(\boldsymbol{\Sigma}_i)^{1/2}}. \tag{4.5}$$

The parameters of the GM, $\boldsymbol{\gamma}$, are computed from the measurements via expectation-maximization (EM) algorithm. Given the nonlinear mapping between the states and measurements and GM model for the error, the posterior states follow non-standard distributions and Markov chain Monte Carlo (MCMC) methods are relevant techniques for posterior sampling.

In DSSE, we require real-time state estimation and MCMC becomes computationally intensive due to expensive measurement model and high-dimensionality of the states. To reduce the computational complexity we project random field on to a sub-space via KPCA as described in Section 2.2 and map the corresponding feature variables to Gaussian random variables via PCE similar to Section 2.3. Figure 4.5 illustrates mapping of the state space to the Gaussian space via KPCA and PCE.

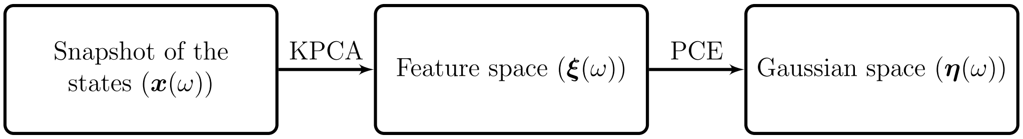

Snapshot of the states ($\boldsymbol{x}(\omega)$)
KPCA
Feature space ($\boldsymbol{\xi}(\omega)$)
PCE
Gaussian space ($\boldsymbol{\eta}(\omega)$)

Fig. 4.5. Mapping state space to the Gaussian space

Instead of performing the Bayesian inversion in the original state space we run the inversion in the $\boldsymbol{\eta}$ space and corresponding posterior can be obtained via,

$$\pi_{post}(\boldsymbol{\eta}) := \pi(\boldsymbol{\eta}|\boldsymbol{z}) \propto \pi_{prior}(\boldsymbol{\eta})\pi_{lik}(\boldsymbol{z}|\boldsymbol{\eta}), \tag{4.6}$$

here the computation of the likelihood use following transformation to map $\boldsymbol{\eta}$ to measurements $\boldsymbol{z}$

$$\boldsymbol{\eta} \xrightarrow{\text{PCE}} \boldsymbol{\xi} \xrightarrow{\text{Pre-image}} \boldsymbol{x} \xrightarrow{\text{forward model}} \boldsymbol{z}. \tag{4.7}$$

We employ Metropolis-Hasting MCMC (MHMCMC) to sample form the posterior4.3 that scales linearly $(O(n))$ with the dimension of state space. The MHMCMC algorithm accepts or rejects a sample from the proposal using α,

$$\alpha = \min\{1, \frac{\pi_{post}(\boldsymbol{\eta}_{k+1})q(\boldsymbol{\eta}_k|\boldsymbol{\eta}_{k+1})}{\pi_{post}(\boldsymbol{\eta}_k)q(\boldsymbol{\eta}_{k+1}|\boldsymbol{\eta}_k)}\} \tag{4.8}$$

where q is the proposal distribution.

4.2.2 Case Studies

The snapshots of the states are generated by solving standard weighed least-squares state estimation procedure multiple times, via perturbing the measurements with uniform noise. Fig 4.6 depicts the corresponding histogram of the prior states and Fig. 4.7 shows the Eigenvalue spectrum of the snapshots computed with KPCA with polynomial kernels of degree 1, 2, 3, 4 and 5.

Figures 4.8 and 4.9 show the box-plot of the states and Fig. 4.10 depicts the posteriors of the posteriors of the states computed using KPCA-MCMC approach. It is found that the posterior distributions of the states are non-Gaussian asymmetric distributed. The median values obtained by the proposed method matches the true values well. However, voltage angle's corresponding confidence interval is relatively large, yielding an estimation results with low confidence. In practice, the high-confidence estimation is preferred. This motivates us to use the weighted Bayesian inference. By this way, the oberservability for differences states can be improved. By this way,the confidence interval for the voltage angles is reduced and the confidence interval for the magnitude estimation is increased.

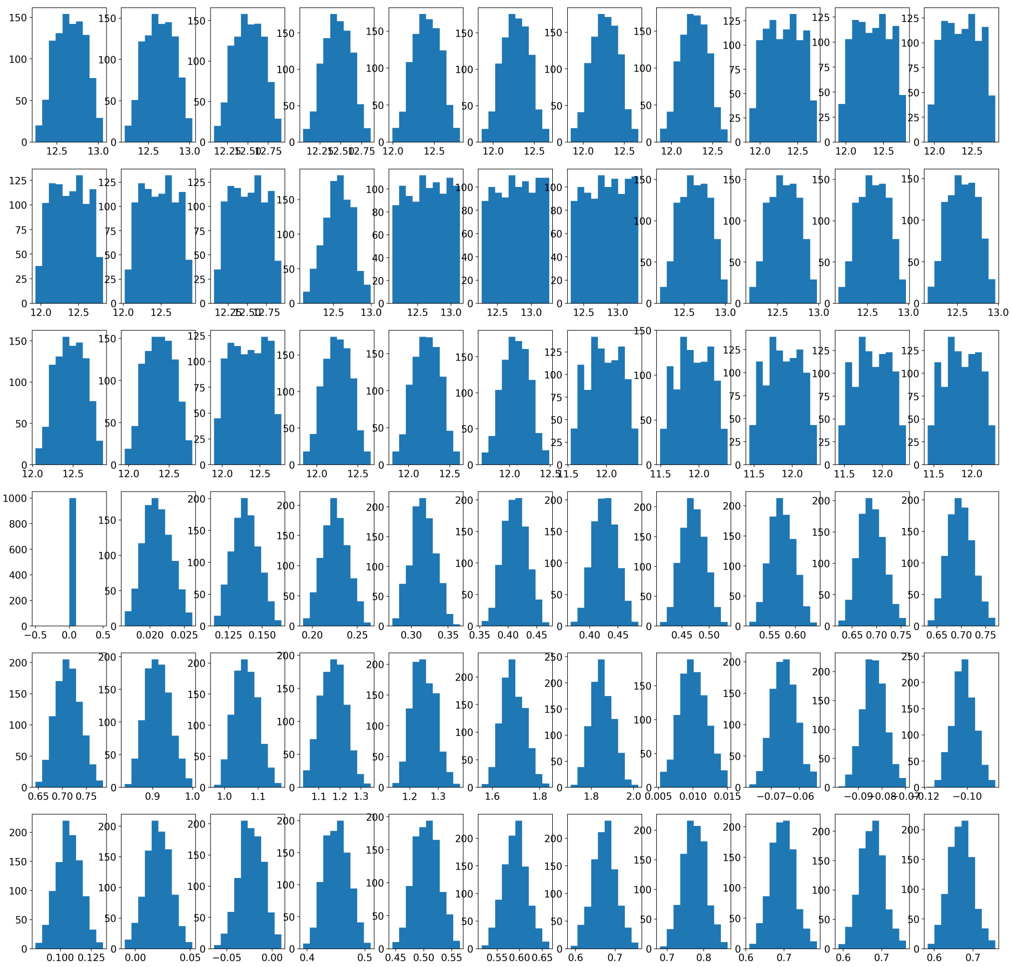


Fig. 4.6. Histograms of the prior state vectors

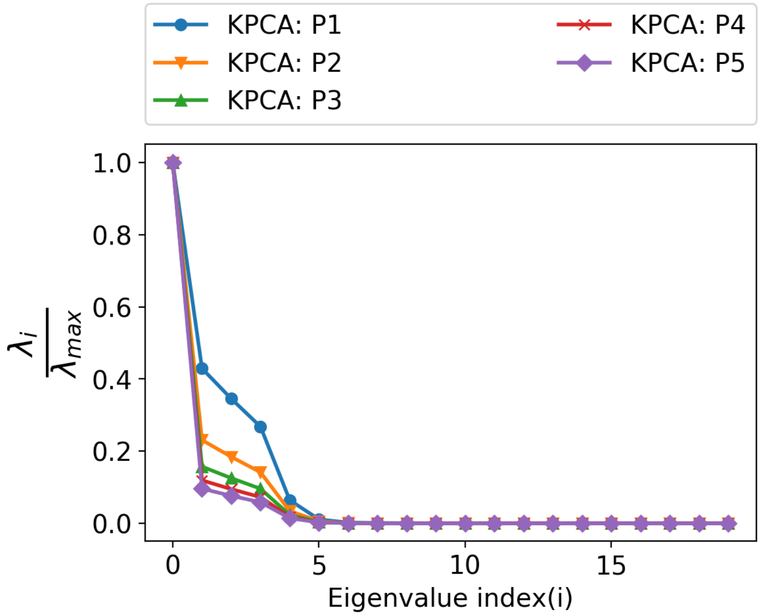


Fig. 4.7. Eigenvalue spectrum of the snapshots

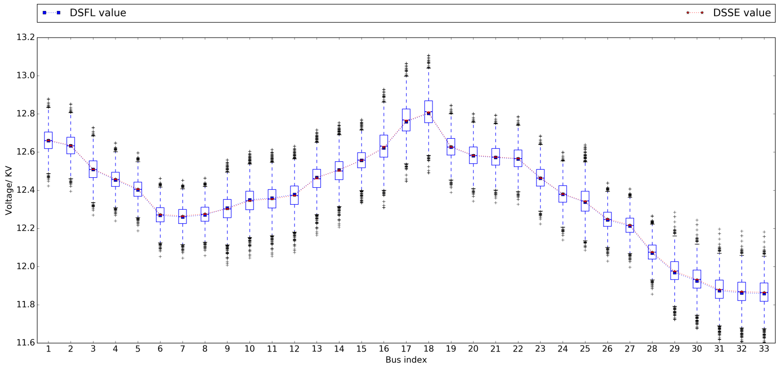


Fig. 4.8. Box-plot of the voltage

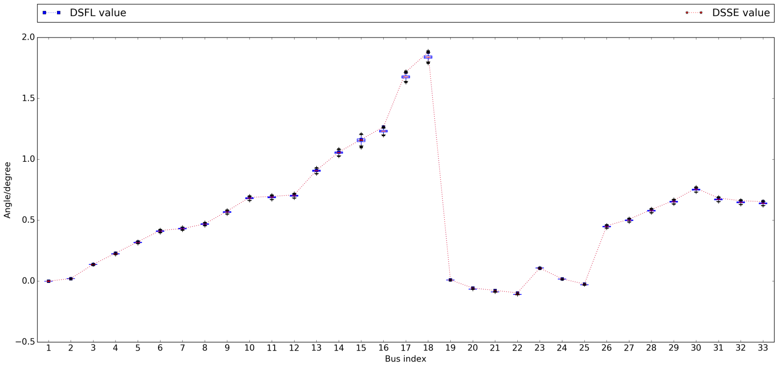


Fig. 4.9. Box-plot of the angles

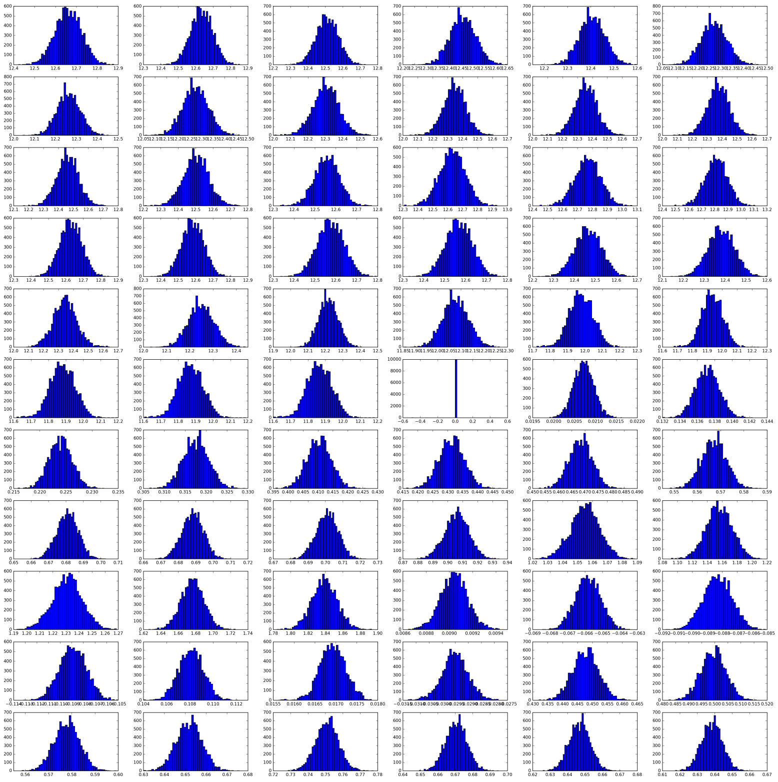


Fig. 4.10. Histograms of the posterior state vectors

4.3 Parameter estimation

Motivation Accurately estimating the key parameters of generator dynamic models is of paramount importance to power system dynamic security analysis. Indeed, inaccurate parameter values may lead to inappropriate control actions, which in turn may result in power system failures. It was found that the well-known 1996 blackout in the Western Electricity Coordinating Council (WECC) system was caused by severe model discrepancy. This has prompted the North American Electric Reliability Corporation (NERC) to issue reports that require the validation and the calibration of the models of large generators and their controllers in North America every five years, aiming to have the dynamic model responses match the recorded measurements reasonably well [72, 73, 74]. To address this need, this work proposes a novel method for estimating generator dynamic model parameters using Bayesian formalism.

Since traditional offline stage-testing-based methods are very costly, time-consuming, and labor intensive for generator model validation and parameter estimation [75], a number of low-cost PMU-based methods have been proposed recently. Some of them are focused on estimating the generator moment of inertia [76, 77, 78], [79] while others are focused on estimating the aggregated inertia of the system[80]. However, the dynamic responses of the power systems are not governed by the moment of inertia only. Simply using the swing equations to describe the dynamic responses may lead to inaccurate estimation of the generator model parameters. In [75], [81], it has been shown in the sensitivity-based methods that other generator model parameters also play key roles in influencing the power system dynamic performance. To estimate these key parameters when the system is subject to disturbances, methods based on the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the ensemble Kalman filter (EnKF) have been advocated in [75, 81, 74]. While overall these methods exhibit good performances, in practice, they still suffer from several weaknesses such as poor convergence rate and Gaussian assumption of the process and measurements error.

Unlike the weighted-least-squares (WLS)-based methods proposed, for example, in [82, 83], which provide only point estimated values for the generator model parameters without providing the confidence intervals, Bayesian-inference based methods provide maximum-a-posteriori (MAP) parameter estimated values along with the corresponding posterior probability distributions [79],[84]. Furthermore, they are able to cope with non-Gaussian errors [84]. For these reasons, Bayesian methods have gained a great deal of attention among power researchers. However, the posterior distribution in Bayesian inference is typically obtained by Monte-Carlo-based methods, which require prohibitive computing time when they are performed on detailed generator models.

To overcome the aforementioned weaknesses, this work proposes a response-surface-based Bayesian inference algorithm for power system dynamic parameter estimation, resulting in the following contributions:

- To eliminate the uncertainties of the line and transformer models and the loads within a centralized system model[79], a decentralized generator model is used [85], [86]. This significantly reduces the number of parameters, allowing effective Bayesian inference for power system dynamic parameter estimation.

- To deal with the non-Gaussian posterior distribution, the Metropolis-Hasting sampler is adopted. This is a Markov chain Monte Carlo (MCMC) method aimed at finding the global optima for the MAP estimator. This overcomes the shortcomings of the local optimization approach [79], in which the local optima might be highly biased when the initial guess is far from the true value or the posterior distribution is non-Gaussian.

- To greatly accelerate the MCMC algorithm [84], the response surfaces of the dynamical system represented by the state-of-the-art polynomial chaos expansion (PCE) are developed [87].

- To identify the key generator model parameters that satisfy system observability, a polynomial-chaos-based ANOVA algorithm is proposed.

Simulation results carried out on the New-England system reveal that our proposed method can accurately and simultaneously estimate several key generator dynamic model parameters with almost two order of magnitude improved in speed compared with the traditional MCMC-based Bayesian inference method for non-Gaussian posterior distribution.

4.3.1 Problem formulation

This section formulates the Bayesian inference framework for the decentralized synchronous generator model and the measurement model. Following Sauer and Pai [88], the two-axis model with a IEEE-DC1A exciter and a

TGOV1 turbine-governor is considered. It is represented by the following differential and algebraic equations:

$$T'_{d_0}\frac{dE'_q}{dt} = -E'_q - (X_d - X'_d)I_d + E_{fd}, \tag{4.9}$$

$$T'_{q_0}\frac{dE'_d}{dt} = -E'_d - (X_q - X'_q)I_q, \tag{4.10}$$

$$\frac{d\delta}{dt} = \omega - \omega_s, \tag{4.11}$$

$$\frac{2H}{\omega_s}\frac{d\omega}{dt} = T_M - P_e - D(\omega - \omega_s), \tag{4.12}$$

$$T_E\frac{dE_{fd}}{dt} = -(K_E + S_E(E_{fd}))E_{fd} + V_R, \tag{4.13}$$

$$T_F\frac{dV_F}{dt} = -V_F + \frac{K_F}{T_E}V_R - \frac{K_F}{T_E}(K_E + S_E(E_{fd}))E_{fd}, \tag{4.14}$$

$$T_A\frac{dV_R}{dt} = -V_R + K_A(V_{\text{ref}} - V_F - V), \tag{4.15}$$

$$T_{CH}\frac{dT_M}{dt} = -T_M + P_{SV}, \tag{4.16}$$

$$T_{SV}\frac{dP_{SV}}{dt} = -P_{SV} + P_C - \frac{1}{R_D}(\frac{\omega}{\omega_s} - 1), \tag{4.17}$$

$$V_d = V\sin(\delta - \theta), V_q = V\cos(\delta - \theta), \tag{4.18}$$

$$I_d = \frac{E'_q - V_q}{X'_d}, I_q = \frac{V_d - E'_d}{X'_q}, \tag{4.19}$$

$$P_e = V_dI_d + V_qI_q, Q_e = -V_dI_q + V_qI_d, \tag{4.20}$$

where T'_{d_0}, T'_{q_0}, T_E, T_F, T_A, T_{CH} and T_{SV} are the time constants; K_E, K_F and K_A are the controller gains; E'_d, E'_q, E_{fd}, V_F, V_R, T_M and P_{SV} are the d-axis and q-axis transient voltages, field voltage, scaled output of the stabilizing transformer and scaled output of the amplifier, synchronous machine mechanical torque and steam valve position, respectively; X_d, X'_d, X_q and X'_q are the generator parameters; H, D and R_D are the inertia constant in seconds, damping ratio and droop respectively; V_{ref} and P_C are the known voltage reference and power references of the exciter and speed governor, respectively.

Following Zhao and Mili [86], once the local PMU measurements are recorded and the parameters of the ith generator model to be estimated are selected while satisfying state observability, we take the voltage phasors $V_i\angle\theta_i$ as the known model inputs and the current phasors $I_i\angle\phi_i$ as the model outputs. Based on (4.18) and (4.19), we have $V_{di} = V_i\sin(\delta_i - \theta_i)$, $V_{qi} = V_i\cos(\delta_i - \theta_i)$, $I_{di} = (E'_{qi} - V_{qi})/X'_{di}$ and $I_{qi} = (V_{di} - E'_{di})/X'_{qi}$. However, paper [89] shows that the terminal real and reactive powers enable a better observability of the generator states than the current phasors. From (4.20), we further take the active power and reactive power for the ith decentralized generator as model outputs expressed as

$$P_{ei} = V_{di}I_{di} + V_{qi}I_{qi} + e_{Pi}, \tag{4.21}$$

$$Q_{ei} = -V_{di}I_{qi} + V_{qi}I_{di} + e_{Qi}, \tag{4.22}$$

where e_{Pi} and e_{Qi} are the measurement noise. Up to now, we have formulated the decentralized generator model. Once we capture the local voltage phasor $V_i\angle\theta_i$ for the ith generator in the selected time period, given its corresponding generator parameters, we can obtain the trajectories of its active and reactive power by calculating the ith generator model's output using the aforementioned differential and algebraic equations given by (1)-(14). For more details and discussions, the reader is referred to [86].

Tarantola [90] formulated the model for Bayesian inference as

$$\boldsymbol{d} = \boldsymbol{f}(\boldsymbol{m}) + \boldsymbol{e}, \tag{4.23}$$

where $\boldsymbol{d}$ contains the observations, which consists of d_1 that contains the active power measurements, P_{ei}, and d_2 that contains the reactive power measurements, Q_{ei}; $\boldsymbol{m} \in \mathbb{R}^N$ are the parameters to be estimated; N is the number of parameters to be estimated that depends on the specific applications; $\boldsymbol{f}(\cdot)$ is the vector-valued forward function that represents the aforementioned differential and algebraic equations, which map the model parameters $\boldsymbol{m}$ to the observations $\boldsymbol{d}$; $\boldsymbol{e} \in \mathbb{R}^2$ stands for the measurement error vector whose components are assumed to be mutually independent random variables with the joint probability density functions $\pi_{\boldsymbol{e}}$ defined as $\pi_{\boldsymbol{e}} = \prod_{i=1}^{2} \pi_{e_i}(e_i)$. In the Bayesian inference, each parameter m_i is also viewed as a random variable with a given prior probability distribution, whose probability density function (pdf) is denoted as $\pi_i(m_i)$. The corresponding joint prior density function for a vector $\boldsymbol{m}$ is given by

$$\pi_{\text{prior}}(\boldsymbol{m}) = \prod_{i=1}^{N} \pi_i(m_i). \tag{4.24}$$

Note that here $\boldsymbol{e}$ and $\boldsymbol{m}$ are also assumed to be mutually independent.

Given the observation $\boldsymbol{d}$, the posterior pdf $\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})$ for the parameters $\boldsymbol{m}$ is derived as [90]

$$\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d}) \propto \pi_{\text{like}}(\boldsymbol{d}|\boldsymbol{m})\pi_{\text{prior}}(\boldsymbol{m}). \tag{4.25}$$

Here $\pi_{\text{like}}(\boldsymbol{d}|\boldsymbol{m})$ denotes the likelihood function, expressed as

$$\pi_{\text{like}}(\boldsymbol{d}|\boldsymbol{m}) = \prod_{i=1}^{2} \pi_{e_i}(d_i - f_i(\boldsymbol{m})). \tag{4.26}$$

Given a set of parameters contained in $\boldsymbol{m}$, we obtain the trajectories of P_{ei} and Q_{ei} from the forward solver $\boldsymbol{f}(\cdot)$. By comparing them to the PMU metered values for the simulated time period t_{end}, the likelihoods for the corresponding trajectories are evaluated. Let us denote $\pi_{e_i}^t$, d_i^t and f_i^t as the likelihood, the observation and the realization at time t, respectively. The likelihood function for the trajectories in the log-form is then expressed as

$$\log \pi_{e_i}(d_i - f_i(\boldsymbol{m})) = \sum_{t=0}^{t_{\text{end}}} \log \pi_{e_i}^t(d_i^t - f_i^t(\boldsymbol{m})). \tag{4.27}$$

Thanks to the high speed sampling rate of the PMUs, which is typically equal to 60 samples/s, for a short time period such as 3 seconds, we get 180 samples for P_{ei} and for Q_{ei}, which provide good tracking information of the dynamic responses of a power system following a disturbance. Now the relationship given by (4.25) can be put into the following form:

$$\log(\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})) = \sum_{i=1}^{2} \log \pi_{e_i}(d_i - f_i(\boldsymbol{m})) + \sum_{i=1}^{N} \log(\pi_i(m_i)), \tag{4.28}$$

yielding the MAP estimator defined as

$$\hat{\boldsymbol{m}}_{\text{MAP}} = \arg\min_{\boldsymbol{m}} \quad \{-\log(\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d}))\}. \tag{4.29}$$

Note that due to the non-linearity of $\boldsymbol{f}(\cdot)$, even if the prior assumption is Gaussian, $\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})$ may be non-Gaussian. This motivates us to use the MCMC method instead of simply making the Gaussian assumptions for the posterior pdfs of the parameters $\boldsymbol{m}$. The detailed procedure to calculate the MCMC via the Metropolis-hasting sampler will be described in Section IV.

Response-Surface-Based Bayesian Inference The MCMC method is very time-consuming when the forward solver is complex. To accelerate it, we propose to replace the forward solver with a PCE-based response surface, as explained next.

The generalized polynomial chaos expansion has been shown to be a cost-effective tool in modeling response surfaces [91], [92], [93]. In the gPC method, the stochastic outputs are represented as a weighted sum of a given set of orthogonal polynomial chaos basis functions constructed from the probability distribution of the input random variables. Let $\boldsymbol{\xi} = [\xi_1, \xi_2, ...\xi_N]$ be a vector of random variables following a standard probability distribution (e.g. the Gaussian or the beta distribution), to which, as shown in [**?**], a unique orthogonal polynomial is associated. Let $\Phi_i(\xi_1, \xi_2, ...\xi_N)$ denote this procedure's corresponding polynomial chaos basis and let a_i denote the ith polynomial chaos coefficient. Formally, we have

$$z = \sum_{i=0}^{\infty} a_i \Phi_i(\boldsymbol{\xi}). \tag{4.30}$$

In practice, a truncated expansion is used such that

$$z = \sum_{i=0}^{N_P} a_i \Phi_i(\boldsymbol{\xi}), \tag{4.31}$$

where $N_P = (N+P)!/(N!P!) - 1$, N is the total number of the random variables involved in the gPC and P is the maximum order of the polynomial chaos basis functions. It is found that a relatively low maximum polynomial chaos order, typically 2, is found to provide output results with enough accuracy [92], [93], [94]. From the polynomial chaos coefficients, the mean, μ, and the variance, σ^2, of the output z can be determined as follows:

$$\mu = a_0, \quad \sigma^2 = \sum_{i=1}^{N_P} a_i^2 E[\Phi_i^2], \tag{4.32}$$

Table 4.4: UNIVARIATE GPC POLYNOMIAL BASIS OF DIFFERENT RANDOM VARIABLES

Random Variable	Polynomial Basis Function	Support
Gaussian	Hermite	$(-\infty, +\infty)$
Gamma	Laguerre	$[0, +\infty)$
Beta	Jacobi	$[0, 1]$
Uniform	Legendre	$[-1, 1]$

where $E[.]$ is the expectation operator.

A set of one-dimensional polynomial chaos basis functions $\{\Phi_i(\xi), i = 0, 1, 2, 3....\}$ with respect to some real positive measure satisfy the following relation:

$$\int \Phi_r(\xi)\Phi_s(\xi)\mathrm{d}\lambda = \begin{cases} 0 & \text{if } r \neq s \ , \\ > 0 & \text{if } r = s. \end{cases} \tag{4.33}$$

Here λ is a probability measure defined as the cumulative probability distribution function (CPDF) of ξ. For every CPDF, the associated orthogonal polynomials are unique.

Similarly, any set of multi-dimensional polynomial chaos basis functions $\{\Phi_i(\boldsymbol{\xi}), i = 1, 2, 3....\}$, are orthogonal to each other with respect to their joint probability measure.

Construction of the Polynomial Chaos Basis A set of multi-dimensional polynomial chaos basis functions can be constructed as the tensor product of the one-dimensional polynomial chaos basis associated with each input random variable. Formally, we have

$$\Phi(\boldsymbol{\xi}) = \Phi(\xi_1) \otimes \Phi(\xi_2) \otimes \cdots \otimes \Phi(\xi_N), \tag{4.34}$$

where $\Phi(\xi_i)$ denotes the one-dimensional polynomial chaos basis for the ith random variable.

Collocation Points collocation points can be regarded as a finite sample of $\boldsymbol{\xi} = [\xi_1, \xi_2, \cdots, \xi_N]$ that are chosen to approximate the polynomial chaos coefficients. The elements of the collocation points are generated by using the union of the zeros and the roots of one higher-order, one-dimensional polynomial for every random variable [93, 87]. For example, for a 2nd-order Hermite polynomial, its one higher-order polynomial is $\phi_3(\xi) = \xi^3 - 3\xi$. The elements of the collocation points are $\{\sqrt{3}, -\sqrt{3}, 0\}$. With these 3 collocation point elements, if there are N random variables, the number of possible combinations is 3^N. For example, the $N_P + 1$ unknown coefficients can be estimated by the WLS method introduced below. Other methods to generate collocation points such as sparse grid can be found in [87].

Building PCE-based Response Surface for Dynamic Power Systems
In the Bayesian inference, the parameters $\boldsymbol{m}$ are viewed as random variables and hence, are given prior PDFs. By mapping the parameters $\boldsymbol{m}$ into $\boldsymbol{\xi}$, we can build a PCE as the response surface of the dynamic power system model. The detailed gPC procedure is as follows:

1. Map the ith random parameter, m_i, to a given random variable, ξ_i, as follows:

$$P_i = F_i^{-1}(T(\xi_i)), \tag{4.35}$$

   where F_i^{-1} is the inverse cumulative probability distribution function of m_i and T is the cumulative probability distribution function of ξ_i.

2. Construct the polynomial chaos basis, then express the output z in the gPC expansion form of (4.31).

3. Construct M combinations of collocation points and put them into the polynomial chaos basis $(M \times (N_P + 1))$ matrix H_{pc}. Formally, we have

$$H_{pc} = \begin{pmatrix} \Phi_0(\boldsymbol{\xi_1}) & \Phi_1(\boldsymbol{\xi_1}) & \dots & \Phi_{N_P}(\boldsymbol{\xi_1}) \\ \Phi_0(\boldsymbol{\xi_2}) & \Phi_1(\boldsymbol{\xi_2}) & \dots & \Phi_{N_P}(\boldsymbol{\xi_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_0(\boldsymbol{\xi_M}) & \Phi_1(\boldsymbol{\xi_M}) & \dots & \Phi_{N_P}(\boldsymbol{\xi_M}) \end{pmatrix}; \tag{4.36}$$

4. Compute the power system dynamic model output for the selected collocation points to get the $(M \times 1)$ output Z matrix given by

$$Z = \begin{pmatrix} z(t, \boldsymbol{\xi_1}) & z(t, \boldsymbol{\xi_2}) & \dots & z(t, \boldsymbol{\xi_M}) \end{pmatrix}^T; \tag{4.37}$$

5. Estimate the unknown coefficients A based on the collocation points that are selected and the model output from:

$$Z = H_{pc}A. \tag{4.38}$$

   A is the $(N_P \times 1)$ coefficient vector expressed as

$$A = \begin{pmatrix} a_0(t) & a_1(t) & \dots & a_{N_P}(t) \end{pmatrix}^T ; \tag{4.39}$$

6. Let $\hat{A}$ denote the estimated coefficient vector and let us define a residual vector r as $r = Z - H_{pc}\hat{A}$. Let us minimize the 2-norm of the residual vector to estimate $\hat{A}$, that is,

$$\hat{A} = \underset{\hat{A}}{\arg\min} \quad r^T r, \tag{4.40}$$

   which yields $\hat{A} = (H_{pc}^T H_{pc})^{-1} H_{pc}^T Z$.

With the coefficients estimated and the bases selected, we can build the PCE for the target output. The system response surface can now be represented as polynomial form.

Incorporating Polynomial Chaos Expansion into the Bayesian Inference Framework In the gPC-based Bayesian inference [91, 92], we use the approximated gPC solution in (4.31) to replace the exact forward solver solution $\boldsymbol{f(m)}$ in (4.28) as

$$\log(\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})) = \sum_{i=1}^{2} \log \pi_{e_i}(d_i - z(\boldsymbol{\xi})) + \sum_{i=1}^{N} \log(\pi_i(m_i)). \tag{4.41}$$

Once a sample point $\boldsymbol{m}$ is proposed in MCMC, we can first map $\boldsymbol{m}$ to $\boldsymbol{\xi}$ via (4.35) and then evaluate $\boldsymbol{\xi}$ with the PCE-based response surface without resorting to actual simulations of the forward solver. Therefore, we achieve very high accuracy in sampling the posterior distribution at a much less computational cost.

Application of the Proposed Response-Surface-Based Bayesian Inference Algorithm This section illustrate the application of the proposed response-surface-based Bayesian inference algorithm via the Metropolis-Hastings (M-H) algorithm to achieve a decentralized power system dynamic parameter estimation. Starting from the initial guess, which can be the manufactured data, the M-H method employs a given PDF, $q(\boldsymbol{m_k}, \cdot)$ at each sample point $\boldsymbol{m}_k$ to generate a proposed sample point $\boldsymbol{m_{k+1}}$. Once generated, the sample point is either accepted or rejected by the M-H method. This procedure is then applied to the next sample point, yielding a chain of sample points from the posterior pdf $\pi_{\text{post}}(\boldsymbol{m})$ [90]. To further stabilize the numerical computation of the M-H algorithm, the posterior pdf is transformed into the log-form as suggested in [84]. The M-H algorithm used in the classical Bayesian inference is described in Algorithm 1. In this algorithm, the most time-consuming step happens in step 5, this is accelerated by the PCE-based response surface, as shown in the modified M-H algorithm given in Algorithm 2. In this modified version, the sample size needed for PCE surrogate construction is much smaller than the samples required by the original M-H algorithm. This step can also be executed offline. In the following, we compare the two Bayesian-inference-based parameter estimation methods.

Algorithm 3 The Bayesian Inference using the M-H Algorithm

```
Choose the initial guess of the parameters $\boldsymbol{m_0}$ from the manufactured data as the Bayesian prior $\boldsymbol{m}_{\text{prior}}$;
Compute $\log(\pi_{\text{post}}(\boldsymbol{m_0}|\boldsymbol{d}))$ for the exact decentralized dynamic model from (4.28);
for $k = 0, \ldots, N_{\text{samples}} - 1$ do
    Generate new sample $\boldsymbol{m_{k+1}}$ from the proposal function $q(\boldsymbol{m_k}, \cdot)$;
    Compute $\log\left(\boldsymbol{\pi_{\text{post}}}(\boldsymbol{m_{k+1}}|\boldsymbol{d})\right)$ from the exact decentralized dynamic model by (4.28);
    Calculate the correction factor $c = \frac{q(\boldsymbol{m_{k+1}},\boldsymbol{m_k})}{q(\boldsymbol{m_k},\boldsymbol{m_{k+1}})}$;
    Compute $\alpha(\boldsymbol{m_k}, \boldsymbol{m_{k+1}})$ defined as
        $$\alpha(\boldsymbol{m_k}, \boldsymbol{m_{k+1}}) = \log(min\{1, \frac{\pi_{\text{post}}(\boldsymbol{m_{k+1}}|\boldsymbol{d})}{\pi_{\text{post}}(\boldsymbol{m_k}|\boldsymbol{d})} \cdot c\});$$
    Draw $u \sim \mathcal{U}([0,1))$;
    if $\log(u) < \alpha(\boldsymbol{m_k}, \boldsymbol{m_{k+1}})$ then
        Accept: Set $\boldsymbol{m}_{k+1} = \boldsymbol{m_{k+1}}$;
    else
        Reject: Set $\boldsymbol{m}_{k+1} = \boldsymbol{m}_k$;
    end if
end for
Plot the PDF of the $N_{\text{samples}}$ of $\boldsymbol{m}$ obtained via the above procedure as the $\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})$ and find the MAP points.
```

Algorithm 4 The Response-surface-based Bayesian Inference using the M-H Algorithm

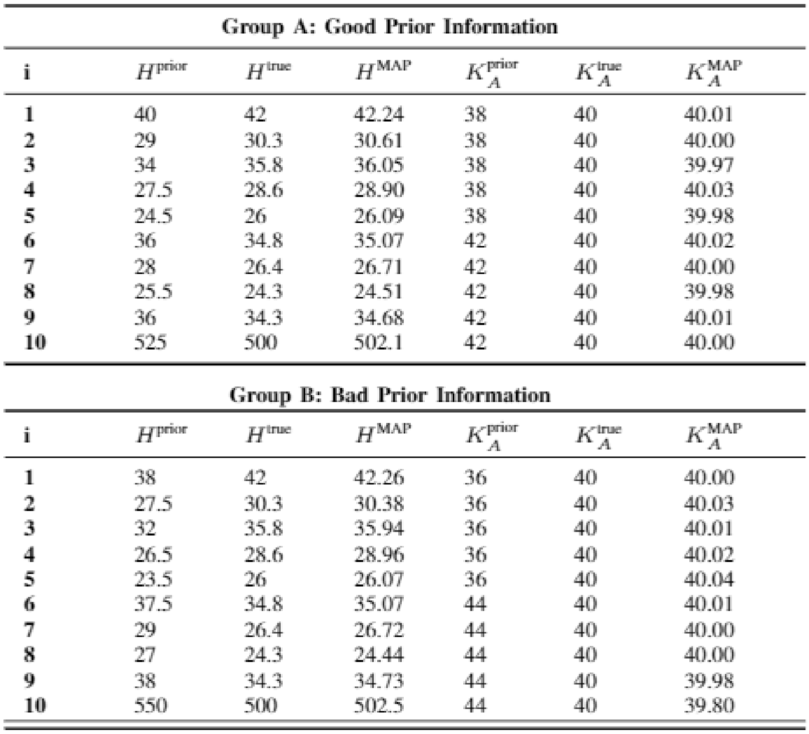
```
Choose the initial guess of the parameters $\boldsymbol{m_0}$ from the manufactured data as the Bayesian prior $\boldsymbol{m}_{\text{prior}}$;
Build the PCE surrogates as the response surface of the decentralized dynamic model;
Compute $\log(\pi_{\text{post}}(\boldsymbol{m_0}|\boldsymbol{d}))$ from the PCE surrogate via (4.41);
for $k = 0, \dots, N_{\text{samples}} - 1$ do
    Generate new sample $\boldsymbol{m_{k+1}}$ from $q(\boldsymbol{m_k}, \cdot)$;
    Compute $\log(\boldsymbol{\pi_{\text{post}}}(\boldsymbol{m_{k+1}}|\boldsymbol{d}))$ from the PCE surrogate via (4.41);
    Calculate the correction factor $c = \frac{q(\boldsymbol{m_{k+1}}, \boldsymbol{m_k})}{q(\boldsymbol{m_k}, \boldsymbol{m_{k+1}})}$;
    Compute $\alpha(\boldsymbol{m_k}, \boldsymbol{m_{k+1}})$ defined as
        $$\alpha(\boldsymbol{m_k}, \boldsymbol{m_{k+1}}) = \log(\min\{1, \frac{\pi_{\text{post}}(\boldsymbol{m_{k+1}}|\boldsymbol{d})}{\pi_{\text{post}}(\boldsymbol{m_k}|\boldsymbol{d})} \cdot c\});$$
    Draw $u \sim \mathcal{U}([0, 1))$;
    if $\log(u) < \alpha(\boldsymbol{m_k}, \boldsymbol{m_{k+1}})$ then
        Accept: Set $\boldsymbol{m}_{k+1} = \boldsymbol{m_{k+1}}$;
    else
        Reject: Set $\boldsymbol{m}_{k+1} = \boldsymbol{m}_k$;
    end if
end for
Plot the PDF, $\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})$, and find the MAP points.
```

4.3.2 Case Studies

Case studies are conducted on the New England system using the aforementioned two-axis generator model with the IEEE-DC1A exciter and the TGOV1 turbine governor as described in [88]. Parameter estimation studies are conducted to evaluate the accuracy, the calculation efficiency, the impact of the Bayesian prior knowledge, the high dimension performance, and the applicability of the methods in different realistic scenarios.

Studies on the Prior Information The posterior distribution of the Bayesian inference is determined by the likelihood function and the prior PDF. For the practice implementation, the prior PDF of the parameters can be chosen using the data provided by the manufacturer, which is considered to be reasonable in general. However, the errors are still inevitable. This study analyzes the estimation accuracy under different prior information. Suppose the parameters to be estimated are the moment of inertia H and the amplifier gain K_A in the exciter. The errors from the original data provided by the manufacturer for every generators are assumed to have about 5% deviations from of their true values in Group A and 10% for the Group B. These two parameters are assumed to follow the Gaussian distributions with their means equal to the manufacturer's data and the standard deviations being set to 10% of the means to account for the parameter uncertainties. The random vector q is assumed to follow a multivariate Gaussian probability distribution with zero means and standard deviations equal to 10% of the original values; $\boldsymbol{e}$ are assumed to be identical, independent Gaussian noise with 0.01 standard deviation. The transmission line between Bus 19 and Bus 33 is removed after 0.5 seconds. The time interval is selected as 3 seconds. The parameter estimation is conducted separately for every generator in a decentralized manner. The sample size for the M-H sampler is set to $100,000$. We choose the Hermite polynomials as surrogate simulators for the Bayesian Inference. Our proposed method completes the calculation for every single generator in just 6s while the traditional method takes around 29 minutes, thus achieving two orders of magnitude in speed. The simulation results from the proposed response-surface-based method are given in Table 4.5. It shows that the MAP estimates for both groups match the true values quite well. We also observe that the prior PDF of the parameters has a negligible influence on the accuracy of the MAP estimates. This is because, for every generator, the sum of the log likelihoods given by (4.41) that are associated with just a few uncertain parameters and are provided by 360 observations in total are negligible. Evidently, the fast sampling rate of the PMUs has greatly decreased the estimation error brought by an inappropriate prior knowledge. Given a good prior information, the estimation algorithm converges faster.

Table 4.5: Comparison of the MAPs with the Actual Values for the ith Machines under Different Prior Information



Group A: Good Prior Information

i	H^{prior}	H^{true}	H^{MAP}	K_A^{prior}	K_A^{true}	K_A^{MAP}
1	40	42	42.24	38	40	40.01
2	29	30.3	30.61	38	40	40.00
3	34	35.8	36.05	38	40	39.97
4	27.5	28.6	28.90	38	40	40.03
5	24.5	26	26.09	38	40	39.98
6	36	34.8	35.07	42	40	40.02
7	28	26.4	26.71	42	40	40.00
8	25.5	24.3	24.51	42	40	39.98
9	36	34.3	34.68	42	40	40.01
10	525	500	502.1	42	40	40.00

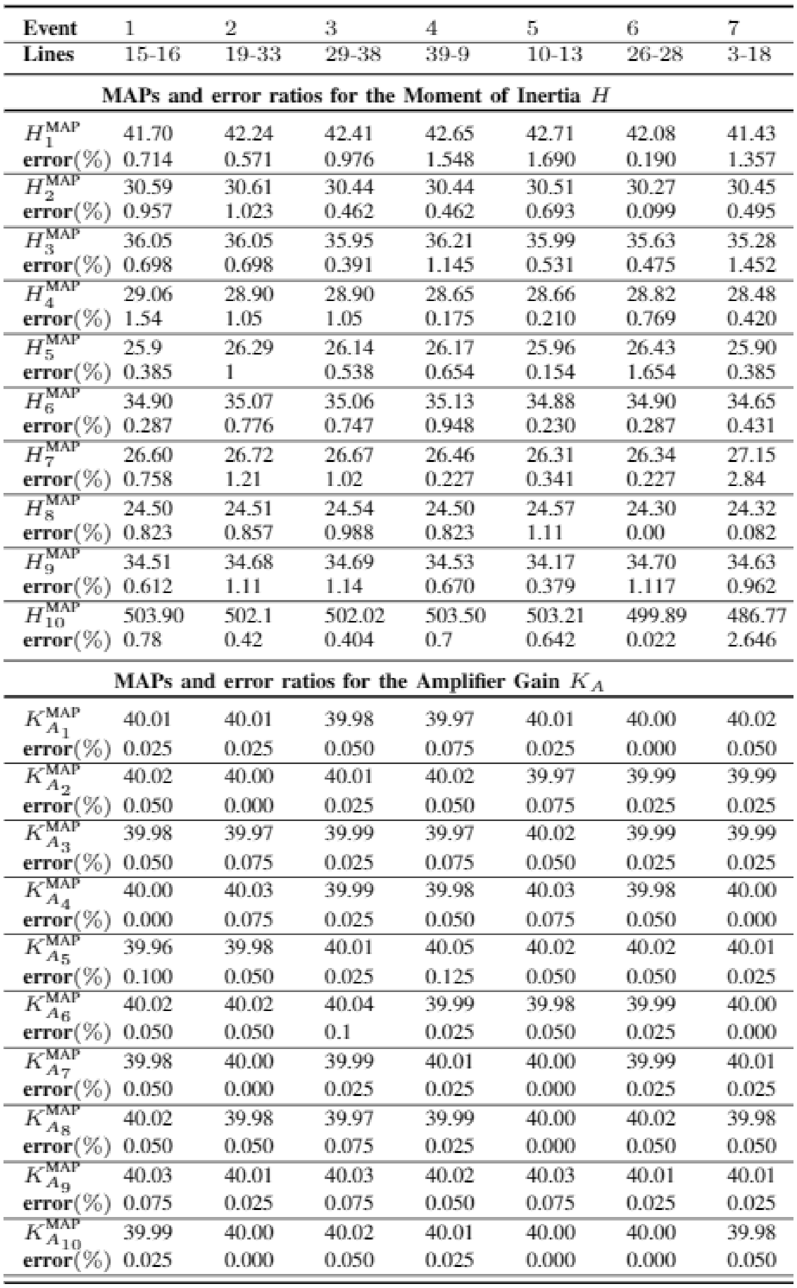
Group B: Bad Prior Information

i	H^{prior}	H^{true}	H^{MAP}	K_A^{prior}	K_A^{true}	K_A^{MAP}
1	38	42	42.26	36	40	40.00
2	27.5	30.3	30.38	36	40	40.03
3	32	35.8	35.94	36	40	40.01
4	26.5	28.6	28.96	36	40	40.02
5	23.5	26	26.07	36	40	40.04
6	37.5	34.8	35.07	44	40	40.01
7	29	26.4	26.72	44	40	40.00
8	27	24.3	24.44	44	40	40.00
9	38	34.3	34.73	44	40	39.98
10	550	500	502.5	44	40	39.80

Validation under Different Events Several papers in the literature discuss the importance of validating the proposed methods under different events [74], [75], [85] since the uncertain parameters can lead to very different model output for different cases. To verify this result, we create seven different events by removing seven different transmission lines. All the other settings are the same with Group A described in the previous case study. The MAP estimates and their corresponding error rates in percentages obtained by our proposed response surface method are displayed in Table 4.6. From the simulation results, we observe that our proposed method yields good accuracy in calculating the MAP parameter estimates for every generator in most of the cases under consideration. However, for Event 7, considering the estimation results for the moment of the inertia H, there are two cases with an error rate higher than 2%. This is because H mainly influences the active power output P while K_A mainly influences the reactive power Q. For Event 7, the active power output has a much smaller variation compared with the variation in the reactive power output. Under this circumstance, with different H values proposed in the M-H sampler, the trajectories of P do not change much, and the corresponding likelihood functions vary little as a result. This leads to potential inaccuracy in the MAP estimates. Luckily, this inaccuracy can be self-detected from the posterior distribution of the Bayesian inference. Unlike the WLS-based estimator that only provides point estimates for the generator parameters, the Bayesian method provides their probability distributions. For example, let us plot the PDFs of the H_2 for Event 3 and Event 7 and the PDF for the H_7 for Event 7, respectively, in Fig. 4.11. Comparing Fig. 4.11.(a) and 4.11.(b), we find that, for H_2, the confidence interval provided by the PDFs under different events can be very different. For Event 3, the PDF mainly covers the range of $30.2-30.6$, which is much smaller than that provided by Event 7. This means Event 3 provides a more accurate parameter estimation than that of Event 7, even though the MAP estimators under these two events are very similar. Comparing Fig. 4.11.(b) and 4.11.(c), we find that, under the same Event 7, the PDFs for both H_2 and H_7 estimate larger confidence intervals. Even though the MAP estimator of H_2 is accurate, it can not always be guaranteed. This is demonstrated in 4.11.(c) that the MAP estimate of H_7 is not accurate. This means Event 7 is not an appropriate event to estimate the parameter H. When obtaining estimation results with large confidence intervals, we should also test other events to provide more reliable results. These scenarios have also been discussed in [74]. The Bayesian posterior distribution gives us a good guidance in distinguishing these inappropriate events.

Studies in Higher Dimension In this part, we are demonstrating the performance of the proposed method with six key parameters that are known to influence the dynamic response of the system and can not be directly measured easily. These are the moment of inertia H, three gains in the exciter, namely, K_A, K_E, K_F, the damping ratio D, and the droop R_D. Let

Table 4.6: Comparison of the MAPs with the Actual Values for the Transmission Lines Between the Different Buses removed for All the Generators



Event	1	2	3	4	5	6	7
Lines	15-16	19-33	29-38	39-9	10-13	26-28	3-18
MAPs and error ratios for the Moment of Inertia H							
H_1^{MAP}	41.70	42.24	42.41	42.65	42.71	42.08	41.43
error(%)	0.714	0.571	0.976	1.548	1.690	0.190	1.357
H_2^{MAP}	30.59	30.61	30.44	30.44	30.51	30.27	30.45
error(%)	0.957	1.023	0.462	0.462	0.693	0.099	0.495
H_3^{MAP}	36.05	36.05	35.95	36.21	35.99	35.63	35.28
error(%)	0.698	0.698	0.391	1.145	0.531	0.475	1.452
H_4^{MAP}	29.06	28.90	28.90	28.65	28.66	28.82	28.48
error(%)	1.54	1.05	1.05	0.175	0.210	0.769	0.420
H_5^{MAP}	25.9	26.29	26.14	26.17	25.96	26.43	25.90
error(%)	0.385	1	0.538	0.654	0.154	1.654	0.385
H_6^{MAP}	34.90	35.07	35.06	35.13	34.88	34.90	34.65
error(%)	0.287	0.776	0.747	0.948	0.230	0.287	0.431
H_7^{MAP}	26.60	26.72	26.67	26.46	26.31	26.34	27.15
error(%)	0.758	1.21	1.02	0.227	0.341	0.227	2.84
H_8^{MAP}	24.50	24.51	24.54	24.50	24.57	24.30	24.32
error(%)	0.823	0.857	0.988	0.823	1.11	0.00	0.082
H_9^{MAP}	34.51	34.68	34.69	34.53	34.17	34.70	34.63
error(%)	0.612	1.11	1.14	0.670	0.379	1.117	0.962
H_{10}^{MAP}	503.90	502.1	502.02	503.50	503.21	499.89	486.77
error(%)	0.78	0.42	0.404	0.7	0.642	0.022	2.646
MAPs and error ratios for the Amplifier Gain K_A							
$K_{A_1}^{\text{MAP}}$	40.01	40.01	39.98	39.97	40.01	40.00	40.02
error(%)	0.025	0.025	0.050	0.075	0.025	0.000	0.050
$K_{A_2}^{\text{MAP}}$	40.02	40.00	40.01	40.02	39.97	39.99	39.99
error(%)	0.050	0.000	0.025	0.050	0.075	0.025	0.025
$K_{A_3}^{\text{MAP}}$	39.98	39.97	39.99	39.97	40.02	39.99	39.99
error(%)	0.050	0.075	0.025	0.075	0.050	0.025	0.025
$K_{A_4}^{\text{MAP}}$	40.00	40.03	39.99	39.98	40.03	39.98	40.00
error(%)	0.000	0.075	0.025	0.050	0.075	0.050	0.000
$K_{A_5}^{\text{MAP}}$	39.96	39.98	40.01	40.05	40.02	40.02	40.01
error(%)	0.100	0.050	0.025	0.125	0.050	0.050	0.025
$K_{A_6}^{\text{MAP}}$	40.02	40.02	40.04	39.99	39.98	39.99	40.00
error(%)	0.050	0.050	0.1	0.025	0.050	0.025	0.000
$K_{A_7}^{\text{MAP}}$	39.98	40.00	39.99	40.01	40.00	39.99	40.01
error(%)	0.050	0.000	0.025	0.025	0.000	0.025	0.025
$K_{A_8}^{\text{MAP}}$	40.02	39.98	39.97	39.99	40.00	40.02	39.98
error(%)	0.050	0.050	0.075	0.025	0.000	0.050	0.050
$K_{A_9}^{\text{MAP}}$	40.03	40.01	40.03	40.02	40.03	40.01	40.01
error(%)	0.075	0.025	0.075	0.050	0.075	0.025	0.025
$K_{A_{10}}^{\text{MAP}}$	39.99	40.00	40.02	40.01	40.00	40.00	39.98
error(%)	0.025	0.000	0.050	0.025	0.000	0.000	0.050

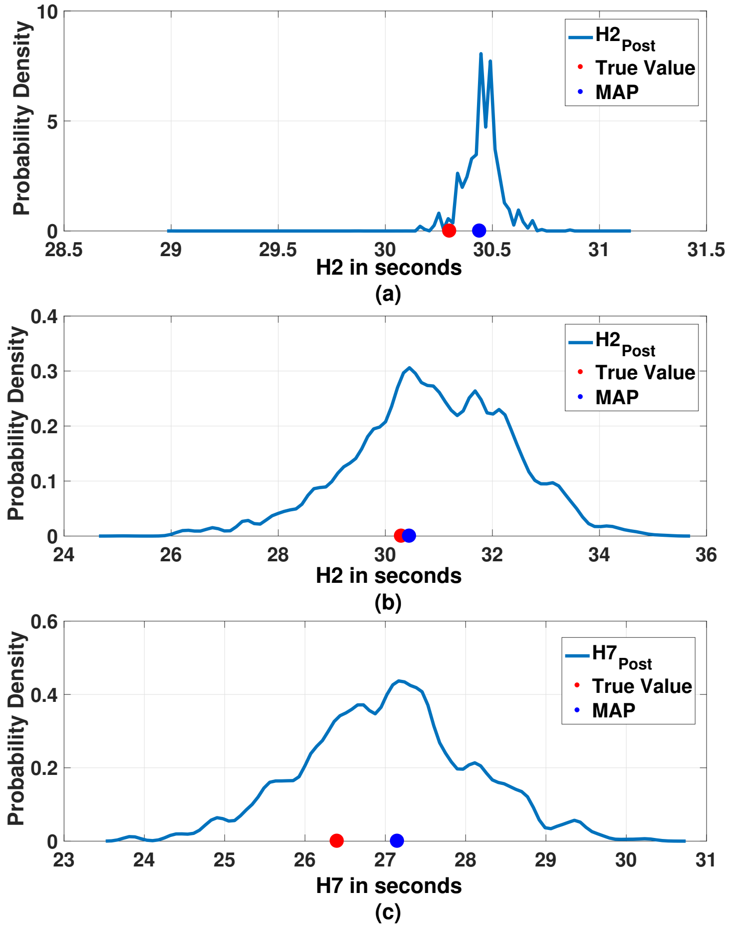


Fig. 4.11. Plots for the PDFs, the true values and the MAPs: (a) for $H1$ under Event 3; (b) for $H1$ under event 7; (c) for $H7$ under Event 7.

us take Generator 5 as an example. These six parameters are assumed to follow Gaussian distributions with means being the original data provided by the manufacturer, that is, $\{26, 40, 82.5, 1, 1, 0.067\}$ and standard deviations being 10% of the means to account for the parameter uncertainties. The disturbances are caused by Event 2. The simulation results are shown in Fig. 4.12. It is found that the proposed method yields good accuracy of the MAP estimates, which are equal to $\{26.07, 39.9, 82.7, 1.014, 0.999, 0.0634\}$. Even though there are several peaks regarded as local optima, the M-H sampler still provides the global optima, namely the MAP estimates. However, the calculation efficiency is decreased. To guarantee the accuracy, the sample number is increased to 1,000,000. With our proposed response-surface-based method, the calculation can still be completed within 2 minutes. By contrast, traditional method will take about 5 hours. This further demonstrates the high efficiency of our proposed method.

Effective Dimension Analysis Not all the parameters can be analyzed by the proposed method due to the observability limitation [89]. Some parameters have negligible influences on the output that makes the accurate prediction impossible. To address this issue, we propose to use the ANOVA index to screen out the key model parameters. The ANOVA expansion represents a function f with N random variable in the form of as

$$\begin{aligned} f(\xi_1, \ldots, \xi_N) &= f_0 + \sum_{1 \le j_1 \le N} f_{j_1}(\xi_{j_1}) \\ &+ \sum_{1 \le j_1 < j_2 \le N} f_{j_1, j_2}(\xi_{j_1}, \xi_{j_2}) + \ldots + f_{1,2,\ldots,N}. \end{aligned} \tag{4.42}$$

The functions $f_{j_k}(\xi_{j_k})$, $1 \le j_k \le N$, $f_{j_k, j_l}(\xi_{j_k}, \xi_{j_l})$, $1 \le j_k \le j_l \le N$, etc., are called the first-order, second-order,... ANOVA components, respectively. Let $T_{j_1,\ldots,j_s}$ denote the fraction of the variance $\sigma^2(f)$ that is contributed by $f_{j_1,\ldots,j_s}$, which is defined as

$$T_{j_1,\ldots,j_s} = \frac{\sigma^2(f_{j_1,\ldots,j_s})}{\sigma^2(f)}. \tag{4.43}$$

Here T_{j_1} denotes the variance contributed by the single variable ξ_{j_1}, T_{j_1, j_2} denotes the variance contributed by the coupling effect of the two random inputs, (ξ_{j_1}, ξ_{j_2}), and so on for the higher-order terms. Sudret [94] proposes

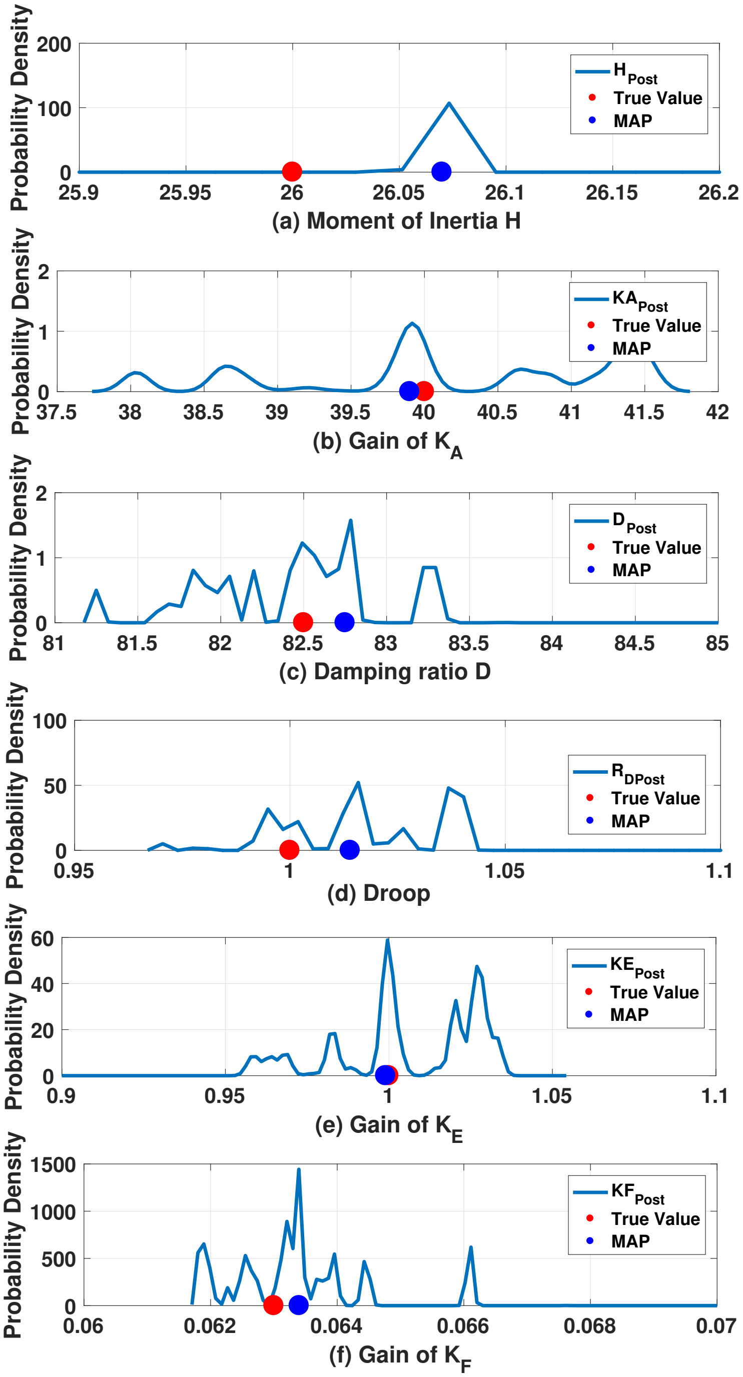
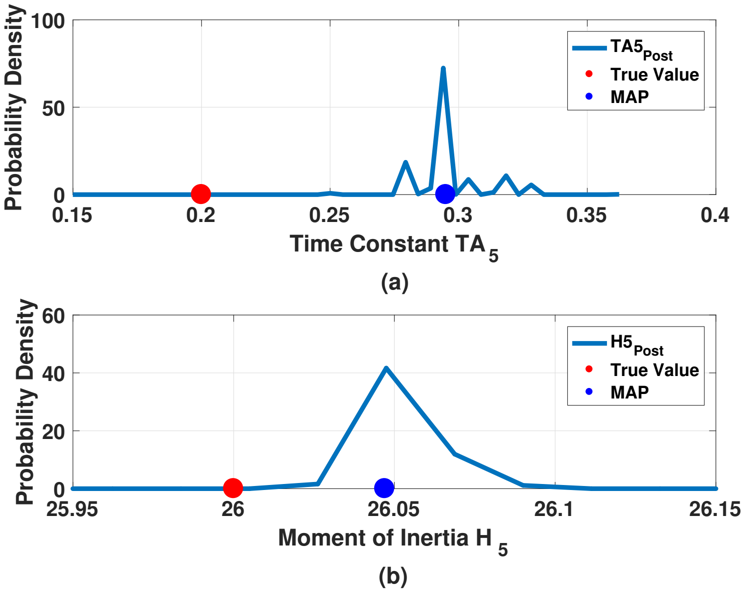


Fig. 4.12. Plots for the pdfs, the true values and the MAPs: (a) for moment of inertia H; (b) for Gain KA; (c) for Damping ratio D;(d) for Droop R_D; (e) for Gain KE; (f) for Gain K_F.



(a)

(b)

Fig. 4.13. Plots for the pdfs, the true values and the MAPs: (a) for Time constant T_A; (b) for moment of inertia H.

the ANOVA index under the PCE framework as follows:

$$T_{j_1} = \frac{\sigma^2(f_{j1)}}{\sigma^2(f)} = \frac{a_i^2 E[\phi(\xi_i)^2] + a_{i,j}^2 E[\phi(\xi_i, \xi_j)^2]_{i=j}}{\sigma^2}, \tag{4.44}$$

$$T_{j_1,j_2} = \frac{\sigma^2(f_{j_1,j_2})}{\sigma^2(f)} = \frac{a_{i,j}^2 E[\phi(\xi_i, \xi_j)^2]_{i\neq j}}{\sigma^2}. \tag{4.45}$$

Let us take the estimation of the time constant T_{A5} and the moment of inertia H_5 for Generator 5 under Event 2 as an example. The manufacturer's data for T_{A5} and H_5 are $\{0.3, 28\}$ while the true values are $\{0.2, 26\}$. As observed from the simulation results displayed in Fig. 4.13, the MAP estimate of H_5 is accurate while the MAP estimate of T_{A5} has almost no shift from the original data. If we calculate the ANOVA index after Step 2 in Algorithm 2, we find that the averaged T_{j_2} for the given time period for H_5 equals to 99.99% for the active power and the averaged T_{j_2} for H_5 equals to 99.6% for the reactive power. This means that T_{j_1} for T_{A5} has a much smaller value compared with T_{j_2}. The variation in the output is dominated by the H_5. Here we call T_{A5} an ineffective parameter that cannot be estimated by the proposed method.

In this work, a response-surfaced-based Bayesian inference via MCMC for power system dynamic parameter estimation with full pdfs is proposed. The Bayesian inference for the decentralized power system model is first formulated. The use of the MCMC makes this method applicable to non-Gaussian distributions. The PCE surrogate is further built to speed up the MCMC. Simulation results obtained on the New England system demonstrate the good performance of the proposed method for credible power system dynamic security analysis. Future work will be focused on further improving the convergence rate and robustness of the proposed method.

Chapter 5

Stochastic Seismic Inversion

5.1 Classical adjoint tomographic inversion

A classical adjoint tomographic inverse approach for seismic problems relies on full waveform simulations and back projection to compute finite frequency sensitivity kernels. These kernels describe the variation of the discrepancy, or misfit, between observed seismic data and modeled synthetic wavefoms as a function of the model parameters at each model intgeration points. Then they are used in an iterative fashion to minimize the misfit function to constrain the model parameters, thereby recovering medium properties.

We can consider a least-squares full waveform misfit function, in order to measure how well the data and corresponding synthetics fit. The waveform misfit writes as

$$\chi(\mathbf{m}) = \frac{1}{2} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \int_0^T ||\mathbf{s}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{m}, t) - \mathbf{d}(\mathbf{x}_r, \mathbf{x}_s, t)||^2 \,\mathrm{dt} \tag{5.1}$$

where $\mathbf{d}$ and $\mathbf{s}$ denote the observed data and the corresponding synthetic displacements, respectively. The model vector $\mathbf{m}$ refers to any combination of parameters defining the model, e.g., density ρ, bulk modulus κ, and shear modulus μ, or density ρ, compressional wave speed α, and shear wave speed β. N_s and N_r are the number of sources and receivers, respectively, while $\mathbf{x}_r$ and $\mathbf{x}_s$ refer to the position of receivers and sources, respectively.

Changes in the misfit function, $\delta\chi$, may be expressed in terms of, e.g., relative changes in density $\delta \ln \rho = \delta\rho/\rho$, bulk modulus $\delta \ln \kappa = \delta\kappa/\kappa$, and shear modulus $\delta \ln \mu = \delta\mu/\mu$ as

$$\delta\chi = \int (K_\rho \delta \ln \rho + K_\kappa \delta \ln \kappa + K_\mu \delta \ln \mu)\, \mathrm{d}^3\mathbf{x} \tag{5.2}$$

The associated sensitivity kernels, or Fréchet derivatives are defined as

$$K_\rho(\mathbf{x}) = \int_0^T \rho(\mathbf{x}) \partial_t \mathbf{s}^\dagger(\mathbf{x}, T-t) \cdot \partial_t \mathbf{s}(\mathbf{x}, T-t)\, \mathrm{dt}\,, \tag{5.3}$$

$$K_\kappa(\mathbf{x}) = -\int_0^T \kappa(\mathbf{x}) \nabla \cdot \mathbf{s}^\dagger(\mathbf{x}, T-t) \nabla \cdot \mathbf{s}(\mathbf{x}, T-t)\, \mathrm{dt}\,, \tag{5.4}$$

$$K_\mu(\mathbf{x}) = -\int_0^T 2\mu(\mathbf{x}) \mathbf{D}^\dagger(\mathbf{x}, T-t) \,:\, \mathbf{D}(\mathbf{x}, T-t)\, \mathrm{dt}\,, \tag{5.5}$$

where $\mathbf{D} = \frac{1}{2}[\nabla\mathbf{s} + (\nabla\mathbf{s})^T] - \frac{1}{3}(\nabla \cdot \mathbf{s})\mathbf{I}$ and $\mathbf{D}^\dagger$ are the strain deviator and its adjoint, respectively. $\mathbf{s}^\dagger$ denotes the adjoint wavefield solution of the wave equation driven by the source term

$$\mathbf{f}^\dagger(\mathbf{x}, t) = \sum_{r=1}^{N_r} [\mathbf{s}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{m}, T-t) - \mathbf{d}(\mathbf{x}_r, \mathbf{x}_s, T-t)]\, \delta(\mathbf{x} - \mathbf{x}_r)\,, \tag{5.6}$$

where all the receivers act as simultaneous sources.

Chapter 6

Codes

Folder structure:

My version of the code contains following folders:

- Src: KPCA class
- Src-adj: MCMC, Adjoint modules, ADAM and Gradient-decent methods
- test: test problems for the elasticity-inversion, powergrid and sesimic-inversion
- Makefile: Make file
- ADFirstAidKit, Data, Include, Lib, README: Other supporting files and libraries

Compiling DASSI:

To compile DASSI go into the main directory and issue "make" command in the linux/unix shell, it will create *libdassi.a*, *libdassi_wz.a*, *liblapack.a*, *libnewuoa.a* libraries.

General notes on the test cases:

- The test cases are located on test directory, to compile the example issue "*make*" command in the linux/unix shell.

- Each example directory contains "makefile.inc" that specifies the required library files locations.
- You can submit the run on cab using *msubscript_msub*. Yo may edit the numbers of nodes (=1), partition(=cab) and walltime(=16:00:00) in this file to suit your needs.
- If the make command creates any error make sure that lines 5 (*dassi_path*=../../../Src), 6 (*dassi_wz_path* =../../../Src-adj) and 23 (23 *Eigen*3*_Include* = -I../../../Include/Eigen3.2.9/include/eigen3) of "makefile.inc" are pointing to write directories.
- Default command to run the forward solver is *"python wrapper.py"*, edit the codes wrapper.py to suit your needs.

Elasticity inversion:

- Forward solver: wrapper.py
- Input snapshots: log of the original snapshots to avoid negative values to the elasticity parameters
- By default dassi assumes, it has given log snapshots and takes exponential before generating inputs to the forward solver

Examples:

- check-kpca : KPCA kernel choosing
- check-kpca-snap3 :KPCA for snap3 with polynomial kernel of order 5
- jcp-plots : JCP paper plots
- langevin-mcmc-dim-20 : LHMCMC with chain starting at 0
- langevin-mcmc-dim-20-eta-2: LHMCMC with chain starting at 2
- langevin-mcmc-dim-20-eta-ve2: LHMCMC with chain starting at -2
- langevin-mcmc-dim-20-good-initial-guess : Quick check of LHMCMC
- mcmc-dim-20 :MCMC with chain starting at 0

- mcmc-dim-20-eta-2: MCMC with chain starting at 2
- mcmc-dim-20-eta-ve2: MCMC with chain starting at -2
- mcmc-dim-20-good-initial-guess: Quick check of MCMC
- langevin-mcmc-dim-20-pca : LHMCMC with pca

Powergrid state estimation:

- Forward solver: wrapper.py which executes the command *matlab -nodesktop -nojvm -nosplash -r "H_function_program; exit"* and produces *error.txt* that contains norm of the error.
- Snapshots are generated by solving WLS multiple time via adding uniform noise to measurements (WLS_DSE_main_program_samples_uniform.m)
- Powergrid states can be negative values and we cannot provide log snapshots. A function "*genSnapshotFromStdRV(VectorXd &Eigen_xiVec, VectorXd &Eigen_snapVec, int noexp)"* is added to *"sReduceModelviaKPCA"* module that facilitate dassi to read true snapshots instead of log snapshots
- To reflect above changes in powergrid MCMC, the functions "powergrid_likelihood_pdf" and "powergrid_posterior_pdf" are added to AdjointsRMvKPCA module and "powergrid_target_density_function" and "powergrid_mcmc_run" are added to MetropolisHastingsMCMC module.
- "plot_results.py" can be used to plot the final results and generate BoxPlots and posterior state histograms

Examples

- gaussian-lik-no-weight: MCMC with Gaussian error model with no weight matrix
- gaussian-lik-no-weight: MCMC with Gaussian error model with weight matrix provided by Can Huang

Seismic inversion:

Running the code:

- step 1: bash rem (removes old files, run this command only if you are starting from iteration 0)
- step 2: make cleanall (removes old compiled files)
- step 3: make (compiles the code)
- step 4: msub script_msub (submit the job on cab)

Results:

- all_Snapshots.txt: Contains Vp andVs solutions from all the runs
- all_etaSnapshots.txt: Contains solutions from all the runs in the feature space
- all_misfits.txt: Contains objective function values from all the runs
- etaSnapshot: feature values from the last run, useful if you are rerunning the job
- etaSnapshot: Vp, Vs values from the last run, useful if you are rerunning the job
- plot_snap.pyplot_snap_0.py: to plot the results
- Forward solver: wrapper.py
- Goal of the problem is to perform deterministic optimization, "FuncDetOpt.cpp" contains corresponding functions
- Adaptive Moment Estimation (ADAM) optimization method is used to solve the optimization problem
- Optimization is carried out on ξ space (default is η space) and corresponding gradient computation function "sesimic_grad_pdf" is added to AdjointsRMvKPCA module
- Snapshot mesh is different from the Specfem mesh i.e., we need to convert

  - VP, Vs value on the snapshot mesh→ VP, Vs value on t specfem mesh (Chris' module)
  - gradients from the snapshot mesh ⇐ gradients on the snapshot mesh (performed via nearest neighborhood interpolation using pyhton code at GtoMnMtoG/gridToModel.py)

- In each iteration Vp and Vs values are limited (similar to gradient clipping) using GtoMnMtoG/ modelToGrid_limit.py

Acknowledgment

This work was funded by the laboratory directed research and development (LDRD; 16-ERD-023) program and conducted under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-7NA27344.

Bibliography

[1] A. Anandarajah, Computational methods in elasticity and plasticity: solids and porous media, Springer Science & Business Media, 2011.

[2] B. Kirtman, S. Power, A. Adedoyin, G. Boer, R. Bojariu, I. Camilloni, F. Doblas-Reyes, A. Fiore, M. Kimoto, G. Meehl, et al., Near-term climate change: projections and predictability.

[3] G. Dagan, S. P. Neuman, Subsurface flow and transport: a stochastic approach, Cambridge University Press, 2005.

[4] B. Kennett, Seismic wave propagation in stratified media, ANU Press, 2013.

[5] R. W. Graves, Simulating seismic wave propagation in 3d elastic media using staggered-grid finite differences, Bulletin of the Seismological Society of America 86 (4) (1996) 1091–1106.

[6] P. Kundur, N. J. Balu, M. G. Lauby, Power system stability and control, Vol. 7, McGraw-hill New York, 1994.

[7] A. Tarantola, Inverse problem theory and methods for model parameter estimation, SIAM, 2005.

[8] J. Martin, L. C. Wilcox, C. Burstedde, O. Ghattas, A stochastic newton mcmc method for large-scale statistical inverse problems with application to seismic inversion, SIAM Journal on Scientific Computing 34 (3) (2012) A1460–A1487.

[9] P. J. Green, A. Mira, Delayed rejection in reversible jump metropolis-hastings, Biometrika (2001) 1035–1053.

[10] A. Mira, Ordering and improving the performance of monte carlo markov chains, Statistical Science (2001) 340–350.

[11] H. Haario, E. Saksman, J. Tamminen, Adaptive proposal distribution for random walk metropolis algorithm, Computational Statistics 14 (3) (1999) 375–396.

[12] H. Haario, E. Saksman, J. Tamminen, An adaptive metropolis algorithm, Bernoulli (2001) 223–242.

[13] L. Tierney, A. Mira, Some adaptive monte carlo methods for bayesian inference, Statistics in medicine 18 (1718) (1999) 2507–2515.

[14] G. O. Roberts, J. S. Rosenthal, Optimal scaling of discrete approximations to langevin diffusions, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60 (1) (1998) 255–268.

[15] H. Haario, M. Laine, A. Mira, E. Saksman, Dram: efficient adaptive mcmc, Statistics and computing 16 (4) (2006) 339–354.

[16] M. Parno, Y. Marzouk, Transport map accelerated markov chain monte carlo, arXiv preprint arXiv:1412.5492.

[17] R. G. Ghanem, P. D. Spanos, Stochastic finite elements: a spectral approach (2003).

[18] Y. Marzouk, D. Xiu, A stochastic collocation approach to bayesian inference in inverse problems.

[19] Y. M. Marzouk, H. N. Najm, L. A. Rahn, Stochastic spectral methods for efficient bayesian solution of inverse problems, Journal of Computational Physics 224 (2) (2007) 560–586.

[20] N. Bliznyuk, D. Ruppert, C. A. Shoemaker, Local derivative-free approximation of computationally expensive posterior densities, Journal of Computational and Graphical Statistics 21 (2) (2012) 476–495.

[21] V. R. Joseph, Bayesian computation using design of experiments-based interpolation technique, Technometrics 54 (3) (2012) 209–225.

[22] C. E. Rasmussen, Gaussian processes for machine learning.

[23] K.-I. Funahashi, On the approximate realization of continuous mappings by neural networks, Neural networks 2 (3) (1989) 183–192.

[24] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359–366.

[25] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: International Conference on Artificial Neural Networks, Springer, 1997, pp. 583–588.

[26] P. Sarma, L. J. Durlofsky, K. Aziz, Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics, Mathematical Geosciences 40 (1) (2008) 3–32.

[27] X. Ma, N. Zabaras, Kernel principal component analysis for stochastic input model generation, J. Comput. Phys. 230 (19) (2011) 7311–7331.

[28] S. Strebelle, Conditional simulation of complex geological structures using multiple-point statistics, Mathematical Geology 34 (1) (2002) 1–21.

[29] R. A. Adams, J. J. F. Fournier, Sobolev spaces, 2nd Edition, Vol. 140 of Pure and Applied Mathematics (Amsterdam), Elsevier/Academic Press, Amsterdam, 2003.

[30] A. A. Oberai, N. H. Gokhale, G. R. Feijóo, Solution of inverse problems in elasticity imaging using the adjoint method, Inverse problems 19 (2) (2003) 297.

[31] R. Courant, D. Hilbert, Methods of mathematical physics, Vol. 1, CUP Archive, 1966.

[32] W. H. Press, Numerical recipes 3rd edition: The art of scientific computing, Cambridge university press, 2007.

[33] C. M. Bishop, Pattern recognition, Machine Learning 128 (2006) 1–58.

[34] N. Cressie, The origins of kriging, Mathematical geology 22 (3) (1990) 239–252.

[35] E. H. Isaaks, et al., Applied geostatistics, Tech. rep., Oxford University Press (1989).

[36] G. Matheron, Principles of geostatistics, Economic geology 58 (8) (1963) 1246–1266.

[37] C. A. Thimmisetty, R. G. Ghanem, J. A. White, X. Chen, High-dimensional intrinsic interpolation using gaussian process regression and diffusion maps, Mathematical Geosciences.

[38] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

[39] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997, pp. 583–588.

[40] I. T. Jolliffe, Principal component analysis, 2nd Edition, Springer Series in Statistics, Springer-Verlag, New York, 2002.

[41] J. T. Kwok, I. W. Tsang, The pre-image problem in kernel methods, in: ICML, 2003, pp. 408–415.

[42] R. Lebrun, A. Dutfoy, A generalization of the nataf transformation to distributions with elliptical copula, Probabilistic Engineering Mechanics 24 (2) (2009) 172–178.

[43] M. Rosenblatt, Remarks on a multivariate transformation, The annals of mathematical statistics 23 (3) (1952) 470–472.

[44] R. G. Ghanem, A. Doostan, On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data, Journal of Computational Physics 217 (1) (2006) 63–81.

[45] G. Stefanou, A. Nouy, A. Clement, Identification of random shapes from images through polynomial chaos expansion of random level set functions, International Journal for Numerical Methods in Engineering 79 (2) (2009) 127–155.

[46] N. Wiener, The homogeneous chaos, American Journal of Mathematics 60 (4) (1938) 897–936.

[47] M. Arnst, R. Ghanem, C. Soize, Identification of bayesian posteriors for coefficients of chaos expansions, Journal of Computational Physics 229 (9) (2010) 3134–3154.

[48] M. Eldred, J. Burkardt, Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification, AIAA paper 976 (2009) (2009) 1–20.

[49] G. Stefanou, A. Nouy, A. Clement, Identification of random shapes from images through polynomial chaos expansion of random level set functions, International Journal for Numerical Methods in Engineering 79 (2) (2009) 127–155.

[50] J. L. Jin Qin, Empirical likelihood and general estimating equations, The Annals of Statistics 22 (1) (1994) 300–325.

[51] M. Jones, The performance of kernel density functions in kernel distribution function estimation, Statistics & Probability Letters 9 (2) (1990) 129–132.

[52] R. Giering, T. Kaminski, Recipes for adjoint code construction, ACM Transactions on Mathematical Software (TOMS) 24 (4) (1998) 437–474.

[53] L. Hascoët, V. Pascual, The Tapenade Automatic Differentiation tool: Principles, Model, and Specification, ACM Transactions On Mathematical Software 39 (3).

[54] C. Thimmisetty, A. Khodabakhshnejad, N. Jabbari, F. Aminzadeh, R. Ghanem, K. Rose, J. Bauer, C. Disenhof, Multiscale stochastic representation in high-dimensional data using gaussian processes with implicit diffusion metrics, in: Dynamic Data-Driven Environmental Systems Science, Springer International Publishing, 2015, pp. 157–166.

[55] H. Gharavi, B. Hu, Synchrophasor sensor networks for grid communication and protection, Proc. IEEE 105 (7) (2017) 1408–1428.

[56] Y. P. Agalgaonkar, D. J. Hammerstrom, Evaluation of smart grid technologies employed for system reliability improvement: Pacific northwest smart grid demonstration experience, IEEE Power and Energy Technology Systems Journal 4 (2) (2017) 24–31. doi:10.1109/JPETS.2017.2683502.

[57] M. Esmaili, M. Ghamsari-Yazdel, Voltage stability-constrained optimal simultaneous placement of pmus and channels enhancing measurement

reliability and redundancy, IEEE Power and Energy Technology Systems Journal 4 (2) (2017) 32–39.

[58] V. Murugesan, Y. Chakhchoukh, V. Vittal, G. T. Heydt, N. Logic, S. Sturgill, Pmu data buffering for power system state estimators, IEEE Power and Energy Technology Systems Journal 2 (3) (2015) 94–102. doi:10.1109/JPETS.2015.2448115.

[59] C. Huang, F. Li, L. Zhan, Y. Xu, Q. Hu, D. Zhou, Y. Liu, Data quality issues for synchrophasor applications part ii: Problem formulation and potential solutions, J. Mod. Power Syst. Clean Energy 4 (3) (2016) 353–361.

[60] C. Huang, F. Li, D. Zhou, J. Guo, Z. Pan, Y. Liu, Y. Liu, Data quality issues for synchrophasor applications part i: a review, J. Mod. Power Syst. Clean Energy 4 (3) (2016) 342–352.

[61] S. Wang, J. Zhao, Z. Huang, R. Diao, Assessing gaussian assumption of pmu measurement error using field data, IEEE Trans. Power Syst.

[62] J. Zhao, L. Mili, A framework for robust hybrid state estimation with unknown measurement noise statistics, IEEE Trans. Ind. Informat.

[63] P. A. Pegoraro, A. Angioni, M. Pau, A. Monti, C. Muscas, F. Ponci, S. Sulis, Bayesian approach for distribution system state estimation with non-gaussian uncertainty models, IEEE Trans. Instrum. Meas.

[64] Synchrophasor monitoring for distribution systems: Technical foundations and applications, Tech. rep., NASPI DisTT (2017).

[65] Power Standards Lab., URL: http://pqubepmu.com/about.php.

[66] A. Meier, E. Stewart, A. McEachern, M. Andersen, L. Mehrmanesh, Precision micro-synchrophasors for distribution systems: A summary of applications, IEEE Trans. Smart Grid.

[67] Y. Liu, S. You, W. Yao, Y. Cui, L. Wu, D. Zhou, J. Zhao, H. Liu, Y. Liu, A distribution level wide area monitoring system for the electric power grid-fnet/grideye, IEEE Access 5 (2017) 2329–2338.

[68] S. Wang, J. Zhao, Z. Huang, R. Diao, Assessing gaussian assumption of pmu measurement error using field data, IEEE Trans. Power Syst.

[69] N. M. Razali, Y. B. Wah, Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests, J. Stat. Model Anal. 2 (1) (2011) 21–33.

[70] A. Ghasemi, S. Zahediasl, Normality tests for statistical analysis: A guide for non-statisticians, Int. J. Endocrinol. Metab. 10 (2) (2012) 486–489.

[71] A. Meier, E. Stewart, A. McEachern, M. Andersen, L. Mehrmanesh, Precision micro-synchrophasors for distribution systems: A summary of applications, IEEE Trans. Smart Grid.

[72] Nerc., “verification of models and data for generator excitation control system or plant volt/var control functions,” 2014.

[73] Nerc., “verification of models and data for turbine/governor and load control or active power/frequency control functions” 2013.

[74] R. huang *et al.*, “calibrating parameters of power system stability models using advanced ensemble kalman filter,” *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2895-2905, may 2018.

[75] Z. huang, p. du, d. kosterev, and s. yang, “generator dynamic model validation and parameter calibration using phasor measurement at the point of connection,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1939-1949, may 2013.

[76] P. m. ashton *et al.*, “inertia estimation of the gb power system using synchrophasor measurements,” *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 701-709, mar. 2015.

[77] X. cao *et al.*, “switching markov gaussian models for dynamic power system inertia estimation,” *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3394-3403, sept. 2016.

[78] P. wall and v. terzjia, “simultaneous estimation of the time of disturbance and inertia in power systems,” *IEEE Trans. Power Del.*, vol. 29, no. 4, pp. 2018-2031, aug. 2014.

[79] N. petra *et al.*, “a bayesian approach for parameter estimation with uncertainty for dynamic power systems,” *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2735-2743, july 2017.

[80] A. chakrabortty *et al.*, “a measurement-based framework for dynamic equivalencing of large power systems using wide-area phasor measurement,” *IEEE Trans. Power Del.*, vol. 29, no. 4, pp. 2018-2031, aug. 2014.

[81] M. a. m. ariff, b. c. pal, and a. k. singh, “estimating dynamic model parameters for adaptive protection and control in power system,” *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 829-839, mar. 2015.

[82] I. a. hiskens, “power system modeling for inverse problems,” *IEEE Trans. Circuits and Syst.*, vol. 51, no. 3, pp. 539-551, mar. 2004.

[83] S. guo *et al.*, “adaptive parameter estimation of power system dynamic model using modal information,” *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 2854-2861, nov. 2014.

[84] P. pegoraro *et al.*, “bayesian approach for distribution system state estimation with non-gaussian uncertainty models,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 11, pp. 2957-2966, nov. 2017.

[85] A. k. singh and b. c. pal, “ decentralized dynamic state estimation in power systems using unscented transformation,” *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 794-804, mar. 2014.

[86] J. zhao and l. mili, “power system robust decentralized dynamic state estimation based on multiple hypothesis testing,” *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 4553-4562, july 2018.

[87] D. xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, princeton university press, princeton, nj, 2010.

[88] P. w. sauer and m. a. pai, *Power System Dynamics and Stability*, upper saddle river, nj: Prentice-hall, 1998.

[89] A. rouhani and a. abur, “observabiliy analysis for dynamic state estimation of synchronous machines,” *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3168-3175, july 2017.

[90] A. tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, philadelphia, pa, usa: Siam, 2005.

[91] Y. marzouk and h. n. najm, "dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems," *J. Comput. Phys.*, vol. 228, no. 6, pp. 1862-1902, apr. 2009.

[92] Y. marzouk and d. xiu, "a stochastic collocation approach to bayesian inference in inverse problems," *Commun. Comput. Phys.*, vol. 6, no. 4, pp. 826-847, oct. 2009.

[93] Z. ren, w. li, r. billinton, and w. yan, "probabilistic power flow analysis based on the stochastic response surface method," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2307-2315, may 2016.

[94] B. sudret, "global sensitivity analysis using polynomial chaos expansions", *Reliab. Eng. Syst. Saf.*, vol. 93, no. 7, pp. 964-979, july 2008.