DATA ASSIMILATION AND FORECASTING FOR COMPLEX NONLINEAR SYSTEMS WITH SUITABLE APPROXIMATE MODELS

by

Yingda Li

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Mathematics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2021

Date of final oral examination: 11/17/21

The dissertation is approved by the following members of the Final Oral Committee:

Nan Chen, Assistant Professor, Mathematics

Stephanie Henderson, Assistant Professor, Atmospheric and oceanic sciences

Qin Li, Associate Professor, Mathematics

Samuel N. Stechmann, Professor, Mathematics

To my parents and my husband

ACKNOWLEDGMENTS

It is beyond my expression to describe my gratitude to my advisor, Professor Nan Chen. Nan has offered me a tremendous amount of guidance, caring, encouragement, and patience during my pursuit of this degree. His teachings of all aspects, from conducting research to academic writing and presentation, have transformed me from a juvenile student to a relatively mature researcher. I could not imagine a better advisor than him.

I am also indebted to my other committee members: Professor Qin Li, Professor Samuel Stechmann, and Professor Stephanie Henderson. Qin has always been an inspiring role model to me. I have heard many stories about her being talented and hard-working. I am also very thankful for her insightful questions during my talks and valuable advice throughout these years. I am immensely grateful to Sam, who generously held an extremely long office hour to help me prepare for the qualify exam, one of the crucial factors I was admitted to UW-Madison. I am sincerely grateful to Steph for serving on my committee and for her invaluable expertise and advice to my dissertation.

I must express my sincere gratefulness to my collaborator, Professor Honghu Liu, for his valuable feedback, insightful ideas, and precious time and effort on our project. This dissertation would not be possible without his help.

My gratitude must be extended to the advisor of my Master's program, Professor Shi Jin. He prepared and helped me in every possible way to pursue this doctorate. His enthusiasm for math and music has always been an inspiration to me.

There was a multitude of individuals I met in Madison who helped me to arrive

at this point. Special thanks to Hongfei Chen, Xiao Hou, Ying Li, Ranran Zhu Mi, and Yuhua Zhu for their friendship and support.

Until this day, I still feel incredibly lucky that I attended that skiing event in 2016, where I met Wei, my love, husband, and best friend. He has witnessed all my struggles and achievements during the past five years. Without his steadfast love, constant companion, unconditional sacrifice, as well as his brilliant ideas and constructive feedback to my research, I would not get through this whole process of completing this degree.

I owe a debt of gratitude to my parents, my safest harbor. They have been encouraging me to follow my passion, supporting my every single decision, and providing the best they can offer—from childhood, to now, and beyond.

Last but not least, I would like to thank my God for bringing me to Resonate church at the perfect time, where the spiritual brothers and sisters gave me enormous comfort and support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Complex nonlinear turbulent systems are ubiquitous in geoscience, engineering, neural and material sciences. This thesis focuses on three of cental topics related to complex nonlinear turbulent dynamical systems, the effective prediction, uncertainty quantification, and data assimilation.

In the first part of this thesis, a simple but effective Bayesian Machine learning advanced forecast ensemble method is developed, which combines an available imperfect physics-informed model with data assimilation to facilitate the ML ensemble forecast. In the BAMCAFE framework, a Bayesian ensemble DA is applied to create the training data of the ML model, which reduces the intrinsic error in the imperfect physics-informed model simulations and provides the training data of the unobserved variables. Then a generalized DA is employed for the initialization of the ML ensemble forecast. Besides, the BAMCAFE also provides an accurate quantification of the forecast uncertainty utilizing a non-Gaussian probability density function that characterizes the intermittency and extreme events.

In the second part, the skill of a rich class of nonlinear stochastic models, known as the "conditional Gaussian nonlinear system" (CGNS), as both a cheap surrogate model and a fast preconditioner is explored to advance many computationally challenging tasks in complex nonlinear systems. The CGNS preserves the underlying physics to a large extent and reproduces the observed intermittency, extreme events and other non-Gaussian features of nature. Second, the CGNS allows the development of a fast algorithm for simultaneously estimating the parameters and the unobserved variables with uncertainty quantification in the presence of only

partial observations. Utilizing an appropriate CGNS as a preconditioner significantly reduces the computational cost in accurately estimating the parameters in the original complex system. Finally, the CGNS advances rapid and statistically accurate algorithms for computing the probability density function and sampling the trajectories of the unobserved state variables. These fast algorithms facilitate the development of an efficient and accurate data-driven method for predicting the linear response of the original system with respect to parameter perturbations based on a suitable CGNS preconditioner.

In the third part, a nonlinear optimal filter, a nonlinear optimal smoother, and the associated filter-based forward sampling and smoother-based backward sampling algorithms are developed for CGNS. The optimal nonlinear smoother also outweighs the optimal nonlinear filter in detecting the hidden mechanism of triggering observed extreme events. In addition, the optimal nonlinear smoother improves the state estimation in stochastic parameterizations and Lagrangian data assimilation. Next, an efficient and accurate online forward-in-time smoother is developed. Comparing with the nonlinear filter, the sequential update of the existing state estimates using the online nonlinear smoother leads to a more accurate recovery of intermittent time series and the associated non-Gaussian features. Finally, the online nonlinear smoother is incorporated into an efficient expectation-maximization algorithm for online parameter estimation.

1 INTRODUCTION

Complex nonlinear turbulent systems are ubiquitous in geoscience, engineering, neural and material sciences.

Chapter 2 introduced a simple but effective Bayesian machine learning advanced forecast ensemble (BAMCAFE) method that combines an available imperfect physics-informed model with DA to facilitate the ML ensemble forecast. Ensemble forecast based on physics-informed models is one of the most widely used forecast algorithms for complex nonlinear turbulent systems. A major difficulty in such a method is the model error that is ubiquitous in practice. Data-driven machine learning (ML) forecasts can overcome the model error but they often suffer from the partial and noisy observations. In this paper, a simple but effective Bayesian Machine learning advanced forecast ensemble (BAMCAFE) method is developed, which combines an available imperfect physics-informed model with data assimilation (DA) to facilitate the ML ensemble forecast. In the BAMCAFE framework, a Bayesian ensemble DA is applied to create the training data of the ML model, which reduces the intrinsic error in the imperfect physics-informed model simulations and provides the training data of the unobserved variables. Then a generalized DA is employed for the initialization of the ML ensemble forecast. In addition to forecasting the optimal point-wise value, the BAMCAFE also provides an accurate quantification of the forecast uncertainty utilizing a non-Gaussian probability density function that characterizes the intermittency and extreme events. It is shown using a two-layer Lorenz 96 model that the BAMCAFE method can significant improve the forecasting skill compared to the typical reduced-order

imperfect models with bare truncation or stochastic parameterization for both the observed and unobserved large-scale variables. It is also shown via a nonlinear conceptual model that the BAMCAFE leads to a comparable non-Gaussian forecast uncertainty as the perfect model while the associated imperfect physics-informed model suffers from large forecast biases.

In chapter 3, the skill of a rich class of nonlinear stochastic models, known as the "conditional Gaussian nonlinear system" (CGNS), as both a cheap surrogate model and a fast preconditioner is explored to advance many computationally challenging tasks in complex nonlinear systems. Developing suitable approximate models for analyzing and simulating complex nonlinear systems is a practically important topic. The objective of this paper is to explore the skill of a rich class of nonlinear stochastic models, known as the conditional Gaussian nonlinear system (CGNS), as both a cheap surrogate model and a fast preconditioner for facilitating many computationally challenging tasks. The CGNS preserves the underlying physics to a large extent and reproduces the observed intermittency, extreme events and other non-Gaussian features of nature. First, the closed analytic formulae of solving the conditional statistics provide an efficient and accurate data assimilation scheme. It is shown that the data assimilation skill of a suitable CGNS approximate forecast model outweighs that by applying an ensemble based method even to the perfect model in the presence of strong nonlinearity and turbulence, where the latter suffers from filter divergence. Second, the CGNS allows the development of a fast algorithm for simultaneously estimating the parameters and the unobserved variables with un- certainty quantification in the presence of only partial observa-

tions. Utilizing an appropriate CGNS as a preconditioner significantly reduces the computational cost in accurately estimating the parameters in the original complex system. Finally, the CGNS advances rapid and statistically accurate algorithms for computing the probability density function and sampling the trajectories of the unobserved state variables. These fast algorithms facilitate the development of an efficient and accurate data-driven method for predicting the linear response of the original system with respect to parameter perturbations based on a suitable CGNS preconditioner.

In chapter 4, efficient nonlinear filtering, nonlinear smoothing, forward sampling and backward sampling algorithms are developed and compared for a rich class of nonlinear complex turbulent dynamical systems using only partial observations. Point-wise state estimation and path-wise optimal sampling of unobserved processes in complex nonlinear turbulent dynamical systems, given partially observed time series, are important topics in practice. A nonlinear optimal filter, a nonlinear optimal smoother, and the associated filter-based forward sampling and smoother-based backward sampling algorithms are developed for a rich class of highly nonlinear systems with significant non-Gaussian features. Both the state estimation and the optimal sampling algorithms involve closed analytic formulae, which allow rigorous mathematical analysis and efficient numerical simulations. It is shown that there exist intrinsic barriers in the posterior mean time series and the trajectories from the filter-based forward sampling algorithm in reproducing the nonlinear dynamical and statistical features of nature while the smoother-based sampling algorithm succeeds in recovering the salient non-Gaussian statistics and

the crucial dynamical characteristics. The optimal nonlinear smoother also outweighs the optimal nonlinear filter in detecting the hidden mechanism of triggering observed extreme events. In addition, the optimal nonlinear smoother improves the state estimation in stochastic parameterizations and Lagrangian data assimilation. Next, an efficient and accurate online forward-in-time smoother is developed. Comparing with the nonlinear filter, the sequential update of the existing state estimates using the online nonlinear smoother leads to a more accurate recovery of intermittent time series and the associated non-Gaussian features. Finally, the online nonlinear smoother is incorporated into an efficient expectation-maximization algorithm for online parameter estimation. Numerical simulations show that the observed extreme events facilitate a rapid convergence of the iterations and provide an accurate estimation of model parameters.

2 BAMCAFE: A BAYESIAN MACHINE LEARNING ADVANCED FORECAST ENSEMBLE METHOD FOR COMPLEX TURBULENT SYSTEMS WITH PARTIAL OBSERVATIONS

2.1 Introduction

Forecasting complex turbulent systems is an important task in many areas, particularly in geophysics, engineering, neuroscience, and climate science (Majda, 2016; Strogatz, 2018; Wilcox, 1988; Sheard and Mostashari, 2009). These systems are often characterized by a large dimensional phase space with strong nonlinear interactions between different spatial and temporal scales, which transfer energy throughout the system. Intermittent instability, extreme events, non-Gaussian probability density functions (PDFs), and multiscale dynamics are typical characteristics in these systems (Farazmand and Sapsis, 2019; Moffatt, 2021). Because of such features, errors and uncertainties from various sources (e.g., initializations, parameters and approximations) may get amplified in the forecast stage, which leads to big challenges in effectively forecasting these complex systems. In addition to predicting the optimal point-wise value, an accurate quantification of the forecast uncertainty of these turbulent systems is an equally important issue.

Ensemble forecast based on physics-informed (or parametric) models is one of the most widely used forecast algorithms for predicting turbulent signals (Palmer, 2019; Toth and Kalnay, 1997; Leutbecher and Palmer, 2008). Starting from a given initial condition, multiple model simulations are conducted resulting in a forecast

ensemble. However, developing effective ensemble forecast algorithms that enable a skillful forecast with an accurate uncertainty quantification is very challenging mainly for the following three reasons. First, only partial and noisy observations are available in many real-world situations (Kalnay, 2003; Lau and Waliser, 2011). For example, temperature at the sea surface are more accessible than inside the deep ocean from satellite observations, and large-scale components of many geophysical systems are more observable than small-scale components. Such partial and noisy observations often introduce large biases and uncertainties at the initialization stage, which significantly affect the accuracy of the ensemble forecast as time evolves. Second, due to the high dimensionality and the complexity of many turbulent systems, it is computationally expensive for even a single run of the forecast, prohibiting the forecast being repeated multiple times to create the ensemble. Third, one of the major and ubiquitous difficulties in applying the parametric model-based ensemble forecast is the model error (Majda and Chen, 2018; Allen et al., 2002). The presence of the model error is often due to a lack of the perfect understanding of nature and the inadequate resolution in the models because of the limited computing power, where model reduction techniques and parameterizations are often adopted when developing practical approximate models (Palmer, 2001; Phillips et al., 2004; Rotstayn, 2000; Mou et al., 2020, 2021a).

Purely data-driven approaches, such as machine learning (ML) or other non-parametric models have gained great interest in the past decades (Vlachas et al., 2018; Chattopadhyay et al., 2020a,c; Scher, 2018; Scher and Messori, 2018; Weyn et al., 2020; Pathak et al., 2018; Rasp and Thuerey, 2021; Raissi et al., 2019; Beucler

et al., 2019; Chattopadhyay et al., 2020b). Given a sufficiently large amount of accurate training data, ML models with well-designed architectures can extract key information from the high-dimensional complex systems, which allows them for effective forecasts (Vlachas et al., 2018; Chattopadhyay et al., 2020a,c; Scher, 2018; Scher and Messori, 2018). It is also worthwhile to notice that once the model has been trained, the computational cost for executing these data-driven models is much cheaper (or even negligible) compared to the cost in a typical numerical solver for the parametric models (Chattopadhyay et al., 2020c). Popular ML models include the long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997; Vlachas et al., 2018), the convolutional neural networks (LeCun et al., 1998; Weyn et al., 2020), and the echo state networks (Jaeger, 2007; Pathak et al., 2018). However, these ML models may suffer from the polluted, incomplete, and insufficient training data, which appear in many real applications resulting from the partial and noisy observations (Rasp and Thuerey, 2021; Chattopadhyay et al., 2021; Farchi et al., 2020; Brajard et al., 2021). Physics-informed ML models partially solve this issue by enforcing some key physics knowledge (e.g., conservation law) as constraints or including them into the model architecture design (Raissi et al., 2019; Beucler et al., 2019; Chattopadhyay et al., 2020b; De Bézenac et al., 2019).

Data assimilation (DA), which combines parametric models with observations, plays a vital role in assisting physics-based model forecast of turbulent systems (Law et al., 2015; Ghil and Malanotte-Rizzoli, 1991; Kalnay, 2003). The main benefits of DA are two-fold. First, DA recovers the states of the unobserved variables. Second, by incorporating the information from observations into the available imperfect

model, the model error and the observational noise are simultaneously mitigated. Therefore, DA improves the initialization of both the observed and the unresolved state variables that facilitates the ensemble forecast. However, utilizing partial observations to correct the model error via DA only applies to the initialization stage of the standard physics-based model ensemble forecast; the model error continuously enters into the forecast ensembles as time evolves. On the other hand, DA can also be incorporated into the ML forecast, where promising results have been shown in various approaches (Chattopadhyay et al., 2021; Tomizawa and Sawada, 2020; Bocquet et al., 2020; Farchi et al., 2020; Brajard et al., 2021; Chen, 2020a). In iterative methods, the ML model is treated as a surrogate model in DA (Brajard et al., 2020; Wikner et al., 2021; Bocquet et al., 2020). However, the so-called cold start problem may arise in this type of approach that leads to the potential numerical instability. In addition, the algorithms may take a long time to converge. One remedy is to add a neural network (NN) as a residual to correct the imperfect knowledge-informed outcomes, which helps reduce the number of cycles as well (Farchi et al., 2020; Brajard et al., 2021). Note that some methods also attempt to predict the uncertainty by training an ML model on the error residual (Scher and Messori, 2018). Other ML forecast approaches involving perturbing the initial conditions are also developed to mimic the traditional ensemble forecast approach. However, as was pointed in (Scher and Messori), the uncertainty obtained in these methods can be systematically lower than that in the physics-based numerical prediction models.

This paper develops a simple but effective Bayesian Machine Learning Advanced

Forecast Ensemble (BAMCAFE) method. Assume that a partial and noisy observational time series and a physics-based but imperfect parametric model are available, as in many realistic situations. The BAMCAFE combines ML with DA to improve the predictions utilizing the standard ensemble forecast by running the imperfect parametric model forward. Different from many purely data-driven methods, the BAMCAFE takes advantage of the available imperfect parametric model to extract useful information that feeds into a ML model via DA. The BAMCAFE starts with a Bayesian ensemble DA that aims to reduce the observational noise and recover the time series of the unobserved variables, which often allow the resulting assimilated trajectories to more accurately represent the underlying dynamics of nature than the imperfect parametric model. Therefore, if these assimilated trajectories are used to build a new forecast model, then the associated forecast is expected to be improved. Since it is in general very challenging to write down a set of physics-informed parametric equations to describe the time evolution of these assimilated trajectories, ML models are used to characterize the underlying dynamics in the BAMCAFE framework. Next, in addition to forecasting the optimal point-wise value, the BAMCAFE is also designed to quantify the forecast uncertainty. The forecast uncertainty in the BAMCAFE is represented by a non-Gaussian PDF, which is computed via an inexpensive algorithm from a mixture distribution. Such a non-Gaussian distribution is particularly appropriate for characterizing the uncertainty in complex turbulent systems in the presence of intermittency and extreme events. Then in the forecast initialization stage of the BAMCAFE approach, a generalized ensemble DA is utilized, which provides an ensemble of time series that serve as the input of the

ML model. Subsequently, the ML forecast is carried out for each ensemble member to obtain both the point-wise forecast value and the forecast uncertainty. Overall, the BAMCAFE is computationally efficient in both the training and forecasting stages. It collects the useful information from the partial and noisy observations as well as the imperfect physics-informed model to facilitate the ML ensemble forecast.

The rest of the paper is organized as follows. The BAMCAFE algorithm is developed in Section 2.2. Section 2.3 includes nonlinear and non-Gaussian test examples comparing the forecast skill of the BAMCAFE algorithm and the associated imperfect models. The paper is concluded in Section 2.4.

2.2 The Bayesian Machine Learning Advanced Forecast Ensemble (BAMCAFE) Framework

2.2.1 Overview

The ensemble forecast is a popular prediction approach for turbulent systems that employs a collection, known as the "ensemble", of multiple individual forecasts from a parametric model. A skillful ensemble forecast requires an accurate representation of the underlying dynamics as well as a reliable forecast initialization. DA is used to generate a more accurate initialization by combining partial and noisy observations with the given imperfect model. However, the forecast error can grow up quickly as time evolves due to the model bias, which often results in

an inaccurate quantification of the forecast uncertainty as well.

In the Bayesian Machine Learning Advanced Forecast Ensemble (BAMCAFE) framework, the intrinsic error in the imperfect physics-based forecast model is alleviated by training a ML model (e.g., a NN) based on a set of assimilated trajectories, which are obtained by applying a Bayesian sampling method (i.e., a Bayesian ensemble DA) to the imperfect physics-based model with the help from the available partial and noisy observational time series. Combining the information from both the imperfect model and the noisy observations, the assimilated trajectories achieve trajectory-wise improvement compared with the signals generated from the imperfect model in terms of both the dynamical and statistics features. Specifically, the BAMCAFE involves the following four steps:

1. Generating the ML training data using a Bayesian sampling approach.
2. Training a ML model (e.g., a NN) utilizing the training data from Step 1.
3. Employing a generalized DA for the initialization of the ML model.
4. Applying a ML ensemble forecast.

The generalized DA in Step 3 aims at recovering a short piece of time series before the initial time instant for forecast, serving as the initialization of the ML model. It plays a similar role as the traditional physics-based parametric model forecast, but the initialization is not just at a single time instant. Figure 2.1 includes a schematic illustration of the traditional physics-informed parametric model based ensemble forecast and the BAMCAFE approach. The details of each of the four steps in the BAMCAFE algorithm will be explained in the following subsections.

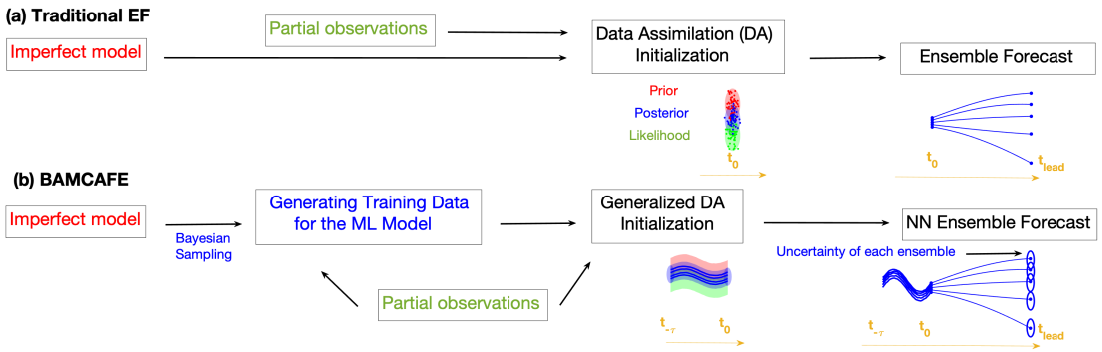

Figure 2.1: A schematic illustration of the traditional physics-informed parametric model based ensemble forecast and the BAMCAFE approach.

2.2.2 Generating the ML training data using a Bayesian sampling approach

The first step of the BAMCAFE algorithm is to generate a set of trajectories that will be used to train the ML model. This is achieved by exploiting a Bayesian sampling approach that takes into account the information from both the given imperfect physics-based parametric model and the available partial and noisy observational time series. Denote by $\mathbf{x}_j$ the state variable and $\mathbf{y}_j$ the observational vector at time t_j, where the dimension of $\mathbf{y}_j$ is often smaller than that of $\mathbf{x}_j$ due to the partial observations. The training data for the ML model is generated from the following conditional distribution,

$$p(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N), \qquad 0 \leqslant j \leqslant N, \tag{2.1}$$

where $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ stands for all the available observations in a given time interval. For the rest of this paper, we define $\mathbf{y}_{i:j} = (\mathbf{y}_i^\top, \mathbf{y}_{i+1}^\top, \ldots \mathbf{y}_j^\top)^\top$ for $0 \leqslant i \leqslant j$. In the Bayesian framework, the statistical information provided by the available imperfect parametric model is called the prior distribution while the conditional distribution (2.1) is named as the posterior distribution. Many existing Bayesian sampling algorithms can be applied to compute the posterior distribution (2.1) (Tong et al., 2020; Bédard, 2017; Johnson et al., 2013; Ottobre et al., 2016; Agapiou et al., 2017). In this work, the ensemble Kalman smoother (EnKS) (Evensen and Van Leeuwen, 2000), which is one of the commonly used DA approaches, is adopted to sample from the posterior distribution. The technique details of sampling from the posterior distribution utilizing the EnKS are summarized in the Appendix A.1. Denote by $\{\mathbf{x}_{0|N}^k, \ldots, \mathbf{x}_{N|N}^k\}$ for $k = 1, \ldots, K$ the resulting K ensemble members from the EnKS, each of which is a time series. The trajectories of these ensemble members will be used as the training data for the ML model. By using the Bayesian DA, the model error is mitigated in these posterior time series due to the extra information from observations. In addition, the training data are now also available for the unobserved variables, without which it is quite challenging to apply the ML models to predict the entire system.

It is important to note that despite the ensemble mean time series from the EnKS $\{\boldsymbol{\mu}_{0|N}, \ldots, \boldsymbol{\mu}_{N|N}\}$ being a widely used surrogate for the true signal as in the standard reanalysis approaches, the ensemble mean time series is not always a suitable training data for the ML model because the fluctuation of the original turbulent dynamics is smoothed out in the ensemble mean time series. In other words, the

dynamical and statistical features of the underlying dynamics are not fully reflected in the ensemble mean time series. Different from the smoother mean, each ensemble member $\{\mathbf{x}^k_{0|N}, \ldots, \mathbf{x}^k_{N|N}\}$, referred as a sampled trajectory, is more appropriate for training the ML models. This is one of the fundamental differences between the training data in the BAMCAFE framework and the traditional reanalysis outcome that is often given by the ensemble mean time series. A comparison between the time series of the ensemble members and the smoother mean as well as the associated ML forecast results will be presented in Section 2.3.3.4.

It is also worthwhile to note that the EnKS is a more suitable method than the ensemble Kalman filter (EnKF), which takes into account only the information in the past, to create the ML training time series. The state estimation using the EnKF is often less accurate than that using the EnKS, especially in the presence of strong turbulence, intermittency and extreme events (Evensen and Van Leeuwen, 2000; Chen and Majda, 2020). Since the training procedure is offline, where the observational time series in a given interval $\mathbf{y}_{1:N}$ is in hand, it is natural to adopt the EnKS for creating the training data for the ML models.

2.2.3 Training a ML model

Given the sampled trajectories from Step 1, the second step of the BAMCAFE algorithm is to build a new model that captures the key features of these sampled trajectories. Since it is often quite difficult to develop a physics-based parametric model that perfectly captures the underlying dynamics of these sampled trajectories, it is natural to employ a ML model to characterize their time evolutions. For the

convenience of discussion, a NN will be utilized as the ML model throughout the paper. It is worthwhile to note that numerous works have been shown that NNs can well approximate many complex dynamics (Chen and Chen, 1995; Hornik, 1993; Sonoda and Murata, 2017).

Denote by $\mathbf{x}^k_{0:N}$ the k-th member of the sampled trajectories from t_0 to t_N, as a simpler notation for $\{\mathbf{x}^k_{0|N}, \ldots, \mathbf{x}^k_{N|N}\}$. The entire sampled trajectory is further separated as the training and the validation periods, denoted by $\mathbf{x}^k_{0:N_{tr}}$ and $\mathbf{x}^k_{N_{tr}:N}$, respectively. The NN model is defined as $g(\mathbf{x}; \theta)$, where θ is a set of trainable parameters in the network. The ML task is to predict the value that is l steps forward in time, which can be expressed as follows

$$\hat{\mathbf{x}}^k_{n+l} = g(\mathbf{x}^k_{0:n}; \theta), \text{ for } N_0 \leqslant n \leqslant N_{tr} - l, \tag{2.2}$$

where l is the lead time or forecast horizon and N_0 is the length of the initial period of the NN. We aim to introduce the framework instead of sophisticated NN architectures, a one-layer LSTM model followed by a fully connected layer will be used in the numerical experiments of this work, the NN structure of which is shown in Figure 2.2. The LSTM here can be replaced by other non-parametric models such as convolutional neural networks depending on the applications. Note that the right-hand side of (2.2) is written in a general form, in which the recurrent neural network can take varying length sequences as the input. A simpler version of the general form (2.2) replaces $\mathbf{x}^k_{0:n}$ by $\mathbf{x}^k_{n-N_{init}:n}$ on the right-hand side, which is widely

used in practice. The loss function used in the training phase is given as follows,

$$J(\theta) = \sqrt{\frac{1}{K(N_{tr} - l + 1 - N_0)} \sum_{k=1}^{K} \sum_{n=N_0}^{N_{tr}-l} \|\hat{\mathbf{x}}_n^k(\theta) - \mathbf{x}_n^k\|^2}. \tag{2.3}$$

The loss function (2.3) is the mean-squared error, which is a commonly used loss function in time series predictions. Again, other loss functions can be utilized in the BAMCAFE framework. Note that the error $\hat{\mathbf{x}}_n^k(\theta) - \mathbf{x}_n^k$ evaluated on validation set is crucial in quantifying the uncertainty in the forecast stage, which will be discussed in Section 2.2.5.

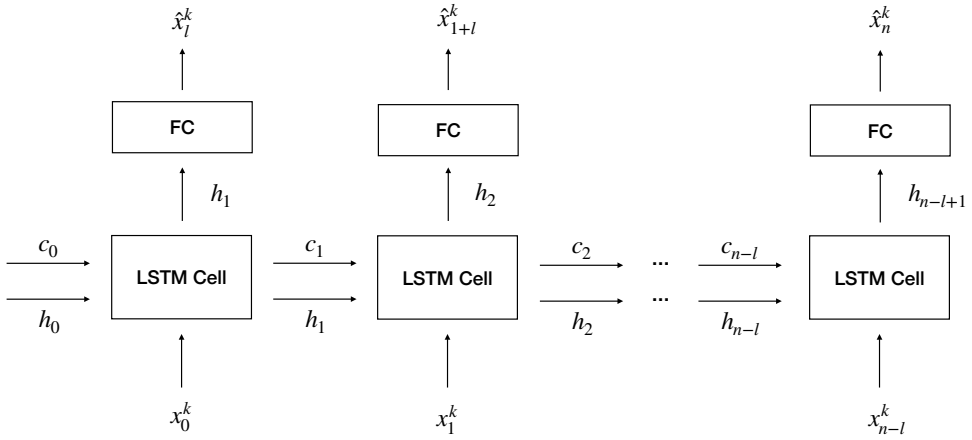

Figure 2.2: Structure of the one layer LSTM model used in this work. FC means a fully-connected layer.

2.2.4 Employing a generalized DA for the initialization of the ML model

With the trained NN in hand, what remains is to apply the BAMCAFE for the ensemble forecast. The goal is to predict both the point-wise value and the associated uncertainty at the lead time l, i.e., to predict $\mathbf{x}_{N+l}$ given an observational sequence $\mathbf{y}_{1:N}$.

Following (2.2), the predicted value for each ensemble member is given by

$$\hat{\mathbf{x}}^k_{N+l} = g(\mathbf{x}^k_{N-L_{init}:N}; \theta^*), \tag{2.4}$$

for $k = 1, \ldots, K_p$, where K_p is the total number of the ensembles used in prediction that can be different from the number of the training trajectories K. In (2.4), θ^* is the value minimizing the loss function obtained from the Step 2 while L_{init} is the length of the initial period. Adopting a period of time series $\mathbf{x}^k_{N-L_{init}:N}$ as the input is necessary for the LSTM and other networks to take into account the memory effects of the dynamics that significantly facilitates the forecast, which is also consistent with the input data in (2.2).

Similar to the training data, only partial observations $\mathbf{y}_{1:N}$ are available for the state variable $\mathbf{x}_{1:N}$ at the forecast initialization stage. Therefore, an initialization procedure has to be applied to obtain $\mathbf{x}^k_{N-L_{init}:N}$ that is the prerequisite for implementing (2.4). This initialization can be regarded as a generalization of the traditional DA because 1) the initial condition $\mathbf{x}^k_{N-L_{init}:N}$ can be obtained via a Bayesian ensemble DA formula; and 2) the value at the end point $\mathbf{x}^k_N$ is exactly the initial values

used in the traditional ensemble forecast. The main difference is that the initial condition in the traditional approaches is a group of point-wise samples at only the time instant t_N yet those in the BAMCAFE framework is a group of time series from $t_{N-L_{init}}$ to t_N. The generalized DA for the initialization of the NN is carried out utilizing the following Bayesian formula

$$p(\mathbf{x}_{N-L_{init}}, \mathbf{x}_{N-L_{init}+1}, \ldots, \mathbf{x}_N | \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N), \qquad 0 \leqslant j \leqslant N, \tag{2.5}$$

where the prior information is still provided by the imperfect parametric model, as in (2.1). Again, the EnKS is applied in this work to sample the trajectories from the posterior distribution (2.5), serving as the initialization of the NN forecast model in (2.4). Note that the value of $\mathbf{x}_N^k$ from the EnKS is the same as that by applying the EnKF but those $\mathbf{x}_{N-L_{init}}^k, \mathbf{x}_{N-L_{init}+1}^k, \ldots, \mathbf{x}_{N-1}^k$ are statistically more accurate utilizing the EnKS.

2.2.5 Applying a ML ensemble forecast

Given the initialization from the generalized DA, it is ready to run (2.4) for the ensemble forecast.

The ensemble mean, which is the most widely used surrogate for the point-wise prediction value, is calculated as follows,

$$\bar{\hat{\mathbf{x}}}_{N+l} = \frac{1}{K_p} \sum_{k=1}^{K_p} \hat{\mathbf{x}}_{N+l}^k, \tag{2.6}$$

where $\hat{\mathbf{x}}^k_{N+l}$ is calculated from (2.4). Note that the mode or other statistical measures can also be used as the optimal point-wise prediction value depending on the quantity of interest of the applications.

In addition to the ensemble mean, predicting the uncertainty is also essential, especially for complex turbulent dynamics with intermittency and extreme events. Unlike many ML approaches that focus only on the optimal point-wise forecast value, the BAMCAFE also provides the quantification of the forecast uncertainty. The total forecast uncertainty contains two parts. The first part is the ensemble spread of the point-wise forecasts, namely the $\hat{\mathbf{x}}^k_{N+l}$ with $k = 1, \dots K_p$. The second and indispensable component is the intrinsic uncertainty associated with each ensemble member. The ensemble spread of the point-wise forecasts is the dominant source of the uncertainty at short lead times, where the spread comes from DA. As the lead time increases, the point-wise forecast becomes less accurate, the associated error of which contributes to the total uncertainty of the forecast. Because of this, the BAMCAFE exploits the validation error obtained in the ML training period as the measurement of the forecast uncertainty associated with each ensemble member. Such a simple criterion is a natural choice for quantifying the forecast uncertainty since it represents the residual part of the dynamics which cannot be well characterized and forecasted by the LSTM model. Some quantitative studies between the forecast uncertainty in the LSTM forecast based on the validation error and that in the ensemble forecast using the perfect parametric model will be illustrated in Section 2.3.2.4. Note that the validation error here can be replaced by the training error, assuming the NN is appropriate, i.e., without being overfitted or

underfitted. Specifically, the forecast uncertainty associated with the BAMCAFE is represented by a non-Gaussian PDF. This non-Gaussian PDF is constructed by a mixture distribution, where each mixture component is another non-Gaussian distribution that is associated with one forecast ensemble member in (2.4). The k-th mixture component is given by adding the point-wise forecast value $\hat{\mathbf{x}}^k_{N+l}$ to a non-Gaussian distribution ϵ,

$$p(\mathbf{x}^k_{N+l}) = \hat{\mathbf{x}}^k_{N+l} + \epsilon, \tag{2.7}$$

where ϵ is the distribution of the validation error,

$$\epsilon \sim \text{PDF of } \left(\mathbf{x}^{k'}_{n+l} - g(\mathbf{x}^{k'}_{N_{tr}:n}; \theta^*), \quad \text{for } k' = 1, \ldots, K, \quad N_{tr} + 1 \leqslant n \leqslant N - l\right). \tag{2.8}$$

As a final remark, a connection can be built between using the parametric model and the BAMCAFE algorithm to characterize forecast uncertainty. In fact, the ensemble ML forecast algorithm of the BAMCAFE can be regarded as a surrogate of the ensemble forecast using physics-informed parametric models but using a different approach of computing the forecast uncertainty. To see such a connection, suppose that we have an ensemble of the forecast value at time t_{N+l}. Consider the following mean-fluctuation decomposition of the forecast value for each ensemble with an index k,

$$\hat{\mathbf{x}}^k_{N+l} = \bar{\hat{\mathbf{x}}}_{N+l} + (\hat{\mathbf{x}}^k_{N+l} - \bar{\hat{\mathbf{x}}}_{N+l}), \tag{2.9}$$

where the first term on the right-hand side is the ensemble mean and the second

term is the fluctuation. The uncertainty is essentially given by the statistics, namely the PDF, of the fluctuation part. In the traditional forecast using physics-informed parametric models, each ensemble member is given directly by running either stochastic or deterministic chaotic models forward. The results of such an ensemble forecast can be directly used to form the forecast PDF. Nevertheless, in most of the ML forecasts, the ML model/mapping is deterministic and non-chaotic. The ML forecast output is typically a deterministic value provided by a certain averaging process (i.e., the optimal mapping) inside the complicated ML architecture. There is a residual from such an averaging process, which is characterized by the validation error. The validation error for each sample is essentially the second part on the right-hand side of (2.9). In other words, the validation error mimics the ensemble spread in the traditional ensemble forecast using physics-informed parametric models. If the training and testing data have the same features, then it is expected that the residual in the testing period should be similar to that in the validation period. Therefore, it is natural to use the validation error as a characterization of the uncertainty. Notably, a more accurate ML model in terms of representing the perfect model dynamics is expected to provide a closer result of the forecast uncertainty as the perfect model. This fact implies that the Bayesian DA is crucial in creating an improved data set for training the ML model.

2.3 Test Examples

This section will illustrate how the BAMCAFE improves ensemble forecast results by employing testing models that mimic many desirable features of complex turbulent systems in reality. In each test case, we demonstrate a perfect model with specific parameters that produces the true dynamics. The baseline is the ensemble forecasts using one or two commonly used imperfect parametric models. Based on 50 sampled trajectories, a NN model, as was shown in Figure 2.2, with a one-layer LSTM network followed by a fully connected layer is trained. The 'Adam' optimization algorithm and the mean-squared error loss function (2.3) are utilized to train different LSTM models for different lead times. The details of hyperparameters are included in Appendix A.3.

The two test experiments used below will emphasize different aspects of the BAMCAFE framework. In the first case, the goal is to illustrate that the BAMCAFE algorithm can improve the forecasting skill compared to the typical reduced-order imperfect models with either bare truncation or stochastic parameterization. In the second case, it is highlighted that the BAMCAFE facilitates the quantification of the forecast uncertainty in a strongly turbulent system with intermittency and extreme events.

2.3.1 General experiment setup

The true dynamics are integrated from the perfect model using the Euler-Maruyama scheme (Gardiner, 2004), with the integrated time step Δt being 0.001 for both the

perfect and imperfect models. The discrete observations are collected for every 0.05 time units.

The two criteria for quantifying the overall point-wise prediction skill are the root-mean-square error (RMSE) and the pattern correlation (Corr) between the prediction $\hat{x}$ and the true signal x. These criteria are defined as (Hyndman and Koehler, 2006):

$$\begin{aligned} \text{Corr} &= \frac{\sum_{i=1}^{I}(\hat{x}_i - \bar{\hat{x}})(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{I}(\hat{x}_i - \bar{\hat{x}})^2}\sqrt{\sum_{i=1}^{I}(x_i - \bar{\hat{x}})^2}}, \\ \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^{I}(\hat{x}_i - x_i)^2}{I}}, \end{aligned} \tag{2.10}$$

where $\bar{\hat{x}}$ and $\bar{x}$ are the averages of the scalar quantities $\hat{x}_i$ and x_i over $i = 1, \ldots, I$, respectively. In general, if RMSE is below one standard deviation of the true signal and Corr is above the threshold value $\text{Corr} = 0.5$, then the prediction is said to be skillful.

On the other hand, the relative entropy (also known as the Kullback–Leibler divergence) is one of the most suitable measurements that quantifies the accuracy of the forecast uncertainty (Kleeman, 2011; Majda et al., 2002; Xu, 2007; Branicki et al., 2013). It measures the statistical distance between the PDF $p(\mathbf{x})$ associated with the perfect model ensemble forecast and that $p(\hat{\mathbf{x}})$ associated with either the imperfect parametric model ensemble forecast or that from the BAMCAFE at a given time instant. The relative entropy is defined as (Kullback and Leibler, 1951;

Kullback, 1987; Cover and Thomas, 1991):

$$\mathcal{P}(p(\mathbf{x}), p(\hat{\mathbf{x}})) = \int p(\mathbf{x}) \log \left(\frac{p(\mathbf{x})}{p(\hat{\mathbf{x}})} \right). \tag{2.11}$$

The relative entropy is zero if the two PDFs equal with each other. The relative entropy increases monotonically as the difference between the two PDFs becomes large.

2.3.2 Improving the ensemble forecasts from the reduced-order models with bare truncation or stochastic parameterization

The first numerical test example is the two-layer Lorenz 96 model, which aims at showing that the BAMCAFE algorithm can improve the forecasting skill compared to the typical reduced order imperfect models with either bare truncation or stochastic parameterization (Majda and Grote, 2009; Grooms and Majda, 2014b,a; Majda and Grooms, 2014).

2.3.2.1 The perfect model

The two-layer Lorenz 96 (L96) model is a conceptual representation of geophysical turbulence that is widely used in numerical weather forecasting as a testbed for DA and parameterization (Lorenz, 1996; Lee and Majda, 2017; Wilks, 2005; Arnold et al., 2013). The model mimics a coarse discretization of atmospheric flow on a latitude circle, which exhibits complex wave-like and chaotic behavior. It illustrates the interactions between small-scale fluctuations and larger-scale motions schematically.

The noisy version of the model reads

$$\frac{du_i}{dt} = \left(-u_{i-1}\left(u_{i-2} - u_{i+1}\right) - u_i + f - \frac{hc}{J}\sum_{j=1}^{J} v_{i,j}\right) + \sigma_{u_i}\dot{W}_{u_i}, \quad i = 1, \ldots, I, \tag{2.12a}$$

$$\frac{dv_{i,j}}{dt} = \left(-bcv_{i,j+1}\left(v_{i,j+2} - v_{i,j-1}\right) - cv_{i,j} + \frac{hc}{J}u_i\right) + \sigma_{v_{i,j}}\dot{W}_{v_{i,j}}, \quad j = 1, \ldots, J, \tag{2.12b}$$

where I, J, f, h, c, b, σ_{u_i} and $\sigma_{v_{i,j}}$ are given parameters while $\dot{W}_{u_i}$ and $\dot{W}_{v_{i,j}}$ are white noise. The large-scale variables u_i are periodic in i with $u_{i+I} = u_{i-I} = u_i$. The corresponding small-scale variables $v_{i,j}$ are periodic in i with $v_{i+I,j} = v_{i-I,j} = v_{i,j}$ and satisfy the following conditions in j: $v_{i,j+J} = v_{i+1,j}$, and $v_{i,j-J} = v_{i-1,j}$. The model discussed here uses variables u_i to describe large-scale or slow movements which are resolved; small scales or rapid fluctuations represented by $v_{i,j}$ are often unresolved ones. The coupling of fast and slow variables is regulated by the parameter h. The parameter c specifies how quickly the fast variables are damped in comparison to the slow variables. The parameter f controls the magnitude of external large-scale forcing, while b determines the amplitude of nonlinear interactions between the fast variables. As in the standard L96 model, we take $I = 40$. There are $J = 4$ small-scale variables associated with each u_i. Thus, the total number of the state variables is 200. The constant forcing $f = 4$ makes the system to be chaotic. The parameters h, c, and b are chosen in such a way that

the small-scale variables have a comparatively significant impact on the large-scale ones. In other words, the perfect model only has a weak scale separation. The reason that we consider such a weak scale separation is that it better mimics the real atmosphere with chaotic/turbulent behavior, the associated forecast of which is often very difficult. Finally, additional stochastic noise is added to the system, representing the contribution of the variables that are not explicitly modeled. The noise also interacts with the deterministic part via nonlinear terms, introducing additional complexity that mimics nature. To summarize, the parameters used in the perfect model (2.12) are as follows,

$$I = 40, \quad J = 4, \quad h = 2, \quad c = 2, \quad b = 2, \quad f = 4, \quad \sigma_{u_i} = \sigma_u = 1, \quad \sigma_{v_{i,j}} = \sigma_v = 1. \tag{2.13}$$

With these parameters, the spatiotemporal patterns are shown in Figure 2.3, together with the time series of the variables at a fixed location $i = 1$, i.e., the large-scale variable u_1 and the small-scale variables $v_{1,1}, \ldots, v_{1,4}$, as well as the associated equilibrium PDFs and the autocorrelation functions (ACFs). The ACFs, which measure the memory of the system of each component, validate the weak scale separation adopted here. The average value of the ACF decay time, i.e., the decorrelation time, for large scales is 0.76.

2.3.2.2 The imperfect parametric models

Since the perfect knowledge of nature is never known or it is too complicated to be used in practice, approximate models with reduced dimensions or simpler struc-

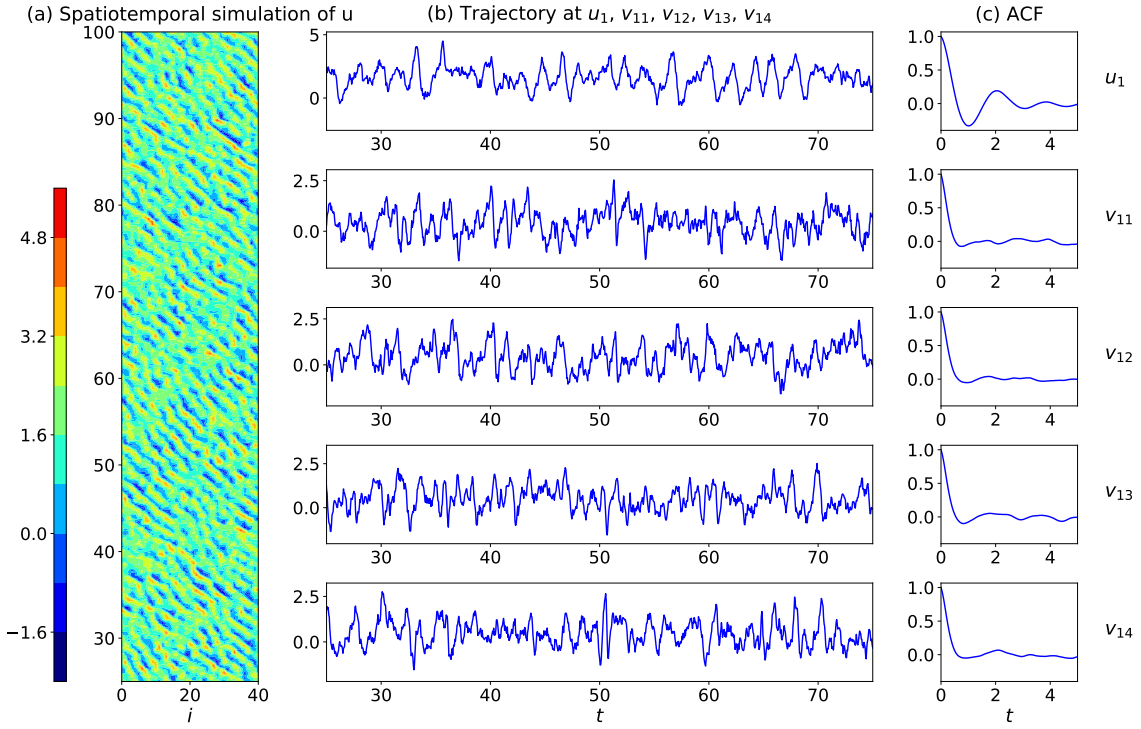

Figure 2.3: The two-layer L96 model (2.12) with parameters in (2.13). Panel (a): The spatiotemporal patterns of large-scale variables u_i. Panel (b): the time series of the large-scale variable u_1 and the associated small-scale variables $v_{11}, \ldots, v_{14}$. Panel (c): the ACFs corresponding to the variables in Panel (b).

tures are often utilized as the forecast models. Two imperfect (i.e., approximate) parametric models are introduced here, which are developed by applying two widely used approximation approaches as in many applications. The goal here is to make use of these imperfect models to predict the large-scale variables u_i at all grid points, i.e., for $i = 1, \ldots, I$.

The first imperfect model is the one-layer L96 (L96-1LYR) model, which is a reduced order model of the perfect system. It is also unknown as the bare truncation model, which is one of the simplest approximations by completely ignoring the

small-scale variables. The L96-1LYR model reads

$$\frac{du_i}{dt} = (-u_{i-1}(u_{i-2} - u_{i+1}) - u_i + f) + \sigma_{u_i}\dot{W}_{u_i}, \quad i = 1\ldots, I. \tag{2.14}$$

Bare truncation models are widely used in practice due to its reduced computational cost. In fact, the dimension of the L96-1LYR model (2.14) is only 40, which is 5x smaller than that of the 200-dimensional perfect model. However, the feedback from the small-scale variables $v_{i,j}$ to the large-scale variables u_i is non-negligible here as in many real-world applications, which is the main source of the model error. Note that the bare truncation models may sometimes suffer from a finite-time blowup issue (Majda and Yuan, 2012), which is however not a problem here as the model only contains advection and dissipation terms beyond a constant forcing.

The second imperfect parametric model is developed as follows. Instead of completely ignoring the small-scale variables, the equations of these unresolved scale variables are replaced by simple stochastic parameterized equations. The specific parameterization form adopted here follows the one that is widely-used in the stochastic parametrized extended Kalman filters (SPEKFs), which have been shown to be skillful for improving the DA and prediction skill (Gershgorin et al., 2010a,b; Majda and Harlim, 2012a). The stochastic parameterized imperfect model (L96-SP) has the following form:

$$\frac{du_i}{dt} = \left(-u_{i-1}\left(u_{i-2} - u_{i+1}\right) - u_i + f - \frac{hc}{J}\sum_{j=1}^{J} v_{i,j} \right) + \sigma_{u_i}\dot{W}_{u_i}, \quad i = 1, \ldots, I, \tag{2.15a}$$

$$\frac{dv_{i,j}}{dt} = -\hat{d}_{i,j}(v_{i,j} - \hat{v}_{i,j}) + \hat{\sigma}_{v_{i,j}}\dot{W}_{v_{i,j}}, \quad j = 1, \ldots, J. \tag{2.15b}$$

In (2.15), the unresolved small scales $v_{i,j}$ have been reduced to linear processes with only Gaussian additive noise providing statistically accurate feedbacks from the unresolved scales to the resolved ones. The parameters in (2.15b) can be calibrated by matching the mean, the variance, and the decorrelation time of $v_{i,j}$ with those in the perfect system, which provides the optimal Gaussian fit of each $v_{i,j}$ in (2.15b) with that in the perfect model (2.12). The comparison between the perfect model (2.12) and two imperfect models (2.14) and (3.29) is shown in Figure 2.4. It is clear that both the imperfect parametric models capture certain features of the perfect dynamics but the model errors are also obvious. The L96-SP model is more accurate than the L96-1LYR model, as is expected. Note that the random numbers generated from the noise sources $\dot{W}_{u_i}$ and $\dot{W}_{v_{i,j}}$ when performing the ensemble forecast using both the perfect and imperfect models are different from the truth, which is a realistic setup.

2.3.2.3 The experiment setup

In this experiment, the two imperfect models (i.e., the L96-1LYR model and the L96-SP model) discussed above will be utilized to provide the baseline of the ensemble

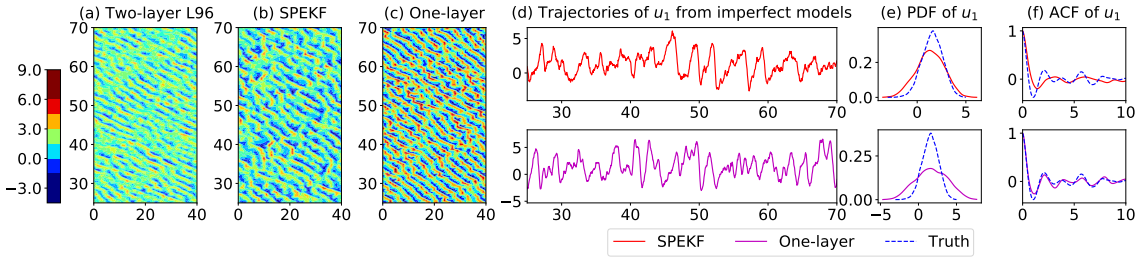

Figure 2.4: Comparison between the perfect model and two imperfect models associated with the two-layer L96 model. Panel (a)–(c): spatiotemporal patterns of the large-scale variables u_i. Panel (d)–(f): the time series, PDFs and ACFs of u_1, where the red curve shows those of the L96-SP (2.15), the magenta curve shows those of the L96-1LYR (2.14) and the blue dashed curve corresponds to those of the perfect model (2.12) as a reference.

forecast results. They will then be utilized as the prior models to create sampled trajectories for training two LSTM models within the BAMCAFE framework. We refer to the LSTM models trained from the sampled trajectories generated from the L96-1LYR model (2.14) and the one from the L96-SP model (2.15) as the LSTM-L96-1LYR and the LSTM-L96-SP, respectively.

The observations are adopted only for the large-scale variables and are only on the even grid points. In other words, $u_2, u_4, \ldots, u_{40}$ are the observed variables while $u_1, u_3, \ldots, u_{39}$ and all the small-scale variables have no observations. The observation noise is assumed to be Gaussian with zero mean and a standard deviation of 1. The total length of the observational sequence is 8000 (400 time units) and is separated into the training and the validation data sets with a ratio 9:1. The testing set is the next 1600 observation steps (from 400 to 480 time units). The LSTM models are trained and are applied for forecast for all the large-scale variables. Sensitivity tests have been performed, which justifies that 400 units are sufficiently

long to train the current one-layer LSTM models.

For the conciseness of presentation, the phrase "imperfect models" below always stands for the imperfect parametric models, i.e., the L96-1LYR model (2.14) and the L96-SP model (2.15). The NN models will always have a prefix 'LSTM'.

2.3.2.4 The prediction skill

The RMSE and the Corr of the predicted ensemble mean time series related to the truth as a function of the forecast lead time are shown in Figure 2.5. The skill scores averaged over all the unobserved large-scale variables (in panel (a)) and over all the observed variables (in panel (b)) are shown using the perfect L96 model, the imperfect L96-1LYR model, and the LSTM-L96-1LYR model. Comparison between the perfect L96 model, the imperfect L96-SP model, and the LSTM-L96-SP model is shown in panel (c)–(d). In each panel, prediction from the perfect model with perfect initial condition (the true values initially at each time) is shown as a reference. When the two different imperfect models are compared to their corresponding LSTM models, almost all the skill scores from the LSTM models outperform those from the imperfect models, owing to the model error being significantly reduced in the training data. It is important to note that the skill scores of the LSTM models are also quite close to those associated with the perfect model, where the useful prediction lasts for more than 1 unit that is slightly longer than the average decorrelation time. In contrast, the errors in the imperfect models increase quickly. The forecast of the observational variables are more skillful than those in the unobserved large-scale variables at short lead times because of the larger

initialization uncertainty of the latter. Besides, since the L96-1LYR model (2.14) completely ignores the damping effects of the small scales, the predicted values have larger amplitudes than the true values and the predictions have phase shifting tendencies.

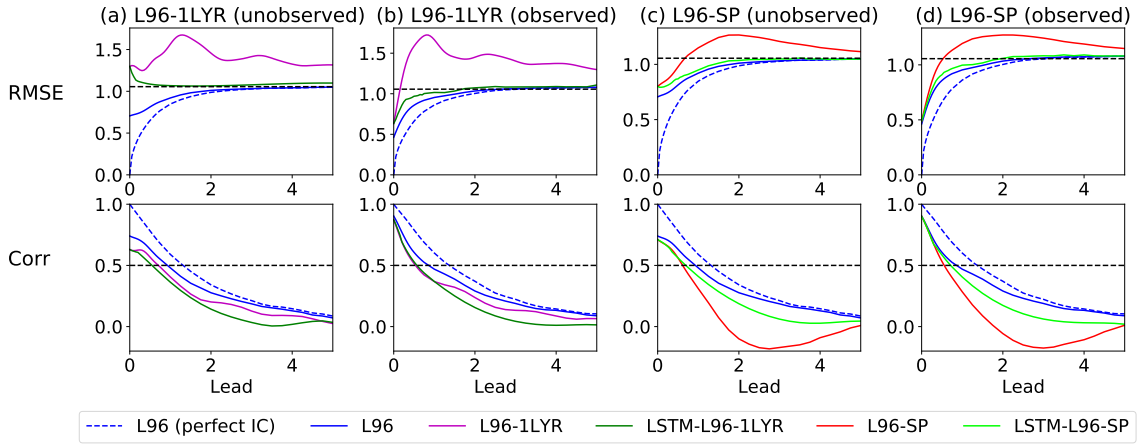

Figure 2.5: The RMSE and the Corr of the ensemble mean prediction as a function of lead time, where the true signal is generated from the two-layer L96 system (2.12). Panels (a)–(b): the skill scores averaged over all the 20 unobserved large-scale variables and those averaged over all the 20 observed variables using the one-layer L96 imperfect model (2.14) and the associated LSTM model. Panels (c)–(d): similar results but for the stochastic parameterized L96 model (2.15). The blue dashed curves indicate the ensemble prediction using perfect model from perfect initial conditions as a reference. The black dashed lines in the RMSE panels represent the one standard deviation of the true signal and those in the Corr panels show the Corr = 0.5 threshold.

Figure 2.6 depicts a case study in which the prediction begins at $t = 410$ with the assimilate initial condition. The ensemble prediction is run for three time units forward in each subfigure. The time evolution of the forecast spread illustrated here is constructed using 2500 ensemble members for all physics-based models. For the LSTM models, the ensemble is made use of 50 components where each

component is constructed by generating 50 samples from the distribution in (2.8). The black solid line is the ensemble mean and the dark and light shading areas show the one and two standard deviations of the uncertainty in the prediction, respectively. First, it justifies the results in Figure 2.5 that, compared to the imperfect models, the associated LSTM models can predict at longer lead times in terms of the ensemble mean forecast. For example, in Rows 2 and 3 of Panel (b), with the same initial condition, the ensemble mean prediction from the imperfect L96-1LYR model starts away from the truth quickly and ends up with larger uncertainty at $t = 413$. In contrast, the LSTM-L96-1LYR model can provide longer lead times for the ensemble mean prediction (nearly identical to the perfect model) and smaller uncertainty completely covering the true signal. Another example to illustrate this point is to compare Rows 4 and 5 of Panel (d). The LSTM-L96-SP model can predict the ascending trends from $t = 410.5$ to $t = 411$ as the perfect model while the imperfect L96-SP model decays immediately. Second, despite the possible initialization errors in the unobserved variables, the ensemble evolutions of the LSTM models can quickly adjust to follow those of the perfect model as is shown in Panel (c) row of the LSTM-L96-1LYR model, which is not the case for the associated imperfect L96-1LYR model. Third, although the ensemble mean predictions from the two LSTM models are similar, which can also be found in score curves in Figure 2.5, the prediction uncertainty for the unobserved variables in LSTM-L96-1LYR is slightly larger due to larger variance of the training data generated by the bare truncation model which totally ignores the small scales. Nevertheless, the forecast uncertainty in the LSTM-L96-1LYR has been significantly reduced

compared with the associated imperfect L96-1LYR model, indicating the success of the BAMCAFE in characterizing the forecast uncertainty even in the presence of a heavily biased imperfect parametric forecast model. The relative entropy skill scores (the last row) that describe the time evolutions of the forecast uncertainty of the imperfect and LSTM models related to the perfect model forecast also confirm the improvement of the LSTM models in quantifying the forecast uncertainty.

The time evolution of the validation errors using the LSTM-L96-SP model are illustrated in Figure 2.7, where the dashed and solid green curves show the averaged validation error over the unobserved and observed large-scale variables, respectively. Recall in Section 2.2.5 that the validation error is the forecast uncertainty of each ensemble member in the LSTM model. For comparison, the averaged forecast ensemble spread using the perfect model (2.12) is shown by the solid black curve. The 'average' here means averaging the forecast deviation from the truth based on multiple forecasts starting from different time instants in a very long trajectory and averaging over all the u_i. Perfect initial conditions are used in the perfect model ensemble forecast to exclude the uncertainty due to the initial ensemble spread. In such a way, both the LSTM-L96-SP and the perfect model illustrate the time evolution of the intrinsic uncertainty instead of the uncertainty resulting from the initializations. It is shown that the time evolutions of the validation errors of the LSTM-L96-SP model are very similar to the averaged forecast uncertainty using the perfect model. Specifically, the validation errors of the LSTM-L96-SP model increase as a function of time and converge to the one standard deviation of the true signal at the statistical equilibrium state. This means that the LSTM-L96-SP

model succeeds in reproducing the time evolution of the forecast uncertainty of the perfect model. The validation error in the unobserved large-scale variables is slightly larger than that in the observed ones. This is because the training data (i.e., the sampled trajectories) of the former contain slightly larger biases. Finally, despite the fact that Figure 2.7 only compares the validation errors in the LSTM model with the perfect model using the perfect initial conditions, the LSTM-L96-SP also has almost the identical uncertainty spread as the perfect model forecast when DA is applied to the initialization, which can be seen from the second last row of Figure 2.6.

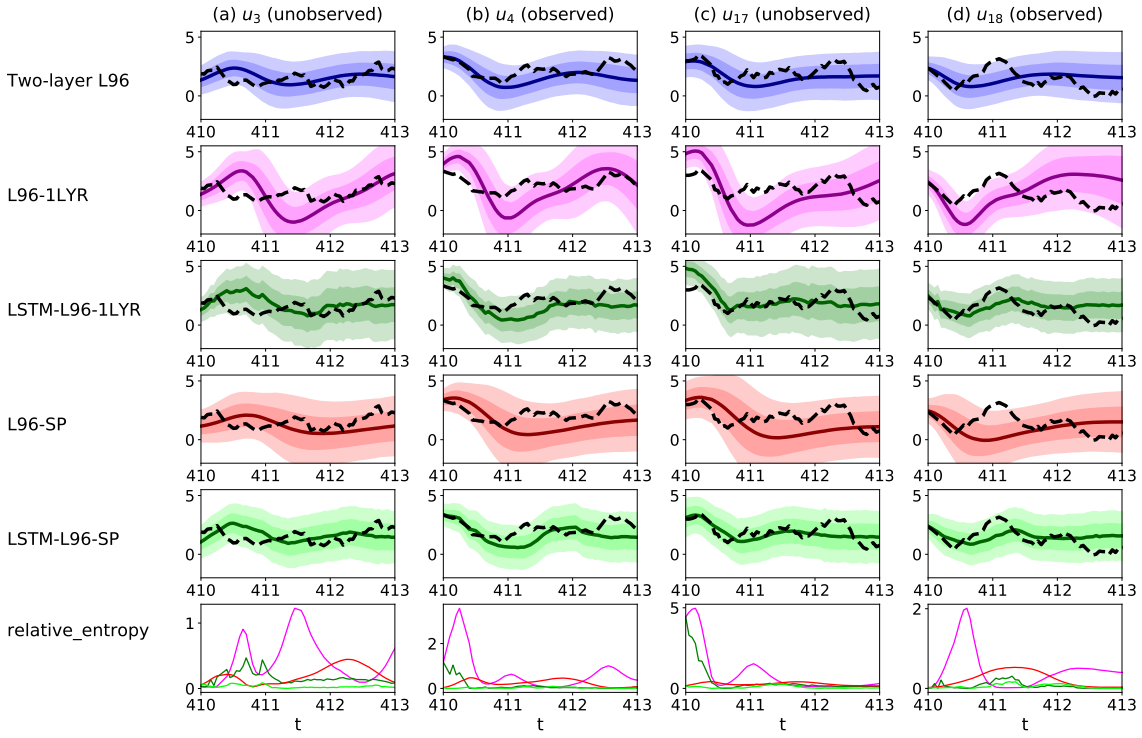

Figure 2.6: Ensemble forecasts of four large-scale variables of the two-layer L96 model. Columns (a) and (c) show the forecasts of the two unobserved variables u_3 and u_{17} while Columns (b) and (d) show those of the two observed variables u_4 and u_{18}. Different rows (except the last one) show the forecasts using different models with assimilated initial conditions starting from $t = 410$. The solid curves are ensemble mean for different models while the black dashed curve are the true trajectories. The dark and light shading areas show the one and two standard deviations of the uncertainty in the prediction. The last row show the time evolutions of the relative entropy between the perfect model ensemble forecast and the ensemble forecast using different imperfect parametric or LSTM models. The pink, dark green, red, and light green curves show the results using the L96-1LYR model, the LSTM-L96-1LYR model, the L96-SP model and the LSTM-L96-SP model, respectively.

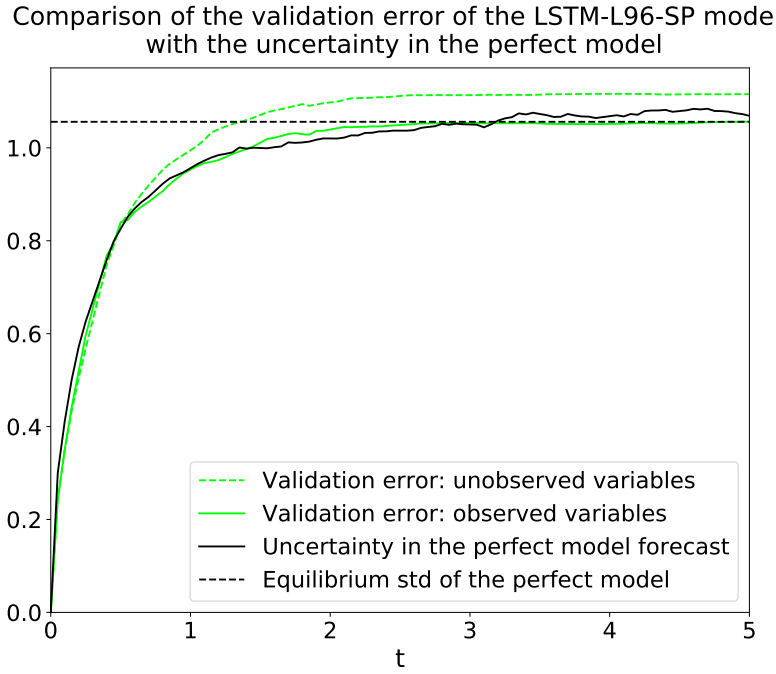

Figure 2.7: The time evolution of the validation errors using the LSTM-L96-SP model. The dashed and solid green curves show the averaged validation error over the unobserved and observed large-scale variables, respectively. For comparison, the averaged forecast ensemble spread using the perfect model (2.12) is shown by the solid black curve. The 'average' here means averaging the forecast deviation from the truth based on multiple forecasts starting from different time instants in a very long trajectory and averaging over all the u_i. Perfect initial conditions are used in the perfect model ensemble forecast to exclude the uncertainty due to the initial ensemble spread. The one standard deviation is used here to denote the averaged ensemble spread. The black dashed line is the one standard deviation of the forecast uncertainty at the statistical equilibrium state of the perfect model.

2.3.3 Quantifying the forecast uncertainty in a conceptual nonlinear model with strongly non-Gaussian features

The purpose of this subsection is to demonstrate the skill of the BAMCAFE algorithm in quantifying the forecast uncertainty in a strongly turbulent systems with intermittency and extreme events.

2.3.3.1 The perfect model

The perfect model used here is the a nonlinear triad model with energy-conserving nonlinear interaction (Majda, 2015; Chen and Majda, 2016a),

$$\frac{du_1}{dt} = \big(-\gamma_1 u_1 + L_{12}u_2 + L_{13}u_3 + I u_1 u_2 + F \big) + \sigma_1 \dot{W}_1, \tag{2.16a}$$

$$\frac{du_2}{dt} = \big(-L_{12}u_1 - \frac{\gamma_2}{\delta}u_2 + L_{23}u_3 - I u_1^2 \big) + \frac{\sigma_2}{\delta^{1/2}}\dot{W}_2, \tag{2.16b}$$

$$\frac{du_3}{dt} = \big(-L_{13}u_1 - L_{23}u_2 - \frac{\gamma_3}{\delta}u_3 \big) + \frac{\sigma_3}{\delta^{1/2}}\dot{W}_3, \tag{2.16c}$$

where γ_i, L_{12}, L_{13}, L_{23}, I, F, σ_i, and δ are given constants while $\dot{W}_i$ are independent white noise. This nonlinear triad system is a simple prototype nonlinear stochastic model that mimics structural features of low-frequency variability of general circulation models with non-Gaussian features (Majda et al., 2009) and it was used to test the skill for reduced nonlinear stochastic models for fluctuation dissipation theorem (Majda et al., 2010). The triad model (2.16) involves a quadratic nonlinear interaction between u_1 and u_2 with energy-conserving property that induces intermittent instabilities. On the other hand, the coupling between u_2 and u_3 is linear and is through the skew-symmetric term with coefficient $-L_{23}$, which represents

an oscillation structure of u_2 and u_3.

The parameter δ controls the scale separation of the system. If $\delta \ll 1$, then (2.16) becomes a slow-fast system. If δ is of order $O(1)$, then all the three variables lie in a similar time scale. This model has been widely used as a testbed for the DA and forecast of complex turbulent systems (Majda and Chen, 2018; Chen and Majda, 2018b, 2016a). The following parameters are adopted for the nonlinear triad model (2.17) in this experiment,

$$\gamma_1 = 2, \quad \gamma_2 = 0.2, \quad \gamma_3 = 0.4, \quad L_{12} = 0.2, \quad L_{13} = 0.1 \quad L_{23} = 0, \quad I = 5, \quad \delta = 1$$
$$\sigma_1 = 0.5, \quad \sigma_2 = 1.2, \quad \sigma_3 = 0.8, \quad F = 2 \tag{2.17}$$

The blue curves in Figure 2.8 show one realization of the model where the model trajectories are given in Panel (a) and the equilibrium PDFs are given in Panel (b). The black dashed curves in Panel (b) are the Gaussian fits of the true equilibrium distributions. It is clear that strong intermittency and extreme events in the time series of u_1 lead to a highly non-Gaussian PDF with an one-sided fat tail. The strong nonlinear feedback from u_1 to u_2 also introduces a skewed PDF of u_2.

2.3.3.2 The imperfect model

In various applications (Majda et al., 2009, 2010; Chen and Majda, 2018b) , it is often assumed that u_1 is the observed variables while u_2 and u_3 represent the unresolved processes. Therefore, approximate models are developed, which includes simple parameterizations for these two unresolved processes. One natural way of proposing an imperfect approximate model is to use linear stochastic processes

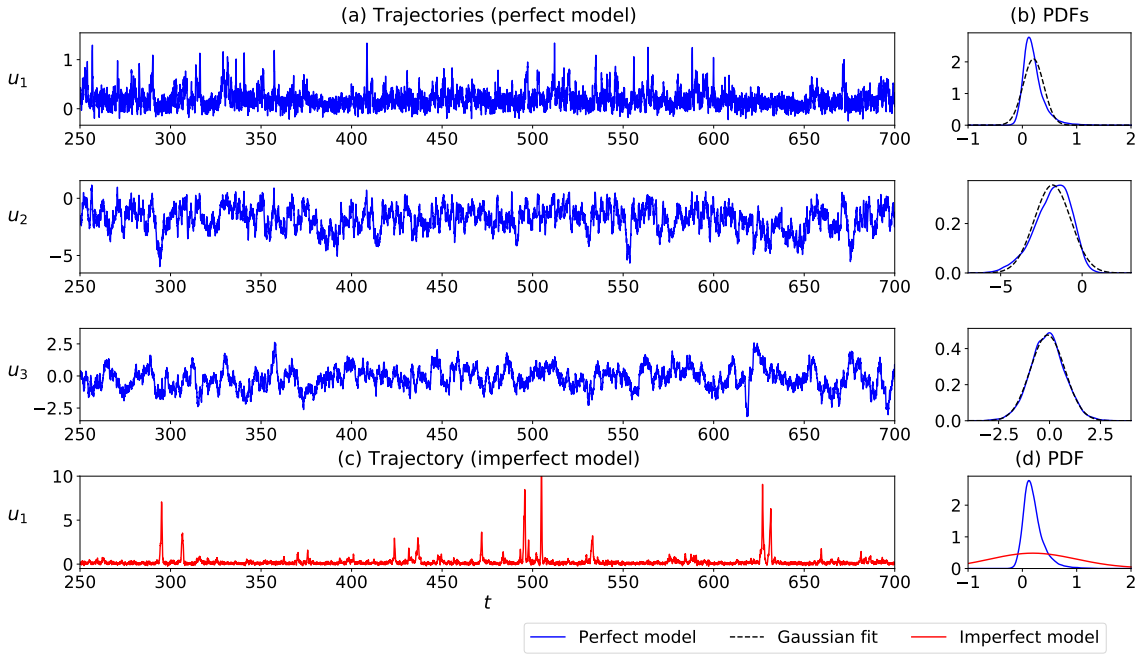

Figure 2.8: Panel (a): A realization of the triad model (2.16) with the parameters in (2.17). Panel (b): the associated equilibrium PDFs, where the black dashed curves are the Gaussian fits of the true PDFs. Panel (c)–(d): A realization and the associated PDF of u_1 from the imperfect model (2.18).

to replace the nonlinear dynamics of u_2 and u_3 (Majda and Harlim, 2012a). The resulting model reads,

$$\frac{du_1}{dt} = \big(-\gamma_1 u_1 + L_{12}u_2 + L_{13}u_3 + I u_1 u_2 + F\big) + \sigma_1 \dot{W}_1, \tag{2.18a}$$

$$\frac{du_2}{dt} = -d^M_{u_2}(v - \bar{u}^M_2) + \sigma^M_{u_2}\dot{W}_{u_2}, \tag{2.18b}$$

$$\frac{du_3}{dt} = -d^M_{u_3}(v - \bar{u}^M_3) + \sigma^M_{u_3}\dot{W}_{u_3}, \tag{2.18c}$$

Assume that the parameters in (2.18a) for the observed variable u_1 are the same as those in the perfect model (2.16) while the parameters in (2.18b)–(2.18c) for the

unobserved variables u_2 and u_3 are calibrated by matching the mean, variance, and decorrelation time with those in (2.16). Note that due to violation of the energy-conserving constraint of the nonlinear terms in (2.18), the amplitude of u_1 from the imperfect model is much larger than the one from the perfect model. See Panel (c) in Figure 2.8. The pathological behavior associated with such a model error exists in many ad hoc data-driven statistical models for time series of partial observations of nature (Majda and Yuan, 2012). In addition, the PDF of u_2 in the imperfect model (2.18) is Gaussian by design, which is also different from the skewed PDF as in the perfect model.

2.3.3.3 The experiment setup

In this experiment, u_1 is the observed variable while u_2 and u_3 are unobserved in the triad model (2.16). The standard deviation of the observational noise is 0.2. The total length of the observational sequence is 16,000 (800 time units) and is separated into training and validation data set with a ratio 9:1. The testing set is the next 1,600 observational steps (from 800 to 880 time units). The LSTM model is trained based on 50 sampled trajectories of u_1, u_2, and u_3.

2.3.3.4 The prediction skill

Before discussing the prediction skill using the imperfect and the LSTM model, we start with showing the sampled trajectories and the posterior mean time series, which is the typical outcome from the traditional reanalysis techniques. The purpose here is to illustrate that the latter fails to capture the basic dynamical and

statistical features in this tough test experiment. Therefore, the ML prediction can be biased and result in an inaccurate uncertainty quantification if it is trained on the posterior mean time series. The smoother posterior mean time series (orange curves) are shown in Panel (a) of Figure 2.9. It is clear that the dynamical behavior of the posterior mean time series is quite different from the true signals. As is shown in Panel (b), the large error in the PDFs, especially for u_2 and u_3, is as expected since the posterior mean time series smoothed out all the fluctuations that are also crucial for characterizing the underlying dynamics. In contrast, the sampled trajectories (green curves) resemble the truth in terms of both the dynamical and statistical features. Furthermore, they are very different from a free run of the imperfect model, as the observations serve as a regularizer in the Bayesian sampling process that completely resolved the instability problem that occurs in the imperfect model simulations.

In Figure 2.10, the time evolutions of the predicted PDFs for u_1 and u_2 starting from $t = 805$ with the assimilated initial conditions are shown in Panels (a) and (c), respectively. For ensemble forecast using both the perfect and the imperfect parametric models, each PDF is constructed using 2500 ensemble members. The PDFs from the LSTM forecast model are constructed by 50 non-Gaussian mixture components where each component is formed by 50 points from the distribution (2.8). It is clear that, as time increases, the PDFs of u_1 constructed from the imperfect model become much more fat-tailed than those from the perfect model. This is because the nonlinear feedback $-Iu_1^2$ is dropped in the imperfect model (2.18) such that the energy-conserving nonlinear constraint is broken. The consequence is that u_1

can stay in the unstable phase for a longer time, which triggers extremely large amplitudes of u_1 intermittently. On the other hand, since u_2 is approximated by a linear Gaussian process in the imperfect model (2.18), the PDF of u_2 of the imperfect model can never reflect any non-Gaussian information such as the skewness in the truth. In contrast, the PDFs constructed from the LSTM model are closer to the true PDFs since the training data is improved by the Bayesian sampling in the BAMCAFE algorithm. The skewness for the last row of u_2 from the perfect model, the imperfect model, and the LSTM model are -0.581, -0.035, -0.366, respectively.

Panels (b) and (d) of Figure 2.10 show 5 out of the 50 components of the mixture distribution, corresponding to the 1st, 16th, 50th, 84th, 99th percentile of the point-wise forecast values i.e., $\hat{x}^k_{N+l}$ in (2.7), among all the mixture components. The black cross marks are these point-wise forecast values. These mixture components show that the uncertainty of the prediction comes from two places. One is from the spread of all the mixture components. The other is from the LSTM model's uncertainty, quantified by the variability of each mixture component that is computed in light of the validation error (2.8). When the forecast lead time is small, the uncertainty of the LSTM model is tiny. In such a situation, the spread of the mixture components due to the use of DA for initialization is the dominant contribution to the uncertainty. On the other hand, as the forecast lead time increases, different mixture components gradually converge to each other and the uncertainty is explained more and more by the single LSTM model's forecast error (2.8).

Finally, since in traditional reanalysis, the smoother mean is usually used as a surrogate of the true signal, we compare the performance of the LSTM model trained

from the sampled trajectories and the LSTM model trained from the smoother mean (generated from the same period of observations) in Figure 2.11. It is clear that the PDFs constructed from the LSTM model trained from the smoother mean time series strongly underestimates the uncertainty. The fundamental reason is that the posterior mean time series has much less variability than the true signal, which has been shown in Figure 2.9. It is also worthwhile to note that the long-term forecast uncertainty should be the same as the equilibrium PDF of the training time series. Yet, the orange curve of u_2 at $t = 809$ in Figure 2.11, which corresponds to the forecast at the lead time of 4 units that is much longer than the decorrelation time, does not equal to the equilibrium PDF of u_2 associated with the posterior mean in Figure 2.9. Such an error probably comes from the insufficient number of the data in the training set. In fact, if the posterior mean time series is used as training, then only one time series is contained in the training data set. In contrast, using multiple sampled trajectories effectively increases the size of the training data, which facilitate the training of the ML.

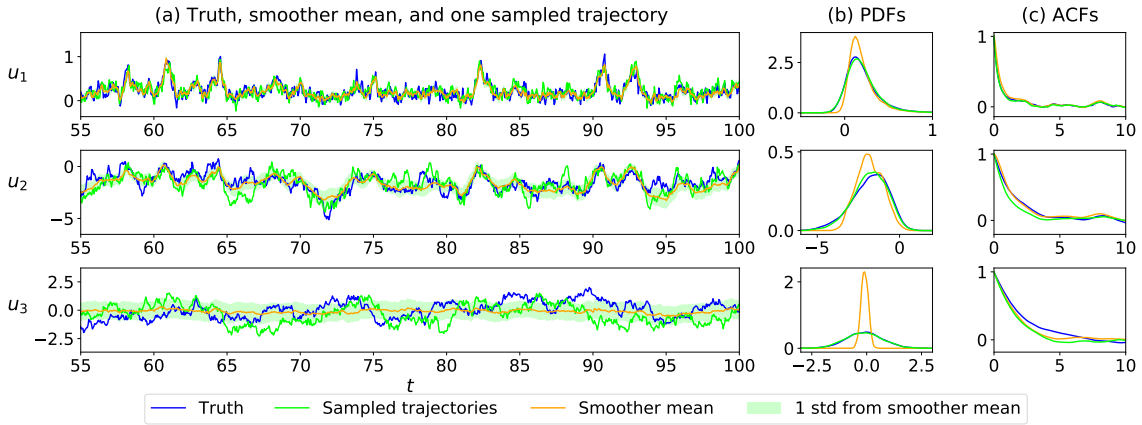

Figure 2.9: Comparison of the true signal from the triad model (2.16) with the smoother mean and the sampled trajectories based on the imperfect model (2.18). Blue curves show the true trajectories, the associated PDFs and ACFs. The orange curves show the posterior mean time series from the EnKS. The green curves show one sampled trajectories. The uncertainty of the smoother estimates, represented by the one standard deviation, is shown by the green shading areas. Panel (a): the time series of true signal, smoother mean, and one sampled trajectory. Panel (b)-(c): the PDFs and ACFs of the true signal, the smoother mean, and the sampled trajectory.

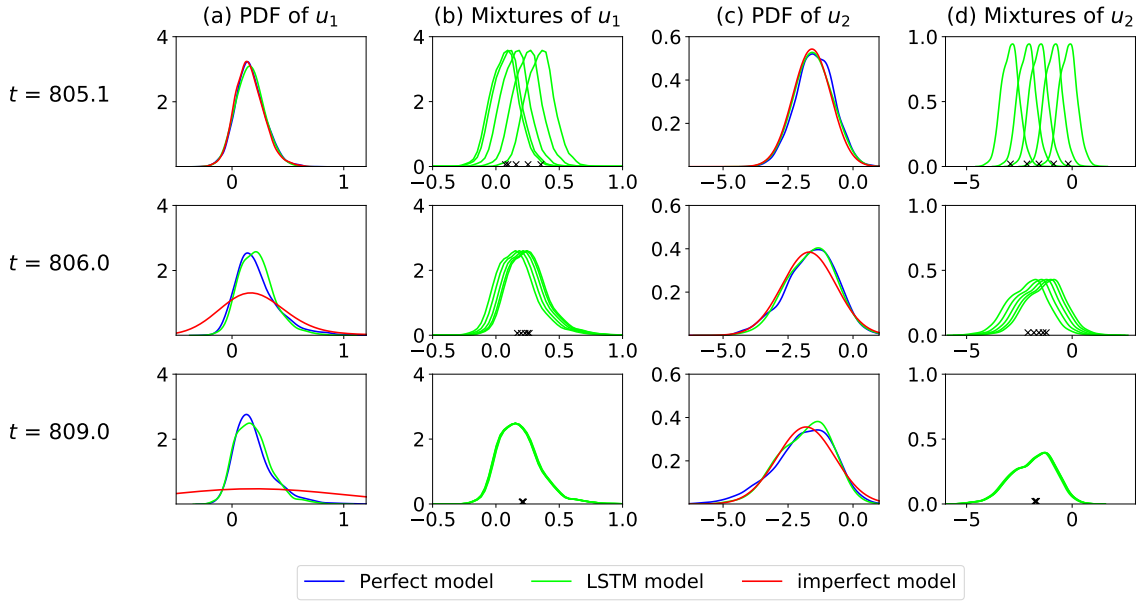

Figure 2.10: The time evolutions of the forecast uncertainty starting from $t = 805$, where the true signal is generated from the triad model (2.16). In Panel (a), the forecast PDFs of u_1 from the perfect model (2.16) and from the imperfect model (2.18) are constructed using 2500 ensemble members whereas the PDFs from the LSTM model are constructed by 50 non-Gaussian mixture components where each component is formed by 50 points from the distribution (2.8). Panel (b) shows 5 out of the 50 components of the mixture distribution, corresponding to the 1st, 16th, 50th, 84th, 99th percentile of the point-wise forecast values i.e., $\hat{x}^k_{N+l}$ in (2.7), among all the mixture components. The black cross-marks are these point-wise forecast values. Panel (c)–(d): similar to Panel (a)–(b) but for u_2.

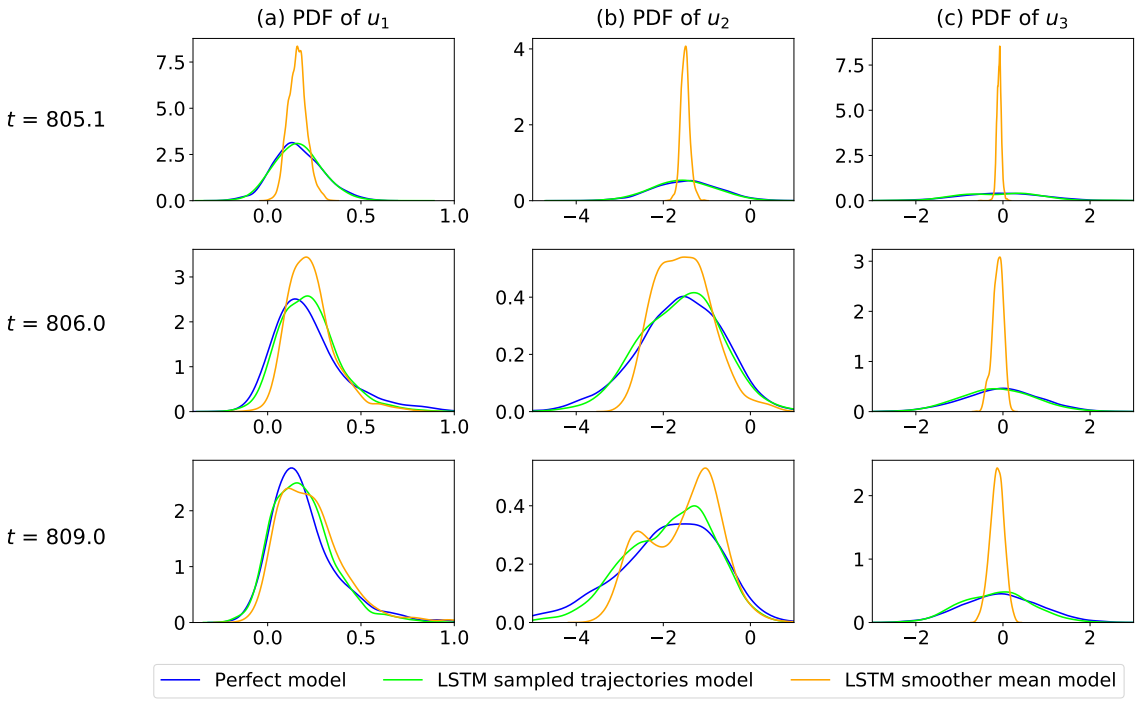

Figure 2.11: Comparison of the time evolutions of the forecast uncertainty starting from t = 805 using different methods, where the true signal is generated from the triad model (2.16). Panel (a)–(c): PDFs of u_1, u_2, and u_3, respectively. The blue curves are the forecasts using the perfect model (2.16). The green curves are those using the LSTM model trained from sampled trajectories, as in Figure 2.10. The pink curves are the forecasts using the LSTM model but is trained from the smoother mean time series.

2.4 Discussions and conclusion

A simple but effective Bayesian machine learning advanced forecast ensemble (BAMCAFE) method is developed that combines an available imperfect physics-informed model with DA to facilitate the ML ensemble forecast. In the BAMCAFE framework, a Bayesian ensemble DA is applied to create the training data of the ML model, which reduces the intrinsic error in the imperfect physics-informed model and provides the training data of the unobserved variables. Then a generalized DA is employed for the initialization of the ML ensemble forecast. In addition to providing the optimal point-wise forecast value, the BAMCAFE also improves the accuracy of quantifying the forecast uncertainty utilizing a non-Gaussian probability density function that characterizes the intermittency and extreme events. Furthermore, one advantage of using ML model to forecast is efficiency. For example, one can generate 2500-ensemble members of 5-time units forecast in about 30 minutes using the perfect L96 model on a server with a 64-core CPU, 256G memory. In contrast, it only takes a few seconds to generate the same forecast using new LSTM-based models.

Compared to the past studies that have already combined DA and ML, the novelties of this paper are as follows.

1. An effective offline DA is applied to create an improved data set compared with the simulations from the available imperfect model. The new data set contains a recovery of the unobserved state variables. It also mitigates the model error. The augmented data is used as the training data for the ML models.

2. Different from the traditional re-analysis technique, where only the optimal point-wise state estimation is often retained, the trajectories of all the ensemble members are utilized here as the ML training data. These sampled trajectories contain crucial temporal information of the underlying dynamics and the multiple trajectories facilitate a relatively short ML training period.
3. A generalized DA is developed for the ML forecast using only partial observations.
4. The forecast uncertainty is characterized in light of a non-Gaussian PDF via a mixture distribution.

Future work includes the theoretical studies of the BAMCAFE algorithm, especially the quantification of the uncertainty, and applying the BAMCAFE algorithm to more complicated and realistic systems, such as those for numerical weather forecasting. As is expected, there will be some additional challenges of the latter. One such challenge is the high dimensionality of the system, which can affect the accuracy of the ensemble data assimilation results. Natural remedies include incorporating the localization and noise inflation techniques (Evensen, 2009) into the ensemble data assimilation schemes or developing suitable approximate data assimilation algorithms that exploit semi-analytic formulae for the state estimation (Chen and Majda, 2018a). Another difficulty for dealing with more realistic applications is the development of suitable approximate models. Skillful stochastic parameterizations and systematic model reduction strategies facilitate the application of the BAMCAFE. More sophisticated ML model architectures and training techniques, such as the mixed-scale dense convolutional neural network (Pelt and

Sethian, 2018), are also required for the BAMCAFE to forecast more efficiently and accurately.

3 CONDITIONAL GAUSSIAN NONLINEAR SYSTEM: A FAST PRECONDITIONER AND A CHEAP SURROGATE MODEL FOR COMPLEX NONLINEAR SYSTEMS

3.1 Introduction

Complex nonlinear systems are ubiquitous in many areas, including geophysics, climate science, engineering, neuroscience and material science (Vallis, 2017; Strogatz, 2018; Wilcox, 1988; Sheard and Mostashari, 2009; Ghil and Childress, 2012). Mathematical modeling plays an important role in characterizing and discovering the underlying physics of these complex systems (Edwards, 2011; Majda, 2012). With suitable mathematical models in hand, effective parameter inference, state estimation and data assimilation then become fundamental tasks that serve as the prerequisites for analyzing these systems (Asch et al., 2016; Kalnay, 2003; Majda and Harlim, 2012a; Law et al., 2015; Ghil and Malanotte-Rizzoli, 1991). Accurate forecast of future states and successful prediction of the system response due to external perturbations are topics that have many practical implications (Lucarini et al., 2017; Ragone et al., 2016; Majda et al., 2010; Toth and Kalnay, 1997; Leutbecher and Palmer, 2008).

However, there exist quite a few mathematical and computational challenges in analyzing and simulating complex nonlinear systems. First, the intrinsic nonlinearity in these complex nonlinear systems often triggers strongly chaotic or turbulent behavior (Salmon, 1998; Dijkstra, 2013; Palmer, 1993). As a consequence, inter-

mittency, extreme events and non-Gaussian probability density functions (PDFs) are some of the typical features in these systems (Farazmand and Sapsis, 2019; Trenberth et al., 2015; Moffatt, 2021; Majda, 2003; Manneville and Pomeau, 1979), which impede the use of many traditional mathematical tools to analyze the model properties. Second, due to the nonlinear interactions between state variables across different scales, many of these complex nonlinear systems are high dimensional and have multiscale spatiotemporal structures (Wilcox, 1988; Majda, 2016; Tao et al., 2009; Majda and Grooms, 2014). Therefore, developing new efficient numerical algorithms to accelerate the computational efficiency becomes essential. Third, it is often the case in practice that only partial observations of the state variables are available (Kalnay, 2003; Lau and Waliser, 2011), which result in additional difficulties for model calibration, state estimation and prediction where systematic uncertainty quantification needs to be addressed (Curry and Webster, 2011; DelSole, 2004; Edwards, 1999; Majda and Branicki, 2012; Majda and Chen, 2018).

Since many complex dynamical systems are too expensive to be handled directly, it is of practical importance to develop suitable approximate models, which capture certain features of nature and are easier to deal with. Linear regression models are arguably the simplest class of approximate models (Yan and Su, 2009; Freedman, 2009), which can already provide certain skill for short-term forecasts although they usually suffer in characterizing the underlying nonlinear physics. Physics-constrained regression models are a set of nonlinear approximate models (Majda and Harlim, 2012b; Harlim et al., 2014; Kondrashov et al., 2015), which take into account the energy conserving nonlinear interactions in the model development

that guarantees the well-posednss of long-term behavior of the system. Another commonly used approach to developing approximate models is to project the starting complex nonlinear system to the leading a few energetic modes in light of the Galerkin proper orthogonal decomposition methods (Holmes et al., 1996) or other empirical basis functions such as the principal interaction patterns (Hasselmann, 1988; Kwasniok, 1996) and the dynamic mode decomposition (Rowley et al., 2009; Schmid, 2010). With a careful design of the closure terms to compensate the truncation error, these reduced order models are skillful in resolving certain problems in fluids and turbulence (Carlberg et al., 2013; Noack et al., 2011; Taira et al., 2020; Xie et al., 2018). On the other hand, many data-driven approximate modeling strategies have recently been developed (Ahmed et al., 2021; Chekroun and Kondrashov, 2017; Lin and Lu, 2021; Mou et al., 2021a; Peherstorfer and Willcox, 2015; Hijazi et al., 2020; Smarra et al., 2018). One popular method is the sparse identification of nonlinear dynamical systems (SINDy) (Brunton et al., 2016), which is a set of nonlinear regression models that take the advantage of an automatic regularization to retain a parsimonious model structure. Other approximate modeling approaches have been designed for specific scientific purposes. For example, the past noise forecasting method (Chekroun et al., 2011) was developed as an approximate forecast model for stochastic systems. Reduced-space Gaussian process regression forecast (Wan and Sapsis, 2017) was designed for data-driven probabilistic forecast of chaotic dynamical systems. Small-scale parameterization based on a data-informed optimal homotopic deformation of invariant manifolds was developed to design low-dimensional models of the large-scale variables for both

deterministic chaotic systems and stochastic systems (Chekroun et al., 2015, 2020). Physically consistent data-driven weather forecasting techniques were proposed and applied to operational models (Chattopadhyay et al., 2020b, 2021). In addition, machine learning methods nowadays have been extensively incorporated into the reduced order models to further improve the approximation and forecast skill (San and Maulik, 2018; Chattopadhyay et al., 2020c; Chen and Li, 2021; Pawar et al., 2020; Moosavi et al., 2015).

The objective of this paper is to explore the skill of a rich class of nonlinear stochastic models, known as the "conditional Gaussian nonlinear system" (CGNS) (Chen and Majda, 2018a), as approximate models for complex nonlinear systems. The CGNS includes many physics-constrained nonlinear stochastic models (e.g., the stochastic versions of various Lorenz models, low-order models of Charney-DeVore flows, and a paradigm model for topographic mean flow interaction), quite a few stochastically coupled reaction-diffusion models in neuroscience and ecology (e.g., stochastically coupled FitzHugh-Nagumo (FHN) models and stochastically coupled SIR epidemic models), and several large-scale dynamical models in engineering and geophysical flows (e.g., the Boussinesq equations with noise and stochastically forced rotating shallow water equation). See the article (Chen and Majda, 2018a) for a gallery of examples of the CGNS. The CGNS has also been applied to modeling and forecasting several important climate phenomena, such as the Madden-Julian oscillation and the monsoon (Chen et al., 2014a, 2018a), and has been utilized for Lagrangian data assimilation (Chen et al., 2014b). Yet, most of the previous work focused on perfect model scenarios, where utilizing

the CGNS as an approximate model has not been systematically studied.

The CGNS has several unique features that allows it to be distinct from many existing approximate modeling strategies. First, the CGNS aims at preserving the underlying physical mechanism to the greatest extent. Specifically, the nonlinearity involving the large-scale or slow variables are by design retained, which includes not only self-interactions among the large-scale variables but also cross-scale interactions between large- and small-scale variables, while suitable approximations are imposed primarily on the nonlinear self-interactions between small-scale, fast-varying or unresolved state variables. This is fundamentally different from many purely data-driven nonlinear regression models, which may miss certain crucial underlying physics of the original complex nonlinear systems. Second, the self interactions between small-scale variables are effectively parameterized by stochastic noise, which leads to the important feature that the distribution of the small-scale variables conditioned on the large-scale ones is Gaussian. One remarkable consequence is that the associated conditional Gaussian distribution can be calculated using closed analytic formulae (Liptser and Shiryaev, 2013), which considerably facilitate the mathematical analysis and numerical simulations of the CGNS. In fact, the closed analytic formulae of the conditional Gaussian distributions allow the development of efficient and statistically accurate algorithms for parameter estimation, data assimilation and ensemble forecast in light of only partial observations. Note that, despite the conditional Gaussianity, the joint and marginal distributions of the CGNS remain highly non-Gaussian. Thus, the intermittency, extreme events and turbulent features can all be preserved in a suitably designed CGNS. Third,

the CGNS is also adaptable to many data-driven scenarios. Physics constraints, localizations and sparse identification together with many other mathematical and computational strategies can be possibly incorporated into the CGNS. Finally, information theory (Kleeman, 2011; Majda et al., 2005) can be applied to quantify the uncertainty and the statistical error of the CGNS in approximating the original complex nonlinear systems.

The specific goal of this paper is two-fold. First, by taking advantage of its analytically solvable properties, the CGNS can be served as a fast preconditioner for facilitating many computationally challenging tasks associated with the original complex nonlinear system. Important applications include estimating the parameters of the original system in the presence of only partial observations and recovering the non-Gaussian PDFs as a crucial intermediate step for computing the model sensitivity and response. Second, the CGNS is exploited as a surrogate model, aiming at spending a much lower computational cost to create comparably accurate results as those obtained from the original complex system. This includes for example the state estimation of unobserved variables and the statistical forecast.

The rest of the paper is organized as follows. The general mathematical framework of CGNS and its analytic properties are described in Section 3.2. Several systematic strategies for the development of the CGNS are included in Section 3.3. Sections 3.4 – 3.6 consist of three important tasks in complex nonlinear systems, showing the roles of the CGNS both as a surrogate model and a preconditioner for the original complex system. Specifically, Section 3.4 focuses on the data assimilation and ensemble forecast, Section 3.5 aims at efficient parameter estimation, and

Section 3.6 illustrates the use of CGNS in facilitating the study of model sensitivity and response theory. The paper is concluded in Section 3.7.

3.2 General Mathematical Framework of the CGNS

Let us start with the general formulation of the turbulent dynamical systems motivated from fluid applications (Vallis, 2017; Salmon, 1998; Kalnay, 2003; Majda, 2016),

$$\frac{d\mathbf{u}}{dt} = (L + D)\,\mathbf{u} + B(\mathbf{u}, \mathbf{u}) + \mathbf{F}(t) + \boldsymbol{\sigma}(\mathbf{u}, t)\,\dot{\mathbf{W}}(t), \tag{3.1}$$

where the state variable $\mathbf{u} \in \mathbb{C}^N$ is in a high dimensional phase space. In (3.1), the first two components, $(L + D)\,\mathbf{u}$, represent linear dispersion and dissipation effects, where $L^* = -L$ is a skew-symmetric operator; and D is a negative-definite matrix. The nonlinear effect is introduced through an energy-conserving quadratic form, $B(\mathbf{u}, \mathbf{u})$. Besides, the system is subject to external forcing effects that are decomposed into a deterministic component, $\mathbf{F}(t)$, and a stochastic component represented by a Gaussian random process, $\boldsymbol{\sigma}(\mathbf{u}, t)\,\dot{\mathbf{W}}(t)$, where $\boldsymbol{\sigma} \in \mathbb{R}^{N\times K}$ is the noise matrix and $\dot{\mathbf{W}} \in \mathbb{R}^K$ is the white noise. The two components $(L + D)\,\mathbf{u} + B(\mathbf{u}, \mathbf{u}) + \mathbf{F}(t)$ and $\boldsymbol{\sigma}(\mathbf{u}, t)$ on the right hand side of (3.1) are also known as the drift part and the diffusion coefficients, respectively.

3.2.1 The CGNS

Despite being highly nonlinear and possessing strongly non-Gaussian statistics in both the marginal and joint distributions of the state $\mathbf{u}$, many complex nonlinear

dynamical systems (3.1) have or can be approximated by the following nonlinear system with conditional Gaussian structures. The general mathematical framework of the CGNS is as follows (Liptser and Shiryaev, 2013; Chen and Majda, 2016a, 2018a),

$$\frac{d\mathbf{X}}{dt} = \Big[\mathbf{A_0}(\mathbf{X},t) + \mathbf{A_1}(\mathbf{X},t)\mathbf{Y}(t)\Big] + \mathbf{B_1}(\mathbf{X},t)\dot{\mathbf{W}}_\mathbf{1}(t), \tag{3.2a}$$

$$\frac{d\mathbf{Y}}{dt} = \Big[\mathbf{a_0}(\mathbf{X},t) + \mathbf{a_1}(\mathbf{X},t)\mathbf{Y}(t)\Big] + \mathbf{b_2}(\mathbf{X},t)\dot{\mathbf{W}}_\mathbf{2}(t), \tag{3.2b}$$

where the original model state $\mathbf{u}$ is decomposed into multi-dimensional state variables $\mathbf{X} \in \mathbb{C}^{N_1}$ and $\mathbf{Y} \in \mathbb{C}^{N_2}$, with $N_1 + N_2 = N$. In (3.2), $\mathbf{A}_0, \mathbf{a}_0, \mathbf{A}_1, \mathbf{a}_1, \mathbf{B}_1$ and $\mathbf{b}_2$ are vectors or matrices that can depend nonlinearly on the state variables $\mathbf{X}$ and time t while $\dot{\mathbf{W}}_\mathbf{1}$ and $\dot{\mathbf{W}}_\mathbf{2}$ are independent white noise sources.

The name "conditional Gaussian" comes from the fact that once a time series of $\mathbf{X}(s)$ for $s \leqslant t$ is given, then the conditional distribution $p(\mathbf{Y}(t)|\mathbf{X}(s \leqslant t))$ is Gaussian. This can be seen by noticing that, with a given $\mathbf{X}$, the process of $\mathbf{Y}$ is linear (with respect to the variable $\mathbf{Y}$ itself since $\mathbf{X}$ has been given) with Gaussian white noises. It is worthwhile to highlight that, from the general form of the complex turbulent system (3.1) to the CGNS (3.2), the nonlinear interactions between $\mathbf{X}$ itself and those between $\mathbf{X}$ and $\mathbf{Y}$ in (3.1) are both completely retained. The only simplification in the CGNS is to approximate the nonlinear self interactions between $\mathbf{Y}$ by a combination of nonlinear functions of $\mathbf{X}$, conditional linear functions of $\mathbf{Y}$ and effective stochastic noises. Nevertheless, if $\mathbf{Y}$ represents small-scale or fast variables, then such a manipulation is expected to be an effective approximation

that preserves the underlying physics to a large extent.

It should be noted that the CGNS in (3.2) is still highly nonlinear due to the nonlinearity in $\mathbf{A}_0, \mathbf{a}_0, \mathbf{A}_1$ and $\mathbf{a}_1$ as well as the nonlinear coupling between the latter two with $\mathbf{Y}$. Such nonlinearities preserve the non-Gaussian statistics in (3.1) and allow to reproduce many observed features in nature such as extreme events with the more tractable conditional Gaussian structure. A gallery of examples of the CGNS, including many physics-constrained nonlinear stochastic models, quite a few stochastically coupled reaction-diffusion models in neuroscience and ecology, and some large-scale dynamical models in engineering and geophysical flows can be found in (Chen and Majda, 2018a).

Despite being highly nonlinear and non-Gaussian, one of the important features of the CGNS (3.2) is that the conditional distribution of $\mathbf{Y}$ given one realization of the time series $\mathbf{X}$ can be solved via closed analytic formulae. Such a unique analytic property significantly facilitates the analysis and calculations of state estimation, data assimilation and forecast. This feature also makes the CGNS to be quite different from general nonlinear or non-Gaussian systems. For the latter, particle methods have to be applied for state estimation (Carpenter et al., 1999; Hol et al., 2006; Nummiaro et al., 2003), in which many empirical tunings have to be involved to mitigate the numerical sampling errors.

3.2.2 Closed analytic formulae for computing the conditional statistics and data assimilation

3.2.2.1 Nonlinear filter

For the CGNS (3.2), given one realization of the time series $\mathbf{X}(s)$ for $s \in [0, t]$, the conditional distribution

$$p(\mathbf{Y}(t)|\mathbf{X}(s), s \leqslant t) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{f}}(t), \mathbf{R}_{\mathbf{f}}(t)) \tag{3.3}$$

becomes Gaussian, where the conditional mean $\boldsymbol{\mu}_{\mathbf{f}}$ and the conditional covariance $\mathbf{R}_{\mathbf{f}}$ are given by the following explicit formulae (Liptser and Shiryaev, 2013)

$$\frac{d\boldsymbol{\mu}_{\mathbf{f}}}{dt} = (\mathbf{a_0} + \mathbf{a_1}\boldsymbol{\mu}_{\mathbf{f}}) + (\mathbf{R}_{\mathbf{f}}\mathbf{A}_{\mathbf{1}}^{*})(\mathbf{B}_1\mathbf{B}_1^{*})^{-1}\left(\frac{d\mathbf{X}}{dt} - (\mathbf{A_0} + \mathbf{A_1}\boldsymbol{\mu}_{\mathbf{f}})\right), \tag{3.4a}$$

$$\frac{d\mathbf{R}_{\mathbf{f}}}{dt} = \mathbf{a_1}\mathbf{R}_{\mathbf{f}} + \mathbf{R}_{\mathbf{f}}\mathbf{a}_{\mathbf{1}}^{*} + \mathbf{b}_2\mathbf{b}_2^{*} - (\mathbf{R}_{\mathbf{f}}\mathbf{A}_{\mathbf{1}}^{*})(\mathbf{B}_1\mathbf{B}_1^{*})^{-1}(\mathbf{A_1}\mathbf{R}_{\mathbf{f}}), \tag{3.4b}$$

with $\cdot^*$ being the complex conjugate transpose. The subscript '$\mathbf{f}$' in the conditional mean $\boldsymbol{\mu}_{\mathbf{f}}$ and conditional covariance $\mathbf{R}_{\mathbf{f}}$ is an abbreviation for 'filter'. The explicit formulae in (3.3)-(3.4) correspond to the optimal nonlinear filter solution of the state variable $\mathbf{Y}(t)$ given a realization of the observed time series $\mathbf{X}(s)$ for $s \in [0, t]$. Thus, $\boldsymbol{\mu}_{\mathbf{f}}$ and $\mathbf{R}_{\mathbf{f}}$ in (3.4) are also known as the filter posterior mean and the filter posterior covariance. The classical Kalman-Bucy filter (Kalman and Bucy, 1961) is the simplest special example of (3.4).

The closed analytic formula (3.4) provides an efficient algorithm for the nonlinear data assimilation of the CGNS, which avoids using the ensemble or particle

methods that may suffer from sampling errors. In Section 3.4, the closed analytic data assimilation formula (3.4) will be used for an accurate state estimation of the initial value that facilitates effective ensemble forecast. It also allows the development of an efficient Gaussian mixture algorithm for calculating the non-Gaussian PDFs of the CGNS (see Section 3.2.3), which overcomes the curse of dimensionality. Such non-Gaussian PDFs are crucial in analyzing the model sensitivity and predicting the system response, the details of which will be discussed in Section 3.6.

3.2.2.2 Nonlinear smoother

Filtering exploits the observational information up to the current time instant for an online state estimation. On the other hand, given the observational time series within an entire time interval, the state estimation can become more accurate. This is known as the smoother (Särkkä, 2013).

Given one realization of the observed variable $\mathbf{X}(t)$ for $t \in [0, T]$, the optimal smoother estimate $p(\mathbf{Y}(t)|\mathbf{X}(s), s \in [0, T])$ of the CGNS (3.2) is also Gaussian (Chen, 2020b),

$$p(\mathbf{Y}(t)|\mathbf{X}(s), s \in [0, T]) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{s}}(t), \mathbf{R}_{\mathbf{s}}(t)), \tag{3.5}$$

where the conditional mean $\boldsymbol{\mu}_{\mathbf{s}}(t)$ and conditional covariance $\mathbf{R}_{\mathbf{s}}(t)$ of the smoother at time t satisfy the following backward equations

$$\frac{\overleftarrow{d\boldsymbol{\mu}_{\mathbf{s}}}}{dt} = -\mathbf{a_0} - \mathbf{a_1}\boldsymbol{\mu}_{\mathbf{s}} + (\mathbf{b}_2\mathbf{b}_2^*)\mathbf{R}_{\mathbf{f}}^{-1}(\boldsymbol{\mu}_{\mathbf{f}} - \boldsymbol{\mu}_{\mathbf{s}}), \tag{3.6a}$$

$$\frac{\overleftarrow{d\mathbf{R}_{\mathbf{s}}}}{dt} = -(\mathbf{a_1} + (\mathbf{b}_2\mathbf{b}_2^*)\mathbf{R}_{\mathbf{f}}^{-1})\mathbf{R}_{\mathbf{s}} - \mathbf{R}_{\mathbf{s}}(\mathbf{a}_1^* + (\mathbf{b}_2\mathbf{b}_2^*)\mathbf{R}_{\mathbf{f}}) + \mathbf{b}_2\mathbf{b}_2^*, \tag{3.6b}$$

with $\boldsymbol{\mu}_{\mathbf{f}}$ and $\mathbf{R}_{\mathbf{f}}$ being given by (3.4). Here, the subscript '$\mathbf{s}$' in the conditional mean $\boldsymbol{\mu}_{\mathbf{s}}$ and conditional covariance $\mathbf{R}_{\mathbf{s}}$ is an abbreviation for 'smoother', which should not be confused by the time variable s in $\mathbf{X}(s)$. In (3.6), the terms on the left hand side are understood as

$$\overleftarrow{d\boldsymbol{\mu}_{\mathbf{s}}} = \lim_{\Delta t \to 0} \boldsymbol{\mu}_{\mathbf{s}}(t) - \boldsymbol{\mu}_{\mathbf{s}}(t + \Delta t), \qquad \overleftarrow{d\mathbf{R}_{\mathbf{s}}} = \lim_{\Delta t \to 0} \mathbf{R}_{\mathbf{s}}(t) - \mathbf{R}_{\mathbf{s}}(t + \Delta t),$$

while the terms on the right hand side of (3.6) takes values at t. The starting value of the nonlinear smoother $(\boldsymbol{\mu}_{\mathbf{s}}(T), \mathbf{R}_{\mathbf{s}}(T))$ is the same as the filter estimate at the endpoint $(\boldsymbol{\mu}_{\mathbf{f}}(T), \mathbf{R}_{\mathbf{f}}(T))$.

The nonlinear smoother plays an important role for an unbiased state estimation and postprocessing of the data. It is also able to quantify the uncertainty in the unobserved variables in the parameter estimation given only partial observations, which will be a topic to be studied in Section 3.5. In addition, the nonlinear smoother is also the basis to the development of a nonlinear sampling formula, which will be shown below and is a necessary step in analyzing the model sensitivity and predicting the system response in Section 3.6.

3.2.2.3 Nonlinear sampling formula

Associated with the nonlinear smoother, a nonlinear conditional sampling formula can be derived. In addition to satisfying the point-wise optimal estimate (3.6), i.e., a distribution formed by conditional mean and conditional distribution at each time stamp, the conditional sampled trajectories further take into account the temporal correlation. These sampled trajectories in the CGNS framework can be regarded as

the analogs of the ensemble members in the ensemble Kalman smoother (Evensen, 2009), but the former can be obtained via a closed analytic formula.

Conditioned on one realization of the observed variable $\mathbf{X}(s)$ for $s \in [0, T]$, the optimal strategy of sampling the trajectories associated with the unobserved variable $\mathbf{Y}$ satisfies the following explicit formula (Chen and Majda, 2020),

$$\frac{\overleftarrow{d\mathbf{Y}}}{dt} = \frac{\overleftarrow{d\boldsymbol{\mu}_s}}{dt} - \left(\mathbf{a_1} + (\mathbf{b}_2\mathbf{b}_2^*)\mathbf{R}_f^{-1}\right)(\mathbf{Y} - \boldsymbol{\mu}_s) + \mathbf{b}_2\dot{\mathbf{W}}_{\mathbf{Y}}(t), \tag{3.7}$$

where $\dot{\mathbf{W}}_{\mathbf{Y}}(t)$ is a random noise that is independent from $\dot{\mathbf{W}}_2(t)$ in (3.2). The conditional sampling formula is another necessary component in analyzing the model sensitivity and predicting the system response in Section 3.6.

3.2.3 Semi-analytic and statistically accurate formulae for solving the non-Gaussian PDFs via mixtures

The closed analytic formula (3.4) in calculating the conditional distribution $p(\mathbf{Y}(t)|\mathbf{X}(s), s \leqslant t)$ in (3.3) also provides an extremely useful way to compute the marginal distribution $p(\mathbf{Y}(t))$. In fact, assuming there are L trajectories of $\mathbf{X}(s \leqslant t)$, denoted by $\mathbf{X}_i^{obs}(s \leqslant t)$ for $i = 1, \ldots, L$, then in the limit with $L \to \infty$, the marginal distribution $p(\mathbf{Y}(t))$ is given by

$$p(\mathbf{Y}(t)) = \lim_{L\to\infty} \frac{1}{L} \sum_{i=1}^{L} p\Big(\mathbf{Y}(t)|\mathbf{X}_i^{obs}(s \leqslant t)\Big). \tag{3.8}$$

While the above identity is in the asymptotic form with $L \to \infty$, it has been shown that the error bound in approximating $p(\mathbf{Y}(t))$ with a finite L does not depend on the dimension of $\mathbf{Y}$ (Chen et al., 2018b). In other words, fundamentally different from the traditional Monte Carlo simulations, the method in (3.8) avoids the curse of dimensionality. If it is further assumed that the dimension of $\mathbf{X}$ is low, then the following efficient and statistically accurate approach can be utilized to compute the joint PDF at any transient phase $p(\mathbf{X}(t), \mathbf{Y}(t))$ (Chen and Majda, 2017),

$$p(\mathbf{X}(t), \mathbf{Y}(t)) = \lim_{L\to\infty} \frac{1}{L} \sum_{i=1}^{L} \Big(K_{\mathbf{H}}(\mathbf{X}(t) - \mathbf{X}_i^{obs}(t))\, p(\mathbf{Y}(t)|\mathbf{X}_i^{obs}(s \leqslant t)) \Big). \tag{3.9}$$

Here the distribution of $\mathbf{X}$ is solved via a kernel density estimation, where each kernel component is a Gaussian function centered at $\mathbf{X}_i^{obs}(t)$ with a bandwidth $\mathbf{H}$ (Botev et al., 2010). In addition to the advantage of applying (3.9) in solving high-dimensional PDF (especially when the dimension of Y is large), the semi-analytic formula in (3.9) also allows a smoothed PDF, which reduces the sampling error compared with other approaches even for systems with moderate or low dimensions.

It is important to note that if the underlying system contains model error, then the PDF provided by (3.9) is very different from the one created by simply running the imperfect model. This is because the model error in the marginal distribution $p(\mathbf{Y}(t))$ is mitigated with the help of the observations $\mathbf{X}^i(s \leqslant t)$. Such a unique feature plays a crucial role in improving the results for computing the model response and sensitivity analysis, where the perfect model is seldom known in

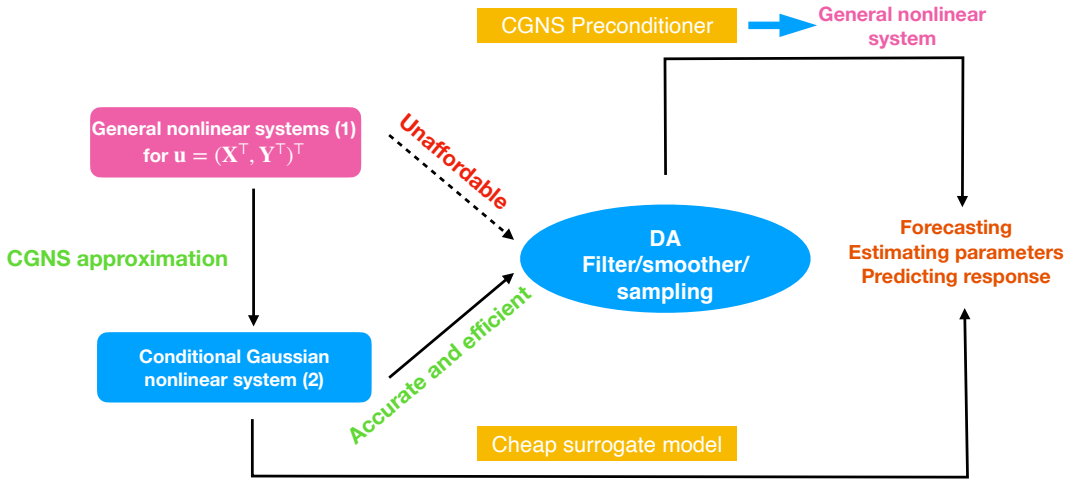

Figure 3.1: A schematic illustration of utilizing the CGNS as a fast preconditioner and a cheap surrogate model for general nonlinear system.

practice. The details will be illustrated in Section 3.6. As a final remark, only the equilibrium PDF is required in many applications, including the study of the model response. Therefore, if the system is ergodic, then only a single (sufficiently long) trajectory of $\mathbf{X}(0 \leqslant t \leqslant T)$, denoted by $\mathbf{X}^{\text{obs}}(0 \leqslant t \leqslant T)$, is needed in computing the equilibrium distribution $p_{\text{eq}}(\mathbf{X}, \mathbf{Y})$

$$p_{\text{eq}}(\mathbf{X}, \mathbf{Y}) = \lim_{J\to\infty} \frac{1}{J} \sum_{j=1}^{J} \Big(K_{\mathbf{H}}(\mathbf{X} - \mathbf{X}^{\text{obs}}(t_j))\, p(\mathbf{Y}|\mathbf{X}^{\text{obs}}(s \leqslant t_j)) \Big), \tag{3.10}$$

where $[t_1, \ldots, t_J]$ is a partition of the time interval $[T_0, T]$ with some burn-in time T_0.

Finally, Figure 3.1 shows a schematic illustration of utilizing the CGNS as a fast preconditioner and a cheap surrogate model for general nonlinear system.

3.3 Strategies of Developing CGNS

The goals of developing approximate models for solving different problems in practice are often distinct to each other. For example, some applications require a skillful forecast model while others seek for a suitable model to reproduce certain statistics. While there is not a universal criterion to build the "optimal" CGNS as an approximate model for all applications, a few potentially useful strategies are provided below for constructing CGNS, which will be applied in the following sections.

3.3.1 Fast wave averaging

Recall the general form of the complex systems with quadratic nonlinearity (3.1). Writing it into the form of state variables $(\mathbf{X}, \mathbf{Y})$ yields

$$\begin{aligned}\frac{d\mathbf{X}}{dt} &= L_{11}\mathbf{X} + L_{12}\mathbf{Y} + B_{11}^{1}(\mathbf{X},\mathbf{X}) + B_{12}^{1}(\mathbf{X},\mathbf{Y}) + B_{22}^{1}(\mathbf{Y},\mathbf{Y}) + \mathbf{F}_1 + \boldsymbol{\sigma}_1(\mathbf{X},\mathbf{Y})\dot{\mathbf{W}}_1,\\ \frac{d\mathbf{Y}}{dt} &= L_{21}\mathbf{X} + L_{22}\mathbf{Y} + B_{11}^{2}(\mathbf{X},\mathbf{X}) + B_{12}^{2}(\mathbf{X},\mathbf{Y}) + B_{22}^{2}(\mathbf{Y},\mathbf{Y}) + \mathbf{F}_2 + \boldsymbol{\sigma}_2(\mathbf{X},\mathbf{Y})\dot{\mathbf{W}}_2,\end{aligned} \tag{3.11}$$

where for the notation simplicity the explicit time dependence has been omitted. In (3.11), L_{ij} are constant matrices while B_{ij}^{k} are vector functions with the entries being quadratic functions of the state variables. In some applications, there exists a scale separation of the state variables, where $\mathbf{X}$ and $\mathbf{Y}$ represent slow and fast variables, respectively. In such a case, it is natural to apply a fast wave average such that the terms representing self interaction between the fast variables, i.e.,

$B_{22}^1(\mathbf{Y},\mathbf{Y})$ and $B_{22}^2(\mathbf{Y},\mathbf{Y})$, are approximated by stochastic damping and noise (Majda et al., 2001, 1999). By further approximating the diffusion coefficients $\boldsymbol{\sigma}_1(\mathbf{X},\mathbf{Y})$ and $\boldsymbol{\sigma}_2(\mathbf{X},\mathbf{Y})$ by functions of only $\mathbf{X}$, the resulting system becomes

$$\begin{aligned}\frac{d\mathbf{X}}{dt} &= \widetilde{L}_{11}\mathbf{X} + \widetilde{L}_{12}\mathbf{Y} + B_{11}^1(\mathbf{X},\mathbf{X}) + B_{12}^1(\mathbf{X},\mathbf{Y}) + \mathbf{F}_1 + \widetilde{\boldsymbol{\sigma}}_1(\mathbf{X})\dot{\mathbf{W}}_1,\\ \frac{d\mathbf{Y}}{dt} &= \widetilde{L}_{21}\mathbf{X} + \widetilde{L}_{22}\mathbf{Y} + B_{11}^2(\mathbf{X},\mathbf{X}) + B_{12}^2(\mathbf{X},\mathbf{Y}) + \mathbf{F}_2 + \widetilde{\boldsymbol{\sigma}}_2(\mathbf{X})\dot{\mathbf{W}}_2,\end{aligned} \tag{3.12}$$

which belongs to the CGNS (3.2). Note that, if the scale separation is not strong enough to apply the fast wave averaging, then $B_{22}^1(\mathbf{Y},\mathbf{Y})$ and $B_{22}^2(\mathbf{Y},\mathbf{Y})$ can be approximated by additional closure terms (San and Maulik, 2018; Mou et al., 2021b) that include nonlinear functions of $\mathbf{X}$ and bilinear functions of $\mathbf{X}$ and $\mathbf{Y}$ to fit the CGNS framework.

3.3.2 Stochastic parameterizations

The fast wave averaging or closure approximations are suitable approaches to build CGNS if the starting complex nonlinear system is completely known. However, in many practical applications, the information of the perfect model is not entirely available. Specifically, while the large-scale dynamics of nature is often accessible, the details of the small- or unresolved-scale features are not fully understood. In such a situation, suitable stochastic parameterizations can be adopted to approximate the processes of the unobserved variables $\mathbf{Y}$ such that the feedback from small/unresolved to large/resolved scales are well characterized and the parameterized system follows the CGNS structure.

One of the simplest strategies is to apply a linear stochastic model with Gaussian white noise to describe each component of the hidden processes of $\mathbf{Y}$ while the processes of $\mathbf{X}$ remain highly nonlinear. This parameterization strategy has been utilized in data assimilation and short-term statistical prediction (Gershgorin et al., 2010b,a; Branicki et al., 2013). The model with such a simple stochastic parameterization automatically fits the CGNS as there is no quadratic function of $\mathbf{Y}$ involved. A more sophisticated stochastic parameterization is to incorporate the physics-constrained nonlinear regression model (Majda and Harlim, 2012b; Harlim et al., 2014) into the CGNS. This allows a more accurate way in characterizing the nonlinear dynamics of the small-scale features in $\mathbf{Y}$, influenced by the large-scale variables $\mathbf{X}$. In addition, the coupled system with physics constraints also prevents finite-time blow up of the solutions and facilitates a skillful medium- to long-range forecast.

3.3.3 System augmentation

As has been discussed in Section 3.3.1 that the most significant difference between the general nonlinear system (3.1) and the CGNS (3.2) is the quadratic nonlinear self interactions of $\mathbf{Y}$, namely the $\mathbf{Q}(\mathbf{Y}(t),\mathbf{Y}(t))$ and $\mathbf{q}(\mathbf{Y}(t),\mathbf{Y}(t))$ terms in the following system, which has been written in a form that is adaptive to the CGNS (3.2),

$$\frac{d\mathbf{X}(t)}{dt} = \Big[\mathbf{A_0}(\mathbf{X},t) + \mathbf{A_1}(\mathbf{X},t)\mathbf{Y}(t) + \mathbf{Q}(\mathbf{Y}(t),\mathbf{Y}(t))\Big] + \mathbf{B_1}(\mathbf{X},t)\dot{\mathbf{W}}_1(t), \tag{3.13a}$$

$$\frac{d\mathbf{Y}(t)}{dt} = \Big[\mathbf{a_0}(\mathbf{X},t) + \mathbf{a_1}(\mathbf{X},t)\mathbf{Y}(t) + \mathbf{q}(\mathbf{Y}(t),\mathbf{Y}(t))\Big] + \mathbf{b_2}(\mathbf{X},t)\dot{\mathbf{W}}_2(t), \tag{3.13b}$$

where the dependence of $\mathbf{Y}$ in the diffusion coefficients have been eliminated for simplicity. Different from approximating the two quadratic terms $\mathbf{Q}(\mathbf{Y}(t), \mathbf{Y}(t))$ and $\mathbf{q}(\mathbf{Y}(t), \mathbf{Y}(t))$ directly, the coupled system (3.13) can be augmented to a new system, where an additional variable $\mathbf{Z} = \mathbf{Y}\mathbf{Y}^*$ is introduced. Applying Ito's formula then yields (Gardiner, 2004)

$$\begin{aligned}\frac{d\mathbf{Z}(t)}{dt} = \Big[\mathbf{Y}(t)\mathbf{a}_0^*(\mathbf{X}, t) + \mathbf{Z}(t)\mathbf{a}_1^*(\mathbf{X}, t) + \underline{\mathbf{Y}(t)\mathbf{q}(\mathbf{Y}(t), \mathbf{Y}(t))} + \mathbf{b}_2(\mathbf{X}, t)\mathbf{b}_2^*(\mathbf{X}, t)\Big] \\ + \underline{\mathbf{Y}(t)}\mathbf{b}_2^*(\mathbf{X}, t)\dot{\mathbf{W}}_2^*(t).\end{aligned} \tag{3.14}$$

The conditional distribution $p(\mathbf{Y}(t)|\mathbf{X}(s \leqslant t))$ can be solved by first computing the conditional distribution $p(\mathbf{Y}(t), \mathbf{Z}(t)|\mathbf{X}(s \leqslant t))$ and then marginalizing it over $\mathbf{Z}$. One of the advantages of such a strategy is that both $\mathbf{Q}$ and $\mathbf{q}$ can be written as a function of $\mathbf{Z}$. Therefore, the only term in the drift part that breaks the structure of the CGNS is the cubic term $\mathbf{Y}(t)\mathbf{q}(\mathbf{Y}(t), \mathbf{Y}(t))$, which together with the diffusion term in the $\mathbf{Z}$ equation, needs to be approximated. Nevertheless, since the right hand size of $\mathbf{X}$ and $\mathbf{Y}$ are unchanged, it is expected that the coupled system (3.13)–(3.14) with a suitable approximation has reasonable data assimilation skill. Note that if $\mathbf{q} \equiv 0$, then there is no approximation involved in the drift part of (3.13)–(3.14). Although the dimension of the augmented system may increase dramatically for high dimensional systems, the strategy is effective for models with low to moderate dimensions.

3.4 Data Assimilation and Ensemble Forecast

Data assimilation concerns the problem of estimating the state variables of a given, usually nonlinear and possibly stochastic, dynamical system when observations of certain related output variables are available (Evensen, 2009; Kalnay, 2003; Law et al., 2015; Majda and Harlim, 2012a). One major challenge in data assimilation is the strong nonlinearity and the associated non-Gaussian statistics in the underlying dynamics, in which a direct application of the particle methods may be inaccurate especially in the high dimensional situations. The development of cheap and effective approximate models that capture the main characteristics of the underlying dynamics is thus an important topic in state estimation and data assimilation. Since the data assimilation solution corresponds to the initialization of the subsequent forecast, an efficient and accurate data assimilation scheme is also essential to advancing the forecast skill. Note that there is usually a stronger demand in developing suitable approximate models for data assimilation than the subsequent short- or medium-range forecast since the former often involves many numerical or sampling issues in the presence of strong nonlinearity and non-Gaussianity.

In this section, a particular type of approximate models for this purpose obtained by the method of system augmentations presented in Section 3.3.3 is studied. The resulting approximate model has the form of a CGNS. Thus, the associated data assimilation solutions can be calculated using the closed analytic formula (3.4) as was discussed in Section 3.2.2. To simplify the presentation, the idea is illustrated using a low-dimensional SDE system with energy-conserving quadratic terms. The data assimilation results from the CGNS, which is an approximate model, will be

compared with that by applying a classical ensemble data assimilation method directly to the perfect model. The goal is to illustrate the efficiency and accuracy of the data assimilation scheme using the CGNS, especially in avoiding the potential sampling and other numerical issues that appear in the ensemble-based approaches.

3.4.1 A truncated stochastic quadratic system and its CGNS approximation through system augmentation

The model considered here is the following three-dimensional SDEs with energy-conserving quadratic nonlinear terms and subject to additive white noise forcing

$$\frac{dx}{dt} = \beta_x x + \alpha xy + \alpha yz + \sigma_x \dot{W}_x, \tag{3.15a}$$

$$\frac{dy}{dt} = \beta_y y - \alpha x^2 + 2\alpha xz + \sigma_y \dot{W}_y, \tag{3.15b}$$

$$\frac{dz}{dt} = \beta_z z - 3\alpha xy + \sigma_z \dot{W}_z. \tag{3.15c}$$

Here, the coefficients for the linear terms are chosen such that β_x is positive to introduce linear instability into the system, while β_y and β_z are negative, representing linear damping effects. The coefficient $\alpha > 0$ controls the strength of the nonlinearity; and the noise strength coefficients σ_x, σ_y, and σ_z are positive constants. This system can for instance be obtained as a Fourier-Galerkin projection of the stochastic Burgers-Sivashinsky equation

$$\frac{\partial u}{\partial t} = \left(\nu \partial_{xx} u + \lambda u - u \partial_x u\right) + \dot{W}(t, x)$$

posed on a bounded interval $x \in (0, L)$ subject to homogeneous Dirichlet boundary conditions. In this context, β_x, β_y, and β_z are simply the three largest eigenvalues of the linear operator and α is linked to the domain size L via $\alpha = \pi/(\sqrt{2}L^{3/2})$; see e.g. (Chekroun et al., 2015, Chapter 6).

In the following, the largest scale variable x is treated as the observed variable while there is no direct observations for the state variables (y, z). Under this splitting of the state variables, system (3.15) does not have the conditional Gaussian structure due to the quadratic nonlinear term αyz between the unobserved variables that appears in (3.15a). Following the idea presented in Section 3.3.3, in order to obtain a CGNS to approximate the system (3.15), three auxiliary variables are introduced for the possible quadratic interactions between the two unobserved variables:

$$p = y^2, \quad q = yz, \quad r = z^2. \tag{3.16}$$

Using (3.15) and apply Itô's formula yields

$$\begin{aligned}
\frac{dp}{dt} &= (\sigma_y)^2 + 2\big(\beta_y p - \alpha x^2 y + 2\alpha x q\big) + 2\sigma_y y \dot{W}_y, \\
\frac{dq}{dt} &= (\beta_y + \beta_z) q - \alpha x^2 z - 3\alpha x p + 2\alpha x r + \sigma_y z \dot{W}_y + \sigma_y y \dot{W}_z, \\
\frac{dr}{dt} &= (\sigma_z)^2 + 2\big(\beta_z r - 3\alpha x q\big) + 2\sigma_z z \dot{W}_z.
\end{aligned} \tag{3.17}$$

Assume that the global mean values of the unobserved variables y and z are accessible (from a period of training data). Then, in combination with (3.15), and after replacing y and z in the state-dependent noise terms of (3.17) by their

respective global mean, the following augmented system is arrived at:

$$
\begin{aligned}
\frac{dx}{dt} &= \beta_x x + \alpha x y + \alpha q + \sigma_x \dot{W}_x, \\
\frac{dy}{dt} &= \beta_y y - \alpha x^2 + 2\alpha x z + \sigma_y \dot{W}_y, \\
\frac{dz}{dt} &= \beta_z z - 3\alpha x y + \sigma_z \dot{W}_z, \\
\frac{dp}{dt} &= (\sigma_y)^2 + 2\big(\beta_y p - \alpha x^2 y + 2\alpha x q\big) + 2\sigma_y \overline{y} \dot{W}_y, \\
\frac{dq}{dt} &= (\beta_y + \beta_z) q - \alpha x^2 z - 3\alpha x p + 2\alpha x r + \sigma_y \overline{z} \dot{W}_y + \sigma_z \overline{y} \dot{W}_z, \\
\frac{dr}{dt} &= (\sigma_z)^2 + 2\big(\beta_z r - 3\alpha x q\big) + 2\sigma_z \overline{z} \dot{W}_z,
\end{aligned}
\tag{3.18}
$$

where yz in (3.15a) becomes the state variable q. This augmented system (3.18) fits into the CGNS form of (3.2) with now the unobserved variables taken to be $\mathbf{Y} = (y, z, p, q, r)^\top$.

Although the dimension of the approximate system is increased compared with the original system, closed analytic equations are now accessible for the evolution of the corresponding conditional statistics for the data assimilation solutions (see equation (3.4) in Section 3.2.2). As will be shown below, the approximate system (3.18) can provide a significantly more accurate estimation of (y, z) compared with another conditional Gaussian approximation obtained by simply removing the term αyz in (3.15a), called the bare truncation (BT) system below. The skill of the proposed method is comparable and sometimes even more accurate than the ensemble Kalman-Bucy filter (EnKBF) (Bergemann and Reich, 2012), while being more efficient thanks to the availability of analytic formulae.

3.4.2 Dynamical regimes and numerical setup

In the following, we consider two dynamical regimes:

$$\begin{aligned} &\text{Regime I:} \quad \sigma_x = 1, \sigma_y = 1, \sigma_z = 2, \beta_x = 0.1, \beta_y = -0.5, \beta_z = -1, \alpha = \pi/\sqrt{2}, \\ &\text{Regime II:} \quad \sigma_x = 0.1, \sigma_y = 1, \sigma_z = 2, \beta_x = 0.1, \beta_y = -0.5, \beta_z = -1, \alpha = \pi/\sqrt{2}. \end{aligned} \tag{3.19}$$

In particular, we have a relatively strong nonlinear effects with $\alpha \approx 2.2$, and a relatively small spectral gap between the observed and the hidden variables with $\beta_x - \beta_y = 0.6$. Both of the two hidden variables are subject to strong noise perturbations. The two regimes differ only in the value of the noise strength σ_x in the x-equation.

The same numerical setup is adopted for both parameter regimes. The true signal is obtained by integrating the original SDE system (3.15) for an arbitrarily fixed noise path using the Euler-Maruyama scheme with a uniform time step size $\delta t = 5 \times 10^{-4}$ and initialized at $(x, y, z) = (0, 0, 0)$.

For the state estimation of the unobserved variables (y, z), we compare three methods:

> **Method 1**: Apply the nonlinear filtering formulae (3.4) for the general CGNS (3.2) to the augmented system (3.18), with $\mathbf{X} = x$ and $\mathbf{Y} = (y, z, p, q, r)^\top$. This method will be referred as the CG method below.
>
> **Method 2**: Apply the nonlinear filtering formulae (3.4) to a bare truncation of (3.15) in which we simply remove the term αyz in (3.15a) to obtain a CGNS,

with $\mathbf{X} = x$ and $\mathbf{Y} = (y, z)^\top$. This method will be referred as the BT method below.

Method 3: Apply the ensemble Kalman-Bucy filtering (EnKBF) method to (3.15). See (3.23) below for its formulation.

The data assimilation for each of the above methods is performed over the time window $[0, 400]$ with the same time step size δt as the true signal. For the CG method, the global mean values $\overline{y}$ and $\overline{z}$ in (3.18) are taken to be the mean values of the corresponding true signal over the interval $[0, 200]$. For both CG and BT, the initial values of the conditional mean and conditional covariance are taken to be zero. For EnKBF, the size of ensemble is taken to be $N = 100$ and the unobserved variables are initialized at $(y, z) = (0, 0)$. For the sake of clarity, we provide below some details about the EnKBF applied to (3.15). We introduce the following notations for the drift part of the system (3.15):

$$\begin{aligned} g(x, y, z) &= \beta_x x + \alpha x y + \alpha y z, \\ f_1(x, y, z) &= \beta_y y - \alpha x^2 + 2\alpha x z, \\ f_2(x, y, z) &= \beta_z z - 3\alpha x y. \end{aligned} \tag{3.20}$$

Denote by $\mathbf{y} = (y_1, y_2, \ldots, y_N)^\top$ and $\mathbf{z} = (z_1, z_2, \ldots, z_N)^\top$ the collection of all the N

ensemble members. We define also

$$\begin{aligned}\mathcal{N}_1(x_{obs}(t),\mathbf{y},\mathbf{z}) &= \frac{1}{\sigma_x^2(N-1)}\sum_{j=1}^{N}(y_j-\overline{\mathbf{y}}(t))(g(x_{obs}(t),y_j,z_j)-\overline{g}(x_{obs}(t),\mathbf{y},\mathbf{z})),\\ \mathcal{N}_2(x_{obs}(t),\mathbf{y},\mathbf{z}) &= \frac{1}{\sigma_x^2(N-1)}\sum_{j=1}^{N}(z_j-\overline{\mathbf{z}}(t))(g(x_{obs}(t),y_j,z_j)-\overline{g}(x_{obs}(t),\mathbf{y},\mathbf{z})),\end{aligned} \tag{3.21}$$

where

$$\overline{\mathbf{y}}(t)=\frac{1}{N}\sum_{\ell=1}^{N}y_\ell(t),\quad \overline{\mathbf{z}}(t)=\frac{1}{N}\sum_{\ell=1}^{N}z_\ell(t),\quad \overline{g}(x_{obs}(t),\mathbf{y},\mathbf{z})=\frac{1}{N}\sum_{\ell=1}^{N}g(x_{obs}(t),y_\ell,z_\ell). \tag{3.22}$$

Then, each ensemble member (y_i,z_i), $i=1,2,\dots,N$, of the EnKBF is computed using

$$\begin{aligned}\frac{dy_i}{dt} &= f_1(x_{obs}(t),y_i,z_i)+\sigma_y\dot{W}_{y,i}\\ &\quad -\mathcal{N}_1(x_{obs}(t),\mathbf{y},\mathbf{z})\big[g(x_{obs}(t),y_i,z_i)-\dot{x}_{obs}(t)+\sigma_x\dot{W}_{x,i}\big],\\ \frac{dz_i}{dt} &= f_2(x_{obs}(t),y_i,z_i)+\sigma_z\dot{W}_{z,i}\\ &\quad -\mathcal{N}_2(x_{obs}(t),\mathbf{y},\mathbf{z})\big[g(x_{obs}(t),y_i,z_i)-\dot{x}_{obs}(t)+\sigma_x\dot{W}_{x,i}\big],\end{aligned} \tag{3.23}$$

where $W_{x,i}$, $W_{y,i}$, and $W_{z,i}$, $i=1,2,\dots,N$, are all mutually independent one-dimensional Brownian motions, and x_{obs} is the observed signal of x.

For Regime II, we will also compare the ensemble forecast skills. The forecast is performed over the time window $[200,400]$, which is chosen to avoid overlap with the training window $[0,200]$ from which the global mean values of y and z

appearing in (3.18) are computed. The forecast model is taken to be the true SDE system (3.15), and the initial conditions (IC) of (y, z) are drawn from multivariate Gaussian distributions with mean and covariance estimated respectively from BT, CG and EnKBF described above. For x, its initial value is taken to be that of the true signal at the corresponding time instant. We will also compute the results when the forecast is initialized with the true signal for all the three variables, which serves as the reference of the theoretic forecast/predictability limit and will be referred as the case with the perfect IC. The time locations at which to issue the forecasts are equally spaced over the chosen time interval, with a gap of 0.01 between two adjacent forecasts, leading thus to a total of 2×10^4 forecast locations. Each forecast is computed up to a lead time of 1 time unit, and a total of 40 ensemble members are generated at each forecast location. This procedure is repeated for each of the methods used to construct the IC.

3.4.3 Numerical results

We present now the results obtained based on the numerical procedure described above. For the two regimes given by (3.19), due to the larger noise strength parameter σ_x used in Regime I for the observed variable, the corresponding DA exercise is less challenging and will be presented first.

Results for Regime I. As is shown in Figures 3.2 for Regime I, the dynamics of x exhibits intermittent behavior with relatively quiescent episodes punctuated by large excursion events. Due to the relatively small spectral gap, the dynamics of y also exhibits highly nonlinear oscillations sustained by noise. In contrast, the

dynamics of z is mainly a damped oscillation sustained by noise due to the relatively strong linear stabilizing effects, and it is the variable that decays the fastest.

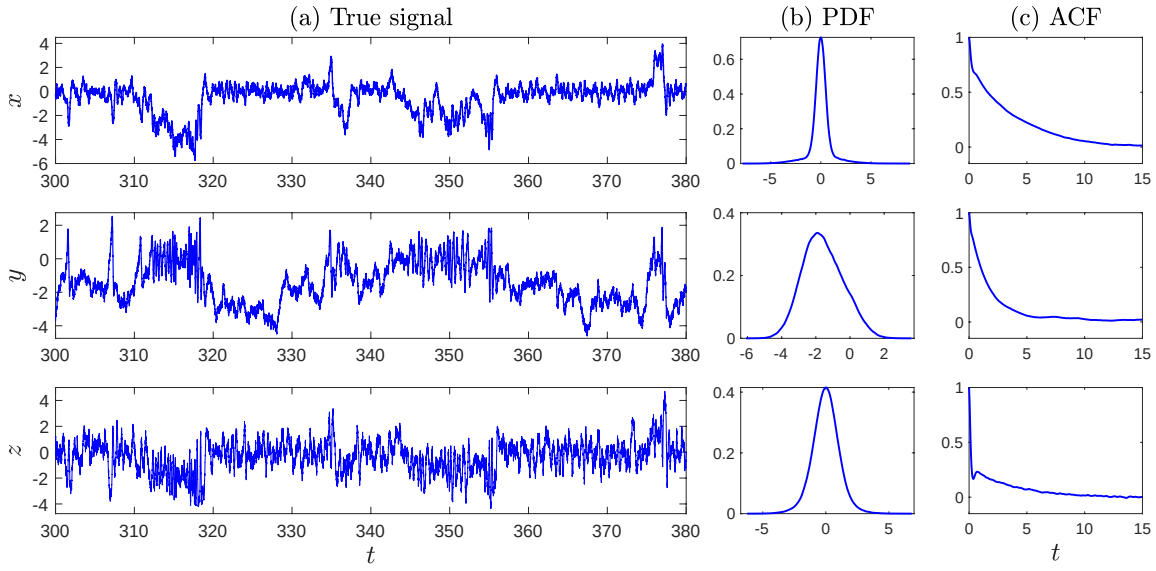

Figure 3.2: Solution of (3.15) in Regime I given by (3.19) for one arbitrarily fixed realization of the noise, and the corresponding probability density functions (PDFs) and autocorrelation functions (ACFs). This solution trajectory is taken to be the true signal. See Section 3.4.2 for details about the numerical setup. The ACFs and PDFs are estimated based on the solution trajectory over the time window $[0, 10^4]$, corresponding to 2×10^7 data points for the time step used.

The posterior mean states of (y, z) for Regime I obtained by CG and EnKBF are fairly close to each other as is shown in Figure 3.3. The conditional covariance matrices of (y, z) estimated by these two methods are also close to each other for this regime (not shown). In contrast, for BT, there are prolonged time windows over which the posterior mean of y deviates significantly from the true signal as is shown in Panel (a) of Figure 3.4. An inspection of the time series of yz shown in Panel (b) of Figure 3.4 reveals that such deviation typically occurs when the value

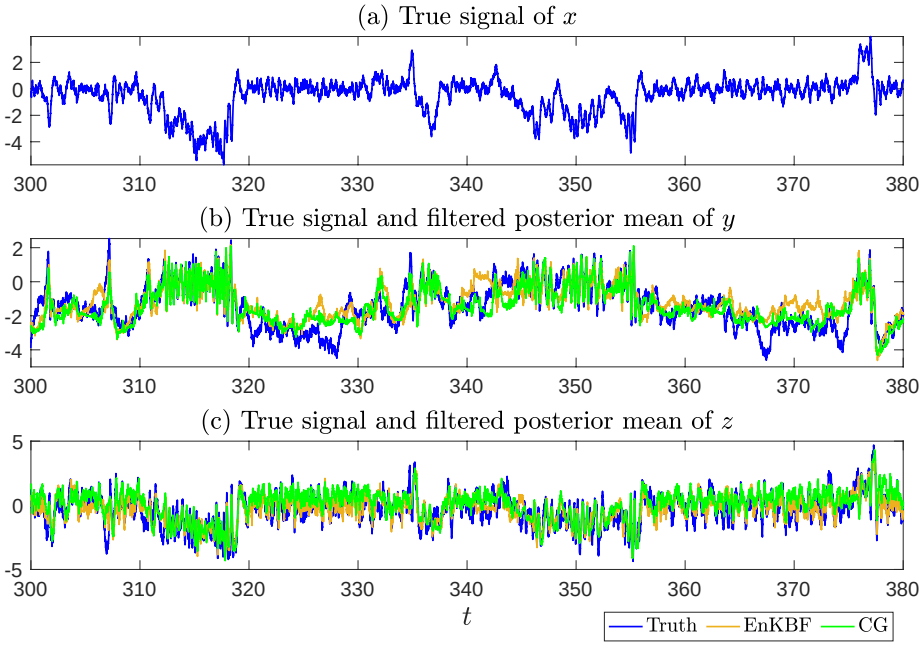

Figure 3.3: Panel (b): the filtered posterior mean of y for Regime I obtained from CG (green) and EnKBF (orange); the corresponding true signal previously shown in Figure 3.2 is plotted in blue. Panel (c): analogue of Panel (b) for z.

of yz is large, which is expected, since the omission of the term αyz in (3.15a) is the only difference between the BT system and the full system. The posterior mean of z obtained by BT is similar to those obtained by CG and EnKBF shown in Panel (c) of Figure 3.3, which is thus not presented.

Results for Regime II. The dynamics of the true system (3.15) in Regime II exhibits similar features as in Regime I shown in Figures 3.2, although the amplitude of each variable is slightly reduced due to the smaller noise intensity σ_x used for this regime. The shape of both the PDFs and ACFs of all the three variables are similar to those shown in Figures 3.2, with although the decorrelation time of x becoming comparable with that of y in this regime. The analogue of Figures 3.2 for Regime II

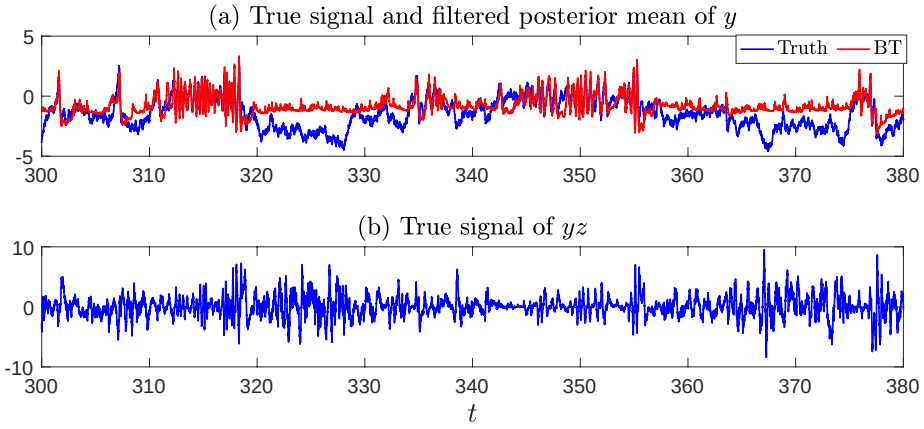

Figure 3.4: Panel (a): The filtered posterior mean of y obtained from BT (red) and the true signal of y (blue). Panel (b): The true signal of yz. The deterioration of the estimated y using BT occurs over time windows when the magnitude of yz gets large.

is thus omitted.

For this regime, CG and EnKBF still provide comparable posterior mean state of z as shown in Panel (c) of Figure 3.5, although the PDF of the posterior mean obtained by EnKBF approximates slightly better the PDF of the true signal of z. However, CG is significantly more skillful in estimating the conditional mean of y (Figure 3.5, Panels (b) and (d)).

The deterioration of the skill from EnKBF in this regime is associated with a false bimodal behavior appearing in the posterior mean of the y variable (see Panel (d) of Figure 3.5), whereas the true signal is unimodal, skewed towards negative values. It has also been checked that the bimodality is always there for EnKBF by further increasing the total number of ensemble members N or decreasing the numerical integration time step δt. Such a pathological behavior is associated with the filter divergence (Gottwald and Majda, 2013; Kelly et al., 2015), which often

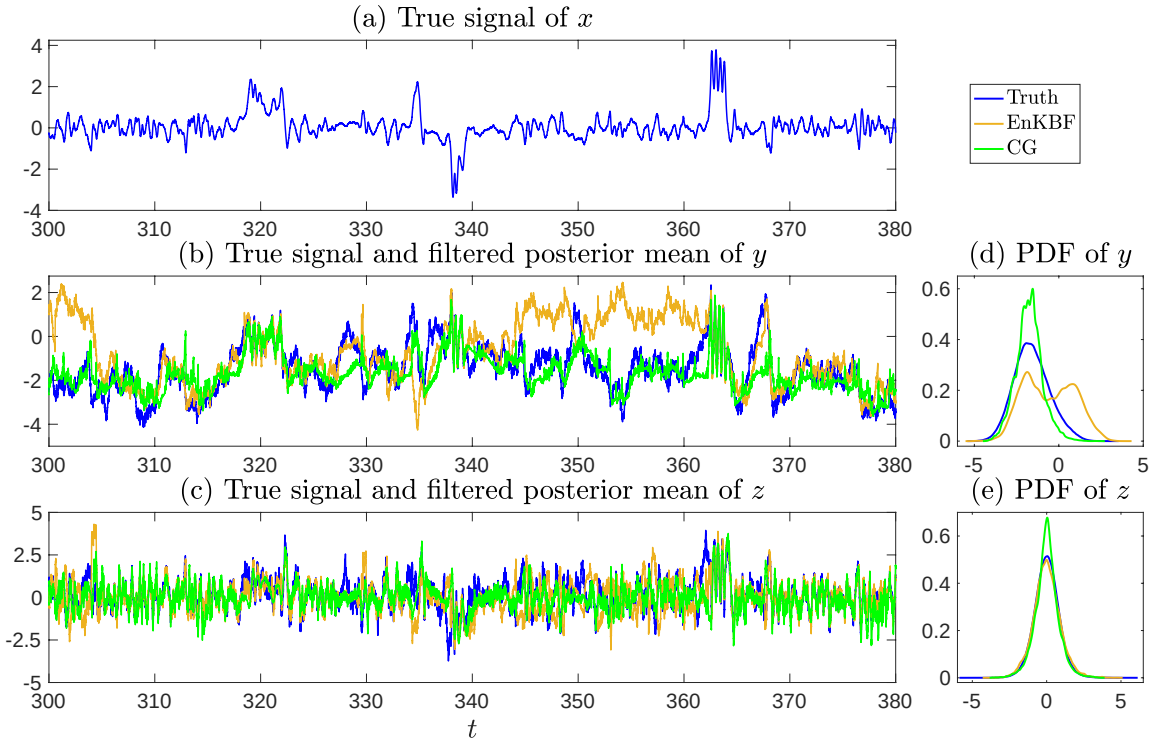

Figure 3.5: Panel (b): the filtered posterior mean of y for Regime II obtained from CG (green) and EnKBF (orange) with the corresponding true signal shown in blue. Panel (c): the analogue of Panel (b) for z. Panel (d): PDFs of the filtered posterior mean of y from CG (green) and EnKBF (orange) compared with that of the true signal. Panel (e): the analogue of Panel (d) for z.

occurs for the ensemble-based filters when the noise in the observational process is small and the observational process is highly nonlinear. In fact, the small noise in the observational process x makes the filter trusts more towards the information provided by the observations. However, the strong nonlinear and non-Gaussian features of x make it very difficult to accurately recover the states of y and z by inferring mainly from the x process.

In contrast, CG tracks well the modulations of the true signal, leading to a

much better reproduction of the PDF of the true signal of y; see again Panel (d) of Figure 3.5. It is worth pointing out that the original system (3.15) can also exhibit bimodal dynamics in a broad range of dynamical regimes, even though bimodality is not observed in the true signals of (x, y, z) for neither of the two regimes considered here. This bimodality that can occur in the dynamics of (3.15) is induced by the additive noise that drives the system to switch from the two locally stable steady states of the corresponding deterministic system produced from a supercritical pitchfork bifurcation, although when occurs, the bimodality is mainly visible in the x variable.

Regarding BT, compared with the corresponding result shown in Figure 3.4 for Regime I, its performance here is even worse and is thus not shown. In particular, the posterior mean state of y not only deviates significantly from the true signal but also has spurious fast oscillations presenting throughout the whole time window. Such fast oscillations also appear in the filtered posterior mean of z, although to a lesser extent.

For this regime, we also compared the skills of ensemble forecast with the forecast model simply taken to be the true SDE system (3.15), and the initial conditions (IC) of (y, z) drawn from multivariate Gaussian distributions in which the mean and covariance are estimated respectively from BT, CG and EnKBF described above; see Section 3.4.2 for further details. In addition, we compute the results when the forecast is initialized with the true signal for all the three variables, which serves as the reference of the theoretic forecast limit.

In Figure 3.6, we presented the normalized root-mean-square error $(RMSE)$

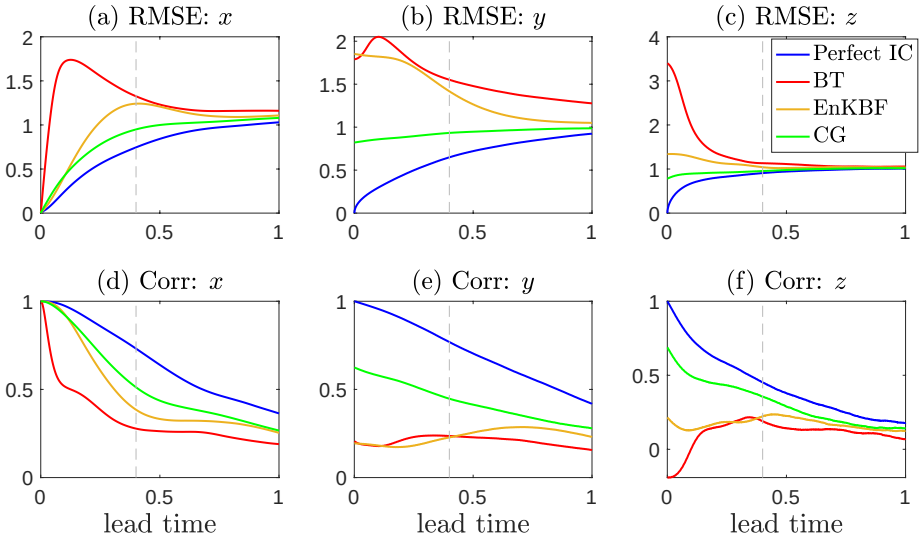

Figure 3.6: Panels (a)-(c): RMSE of the forecast skills for Regime II when the ICs are chosen either to be the perfect values from the true signals, or are drawn from multivariate Gaussian distributions in which the mean and covariance are estimated respectively from BT, CG and EnKBF; the RMSE are normalized by dividing by the standard deviation of the respective true signal. Panels (d)-(f): the correlation coefficients of the forecast skills. The vertical dashed line corresponds to lead time $\tau = 0.4$ for which the corresponding forecasted ensemble mean time series are shown in Figure 3.7.

and correlation coefficients of the forecasts for all the methods used. As expected, the better skills of CG at the DA stage carries over to the ensemble forecast as well. BT performs the worst due to large spurious oscillations appearing in its DA stage. While the RMSE is a convenient way of ranking the performance, to provide a better visualization of the skills, we also show in Figure 3.7 the forecasted ensemble mean trajectories at a given lead time, chosen here to be $\tau = 0.4$ time unit, which corresponds roughly to one half of the decorrelation time of the y-variable. The results in Figure 3.7 show that the forecast based on initial conditions

provided by CG (middle row) actually performs fairly well for all the three variables compared with those when perfect initial conditions are used (top row), although the uncertainty in the y variable is slightly higher for CG. The results for EnKBF are correlated with its performance at the DA stage, with significant error in the x and y variables over the time windows when the filtered posterior mean of y deviates from the true signal as was previously shown in the middle panel of Figure 3.5. For BT, large spurious oscillations appear in the forecasted ensemble mean time series for all the variables, especially for x and y. Such oscillations are inherited from those appearing in the assimilated y variables, which propagate to the other two variables due to nonlinear interactions.

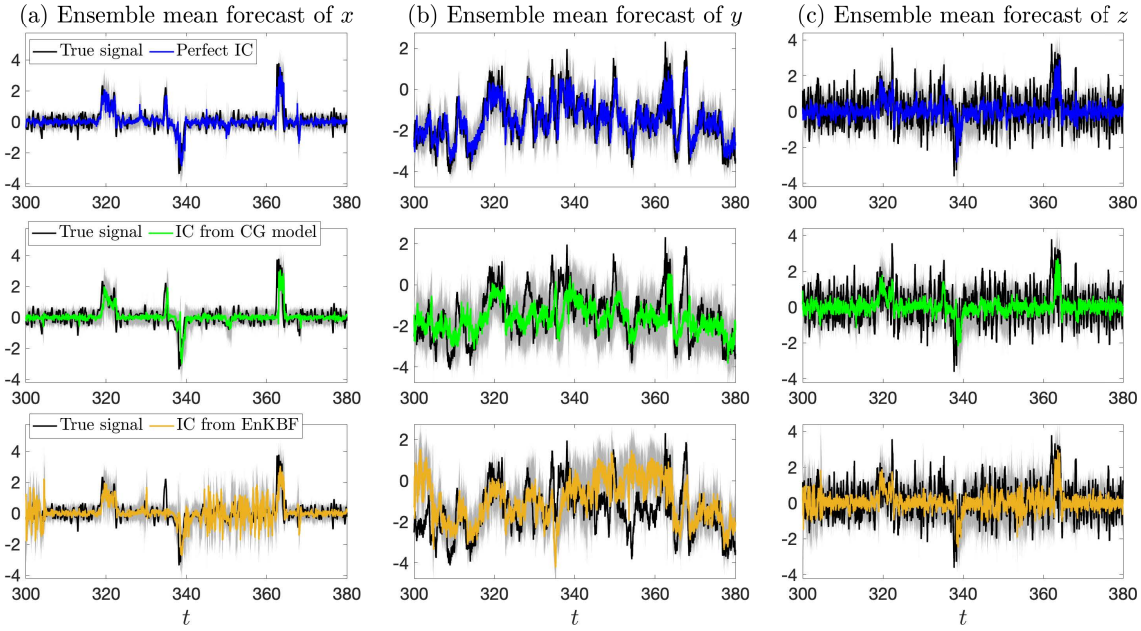

Figure 3.7: The forecasted ensemble mean time series for Regime II at lead time $\tau = 0.4$ time unit, when the ICs are from either the true signal (top row), or assimilated from the CG method (middle row), or the EnKBF (bottom row). The gray area on each panel marks the spread between 5 percentile and 95 percentile of the corresponding ensemble forecast. The true signals are plotted in black. For BT, large spurious oscillations appear in the forecasted ensemble mean time series for all the variables; and the results are not shown here.

3.5 Parameter Estimation

Parameter estimation is an important topic and a necessary precursor for effective state estimation, data assimilation, and prediction. Maximum likelihood estimation (MLE) and maximum a posterior (MAP) are often adopted to infer the parameter values if the observed time series for all the state variables are accessible (Myung, 2003; Kristensen et al., 2004; Coleman and Block, 2006). However, only partial observations are available in many complex nonlinear systems. In such a situation, data augmentation is widely used to simultaneously estimate the model parameter and recover the unobserved state variables (Tanner and Wong, 1987). In particular, data augmentation has been extensively incorporated into the Markov Chain Monte Carlo (MCMC) algorithms for improving the Bayesian inference (Wei and Tanner, 1990; Golightly and Wilkinson, 2008; Eraker, 2001; Papaspiliopoulos et al., 2013). In contrast to targeting the global optimal solution based on the MCMC, many other parameter estimation algorithms seek locally optimal solutions. One of the widely used local optimal parameter estimation approaches is the expectation-maximization (EM) algorithm (Ghahramani and Roweis, 1999; Ghahramani and Hinton, 1996; Dembo and Zeitouni, 1986). The EM algorithm is an iteration method that aims to find the parameter values that maximize the likelihood function and compute certain statistical expectations of the unobserved state variables in an alternating fashion. Note that, due to the local optimality property, the EM algorithm often requires fewer iterations than the MCMC algorithm. Unfortunately, both methods are computationally expensive for general complex nonlinear systems since neither the data augmentation in MCMC nor the computation of the statistical

expectation in the EM is easily obtained.

Unlike the general complex nonlinear systems, the closed analytic formulae of the CGNS facilitate the acceleration of the computational efficiency in parameter estimation. In particular, the conditional sampling formula (3.7) has the potential to allow a rapid data augmentation in the MCMC method while the analytic state estimation formula (3.6) offers an exact and accurate way to compute the statistical expectation, which is an essential component in the EM algorithm. Note that appropriately incorporating the CGNS into the MCMC may require several additional manipulations, which deserves a separate topic to study. Therefore, the focus below is on applying the CGNS to accelerate the parameter estimation utilizing the EM algorithm, where detailed mathematical justifications are more accessible. The CGNS in this entire procedure acts as a preconditioner to seek a suitable approximation of the statistical expectation of the unobserved state variables given the observational time series via the EM algorithm. With such a statistical expectation available from the CGNS, the original complex nonlinear system is only essential in computing the maximum likelihood solution at the last iteration step, leading thus to a significant speedup of the computational time required.

3.5.1 Accelerating the EM algorithm with a CGNS preconditioner

Denote by $(\mathbf{X}, \mathbf{Y})$ the state variables of a complex nonlinear system, where only a time series of $\mathbf{X}$ is observed while there is no direct observations for the state variable $\mathbf{Y}$. Let $\theta = (\theta^D, \theta^S)$ be the collections of the model parameters, where

θ^D and θ^S denote the model parameters in the drift part and the diffusion coefficients, respectively. Denote by $\widehat{\mathbf{X}} = \{\mathbf{X}^0, \ldots, \mathbf{X}^j, \ldots, \mathbf{X}^J\}$ and $\widehat{\mathbf{Y}} = \{\mathbf{Y}^0, \ldots, \mathbf{Y}^j, \ldots, \mathbf{Y}^J\}$ a discrete approximation of the continuous time series of $\mathbf{X}$ and $\mathbf{Y}$, respectively, within the time interval $t \in [0, T]$, where $T = J\Delta t$, $\mathbf{X}^j = \mathbf{X}(t_j)$ and $\mathbf{Y}^j = \mathbf{Y}(t_j)$, with $t_j = j\Delta t$.

The target is to seek an optimal estimation of the unknown parameters θ by maximizing the log-likelihood function. Since only the time series of $\mathbf{X}$ is observed, the maximum log-likelihood estimate is given by the solution that averages over the state variable $\mathbf{Y}$,

$$\mathcal{L}(\theta) = \log q(\mathbf{X}|\theta) = \log \int_{\mathbf{Y}} p(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{Y}. \tag{3.24}$$

Due to the unknown state variable $\mathbf{Y}$, there is no simple formula for a direct calculation of the log-likelihood. Nevertheless, the EM iteration algorithm can be applied to solve the optimization in (3.24). The EM algorithm alternates between performing an expectation (E) step, which estimates the unobserved state variable $\mathbf{Y}$ with an uncertainty quantification using the current estimate for the parameters, and a maximization (M) step, which updates the parameters by maximizing the expected log-likelihood found on the E step (Dembo and Zeitouni, 1986; Kokkala et al., 2014). Denote θ_k the updated parameters after the k-th iteration. The EM algorithm at step $k+1$ is the following:

E-Step. Computing the conditional distribution $p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \theta_k)$ using the previously estimated parameters θ_k and then plugging it into the cost function to compute

the function expectation

$$\mathcal{Q}(\theta;\theta_k)=\int_{\widehat{\mathbf{Y}}} p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}},\theta_k)\log p(\widehat{\mathbf{Y}},\widehat{\mathbf{X}}|\theta)d\widehat{\mathbf{Y}}. \tag{3.25}$$

M-Step. Updating the parameters θ_{k+1} utilizing the result from the E-Step,

$$\theta_{k+1}=\arg\max_{\theta}\mathcal{Q}(\theta;\theta_k). \tag{3.26}$$

In many situations, the M-Step usually involves solving a quadratic optimization problem, the analytic formula of which is available. However, for general nonlinear systems, the conditional distribution $p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}},\theta_k)$ in the E-Step is extremely difficult to solve. Note that such a conditional distribution is precisely the smoother estimate of the complex nonlinear system. Particle methods can be applied. Yet, repeatedly using these particle methods through the iteration procedure can be computationally expensive, and careful tuning is required, especially for systems with large dimensions.

To overcome the most significant computational barrier in the above EM algorithm, namely computing the conditional distribution $p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}},\theta_k)$, it is natural to exploit a suitable CGNS model as a preconditioner to accelerate the calculation of such a conditional distribution in the E-Step. We denote the distribution of the state variables of CGNS as $p^M(\widehat{\mathbf{Y}},\widehat{\mathbf{X}}|\theta_k^M)$. Here M indicates that the model is an imperfect model. Note that since the CGNS is only an approximate model, the collection of the parameters θ^M associated with the CGNS can be different from that θ associated with the perfect system. One major advantage of the CGNS is

that the closed analytic formula in (3.6) substantially advances the computational efficiency of the EM algorithm. After the conditional distribution $p^{M}(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \theta_k^M)$ being converged after a few EM iterations based on the CGNS, the original complex nonlinear system is then applied to compute the expectation and the maximum log-likelihood solution at the last iteration step. The technical details are included in the Appendix B.1.

Physics constraints or other constraints can be naturally incorporated into the CGNS for parameter estimation, which preserves the closed analytic formulae in the corresponding EM algorithm. In addition, a block decomposition of the conditional covariance can be further applied to deal with the complex nonlinear systems in high dimensions (Chen, 2020b). Both strategies are adopted in the following numerical tests, where the mathematical details are again included in

the Appendix B.1.1 and Appendix B.1.2.

Algorithm 1: EM with CGNS

1 Start with a given realization of the observations $\widehat{\mathbf{X}}$;

2 Propose an approximate model $p^{M}(\widehat{\mathbf{Y}}, \widehat{\mathbf{X}}|\theta^{M})$ that belongs to CGNS (3.2);

3 Assign an initial guess of the parameters θ_0^M;

4 **for** $k = 1 : K$ **do**

5 E-step: compute the conditional distribution $p^{M}(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \theta_{k-1}^{M})$ using the previously estimated parameters θ_{k-1}^{M} and then plugging it into the cost function to compute the function expectation $\widetilde{\mathcal{Q}}(\theta^{M}; \theta_{k-1}^{M}) = \int_{\mathbf{Y}} p^{M}(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \theta_{k-1}^{M}) \log p^{M}(\widehat{\mathbf{Y}}, \widehat{\mathbf{X}}|\theta^{M}) d\widehat{\mathbf{Y}}$;

6 M-step: update the parameters θ_k^M where $\theta_k^M = \arg\max_{\theta^M} \widetilde{\mathcal{Q}}(\theta^{M}; \theta_{k-1}^{M})$.

7 E-step: compute the conditional distribution $p^{M}(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \theta_K^M)$ and then compute the function expectation $\mathcal{Q}(\theta; \theta_K^M) = \int_{\mathbf{Y}} p^{M}(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \theta_K^M) \log p(\widehat{\mathbf{Y}}, \widehat{\mathbf{X}}|\theta) d\widehat{\mathbf{Y}}$;

8 M-step: update the parameters θ_{K+1} where $\theta_{K+1} = \arg\max_{\theta} \mathcal{Q}(\theta; \theta_K^M)$

3.5.2 A multiscale turbulent test model

In this subsection, the two-layer inhomogeneous Lorenz model is utilized to demonstrate that a suitable CGNS can be both a preconditioner and a surrogate model in parameter estimation. First, we show that a simple approximate model that belongs to CGNS can accelerate the EM algorithm as a preconditioner. Then, we show that this approximate model itself can be used as a surrogate model for prediction.

3.5.2.1 The perfect model

The two-layer Lorenz 96 (L96) model (Lorenz, 1996) is a conceptual representation of geophysical turbulence that is commonly used as a testbed for DA and parameterization in numerical weather forecasting (Majda and Grote, 2009; Grooms and Majda, 2014b,a; Majda and Grooms, 2014). The model mimics a coarse discretization of atmospheric flow on a latitude circle. It supports complex wave-like and chaotic behavior, and the two-layer structure schematically depicts the interactions between small-scale fluctuations and large-scale motions. The stochastic version of the model subject to additive noise forcing reads

$$\frac{du_i}{dt} = \left(-u_{i-1}\left(u_{i-2} - u_{i+1}\right) - u_i + f - \frac{hc_i}{J}\sum_{j=1}^{J} v_{i,j}\right) + \sigma_{u_i}\dot{W}_{u_i}, \quad i = 1, \ldots, I, \tag{3.27a}$$

$$\frac{dv_{i,j}}{dt} = \left(-bc_i v_{i,j+1}\left(v_{i,j+2} - v_{i,j-1}\right) - c_i v_{i,j} + \frac{hc_i}{J}u_i\right) + \sigma_{v_{i,j}}\dot{W}_{v_{i,j}}, \quad j = 1, \ldots, J, \tag{3.27b}$$

where I denotes the total number of large-scale variables, J the number of small-scale variables corresponding to each large-scale variable, f, h, c_i, b, σ_{u_i} and $\sigma_{v_{i,j}}$ are given scalar parameters while $\dot{W}_{u_i}$ and $\dot{W}_{v_{i,j}}$ are white noise. The large-scale variables u_i are periodic in i with $u_{i+I} = u_{i-I} = u_i$. The corresponding small-scale variables $v_{i,j}$ are periodic in i with $v_{i+I,j} = v_{i-I,j} = v_{i,j}$ and satisfy the following

conditions in j: $v_{i,j+J} = v_{i+1,j}$, and $v_{i,j-J} = v_{i-1,j}$.

The model discussed here uses variables u_i to describe large-scale or slow movements which are resolved; small scales or rapid fluctuations represented by $v_{i,j}$ are often unresolved ones. The coupling of fast and slow variables is regulated by the parameter h. The parameter f controls the magnitude of external large-scale forcing, while b determines the amplitude of nonlinear interactions between the fast variables. The parameter c_i specifies how quickly the fast variables are damped in comparison to the slow variables. As in the standard L96 model, we take $I = 40$, corresponding to a discretization of the latitude circle by a total of 40 sites. There are $J = 4$ small-scale variables associated with each u_i. The constant forcing $f = 4$ makes the system to be chaotic. The parameters h, c_i, and b are chosen in such a way that the small-scale variables have a comparatively significant impact on the large-scale ones. In other words, the perfect model only has a weak scale separation. The reason that we consider such a weak scale separation is that it better mimics the real atmosphere with chaotic/turbulent behavior, the effect that the small-scale variables to the large ones cannot purely replaced by a simple parameterization. The parameter c_i varies across the spatial sites, which aims to mimic the fact that the coupling across the variables above the ocean is weaker than that above the land since the latter usually have stronger friction or dissipation. In this sense, the model is inhomogeneous. Finally, additional stochastic noise is added to the system, representing the contribution of the variables that are not explicitly modeled. The noise also interacts with the deterministic part via nonlinear terms, introducing additional complexity that mimics nature. To summarize, the parameters used in

the perfect model (3.27) are as follows,

$$\begin{gathered} I = 40, \quad J = 4, \quad h = 2, \quad c_i = 2 + 0.7\cos(2\pi i/I), \quad b = 2, \quad f = 4, \\ \sigma_{u_i} = \sigma_u = 0.2, \quad \sigma_{v_{i,j}} = \sigma_v = 1. \end{gathered} \tag{3.28}$$

3.5.2.2 The approximate model

Since in general the perfect model is not always fully known, or it is too complicated to be used in practice, it is essential to develop a simple and computationally tractable approximate model, which is nevertheless able to capture the key nonlinear feedback from the unobserved variables ($v_{i,j}$ here) to the observed variables (u_i here). As was discussed in Section 3.3, stochastic parameterization is widely used in describing chaotic signals (Chen and Majda, 2016a), which replaces the nonlinear eddy terms by quasilinear stochastic processes on formally infinite embedded domains where the stochastic processes are Gaussian conditional to the large scale mean flow. In addition, physics-constraint is adopted in designing approximate models, which also includes the effects from the large-scale u_i to the small-scale variables $v_{i,j}$. Therefore, such approximate models can potentially be used as surrogate models of the perfect model. The approximate model that we introduce

for (3.27) is as follows

$$\frac{du_i}{dt} = \left(-u_{i-1}(u_{i-2} - u_{i+1}) - u_i + \hat{f}_i - \hat{a}_i \sum_{j=1}^{J} v_{i,j} \right) + \hat{\sigma}_{u_i} \dot{W}_{u_i}, \quad i = 1, \ldots, I, \tag{3.29a}$$

$$\frac{dv_{i,j}}{dt} = -\hat{d}_{i,j} v_{i,j} + \hat{v}_{i,j} + \hat{c}_i u_i + \hat{\sigma}_{v_{i,j}} \dot{W}_{v_{i,j}}, \quad j = 1, \ldots, J, \tag{3.29b}$$

where $\hat{f}_i$, $\hat{a}_i$, $\hat{\sigma}_{u_i}$, $\hat{d}_{i,j}$, $\hat{v}_{i,j}$, $\hat{c}_i$, and $\hat{\sigma}_{v_{i,j}}$ are unknown constants. Compared with the original system (3.27), the main simplification here is in the small-scale equations, where we have replaced the nonlinear interactions $bc_i v_{i,j+1}(v_{i,j+2} - v_{i,j-1})$ in (3.27b) by simpler linear terms involving $v_{i,j}$ and u_i. The equations for u_i are essentially the same as before, although $\hat{f}_i$ is allowed now to vary from site to site. In the numerical experiments below, we will enforce $\hat{a}_i = \hat{c}_i$ for simplicity.

One desirable feature of this approximate model is that the direct coupling of the state variables only involves u_i and the corresponding $v_{i,j}$ for each fixed i. This is different from the original system (3.27) where u_i can have direct interactions with $v_{i+1,1}$ and $v_{i-1,J}$ given periodic conditions of the small scale $v_{i,j}$ over both i and j. Such a property allows to use a block decomposition of the covariance matrix of the smoother estimate during both E-step and M-step (Chen, 2020b). The entire state space for all the variables $\{u_i, v_{i,j} \mid i = 1, \ldots, I, j = 1, \ldots J\}$ can be decomposed into I subspaces, where each subspace can be dealt with in parallel. The technical details are included in the Appendix B.1.2.1.

3.5.2.3 Setup of the numerical simulations

The true signal is obtained by integrating the inhomogeneous L96 model (3.27) using the Euler-Maruyama scheme with the parameters given by (3.28), a uniform time step size $\delta t = 2 \times 10^{-3}$, and zero initial condition. The same time step size and initial condition are adopted for the approximate model (3.29) as well as the identified model, where the latter takes the same form as (3.27) but with estimated parameters. The true signals of u_i with 100 time units are used as the observations for parameter estimation while longer data with 1000 time units are used to compute their statistics. This latter length is also adopted when computing the statistics of the variables in the approximate and the identified models. The total number of the EM loops with CGNS is fixed to be 200 in this experiment.

3.5.2.4 CGNS as a preconditioner for identifying parameters in the perfect model

We first discuss the results of applying the EM algorithm to estimate the parameters in the approximate model (3.29). Figure 3.8 shows the trace plots at the site $i = 2$ corresponding to the variables u_2 and $v_{2,j}$ (in the first and the third rows) and the final estimation of the parameters in (3.29) (in the second and the fourth rows). The trace plots at other spatial locations have similar behavior; the parameters involved in the u_i-equations all converge quickly (within 10 iteration steps), and those involved in the $v_{i,j}$-equations converge at a relatively slower speed, but all stabilized after about 100 iterations. The black curves are shown as a reference, assuming $\hat{f}_i$, $\hat{a}_i$ $(= hc_i/J)$, $\hat{\sigma}_{u_i}$, and $\hat{c}_i$ are known and equal to those in the perfect

model and $\hat{d}_{i,j} = \hat{d}_i$, $\hat{\nu}_{i,j} = \hat{\nu}_i$, and $\hat{\sigma}_{\nu_{i,j}} = \hat{\sigma}_{\nu_i}$ are calibrated by the true statistics, i.e., the mean, the variance, and the decorrelation time, in the perfect model of $\nu_{i,j}$ averaged over j. The EM algorithm with the approximate model (3.29) can provide an accurate approximation of the parameters corresponding to the Gaussian fit of $\nu_{i,j}$ in the true signal. The conditional distribution of approximate model (3.29) with the estimated parameters is shown in Figure 3.9. Panel (a) and (c) show the true signal of two large-scale modes u_{10} and u_{20}, and Panel (c) and (d) show the true signal, and smoother estimate of the two hidden modes, where the smoother mean is given the black dashed curves, and one, two, and three standard derivations of the uncertainty are shown in the light, moderate, and dark shading areas. The shading areas cover most of the true signal, which indicates appropriate amount of uncertainties are obtained by combining the true observations and the approximate model. In fact, characterizing appropriate amount of uncertainty plays an important role in the EM algorithm when the hidden process contains large uncertainty. If uncertainty is totally ignored in the EM loop, the solution of conditional distribution of the approximate model can easily blow up.

Figure 3.10 shows the estimated parameters for the perfect model, where the CGNS is utilized as a preconditioner following Algorithm 1. Due to the intrinsic model error of the approximate model where the hidden variables are fully decomposed, in the sense that the correlation between the small scales corresponding to different large scales are omitted, resulting in 0 value of the bc_i term. However, other parameters, for example f and c_i, are adjusted accordingly. As a result, the identified model with estimated parameters resembles the truth to a remarkable

extent. Figure 3.11 shows the hovmoller plot of the large scales from different models. Although the estimated parameters is not perfect, the identified perfect model (Panel (b)) are almost the same as the true signal (Panel (a)). One realization of the identified perfect model with estimated parameters and that with true parameters (3.28) are shown in the first four row of Figure 3.12. The overall behavior of the trajectories from the model with estimated parameters resembles the truth. In addition, despite the small error, the ACFs and the PFDs are recovered with high accuracy.

3.5.2.5 CGNS as a surrogate model

The approximate model (3.29) with estimated parameters itself can be exploited as a surrogate model of the perfect system, which can be applied for ensemble forecast and other tasks. In fact, as is shown in Panel (c) of the Figure 3.11, the hovmoller plot of approximate model is almost the same as that of the perfect model (3.27). The performance of the approximate model serving as a surrogate model can also be found from the last four row of Figure 3.12, where a single realization of the trajectory, and the statistics from the approximate model with estimated parameters are all similar to the truth. Note that the single realization from the approximate model is not expected to follow the truth at all time instants since the system is chaotic.

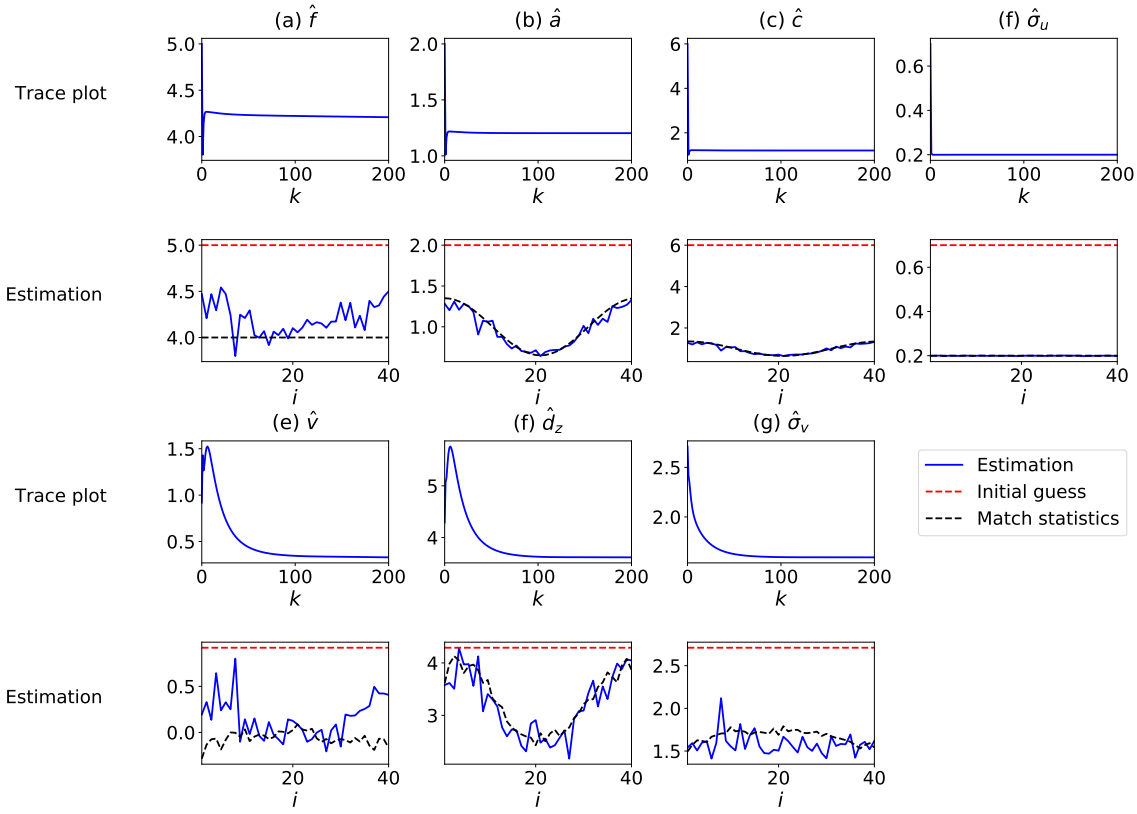

Figure 3.8: Learning the parameters of the approximate L96 model (3.29). The panels (a)–(d) and (i)–(k) in the 1st and 3rd rows show the trace plots of the estimated parameters of the EM algorithm at site $i = 2$. The panels (e)–(h) and (l)–(n) in the 2nd and 4th rows show the final estimated parameters θ_K^M (blue) after the K-th step of the EM algorithm with K taken here to be 200, the initial guesses of the parameters (red), and the reference parameters assuming $\hat{f}_i$, $\hat{a}_i$ $(= hc_i/J)$, $\hat{\sigma}_{u_i}$, and $\hat{c}_i$ are known and equal to those in the perfect model and $\hat{d}_{i,j} = \hat{d}_i$, $\hat{v}_{i,j} = \hat{v}_i$, and $\hat{\sigma}_{v_{i,j}} = \hat{\sigma}_{v_i}$ are calibrated by the true statistics, i.e., the mean, the variance, and the decorrelation time, in the perfect model of $v_{i,j}$ averaged over j.

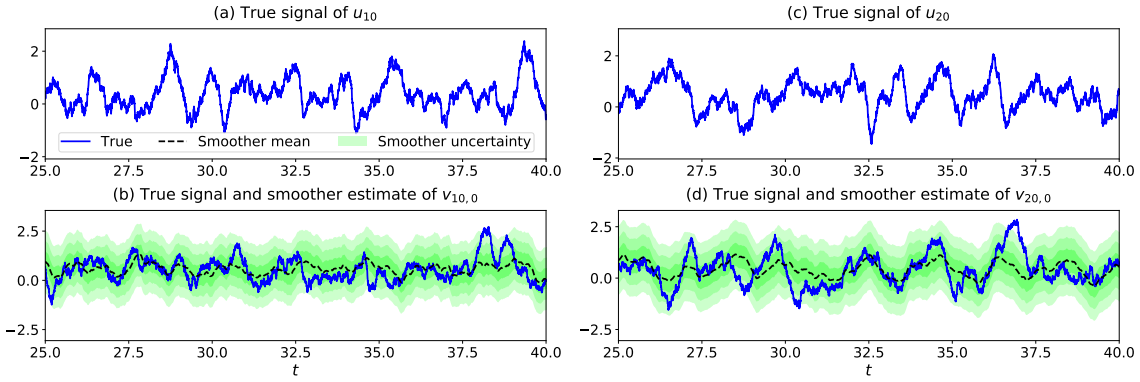

Figure 3.9: Smoother estimate of the approximate model in the K-th iteration. The blue curves: true signals; the black dashed curves: the smoother mean time series of the hidden variable; the light, moderate, and dark shading areas show the one, two, and three standard derivations (STDs) of the uncertainty in the smoother estimate. Panel (a): true signal of u_{10}; Panel (b): true signal of $v_{10,1}$ with one, two, and three, standard derivations of the uncertainty. Panel (c)–(d): similar to Panel (a)–(b) but for u_{20} and $v_{20,1}$.

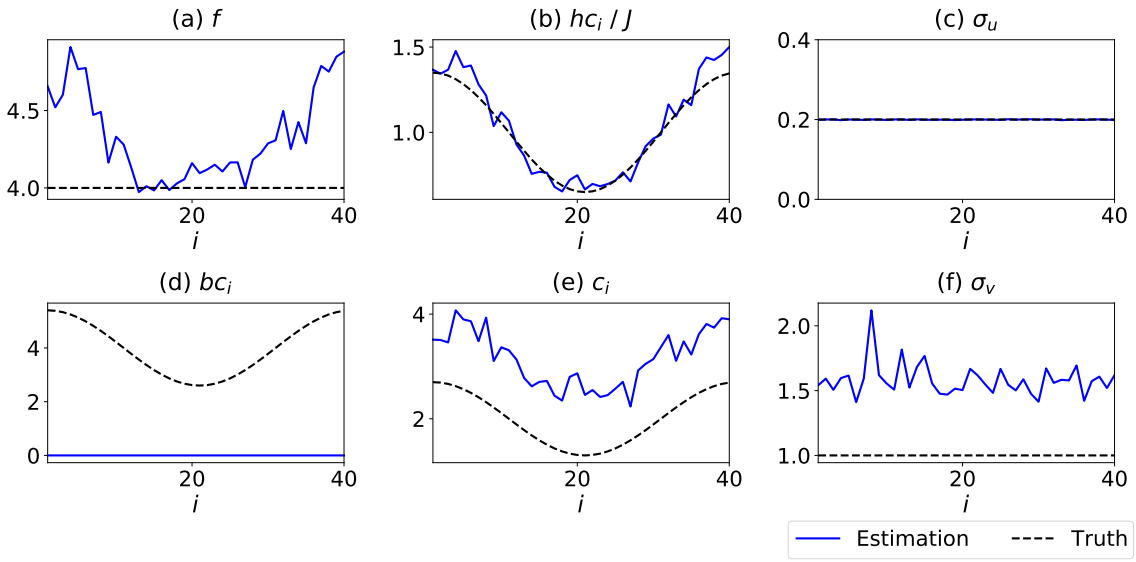

Figure 3.10: The estimated parameters of the perfect L96 model (3.27) after line 8 using Algorithm 1 (,i.e., θ_{K+1}). The black dash curves: the true values in the perfect model; the blue curves: the estimated parameters. Panel (a)–(f): parameters f, hc_i/J, σ_u, bc_i, c_i, σ_v, respectively.

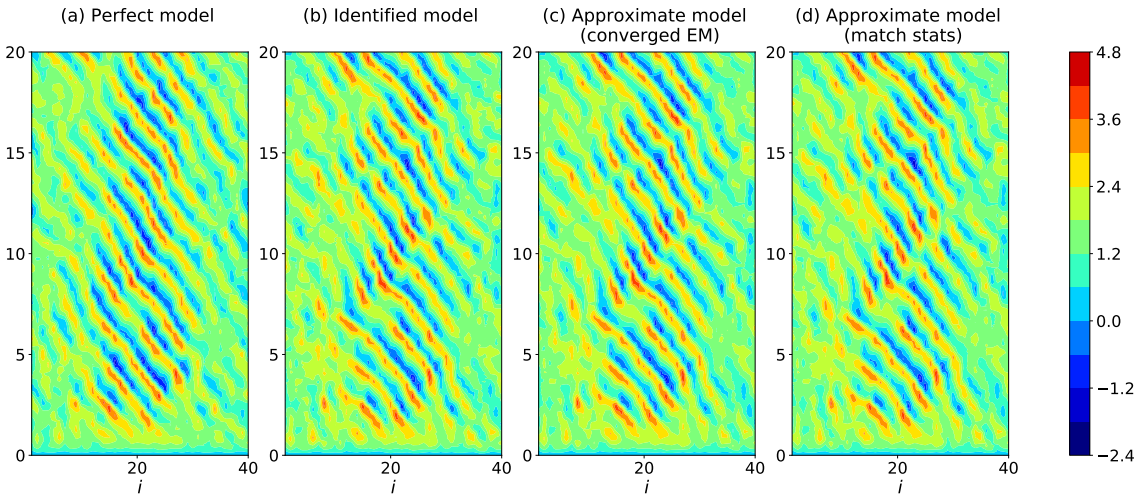

Figure 3.11: Hovmoller of large scales from different models. Panel (a): perfect model (3.27) with parameters (3.28); Panel (b): perfect model (3.27) with estimated parameters θ_{K+1} after line 8 using Algorithm 1; Panel (c): approximate model (3.29) with parameters θ_K^M with K.

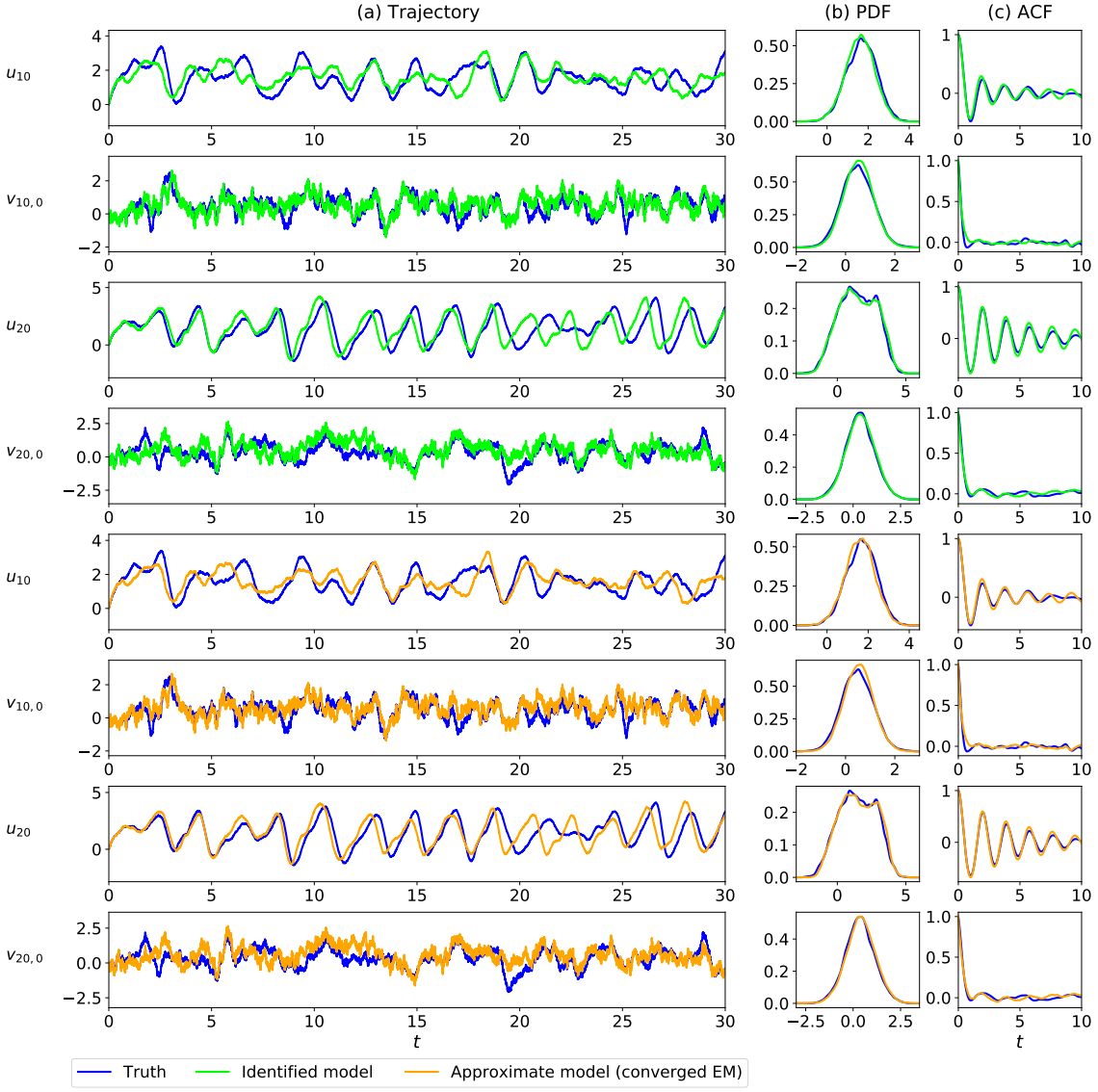

Figure 3.12: Comparison of the perfect model two-layer L96 model (3.27) with parameters (3.28) (blue), identified model (3.27) with estimated parameters θ_{K+1} (green), and approximate model (3.29) with parameters θ_K^M (orange) after K-th iteration of Algorithm 1. Panel (a)–(c): trajectory, PDF, and ACF, respectively.

3.6 Predicting the Statistical Response

Yet another important topic in studying complex nonlinear systems is to predict the model response to the perturbation of the external forcing. Developing efficient and accurate approaches to study such a key issue facilitates understanding model sensitivity, regime-switching behavior, and nonlinear interactions across different scales. Resolving this problem using advanced mathematical tools also has significant practical implications, such as coping with the climate change scenario. Due to various uncertainties from the internal instability and external forcing, a probabilistic description is more suitable for characterizing the complex turbulent systems (3.1). However, there exist several challenges in predicting the statistical response of complex nonlinear systems. First, solving the high-dimensional Fokker-Planck equation is the prerequisite of obtaining the model statistics, which, however, often suffers from the curse of dimensionality. Second, since the perfect system is not always computationally feasible in practice, the predicted model response may become inaccurate when an approximate model is utilized. It is then important to take advantage of a suitable combination of partial observations with the approximate model to mitigate error in calculating the model statistics. Third, due to the nonlinear nature of the systems, computing the statistical response in terms of the perturbations with different strengths and categories requires repeatedly solving the Fokker-Planck equation. As a consequence, even with a relatively fast solver of the Fokker-Planck equation, the total computational cost can still remain significant.

The advantage of the exact and statistically accurate solver of the equilibrium

PDF (3.9) makes the CGNS a natural framework for the development of approximate models for efficiently computing the model statistics and the associated response. One important feature in finding the PDF of the CGNS based on the formula (3.9) is that the available partially observed time series $\mathbf{X}$ is used to compute the conditional distribution of $\mathbf{Y}$ given $\mathbf{X}$. As a consequence, the model error in the PDF associated with the direct simulation of the approximate model is mitigated with the help of observations in computing the PDF based on (3.9). In other words, the resulting PDF from (3.9) is in general closer to the truth than that computed purely based on the approximate model without taking into account any input information from observations. What remains is the third challenge mentioned above. To overcome such a difficulty, the linear statistical response is utilized as an approximate method for computing the exact statistical response. The linear response only requires a linearization of the statistical equation while there is no linearization involved in the original nonlinear dynamics. Therefore, the nonlinear features of the underlying dynamics is preserved. In addition, the linear response can be computed utilizing the fluctuation-dissipation theorem (FDT) (Majda et al., 2005), which involves only a single PDF for computing different linear responses. Note that such a PDF is the equilibrium distribution of the unperturbed state, which can be efficiently solved using the formula (3.10).

3.6.1 Computing the linear statistical response via the fluctuation-dissipation theorem (FDT)

Consider the general complex nonlinear systems in (3.1). Defining $\mathbf{G}(\mathbf{u}, t) = (L + D)\,\mathbf{u} + B\,(\mathbf{u}, \mathbf{u}) + \mathbf{F}(t)$, the model can be written in a concise form

$$\frac{d\mathbf{u}}{dt} = \mathbf{G}(\mathbf{u}, t) + \boldsymbol{\sigma}(\mathbf{u}, t)\dot{\mathbf{W}}. \tag{3.30}$$

The equilibrium statistics of some functional $A(\mathbf{u})$ associated with (3.30) is formulated as

$$\langle A(\mathbf{u})\rangle = \int A(\mathbf{u}) p_{eq}(\mathbf{u}) d\mathbf{u}, \tag{3.31}$$

where $p_{eq}(\mathbf{u})$ is the equilibrium PDF of $\mathbf{u}$ in (3.30). Now consider the dynamics in (3.30) by a small time separable external forcing perturbation $\delta\mathbf{w}(\mathbf{u})f(t)$, where δ is a small scalar $\mathbf{w}$ is a general nonlinear function of $\mathbf{u}$. The perturbed system reads

$$\frac{d\mathbf{u}}{dt} = \mathbf{G}(\mathbf{u}, t) + \delta\mathbf{w}(\mathbf{u})f(t) + \boldsymbol{\sigma}(\mathbf{u}, t)\dot{\mathbf{W}}. \tag{3.32}$$

The FDT states that if δ is small enough, then the leading-order correction to the statistics in (3.31) becomes

$$\delta\langle A(\mathbf{u})\rangle(t) = \delta \int_0^t \mathbf{R}(t - s) f(s) ds, \tag{3.33}$$

where $\mathbf{R}(t)$ is the linear response operator, which is calculated through correlation functions in the unperturbed dynamics:

$$\mathbf{R}(t) = \langle A(\mathbf{u}(t))\mathcal{B}(\mathbf{u}(0))\rangle, \qquad \mathcal{B}(\mathbf{u}) = -\frac{\mathrm{div}_{\mathbf{u}}(\mathbf{w}(\mathbf{u})p_{eq}(\mathbf{u}))}{p_{eq}(\mathbf{u})}. \tag{3.34}$$

See (Majda et al., 2005) for a rigorous derivation of (3.33)–(3.34). In particular, the above procedure of computing the linear response via the FDT does not require the linearization of the underlying complex nonlinear systems. Therefore, the features of the nonlinear dynamics are preserved. If the functional $A(\mathbf{u})$ in (3.33) is given by $A(\mathbf{u}) = \mathbf{u}$, then the response computed is for the statistical mean. Likewise, $A(\mathbf{u}) = (\mathbf{u} - \bar{\mathbf{u}})^2$ is used for computing the response in the variance.

3.6.2 Calculating the linear statistical response via the CGNS preconditioner

According to (3.34), the calculation of the linear statistical response via the FDT requires the information of

1. the equilibrium PDF $p_{eq}(\mathbf{u})$,
2. the time series $A(\mathbf{u})$, and
3. the correct formulation of $\mathcal{B}(\mathbf{u})$.

Even if the perfect model is known, directly solving the high-dimensional Fokker-Planck equation is often not computationally affordable. Therefore, a suitable CGNS, serving as a preconditioner, is utilized to find a suitable approximation of

the non-Gaussian equilibrium PDF p_{eq} in an efficient way. Specifically, the explicit formula in (3.10) is utilized to achieve this goal. Besides, the observations are also need to be incorporated in computing the equilibrium p_{eq} (and later in recovering the hidden components in $A(\mathbf{u})$) to reduce model error from the approximate model free run. We denote one realization of the observations by $\mathbf{X}^{obs}$, the posterior distribution (filter and smoother) given $\mathbf{X}^{obs}$ by $p^{M|obs}$ where M indicates that the approximate model is used in computing the filter distribution (3.4), the explicit formula (3.10) becomes

$$p_{eq}^{M|obs}(\mathbf{X},\mathbf{Y}) = \lim_{J\to\infty}\frac{1}{J}\sum_{j=1}^{J}\Big(K_{\mathbf{H}}(\mathbf{X}-\mathbf{X}^{obs}(t_j))p^{M|obs}(\mathbf{Y}|\mathbf{X}^{obs}(s\leqslant t_j))\Big), \tag{3.35}$$

where $p_{eq}^{M|obs}$ is an efficient and effective approximation of the true equilibrium PDF p_{eq}.

Next, in the presence of partial observations, the conditional sampling formula (3.7) is exploited to calculate the unobserved component of the time series in $A(\mathbf{u})$. Note that the approximate model is used to compute the filter distribution (3.4), the smoother distribution formulae (3.6) and the conditional sampling formula (3.7). It is remarkable to note that the partial observations are involved in computing both $p_{eq}(\mathbf{u})$ and $A(\mathbf{u})$, which automatically mitigate the model error in the approximate model in both the equilibrium PDF and time series of the unobserved components. In addition, with $p_{eq}(\mathbf{u})$ and $A(\mathbf{u})$ obtained from the CGNS preconditioner, the original nonlinear system structure is utilized to form $\mathcal{B}(\mathbf{u})$ to compute the linear response $\mathbf{R}(t)$. The entire procedure of the FDT via CGNS is

given in Algorithm 2.

Algorithm 2: FDT with the CGNS preconditioner

1 Start with a given realization of the observations $\mathbf{X}^{obs}$;

2 Propose an approximate model that belongs to CGNS (3.2);

3 Compute the filter posterior distribution $p^{\mathcal{M}|obs}(\mathbf{Y}|\mathbf{X}^{obs}(s \leqslant t_i))$ via (3.4) ;

4 Form the equilibrium $p_{eq}^{\mathcal{M}|obs}$ via equation (3.35);

5 Compute the smoother posterior distribution $p^{\mathcal{M}|obs}(\mathbf{Y}(t)|\mathbf{X}^{obs}(s), s \in [0, T])$ from (3.6);

6 Sample one realization of the hidden time series via (3.7) that is used to approximate the unobserved component of $A(\mathbf{u})$;

7 Compute the response operator $\mathbf{R}(t)$ via (3.34) and compute the linear response via (3.33).

3.6.3 A 4D stochastic climate model

This section utilizes a four-mode stochastic model with key features of atmospheric low-frequency variability to show how to use CGNS as a preconditioner incorporated with partial observations to calculate the linear statistical response.

3.6.3.1 The perfect model

The stochastic climate model is designed in such a way that it involves many of the major dynamical properties of comprehensive global circulation models (GCMs) but with only four degree of freedom (Majda et al., 2008, 2005, 1999, 2001). The

model reads as follows

$$\frac{dx_1}{dt} = (-x_2(L_{12} + a_1x_1 + a_2x_2) - d_1x_1 + F_1 + L_{13}y_1 + b_{123}x_2y_1) + \sigma_1\dot{W}_{x_1}, \quad (3.36a)$$

$$\frac{dx_2}{dt} = (+x_1(L_{12} + a_1x_1 + a_2x_2) - d_2x_2 + F_2 + L_{24}y_2 + b_{213}x_1y_1) + \sigma_2\dot{W}_{x_2}, \quad (3.36b)$$

$$\frac{dy_1}{dt} = (-L_{13}x_1 + b_{312}x_1x_2 + F_3 - \gamma_1y_1) + \sigma_3\dot{W}_{y_1}, \quad (3.36c)$$

$$\frac{dy_2}{dt} = (-L_{24}x_2 + F_4 - \gamma_2y_2) + \sigma_4\dot{W}_{y_2}, \quad (3.36d)$$

where $b_{123} + b_{213} + b_{312} = 0$. Consistent with many geophysical flow models, the model has energy-conserving quadratic nonlinear terms, a linear operator, and external forcing terms. The linear operator contains two parts: one is a skew-symmetric component formally related to the Coriolis effect and topographic Rossby wave propagation; the other is a negative definite symmetric portion conceptually analogous to dissipative processes such as surface drag and radiative damping. The coupling in different variables is through both linear and nonlinear terms, where the nonlinear coupling through b_{ijk} produces multiplicative noise effects. In fact, the strategies described in Section 3.3.1 are applied to y_1 and y_2 that introduce the stochastic noise and damping terms. The variables x_1 and x_2 can be regarded as the climate variables and y_1 and y_2 represents the weather variables. The parameters

used to generate the true dynamics are as follows

$$\begin{aligned}
&d_1 = 1, \quad d_2 = 0.4, \quad \gamma_1 = 0.5, \quad \gamma_2 = 0.5, \quad L_{12} = 1, \quad L_{13} = 0.5, \quad L_{24} = 0.5, \\
&a_1 = 2, \quad a_2 = 1, \quad b_{123} = 1.5, \quad b_{213} = 1.5, \\
&\sigma_1 = 0.5, \quad \sigma_2 = 2, \quad \sigma_3 = 0.5, \quad \sigma_4 = 1, \quad F_1 = F_2 = F_3 = F_4 = 0.
\end{aligned} \tag{3.37}$$

One realization of the true signal is shown in black color in Figure 3.13. Both climate variable x_1 and weather variable y_1 have intermittent behavior with non-Gaussian PDFs. Note that this stochastic model is CGNS with $\mathbf{X} = (x_1, x_2)^\top$ and $\mathbf{Y} = (y_1, y_2)^\top$. We use a CGNS as the perfect model because the analytical solver is available, which helps to visualize presenting the results. Besides, the sampling error from computing the linear response via the FDT can be removed at this moment. In (3.36), we assume the variable x_1 and x_2 are observed variables.

3.6.3.2 The approximate model

In practice, running the entire perfect model is prohibitively costly. As a result, simpler or reduced models are commonly utilized in computing the responses. Linear stochastic models are widely used as approximate models for the unresolved variables (DelSole, 2005). Therefore, the hidden processes are replaced by two linear Gaussian equations, the parameters of which are calibrated by the true equilibrium statistics, i.e., the mean, the variance, and the decorrelation time. Besides, the parameters in the observed processes are assumed to be the same as those in the

perfect model (3.36). The approximate model reads,

$$\frac{dx_1}{dt} = (-x_2(L_{12} + a_1x_1 + a_2x_2) - d_1x_1 + F_1 + L_{13}y_1 + b_{123}x_2y_1) + \sigma_1\dot{W}_{x_1}, \quad (3.38a)$$
$$\frac{dx_2}{dt} = (+x_1(L_{12} + a_1x_1 + a_2x_2) - d_2x_2 + F_2 + L_{24}y_2 + b_{213}x_1y_1) + \sigma_2\dot{W}_{x_2}, \quad (3.38b)$$
$$\frac{dy_1}{dt} = -\hat{\gamma}_3\,(y_1 - \hat{y}_1) + \hat{\sigma}_3\dot{W}_{y_1}, \quad (3.38c)$$
$$\frac{dy_2}{dt} = -\hat{\gamma}_4\,(y_2 - \hat{y}_2) + \hat{\sigma}_4\dot{W}_{y_1}, \quad (3.38d)$$

which belongs to the CGNS. Note that despite the simplicity of utilizing linear Gaussian models to approximate the hidden processes, one major issue in (3.38) is that the physics constraint is no longer satisfied in (3.38). Therefore, the model in (3.38) can contain large errors for a long simulation.

3.6.3.3 Calculating linear response via FDT

In this example, we aim at calculating the linear response to the perturbation of the external forcing and linear interaction parameters. Specifically, the following two perturbation cases are considered:

Case 1: perturb parameters of forcing in the observed processes, i.e., $F_1^\delta = F_2^\delta = 0.3$, $\delta\mathbf{w}(\mathbf{u})f(t) = (0.3, 0.3, 0, 0)^\top$;

Case 2: perturb parameters in linear interaction terms, i.e., $L_{13}^\delta = L_{24}^\delta = 0.1$, $\delta\mathbf{w}(\mathbf{u})f(t) = (0.1y_1, 0.1y_2, -0.1x_1, -0.1x_2)^\top$.

Note that in the second case, the parameters appear in both the observed and hidden processes. We compare the linear response in the following models:

Perfect FDT (or perfect model): The equilibrium PDF $p_{eq}(\mathbf{u})$, the time series

$A(\mathbf{u})$ and the formulation of $\mathcal{B}(\mathbf{u})$ are all from the perfect model (3.36).

Imperfect FDT (or imperfect model): The equilibrium PDF $p_{eq}(\mathbf{u})$, the time series $A(\mathbf{u})$ and the formulation of $\mathcal{B}(\mathbf{u})$ are all from the approximate model (3.38).

Concatenate FDT (or concatenate model): The equilibrium PDF $p_{eq}(\mathbf{u})$, the time series $A(\mathbf{u})$ are from the simple concatenation of the observations, i.e., the trajectories of observed variables x_1 and x_2 from the perfect model (3.36), and the trajectories of the hidden variables y_1 and y_2 from the approximate model (3.38) free-run. The formulation of $\mathcal{B}(\mathbf{u})$ is from the approximate model (3.38).

FDT with preconditioner (or preconditioner model): The equilibrium PDF $p_{eq}(\mathbf{u})$ is calculated via (3.10) where true observations and the CGNS approximate model (3.38) are utilized in computing $K_{\mathbf{H}}(\mathbf{X} - \mathbf{X}^{obs}(t_i))$ and $p^{M|obs}(\mathbf{Y}|\mathbf{X}(s \leqslant t_i))$. The time series $A(\mathbf{u})$ are generated by equation (3.7) with the CGNS approximate model and the true observations. The formulation of $\mathcal{B}(\mathbf{u})$ is from the perfect model.

The details of the $\mathcal{B}(\mathbf{u})$'s forms from the perfect model and approximate model can be found in Appendix B.2. In this experiment, the true signal is obtained using Euler-Maruyama scheme with a uniform time step $\delta t = 5 \times 10^{-3}$ and the total length is 1000 time units.

Before discussing the linear response using CGNS as a preconditioner, we start by showing some pre-requisites, the equilibrium covariance, and time series $A(\mathbf{u})$ from different models. The trajectories from the approximate model (3.38) free-run (red color) are shown in Figure 3.13. Note that due to the violation of the

energy-conserving constraint of the nonlinear terms in (3.38), the amplitude of x_1 and x_2 from the approximate model is much larger than the one from the perfect model (3.36). In addition, the PDF of y_1 is Gaussian by design, which is also different from the skewed PDF as in the perfect model. The model error can also be found in the comparison of the equilibrium covariance matrices of the perfect model (Panel (a)) and the approximate model (Panel (d)) in Figure 3.14. Due to large error caused by the simple parameterization of the hidden processes, one may consider concatenating the observations (from the perfect model) and the trajectories of the hidden variables y_1 and y_2 from the approximate model (3.38) free-run to approximate the required equilibrium PDF p_{eq} and the unobserved time series. However, it is expected that the correlation between the observations from the perfect model and trajectories from the approximate model free-run is neglected. See Panel (c) of Figure 3.14. In contrast, in light of the desired structures of CGNS, the partial observations can be incorporated with both approximating the equilibrium PDF and recovering the unobserved times series. Therefore, the model error is significantly mitigated, and the correlation between the observed and unobserved variables is reserved. The green curves in Figure 3.13 show one sampled trajectories of y_1 and y_2 using conditional sampling formula (3.7) from the approximate model (3.38) and observations. The overall dynamics of the recovered sampled trajectories are very similar to those in the perfect model. In addition, the skewed PDF of y_1 can be found in the green curve but the appropriate model free run only brings Gaussian statistics by design. More importantly, the correlation between the sampled trajectories and the observations is consistent with that in

the perfect model as shown in Panel (b) of Figure 3.14. A final remark is that the smoother (or filter) mean time series is widely used as a surrogate of the true signals, which, however, underestimate the uncertainty as shown in the orange color in Figure 3.13. Therefore, conditional sampling is essentially in approximating the time series of $A(\mathbf{u})$.

Figure 3.15 shows the linear response operator $R(t)$ in (3.34) for the response of the first four moments when perturbing the external forcing parameters in the observed processes. Here, the blue and red curves show the linear response from the perfect model (3.36) and the free-run of the approximate model (3.38), respectively. The magenta curves show the linear response from simply concatenating the trajectories of observed variables x_1 and x_2 from the perfect model (3.36) and the trajectories of the hidden variables y_1 and y_2 from the approximate model (3.38). The green curves show the linear response from the procedure discussed in Section 3.6.2. The imperfect FDT contains huge errors in capturing higher moments for the observed variables x_1 and x_2. Concatenate FDT works well for computing the four moments of x_1; however, it is gradually away from the perfect FDT from lower moments to higher moments. For example, the gap between the concatenate FDT and the perfect FDT is obvious in the last row of x_2. This is because the strong correlation between the x_2 and y_1 (shown in Panel (a) in Figure 3.14) is omitted in this simple concatenation (shown in Panel (c) in Figure 3.14). In contrast, the response operator $R(t)$ of the two observed variables from preconditioner FDT is very close to the perfect FDT. Due to the indirect perturbation of the unresolved variables, the response is expected to be small. The preconditioner FDT can still capture the trend

of the linear response operator as that using perfect FDT. Figure 3.16 shows the second perturbation case, i.e., perturbing the linear interaction parameters appearing in both the observed and the hidden processes. In addition to the model error of the approximate model being mitigated and correlation between the observations and hidden dimensions of the approximate model being reserved, the perfect model structure is utilized to calculate the $\mathcal{B}(\mathbf{u})$ in FDT with preconditioner. Therefore, the performance of the FDT with preconditioner outperforms concatenate FDT and imperfect FDT.

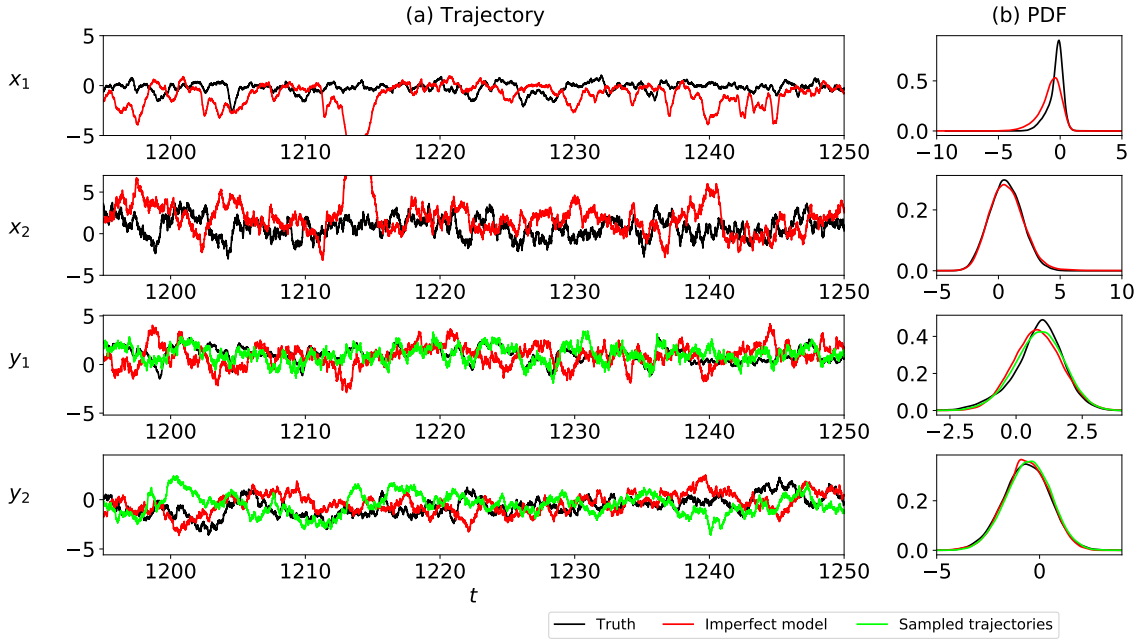

Figure 3.13: Comparison of the trajectories from the perfect 4D stochastic climate model (3.36), approximate model (3.38), the smoother mean time series, and the sampled trajectories based on the approximate model. Blue curves: perfect model trajectories; red curves: approximate model trajectories; orange curves: the smoother mean time series; green curves: sampled trajectories. Panel (a): trajectories; Panel (b): PDFs.

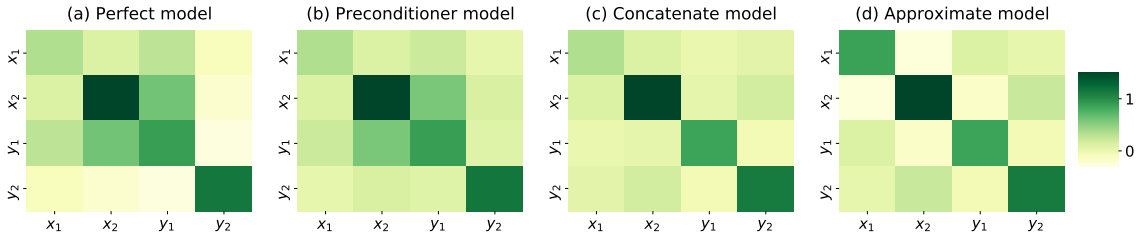

Figure 3.14: Comparison of covariance matrices in different scenarios. Panel (a): the perfect model where the equilibrium PDF is from the perfect model (3.36); Panel (b): preconditioner model where equilibrium PDF is from (3.4) where true observations and approximate model are utilized in computing $K_{\mathbf{H}}(\mathbf{X} - \mathbf{X}^{\text{obs}}(t_i))$ and $p^{\mathcal{M}|\text{obs}}(\mathbf{Y}|\mathbf{X}(s \leqslant t_i))$; Panel (c): concatenate model where the equilibrium PDF p_{eq} is from the simple concatenation of the observations, i.e., the trajectories of observed variables x_1 and x_2 from the perfect model (3.36) and the trajectories of the hidden variables y_1 and y_2 from the approximate model (3.38) free-run; Panel (d): the imperfect model where the equilibrium PDF is from the approximate model (3.38).

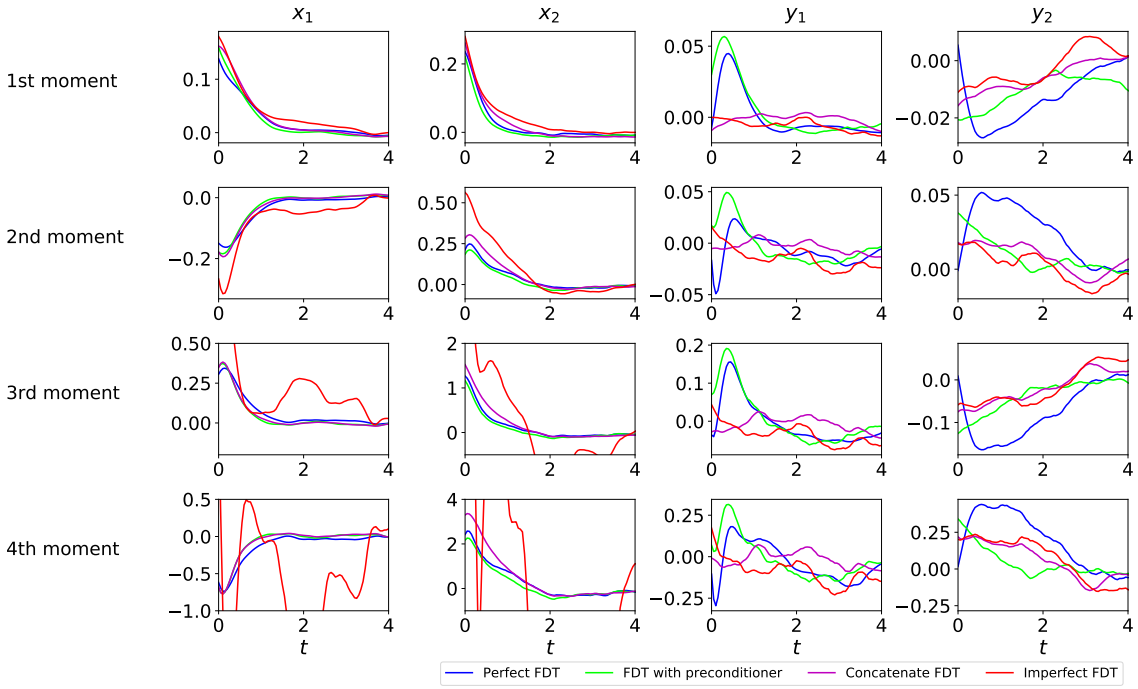

Figure 3.15: Response operator $\mathbf{R}(t)$ in (3.34) for the response of the first four moments of the 4D climate model (3.36) when perturbing parameters in the observed processes with $F_1^\delta = F_2^\delta = 0.3$. In each panel, the blue and red curves show the linear response from the perfect model (3.36) and the free-run of the approximate model (3.38), respectively. The magenta curves show linear response from simply concatenating the trajectories of observed variables x_1 and x_2 from the perfect model (3.36) and the trajectories of the hidden variables y_1 and y_2 from the approximate model (3.38). The green curves show the linear response from the procedure discussed in Section 3.6.2. Panel (a)–(d): x_1, x_2, y_1, and y_2, respectively.

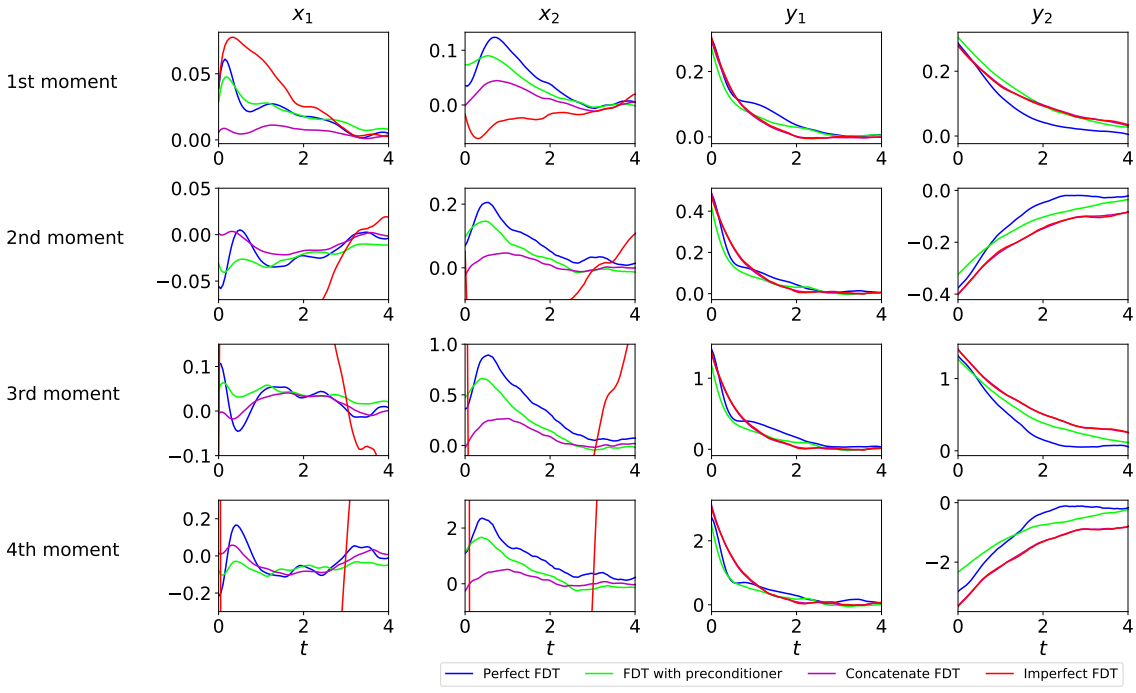

Figure 3.16: Similar to Figure 3.15 but perturbing parameters in the linear interaction terms $L^{\delta}_{13} = L^{\delta}_{24} = 0.1$. Panel (a)–(d): x_1, x_2, y_1, and y_2, respectively.

3.7 Discussion and Conclusions

In this paper, the skill of a rich class of nonlinear stochastic models, known as the "conditional Gaussian nonlinear system" (CGNS) (3.2), as both a cheap surrogate model and a fast preconditioner is explored to advance many computationally challenging tasks in complex nonlinear systems. The CGNS not only preserves the main underlying physics of nature but reproduces the observed intermittency, extreme events and other non-Gaussian features as well. The closed analytic formula of solving the conditional statistics facilitates the development of many mathematical theories and fast numerical algorithms. Three topics are covered in this paper. First, the closed analytic formulae of the conditional statistics of the CGNS allow an efficient and accurate data assimilation scheme. It is shown in Section 3.4 that the data assimilation skill of a suitable CGNS approximate forecast model outweighs the EnKBF applying directly even to the perfect model in the presence of strong nonlinear and turbulent features. The latter may suffer from filter divergence when the observational process is highly nonlinear with small observational uncertainties. Second, as is shown in Section 3.5, the CGNS allows the development of a fast algorithm for simultaneously estimating the parameters and the unobserved state variables with uncertainty quantification in the presence of only partial observations. Utilizing an appropriate CGNS as a preconditioner significantly reduces the computational cost in accurately estimating the parameters in the original complex system. The same CGNS can also serve as a surrogate model for reproducing the large-scale dynamics and statistics of nature and can be applicable to ensemble forecast. Finally, the CGNS advances rapid and statistically accurate algorithms for

both computing the probability density function and sampling the trajectories of the unobserved state variables. Showing in Section 3.6, these fast algorithms facilitate the development of an efficient and accurate data-driven method for predicting the linear response of the original system with respect to parameter perturbations based on a suitable CGNS preconditioner.

Several important topics are remained as future work. First, it is crucial to systematically determine the CGNS approximate model. A promising approach is to write down the general structure of (3.12) and then apply a parameter estimation algorithm together with certain sparse identification method to prevent the overfitting issue. Second, it is of practical significance to develop further pathways that allow to apply the CGNS to more complicated systems and explore the approximate errors. Incorporating the CGNS into intermediate complicated models or even certain versions of the general circulation models (GCMs), say for climate science or geophysics, can be interesting and practically useful tasks. Finally, the CGNS may have the potential to combine with machine learning algorithms to advance the ensemble forecast. One possible direction is to exploit the analytic formula in (3.9) for the ensemble forecast, where the complicated nonlinear interactions in solving the conditional distributions can be replaced by a cheaper machine learning architecture.

4 EFFICIENT NONLINEAR FILTERING, SMOOTHING, FORWARD AND BACKWARD SAMPLING ALGORITHMS FOR PARTIALLY OBSERVED COMPLEX TURBULENT NONLINEAR DYNAMICAL SYSTEMS WITH INTERMITTENCY AND EXTREME EVENTS

4.1 Introduction

State estimation of unobserved or unresolved variables in complex nonlinear turbulent dynamical systems, given partially observed time series, is an important topic in practice. Filtering is one of the most widely used approaches for state estimation (Kalnay, 2003; Lahoz et al., 2010; Majda and Harlim, 2012a; Evensen, 2009; Law et al., 2015), where the estimated state is given by the so-called posterior distribution. One major advantage of the filtering technique is that it adopts the observational information only up to the current time instant and thus the filter estimate is used as an improved initialization for real-time prediction. On the other hand, the smoothing technique (Rauch et al., 1965; Särkkä, 2013) exploits the information in the entire observational period, including both the past and the future data, and it is extensively used for postprocessing. Due to the extra observational information, the state estimation from smoothing is expected to be more accurate and less uncertain than that from filtering, at least in the perfect model setup. When the underlying dynamics and the observational operator are linear and the system and observational noises are Gaussian, the corresponding filtering

and smoothing techniques are the celebrated Kalman filter and the Rauch-Tung-Striebel (RTS) smoother (Kalman and Bucy, 1961; Bucy and Joseph, 2005; Rauch et al., 1965). On the other hand, strong nonlinear and non-Gaussian phenomena, including intermittency and extreme events (Farazmand and Sapsis, 2018; Denny et al., 2009; Mohamad and Sapsis, 2015), are key features in nature. Thus, a large number of numerical or approximate methods have been proposed for filtering nonlinear and non-Gaussian systems, such as the ensemble Kalman filters and the particle (or sequential Monte Carlo) filters (Evensen, 2009; Law et al., 2015; Anderson, 2001; Del Moral, 1997; Liu and Chen, 1998). Correspondingly, the ensemble Kalman smoother (Evensen and Van Leeuwen, 2000) and the Monte Carlo smoother (Kitagawa, 1996) have also been developed. These numerical methods are widely applied in different areas and lead to great success in many practical issues.

However, due to the considerable computational cost of simulating complex nonlinear dynamical systems, especially in high dimensions, the number of the particles or ensembles that are allowed to use in these numerical methods are very limited. This may lead to a significant bias in the state estimation of nonlinear systems and even a finite-time blowup of the solution (Harlim et al., 2010; Gottwald and Majda, 2013). In addition, the Gaussian approximation in the ensemble Kalman filter/smoother and the insufficient number of samples in the particle methods may end up with large errors in recovering the non-Gaussian probability density functions (PDFs) and predicting extreme events. Various remedies, such as noise inflation, localization and resampling, have been developed (Greybush et al., 2011;

Anderson, 2001; Hol et al., 2006), which improve the numerical behaviors of these particle based methods. Nevertheless, incorporating closed analytic formulae into the state estimation algorithms is highly preferable for enhancing the computational efficiency as well as promoting the accuracy of capturing salient non-Gaussian features such as rare and extreme events. The closed analytic formulae also facilitate the theoretic study of the error and uncertainty in state estimation.

In this article, a recently developed nonlinear modeling framework (Liptser and Shiryaev, 2013; Chen and Majda, 2016a, 2018a), called conditional Gaussian nonlinear models, are used to study the filter-based and smoother-based nonlinear state estimation. One of the main advantages of such models is that closed analytic formulae are available for the nonlinear optimal filter and nonlinear optimal smoother, which facilitate rigorous mathematical analysis and efficient numerical simulations. The conditional Gaussian models are highly nonlinear and non-Gaussian, where both the joint and marginal PDFs can be skewed with fat tails. Extreme events, intermittency and highly nontrivial nonlinear interactions between different variables all appear in the conditional Gaussian systems. The name "conditional Gaussian" comes from the fact that once the trajectories of a subset of the variables are given, the statistics of the remaining variables conditioned on the given trajectories are Gaussian. A gallery of examples of conditional Gaussian systems in engineering, neuroscience, ecology, fluids and geophysical flows can be found in (Chen and Majda, 2018a).

In addition to the point-wise state estimation using filtering or smoothing techniques, another important and highly relevant topic is to sample the trajectories of

the hidden variables conditioned on the observations. These conditional sampled trajectories are completely different from the time series by connecting independent samples drawn from the posterior distributions at different time instants. Instead, each point in the sampled trajectories takes into account the nonlinear temporal dependence with those at nearby time instants. The point-wise statistics of the sampled trajectories (namely, the statistics at each time instant) is nevertheless consistent with the posterior distribution from filtering or smoothing. Notably, the path-wise information provided by the sampled trajectories allows the study of many crucial dynamical characteristics, such as the temporal autocorrelation function (ACF) and the causality between the observed and hidden extreme events, which are not available from the point-wise statistics using filtering or smoothing. While sampling the hidden trajectories in general nonlinear systems can be an extremely computationally challenging task, the conditional Gaussian nonlinear models allow closed analytic formulae for both the filter-based forward sampling and smoother-based backward sampling. These explicit formulae facilitate a systematic comparison of the two methods in reproducing various nonlinear and non-Gaussian features in the hidden processes and advance the understanding of the onset, demise and the dynamical evolution of the intermittent extreme events. The sampled trajectories can also be used to illustrate the lack of information in the posterior mean time series and the approximation error in the affecting the overall dynamics by treating the posterior mean time series as a path-wise surrogate of the hidden trajectories, which is widely used in practice.

The traditional smoother-based state estimation and the associated sampling

approach require a forward pass of filtering the entire period before running a backward pass for smoothing (Rauch et al., 1965). However, this is often hard to be applied in practice since the method requires to restore the filtered state estimation in the entire period before running the backward pass. As the dimension of the system and the length of the observational period increase, a significant amount of storage is required. In addition, in many practical situations new observational data are often available in a sequential manner. Recalculating the entire forward-backward procedure from scratch when a new observation is included can be quite cumbersome. Therefore, it is highly desirable to develop an online version of the nonlinear smoother and the associated online smoother-based sampling algorithm that involve only a forward-in-time pass. As the traditional offline version, the recursive online smoother can be expressed by closed analytic formulae with the help of the analytically solvable conditional Gaussian statistics. The explicit formulae of the online smoother allows using rigorous mathematical analysis and accurate computational algorithms to understand the sequential update of the existing state estimates, especially for the recovery of extreme events, which can be quite challenging using the ensemble based methods. The closed form of the online nonlinear smoother also facilitates the online parameter estimation of complex nonlinear systems. In particular, it can be used to study the role of the observed extreme events in improving the online parameter estimation skill.

The rest of the article is organized as follow. The conditional Gaussian nonlinear modeling framework is presented in Section 4.2. The filter-based forward sampling and the smoother-based backward sampling of the hidden processes are presented

in Section 4.3. The online forward-in-time nonlinear optimal smoother and the associated sampling algorithm are shown in Section 4.4. Section 4.5 aims at using numerical simulations to compare the nonlinear filter and nonlinear smoother as well as the associated sampling strategies. The focus is on the state estimation, sampling hidden intermittent time series and detecting hidden extreme events in various nonlinear and non-Gaussian systems. A smoother-based online parameter estimation algorithm is shown in Section 4.6. The article is concluded in Section 4.7. All the proofs are included in Appendix.

4.2 Conditional Gaussian Nonlinear Systems

4.2.1 Special cases

4.2.1.1 The Kalman-Bucy filter

Consider the simplified version of (3.2), where all the matrices and vectors have no dependence on the variable $\mathbf{X}$ and the noises in the two processes are independent with each other. The resulting system is then a linear model with deterministic coefficients. In particular, the process of $\mathbf{Y}$ becomes independent of $\mathbf{X}$. Such a system is the classical Kalman-Bucy filter model (Kalman and Bucy, 1961; Bucy and Joseph, 2005),

$$d\mathbf{X}(t) = \Big[\mathbf{A_0}(t) + \mathbf{A_1}(t)\mathbf{Y}(t)\Big]dt + \mathbf{B_2}(t)d\mathbf{W_2}(t), \tag{4.1a}$$

$$d\mathbf{Y}(t) = \Big[\mathbf{a_0}(t) + \mathbf{a_1}(t)\mathbf{Y}(t)\Big]dt + \mathbf{b_1}(t)d\mathbf{W_1}(t), \tag{4.1b}$$

which aims at solving the conditional distribution $p(\mathbf{Y}(t)|\mathbf{X}(s), s \leqslant t)$. Note that in the general conditional Gaussian nonlinear framework, the filter estimate (3.4) involves solving a random Riccati equation for the covariance, which is not the case in the Kalman-Bucy model since all the coefficients of the filter estimate are state independent.

As an analog, the classical Kalman filter (Kalman and Bucy, 1961; Kalman, 1960) is a special case of the discrete version of the conditional Gaussian systems.

4.2.1.2 Stochastic parameterizations and Lagrangian data assimilation

In many situations, the coupled system (3.2) has the following specific form,

$$d\mathbf{X}(t) = \Big[\mathbf{A_0}(\mathbf{X}, t) + \mathbf{A_1}(\mathbf{X}, t)\mathbf{Y}(t)\Big]dt + \mathbf{B_2}(\mathbf{X}, t)d\mathbf{W_2}(t), \tag{4.2a}$$

$$d\mathbf{Y}(t) = \Big[\mathbf{a_0}(t) + \mathbf{a_1}(t)\mathbf{Y}(t)\Big]dt + \mathbf{b_1}(t)d\mathbf{W_1}(t), \tag{4.2b}$$

where $\mathbf{X}$ does not appear in the process of $\mathbf{Y}$. However, different from the Kalman-Bucy filter (4.1), the process of $\mathbf{X}$ in (4.2a) remains highly nonlinear. The coupled system (4.2a) has several important applications in practice.

a) Stochastic parameterization Many reduced order models involve parameterizing certain complicated processes, such as those related to the unresolved or the small-scale variables. However, the deterministic parameterization may not be sufficient to capture crucial turbulent and chaotic features. Therefore, the stochastic parameterization, which is designed to include the small-scale variability and uncertainties, becomes a widely used method in practice (Gershgorin et al., 2010b;

Berner et al., 2017; Andrejczuk et al., 2016; Plant and Craig, 2008). One of the typical stochastic parameterization approaches involves using linear processes with Gaussian noises (4.2b) for describing the unresolved variables, which nevertheless interact with the nonlinear large-scale dynamics and result in various non-Gaussian characteristics in (4.2a).

b) Lagrangian data assimilation In Lagrangian data assimilation (Apte et al., 2008b; Ide et al., 2002; Apte et al., 2008a), the process $\mathbf{X}$ represents the time evolution of the Lagrangian trajectories, which is usually highly nonlinear. On the other hand, the process $\mathbf{Y}$ describes the underlying flow field, which can be linear. The variable $\mathbf{Y}$ drives the observed tracer trajectories but is not affected by the tracers. Despite the intrinsic nonlinearity in the observed process, the coupled system belongs to the conditional Gaussian framework, which allows an explicit formula for obtaining the posterior estimate from the nonlinear Lagrangian data assimilation (Chen et al., 2014b, 2015; Chen and Majda, 2016b).

4.2.1.3 Nonlinear systems retaining the conditional Gaussianity in the reverse way

The conditional Gaussianity of $p(\mathbf{Y}(t)|\mathbf{X}(s), s \leqslant t)$ is an important feature of the nonlinear modeling framework (3.2). However, the conditional Gaussianity is in general not valid in the reverse way, meaning that the conditional distribution $p(\mathbf{X}(t)|\mathbf{Y}(s), s \leqslant t)$ is not guaranteed to be Gaussian. Nevertheless, there is a wide class of the nonlinear models, in which both $p(\mathbf{Y}(t)|\mathbf{X}(s), s \leqslant t)$ and $p(\mathbf{X}(t)|\mathbf{Y}(s), s \leqslant$

t) are Gaussian. These models are summarized as follows,

$$d\mathbf{X}(t) = \Big[\mathbf{L_{XX}}(t)\mathbf{X}(t) + \mathbf{L_{XY}}(t)\mathbf{Y}(t) + \mathbf{N_X}(\mathbf{X}(t), \mathbf{Y}(t)) + \mathbf{F_X}(t)\Big]dt + \mathbf{B_1}(t)d\mathbf{W_1}(t), \tag{4.3a}$$

$$d\mathbf{Y}(t) = \Big[\mathbf{L_{YX}}(t)\mathbf{X}(t) + \mathbf{L_{YY}}(t)\mathbf{Y}(t) + \mathbf{N_Y}(\mathbf{X}(t), \mathbf{Y}(t)) + \mathbf{F_Y}(t)\Big]dt + \mathbf{b_2}(t)d\mathbf{W_2}(t). \tag{4.3b}$$

In (4.3), $\mathbf{L_{XX}}(t)$, $\mathbf{L_{XY}}(t)$, $\mathbf{L_{YX}}(t)$ and $\mathbf{L_{YY}}(t)$ are deterministic functions, which are coefficients associated with the linear terms. The noise coefficients $\mathbf{B_1}(t)$, $\mathbf{B_2}(t)$, $\mathbf{b_1}(t)$ and $\mathbf{b_2}(t)$ are also deterministic and they do not depend on the state variables. The two nonlinear terms $\mathbf{N_X}(\mathbf{X}(t), \mathbf{Y}(t))$ and $\mathbf{N_Y}(\mathbf{X}(t), \mathbf{Y}(t))$ are bilinear.

The energy-conserving nonlinearity, also known as the physics constraints (Majda and Harlim, 2012b; Harlim et al., 2014), is satisfied in many nonlinear turbulent systems. This implies

$$\mathbf{X} \cdot \mathbf{N_X}(\mathbf{X}, \mathbf{Y}) + \mathbf{Y} \cdot \mathbf{N_Y}(\mathbf{X}, \mathbf{Y}) = 0. \tag{4.4}$$

Since the first term in (4.4) in a quadratic function of $\mathbf{X}$ while the second term is a quadratic function of $\mathbf{Y}$, the equation of (4.4) leads to the following conditions

$$\mathbf{X} \cdot \mathbf{N_X}(\mathbf{X}, \mathbf{Y}) = 0 \qquad \text{and} \qquad \mathbf{Y} \cdot \mathbf{N_Y}(\mathbf{X}, \mathbf{Y}) = 0. \tag{4.5}$$

This means the bilinear functions $\mathbf{N_X}(\mathbf{X},\mathbf{Y})$ and $\mathbf{N_Y}(\mathbf{X},\mathbf{Y})$ have the following forms

$$\mathbf{N_X}(\mathbf{X},\mathbf{Y}) = \mathbf{S_X}(\mathbf{Y})\mathbf{X}, \qquad \text{and} \qquad \mathbf{N_Y}(\mathbf{X},\mathbf{Y}) = \mathbf{S_Y}(\mathbf{X})\mathbf{Y}, \tag{4.6}$$

where $\mathbf{S_X}(\mathbf{Y})$ and $\mathbf{S_Y}(\mathbf{X})$ are both skew-symmetric matrices. Examples of such type of systems include many Lorenz-like systems or the Lorenz-Chen systems (Wang and Chen, 2013; Lainscsek, 2012).

4.3 Forward and Backward Sampling of Hidden Trajectories Conditioned on the Observations

4.3.1 Closed analytic formulae of computing the sampled trajectories

The posterior distribution from the nonlinear filter or smoother provides the point-wise optimal statistical state estimation. Here, "point-wise estimation" means the state estimation at a fixed time instant. In many applications, it is also important to sample trajectories of the hidden processes conditioned on the observations. Unfortunately, merely using the information from the posterior distributions is not sufficient to draw unbiased trajectories of the hidden variables conditioned on the observations. This is because, in addition to the point-wise statistical state estimates, each point in the sampled trajectories is required to take into account the nonlinear temporal dependence with those at nearby time instants. In practice, the posterior

mean time series is often used as a surrogate of the optimal trajectories of the hidden processes conditioned on the observations. However, the posterior uncertainty and its temporal correlation are completely ignored in such an approximation. The consequence, which will be shown in Section 4.5, is that the posterior mean time series fails to capture some of the key dynamical features, such as the temporal autocorrelation function (ACF) of the underlying dynamics. Note that a naive way of involving the posterior uncertainty in sampling hidden trajectories conditioned on the observations is to draw independent random numbers from the posterior distributions at different time instants and then connect them together. Yet, such an approach will not only leads to a noisy time series but fails to capture the underlying dynamical features as well due to the lack of incorporating the temporal correlation of the uncertainty.

Since the trajectories of the hidden variables conditioned on the observations allow the understanding of many dynamical features, developing an efficient and unbiased method with rigorous mathematical evidence to sample these trajectories has practical significance. The two theorems below provide closed analytic formulae for sampling the hidden trajectories conditioned on the observations.

Theorem 4.1 (Nonlinear Optimal Forward Sampling Formula)**.** *Conditioned on one realization of the observed variable* $\mathbf{X}(s)$ *for* $t \in [0, t]$*, the optimal strategy of sampling the trajectories associated with the unobserved variable* $\mathbf{Y}$ *at time* t *satisfies the following*

explicit formula,

$$\begin{aligned} d\mathbf{Y} = d\boldsymbol{\mu}_{\mathbf{f}} + \left(\mathbf{a_1} - (\mathbf{R_f}\mathbf{A_1^*})(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\mathbf{R_f^{-1}}\right)(\mathbf{Y}-\boldsymbol{\mu}_{\mathbf{f}})dt \\ + \left(\mathbf{b}\circ\mathbf{b} + (\mathbf{R_f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\right)^{1/2} d\mathbf{W_Y}, \end{aligned} \tag{4.7}$$

where $\mathbf{W_Y}$ *is an independent white noise source. The* $\cdot^{1/2}$ *in the second line of* (4.7) *is the square root of a matrix. Since* $\mathbf{b}\circ\mathbf{b} + (\mathbf{R_f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})$ *is a positive definite matrix, the square root is unique.*

See Appendix C.2 for the proof. Similar to the filtering technique, the forward sampling formula in Theorem (4.7) is a sequential method, which makes use of the information only in the past to compute the path-wise value at the current time instant.

The formula in (4.7) states that the sampled trajectory of $\mathbf{Y}$ is based on the posterior mean estimate of the filter $\boldsymbol{\mu}_{\mathbf{f}}$ (the first term on the right hand side of (4.7)) but it also includes temporal correlated uncertainties that depend on both the observations and the underlying nonlinear model. In fact, as is shown in Appendix C.2, with the mean stability of the system being satisfied, the real part of all the eigenvalues of $\mathbf{a_1} - (\mathbf{R_f}\mathbf{A_1^*})(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{B}\circ\mathbf{b} + \mathbf{A_1}\mathbf{R_f})\mathbf{R_f^{-1}}$ is negative, which implies that $\mathbf{Y}$ has a tendency going towards the filter mean state $\boldsymbol{\mu}_{\mathbf{f}}$. It is also worthwhile to point out that the noise coefficient is not simply $(\mathbf{b}\circ\mathbf{b})^{1/2}$ as in the original model (3.2b). The uncertainty represented by the noise also takes into account the information from observations, which is embodied in $\mathbf{R_f}$, $\mathbf{A_1}$ and $\mathbf{B}$. Despite this noise level being larger than $(\mathbf{b}\circ\mathbf{b})^{1/2}$ in (3.2b), the equation has a much larger damping than the

original model (3.2b) as well. Therefore, the uncertainty, represented by the equilibrium variance, in the sampled $\mathbf{Y}$ conditioned on the observations (4.7) is lower than the $\mathbf{Y}$ resulting from a free run of the coupled model (3.2). The reduction of the uncertainty comes from the information of the given observational time series.

Another useful conclusion that can be drawn from (4.7) is the following. The residual of the sampled trajectory $\mathbf{Y}$ with respect to the filter mean state $\boldsymbol{\mu}_\mathbf{f}$ is $\mathbf{Y}' = \mathbf{Y} - \boldsymbol{\mu}_\mathbf{f}$, which satisfies the equation

$$\begin{aligned} d\mathbf{Y}' = &\left(\mathbf{a_1} - (\mathbf{R_f}\mathbf{A_1^*})(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\mathbf{R_f^{-1}}\right)\mathbf{Y}'dt \\ &+ \left(\mathbf{b}\circ\mathbf{b} + (\mathbf{R_f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\right)^{1/2} d\mathbf{W_Y}. \end{aligned} \tag{4.8}$$

This indicates that time series of the residual, which reflects the uncertainty in the sampled trajectory, does have a highly non-trivial temporal dependence. The temporal dependence of $\mathbf{Y}'$ in (4.8) is very different from that of $\mathbf{Y}$ in the original system (3.2b). In fact, consider the simplest situation that both $\mathbf{X}$ and $\mathbf{Y}$ are one-dimensional variables and all the coefficients are constants with $\mathbf{A}_1 < 0$ and $\mathbf{a_1} < 0$ (and therefore $\mathbf{R_f}$ converges to a constant). Then the decorrelation time, which measures the memory of a stochastic system, is $1/(-\mathbf{a_1})$ and $1/(-\mathbf{a_1} + (\mathbf{R_f}\mathbf{A_1^*})(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\mathbf{R_f^{-1}})$ for the original system (3.2b) and the residual of the sampled trajectory (4.8), respectively. Such a difference is due to the extra information provided by the observations, which is incorporated into the sampled trajectories.

The above forward sampling formula is based on the nonlinear filter estimate (see also Corollary 4.4 below). Since the nonlinear smoother provides a more accurate and less uncertain state estimation, it is natural to study the smoother-

based sampling formula.

Theorem 4.2 (Nonlinear Optimal Backward Sampling Formula)**.** *Conditioned on one realization of the observed variable* $\mathbf{X}(t)$ *for* $t \in [0, T]$*, the optimal strategy of sampling the trajectories associated with the unobserved variable* $\mathbf{Y}$ *at any time* $t \in [0, T]$ *satisfies the following explicit formula,*

$$\overleftarrow{d\mathbf{Y}} = \big(-\mathbf{a_0} - \mathbf{a_1}\mathbf{Y}\big)dt + (\mathbf{b} \circ \mathbf{b})\mathbf{R_f}^{-1}\big(\boldsymbol{\mu}_\mathbf{f} - \mathbf{Y}\big)dt + \big(\mathbf{b} \circ \mathbf{b}\big)^{1/2} d\mathbf{W_Y}, \tag{4.9}$$

where $\mathbf{W_Y}$ *is an independent white noise source. As in Theorem 4.1, the square root of* $\mathbf{b} \circ \mathbf{b}$ *is unique. In* (4.9)*, the left hand side is understood as*

$$\overleftarrow{d\mathbf{Y}} = \lim_{\Delta t \to 0} \mathbf{Y}(t) - \mathbf{Y}(t + \Delta t).$$

The formula (4.9) *starts from* $t = T$ *and it is run backwards towards* $t = 0$*. Therefore, it is named as a backward sampling formula. The initial value of* $\mathbf{Y}$ *in* (4.9) *is drawn from the conditional Gaussian distribution from filtering* $\mathcal{N}(\boldsymbol{\mu}_\mathbf{f}(T), \mathbf{R_f}(T))$ *due to the fact that the filter and the smoother estimates are the same at the end point.*

The proof of Theorem 4.2 is similar to that in Theorem 4.1 but makes use of the nonlinear smoother estimate in (3.6). See (Chen and Majda, 2020) for the details.

Comparing with the true underlying dynamics of $\mathbf{Y}$ in (3.2b), the backward sampling equation (4.9) involves an extra term $(\mathbf{b} \circ \mathbf{b})\mathbf{R_f}^{-1}\big(\boldsymbol{\mu}_\mathbf{f} - \mathbf{Y}\big)dt$. This correction term is similar to the one in the nonlinear smoother mean equation (3.6a), which takes into account the information from observations and builds a connection

between the nonlinear smoother and the backward sampling. In light of (3.6a), an alternative way of backward sampling formula is given as follows.

Corollary 4.3 (An Alternative Backward Sampling Formula)**.**

$$\overleftarrow{\mathrm{d}\mathbf{Y}} = \overleftarrow{\mathrm{d}\boldsymbol{\mu}_{\mathbf{s}}} - \left(\mathbf{a_1} + (\mathbf{b} \circ \mathbf{b})\mathbf{R}_{\mathbf{f}}^{-1}\right)(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{s}})\mathrm{d}t + \left(\mathbf{b} \circ \mathbf{b}\right)^{1/2}\mathrm{d}\mathbf{W}_{\mathbf{Y}}. \tag{4.10}$$

See Appendix C.2.2 for the proof. The result in (4.10) illustrates that the sampled trajectory indeed meanders around the smoother mean state, which is an analogy to (4.7). Similar to the argument in the forward sampling formula, the uncertainty in the trajectory computed from the backward sampling formula (4.9) or (4.10) is smaller than that of $\mathbf{Y}$ from a free run of the original model (3.2), as a result of the extra information provided by the observations.

The following two corollaries make connections between the point-wise statistics of the sampled trajectories and the filter/smoother posterior estimates. The proofs are shown in Appendices C.2.3 and C.2.4.

Corollary 4.4 (Forward sampling and nonlinear filter)**.** *The conditional Gaussian nonlinear filter estimate* (3.4) *can be recovered by applying a mean-fluctuation decomposition to the forward sampling equation* (4.7)*.*

(*a.*) *The ensemble average of the forward sampling equation* (4.7) *is the time evolution of the posterior mean equation* (3.4a)*.*

(*b.*) *The ensemble average of the quadratic form of the residual equation of* (4.7) *is the time evolution of the covariance equation* (3.4b)*.*

Corollary 4.5 (Backward sampling and nonlinear smoother)**.** *The conditional Gaussian nonlinear smoother estimate* (3.6) *can be recovered by applying a mean-fluctuation decomposition to the backward sampling equation* (4.9)*.*

(*a.*) *The ensemble average of the forward sampling equation* (4.9) *is the time evolution of the posterior mean equation* (3.6a)*.*

(*b.*) *The ensemble average of the quadratic form of the residual equation of* (4.9) *is the time evolution of the covariance equation* (3.6b)*.*

4.3.2 Path-wise error in the posterior mean time series and the sampled trajectories

In many applications, the skill of filtering is quantified by the path-wise error in the filtered posterior mean time series related to the truth. The motivation of adopting such an approach is that for Gaussian posterior distribution the posterior mean estimate at each fixed time instant is the point-wise maximum likelihood estimate. In addition, the path-wise error between the posterior mean time series and the truth is easy to compute in practice. However, computing the path-wise error in the posterior mean time series completely ignores the posterior uncertainty, which is also a time-dependent function and contains significant amount of information of the posterior estimate. Besides, the posterior mean time series is simply a collection of the point-wise maximum likelihood estimates, which is not even dynamically consistent with the underlying model. On the other hand, since the sampled trajectories from either (4.7) or (4.9) take into account the posterior uncertainty and

temporal correlations, assessing the path-wise error in these sampled trajectories can be a more suitable choice in certain circumstances, especially for understanding the dynamical behavior of the unobserved processes conditioned on the observations.

This subsection focuses on comparing the path-wise error in the posterior mean time series and that in the sampled trajectories. The root-mean-squared error (RMSE) and the anomaly pattern correlation (Corr) are the two most widely used path-wise measurements in practice (Taylor, 2001; Houtekamer and Mitchell, 1998; Lermusiaux, 1999; Kalnay, 2003; Hendon et al., 2009; Kim et al., 2012). The path-wise error is often computed for each dimension of the state variable. Denote $Y(t)$ the one-dimensional true time series and $\widehat{Y}(t)$ a sampled trajectory of Y from the filter-based forward sampling conditioned on the observations (4.7). Denote $\overline{Y}(t) = \mathbb{E}(\widehat{Y}(t))$ the posterior mean time series from filtering, where the expectation is taken over all the ensembles at a fixed time instant. Denote $\langle Y \rangle = \frac{1}{T}\int_0^T Y(t)dt$ the time average of $Y(t)$. Let $0 = t_1 \leqslant t_2 \leqslant \ldots \leqslant t_{n-1} \leqslant t_n = T$ be a partition of the time interval with equal distance, and Y_i stands for $Y(t_i)$. The RMSE and the Corr between the posterior mean time series and the true trajectory are defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\overline{Y}_i - Y_i)^2}{n}}, \tag{4.11}$$

$$\text{Corr} = \frac{\sum_{i=1}^{n}(\overline{Y}_i - \langle\overline{Y}\rangle)(Y_i - \langle Y\rangle)}{\sqrt{\sum_{i=1}^{n}(\overline{Y}_i - \langle\overline{Y}\rangle)^2}\sqrt{\sum_{i=1}^{n}(Y_i - \langle Y\rangle)^2}}. \tag{4.12}$$

Similarly, the RMSE and the Corr using the sampled trajectory have the same formulae (4.11)–(4.12) except that $\overline{Y}_i$ is replaced by $\widehat{Y}_i$. Clearly, the uncertainty in the posterior estimate does not appear in the path-wise measurements (4.11)–

(4.12) using the posterior mean time series. The following proposition builds a connection between the RMSE in the posterior mean time series and that in the sampled trajectory from (4.7), where the latter takes into account both the averaged error and the uncertainty.

Proposition 4.1. *The expected squared error in* $\widehat{Y}(t)$ *related to the truth* $Y(t)$ *is given by*

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y(t_i)-\widehat{Y}(t_i)\right)^2\right] \\
&\quad=\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y(t_i)-\overline{Y}(t_i)+\overline{Y}(t_i)-\widehat{Y}(t_i)\right)^2\right] \\
&\quad=\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y(t_i)-\overline{Y}(t_i)\right)^2\right]+\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\overline{Y}(t_i)-\widehat{Y}(t_i)\right)^2\right] \\
&\quad=\frac{1}{n}\sum_{i=1}^{n}\left(Y(t_i)-\mathbb{E}\left[\widehat{Y}(t_i)\right]\right)^2+\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\left[\widehat{Y}(t_i)\right]-\widehat{Y}(t_i)\right)^2\right],
\end{aligned} \tag{4.13}$$

where the fact $\mathbb{E}\left[\widehat{Y}(t_i)\right]=\overline{Y}(t_i)$ *is used in the last equality, which comes from Corollary 4.4.*

This is actually the bias-variance decomposition (Kohavi et al., 1996), where the first term is the bias while the second term is the variance. Notably, the bias part is exactly the square of the traditional RMSE in the posterior mean time series (4.11). The variance part quantifies additional uncertainties in the sampled trajectory beyond the variation of the posterior mean time series. Such additional uncertainties are provided by the posterior covariance at each time instant, which are often ignored in many applications that take into account only the posterior mean time series.

The next goal is to study the difference in the anomaly pattern correlation between using the posterior mean time series and the sampled trajectories.

Proposition 4.2. *The anomaly pattern correlation* (*Corr*) *between the sampled trajectory* $\widehat{Y}(t)$ *and the truth* $Y(t)$ *is given by*

$$\begin{aligned} Corr\,(\widehat{Y}, Y) &= \frac{\sum_{i=1}^{n}(\widehat{Y}_i - \langle\widehat{Y}\rangle)(Y_i - \langle Y\rangle)}{\sqrt{\sum_{i=1}^{n}(\widehat{Y}_i - \langle\widehat{Y}\rangle)^2}\sqrt{\sum_{i=1}^{n}(Y_i - \langle Y\rangle)^2}} \\ &= \frac{\sum_{i=1}^{n}(\overline{Y}_i - \langle\overline{Y}\rangle)(Y_i - \langle Y\rangle) + \sum_{i=1}^{n} Y'_i(Y_i - \langle Y\rangle)}{\sqrt{\sum_{i=1}^{n}(\overline{Y}_i - \langle\overline{Y}\rangle)^2 + \sum_{i=1}^{n}(Y'_i)^2}\sqrt{\sum_{i=1}^{n}(Y_i - \langle Y\rangle)^2}} \\ &= Corr\,(\overline{Y}, Y)\cdot\alpha, \end{aligned} \tag{4.14}$$

where the constant $0 \leqslant \alpha \leqslant 1$

$$\alpha = \frac{\sqrt{\sum_{i=1}^{n}(\overline{Y}_i - \langle\overline{Y}\rangle)^2}}{\sqrt{\sum_{i=1}^{n}(\overline{Y}_i - \langle\overline{Y}\rangle)^2 + \sum_{i=1}^{n}(Y'_i)^2}}, \tag{4.15}$$

and $Y'_i = Y_i - \overline{Y}_i$ *is the residual part of the sampled trajectory related to the posterior mean, the time average of which is zero since the posterior distributions are all Gaussian.*

Note that the time series of Y', according to (4.7), satisfies a linear Gaussian equation ($\mathbf{Y}' = \mathbf{Y} - \boldsymbol{\mu}_{\mathbf{f}}$ in the notation there and Y' is one entry of the vector $\mathbf{Y}'$), which does not depend on Y. Therefore, $\mathrm{Corr}(Y', Y)$ is zero. This fact has been used in deriving the last equality of (4.14).

Proposition 4.2 implies that the anomaly pattern correlation between the sampled trajectory and the truth $\mathrm{Corr}(\widehat{Y}, Y)$ equals to that between the posterior mean

time series and the truth $\mathrm{Corr}(\overline{Y}, Y)$ multiplied by a factor α that depends on the filter uncertainty $(Y_i')^2$. When the filter uncertainty increases, α becomes small. Since $0 \leqslant \alpha \leqslant 1$, $\mathrm{Corr}(\widehat{Y}, Y)$ is always smaller than $\mathrm{Corr}(\overline{Y}, Y)$. Only in the limiting situation that the posterior uncertainty goes to zero, $\mathrm{Corr}(\widehat{Y}, Y)$ converges to $\mathrm{Corr}(\overline{Y}, Y)$.

Despite the wide applications of the RMSE and Corr, these two path-wise measurements are not suitable indicators to quantify the temporal dependence of the points on each time series. Another important quantity in assessing the path-wise behavior is the temporal autocorrelation function (ACF). The ACFs associated with the truth and the posterior mean time series are given by

$$\mathrm{ACF}_Y(s) = \frac{\mathbb{E}[(Y_t - Y_\infty)(Y_{t+s} - Y_\infty)]}{\mathrm{Var}(Y_\infty)}, \tag{4.16}$$

$$\mathrm{ACF}_{\overline{Y}}(s) = \frac{\mathbb{E}[(\overline{Y}_t - \overline{Y}_\infty)(\overline{Y}_{t+s} - \overline{Y}_\infty)]}{\mathrm{Var}(\overline{Y}_\infty)}, \tag{4.17}$$

respectively, where $\mathrm{Var}(Y_\infty)$ and $\mathrm{Var}(\overline{Y}_\infty)$ are the equilibrium variances. Note that computing the ACF requires the underlying dynamics to be stationary. In practice, the sample variance is often used as an approximation in calculating the ACF. For the theoretic discussions here, let us assume the underlying coupled system is linear with Gaussian noise, which guarantees the stationarity condition. In such a situation, the posterior variance also goes extensionally to a constant after a short relaxation period.

Below, let us compare the temporal ACF of the posterior mean time series and that of the sampled trajectory.

Proposition 4.3. *The temporal ACF of the sampled trajectory* $\widehat{Y}$ *is given by*

$$\begin{aligned} ACF_{\widehat{Y}}(s) &= \frac{\mathbb{E}[(\widehat{Y}_t - \widehat{Y}_\infty)(\widehat{Y}_{t+s} - \widehat{Y}_\infty)]}{Var(\widehat{Y}_\infty)} \\ &= \frac{\mathbb{E}[(Y'_t + \overline{Y}_t - \overline{Y}_\infty)(Y'_{t+s} + \overline{Y}_{t+s} - \overline{Y}_\infty)]}{Var(Y'_\infty + \overline{Y}_\infty)} \\ &= \frac{\mathbb{E}[(\overline{Y}_t - \overline{Y}_\infty)(\overline{Y}_{t+s} - \overline{Y}_\infty)]}{Var(Y'_\infty) + Var(\overline{Y}_\infty)} + \frac{\mathbb{E}[Y'_t Y'_{t+s}]}{Var(Y'_\infty) + Var(\overline{Y}_\infty)} \\ &= ACF_{\overline{Y}} \cdot \beta_1 + ACF_{Y'} \cdot \beta_2, \end{aligned} \tag{4.18}$$

where the constants $0 \leqslant \beta_1, \beta_2 \leqslant 1$

$$\beta_1 = \frac{Var(\overline{Y}_\infty)}{Var(Y'_\infty) + Var(\overline{Y}_\infty)}, \qquad \beta_2 = \frac{Var(Y'_\infty)}{Var(Y'_\infty) + Var(\overline{Y}_\infty)} \tag{4.19}$$

and $\beta_1 + \beta_2 = 1$.

It is important to note that the temporal ACF of Y' is not zero because Y' satisfies a dynamical equation (4.7) ($Y' = Y - \mu$) with a finite damping rate. This is very different from the anomaly pattern correlation in (4.18) where Y' has no contribution in (4.18). The integration of the ACF is a crucial quantity and is named as the decorrelation time (Gardiner, 2004), which measures the memory of the system. The decorrelation time of the sampled trajectory can thus be smaller, equal or larger than that of the posterior mean time series.

Similar arguments can be made for the sampled trajectories based on the smoother-based backward sampling method (4.9). In fact, the ACF of the backward sampled trajectory can perfectly reproduce the truth while that associated with the (filter or smoother) posterior mean time series contains a bias (see the numerical experi-

ments in Section 4.5). This also justifies the fact that the posterior mean time series is not dynamical consistent with the truth while the sampled trajectory takes into account the uncertainty and its temporal dependence, which is a more suitable way as a path-wise surrogate of the true signal.

4.3.3 Equilibrium PDFs

In addition to the path-wise errors, quantifying the skill in reproducing the model statistics is another important task. Assume the coupled system (3.2) is stationary and ergodic. It is easy to argue that the posterior mean time series from either filtering or smoothing is unable to recover the equilibrium PDF of the unobserved variable $\mathbf{Y}$ due to the lack of including the posterior uncertainty. On the other hand, the sampled trajectories provide a new efficient way of computing such a PDF, which can also be applied to high-dimensional systems and outweighs the classical numerical approaches of solving the Fokker-Planck equation (Gardiner, 2004).

Theorem 4.6. *Assume the coupled system* (3.2) *is ergodic. Further assume one realization of the observed variable* $\mathbf{X}(t)$ *is available, where the length of this time series goes to infinity. Then the equilibrium PDF of* $\mathbf{Y}$ *from the underlying model equals the PDF by collecting all points in the trajectory from the forward sampling* (4.7)*. It also equals the PDF by collecting all points in the trajectory from the backward sampling* (4.9)*.*

The proofs can follow the procedure provided in (Chen and Majda, 2017, 2018b) and use the ergodic property.

4.4 An Online Nonlinear Smoother Algorithm and the Associated Sampling Procedure

The standard smoother technique (Rauch et al., 1965; Simonoff, 2012; Law et al., 2015; Chen and Majda, 2020) involves a forward pass using a specific filtering method followed by a backward pass to obtain the optimal state estimation. This forward-backward algorithm is quite useful for offline postprocessing the data. However, in many practical situations new observational data are often available in a sequential manner. Recalculating the entire forward-backward procedure from scratch when a new observation is included can be quite cumbersome. Therefore, an online version of the nonlinear smoother that requires only forward-in-time model integrations is highly desirable, which aims at updating the existing estimated states when new observational data arrive sequentially. While particle or ensemble methods can be applied to general nonlinear systems (Evensen and Van Leeuwen, 2000; Olsson et al., 2017), the conditional Gaussian nonlinear framework allows us to use closed analytic formulae to derive such an online nonlinear smoother, which facilitates understanding the associated mathematical and numerical properties.

4.4.1 An online forward-in-time nonlinear smoother algorithm

For the convenience of discussion, consider the time discretization of the general conditional Gaussian nonlinear system (3.2) based on the Euler-Maruyama scheme (Gardiner, 2004) with a small time step $\Delta t = t_{j+1} - t_j$. Denote $\mathbf{X} = \{\mathbf{X}^0, \mathbf{X}^1, \mathbf{X}^2, \ldots\}$ and $\mathbf{Y} = \{\mathbf{Y}^0, \mathbf{Y}^1, \mathbf{Y}^2, \ldots,\}$ the time series of the observed and un-

observed variables at discrete time instants, respectively, where $\mathbf{X}^j = \mathbf{X}(t_j)$ and $\mathbf{Y}^j = \mathbf{Y}(t_j)$.

Lemma 4.7. *The conditional distribution*

$$p(\mathbf{Y}^j|\mathbf{Y}^{j+1}, \mathbf{X}^s, s \leqslant j) \sim \mathcal{N}(\mathbf{m}^j, \mathbf{P}^j) \tag{4.20}$$

is Gaussian, where the conditional mean $\mathbf{m}^j$ *and conditional covariance* $\mathbf{P}^j$ *satisfy the following equations*

$$\mathbf{m}^j = \boldsymbol{\mu}_f^j + \mathbf{C}^j\big(\mathbf{Y}^{j+1} - \mathbf{a}_0^j\Delta t - (\mathbf{I} + \mathbf{a}_1^j\Delta t)\boldsymbol{\mu}_f^j\big), \tag{4.21a}$$

$$\mathbf{P}^j = \mathbf{R}_f^j - \mathbf{C}^j\big(\mathbf{b}^j \circ \mathbf{b}^j\Delta t + (\mathbf{I} + \mathbf{a}_1^j\Delta t)\mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j\Delta t)^*\big)(\mathbf{C}^j)^*, \tag{4.21b}$$

and the auxiliary matrix $\mathbf{C}^j$ *is given by*

$$\mathbf{C}^j = \mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j\Delta t)^*\big(\mathbf{b}^j \circ \mathbf{b}^j\Delta t + (\mathbf{I} + \mathbf{a}_1^j\Delta t)\mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j\Delta t)^*\big)^{-1}. \tag{4.22}$$

Denote $\boldsymbol{\mu}_s^{n,j}$ and $\mathbf{R}_s^{n,j}$ the smoother mean and the smoother covariance at time $t = t_j$ conditioned on a realization of $\mathbf{X}$ up to time $t = t_n$, respectively, where $j \leqslant n$. With Lemma 4.7 in hand, the online forward-in-time nonlinear optimal smoother estimate of the conditional Gaussian system (3.2) is given as follows.

Theorem 4.8 (Online Nonlinear Optimal Smoother)**.** *Assume the observational data* $\mathbf{X}^0, \mathbf{X}^1, \mathbf{X}^2, \ldots$ *is given sequentially. When a new observation* $\mathbf{X}^n$ *is available, it is used to update all the existing nonlinear optimal smoother estimates at time instants* $j \leqslant n-1$ *and provide a new state estimate at* $j = n$*. The nonlinear smoother estimate* $p(\mathbf{Y}^j|\mathbf{X}^s, 0 \leqslant s \leqslant n)$

is conditional Gaussian,

$$p(\mathbf{Y}^j|\mathbf{X}^s, 0 \leqslant s \leqslant n) \sim \mathcal{N}(\boldsymbol{\mu}_s^{n,j}, \mathbf{R}_s^{n,j}). \tag{4.23}$$

The conditional mean $\boldsymbol{\mu}_s^{n,j}$ *and the conditional covariance* $\mathbf{R}_s^{n,j}$ *for* $0 \leqslant j \leqslant n-1$ *satisfy the following recursive formulae,*

$$\begin{aligned}
\boldsymbol{\mu}_s^{n,j} &= \boldsymbol{\mu}_s^{n-1,j} - \mathbf{D}^{n-2,j}\boldsymbol{\mu}_s^{n-1,n-1} + \mathbf{D}^{n-1,j}\boldsymbol{\mu}_s^{n,n} + \mathbf{D}^{n-2,j}\mathbf{m}_1^{n-1}, \\
\mathbf{R}_s^{n,j} &= \mathbf{R}_s^{n-1,j} - \mathbf{D}^{n-2,j}\mathbf{R}_s^{n-1,n-1}(\mathbf{D}^{n-2,j})^* \\
&\qquad + \mathbf{D}^{n-1,j}\mathbf{R}_s^{n,n}(\mathbf{D}^{n-1,j})^* + \mathbf{D}^{n-2,j}\mathbf{P}^{n-1}(\mathbf{D}^{n-2,j})^*,
\end{aligned} \tag{4.24}$$

where each smoother estimate $(\boldsymbol{\mu}_s^{n,j}, \mathbf{R}_s^{n,j})$ *is obtained by updating that from the previous step* $(\boldsymbol{\mu}_s^{n-1,j}, \mathbf{R}_s^{n-1,j})$*. The coefficients of the update in* (4.24) *are calculated forward-in-time in the following way*

$$\begin{aligned}
\mathbf{D}^{n-1,n} &= \mathbf{I}, && \textit{for } n \geqslant 0, \\
\mathbf{D}^{n-1,n-1} &= \mathbf{m}_0^{n-1}, && \textit{for } n \geqslant 1, \\
\mathbf{D}^{n-1,j} &= \mathbf{D}^{n-2,j}\mathbf{m}_0^{n-1}, && \textit{for } n \geqslant 2 \textit{ and } j = 0, \ldots, n-2.
\end{aligned} \tag{4.25}$$

In (4.24)–(4.25)*, the auxiliary terms are given by*

$$\begin{aligned}
\mathbf{m}_0^{n-1} &= \mathbf{C}^{n-1}, \\
\mathbf{m}_1^{n-1} &= \boldsymbol{\mu}_f^{n-1} - \mathbf{C}^{n-1}(\mathbf{a_0}^{n-1}\Delta t + (\mathbf{I} + \mathbf{a_1}^{n-1}\Delta t)\boldsymbol{\mu}_f^{n-1}), \\
\mathbf{P}^{n-1} &= \mathbf{R}_f^{n-1} - \mathbf{C}^{n-1}(\mathbf{b}^{n-1} \circ \mathbf{b}^{n-1}\Delta t + (\mathbf{I} + \mathbf{a_1}^{n-1}\Delta t)\mathbf{R}_f^{n-1}(\mathbf{I} + \mathbf{a_1}^{n-1}\Delta t)^*)(\mathbf{C}^j)^*,
\end{aligned} \tag{4.26}$$

and the auxiliary matrix $\mathbf{C}^{n-1}$ *is defined in* (4.22).

Finally, when $j = n$, *the smoother and the filter estimates are the same, namely* $\boldsymbol{\mu}_{\mathbf{s}}^{n,n} = \boldsymbol{\mu}_{\mathbf{f}}^{n}$ *and* $\mathbf{R}_{\mathbf{s}}^{n,n} = \mathbf{R}_{\mathbf{f}}^{n}$ *at the end point.*

The proof is shown in Appendix C.3.2. Theorem 4.8 provides an optimal online update the state estimate $(\boldsymbol{\mu}^{n,\cdot}, \mathbf{R}^{n,\cdot})$ based on the smoother estimate using the observational information up to the previous time instant $(\boldsymbol{\mu}^{n-1,\cdot}, \mathbf{R}^{n-1,\cdot})$ together with some other contributions that depend only on the filter estimates. When a new observation is available at n, all the existing state estimates $(\boldsymbol{\mu}^{n-1,j}, \mathbf{R}^{n-1,j})$ are updated accordingly. The amount of the update from $(\boldsymbol{\mu}^{n-1,j}, \mathbf{R}^{n-1,j})$ to $(\boldsymbol{\mu}^{n,j}, \mathbf{R}^{n,j})$ can be significant when j is close to n but the influence of the new observations on the existing states decays when $j \ll n$. On the other hand, the smoother estimate simply becomes the filter one at the endpoint, namely $j = n$.

Another form of the online nonlinear smoother, which is equivalent to that in Theorem 4.8 but is more intuitive in understanding the recursive relationship, is as follows.

Theorem 4.9 (Equivalent Formulae of the Online Nonlinear Optimal Smoother)**.** *An equivalent form of the online nonlinear optimal smoother is given as follows*

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{s}}^{n,j} &= \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\boldsymbol{\mu}_{\mathbf{s}}^{n,n-1} - \boldsymbol{\mu}_{\mathbf{s}}^{n-1,n-1}), \\ \mathbf{R}_{\mathbf{s}}^{n,j} &= \mathbf{R}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\mathbf{R}_{\mathbf{s}}^{n,n-1} - \mathbf{R}_{\mathbf{s}}^{n-1,n-1})(\mathbf{D}^{n-2,j})^{*}, \end{aligned} \tag{4.27}$$

where the coefficient $\mathbf{D}^{n-2,j}$ *is updated using* (4.25) *and*

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{s}}^{n,n-1} &= \mathbf{m}_0^{n-1}\boldsymbol{\mu}_{\mathbf{s}}^{n,n} + \mathbf{m}_1^{n-1},\\ \mathbf{R}_{\mathbf{s}}^{n,n-1} &= \mathbf{m}_0^{n-1}\mathbf{R}_{\mathbf{s}}^{n,n}(\mathbf{m}_0^{n-1})^* + \mathbf{P}^{n-1},\end{aligned} \tag{4.28}$$

where $\mathbf{m}_0^{n-1}$, $\mathbf{m}_1^{n-1}$, *and* $\mathbf{P}^{n-1}$ *are the same as those in* (4.26).

Based on Theorem 4.8, an algorithm associated with the online smoother update (4.27) is summarized in Algorithm 3.

Figure 4.1 provides an intuitive way of understanding the forward and backward smoother estimates. For the convenience of discussion, only the smoother mean $\boldsymbol{\mu}_{\mathbf{s}}^{\cdot,\cdot}$ is shown. The same argument applies to the smoother covariance $\mathbf{R}_{\mathbf{s}}^{\cdot,\cdot}$.

Each row, for example the last one in the red dashed box, can be understood as the procedure of the backward smoother estimate given a batch realization $\{\mathbf{X}^0, \ldots, \mathbf{X}^n\}$. It requires a forward pass for computing the filter estimate and a backward pass for calculating the smoother estimate. It is known as forward filtering backward smoothing. On the other hand, each column shows the update of the smoother estimate at a fixed time j in the online forward-in-time smoother algorithm (Theorem 4.9) when new observations arrive sequentially.

Figure 4.1 can also be used to explain the forward-in-time update formulae (4.27). Recall that each row represents the smoother estimates based on currently available observations. Assume the current time instant is $n-1$, and therefore the last element is $\boldsymbol{\mu}_{\mathbf{s}}^{n-1,n-1}$. When a new observation is available at time n, the smoother estimate at $j = n-1$ is updated to $\boldsymbol{\mu}_{\mathbf{s}}^{n,n-1}$. Such an update is indicated by the last

green dashed box in Figure 4.1 and it appears on the right hand side of (4.27). Now let us rewrite the first equation in (4.27) in the following form,

$$\boldsymbol{\mu}_{\mathbf{s}}^{n,j} - \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j} = \mathbf{D}^{n-2,j}(\boldsymbol{\mu}_{\mathbf{s}}^{n,n-1} - \boldsymbol{\mu}_{\mathbf{s}}^{n-1,n-1}),$$

where the left hand side is represented by the first a few green dashed boxes in Figure 4.1. Clearly, the amount of the update from $n-1$ to n observations for the existing state estimation at j is proportional to that at the most recent time instant $n-1$ via the matrix $\mathbf{D}^{n-2,j}$.

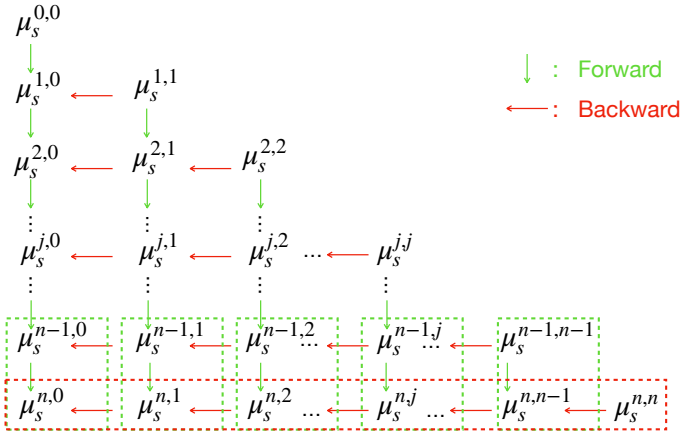

Figure 4.1: Annotation of forward and backward smoother mean. Smoothing parts only.

Algorithm 3: Online Smoother Estimate

Give the first observation $\mathbf{X}^0$; Initialize conditional mean $\boldsymbol{\mu}^0$, and conditional covariance $\mathbf{R}^0$;

for $n = 0, 1, \ldots$ **do**

  Compute the conditional mean at the last time step: $\boldsymbol{\mu}_{\mathbf{s}}^{n,n} = \boldsymbol{\mu}_{\mathbf{f}}^{n}$;

  Compute the conditional covariance at the last time step: $\mathbf{R}_{\mathbf{s}}^{n,n} = \mathbf{R}_{\mathbf{f}}^{n}$;

  Compute $\mathbf{m}_0^{n-1}$, $\mathbf{m}_1^{n-1}$, and $\mathbf{P}^{n-1}$ using (4.26);

  for $j = 0 : n-1$ **do**

    Compute $\mathbf{D}^{n-1,j}$ using (4.25);

    Compute $\boldsymbol{\mu}_{\mathbf{s}}^{n,j}$ and $\mathbf{R}_{\mathbf{s}}^{n,j}$ using (4.24)

Therefore, for the sake of understanding the update in the online smoother, it is natural to study the mathematical properties of the matrix $\mathbf{D}^{n-2,j}$. In light of (4.25), it is easily to see that for $j \leqslant n-2$

$$\mathbf{D}^{n-2,j} = \prod_{i=j}^{n-2} \mathbf{m}_0^i, \text{ and } \mathbf{m}_0^i = \mathbf{C}^i, \tag{4.29}$$

which implies the study of the auxiliary matrix $\mathbf{C}^j$ in (4.22),

$$\mathbf{C}^j = \mathbf{R}_{\mathbf{f}}^j(\mathbf{I} + \mathbf{a}_1^j\Delta t)^*\left(\mathbf{b}^j \circ \mathbf{b}^j\Delta t + (\mathbf{I} + \mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I} + \mathbf{a}_1^j\Delta)t^*\right)^{-1}.$$

The following proposition shows that the impact of the observation obtained from time n on the smoother estimate at a previous time instant j will decay when n becomes larger than j.

Proposition 4.4. *If* $\mathbf{a}_1^j \mathbf{R}_\mathbf{f}^j + \mathbf{b}^j \circ \mathbf{b}^j$ *is a positive definite matrix, then the spectral radius of* $\mathbf{C}^j$ *in* (4.22) *is less than* 1.

The proof is shown in Appendix C.3.4. Note that the filter covariance matrix $\mathbf{R}_\mathbf{f}^j$ combines the uncertainties from both the observed and unobserved variables. If the uncertainty in the observed processes is much smaller than that in the unobserved processes, then the eigenvalues of $\mathbf{b}^j \circ \mathbf{b}^j$ will be much larger than those of $\mathbf{R}_\mathbf{f}^j$. As a consequence, the spectral of $\mathbf{C}^j$ is much less than 1 according to (4.22). In fact, in such a situation, the informative new observation is able to provide a significant amount of the update for the estimated states within a nearby time interval but its impact on the past time instants that are far away from the current one will decay exponentially fast. Numerical examples of a two dimensional linear Gaussian model with exact solutions and a nonlinear dyad model with intermittent instabilities are provided in Appendix C.4. For the latter one, the value of $\mathbf{C}^j$ will be smaller when extreme events happen in the observed time series, since it gives a large signal-to-noise ratio and reduces the uncertainty in the observations.

4.4.2 A fixed-lag online smoother approximation

As the dimension of the system and the length of the observational period increase, the storage of the updates in the online smoothing becomes tremendous. Since the impact of each new observation on the past time instants decays exponentially, it is practically useful to derive a fixed-lag online smoother (Kitagawa and Sato, 2001; Cappé et al., 2006; Olsson et al., 2008), in which the impact region of each new observation is assumed to be finite and is predetermined.

Assume the predetermined size of the impact region (i.e., the lag) is L. The fixed-lag online smoother estimate is given by following formulae.

$$\boldsymbol{\mu}_{\mathbf{s}}^{n,j} = \begin{cases} \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j}, & j \leqslant n-1-L, \\ \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\boldsymbol{\mu}_{\mathbf{s}}^{n,n-1} - \boldsymbol{\mu}_{\mathbf{s}}^{n-1,n-1}), & n-L \leqslant j \leqslant n-1, \end{cases} \tag{4.30a}$$

$$\mathbf{R}_{\mathbf{s}}^{n,j} = \begin{cases} \mathbf{R}_{\mathbf{s}}^{n-1,j}, & j \leqslant n-1-L, \\ \mathbf{R}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\mathbf{R}_{\mathbf{s}}^{n,n-1} - \mathbf{R}_{\mathbf{s}}^{n-1,n-1})(\mathbf{D}^{n-2,j})^*, & n-L \leqslant j \leqslant n-1. \end{cases} \tag{4.30b}$$

The explicit form of $\mathbf{C}^j$ can be used as a reference of choosing the lag L.

4.4.3 A forward-in-time smoother-based sampling algorithm

Theorem 4.8 provides an optimal online way of state estimation at each time instant t_j adjusted from the existing values when new observations are available. Corresponding to this online smoother, an online sampling algorithm can be developed. Note that this online sampling algorithm makes use of the smoother-based state estimation, which is different from the forward sampling based on the filter estimates in Theorem 4.1.

Theorem 4.10 (Nonlinear Optimal Smoother-Based Forward-in-Time Sampling Algorithm)**.** *Assume the observational data* $\mathbf{X}^0, \mathbf{X}^1, \mathbf{X}^2, \ldots$ *is given sequentially. When a new observation* $\mathbf{X}^n$ *is available, it is used to update all the existing the mean and covariance* $(\mathbf{m}_{\mathbf{s}}^{n,j}, \mathbf{P}_{\mathbf{s}}^{n,j})$ *of the conditional distribution* $p(\mathbf{Y}^j, \mathbf{Y}^{j+1} | \mathbf{X}^s, s \leqslant n)$ *at time instants* $j \leqslant n-1$

and provide a new estimate at $j = n$. *Such an estimate leads to the following sampling of the hidden trajectories*

$$\mathbf{Y}^{j+1} \sim \mathcal{N}(\mathbf{m}_s^{n,j}, \mathbf{P}_s^{n,j}), \qquad j \leqslant n-1, \tag{4.31}$$

where

$$\begin{aligned} \mathbf{m}_s^{n,j} &= \boldsymbol{\mu}_s^{n,j+1} + \mathbf{R}_s^{n,j+1}(\mathbf{C}^j)^*(\mathbf{R}_s^{n,j})^{-1}(\mathbf{Y}^j - \boldsymbol{\mu}_s^{n,j}), \\ \mathbf{P}_s^{n,j} &= \mathbf{R}_s^{n,j+1} - \mathbf{R}_s^{n,j+1}(\mathbf{C}^j)^*(\mathbf{R}_s^{n,j})^{-1}(\mathbf{C}^j)\mathbf{R}_s^{n,j+1}. \end{aligned} \tag{4.32}$$

In (4.32)*, the smoother mean and covariance* $(\boldsymbol{\mu}_s^{n,j}, \mathbf{R}_s^{n,j+1})$ *is updated using* (4.24) *from* $n-1$.

The proof is shown in Appendix C.3.5. Clearly, given all the smoother estimates, the sampling step is forward-in-time since $\mathbf{Y}^{j+1}$ depends only explicitly on $\mathbf{Y}^j$. This is fundamentally different from the backward sampling formulae in (4.9).

Comparing with the filter-based forward sampling in (4.7), the smoother-based forward-in-time sampling algorithm developed here provides more accurate sampled trajectories. Yet, the computational cost of the smoother-based forward-in-time sampling algorithm is more expensive since the conditional distribution (4.32) needs to be re-calculated with the sequential observational data. Therefore, if certain error is allowed, then the cheap filter-based forward sampling in (4.7) remains as a good approach to generate sampled trajectories of the hidden variables conditioned on the past observations.

4.5 State Estimation, Sampling Intermittent Time Series, and Detecting Hidden Extreme Events

In this section, numerical examples are used for studying the state estimation, intermittency and extreme events. Throughout this section, the forward sampling stands for the filter-based forward sampling method (4.7) while the backward sampling means the smoother-based backward sampling approach (4.9) (or numerically its online version (4.27)). Note that the forward sampling here should not be confused with the online smoother-based sampling algorithm developed in Section 4.4.

4.5.1 Recovering the dynamical and statistical features of hidden variables

In this subsection, the following nonlinear dyad model is used as a test model to numerically illustrate the difference in the path-wise and statistical behavior between using the filter posterior mean time series, the smoother posterior mean time series, the sampled trajectories from the forward sampling and the sampled trajectories from the backward sampling.

The nonlinear dyad model reads,

$$du = \big((-d_u + cv)u + f_u\big)dt + \sigma_u dW_u, \tag{4.33a}$$

$$dv = (-d_v v - cu^2 + f_v)dt + \sigma_v dW_v, \tag{4.33b}$$

where u and v correspond to the observed variable $\mathbf{X}$ and the unobserved one $\mathbf{Y}$ in

the general conditional Gaussian nonlinear modeling framework (3.2), respectively. The dyad model (4.33) is a simple nonlinear model but it contains many desirable features that are observed in more complex turbulent dynamical systems. First, the nonlinear dyad model has energy-conserving nonlinear interactions via the cvu and $-cu^2$ terms, and therefore the model belongs to the physics-constrained nonlinear modeling framework (Majda and Harlim, 2012b; Harlim et al., 2014). Second, the hidden variable v acts as a stochastic damping of the observed variable u. In other words, v plays the role of triggering the extreme events in u when $-d_u + cv$ becomes positive. This allows the dyad model to be a simple test model for understanding and predicting extreme events. Third, it is the stochastic noise W_v that drives v above the threshold of $v = d_u/c$, which then triggers the intermittent extreme events in u. The recovery of the hidden variable in such a "noise-driven" process is thus a more challenging task than the models which are dominated by the deterministic effects.

In this subsection, the focus is on studying the path-wise and statistical behavior of the model and therefore a long trajectory with 500 units is used to eliminate the error in reconstructing the statistics due to the insufficient number of sample points. A short trajectory will be used in Section 4.5.2 for comparing the filter and the smoother estimates in capturing the transient behavior of extreme events.

The following parameters are used in the tests here:

$$\begin{gathered} f_u = 1, \qquad f_v = 0, \qquad d_v = 0.8, \qquad d_u = 0.8, \\ c = 1.2, \qquad \sigma_v = 2, \qquad \sigma_u = 0.5. \end{gathered} \tag{4.34}$$

Figure 4.2 shows a realization of the model with 500 time units. The true signal of the observed variable u has quite a few intermittent extreme events and the associated PDF is highly non-Gaussian with an one-sided fat tail. The true signal of v is nearly Gaussian and it is not observed.

Panel (a) of Figure 4.3 compares the truth and the recovered PDFs of v based on different approaches. Here, the blue curve represents the true PDF. The green and the red curves are the PDFs associated with the posterior mean time series resulting from the nonlinear optimal smoother (3.6a) and the nonlinear optimal filter (3.4a), respectively. The black and the cyan curves are the sampled trajectories from the smoother-based backward sampling (4.9) and the filter-based forward sampling (4.7). Unlike the posterior mean time series, which is deterministic once the observational time series is given, the randomness still exists in the sampled trajectories. To take into account the uncertainty from such randomness in calculating the statistics due to the use of a finite length of the time series, the PDFs associated with 50 different sampled trajectories are shown for each case. The results indicate that the difference between the PDFs resulting from different sampled trajectories with 500 time units is small. Similar to Panel (a), Panel (b) shows the comparison of the ACFs.

From Panel (a), it is clear that the PDFs associated with the posterior mean time series regardless of filtering or smoothing are quite different from the truth. This is not a surprising result because the variability of v represented by the posterior variance is not included. In contrast, the PDFs associated with the sampled trajectories resulting from both the forward and backward sampling strategies perfectly recover

the truth. This is a numerical evidence that supports the result in Theorem 4.6. In Panel (b), it is shown that the ACFs associated with the posterior mean time series also contain biases, which validate the conclusions in Proposition 4.3 that ignoring the temporal correlated uncertainty leads to errors in recovering the dynamical features of the underlying system. These comparisons indicate that the posterior mean time series is not a good surrogate to reproduce the statistical and dynamical features of the truth. On the other hand, different from a perfect match of the PDF, the trajectories resulting from the forward sampling now leads to an error in reproducing the ACF. Nevertheless, the trajectories from the backward sampling remains a perfect recovery of the truth. The results here are intuitive because the backward sampled trajectories contains extra information in "future".

Panel (c) shows the error PDF, which is computed by first taking the difference between the true signal and a certain recovered trajectory and then forming a PDF by collecting all the points in the resulting time series. The recovered trajectory is either the posterior mean time series or the sampled ones. The overall difference between the two time series is reflected by the variance of the error PDF, which is equivalent to the RMSE but is a more intuitive representation. The results in Panel (c) indicate that the averaged point-wise error in the smoother mean time series is the smallest, followed by the filter mean time series. The sampled trajectories have larger point-wise errors resulting from the uncertainty in the posterior variance, which has been quantified by the variance part in Proposition 4.1. Panel (d) demonstrates the comparison of the anomaly pattern correlation, which has the same qualitative conclusion as the error PDFs, and the numerical results here are consistent with

the conclusions in Proposition 4.2. Note that the anomaly pattern correlation conditioned on the phases with extreme events of u are lower than the overall pattern correlation, where the phases with extreme events of u are defined by those time instants when v is greater than d_u/c such that the damping becomes anti-damping. The deterioration in the skill of recovering the hidden states associated with the observed extreme events is due to the fact that the extreme events are largely triggered by random noise in the unobserved process. Nevertheless, the smoother has a much higher skill than the filter to recover these hidden states. In fact, the extreme events in u happen as a consequence of the signal v being across the threshold d_u/c. In other words, the occurrence of the large bursts in u acts as a delayed response when v becomes greater than d_u/c. Since filtering only uses the information in the past, the state estimation from filtering is expected to miss the onset phase of v being across the threshold d_u/c. A case study will be shown in Section 4.5.2 for a comparison of the state estimation between using filtering and the online smoothing.

To summarize the findings in Figure 4.3, the filter or smoother posterior mean time series contains a smaller averaged point-wise error (Panels (c) and (d)) while the sampled trajectories share more similar dynamical and statistical features as the true dynamics (Panels (a) and (b)). In particular, despite the optimality in the point-wise sense, using the posterior mean time series as the surrogate for the hidden trajectory is not a suitable choice since it misses both the basic dynamical feature (reflected by the temporal ACF) and fails to reproduce the equilibrium statistics of the model (e.g., the PDF). The sampled trajectories from the smoother-

based backward sampling method can reproduce both the ACF and PDF perfectly while those from the filter-based forward sampling method succeed in recovering the equilibrium PDF but may contain errors in reproducing the ACF.

As a final remark of this subsection, a naive way to obtain a set of sampled trajectories of the u that shares certain features with the specific observed one is to plug the sampled trajectories of the hidden variable v back into (4.33a). The PDFs of these sampled trajectories of u are nearly the same as the truth (not shown here). However, the ACFs may contain errors. This is because plugging a fixed trajectory of v back into the u equation to form a time series of u is equivalent to assuming that the uncertainty of v is zero, which is not the case in the original coupled system. On the other hand, the time series of u resulting from plugging the posterior time series into (4.33a) even contains errors in the PDF. In Section 4.5.3, a simple but systematic method will be developed to sample trajectories of the observed variable for the conditional Gaussian models that retain the conditional Gaussianity in the reverse way, as were discussed in Section 4.2.1.3. The sampled trajectories succeed in recovering both the PDFs and ACFs of the true signals from the original model.

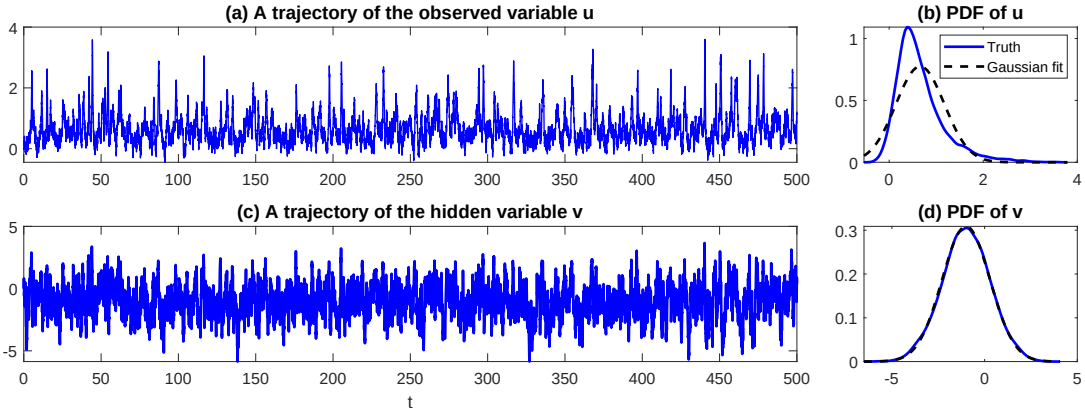

Figure 4.2: A realization of the dyad model (4.33) (Panels (a) and (c)) and the equilibrium PDFs (Panels (b) and (d)), where the parameters are given by (4.34). In Panels (b) and (d), the black dashed curves are the Gaussian fits of the true PDFs.

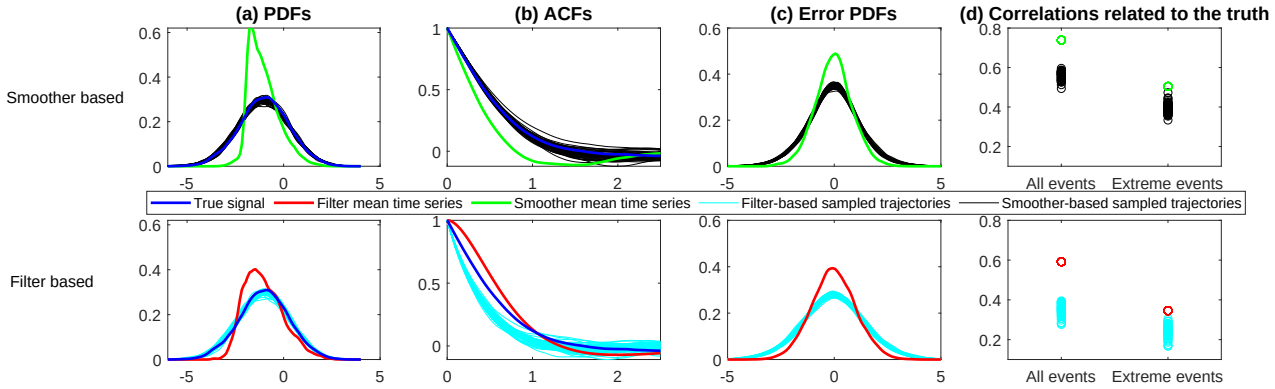

Figure 4.3: Dyad model (4.33) with parameters (4.34). Panel (a): the truth and the recovered PDFs of ν based on different approaches. Here, the blue curve represents the true PDF. The green and the red curves are the PDFs associated with the posterior mean time series resulting from the nonlinear optimal smoother (3.6a) and the nonlinear optimal filter (3.4a), respectively. The black and the cyan curves are 50 sampled trajectories from the smoother-based backward sampling (4.9) and the filter-based forward sampling (4.7). The truth is given in Panel (c) of Figure 4.2 and all the recovered trajectories have 500 time units as the truth. Panel (b): the comparison of the ACFs. Panel (c): the error PDF, which is computed by first taking the difference between the true and a certain recovered trajectory and then forming a PDF by collecting all the points in the resulting time series. Panel (d): the anomaly pattern correlation related to the truth. The label "extreme events" here means the phases with extreme events in u that are defined by the corresponding ν being greater than the threshold value d_u/c such that ν becomes anti-damping in the observed u process.

4.5.2 Comparing the posterior state estimation using the nonlinear smoother and the nonlinear filter

It is important to understand quantitatively the difference between the filter and smoother estimates and the time evolution of the estimated states. In this subsection, the nonlinear dyad model (4.33) is used again to demonstrate the difference between the time evolutions of the filter and smoother estimates, especially at the phases with the observed extreme events. Recall that when a new observation is available at time t, filtering only provides the state estimation at the current time instant t while smoothing updates all the estimated states in the past for $s \leqslant t$. The online smoothing algorithm developed in Section 4.4 is utilized here for studying the online update of the historical states using the nonlinear smoothing technique.

Figure 4.4 shows a comparison between the state estimations using the online smoothing (4.27) and filtering (3.4) techniques. The parameters in (4.34) are used to generate the true signal, which are shown in Panel (a). The black and the blue curves are the true signals of u and v, respectively. Panels (b)–(i) demonstrate the filter and smoother estimates as time evolves from $t = 31.25$ to $t = 33$. This period is indicated by the red box in Panel (a). The online smoother and filter posterior mean states are shown in the green and the red curves, respectively, and the associated uncertainties (represented by one posterior standard derivation) are given by the green and the red shading areas. In Panels (b)–(d), the true signal of v remains negative and the signal u of lies in a quiescent phase. In such a situation, both the filter and smoother mean estimates are non-informative, tracking the equilibrium mean state of v in the original model, with large uncertainties that

cover the true signal. At $t = 32$ (Panel (e)), the true signal of v exceeds the threshold value and starts to trigger the extreme events of u. Yet, the response of u is not instantaneous such that no obvious increase of the amplitude is observed in the signal of u. In other words, the triggering mechanism of the observed extreme events is not observable. Therefore, both the filter and the smoother estimates fail to recover the true signal of v. The estimated states from filtering and smoothing become significantly different when $t = 32.25$ (Panel (f)). At this time instant, the amplitude of u has been greatly increased and thus both the filter and smoother are able to recover the hidden state quite accurately due to a large signal-to-noise ratio in the observed signal u. However, since the filter only estimate the state at the current time, the underestimated amplitude of v in all the previous time instants cannot be corrected. On the other hand, the online smoother at $t = 32.25$ not only accurately detects the instantaneous state of v but makes use of such information to remedy the estimated states in the previous time instants. In fact, the recovered signal of v in the entire period from $t = 31.75$ to $t = 32.25$ has been revised almost perfectly, which indicates the advantage of the smoother over the filter. In Panels (g)–(i), the true signal of v goes back to the negative phase and the signal of u stops increasing. Nevertheless, due to the delayed response in u to the change of v, the large signal-to-noise ratio remains in the time series of u. As a result, both the filter and the smoother posterior mean states capture the truth accurately with a tiny uncertainty.

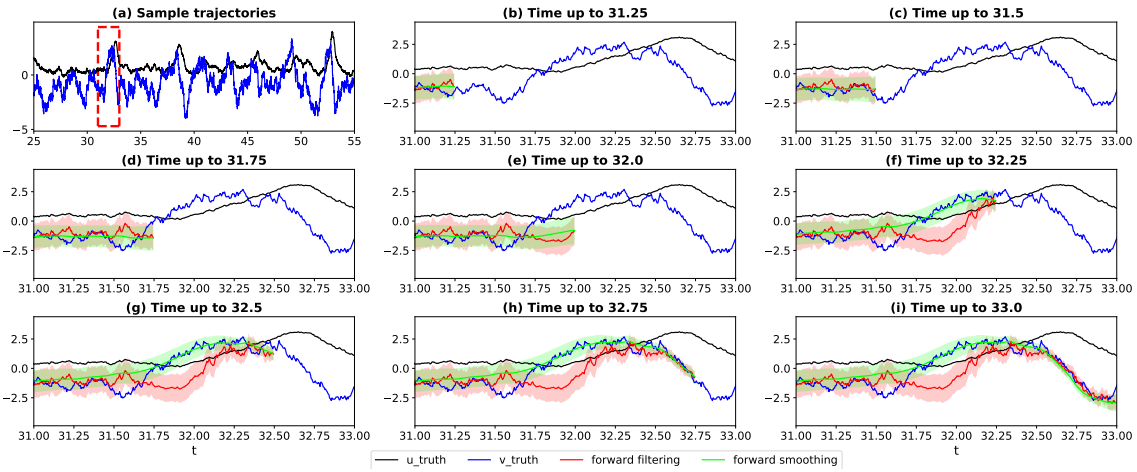

Figure 4.4: The physics-constrained dyad model (4.33) with parameters given by (4.34). Panel (a) shows the true signal of the observed variable u (black) and the hidden variable v (blue). Panels (b)–(i) show the smoother and filter estimates up to the current observed time instant, which moves forward in time. The online smoother and filter mean states are shown in the green and red curves, respectively, and the associated uncertainties (represented by one standard derivation) are given by the green and red shading areas.

4.5.3 Sampling the observed variables

In addition to sampling the trajectories of the unobserved variables $\mathbf{Y}$ given a specific observed trajectory $\mathbf{X}^{obs}(t)$, it is also practically useful to make use of these sampled $\mathbf{Y}$ trajectories to generate different trajectories of the observed variables $\mathbf{X}$. These sampled $\mathbf{X}$ trajectories are different from those associated with a model free run. Instead, they are based on the specific observed trajectory $\mathbf{X}^{obs}(t)$ and take into account the model uncertainty. Meanwhile, they should share the same dynamical and statistical features as the underlying model.

In Section 4.5.1, it has been remarked that plugging the sampled hidden trajectory of $\mathbf{Y}$ back into the equation of the observed variables $\mathbf{X}$ to reach a sampled time series of the latter violates the dynamical structure of the original coupled system. This is because applying this plugging-in method is equivalent to assuming that the uncertainty in the hidden process is zero. In this subsection, a dynamically consistent method is developed for the conditional Gaussian models that retain the conditional Gaussianity in the reverse way (see Section 4.2.1.3). It is based on the following two-step sampling procedure:

Step 1. Given a trajectory of $\mathbf{X}^{obs}(t)$, run either the forward or backward sampling algorithm (4.7) or (4.9) to obtain a sampled path of $\mathbf{Y}$, which is denoted by $\mathbf{Y}^{smp}(t)$.

Step 2. Since $p(\mathbf{X}(t)|\mathbf{Y}(s))$ is also a conditional Gaussian model, the forward or backward sampling algorithm can be applied to sample a trajectory of $\mathbf{X}$ conditioned on the $\mathbf{Y}^{smp}(t)$ from Step 1. The resulting sampled trajectory of

$\mathbf{X}$ is denoted by $\mathbf{X}^{smp}(t)$.

In the procedure of computing $\mathbf{X}^{smp}(t)$, the uncertainties in both $\mathbf{X}$ and $\mathbf{Y}$ have been taken into account. The trajectories $\mathbf{X}^{smp}(t)$ and $\mathbf{X}^{obs}(t)$ also share many common dynamical and statistical features.

As a test model, consider the noisy Lorenz 63 model (Lorenz, 1963),

$$dx = \sigma(y - x)dt + \sigma_x dW_x, \tag{4.35a}$$

$$dy = \big(x(\rho - z) - y\big)dt + \sigma_y dW_y, \tag{4.35b}$$

$$dz = (xy - \beta z)dt + \sigma_z dW_z. \tag{4.35c}$$

It is a simplified mathematical model for atmospheric convection and is widely used as a testbed for validating numerical methods for chaotic systems. The noisy Lorenz 63 model (4.35) is a conditional Gaussian system in two ways, that is, both the conditional distributions $p(x(t)|y(s), z(s), s \leqslant t)$ and $p(y(t), z(t)|x(s), s \leqslant t)$ for filtering are conditional Gaussian (and the same applies for smoothing). The following parameters are adopted in the test here,

$$\sigma = 5, \qquad \rho = 38, \qquad \beta = 2/3, \qquad \sigma_x = \sigma_y = \sigma_z = 15. \tag{4.36}$$

These parameters are different from the classical choices ($\sigma = 10$, $\rho = 28$ and $\beta = 8/3$). The modification of the parameters allows the system to be more turbulent, which together with the moderately large noise coefficients leads to a noticeable difference in the posterior mean time series and the sampled ones. Nevertheless, as

shown in Figure 4.5, the phase plots still exhibit a butterfly profile and the trajectories have irregular oscillation patterns. A model simulation with 500 time units is used for the study here, which is sufficiently long to exclude obvious undersampling bias.

Below, assume one trajectory of x (with 500 time units) is available, which is shown in Panel (a) of Figure 4.5. Following the procedure described at the beginning of this subsection, the forward or backward sampling method can be applied to sample trajectories of y and z given the specific observed time series of x. Then, conditioned on the sampled trajectories of y and z, the forward or backward sampling algorithm is used again for sampling the trajectory of x. The PDFs and ACFs associated with the sampled trajectories of x based on 50 repeated experiments are shown in the first two rows in Figure 4.6. The sampled trajectories of x from the smoother-based backward sampling method lead to a perfect recovery of the statistics while those from the filter-based backward sampling approach are also nearly the same as the truth. In fact, unlike the nonlinear dyad model, in which the intermittent events are triggered by random noise in the hidden process, the noisy Lorenz 63 model still have crucial dynamical structures dominated by irregular oscillations. Thus, the difference in the filter-based sampling and the smoother-based sampling methods is not that significant. As a comparison, the third and the fourth rows show the statistics associated with the trajectories of x which are formed by plugging the sampled trajectories of y and z into the equation of x (4.35a) (hereafter "plugging-in" method). It is clear that non-negligible errors exist in the ACFs.

In addition to the skill of recovering the dynamical and statistical features of the model, the point-wise error represented by the error PDFs in the plugging-in approach of sampling x is significantly larger than the two-step sampling procedure developed here. See Panel (c) in Figure 4.6. Likewise, the anomaly pattern correlation in Panel (d) also indicates the advantage of the two-step sampling procedure developed here, especially for the extreme events. The extreme events are those with amplitude of x being greater (smaller) than 15 (-15), which is two standard deviations from the mean value. In fact, the direct plugging-in approach can be regarded as using an imperfect model with zero uncertainty of y and z in the second step. It is expected that the error becomes large in the presence of model error.

Each panel in Figure 4.7 shows one sampled trajectory (black) and the probability of the area covered by the sampled trajectories. The sampled trajectories in Panels (a) and (b) are based on the two-step sampling procedure using the filter-based forward sampling and the smoother-based backward sampling, respectively. Panels (c) and (d) show the sampled trajectories using the plugging-in approach for x, based on the sampled trajectories of y and z from the forward and backward sampling. The results in this figure confirm the statistical and dynamical behavior shown in Figure 4.6. The sampled paths using the two-step procedure resemble the truth while those using the plugging-in approach do not always capture the crucial dynamical features of the true signal. The latter also have a much larger uncertainty.

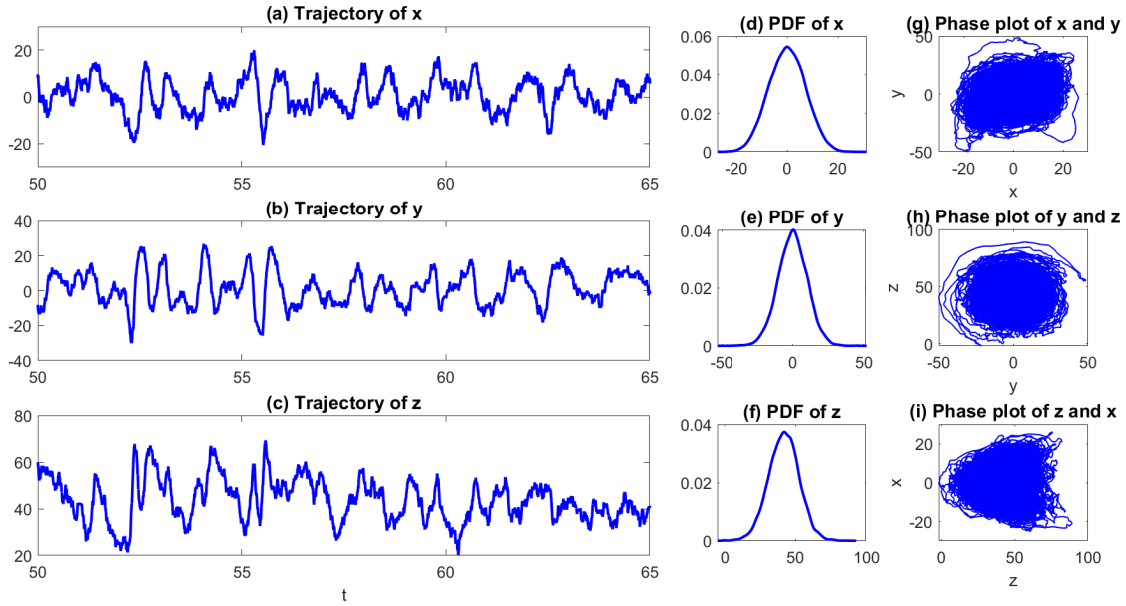

Figure 4.5: Noisy Lorenz 63 model (4.35) with parameters (4.36). Panels (a)–(c): trajectories. Panels (d)–(f): marginal PDFs. Panels (g)–(i): two dimensional phase plots. The statistics are based on a model simulation with 500 time units despite that the trajectories shown here, for the illustration purpose, have only 15 time units long.

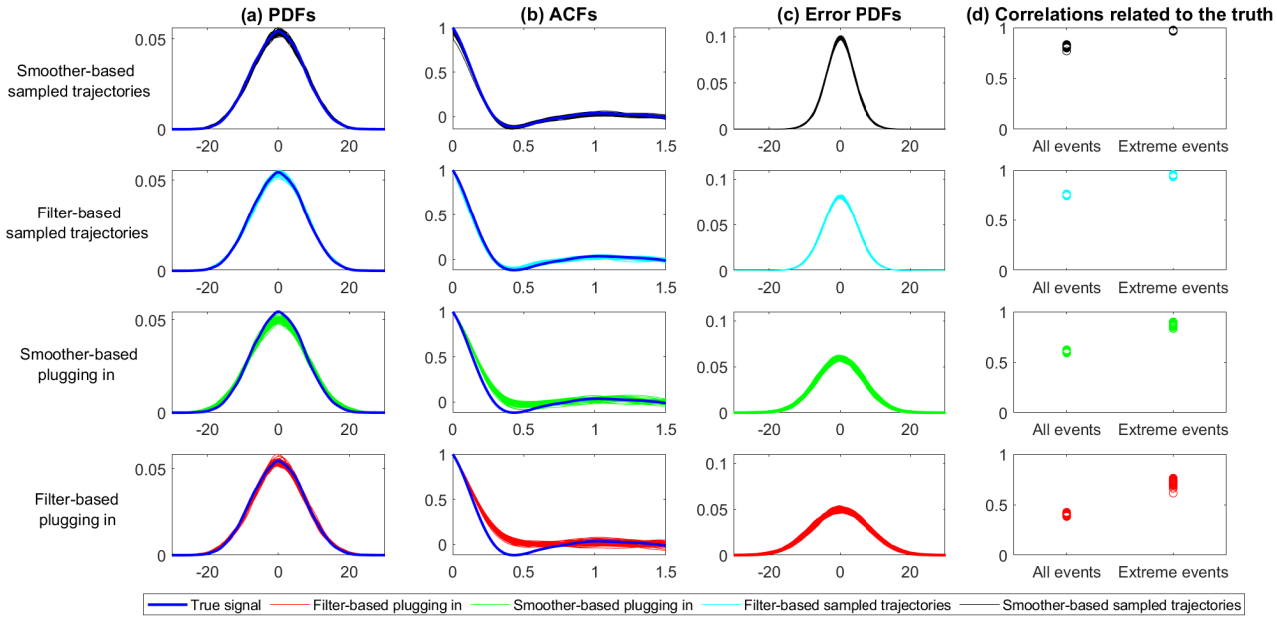

Figure 4.6: Sampling trajectories of the observed variable x in the noisy Lorenz 63 model (4.35) with parameters (4.36) given the specific observed trajectory shown in Panel (a) of Figure 4.5. The first two rows are the statistics associated with the sampled trajectories of x using the two-step procedure developed at the beginning of this subsection. the third and the fourth rows show the statistics associated with the trajectories of x which are formed by plugging the sampled trajectories of y and z in the first step into the equation of x (4.35a). All the statistics are computed based on time series with length of 500 time units. Panel (a): PDFs. Panel (b): ACFs. Panel (c): error PDFs. Panel (d): anomalous pattern correlation.

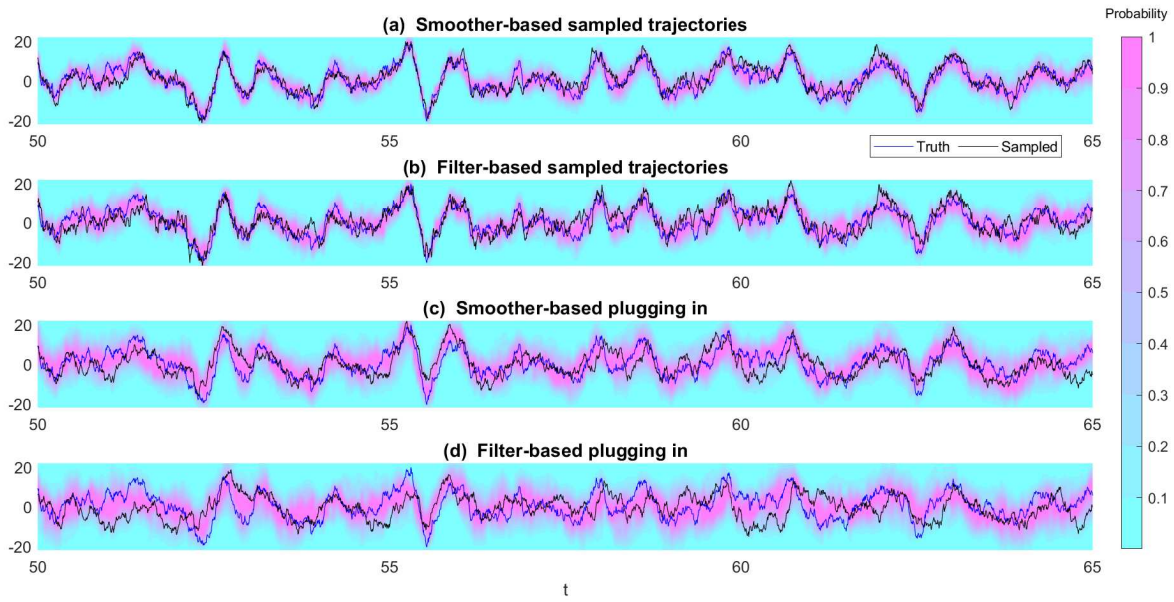

Figure 4.7: The true signal (blue) and a sampled trajectory (black) of the noisy Lorenz 63 model (4.35) with parameters (4.36). The shading shows the probability of the area that all the sampled trajectories cover. The four panels show the situations using different methods to sample the trajectories of x. Panels (a)–(b): using the two-step procedure developed at the beginning of this subsection, using filter-based forward and smoother-based backward sampling, respectively. Panels (a)–(b): plugging the sampled trajectories of y and z in the first step into the equation of x (4.35a).

4.5.4 Multiple observations in recovering stochastic parameterizations

In all the previous examples, the observed and the hidden variables have a mutual dependence. Therefore, one realization of the unobserved variable has only one corresponding realization of the observed variable. On the other hand, for the models belonging to the framework shown in Section 4.2.1.2, where the governing equation of $\mathbf{Y}$ has no dependence on $\mathbf{X}$, multiple observations can be used to recover a single trajectory of the hidden variable.

As a simple illustration, consider the following so-called SPEKF model (stochastic parameterized extended Kalman filter model) (Gershgorin et al., 2010b,a),

$$du = (-\gamma u + f_u + w)dt + \sigma_u dW_u, \tag{4.37a}$$

$$d\gamma = (-d_\gamma \gamma + f_\gamma)dt + \sigma_\gamma dW_\gamma, \tag{4.37b}$$

where u is the observed variable and γ is the stochastic parameterization serving as the stochastic damping of u. The nonlinear SPEKF system was first introduced for filtering multiscale turbulent signals with hidden instabilities and has been used for filtering and predicting complex nonlinear systems (Majda and Harlim, 2012a; Majda, 2016). Other important applications of using the SPEKF model to filter complex spatial-extended systems include stochastic dynamical superresolution (Branicki and Majda, 2013) and effective filters for Navier-Stokes equation (Branicki et al., 2018). It has been shown that the SPEKF model has much higher skill than classical Kalman filters using the so-called mean stochastic model (MSM) (Majda and

Harlim, 2012a). In (4.37a), a control term w is added, which by default is zero. The other parameters in the test here are given as follows,

$$f_u = 0, \qquad \sigma_u = 0.3, \qquad d_\gamma = 1, \qquad f_\gamma = 0.8, \qquad \sigma_\gamma = 1. \tag{4.38}$$

These parameters allow γ to switch between positive and negative values. Once γ becomes negative, the anti-damping leads to intermittent extreme events in u.

A single realization of γ is generated from the SPEKF model, which is illustrated by the blue curve in the second and third row in Figure 4.8. As shown in Panel (a), if a single observation of u is used to recover γ, then the filter estimate (the second row) has a delayed response while the smoother estimate (the third row) can capture the timing of the negative phase of γ, which is the same as the nonlinear dyad model. Both the filter and smoother estimates have a relatively large uncertainty, especially at the quiescent phases. In Panel (b), $L = 10$ observational trajectories are used. Such multiple observations improve the state estimation of γ in the sense that the uncertainty of the posterior estimate is greatly reduced. Notably, the issue of the delayed response in the filter estimate is significantly mitigated. This is because multiple observations play the role of averaging out the noise. Therefore, the onset of the intermittent phase is much easier to be detected. In Panel (c), again $L = 1$ observation is used. But different from the setup in Panel (a), a nonzero control term $w = 2$ is used in the test in Panel (c). This control term plays the same role as the multiple observations that reduce the uncertainty in recovering the hidden trajectory and improves the accuracy in detecting the onset phase of the extreme events. In fact, this control term leads to a larger amplitude of the observational time

series of u, which increases the signal-to-noise ratio and provides the improvement.

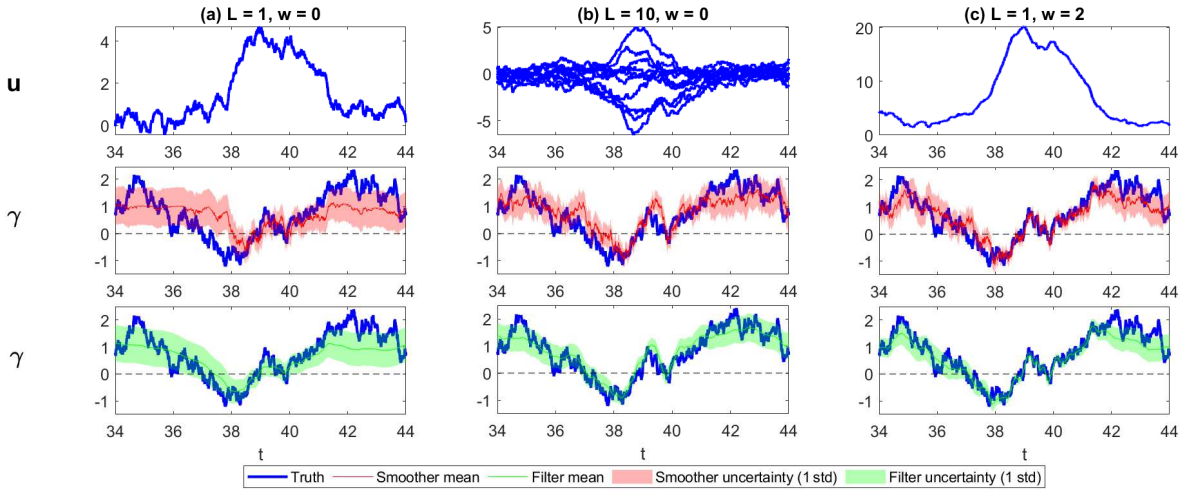

Figure 4.8: The SPEKF model (4.37) with parameters in (4.38). The true signal of the unobserved variable γ is the same in all the test. Panel (a): $L = 1$ observed trajectory of u. Panel (b): multiple $L = 10$ observed trajectories of u. Panel (c): $L = 1$ trajectory of u but a nonzero control term $w = 2$ is imposed. The first row shows the observed time series of u. The second and the third rows show the true hidden trajectory γ (blue), the filter/smoother posterior mean (thin curve) and the filter/smoother posterior uncertainty (shading) in the form of one standard deviation of the posterior distribution. The second row shows the results for filtering while the third row shows those for smoothing.

4.5.5 Lagrangian data assimilation

Lagrangian data assimilation has wide applications in practice. The state estimation from the nonlinear smoothing technique developed here can be used as a comparison to explore the insufficiency in Lagrangian data assimilation in the presence of multiple observations.

Consider a simple random incompressible flow given by the stream function (Vallis, 2017)

$$\psi = \hat{u}_{1,0} e^{ix} + \hat{u}_{0,1} e^{iy} + \text{c.c.}, \tag{4.39}$$

where ψ is represented by 4 Fourier modes with Fourier coefficient $\hat{u}_{\mathbf{k}}$ and 'c.c.' denotes the complex conjugate. The stream function and the velocity field are related via

$$\mathbf{v} = \nabla^{\perp}\psi = \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x}\right)^{T}. \tag{4.40}$$

For both wavenumbers $\mathbf{k} = (1, 0)$ and $\mathbf{k} = (0, 1)$, the underlying dynamics of $\hat{u}_{\mathbf{k}}$ is given by

$$d\hat{u}_{\mathbf{k}} = (-d_{\mathbf{k}}\hat{u}_{\mathbf{k}} + f_{\mathbf{k}})dt + \sqrt{2}\sigma_{\mathbf{k}} dW_{\mathbf{k}}, \tag{4.41}$$

where $W_{\mathbf{k}}$ is a complex random variable and the randomness is used to include the unresolved small-scale features. It is clear that once the Fourier coefficients are known, the underlying velocity is determined since the Fourier bases are deterministic functions. The observation is the tracer trajectory $\mathbf{x} = (x, y)^{T}$, which

satisfies

$$\begin{aligned} d\mathbf{x} &= \mathbf{v}dt + \sigma_{\mathbf{x}} d\mathbf{W}, \\ &= \begin{pmatrix} -ie^{iy}\hat{u}_{0,1} + c.c. \\ ie^{ix}\hat{u}_{1,0} + c.c. \end{pmatrix} dt + \sigma_{\mathbf{x}} \begin{pmatrix} dW_x \\ dW_y \end{pmatrix} \end{aligned} \tag{4.42}$$

where the deterministic part is the Newton's law while the stochastic part accounts for the small-scale noise. The parameters used here are

$$d_{1,0} = 1.0, \quad d_{0,1} = 1.2, \quad \sigma_{1,0} = \sigma_{0,1} = 1, \quad f_{1,0} = f_{0,1} = 0, \quad \sigma_{\mathbf{x}} = 0.5. \tag{4.43}$$

The coupled system (4.41)–(4.42) is a nonlinear system because $\mathbf{x}$ appears in the exponential function in (4.42). Nevertheless, given the observed tracer trajectories, the velocity field is a Gaussian process. Thus, the model belongs to the general conditional Gaussian framework (3.2). In fact, the flow field itself does not depends on the tracer trajectories and therefore the coupled system belongs to the special class of the conditional Gaussian nonlinear model (4.2) and multiple tracer observations can be used to recover the flow field.

Figure 4.9 shows the truth and the recovered flow field from the filter and the smoother estimates. Panels (a)–(b) show the situations using $L = 2$ tracers while Panels (c)–(d) correspond to the case using $L = 12$ tracers. Note that the degree of freedom (DoF) of the underlying flow is DoF $= 4$ (in total 4 Fourier modes). With a small number of tracers, the filter estimate is significantly less accurate than the smoother estimate (see the pattern correlation within each panel). In fact, when

the underlying flow has a sudden change, the filter estimate tends to have a delay related to the truth. This can be seen for example around $t = 12.4$ in the second and the fourth rows. The reason is that the tracer trajectory is the result of an integrated effect of the underlying flow field and therefore the change of the tracer movement does not completely synchronize with that of the underlying flow. Such a delayed effect will be greatly mitigated when the number of the tracers becomes large, say $L = 12$, in which case the filter and the smoother estimates become closer to each other with small uncertainties. Yet, even in the situation that $L \gg$DoF, the recovered signal still intermittently misses the true values, for example at time $t = 10.5$.

Figure 4.10 compares the true velocity field with the recovered ones based on the filter and the smoother posterior mean estimates at three different time instants $t = 10.5, 12.4$ and 14.0. The true flow field alternates between jet flows (e.g., $t = 10.5$) and vortices (e.g., $t = 12.4$). With $L = 2$ tracers (Panels (a)–(c)), despite the fact that the overall patterns of the true signals are recovered by the filter and the smoother estimates and the path-wise error is only moderate, the recovered flow field can have a completely different pattern compared with the truth. For example, the true flow field at $t = 10.5$ (Panel (a)) and $t = 14$ (Panel (c)) have patterns with strong jets and strong vortices, respectively, while the recovered patterns from the filter mean estimate are the opposite. It is important to note that at $t = 12.4$ (Panel (b)), all the three recovered Fourier coefficients $\mathrm{Re}[\hat{u}_{0,1}(t)]$, $\mathrm{Re}[\hat{u}_{1,0}(t)]$ and $\mathrm{Im}[\hat{u}_{1,0}(t)]$ from the filter estimate nearly perfectly match the truth. The only error in the filtered signal is the delayed response of $\mathrm{Im}[\hat{u}_{0,1}(t)]$. However, even the error in this single Fourier mode is able to cause a significant difference between the

recovered flow pattern and the truth, where the truth is dominated by a jet along the y-axis together with a few vortices while the jets in the filtered flow are along the x-axis. In addition, there is no surprising that the recovered flow field from the filter estimate is less accurate than that from the smoother. Yet, with a small number of the tracers, larger errors also occur in the smoother estimate intermittently. Next, when L is increased to $L = 12$ (Panels (d)–(f)), which is much larger than the DoF of the underlying flow, the path-wise error in all the Fourier coefficients become much smaller, where in particular the pattern correlations between the true signals and the filtered ones are all beyond 0.84. However, as shown in Panels (e)–(f), the filtered flow fields are still quite different from the truth. As a comparison, the recovered flow fields from the smoother stay nearly the same as the truth.

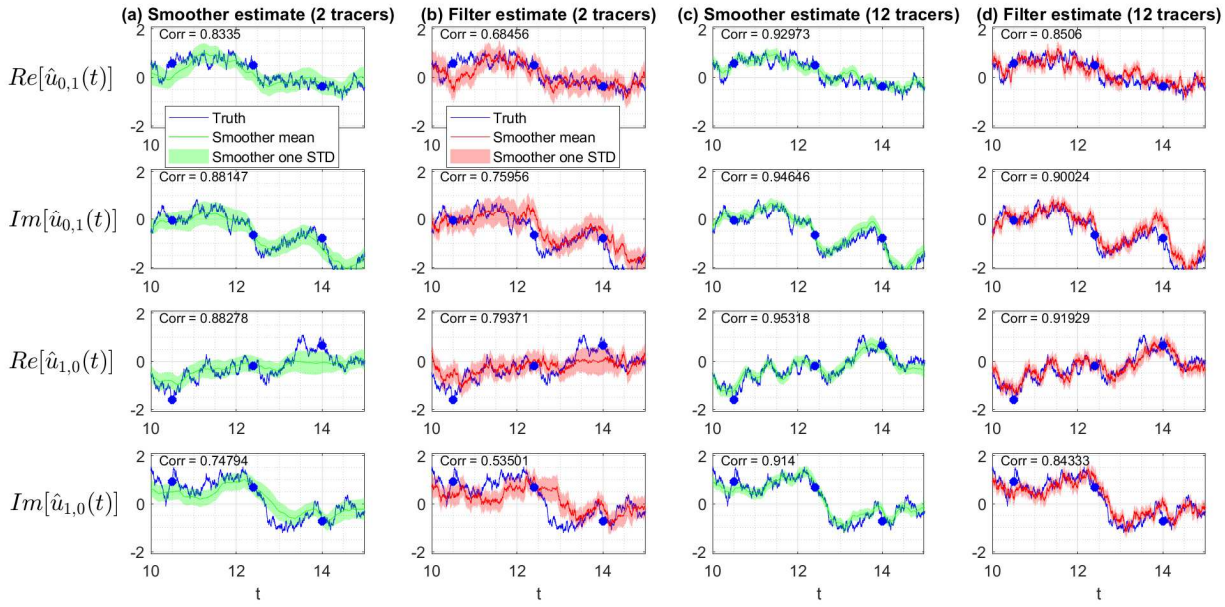

Figure 4.9: Comparison of the truth and the recovered flow field in terms of the Fourier coefficients in the Lagrangian data assimilation framework (4.41)–(4.42). Panels (a): the truth and the smoother estimate with L = 2 tracers. Panels (b): the truth and the filter estimate with L = 2 tracers. Panels (c): the truth and the smoother estimate with L = 12 tracers. Panels (d): the truth and the filter estimate with L = 12 tracers.

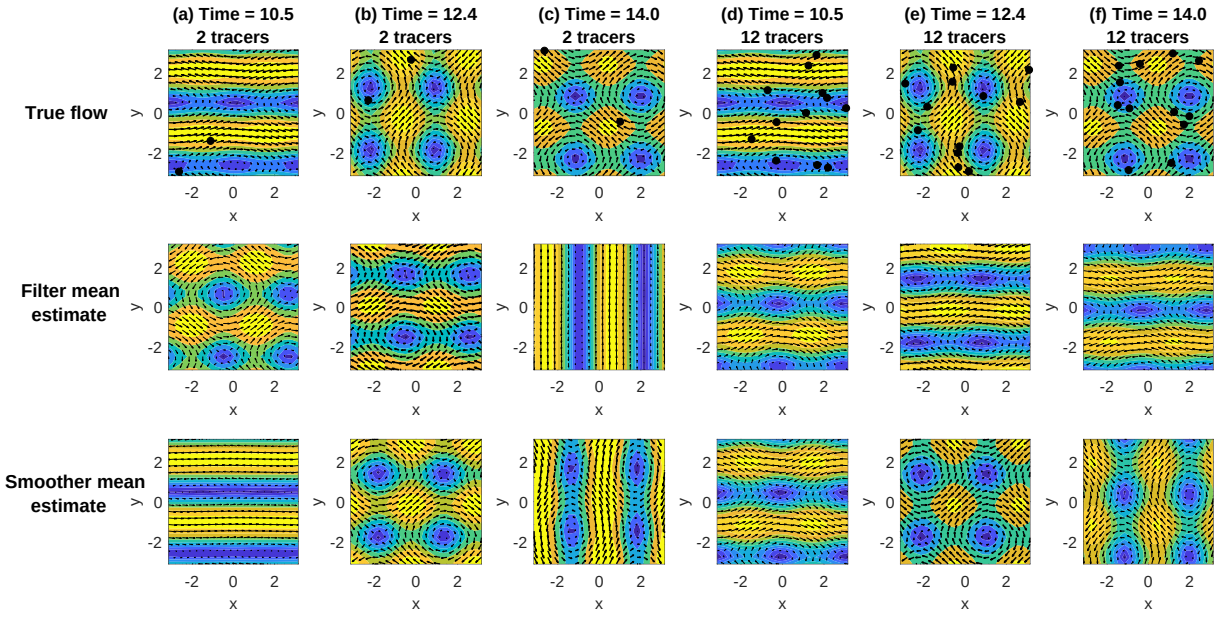

Figure 4.10: Comparison of the truth and the recovered flow field in physical space in the Lagrangian data assimilation framework (4.41)–(4.42). Panels (a)–(c): comparison at t = 10.5, t = 12.4 and t = 14 using L = 2 tracers, which are marked by black dots in the first row. Panels (d)–(f): comparison at t = 10.5, t = 12.4 and t = 14 using L = 12 tracers.

4.6 An Online Parameter Estimation Method

Parameter estimation of complex nonlinear dynamical systems is an important topic and a necessary precursor for effective state estimation, data assimilation and prediction. For partially observed nonlinear systems, parameter estimation and state estimation need to be carried out simultaneously. Approximations and expensive numerical methods are often required for general nonlinear systems in the presence of partial observations. Nevertheless, the closed analytic formulae of the nonlinear smoother in Theorem 4.9 allow an efficient online parameter estimation scheme for the conditional Gaussian nonlinear systems (3.2). In this section, an expectation-maximization (EM) algorithm (Ghahramani and Roweis, 1999; Ghahramani and Hinton, 1996) is developed to estimate the parameters and the hidden states alternatively, where closed analytic formulae are used in estimating both the states and the parameters.

4.6.1 The EM algorithm

The parameter estimation aims at seeking an optimal estimation of the unknown parameters θ by maximizing the log likelihood function. Since only the time series of $\mathbf{X}$ are observed, the maximum log likelihood estimate is given by

$$\mathcal{L}(\theta) = \log p(\mathbf{X}|\theta) = \log \int_{\mathbf{Y}} p(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{Y}. \tag{4.44}$$

The EM iteration alternates between performing an expectation (E) step, which estimates the hidden state of $\mathbf{Y}$ using the current estimate for the parameters, and a

maximization (M) step, which computes parameters by maximizing the expected log-likelihood found on the E step (Dembo and Zeitouni, 1986; Kokkala et al., 2014).

E-Step Compute the conditional distribution $p(\mathbf{Y}|\mathbf{X}, \theta_k)$ based on the current parameter θ_k and calculate the expectation

$$\mathcal{Q}(\theta; \theta_k) = \int_{\mathbf{Y}} p(\mathbf{Y}|\mathbf{X}, \theta_k) \log p(\mathbf{Y}, \mathbf{X}|\theta) d\mathbf{Y}, \tag{4.45}$$

the solution of which is exactly the nonlinear smoother estimate (3.6).

M-Step Update parameters θ by solving

$$\theta_{k+1} = \operatorname{argmax}_{\theta} \mathcal{Q}(\theta; \theta_k). \tag{4.46}$$

Since $\mathbf{Y}$ is unobserved, the conditional expectation needs to be taken for those functions containing $\mathbf{Y}$ when computing the log likelihood estimate. which takes into account the uncertainty of the estimated $\mathbf{Y}$. In addition, the log likelihood estimate only involves linear and quadratic functions of $\mathbf{Y}$, which are provided by the solutions of the smoother mean and smoother covariance (3.6). The technique details of this online parameter estimation algorithm is shown in Appendix C.5.

In practice, the observational data of $\mathbf{X}$ is obtained sequentially. Therefore, the online nonlinear smoother algorithm developed in Section 4.4 can be incorporated into the EM algorithm. Below, numerical examples show the online update of the estimated parameters with sequential observations.

4.6.2 Numerical examples

4.6.2.1 The SPEKF model

The SPEKF model (4.37) is used to understand the update of the estimated parameters when the partial observational data is available sequentially. The true parameters are the same as in (4.38) except $f_u = 1$ and $w = 0$. The entire time series of the observed variable u in the parameter estimation is shown in Panel (a) of Figure 4.11, where the associated unobserved time series of γ is shown in Panel (b). Recall that the unobserved variable γ serves as a stochastic damping of the dynamics of u. When γ is below zero, extreme events are triggered in the observed process u.

In the EM iterations, the initial guesses of the parameters are given as follows:

$$f_u^{(0)} = 3, \qquad \sigma_u^{(0)} = 0.8, \qquad d_\gamma^{(0)} = 4, \qquad f_\gamma^{(0)} = 4, \qquad \sigma_\gamma^{(0)} = 2,$$

which are far away from the true values in (4.38).

Figure 4.12 shows the parameter estimation results. The green curves are the trace plots from the offline EM algorithm given the entire observational time series with 50 time units. The initial errors decay quite fast, and the offline algorithm provides accurate estimations of the parameters.

Next, the online parameter estimation is carried out. A short time series including only the first 10 time units (20% of the entire observations in Panel (a) of Figure 4.11) is used as a start. For each EM iteration, a small amount of extra data with 0.005 units is included sequentially. It is interesting to see that there are

three significant drops around $k = 1600, 4000$ and 5000 (marked as I, II and III in Figure 4.12) towards the true states. The associated time instants of the sequential input of the data are around $t = 18, 30$ and 35 (marked as I, II and III in Figure 4.11), which correspond to the three extreme events in the observed time series u. These extreme events have a large signal-to-noise ratio and are therefore able to provide more dynamical information to the parameter estimation algorithm.

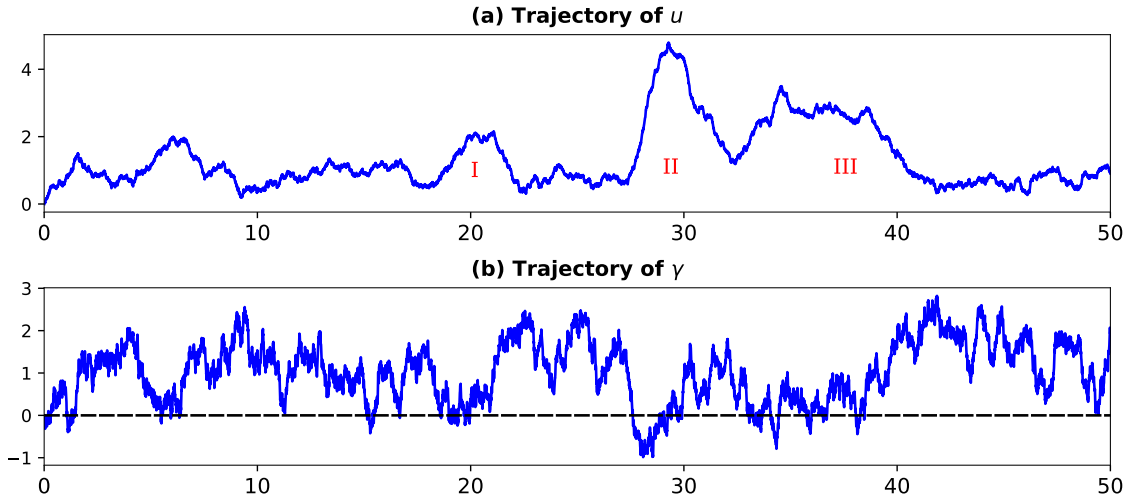

Figure 4.11: True signal (training period) of the SPEKF model (4.37) with true parameter (4.38) (with $f_u = 1$ and $w = 0$). The horizontal dashed line shows the intermittent threshold $v^* = 0$.

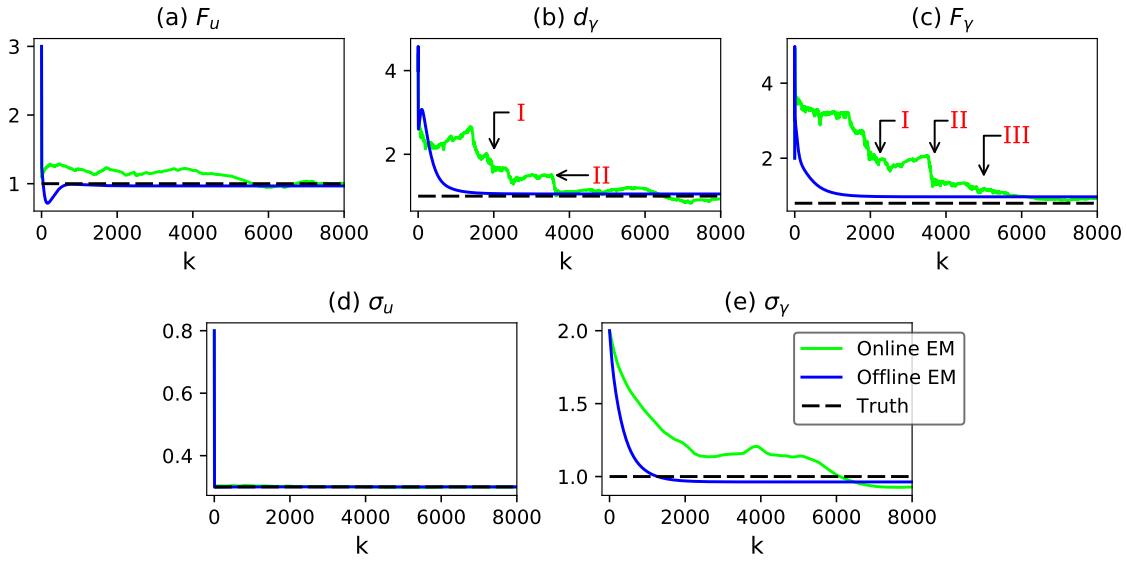

Figure 4.12: Trace plots of the online and offline parameter estimation algorithms (given in green and blue line, respectively) for the SPEKF model (4.37). Here k is the iteration step. The black dashed line in each panel shows the true parameter values in (4.38) with $f_u = 1$ and $w = 0$.

4.6.2.2 The noisy Lorenz 63 model

The second test example here is the noisy Lorenz 63 model (4.35). Different from the SPEFK model where random noise in v is the triggering mechanism of the extreme events in u, the events with large amplitudes in the noisy Lorenz 63 model are triggered by the intrinsic chaotic nature due to the deterministic nonlinearity. Here, the true parameters are the standard ones that allow the butterfly profiles of the solution with some small noises,

$$\rho = 28, \qquad \sigma = 10, \qquad \beta = 8/3, \qquad \text{and} \qquad \sigma_x = \sigma_y = \sigma_z = 1. \tag{4.47}$$

Only the time series of x is observed. The total length of the observations for the offline algorithm is 5 time units. For the online parameter estimation, only 1 unit of the time series is used as the start and then a small amount of extra data with 0.002 units is included sequentially for each EM iteration. The trace plots of the online and offline algorithms are shown by green and blue curves, respectively, in Figure 4.13. Both the online and offline algorithms have high parameter estimation skill using even short training periods, which is due to the fact that the nonlinear dynamics rather than the noise dominates the model behavior.

4.7 Conclusion

In this article, efficient nonlinear filtering, nonlinear smoothing, forward sampling and backward sampling algorithms are developed for a rich class of nonlinear com-

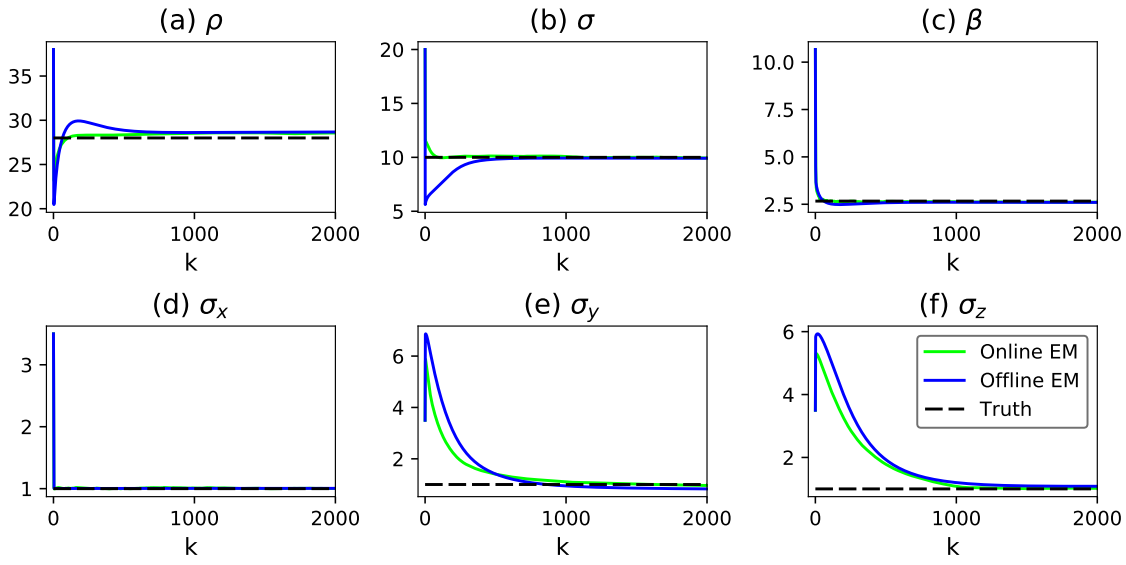

Figure 4.13: Trace plots of the online and offline parameter estimation algorithms (given in green and blue line, respectively) for the noisy Lorenz 63 model (4.35). Here k is the iteration step. The black dashed line in each subplot shows the true parameter values in (4.47).

plex turbulent dynamical systems using only partial observations. The nonlinear filter and the nonlinear smoother provide the optimal point-wise state estimation while the optimal sampling algorithms facilitate to sample hidden trajectories conditioned on observations. These sampled trajectories recover the temporal dependence as well as many other salient nonlinear dynamical features, which cannot be captured by the point-wise state estimation. The closed analytic formulae in these algorithms allow them to be accurately applied to highly nonlinear systems in large dimensions. Since sequential observations appear in many practical situations, an online nonlinear smoother and the associated sampling technique are developed here as well. The applications of these algorithms to state estimation,

understanding hidden extreme events, Lagrangian data assimilation, stochastic parameterizations and parameter estimation have been illustrated with the help of numerical simulations.

All the models considered here are perfect. One future direction of the work is to study the model error in filtering, smoothing and sampling. Another potential application is to make use of the sampled trajectories to quantify the uncertainty in complex nonlinear dynamical systems.

A.1 Details about the ensemble Kalman filter and smoother

Assume that the prior model from time t_{n-1} to t_n is given as following form:

$$\mathbf{x}_n^M = \mathcal{F}^M(\mathbf{x}_{n-1}^M) + \boldsymbol{\sigma}_n^M. \tag{A.1}$$

The goal of filtering problem is to estimate the unknown true state $\mathbf{x}_n$, given noisy and partial observations

$$\mathbf{y}_n = \mathbf{H}\mathbf{x}_n + \boldsymbol{\sigma}_n^o, \tag{A.2}$$

where $\mathbf{y}_n$ is a m-dimensional measurement vector, $\mathbf{H}$ is a m by n matrix, and $\boldsymbol{\sigma}_n^o$ is a vector of unbiased Gaussian noise with variance $\mathbf{R}_n^o$. Thus, the filtering problem can be formulated as maximizing the conditional distribution $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ sequentially for each time $t = 1, 2, \cdots$ Based on the Bayesian's formula, the conditional distribution satisfies

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) \propto p(\mathbf{x}_n|\mathbf{y}_{1:n-1})p(\mathbf{y}_n|\mathbf{x}_n), \tag{A.3}$$

where the prior estimates $p(\mathbf{x}_n|\mathbf{y}_{1:n-1})$ is obtained from underlying model forecast (A.1) and $p(\mathbf{y}_n|\mathbf{x}_n)$ is the likelihood of estimating $\mathbf{x}_n$ given observation $\mathbf{y}_n$.

Ensemble Kalman filter (EnKF) is one of the typical methods used for filtering the nonlinear underlying dynamics. It is based on an assumption that the prior

distribution is approximated by Gaussian and the mean and covariance are formed by a finite number K of sampled members. More specifically, assume that the filtering ensemble at time t_{n-1} is given by $\{\mathbf{x}^k_{n-1|n-1}\}_{k=1,\dots,K}$, the forecast ensemble $\{\mathbf{x}^k_{n|n-1}\}_{k=1,\dots,K}$ is obtained by applying forecast model (A.1) to each ensemble member. Thus, the prior distribution is

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}), \tag{A.4}$$

where $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, and $\boldsymbol{\mu}_{n|n-1}$ and $\boldsymbol{\Sigma}_{n|n-1}$ are sample mean and sample covariance, respectively. Given this Gaussian prior, the posterior distribution is also Gaussian

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{1:n}) &\propto p(\mathbf{x}_n|\mathbf{y}_{1:n-1})p(\mathbf{y}_n|\mathbf{x}_n) \\ &= \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1})\mathcal{N}(\mathbf{y}_n|\mathbf{H}_n\mathbf{x}_n, \mathbf{R}^o_n) \\ &\propto \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}), \end{aligned} \tag{A.5}$$

where the filtered mean $\boldsymbol{\mu}_{n|n}$ and filtered covariance $\boldsymbol{\Sigma}_{n|n}$ are given by

$$\begin{aligned} \boldsymbol{\mu}_{n|n} &= \boldsymbol{\mu}_{n|n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{H}_n\boldsymbol{\mu}_{n|n-1}), \\ \boldsymbol{\Sigma}_{n|n} &= (\mathbf{I} - \mathbf{K}_n\mathbf{H}_n)\boldsymbol{\Sigma}_{n|n-1}, \end{aligned} \tag{A.6}$$

with the Kalman gain $\mathbf{K}_n$

$$\mathbf{K}_n = \boldsymbol{\Sigma}_{n|n-1}\mathbf{H}'_n(\mathbf{H}_n\boldsymbol{\Sigma}_{n|n-1} + \mathbf{R}^o_n)^{-1}. \tag{A.7}$$

The ensemble Kalman filter algorithm is given in Algorithm 4.

Algorithm 4: Ensemble Kalman filter

Start with an initial ensemble $\{\mathbf{x}_{0|0}^{k}\}_{k=1,\dots,K}$ sampled from initial distribution $\mathcal{N}(\boldsymbol{\mu}_{0|0}, \mathbf{R}_{0|0})$;

for $n = 1, \dots$ **do**

  for $k = 1, \dots, K$ **do**

    Compute forecast ensemble members using $\mathbf{x}_{n|n-1}^{k} = \mathcal{F}^{M}(\mathbf{x}_{n-1}^{k}) + \boldsymbol{\sigma}_{n}^{M}$;

    Estimate prior ensemble covariance $\boldsymbol{\Sigma}_{n|n-1}$ by the samples $\{\mathbf{x}_{n|n-1}^{k}\}$;

    Compute the Kalman gain $\mathbf{K}_n = \boldsymbol{\Sigma}_{n|n-1}\mathbf{H}_n'(\mathbf{H}_n\boldsymbol{\Sigma}_{n|n-1}\mathbf{H}_n' + \mathbf{R}_n^{o})^{-1}$;

    Draw a sample $\mathbf{v}_n^{k} \sim \mathcal{N}(0, \mathbf{R}_n^{o})$;

    Update ensemble member $\mathbf{x}_{n|n}^{k} = \mathbf{x}_{n|n-1}^{k} + \mathbf{K}_n(\mathbf{x}_n - \mathbf{H}_n\mathbf{x}_{n|n-1}^{k} - \mathbf{v}_n^{k})$;

For smoothing, it exploits the information in the entire observational window, including the past and the future. Therefore, the smoothing problem optimizes the entire state $\{\mathbf{x}_{0:n}\}$ given all available observations $\{\mathbf{y}_1, \dots, \mathbf{y}_n\}$. Ensemble Kalman smoother (EnKS) Evensen and Van Leeuwen (2000) can be easily extended from EnKF. It starts from the sequential formulation

$$\begin{aligned} p(\mathbf{x}_{0:n}|\mathbf{y}_{1:n}) &\propto p(\mathbf{x}_{0:n-1}|\mathbf{y}_{1:n-1})p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{y}_n|\mathbf{x}_n) \\ &= p(\mathbf{x}_{0:n}|\mathbf{y}_{1:n-1})p(\mathbf{y}_n|\mathbf{x}_n), \end{aligned} \tag{A.8}$$

which is obtained using Bayesian's formula, the definition of the conditional PDF and the Markov property of the system. Thus, the vanilla version of EnKS algorithm is given by Algorithm 5. Evensen and Van Leeuwen (2000); Katzfuss et al. (2020);

Cosme et al. (2012)

Algorithm 5: Ensemble Kalman smoother and sampling (EnKS)

Start with an initial ensemble $\{\mathbf{x}^k_{0|0}\}_{k=1,\dots,K}$ sampled from initial distribution $\mathcal{N}(\mu_{0|0}, \mathbf{R}_{0|0})$;

for $n = 1, \dots$ **do**

  for $k = 1, \dots, K$ **do**

    Compute forecast ensemble members using $\mathbf{x}^k_{n|n-1} = \mathcal{F}^{\mathcal{M}}(\mathbf{x}^k_{n-1}) + \boldsymbol{\sigma}_n$;

    Estimate prior ensemble covariance $\boldsymbol{\Sigma}_{n|n-1}$ by the samples $\{\mathbf{x}^k_{n|n-1}\}$;

    for $n' = 0, \dots, n$ **do**

      Compute cross-covariance $\boldsymbol{\Sigma}_{n',n|n-1}$ between ensemble $\{\mathbf{x}^k_{n'|n-1}\}$ and $\{\mathbf{x}^k_{n|n-1}\}$;

      Compute the Kalman gain $\mathbf{K}_{n',n} = \boldsymbol{\Sigma}_{n',n|n-1}\mathbf{H}'_n(\mathbf{H}_n\boldsymbol{\Sigma}_{n,n|n-1}\mathbf{H}'_n + \mathbf{R}^o_n)^{-1}$;

      Draw a sample $\mathbf{v}^k_n \sim \mathcal{N}(0, \mathbf{R}^o_n)$;

      Update ensemble member $\mathbf{x}^k_{n'|n} = \mathbf{x}^k_{n'|n-1} + \mathbf{K}_{n',n}(\mathbf{x}_n - \mathbf{H}_n\mathbf{x}^k_{n|n-1} - \mathbf{v}^k_n)$;

A.2 Parameters related to perfect model and data assimilation

The parameters used in perfect models and data assimilation are shown in Table A.1.

Table A.1: Parameters used perfect model and data assimilation.

	model parameters												data assimilation	
L96	I	J	h	c	b	f	σ_{u_i}	$\sigma_{v_{i,j}}$					σ^o	$\Delta_{obs}t$
	40	4	2	2	2	4	1	1					1	0.05
Triad model	γ_1	γ_2	γ_3	L_{12}	L_{13}	L_{23}	I	δ	σ_1	σ_2	σ_3	F	σ^o	$\Delta_{obs}t$
	2	0.2	0.4	0.2	0.1	0	5	1	0.5	1.2	0.8	2	0.2	0.05

A.3 Hyperparameters used in the LSTM models

Table A.2: Hyperparameters used in the LSTM models.

L96 LSTM models				
Lead time	[1, 10]	[11, 16]	[16, 70]	[71, 100]
Maximum epochs	300	100	100	100
Hidden dimension	64	64	32	16
Learning rate	0.001			
Batch size	64			
L_{init}	15 (0.75 time units)			
Triad LSTM model				
Lead time	[1, 3]	[4, 15]	[16, 40]	[41, 80]
Maximum epochs	200	100	30	30
Hidden dimension	64	64	32	16
Learning rate	0.0005			
Batch size	64			
L_{init}	15 (0.75 time units)			
Triad LSTM model (smoother mean)				
Lead time	[1, 3]	[4, 15]	[16, 40]	[41, 80]
Maximum epochs	1000	500	500	500
Hidden dimension	64	64	32	16
Learning rate	0.001			
Batch size	64			
L_{init}	15 (0.75 time units)			

The hyperparameters used in the LSTM models associated with the two imperfect L96 models and the imperfect triad model are listed in Table A.2.

B APPENDIX FOR CGNS AS FAST PRECONDITIONER AND CHEAP SURROGATE MODEL

B.1 Details of the EM algorithm for parameters estimation

Consider the discrete fashion of CGNS (3.2) using the Euler-Maruyama scheme Gardiner (2004),

$$\mathbf{X}^{j+1} = \mathbf{X}^j + (\mathbf{A}_0^j(\mathbf{X}, t; \theta) + \mathbf{A}_1^j(\mathbf{X}, t; \theta)\mathbf{Y}^j)\Delta t + \mathbf{B}_1^j(\mathbf{X}, t; \theta)\sqrt{\Delta t}\varepsilon_1^j, \tag{B.1a}$$

$$\mathbf{Y}^{j+1} = \mathbf{Y}^j + (\mathbf{a}_0^j(\mathbf{X}, t; \theta) + \mathbf{a}_1^j(\mathbf{X}, t; \theta)\mathbf{Y}^j)\Delta t + \mathbf{b}_2^j(\mathbf{X}, t; \theta)\sqrt{\Delta t}\varepsilon_2^j, \tag{B.1b}$$

where ε_1^j and ε_2^j are independent and identically distributed Gaussian white noises, and Δt is sufficiently small. Assume all the parameters appear as multiplicative prefactors of some functions of $\mathbf{X}^j$ and $\mathbf{Y}^j$ on the right hand side of (B.1). Thus, the log likelihood function of $p(\mathbf{X}, \mathbf{Y}|\theta)$ in M-step can be solved explicitly.

For (B.1), the local linear Gaussian approximation on the right hand side is

$$\mathcal{N}(\boldsymbol{\mu}^j, \mathbf{R}^j) = \widetilde{C}|\mathbf{R}^j|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{u}^{j+1} - \boldsymbol{\mu}^j)^T(\mathbf{R}^j)^{-1}(\mathbf{u}^{j+1} - \boldsymbol{\mu}^j)\right), \tag{B.2}$$

where $\mathbf{u}^{j+1} = (\mathbf{X}^{j+1}, \mathbf{Y}^{j+1})^T$ and $\boldsymbol{\mu}^j = \mathbf{M}^j\xi + \S^j$. Here, ξ is the parameters in drift part (B.1) and $\S^j$ is those terms that do not involve parameters such as the first terms $\mathbf{X}^j$ or $\mathbf{Y}^j$ in (B.1). The covariance $\mathbf{R}^j$ is a block diagonal matrix with

entries $(\mathbf{B}_1^j(\mathbf{X},t))(\mathbf{B}_1^j(\mathbf{X},t))^*\Delta t$ and $(\mathbf{b}_2^j(\mathbf{X},t))(\mathbf{b}_2^j(\mathbf{X},t))^*\Delta t$, which has an one-to-one correspondence with the parameters in the diffusion terms. The constant $\widetilde{C}$ is due to the normalization of a Gaussian distribution. Since the states $\mathbf{Y}$ is unobserved and it contains uncertainty, an expectation of the log-likelihood function as in (3.25) is adopted, and the overall objective function becomes

$$\widetilde{\mathcal{L}} = \frac{1}{2}\sum_j \left\langle (\mathbf{u}^{j+1} - \mathbf{M}^j\xi - \S^j)^*(\mathbf{R})^{-1}(\mathbf{u}^{j+1} - \mathbf{M}^j\xi - \S^j)\right\rangle - \frac{J}{2}\log|\mathbf{R}|, \tag{B.3}$$

where ξ and $\mathbf{R}$ are parameters in drift terms and diffusion terms, respectively. In (B.3), J is the total numbers of the terms in previous sum, and $\langle\cdot\rangle$ denotes the expectation over the uncertain component of $\mathbf{u}^j$, namely $\mathbf{Y}^j$, at fixed j while the expectation of the deterministic component $\mathbf{X}^j$ is simply itself. Here we assumed $\mathbf{R}^j = \mathbf{R}$ for all j. To find the minimum of $\widetilde{\mathcal{L}}$, we aim at finding $\frac{\partial\widetilde{\mathcal{L}}}{\partial\xi} = 0$ and $\frac{\partial\widetilde{\mathcal{L}}}{\partial\mathbf{R}} = 0$, which leads to

$$\mathbf{R} = \frac{1}{J}\sum_j \left\langle (\mathbf{u}^{j+1} - \mathbf{M}^j\xi - \S^j)(\mathbf{u}^{j+1} - \mathbf{M}^j\xi - \S^j)^*\right\rangle, \tag{B.4a}$$

$$\xi = \mathbf{D}^{-1}\mathbf{c}, \tag{B.4b}$$

where

$$\mathbf{D} = \sum_j \left\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}\mathbf{M}^j\right\rangle \quad \text{and} \quad \mathbf{c} = \sum_j \left\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}(\mathbf{u}^{j+1} - \S^j)\right\rangle. \tag{B.5}$$

B.1.1 Learning with constraints

In certain cases, one may want to constrain the element of ξ. For example, under the restriction

$$\mathbf{H}\xi = \mathbf{g}, \tag{B.6}$$

where $\mathbf{H}$ and $\mathbf{g}$ are constant matrices, we obtain the following objective function

$$\widetilde{\mathcal{L}} = \frac{1}{2}\sum_{j}\left\langle (\mathbf{u}^{j+1} - \mathbf{M}^{j}\xi - \S^{j})^{*}(\mathbf{R})^{-1}(\mathbf{u}^{j+1} - \mathbf{M}^{j}\xi - \S^{j})\right\rangle - \frac{J}{2}\log|\mathbf{R}^{-1}| + \boldsymbol{\lambda}^{*}(\mathbf{H}\xi - \mathbf{g}). \tag{B.7}$$

Therefore, the solution of the minimization problem with the new objective function (B.7) is given as follows

$$\mathbf{R} = \frac{1}{J}\sum_{j}\left\langle (\mathbf{u}^{j+1} - \mathbf{M}^{j}\xi - \S^{j})(\mathbf{u}^{j+1} - \mathbf{M}^{j}\xi - \S^{j})^{*}\right\rangle \tag{B.8a}$$

$$\boldsymbol{\lambda} = \left(\mathbf{H}\mathbf{D}^{-1}\mathbf{H}^{*}\right)^{-1}(\mathbf{H}\mathbf{D}^{-1}\mathbf{c} - \mathbf{g}), \tag{B.8b}$$

$$\xi = \mathbf{D}^{-1}\left(\mathbf{c} - \mathbf{H}^{*}\boldsymbol{\lambda}\right), \tag{B.8c}$$

where $\mathbf{D}$ and $\mathbf{c}$ are defined in (B.5).

B.1.2 Learning with block decomposition

Many complex systems with multiscale structures, multilevel dynamics or state-dependent parameterizations have the following block decomposition features. Now we develop an efficient strategy with block decomposition and incorporate it

into the basic algorithms. Consider the following decomposition of state variables

$$\mathbf{u} = \bigcup_{i=1}^{I} \mathbf{u}_i, \quad \text{with} \quad \mathbf{u}_i = (\mathbf{X}_i, \mathbf{Y}_i), \quad \mathbf{X}_i \in \mathbb{R}^{N_{I,i}} \quad \text{and} \quad \mathbf{Y}_i \in \mathbb{R}^{N_{II,i}}, \tag{B.9}$$

where $1 \leqslant i \leqslant I$, $N_{\mathbf{I}} = \sum_{i=1}^{I} N_{\mathbf{I},i}$ and $N_{\mathbf{II}} = \sum_{i=1}^{I} N_{\mathbf{II},i}$. Correspondingly, the full dynamics are decomposed into L groups, where the variables on the left-hand side of the lth group are $\mathbf{u}_i$. In addition, we assume both $\mathbf{B_1}$ and $\mathbf{b_2}$ are diagonal for notation simplicity. Furthermore, in the dynamics of each $\mathbf{u}_i$ in (3.2), the terms $\mathbf{A}_{0,i}$ and $\mathbf{a}_{0,i}$ can depend only on the components of $\mathbf{X}_i$ while the terms $\mathbf{A}_{1,i}$ and $\mathbf{a}_{1,i}$ are only functions of $\mathbf{Y}_i$; namely

$$\begin{aligned} \mathbf{A}_{0,i} :=& \mathbf{A}_{0,i}(t, \mathbf{X}), & \mathbf{a}_{0,i} :=& \mathbf{a}_{0,i}(t, \mathbf{X}), \\ \mathbf{A}_{1,i} :=& \mathbf{A}_{1,i}(t, \mathbf{X}_i), & \mathbf{a}_{1,i} :=& \mathbf{a}_{1,i}(t, \mathbf{X}_i). \end{aligned} \tag{B.10}$$

In addition, only $\mathbf{Y}_i$ involves with terms $\mathbf{A}_{1,i}$ and $\mathbf{a}_{1,i}$. The initial values of $(\mathbf{X}_i, \mathbf{Y}_i)$ and $(\mathbf{X}_{i'}, \mathbf{Y}_{i'})$ for all $i' \neq i$ are also assumed to be independent with each other. Therefore, we can decompose equation (3.2) as follows,

$$\frac{d\mathbf{X}_i(t)}{dt} = [\mathbf{A}_{0,i}(\mathbf{X}, t; \theta) + \mathbf{A}_{1,i}(\mathbf{X}_i, t; \theta)\mathbf{Y}_i(t)] + \mathbf{B_1}(\mathbf{X}, t; \theta)\dot{\mathbf{W}}_{\mathbf{1}}(t), \tag{B.11a}$$

$$\frac{d\mathbf{Y}_i(t)}{dt} = [\mathbf{a}_{0,i}(\mathbf{X}, t; \theta) + \mathbf{a}_{1,i}(\mathbf{X}_i, t; \theta)\mathbf{Y}_i(t)] + \mathbf{b_2}_{,i}(\mathbf{X}_i, t; \theta)\dot{\mathbf{W}}_{\mathbf{2},i}(t). \tag{B.11b}$$

Correspondingly, the evolution of the conditional mean and covariance and smoothing mean and covariance are given as follows

$$\frac{d\boldsymbol{\mu}_{\mathbf{f},i}(t)}{dt} =(\mathbf{a}_{0,i} + \mathbf{a}_{0,i}\boldsymbol{\mu}_{\mathbf{f},i}) + (\mathbf{R}_{\mathbf{f},i}\mathbf{A}^*_{1,i})(\mathbf{B}_{\mathbf{X},i}\mathbf{B}^*_{\mathbf{X},i})^{-1}\left(\frac{d\mathbf{X}_i}{dt} - (\mathbf{A}_{0,i} + \mathbf{A}_{1,i}\boldsymbol{\mu}_{\mathbf{f},i})\right), \tag{B.12a}$$

$$\frac{d\mathbf{R}_{\mathbf{f},i}(t)}{dt} = \left(\mathbf{a}_{1,i}\mathbf{R}_{\mathbf{f},i} + \mathbf{R}_{\mathbf{f},i}\mathbf{a}^*_{1,i} + \mathbf{b}_{2,i}\mathbf{b}^*_{2,i} - (\mathbf{R}_{\mathbf{f},i}\mathbf{A}^*_{1,i})(\mathbf{B}_{\mathbf{X},i}\mathbf{B}^*_{\mathbf{X},i})^{-1}(\mathbf{A}_{1,i}\mathbf{R}_{\mathbf{f},i})\right), \tag{B.12b}$$

and

$$\frac{d(-\boldsymbol{\mu}_{\mathbf{s},i})}{dt} = \left(-\mathbf{a}_{0,i} - \mathbf{a}_{1,i}\boldsymbol{\mu}_{\mathbf{s},i} + ((\mathbf{b}_{2,i}\mathbf{b}^*_{2,i})\mathbf{R}^{-1}_{\mathbf{f},i}(\boldsymbol{\mu}_{\mathbf{f},i} - \boldsymbol{\mu}_{\mathbf{s},i}))\right), \tag{B.13a}$$

$$\frac{d(-\mathbf{R}_{\mathbf{s},i})}{dt} = -\left((\mathbf{a}_{1,i} + \mathbf{b}_{2,i}\mathbf{b}^*_{2,i})\mathbf{R}^{-1}_{\mathbf{f},i})\mathbf{R}_{\mathbf{s},i} + \mathbf{R}_{\mathbf{s},i}(\mathbf{a}^*_{1,i} + (\mathbf{b}_{2,i}\mathbf{b}^*_{2,i})\mathbf{R}^{-1}_{\mathbf{f},i}) - (\mathbf{b}_{2,i}\mathbf{b}^*_{2,i})\right) \tag{B.13b}$$

When computing the log likelihood, we further assumed that the parameters to have a decomposition $\theta = \bigcup_{i=1}^{I} \theta_i$, where the parameters θ_i appear only in the equations of $\mathbf{X}_i$ and $\mathbf{Y}_i$. Therefore, the objective function (B.3) becomes

$$\begin{aligned}\widetilde{\mathcal{L}} &= \left(\frac{1}{2}\sum_j \left\langle (\mathbf{u}^{j+1} - \mathbf{M}^j\boldsymbol{\xi} - \S^j)^*(\mathbf{R})^{-1}(\mathbf{u}^{j+1} - \mathbf{M}^j\boldsymbol{\xi} - \S^j)\right\rangle - \frac{J}{2}\log|\mathbf{R}^{-1}|\right) \\ &= \sum_i \left(\sum_j \frac{1}{2}\left\langle (\mathbf{u}_i^{j+1} - \mathbf{M}_i^j\boldsymbol{\xi}_i - \S_i^j)^*(\mathbf{R}_i)^{-1}(\mathbf{u}_i^{j+1} - \mathbf{M}_i^j\boldsymbol{\xi}_i - \S_i^j)\right\rangle - \frac{J}{2}\log|\mathbf{R}_i^{-1}|\right).\end{aligned} \tag{B.14}$$

Similarly, the optimal solution of the minimization problem of the new objective function of each block can be achieved via

$$\mathbf{R}_i = \frac{1}{J}\sum_j \left\langle (\mathbf{u}_i^{j+1} - \mathbf{M}_i^j \theta_i - \S_i^j)(\mathbf{u}_i^{j+1} - \mathbf{M}_i^j \xi_i - \S_i^j)^* \right\rangle, \tag{B.15a}$$

$$\xi_i = \left(\sum_j \left\langle (\mathbf{M}_i^j)^* \mathbf{R}_i^{-1} \mathbf{M}_i^j \right\rangle \right)^{-1} \left(\sum_j \left\langle (\mathbf{M}_i^j)^* \mathbf{R}_i^{-1} (\mathbf{u}_i^{j+1} - \S_i^j) \right\rangle \right). \tag{B.15b}$$

B.1.2.1 Block decomposition of the approximate two-layer L96 model (3.29)

Following general block decomposition from previous section, each block $\mathbf{u}_i$ for the approximate two-layer L96 model (3.29) is given as $(u_i, v_{l,1}, v_{l,2}, \cdots, v_{l,J})$. Correspondingly, $\mathbf{A}_{0,i}$, $\mathbf{a}_{0,i}$, $\mathbf{A}_{1,i}$, $\mathbf{a}_{1,i}$, $\mathbf{B}_{\mathbf{1},i}$, and $\mathbf{b}_{\mathbf{2},i}$ are

$$\mathbf{A}_{0,i} = -u_{i-1}(u_{i-2} - u_{i+1}) - u_i + \hat{f}_i, \qquad \mathbf{A}_{1,i} = (-\hat{a}_i, -\hat{a}_i, \ldots, -\hat{a}_i), \qquad \mathbf{B}_{\mathbf{1},i} = \hat{\sigma}_{u_i},$$

$$\mathbf{a}_{0,i} = \begin{pmatrix} u_i + \hat{v}_{i,1} \\ u_i + \hat{v}_{i,2} \\ \vdots \\ u_i + \hat{v}_{i,J} \end{pmatrix}, \quad \mathbf{a}_{1,i} = \begin{pmatrix} -\hat{d}_{i,1} & & & \\ & -\hat{d}_{i,2} & & \\ & & \ddots & \\ & & & -\hat{d}_{i,J} \end{pmatrix}, \quad \mathbf{b}_{\mathbf{2},i} = \begin{pmatrix} \hat{\sigma}_{v_{i,1}} & & & \\ & \hat{\sigma}_{v_{i,2}} & & \\ & & \ddots & \\ & & & \hat{\sigma}_{v_{i,J}} \end{pmatrix}. \tag{B.16}$$

Therefore, the time evolution of mean and covariance for the filter (B.12) and the smoother (B.13) can be computed in parallel.

B.2 Calculating $\mathcal{B}(\mathbf{u})$

In light of (3.34) and (3.35), one has the following explicit expression of $\mathcal{B}(u)$

$$\begin{aligned}\mathcal{B}(\mathbf{u}) &= -\frac{\operatorname{div}_{\mathbf{u}}(\mathbf{w}(\mathbf{u})p_{eq}^{M|obs}(\mathbf{u}))}{p_{eq}^{M|obs}(\mathbf{u})} \\ &= -\sum_{i=1}^{N}\frac{\partial}{\partial \mathbf{u}_i}\mathbf{w}_i(\mathbf{u}) - \sum_{i=1}^{N}\mathbf{w}_i\frac{\partial}{\partial \mathbf{u}_i}p_{eq}^{M|obs}(\mathbf{u})\,.\end{aligned} \tag{B.17}$$

When perturbing the parameters of forcing in the observed processes, the forms of $\mathcal{B}(\mathbf{u})$ from the perfect model and the approximate model are the same, since the parameters F_1 and F_2 appear exactly the same way as in both the perfect and the approximate model. The $\mathcal{B}(\mathbf{u})$ term reads as follows

$$\mathcal{B}(\mathbf{u}) = -\frac{\partial}{\partial x_1}p_{eq}^{M|obs}(\mathbf{u}) - \frac{\partial}{\partial x_2}p_{eq}^{M|obs}(\mathbf{u})\,. \tag{B.18}$$

When perturbing the parameters in linear interactions terms that appear in both the observed and hidden processes, the formulation of $\mathcal{B}(\mathbf{u})$ from the perfect and approximate models are different. Given the perturbation vector $\mathbf{w}(\mathbf{u}) = (y_1, y_2, -x_1, -x_2)^\top$, the $\mathcal{B}(\mathbf{u})$ from the perfect model is as follows

$$\mathcal{B}(\mathbf{u}) = -y_1\frac{\partial}{\partial x_1}p_{eq}^{M|obs}(\mathbf{u}) - y_2\frac{\partial}{\partial x_2}p_{eq}^{M|obs}(\mathbf{u}) + x_1\frac{\partial}{\partial y_1}p_{eq}^{M|obs}(\mathbf{u}) + x_2\frac{\partial}{\partial y_2}p_{eq}^{M|obs}(\mathbf{u})\,. \tag{B.19}$$

However, since there is no L_{13} and L_{24} parameters in the hidden processes of the approximate model, the formulation of $\mathcal{B}(\mathbf{u})$ from the approximate model remains

$$\mathcal{B}(\mathbf{u}) = -y_1 \frac{\partial}{\partial x_1} p_{eq}^{M|obs}(\mathbf{u}) - y_2 \frac{\partial}{\partial x_2} p_{eq}^{M|obs}(\mathbf{u}). \tag{B.20}$$

C.1 Auxiliary formulae

C.1.1 The mean-fluctuation decomposition

The mean-fluctuation decomposition of a random variable $\mathbf{Y}$ at a fixed time instant is given by Adrian et al. (2000); Vallis (2017)

$$\mathbf{Y} = \overline{\mathbf{Y}} + \mathbf{Y}', \tag{C.1}$$

where $\overline{\mathbf{Y}}$ is the ensemble average (namely the ensemble mean) of $\mathbf{Y}$ and $\overline{\mathbf{Y}'} = 0$ is the residual. The covariance of $\mathbf{Y}$ is given by $\overline{\mathbf{Y}'(\mathbf{Y}')^*}$.

C.1.2 Some useful properties of multivariate Gaussian distributions

Let us denote a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{R}$ by $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{R})$, where $\mathbf{x}$ is the random variable.

Lemma C.1. *For two Gaussian distributions,*

$$\int \mathcal{N}(\mathbf{x_2}|\mathbf{F}\mathbf{x_1} + \mathbf{b}, \mathbf{R_2})\mathcal{N}(\mathbf{x_1}|\boldsymbol{\mu_1}, \mathbf{R_1})\mathrm{d}\mathbf{x_1} = \mathcal{N}(\mathbf{x_2}|\mathbf{F}\boldsymbol{\mu_1} + \mathbf{b}, \mathbf{F}\mathbf{R_1}\mathbf{F}^* + \mathbf{R_2}). \tag{C.2}$$

Lemma C.2. *Let the Gaussian random variables be*

$$\mathbf{x} = \begin{pmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{pmatrix},$$

with mean $\boldsymbol{\mu}$ *and covariance* $\mathbf{R}$*,*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{pmatrix}, \qquad \mathbf{R} = \begin{pmatrix} \mathbf{R_{11}} & \mathbf{R_{12}} \\ \mathbf{R_{21}} & \mathbf{R_{22}} \end{pmatrix}.$$

The conditional distribution

$$p(\mathbf{x_1}|\mathbf{x_2}) \sim \mathcal{N}(\overline{\boldsymbol{\mu}}, \overline{\mathbf{R}}),$$

where

$$\begin{aligned} \overline{\boldsymbol{\mu}} &= \boldsymbol{\mu_1} + \mathbf{R_{12}}\mathbf{R_{22}}^{-1}(\mathbf{x_2} - \boldsymbol{\mu}_2), \\ \overline{\mathbf{R}} &= \mathbf{R_{11}} - \mathbf{R_{12}}\mathbf{R_{22}}^{-1}\mathbf{R_{21}}. \end{aligned} \tag{C.3}$$

C.1.3 Discrete approximation

Let us start writing the nonlinear conditional Gaussian system (3.2) in a discrete fashion using the Euler-Maruyama scheme Gardiner (2004),

$$\mathbf{X}^{j+1} = \mathbf{X}^j + (\mathbf{A}_0^j + \mathbf{A}_1^j\mathbf{Y}^j)\Delta t + \mathbf{B}_1^j\sqrt{\Delta t}\epsilon_1^j, \tag{C.4a}$$

$$\mathbf{Y}^{j+1} = \mathbf{Y}^j + (\mathbf{a}_0^j + \mathbf{a}_1^j\mathbf{Y}^j)\Delta t + \mathbf{b}_2^j\sqrt{\Delta t}\epsilon_2^j, \tag{C.4b}$$

where ϵ_1^j and ϵ_2^j are independent Gaussian random noises, and Δt is sufficiently small. All the proofs below will be based on this discrete approximation and eventually let $\Delta t \to 0$ to retrieve the continuous dynamics.

C.2 Proofs of the results related to the forward and backward sampling

Lemma C.3. *The cross-covariance between* $\mathbf{Y}^{j+1}$ *and* $\mathbf{Y}^j$*, conditioned on the observations* $\mathbf{X}, s \leqslant J$ *is given by*

$$\mathbf{R}_s^{j+1,j} = \mathbf{R}_s^{j+1}(\mathbf{C}^j)^*, \tag{C.5}$$

where $\mathbf{C}^j$ *is the same as in* (4.22)*.*

Proof. The standard proof starts from writing down the details of the joint distribution of $\mathbf{Y}^{j+1}$ and $\mathbf{Y}^j$ conditioned on the observations $\mathbf{X}, s \leqslant J$, where $j+1 \leqslant J$. Below, a shortcut of proving the result in (C.5) is shown.

Consider the joint distribution

$$p(\mathbf{Y}^j, \mathbf{Y}^{j+1} | \mathbf{X}^s, s \leqslant J) \sim \mathcal{N}\left(\begin{pmatrix} \mu_s^j \\ \mu_s^{j+1} \end{pmatrix}, \begin{pmatrix} \mathbf{R}_s^j & (\mathbf{R}_s^{j+1,j})^* \\ \mathbf{R}_s^{j+1,j} & \mathbf{R}_s^{j+1} \end{pmatrix}\right), \tag{C.6}$$

where the notation $\mathbf{R}_s^{j+1,j}$ denotes the cross-covariance. In light of Lemma C.2, the conditional distribution is given by

$$p(\mathbf{Y}^j | \mathbf{Y}^{j+1}, \mathbf{X}^s, s \leqslant J) = \mu_s^j + \mathbf{R}_s^{j+1,j}(\mathbf{R}_s^{j+1})^{-1}(\mathbf{Y}^{j+1} - \mu_s^{j+1}). \tag{C.7}$$

Recall that the same conditional distribution appears in Lemma 4.7. Matching the coefficients in front of $\mathbf{Y}^{j+1}$ in (C.7) and (4.21a) leads to (C.5). □

C.2.1 Proof of Theorem 4.1

Proof. Consider the filter estimate at the nearby two time instant j and $j+1$,

$$\begin{aligned} p(\mathbf{Y}^{j+1}|\mathbf{X}^s, s \leqslant j+1) &\sim (\boldsymbol{\mu}_{\mathbf{f}}^{j+1}, \mathbf{R}_{\mathbf{f}}^{j+1}), \\ p(\mathbf{Y}^{j}|\mathbf{X}^s, s \leqslant j) &\sim (\boldsymbol{\mu}_{\mathbf{f}}^{j}, \mathbf{R}_{\mathbf{f}}^{j}). \end{aligned} \tag{C.8}$$

As has been shown in Lemma C.3, the cross-covariance is given by

$$\operatorname{cov}(\mathbf{Y}^{j+1}, \mathbf{Y}^{j}) = \mathbf{R}_{\mathbf{f}}^{j+1}(\mathbf{C}^{j})^*. \tag{C.9}$$

Note that since the observations are given up to only $j+1$, the smoother covariance equals to the filter covariance at this end point and therefore $\mathbf{R}_{\mathbf{f}}^{j+1}$ appears on the right hand side of (C.9).

For simplicity of notation, define

$$\begin{aligned} \mathbf{D_1} &= \mathbf{a}_{\mathbf{1}}^{j}\mathbf{R}_{\mathbf{f}}^{j} + \mathbf{R}_{\mathbf{f}}^{j}(\mathbf{a}_{\mathbf{1}}^{j})^* + \mathbf{b}^{j} \circ \mathbf{b}^{j}, \\ \mathbf{D_2} &= -(\mathbf{R}_{\mathbf{f}}^{j}(\mathbf{A}_{\mathbf{1}}^{j})^*)(\mathbf{B}^{j} \circ \mathbf{B}^{j})^{-1}(\mathbf{A}_{\mathbf{1}}^{j}\mathbf{R}_{\mathbf{f}}^{j}). \end{aligned} \tag{C.10}$$

With the two marginal means and covariances (C.8) as well as the cross-covariance (C.9) in hand, applying (C.2) leads to the conditional distribution $p(\mathbf{Y}^{j+1}|\mathbf{Y}^{j}) \sim \mathcal{N}(\mathbf{m}_{\mathbf{f}}^{j}, \mathbf{P}_{\mathbf{f}}^{j})$, where by slightly abusing the notation $\mathbf{Y}^{j+1}$ and $\mathbf{Y}^{j}$ themselves are the filtered dis-

tribution in (C.8). The solutions of $\mathbf{m}_\mathbf{f}^\mathbf{j}$ and $\mathbf{P}_\mathbf{f}^\mathbf{j}$ are as follows,

$$\mathbf{m}_\mathbf{f}^\mathbf{j} = \boldsymbol{\mu}^{\mathbf{j}+1} + \mathbf{R}_\mathbf{f}^{\mathbf{j}+1}(\mathbf{C}^\mathbf{j})^*(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{Y}^\mathbf{j} - \boldsymbol{\mu}^\mathbf{j}), \tag{C.11a}$$

$$\mathbf{P}_\mathbf{f}^\mathbf{j} = \mathbf{R}_\mathbf{f}^{\mathbf{j}+1} - \mathbf{R}_\mathbf{f}^{\mathbf{j}+1}(\mathbf{C}^\mathbf{j})^*(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{C}^\mathbf{j})\mathbf{R}_\mathbf{f}^{\mathbf{j}+1}, \tag{C.11b}$$

where recall that

$$\begin{aligned}
\mathbf{C}^\mathbf{j} &= \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})^* \left(\mathbf{b}^\mathbf{j} \circ \mathbf{b}^\mathbf{j}\Delta\mathrm{t} + (\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})\mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})^*\right)^{-1} \\
&= \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})^* \left[(\mathbf{R}_\mathbf{f}^\mathbf{j})\left((\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\mathbf{b}^\mathbf{j} \circ \mathbf{b}^\mathbf{j}\Delta\mathrm{t} + \mathbf{I} + (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j} + \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^*)\Delta\mathrm{t}\right)\right]^{-1} \\
&= \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})^* \left[(\mathbf{R}_\mathbf{f}^\mathbf{j})\left(\mathbf{I} + (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j} + \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^* + \mathbf{b}^\mathbf{j} \circ \mathbf{b}^\mathbf{j})\Delta\mathrm{t}\right)\right]^{-1} \\
&= \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})^*(\mathbf{I} - (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j} + \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^* + \mathbf{b}^\mathbf{j} \circ \mathbf{b}^\mathbf{j})\Delta\mathrm{t})(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1} \\
&= \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + (\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta\mathrm{t})(\mathbf{I} - (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\mathbf{D}_\mathbf{1}^\mathbf{j}\Delta\mathrm{t})(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}.
\end{aligned} \tag{C.12}$$

Plugging (C.12) back into (C.11b) yields

$$
\begin{aligned}
\mathbf{P}_\mathbf{f}^\mathbf{j} &= \mathbf{R}_\mathbf{f}^{\mathbf{j}+1} - \mathbf{R}_\mathbf{f}^{\mathbf{j}+1}(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{I} - \mathbf{D}_\mathbf{1}^\mathbf{j}(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\Delta t)(\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta t)\mathbf{R}_\mathbf{f}^\mathbf{j} \\
&\qquad\qquad \times (\mathbf{I} + (\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta t)(\mathbf{I} - (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\mathbf{D}_\mathbf{1}^\mathbf{j}\Delta t)(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\mathbf{R}_\mathbf{f}^{\mathbf{j}+1} \\
&= \mathbf{R}_\mathbf{f}^\mathbf{j} + (\mathbf{D}_\mathbf{1}^\mathbf{j} + \mathbf{D}_\mathbf{2}^\mathbf{j})\Delta t - (\mathbf{I} + (\mathbf{D}_\mathbf{1}^\mathbf{j} + \mathbf{D}_\mathbf{2}^\mathbf{j})(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\Delta t)(\mathbf{I} - \mathbf{D}_\mathbf{1}^\mathbf{j}(\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\Delta t) \\
&\qquad \times (\mathbf{I} + \mathbf{a}_\mathbf{1}^\mathbf{j}\Delta t)\mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{I} + (\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta t)(\mathbf{I} - (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}\mathbf{D}_\mathbf{1}^\mathbf{j}\Delta t)(\mathbf{I} + (\mathbf{R}_\mathbf{f}^\mathbf{j})^{-1}(\mathbf{D}_\mathbf{1}^\mathbf{j} + \mathbf{D}_\mathbf{2}^\mathbf{j})\Delta t) \\
&= (\mathbf{D}_\mathbf{1}^\mathbf{j} + \mathbf{D}_\mathbf{2}^\mathbf{j})\Delta t \\
&\qquad - \big((\mathbf{D}_\mathbf{1}^\mathbf{j} + \mathbf{D}_\mathbf{2}^\mathbf{j})\Delta t - \mathbf{D}_\mathbf{1}^\mathbf{j}\Delta t + \mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j}\Delta t + \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta t - \mathbf{D}_\mathbf{1}^\mathbf{j}\Delta t + (\mathbf{D}_\mathbf{1}^\mathbf{j} + \mathbf{D}_\mathbf{2}^\mathbf{j})\Delta t\big) \\
&= \mathbf{D}_\mathbf{1}^\mathbf{j}\Delta t - \mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j}\Delta t - \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta t - \mathbf{D}_\mathbf{2}^\mathbf{j}\Delta t \\
&= \mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j}\Delta t + \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta t + \mathbf{b}^\mathbf{j} \circ \mathbf{b}^\mathbf{j}\Delta t - \mathbf{a}_\mathbf{1}^\mathbf{j}\mathbf{R}_\mathbf{f}^\mathbf{j}\Delta t - \mathbf{R}_\mathbf{f}^\mathbf{j}(\mathbf{a}_\mathbf{1}^\mathbf{j})^*\Delta t - \mathbf{D}_\mathbf{2}^\mathbf{j}\Delta t \\
&= (\mathbf{b}^\mathbf{j} \circ \mathbf{b}^\mathbf{j} - \mathbf{D}_\mathbf{2}^\mathbf{j})\Delta t.
\end{aligned}
\tag{C.13}
$$

Next, plugging (C.12) back into (C.11a) results in

$$\begin{aligned}
\mathbf{m}_{\mathbf{f}}^{j} - \mathbf{Y}^{j} &= \boldsymbol{\mu}_{\mathbf{f}}^{j+1} + \mathbf{R}_{\mathbf{f}}^{j+1}(\mathbf{C}^{j})^{*}(\mathbf{R}_{\mathbf{f}}^{j})^{-1}(\mathbf{Y}^{j} - \boldsymbol{\mu}_{\mathbf{f}}^{j}) - \mathbf{Y}^{j} \\
&= \boldsymbol{\mu}_{\mathbf{f}}^{j+1} \\
&+ (\mathbf{R}_{\mathbf{f}}^{j} + (\mathbf{D}_{\mathbf{1}}^{j} + \mathbf{D}_{\mathbf{2}}^{j})\Delta t)(\mathbf{R}_{\mathbf{f}}^{j})^{-1}(\mathbf{I} - \mathbf{D}_{\mathbf{1}}^{j}(\mathbf{R}_{\mathbf{f}}^{j})^{-1}\Delta t)(\mathbf{I} + \mathbf{a}_{\mathbf{1}}^{j}\Delta t)(\mathbf{Y}^{j} - \boldsymbol{\mu}_{\mathbf{f}}^{j}) - \mathbf{Y}^{j} \\
&= \boldsymbol{\mu}_{\mathbf{f}}^{j+1} \\
&+ \mathbf{Y}^{j} - \boldsymbol{\mu}_{\mathbf{f}}^{j} - \mathbf{Y}^{j} + (-\mathbf{D}_{\mathbf{1}}^{j}(\mathbf{R}_{\mathbf{f}}^{j})^{-1} + \mathbf{a}_{\mathbf{1}}^{j} + \mathbf{D}_{\mathbf{1}}^{j}(\mathbf{R}_{\mathbf{f}}^{j})^{-1} + \mathbf{D}_{\mathbf{2}}^{j}(\mathbf{R}_{\mathbf{f}}^{j})^{-1})(\mathbf{Y}^{j} - \boldsymbol{\mu}_{\mathbf{f}}^{j})\Delta t \\
&= \boldsymbol{\mu}_{\mathbf{f}}^{j+1} - \boldsymbol{\mu}_{\mathbf{f}}^{j} + (\mathbf{a}_{\mathbf{1}}^{j} - (\mathbf{R}_{\mathbf{f}}^{j}(\mathbf{A}_{\mathbf{1}}^{j})^{*})(\mathbf{B}^{j} \circ \mathbf{B}^{j})^{-1}(\mathbf{A}_{\mathbf{1}}^{j}\mathbf{R}_{\mathbf{f}}^{j})(\mathbf{R}_{\mathbf{f}}^{j})^{-1}) \\
&\qquad \times (\mathbf{Y}^{j} - \boldsymbol{\mu}_{\mathbf{f}}^{j})\Delta t.
\end{aligned} \tag{C.14}$$

Collecting the results in (C.13) and (C.14) and taking the limit $\Delta t \to 0$ lead to

$$\begin{aligned}
d\mathbf{Y} = d\boldsymbol{\mu}_{\mathbf{f}} &+ \big(\mathbf{a}_{\mathbf{1}} - (\mathbf{R}_{\mathbf{f}}(\mathbf{A}_{\mathbf{1}})^{*})(\mathbf{B} \circ \mathbf{B})^{-1}(\mathbf{A}_{\mathbf{1}}\mathbf{R}_{\mathbf{f}})\mathbf{R}_{\mathbf{f}}^{-1}\big)(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{f}})dt \\
&+ \big(\mathbf{b} \circ \mathbf{b} + (\mathbf{R}_{\mathbf{f}}(\mathbf{A}_{\mathbf{1}})^{*})(\mathbf{B} \circ \mathbf{B})^{-1}(\mathbf{A}_{\mathbf{1}}\mathbf{R}_{\mathbf{f}})\big)^{1/2} d\mathbf{W}_{\mathbf{Y}}.
\end{aligned} \tag{C.15}$$

□

C.2.2 Proof of Corollary 4.3

Proof. Recall the smoother posterior mean equation (3.6a)

$$\overleftarrow{d\boldsymbol{\mu}_{\mathbf{s}}} = \big(-\mathbf{a}_{\mathbf{0}} - \mathbf{a}_{\mathbf{1}}\boldsymbol{\mu}_{\mathbf{s}} + (\mathbf{b} \circ \mathbf{b})\mathbf{R}_{\mathbf{f}}^{-1}(\boldsymbol{\mu}_{\mathbf{f}} - \boldsymbol{\mu}_{\mathbf{s}})\big)dt,$$

and the backward sampling equation (4.9)

$$\overleftarrow{\mathrm{d}\mathbf{Y}} = \big(-\mathbf{a_0} - \mathbf{a_1}\mathbf{Y}\big)\mathrm{d}t + (\mathbf{b}\circ\mathbf{b})\mathbf{R}_\mathbf{f}^{-1}\big(\boldsymbol{\mu}_\mathbf{f} - \mathbf{Y}\big)\mathrm{d}t + \big(\mathbf{b}\circ\mathbf{b}\big)^{1/2}\mathrm{d}\mathbf{W}_\mathbf{Y}.$$

Take the difference between these two equations yields,

$$\begin{aligned}\overleftarrow{\mathrm{d}\mathbf{Y}} - \overleftarrow{\mathrm{d}\boldsymbol{\mu}_\mathbf{s}} &= -\mathbf{a_1}(\mathbf{Y} - \boldsymbol{\mu}_\mathbf{s})\mathrm{d}t + (\mathbf{b}\circ\mathbf{b})\mathbf{R}_\mathbf{f}^{-1}\big(\boldsymbol{\mu}_\mathbf{s} - \mathbf{Y}\big)\mathrm{d}t + \big(\mathbf{b}\circ\mathbf{b}\big)^{1/2}\mathrm{d}\mathbf{W}_\mathbf{Y} \\ &= -\big(\mathbf{a_1} + (\mathbf{b}\circ\mathbf{b})\mathbf{R}_\mathbf{f}^{-1}\big)(\mathbf{Y} - \boldsymbol{\mu}_\mathbf{s})\mathrm{d}t + \big(\mathbf{b}\circ\mathbf{b}\big)^{1/2}\mathrm{d}\mathbf{W}_\mathbf{Y}.\end{aligned} \tag{C.16}$$

which is (4.10). □

C.2.3 Proof of Corollary 4.4

Proof. Plugging the mean-fluctuation decomposition (C.1) into (C.15) yields,

$$\mathrm{d}\overline{\mathbf{Y}} = \mathrm{d}\boldsymbol{\mu}_\mathbf{f} + \big(\mathbf{a_1} - (\mathbf{R}_\mathbf{f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R}_\mathbf{f})\mathbf{R}_\mathbf{f}^{-1}\big)(\overline{\mathbf{Y}} - \boldsymbol{\mu}_\mathbf{f})\mathrm{d}t. \tag{C.17}$$

If $\overline{\mathbf{Y}}$ at the initial time instant equals $\boldsymbol{\mu}_\mathbf{f}$, then $\overline{\mathbf{Y}} \equiv \boldsymbol{\mu}_\mathbf{f}$ as time evolves. On the other hand, if $\overline{\mathbf{Y}}$ differs from the filter mean $\boldsymbol{\mu}_\mathbf{f}$ at the initial time instant, then $\overline{\mathbf{Y}} \to \boldsymbol{\mu}_\mathbf{f}$. This is because to guarantee the mean stability of $\mathbf{Y}$ in (3.2b), the time average of $\mathbf{a_1}$ is such a matrix that the real part of its eigenvalues should all be negative. On the other hand, $-(\mathbf{R}_\mathbf{f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R}_\mathbf{f})\mathbf{R}_\mathbf{f}^{-1}$ is also a negative definitely matrix.

Next, subtracting (C.17) from (C.15) yields an equation for the fluctuation part

of $\mathbf{Y}$,

$$\begin{aligned} d\mathbf{Y}' = \big(\mathbf{a_1} - (\mathbf{R_f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\mathbf{R_f}^{-1}\big)\mathbf{Y}'dt \\ + \big(\mathbf{b}\circ\mathbf{b} + (\mathbf{R_f}(\mathbf{A_1})^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\big)^{1/2} d\mathbf{W_Y}. \end{aligned} \tag{C.18}$$

The covariance is given by $\overline{\mathbf{Y}'(\mathbf{Y}')^*}$, which can be solved by using the Ito's formula Gardiner (2004)

$$d\overline{\mathbf{Y}'(\mathbf{Y}')^*} = \overline{\mathbf{Y}'d(\mathbf{Y}')^*} + \overline{(d\mathbf{Y}')(\mathbf{Y}')^*} + \overline{d\mathbf{Y}'d(\mathbf{Y}')^*}. \tag{C.19}$$

Then plugging (C.18) into (C.19) yields,

$$d\overline{\mathbf{Y}'(\mathbf{Y}')^*} = \big(\mathbf{a_1}\mathbf{R_f} + \mathbf{R_f}\mathbf{a_1}^* + \mathbf{b}\circ\mathbf{b} - (\mathbf{R_f}\mathbf{A_1}^*)(\mathbf{B}\circ\mathbf{B})^{-1}(\mathbf{A_1}\mathbf{R_f})\big)dt, \tag{C.20}$$

which is exactly the same as the filter covariance equation (3.4b).

□

C.2.4 Proof of Corollary 4.5

Proof. The proof here is similar to that in Appendix C.2.3 in light of the mean-fluctuation decomposition.

Taking the ensemble average of both the left and right hand sides of (4.9) yields

$$\overleftarrow{d\boldsymbol{\mu}_\mathbf{s}} = \big(-\mathbf{a_0} - \mathbf{a_1}\boldsymbol{\mu}_\mathbf{s} + (\mathbf{b}\circ\mathbf{b})\mathbf{R_f}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_\mathbf{s})\big)dt, \tag{C.21}$$

which is (3.6a).

Next, taking the difference between (4.9) and (C.21) yields,

$$\overleftarrow{\mathrm{d}\mathbf{Y}'} = \big(-\mathbf{a_1} - (\mathbf{b}\circ\mathbf{b})\mathbf{R}_\mathbf{f}^{-1}\big)\mathbf{Y}'\mathrm{d}t + \mathbf{b_1}\mathrm{d}\mathbf{W}_{\mathbf{Y},1} + \mathbf{b_2}\mathrm{d}\mathbf{W}_{\mathbf{Y},2}. \tag{C.22}$$

The covariance can be solved by making use of the Ito's formula Gardiner (2004),

$$\overleftarrow{\mathrm{d}\mathbf{R}_\mathbf{s}} = \mathrm{d}\overline{(-\mathbf{Y}'(\mathbf{Y}')^*)} = \overline{(\mathbf{Y}')^*\mathrm{d}(-\mathbf{Y}')} + \overline{\mathbf{Y}'\mathrm{d}(-(\mathbf{Y}')^*)} + \mathrm{d}\overline{(-\mathbf{Y}')\mathrm{d}(-(\mathbf{Y}')^*)}, \tag{C.23}$$

which combining with (C.22) leads to

$$\overleftarrow{\mathrm{d}\mathbf{R}_\mathbf{s}} = -\big((\mathbf{a_1} + (\mathbf{b}\circ\mathbf{b})\mathbf{R}_\mathbf{f}^{-1})\mathbf{R}_\mathbf{s} + \mathbf{R}_\mathbf{s}(\mathbf{a}_1^* + (\mathbf{b}\circ\mathbf{b})\mathbf{R}_\mathbf{f}^{-1}) - \mathbf{b}\circ\mathbf{b}\big)\mathrm{d}t. \tag{C.24}$$

This is (3.6b). □

C.3 Proofs of the results related to the online smoother and sampler

C.3.1 Proof of Lemma 4.7

Proof. Let us start with the joint distribution $p(\mathbf{Y}^j, \mathbf{Y}^{j+1}|\mathbf{X}^s, s \leqslant j)$. The marginal distribution $p(\mathbf{Y}^{j+1}|\mathbf{X}^s, s \leqslant j)$ is given by

$$p(\mathbf{Y}^{j+1}|\mathbf{X}^s, s \leqslant j) \sim \mathcal{N}\Big(\mathbf{a}_\mathbf{0}^j\Delta t + (\mathbf{I} + \mathbf{a}_\mathbf{1}^j\Delta t)\boldsymbol{\mu}_\mathbf{f}, \mathbf{b}^j \circ \mathbf{b}^j\Delta t + (\mathbf{I} + \mathbf{a}_\mathbf{1}^j\Delta t)\mathbf{R}_\mathbf{f}^j(\mathbf{I} + \mathbf{a}_\mathbf{1}^j\Delta t)^*\Big). \tag{C.25}$$

On the other hand, the other marginal distribution $p(\mathbf{Y}^j|\mathbf{X}^s, s \leqslant j)$ is simply given by the filtering formula,

$$p(\mathbf{Y}^j|\mathbf{X}^s, s \leqslant j) \sim \mathcal{N}(\boldsymbol{\mu}_f^j, \mathbf{R}_f^j). \tag{C.26}$$

The cross covariance term is given by

$$\langle \mathbf{Y}'^{,j+1}(\mathbf{Y}'^{,j})^* \rangle = (\mathbf{I} + \mathbf{a}_1^j \Delta t)\mathbf{R}_f^j, \tag{C.27}$$

where $\mathbf{Y}'^{,j+1}$ and $\mathbf{Y}'^{,j}$ are $\mathbf{Y}^{j+1}$ and $\mathbf{Y}^j$ by removing their mean values. Therefore, collecting (C.25)–(C.27) leads to

$$p(\mathbf{Y}^j, \mathbf{Y}^{j+1}|\mathbf{X}^s, s \leqslant j) \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_i, \tilde{\mathbf{R}}_i\right), \tag{C.28}$$

where

$$\tilde{\boldsymbol{\mu}}_i = \begin{pmatrix} \boldsymbol{\mu}_f^j \\ \mathbf{a}_0^j \Delta t + (\mathbf{I} + \mathbf{a}_1^j \Delta t)\boldsymbol{\mu}_f^j \end{pmatrix}, \tag{C.29}$$

and

$$\tilde{\mathbf{R}}_i = \begin{pmatrix} \mathbf{R}_f^j & \mathbf{R}_f^j(1 + \mathbf{a}_1^j \Delta t)^* \\ (\mathbf{I} + \mathbf{a}_1^j \Delta t)\mathbf{R}_f^j & \mathbf{b}^j \circ \mathbf{b}^j \Delta t + (\mathbf{I} + \mathbf{a}_1^j \Delta t)\mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j \Delta t)^* \end{pmatrix}. \tag{C.30}$$

In light of Lemma C.2, the result in (C.28) yields the conditional distribution,

$$p(\mathbf{Y}^j|\mathbf{Y}^{j+1}, \mathbf{X}^s, s \leqslant J) = p(\mathbf{Y}^j|\mathbf{Y}^{j+1}, \mathbf{X}^s, s \leqslant j) = \mathcal{N}(\mathbf{m}^j, \mathbf{P}^j), \tag{C.31}$$

where

$$\mathbf{m}^j = \boldsymbol{\mu}_f^j + \mathbf{C}^j\big(\mathbf{Y}^{j+1} - \mathbf{a}_0^j \Delta t - (\mathbf{I} + \mathbf{a}_1^j \Delta t)\boldsymbol{\mu}_f^j\big), \tag{C.32a}$$

$$\mathbf{P}^j = \mathbf{R}_f^j - \mathbf{C}^j\big(\mathbf{b}^j \circ \mathbf{b}^j \Delta t + (\mathbf{I} + \ \mathbf{a}_1^j \Delta t)\mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j \Delta t)^*\big)(\mathbf{C}^j)^*, \tag{C.32b}$$

and the auxiliary matrix $\mathbf{C}$ is given by

$$\mathbf{C}^j = \mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j \Delta t)^*\big(\mathbf{b}^j \circ \mathbf{b}^j \Delta t + (\mathbf{I} + \mathbf{a}_1^j \Delta t)\mathbf{R}_f^j(\mathbf{I} + \mathbf{a}_1^j \Delta t)^*\big)^{-1}. \tag{C.33}$$

Note that the first equality in (C.31) is due to the Markovian property of the underlying system Särkkä (2013). In fact, if $\mathbf{Y}^{j+1}$ is known, then the conditional distribution of $\mathbf{Y}^j$ has no dependence on $\mathbf{X}^s, s \geqslant j+1$. More specifically, the information of $\mathbf{X}^s, s > j$ has been included in $\mathbf{Y}^{j+1}$. The conditional terms $\mathbf{Y}^{j+1}$ and $\mathbf{X}^s, s \leqslant j$ represent the information in the future and past, respectively, which can be seen in (C.32). This finishes the proof of Lemma 4.7. □

In light of the Lemma 4.7, an alternative way of calculating the optimal nonlinear smoother is given as follows.

Lemma C.4 (An alternative way of calculating the Optimal Nonlinear Smoother)**.** *Given one realization of the observed variable* $\{\mathbf{X}^0, \ldots, \mathbf{X}^J\}$*, the nonlinear optimal smoother estimate* $p(\mathbf{Y}^j | \mathbf{X}^s, 0 \leqslant s \leqslant J)$ *is conditional Gaussian,*

$$p(\mathbf{Y}^j | \mathbf{X}^s, 0 \leqslant s \leqslant J) \sim \mathcal{N}(\boldsymbol{\mu}_s^j, \mathbf{R}_s^j), \tag{C.34}$$

where the conditional mean $\boldsymbol{\mu}_{\mathbf{s}}^{j}$ *and conditional covariance* $\mathbf{R}_{\mathbf{s}}^{j}$ *of the smoother are given by*

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{s}}^{j} &= \boldsymbol{\mu}_{\mathbf{f}}^{j} + \mathbf{C}^{j}(\boldsymbol{\mu}_{\mathbf{s}}^{j+1} - \mathbf{a}_{0}^{j}\Delta t - (\mathbf{I} + \mathbf{a}_{1}^{j}\Delta t)\boldsymbol{\mu}_{\mathbf{f}}^{j}), \\
\mathbf{R}_{\mathbf{s}}^{j} &= \mathbf{R}_{\mathbf{f}}^{j} + \mathbf{C}^{j}(\mathbf{R}_{\mathbf{s}}^{j+1} - (\mathbf{I} + \mathbf{a}_{1}^{j}\Delta t)\mathbf{R}_{\mathbf{f}}^{j}(\mathbf{I} + \mathbf{a}_{1}^{j}\Delta t)^{*} - \mathbf{b}^{j} \circ \mathbf{b}^{j}\Delta t)(\mathbf{C}^{j})^{*},
\end{aligned} \tag{C.35}$$

where the auxiliary matrix $\mathbf{C}^{j}$ *is the same as in* (4.22)*. The optimal smoother is calculated backwards from* $s = J$ *to* $s = 0$*. The starting value of the smoother estimate* $(\boldsymbol{\mu}_{\mathbf{s}}^{J}, \mathbf{R}_{\mathbf{s}}^{J})$ *is the same as the filter estimate at the endpoint* $(\boldsymbol{\mu}_{\mathbf{f}}^{J}, \mathbf{R}_{\mathbf{f}}^{J})$*.*

C.3.2 Proof of Theorem 4.8

Proof. For any $n \geqslant 1$ and $0 \leqslant j \leqslant n-1$, the following equations are valid

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{s}}^{n,j} &= \mathbf{D}^{n-1,j}\boldsymbol{\mu}_{s}^{n,n} + \mathbf{m}_{1}^{j} + \sum_{m=j+1}^{n-1} \mathbf{D}^{m-1,j}\mathbf{m}_{1}^{m}, \\
\mathbf{R}_{\mathbf{s}}^{n,j} &= \mathbf{D}^{n-1,j}\mathbf{R}_{s}^{n,n}(\mathbf{D}^{n-1,j})^{*} + \mathbf{P}^{j} + \sum_{m=j+1}^{n-1} \mathbf{D}^{m-1,j}\mathbf{P}^{m}(\mathbf{D}^{m-1,j})^{*},
\end{aligned} \tag{C.36}$$

where $\boldsymbol{\mu}_{\mathbf{s}}^{n,j}$ and $\mathbf{R}_{\mathbf{s}}^{n,j}$ denote the smoother mean and smoother covariance at time $t = t_j$ based on one realization of trajectory of $\mathbf{X}$ up to $t = t_n$. In fact, the mean and covariance at two nearby time instants satisfy similar relationship as in (C.35). Thus, the equations (C.36) can be proved by showing

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{s}}^{n,j} &= \boldsymbol{\mu}_{\mathbf{f}}^{j} + \mathbf{C}^{j}(\boldsymbol{\mu}_{\mathbf{s}}^{n,j+1} - \mathbf{a}_{0}^{j}\Delta t - (\mathbf{I} + \mathbf{a}_{1}^{j}\Delta t)\boldsymbol{\mu}_{\mathbf{f}}^{j}), \\
\mathbf{R}_{\mathbf{s}}^{n,j} &= \mathbf{R}_{\mathbf{f}}^{j} + \mathbf{C}^{j}(\mathbf{R}_{\mathbf{s}}^{n,j+1} - (\mathbf{I} + \mathbf{a}_{1}^{j}\Delta t)\mathbf{R}_{\mathbf{f}}^{j}(\mathbf{I} + \mathbf{a}_{1}^{j}\Delta t)^{*} - \mathbf{b}^{j} \circ \mathbf{b}^{j}\Delta t)(\mathbf{C}^{j})^{*},
\end{aligned} \tag{C.37}$$

for $n \geqslant 1$ and $0 \leqslant j \leqslant n-1$. Denote $\mathbf{m}_0^j$, $\mathbf{m}_1^j$, and $\mathbf{P}^j$ as follows

$$\begin{aligned} \mathbf{m}_0^j &= \mathbf{C}^j, \qquad \mathbf{m}_1^j = \boldsymbol{\mu}_\mathbf{f}^j - \mathbf{C}^j(\mathbf{a}_\mathbf{0}^j \Delta t + (\mathbf{I} + \mathbf{a}_\mathbf{1}^j \Delta t)\boldsymbol{\mu}_\mathbf{f}^j), \\ \mathbf{P}^j &= \mathbf{R}_\mathbf{f}^j - \mathbf{C}^j(\mathbf{b}^j \circ \mathbf{b}^j \Delta t + (\mathbf{I} + \mathbf{a}_\mathbf{1}^j \Delta t)\mathbf{R}_\mathbf{f}^j(\mathbf{I} + \mathbf{a}_\mathbf{1}^j \Delta t)^*)(\mathbf{C}^j)^*, \end{aligned} \tag{C.38}$$

and the auxiliary matrix $\mathbf{C}^j$ is the same as in (4.22). Therefore, the right hand side of equation (C.37) can be written as

$$\begin{aligned} \boldsymbol{\mu}_\mathbf{s}^{n,j} &= \mathbf{m}_0^j \boldsymbol{\mu}_\mathbf{s}^{n,j+1} + \mathbf{m}_1^j, \\ \mathbf{R}_\mathbf{s}^{n,j} &= \mathbf{P}^j + \mathbf{m}_0^j \mathbf{R}_\mathbf{s}^{n,j+1} (\mathbf{m}_0^j)^*. \end{aligned} \tag{C.39}$$

Note that for any $n \geqslant 1$ and $0 \leqslant j \leqslant n-1$,

$$\mathbf{m}_0^j \mathbf{D}^{n-1,j+1} = \mathbf{D}^{n-1,j}. \tag{C.40}$$

This is because for $n \geqslant 2$ and $0 \leqslant j \leqslant n-2$, from the third equation of (4.25), it follows

$$\mathbf{D}^{n-1,j} = \mathbf{D}^{n-2,j}\mathbf{m}_0^{n-1} = \mathbf{D}^{n-3,j}\mathbf{m}_0^{n-2}\mathbf{m}_0^{n-1} = \cdots = \mathbf{D}^{j,j}\mathbf{m}_0^{j+1}\ldots\mathbf{m}_0^{n-1} = \prod_{i=j}^{n-1} \mathbf{m}^i. \tag{C.41}$$

Similarly, the left hand side of (C.40) becomes

$$\mathbf{m}_0^j \mathbf{D}^{n-1,j+1} = \mathbf{m}_0^j \prod_{i=j+1}^{n-1} \mathbf{m}_0^i = \prod_{i=j}^{n-1} \mathbf{m}^i, \tag{C.42}$$

which is the same as the right hand side of equation (C.40). When $n \geqslant 2$ and $j = n - 1$, the equation (C.40) becomes

$$\mathbf{m}_0^j \mathbf{D}^{n-1,j+1} = \mathbf{m}_0^{n-1} \mathbf{D}^{n-1,n} = \mathbf{m}_0^{n-1} = \mathbf{D}^{n-1,n-1} = \mathbf{D}^{n-1,j},$$

and when $n = 1$ and $j = 0$, it is

$$\mathbf{m}_0^j \mathbf{D}^{n-1,j+1} = \mathbf{m}_0^0 \mathbf{D}^{0,1} = \mathbf{m}_0^0 = \mathbf{D}^{0,0} = \mathbf{D}^{n-1,j}.$$

Plugging the expression of $\boldsymbol{\mu}_s^{n,j+1}$ in (C.36) into the right hand side of the first equation of (C.39) yields

$$\begin{aligned}
\mathbf{m}_0^j \boldsymbol{\mu}_s^{n,j+1} + \mathbf{m}_1^j =& \mathbf{m}_0^j \mathbf{D}^{n-1,j+1} \boldsymbol{\mu}_s^{n,n} + \mathbf{m}_0^j \mathbf{m}_1^{j+1} + \sum_{m=j+2}^{n-1} \mathbf{m}_0^j \mathbf{D}^{m-1,j+1} \mathbf{m}_1^m + \mathbf{m}_1^j \\
=& \mathbf{D}^{n-1,j} \boldsymbol{\mu}_s^{n,n} + \mathbf{m}_0^j \mathbf{m}_1^{j+1} + \sum_{m=j+2}^{n-1} \mathbf{D}^{m-1,j} \mathbf{m}_1^m + \mathbf{m}_1^j \\
=& \mathbf{D}^{n-1,j} \boldsymbol{\mu}_s^{n,n} + \sum_{m=j+1}^{n-1} \mathbf{D}^{m-1,j} \mathbf{m}_1^m + \mathbf{m}_1^j = \boldsymbol{\mu}_s^{n,j},
\end{aligned} \tag{C.43}$$

where the second equality is due to (C.40) and the third one is by $\mathbf{D}^{j,j} = \mathbf{m}_0^j$. Likewise, plugging $\mathbf{R}_s^{n,j+1}$ in (C.36) into the second equation of (C.39) yields

$$\begin{aligned}\mathbf{P}^j + \mathbf{m}_0^j \mathbf{R}_s^{n,j+1} (\mathbf{m}_0^j)^* =& \mathbf{P}^j + \mathbf{m}_0^j \mathbf{D}^{n-1,j+1} (\mathbf{R}_s^{n,n}) (\mathbf{D}_0^{n-1,j+1})^* (\mathbf{m}_0^j)^* \\ &+ \sum_{m=j+2}^{n-1} \mathbf{m}_0^j \mathbf{D}^{m-1,j+1} \mathbf{P}^m (\mathbf{D}^{m-1,j+1})^* (\mathbf{m}_0^j)^* + \mathbf{m}_0^j \mathbf{P}^{j+1} (\mathbf{m}_0^j)^* \\ =& \mathbf{P}^j + \mathbf{D}^{n-1,j} \mathbf{R}_s^{n,n} (\mathbf{D}^{n-1,j})^* + \sum_{m=j+1}^{n-1} \mathbf{D}^{m-1,j} \mathbf{P}^m (\mathbf{D}^{m-1,j})^* \\ =& \mathbf{R}_s^{n,j},\end{aligned} \tag{C.44}$$

where the second equality is due to (C.40) and $\mathbf{D}^{j,j} = \mathbf{m}_0^j$.

Next, the recursive formulae (4.24) can be verified by showing the difference between $\boldsymbol{\mu}_s^{n,j}$ and $\boldsymbol{\mu}_s^{n-1,j}$ and that between $\mathbf{R}_s^{n,j}$ and $\mathbf{R}_s^{n-1,j}$ are as follows

$$\begin{aligned}\boldsymbol{\mu}_s^{n,j} &= \mathbf{D}^{n-1,j} \boldsymbol{\mu}_s^{n,n} + \mathbf{m}_1^j + \sum_{m=j+1}^{n-1} \mathbf{D}^{m-1,j} \mathbf{m}_1^m \\ &= \mathbf{D}^{n-1,j} \boldsymbol{\mu}_s^{n,n} + \mathbf{m}_1^j + \sum_{m=j+1}^{n-2} \mathbf{D}^{m-1,j} \mathbf{m}_1^m + \mathbf{D}^{n-2,j} \mathbf{m}_1^{n-1} \\ &= \boldsymbol{\mu}_s^{n-1,j} - \mathbf{D}^{n-2,j} \boldsymbol{\mu}_s^{n-1,n-1} + \mathbf{D}^{n-1,j} \boldsymbol{\mu}_s^{n,n} + \mathbf{D}^{n-2,j} \mathbf{m}_1^{n-1},\end{aligned} \tag{C.45}$$

and

$$\begin{aligned}
\mathbf{R}_{\mathbf{s}}^{n,j} =& \mathbf{D}^{n-1,j}\mathbf{R}_{\mathbf{s}}^{n,n}(\mathbf{D}^{n-1,j})^* + \mathbf{P}^j + \sum_{m=j+1}^{n-1} \mathbf{D}^{m-1,j}\mathbf{P}^m(\mathbf{D}^{m-1,j})^* \\
=& \mathbf{D}^{n-1,j}\mathbf{R}_{\mathbf{s}}^{n,n}(\mathbf{D}^{n-1,j})^* + \mathbf{P}^j + \sum_{m=j+1}^{n-2} \mathbf{D}^{m-1,j}\mathbf{P}^m(\mathbf{D}^{m-1,j})^* \\
&+ \mathbf{D}^{n-2,j}\mathbf{P}^{n-1}(\mathbf{D}^{n-2,j})^* \\
=& \mathbf{R}_{\mathbf{s}}^{n-1,j} - \mathbf{D}^{n-2,j}\mathbf{R}_{\mathbf{s}}^{n-1,n-1}(\mathbf{D}^{n-2,j})^* \\
&+ \mathbf{D}^{n-1,j}\mathbf{R}_{\mathbf{s}}^{n,n}(\mathbf{D}^{n-1,j})^* + \mathbf{D}^{n-2,j}\mathbf{P}^{n-1}(\mathbf{D}^{n-2,j})^*.
\end{aligned} \tag{C.46}$$

□

C.3.3 Proof of Theorem 4.9

Proof. By Theorem 4.10, the online smoother update is given by

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{s}}^{n,j} &= \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j} - \mathbf{D}^{n-2,j}\boldsymbol{\mu}_{s}^{n-1,n-1} + \mathbf{D}^{n-1,j}\boldsymbol{\mu}_{\mathbf{s}}^{n,n} + \mathbf{D}^{n-2,j}\mathbf{m}_1^{n-1}, \\
\mathbf{R}_{\mathbf{s}}^{n,j} &= \mathbf{R}_{\mathbf{s}}^{n-1,j} - \mathbf{D}^{n-2,j}\mathbf{R}_{s}^{n-1,n-1}(\mathbf{D}^{n-2,j})^* \\
&\quad + \mathbf{D}^{n-1,j}\mathbf{R}_{s}^{n,n}(\mathbf{D}^{n-1,j})^* + \mathbf{D}^{n-2,j}\mathbf{P}^{n-1}(\mathbf{D}^{n-2,j})^*.
\end{aligned} \tag{C.47}$$

Replacing the coefficient $\mathbf{D}^{n-1,j}$ by $\mathbf{D}^{n-2,j}\mathbf{m}_0^{n-1}$ using the coefficient update formulae (4.25), the mean equation in (C.47) becomes

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{s}}^{n,j} &= \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\mathbf{m}_0^{n-1}\boldsymbol{\mu}_{\mathbf{s}}^{n,n} + \mathbf{m}_1^{n-1} - \boldsymbol{\mu}_{\mathbf{s}}^{n-1,n-1}) \\
&= \boldsymbol{\mu}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\boldsymbol{\mu}_{\mathbf{s}}^{n,n-1} - \boldsymbol{\mu}_{\mathbf{s}}^{n-1,n-1}),
\end{aligned} \tag{C.48}$$

where the second equality is obtained by using the first equation of (C.39). Similarly, the covariance can be written as

$$\begin{aligned}\mathbf{R}_{\mathbf{s}}^{n,j} &= \mathbf{R}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\mathbf{m}_0^{n-1}\mathbf{R}_{\mathbf{s}}^{n,n}(\mathbf{m}_0^{n-1})^* + \mathbf{P}^{n-1} - \mathbf{R}_{\mathbf{s}}^{n-1,n-1})(\mathbf{D}^{n-2,j})^* \\ &= \mathbf{R}_{\mathbf{s}}^{n-1,j} + \mathbf{D}^{n-2,j}(\mathbf{R}_{\mathbf{s}}^{n,n-1} - \mathbf{R}_{\mathbf{s}}^{n-1,n-1})(\mathbf{D}^{n-2,j})^*,\end{aligned} \tag{C.49}$$

where the second equality is due to the second equation of (C.39). □

C.3.4 Proof of Proposition 4.4

Proof. Recall the auxiliary matrix $\mathbf{C}^j$ in form (C.12)

$$\mathbf{C}^j = \mathbf{R}_{\mathbf{f}}^j(\mathbf{I} + (\mathbf{a}_1^j)^*\Delta t)(\mathbf{I} - (\mathbf{R}_{\mathbf{f}}^j)^{-1}\mathbf{D}_1^j\Delta t)(\mathbf{R}_{\mathbf{f}}^j)^{-1}, \tag{C.50}$$

where

$$\mathbf{D}_1^j = \mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j + \mathbf{R}_{\mathbf{f}}^j(\mathbf{a}_1^j)^* + \mathbf{b}^j \circ \mathbf{b}^j. \tag{C.51}$$

By rearranging terms in $\mathbf{C}^j$, it obtains

$$\begin{aligned}\mathbf{C}^j =& \mathbf{R}_{\mathbf{f}}^j(\mathbf{I} + (\mathbf{a}_1^j)^*\Delta t - (\mathbf{R}_{\mathbf{f}}^j)^{-1}(\mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j + \mathbf{R}_{\mathbf{f}}^j(\mathbf{a}_1^j)^* + \mathbf{b}^j \circ \mathbf{b}^j)\Delta t)(\mathbf{R}_{\mathbf{f}}^j)^{-1} \\ =& \big(\mathbf{R}_{\mathbf{f}}^j - \mathbf{R}_{\mathbf{f}}^j(\mathbf{a}_1^j)^*\Delta t - (\mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j + \mathbf{R}_{\mathbf{f}}^j(\mathbf{a}_1^j)^* + \mathbf{b}^j \circ \mathbf{b}^j)\Delta t\big)(\mathbf{R}_{\mathbf{f}}^j)^{-1} \\ =& (\mathbf{R}_{\mathbf{f}}^j - \mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j\Delta t - \mathbf{b}^j \circ \mathbf{b}^j\Delta t)(\mathbf{R}_{\mathbf{f}}^j)^{-1} \\ =& \mathbf{I} - (\mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j\Delta t + \mathbf{b}^j \circ \mathbf{b}^j\Delta t)(\mathbf{R}_{\mathbf{f}}^j)^{-1}.\end{aligned} \tag{C.52}$$

If $\mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j + \mathbf{b}^j \circ \mathbf{b}^j$ is positive definite, so is $(\mathbf{a}_1^j\mathbf{R}_{\mathbf{f}}^j\Delta t + \mathbf{b}^j \circ \mathbf{b}^j\Delta t)(\mathbf{R}_{\mathbf{f}}^j)^{-1}$ since $(\mathbf{R}_{\mathbf{f}}^j)^{-1}$ is also a positive definite matrix. Thus, the spectral radius of $\mathbf{C}^j$ is less than 1. □

C.3.5 Proof of Theorem 4.10

Proof. Given the results from the backward smoothing (C.6) with in total n available observations,

$$p(\mathbf{Y}^{j+1}, \mathbf{Y}^{j} | \mathbf{X} \leqslant n) \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{\mathbf{s}}^{n,j+1} \\ \boldsymbol{\mu}_{\mathbf{s}}^{n,j} \end{pmatrix}, \begin{pmatrix} \mathbf{R}_{\mathbf{s}}^{n,j+1} & \mathbf{R}_{\mathbf{s}}^{n,j+1}(\mathbf{C}^{j})^{*} \\ \mathbf{C}^{j}\mathbf{R}_{\mathbf{s}}^{n,j+1} & \mathbf{R}_{\mathbf{s}}^{n,j} \end{pmatrix} \right), \tag{C.53}$$

the conditional distribution is as follows

$$p(\mathbf{Y}^{j+1} | \mathbf{Y}^{j}, \mathbf{X}^{s} \leqslant n) = \mathcal{N}(\mathbf{m}_{\mathbf{s}}^{n,j}, \mathbf{P}_{\mathbf{s}}^{n,j}), \qquad j \leqslant n-1, \tag{C.54}$$

where the expression of $\mathbf{m}_{\mathbf{s}}^{n,j}$ and $\mathbf{P}_{\mathbf{s}}^{n,j}$ are exactly (4.32). □

C.4 Examples: the behavior of $\mathbf{C}^{j}$ in affecting the online smoother impact regions

It has been shown in Section 4.4.1 that the spectral radius of the auxiliary matrix $\mathbf{C}^{j}$ in (4.22) plays an important role in affecting the impact regions of the online smoother. Here two examples are used to address this issue.

A two dimensional linear Gaussian model Consider the following two dimensional linear Gaussian model

$$\begin{aligned} dx &= (a_{11}x + a_{12}y)dt + \sigma_1 dW_1, \\ dy &= (a_{21}x + a_{22}y)dt + \sigma_2 dW_2, \end{aligned} \tag{C.55}$$

where a_{11}, a_{12}, a_{21}, and a_{22} are constants, and x and y are the observed and unobserved variable, respectively. Here a_{22} needs to be negative in order to guarantee the existence of the statistical equilibrium state of the coupled system. Since the coefficients are constants, the equilibrium solution of the filter covariance R_{eq} can thus be solved via the following steady state Riccati equation (3.4b)

$$2a_{22}R_{eq} + \sigma_2^2 = \frac{(R_{eq}a_{12})^2}{\sigma_1^2}, \tag{C.56}$$

which gives

$$R_{eq} = \frac{a_{22}\sigma_1^2 + \sigma_1\sqrt{\Delta}}{a_{12}^2}, \tag{C.57}$$

where $\Delta = a_{22}^2\sigma_1^2 + a_{12}^2\sigma_2^2$. Plugging (C.57) into the assumption $\mathbf{a}_1^j\mathbf{R}_f^j + \mathbf{b}^j \circ \mathbf{b}^j$, we find the assumption in Proposition 4.4 yields

$$\begin{aligned}\mathbf{a}_1^j\mathbf{R}_f^j + \mathbf{b}^j \circ \mathbf{b}^j &= a_{22}\frac{a_{22}\sigma_1^2 + \sigma_1\sqrt{\Delta}}{a_{12}^2} + \sigma_2^2 \\ &= \frac{a_{22}^2\sigma_1^2 + a_{22}\sigma_1\sqrt{\Delta} + a_{12}^2\sigma_2^2}{a_{12}^2} \\ &= \frac{\Delta + a_{22}\sigma_1\sqrt{\Delta}}{a_{12}^2} \\ &= \frac{\sqrt{\Delta}(\sqrt{\Delta} + a_{22}\sigma_1)}{a_{12}^2} > 0.\end{aligned} \tag{C.58}$$

The last inequality is due to $\sqrt{\Delta} + a_{22}\sigma_1 = \sqrt{a_{22}^2\sigma_1^2 + a_{12}^2\sigma_2^2} + a_{22}\sigma_1 > 0$. This implies that the auxiliary constant $\mathbf{C}^j$ is always less than 1.

A nonlinear dyad model Here, a nonlinear dyad model is used to understand the spectral radius of $\mathbf{C}^j$ at both quiescent and intermittent phases. Recall the physics-constrained nonlinear dyad model in (4.33) with one observed variable u and one unobserved variable v, which is repeated as follows,

$$du = \big((-d_u + cv)u + f_u\big)dt + \sigma_u dW_u, \tag{C.59a}$$

$$dv = (-d_v v - cu^2 + f_v)dt + \sigma_v dW_v. \tag{C.59b}$$

Define two dynamical regimes by controlling the values of the noise as follows (other parameters are the same as in (4.34)).

Gaussian regime:

$$\sigma_u = 2, \qquad \sigma_v = 0.5,$$

Intermittency regime:

$$\sigma_u = 0.5, \qquad \sigma_v = 2.$$

Figure C.1 shows realizations of the model trajectories of u and v as well as the associated $\mathbf{C}^j$. The first column shows the results in the Gaussian regime while the second column illustrates those in the intermittency regime. It is clear that in the Gaussian regime, the matrix $\mathbf{C}^j$ stays below the threshold value $\mathbf{C}^j = 1$ when the system reaches its equilibrium state. On the other hand, $\mathbf{C}^j$ changes significantly as a function of time j in the intermittency regime, depending strongly on the observations. The value of $\mathbf{C}^j$ is smaller at the phases that extreme events appear in the observational time series u. In fact, the signal-to-noise ratio in these intermittent phase is large, implying a relatively smaller uncertainty in the observational process than that in the hidden one, which has been discussed in Section 4.4.1. In such a situation, the impact region will be a shorter time interval. Nevertheless, the amount of the update at intermittent phases is significant and overall the extreme events in the observational processes greatly reduces the uncertainty in the recovered hidden states within a finite neighborhood.

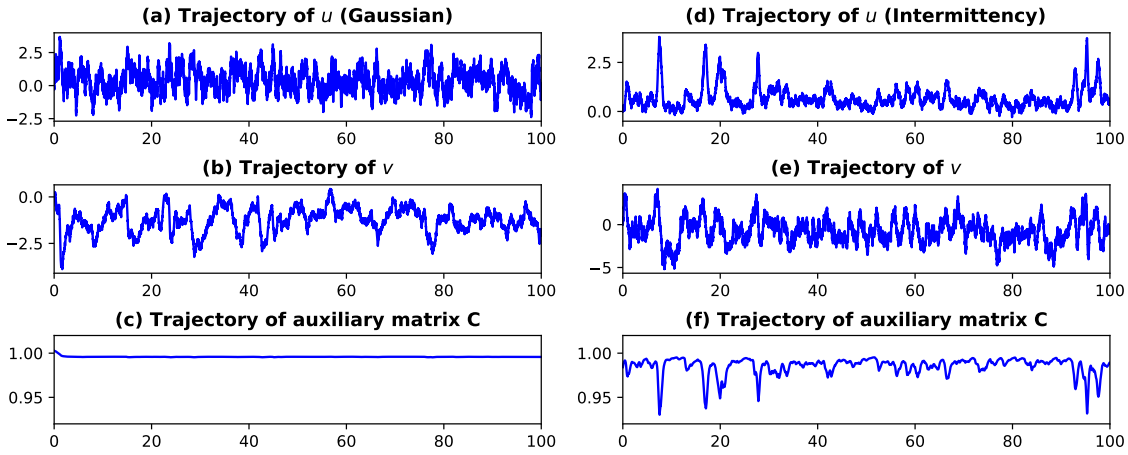

Figure C.1: Panel (a)-(c): the trajectories of observations u, hidden variable v, and auxiliary matrix $\mathbf{C}$ in Gaussian regime. Panel (d)-(f): the trajectories of observations u, hidden variable v, and auxiliary matrix $\mathbf{C}$ in intermittency regime.

C.5 Details of the EM algorithm for parameter estimation

Consider a slightly simpler version of conditional Gaussian nonlinear model (3.2) with $\mathbf{B}_2(t)$ and $\mathbf{b}_1(t)$ being zero matrices. Let us write the discrete approximation of the original continuous system using the Euler-Maruyama scheme,

$$\mathbf{X}^{j+1} = \mathbf{X}^j + (\mathbf{A}_0^j(\mathbf{X}, t; \theta) + \mathbf{A}_1^j(\mathbf{X}, t; \theta)\mathbf{Y}^j)\Delta t + \mathbf{B}_1^j(\mathbf{X}, t; \theta)\sqrt{\Delta t}\,\varepsilon_1^j, \tag{C.60a}$$

$$\mathbf{Y}^{j+1} = \mathbf{Y}^j + (\mathbf{a}_0^j(\mathbf{X}, t; \theta) + \mathbf{a}_1^j(\mathbf{X}, t; \theta)\mathbf{Y}^j)\Delta t + \mathbf{b}_2^j(\mathbf{X}, t; \theta)\sqrt{\Delta t}\,\varepsilon_2^j, \tag{C.60b}$$

where ε_1^j and ε_2^j are independent and identically distributed Gaussian white noises. Assume all the parameters appear as multiplicative prefactors of some functions of $\mathbf{X}^j$ and $\mathbf{Y}^j$ on the right hand side of (C.60). Thus, the log likelihood function of

$p(\mathbf{X}, \mathbf{Y}|\theta)$ can be solved explicitly.

For (C.60), the local linear Gaussian approximation on the right hand side is

$$\mathcal{N}(\mu^j, \mathbf{R}^j) = \widetilde{C}|\mathbf{R}^j|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{Z}^{j+1} - \mu^j)^T(\mathbf{R}^j)^{-1}(\mathbf{Z}^{j+1} - \mu^j)\right), \tag{C.61}$$

where $\mathbf{Z}^{j+1} = (\mathbf{X}^{j+1}, \mathbf{Y}^{j+1})^T$ and $\mu^j = \mathbf{M}^j\xi + \mathbf{S}^j$. Here ξ is the parameters in drift part (C.60) and $\mathbf{S}^j$ is those terms that do not involve parameters such as the first terms $\mathbf{X}^j$ or $\mathbf{Y}^j$ in (C.60). The covariance $\mathbf{R}^j$ is a block diagonal matrix with entries $(\mathbf{B}_1^j(\mathbf{X}, t))(\mathbf{B}_1^j(\mathbf{X}, t))^*\Delta t$ and $(\mathbf{b}_2^j(\mathbf{X}, t))(\mathbf{b}_2^j(\mathbf{X}, t))^*\Delta t$, which has an one-to-one correspondence with the parameters in the diffusion terms. The constant $\widetilde{C}$ is due to the normalization of a Gaussian distribution. Since the states $\mathbf{Y}$ is unobserved and it contains uncertainty, an expectation of the log-likelihood function as in (3.25) is adopted, and the overall objective function becomes

$$\min_\theta \mathcal{L} = \min_\theta \left(\frac{1}{2}\sum_j \overline{(\mathbf{Z}^{j+1} - \mathbf{M}^j\xi - \mathbf{S}^j)^*(\mathbf{R})^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\xi - \mathbf{S}^j)} + \frac{J}{2}\log|\mathbf{R}|\right), \tag{C.62}$$

where $\theta = \{\xi, \mathbf{R}\}$, J is the total numbers of the terms in previous sum, and $\bar{\cdot}$ denotes the expectation over the uncertain component of $\mathbf{Z}^j$, namely $\mathbf{Y}^j$, at fixed j while the expectation of the deterministic component $\mathbf{X}^j$ is simply itself.

The online smoother estimates in Section 4.4 allow us to derive an online EM algorithm using sequential data. In order to avoid blowup issue due to the lack of observations, a small number of observations, $\{\mathbf{X}^0, \ldots, \mathbf{X}^{T_0}\}$, exist before the EM procedure. Denote $\mathbf{X}^{i:j} = \{\mathbf{X}^i, \ldots, \mathbf{X}^j\}$ for $i \leqslant j$, the online EM algorithm is given in

Algorithm 6.

Algorithm 6: An online EM algorithm for parameter estimation

Set an initial guess θ^0;

for $k = 1$ *to converge* **do**

Use Algorithm 3 to compute the conditional distribution

$p(\mathbf{Y}^{0:T_0+k}|\mathbf{X}^{0:T_0+k}, \theta_{k-1})$;

M-step: update the parameters θ_k^M where

$\theta_k^M = \arg\max_{\theta^M} \widetilde{Q}(\theta^M; \theta_{k-1}^M)$;

Compute

$$\left(\frac{1}{2}\sum_{j=0}^{T_0+k-1} \overline{(\mathbf{Z}^{j+1} - \mathbf{M}^j\xi - \mathbf{C}^j)^*(\mathbf{R})^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\xi - \mathbf{C}^j)} + \frac{T_0+k-1}{2}\log|\mathbf{R}|\right);$$

Update $\theta_k = \text{argmax}_\theta Q(\theta; \theta_{k-1})$;

REFERENCES

Adrian, RJ, KT Christensen, and Z-C Liu. 2000. Analysis and interpretation of instantaneous turbulent velocity fields. *Experiments in fluids* 29(3):275–290.

Agapiou, Sergios, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and AM Stuart. 2017. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science* 405–431.

Ahmed, Shady E, Suraj Pawar, Omer San, Adil Rasheed, Traian Iliescu, and Bernd R Noack. 2021. On closures for reduced order models–A spectrum of first-principle to machine-learned avenues. *Physics of Fluids* 33(9):091301.

Allen, Myles R, JA Kettleborough, and DA Stainforth. 2002. Model error in weather and climate forecasting. In *Ecmwf predictability of weather and climate seminar*. European Centre for Medium Range Weather Forecasts, Reading, UK, http://www ....

Anderson, Jeffrey L. 2001. An ensemble adjustment kalman filter for data assimilation. *Monthly weather review* 129(12):2884–2903.

Andrejczuk, M, FC Cooper, S Juricke, TN Palmer, A Weisheimer, and Laure Zanna. 2016. Oceanic stochastic parameterizations in a seasonal forecast system. *Monthly Weather Review* 144(5):1867–1875.

Apte, A, Christopher KRT Jones, AM Stuart, and Jochen Voss. 2008a. Data assimilation: Mathematical and statistical perspectives. *International journal for numerical methods in fluids* 56(8):1033–1046.

Apte, Amit, Christopher KRT Jones, and AM Stuart. 2008b. A bayesian approach to lagrangian data assimilation. *Tellus A: Dynamic Meteorology and Oceanography* 60(2):336–347.

Arnold, HM, IM Moroz, and TN Palmer. 2013. Stochastic parametrizations and model uncertainty in the lorenz'96 system. *Phil. Trans. R. Soc. A* 371(1991): 20110479.

Asch, Mark, Marc Bocquet, and Maëlle Nodet. 2016. *Data assimilation: methods, algorithms, and applications*. SIAM.

Bédard, Mylène. 2017. Hierarchical models: Local proposal variances for rwm-within-gibbs and mala-within-gibbs. *Computational Statistics & Data Analysis* 109: 231–246.

Bergemann, K., and S. Reich. 2012. An ensemble Kalman-Bucy filter for continuous data assimilation. *Meteorologische Zeitschrift* 21:213–219.

Berner, Judith, Ulrich Achatz, Lauriane Batte, Lisa Bengtsson, Alvaro de la Cámara, Hannah M Christensen, Matteo Colangeli, Danielle RB Coleman, Daan Crommelin, Stamen I Dolaptchiev, et al. 2017. Stochastic parameterization: Toward a new view of weather and climate models. *Bulletin of the American Meteorological Society* 98(3): 565–588.

Beucler, Tom, Stephan Rasp, Michael Pritchard, and Pierre Gentine. 2019. Achieving conservation of energy in neural network emulators for climate modeling. *arXiv preprint arXiv:1906.06622*.

Bocquet, Marc, Julien Brajard, Alberto Carrassi, and Laurent Bertino. 2020. Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *arXiv preprint arXiv:2001.06270*.

Botev, Zdravko I, Joseph F Grotowski, and Dirk P Kroese. 2010. Kernel density estimation via diffusion. *The annals of Statistics* 38(5):2916–2957.

Brajard, Julien, Alberto Carrassi, Marc Bocquet, and Laurent Bertino. 2020. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the lorenz 96 model. *Journal of Computational Science* 44:101171.

———. 2021. Combining data assimilation and machine learning to infer unresolved scale parametrization. *Philosophical Transactions of the Royal Society A* 379(2194):20200086.

Branicki, Michal, Nan Chen, and Andrew J Majda. 2013. Non-gaussian test models for prediction and state estimation with model errors. *Chinese Annals of Mathematics, Series B* 34(1):29–64.

Branicki, Michal, and Andrew J Majda. 2013. Dynamic stochastic superresolution of sparsely observed turbulent systems. *Journal of Computational Physics* 241:333–363.

Branicki, Michal, Andrew J Majda, and Kody JH Law. 2018. Accuracy of some approximate gaussian filters for the navier–stokes equation in the presence of model error. *Multiscale Modeling & Simulation* 16(4):1756–1794.

Brunton, Steven L, Joshua L Proctor, and J Nathan Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 113(15):3932–3937.

Bucy, Richard S, and Peter D Joseph. 2005. *Filtering for stochastic processes with applications to guidance*, vol. 326. American Mathematical Soc.

Cappé, Olivier, Eric Moulines, and Tobias Rydén. 2006. *Inference in hidden markov models*. Springer Science & Business Media.

Carlberg, Kevin, Charbel Farhat, Julien Cortial, and David Amsallem. 2013. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics* 242:623–647.

Carpenter, James, Peter Clifford, and Paul Fearnhead. 1999. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation* 146(1): 2–7.

Chattopadhyay, Ashesh, Pedram Hassanzadeh, and Devika Subramanian. 2020a. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics* 27(3):373–389.

Chattopadhyay, Ashesh, Mustafa Mustafa, Pedram Hassanzadeh, Eviatar Bach, and Karthik Kashinath. 2021. Towards physically consistent data-driven weather forecasting: Integrating data assimilation with equivariance-preserving spatial

transformers in a case study with era5. *Geoscientific Model Development Discussions* 1–23.

Chattopadhyay, Ashesh, Mustafa Mustafa, Pedram Hassanzadeh, and Karthik Kashinath. 2020b. Deep spatial transformers for autoregressive data-driven forecasting of geophysical turbulence. In *Proceedings of the 10th international conference on climate informatics*, 106–112.

Chattopadhyay, Ashesh, Adam Subel, and Pedram Hassanzadeh. 2020c. Data-driven super-parameterization using deep learning: Experimentation with multi-scale lorenz 96 systems and transfer learning. *Journal of Advances in Modeling Earth Systems* 12(11):e2020MS002084.

Chekroun, M. D., H. Liu, and J. C. McWilliams. 2020. Variational approach to closure of nonlinear dynamical systems: Autonomous case. *Journal of Statistical Physics* 179:1073–1160.

Chekroun, M. D., H. Liu, and S. Wang. 2015. *Stochastic Parameterizing Manifolds and Non-Markovian Reduced Equations: Stochastic Manifolds for Nonlinear SPDEs II*. Springer Briefs in Mathematics, Springer.

Chekroun, Mickaël D, and Dmitri Kondrashov. 2017. Data-adaptive harmonic spectra and multilayer Stuart-Landau models. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27(9):093110.

Chekroun, Mickaël David, Dmitri Kondrashov, and Michael Ghil. 2011. Predicting stochastic systems by noise sampling, and application to the el niño-southern oscillation. *Proceedings of the National Academy of Sciences* 108(29):11766–11771.

Chen, Nan. 2020a. Can short and partial observations reduce model error and facilitate machine learning prediction? *Entropy* 22(10):1075.

———. 2020b. Learning nonlinear turbulent dynamics from partial observations via analytically solvable conditional statistics. *Journal of Computational Physics* 418: 109635.

Chen, Nan, and Yingda Li. 2021. Bamcafe: A bayesian machine learning advanced forecast ensemble method for complex nonlinear turbulent systems with partial observations. *arXiv preprint arXiv:2107.05549*.

Chen, Nan, and Andrew Majda. 2018a. Conditional gaussian systems for multiscale nonlinear stochastic systems: Prediction, state estimation and uncertainty quantification. *Entropy* 20(7):509.

Chen, Nan, and Andrew J Majda. 2016a. Filtering nonlinear turbulent dynamical systems through conditional Gaussian statistics. *Monthly Weather Review* 144(12): 4885–4917.

———. 2016b. Model error in filtering random compressible flows utilizing noisy Lagrangian tracers. *Monthly Weather Review* 144(11):4037–4061.

———. 2017. Beating the curse of dimension with accurate statistics for the fokker–planck equation in complex turbulent systems. *Proceedings of the National Academy of Sciences* 114(49):12864–12869.

———. 2018b. Efficient statistically accurate algorithms for the fokker–planck equation in large dimensions. *Journal of Computational Physics* 354:242–268.

———. 2020. Efficient nonlinear optimal smoothing and sampling algorithms for complex turbulent nonlinear dynamical systems with partial observations. *Journal of Computational Physics* 109381.

Chen, Nan, Andrew J Majda, and Dimitrios Giannakis. 2014a. Predicting the cloud patterns of the Madden-Julian oscillation through a low-order nonlinear stochastic model. *Geophysical Research Letters* 41(15):5612–5619.

Chen, Nan, Andrew J Majda, CT Sabeerali, and RS Ajayamohan. 2018a. Predicting monsoon intraseasonal precipitation using a low-order nonlinear stochastic model. *Journal of Climate* 31(11):4403–4427.

Chen, Nan, Andrew J Majda, and Xin T Tong. 2014b. Information barriers for noisy Lagrangian tracers in filtering random incompressible flows. *Nonlinearity* 27(9):2133.

———. 2015. Noisy Lagrangian tracers for filtering random rotating compressible flows. *Journal of Nonlinear Science* 25(3):451–488.

———. 2018b. Rigorous analysis for efficient statistically accurate algorithms for solving fokker–planck equations in large dimensions. *SIAM/ASA Journal on Uncertainty Quantification* 6(3):1198–1223.

Chen, Tianping, and Hong Chen. 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks* 6(4):911–917.

Coleman, Matthew C, and David E Block. 2006. Bayesian parameter estimation with informative priors for nonlinear systems. *AIChE journal* 52(2):651–667.

Cosme, Emmanuel, Jacques Verron, Pierre Brasseur, Jacques Blum, and Didier Auroux. 2012. Smoothing problems in a bayesian framework and their linear gaussian solutions. *Monthly Weather Review* 140(2):683–695.

Cover, Thomas M, and Joy A Thomas. 1991. Entropy, relative entropy and mutual information. *Elements of information theory* 2(1):12–13.

Curry, Judith A, and Peter J Webster. 2011. Climate science and the uncertainty monster. *Bulletin of the American Meteorological Society* 92(12):1667–1682.

De Bézenac, Emmanuel, Arthur Pajot, and Patrick Gallinari. 2019. Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment* 2019(12):124009.

Del Moral, Pierre. 1997. Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* 325(6):653–658.

DelSole, Timothy. 2004. Predictability and information theory. part i: Measures of predictability. *Journal of the atmospheric sciences* 61(20):2425–2440.

———. 2005. Predictability and information theory. part ii: Imperfect forecasts. *Journal of the atmospheric sciences* 62(9):3368–3381.

Dembo, A, and O Zeitouni. 1986. Parameter estimation of partially observed continuous time stochastic processes via the em algorithm. *Stochastic Processes and their Applications* 23(1):91–113.

Denny, Mark W, Luke JH Hunt, Luke P Miller, and Christopher DG Harley. 2009. On the prediction of extreme ecological events. *Ecological Monographs* 79(3):397–421.

Dijkstra, Henk A. 2013. *Nonlinear climate dynamics*. Cambridge University Press.

Edwards, Paul N. 1999. Global climate science, uncertainty and politics: Data-laden models, model-filtered data. *Science as culture* 8(4):437–472.

———. 2011. History of climate modeling. *Wiley Interdisciplinary Reviews: Climate Change* 2(1):128–139.

Eraker, Bjørn. 2001. Mcmc analysis of diffusion models with application to finance. *Journal of Business & Economic Statistics* 19(2):177–191.

Evensen, Geir. 2009. *Data assimilation: the ensemble kalman filter*. Springer Science & Business Media.

Evensen, Geir, and Peter Jan Van Leeuwen. 2000. An ensemble kalman smoother for nonlinear dynamics. *Monthly Weather Review* 128(6):1852–1867.

Farazmand, Mohammad, and Themistoklis Sapsis. 2018. Extreme events: Mechanisms and prediction. *Applied Mechanics Reviews*.

Farazmand, Mohammad, and Themistoklis P Sapsis. 2019. Extreme events: Mechanisms and prediction. *Applied Mechanics Reviews* 71(5).

Farchi, Alban, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet. 2020. Using machine learning to correct model error in data assimilation and forecast applications. *arXiv preprint arXiv:2010.12605*.

Freedman, David A. 2009. *Statistical models: theory and practice*. cambridge university press.

Gardiner, Crispin W. 2004. Handbook of stochastic methods for physics, chemistry and the natural sciences, vol. 13 of springer series in synergetics.

Gershgorin, Boris, John Harlim, and Andrew J Majda. 2010a. Improving filtering and prediction of spatially extended turbulent systems with model errors through stochastic parameter estimation. *Journal of Computational Physics* 229(1):32–57.

———. 2010b. Test models for improving filtering with model errors through stochastic parameter estimation. *Journal of Computational Physics* 229(1):1–31.

Ghahramani, Zoubin, and Geoffrey E Hinton. 1996. Parameter estimation for linear dynamical systems. Tech. Rep., Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.

Ghahramani, Zoubin, and Sam T Roweis. 1999. Learning nonlinear dynamical systems using an em algorithm. In *Advances in neural information processing systems*, 431–437.

Ghil, Michael, and Stephen Childress. 2012. *Topics in geophysical fluid dynamics: atmospheric dynamics, dynamo theory, and climate dynamics*. Springer Science & Business Media.

Ghil, Michael, and Paola Malanotte-Rizzoli. 1991. Data assimilation in meteorology and oceanography. *Advances in geophysics* 33:141–266.

Golightly, Andrew, and Darren J Wilkinson. 2008. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis* 52(3):1674–1693.

Gottwald, Georg A, and AJ Majda. 2013. A mechanism for catastrophic filter divergence in data assimilation for sparse observation networks. *Nonlinear Processes in Geophysics* 20(5).

Greybush, Steven J, Eugenia Kalnay, Takemasa Miyoshi, Kayo Ide, and Brian R Hunt. 2011. Balance and ensemble kalman filter localization techniques. *Monthly Weather Review* 139(2):511–522.

Grooms, Ian, and Andrew J Majda. 2014a. Stochastic superparameterization in quasigeostrophic turbulence. *Journal of Computational Physics* 271:78–98.

Grooms, Ian G, and Andrew J Majda. 2014b. Stochastic superparameterization in a one-dimensional model for wave turbulence. *Communications in Mathematical Sciences* 12(3):509–525.

Harlim, John, Adam Mahdi, and Andrew J Majda. 2014. An ensemble kalman filter for statistical estimation of physics constrained nonlinear regression models. *Journal of Computational Physics* 257:782–812.

Harlim, John, Andrew J Majda, et al. 2010. Catastrophic filter divergence in filtering nonlinear dissipative systems. *Communications in Mathematical Sciences* 8(1):27–43.

Hasselmann, K. 1988. PIPs and POPs: The reduction of complex dynamical systems using principal interaction and oscillation patterns. *Journal of Geophysical Research: Atmospheres* 93(D9):11015–11021.

Hendon, Harry H, Eunpa Lim, Guomin Wang, Oscar Alves, and Debra Hudson. 2009. Prospects for predicting two flavors of el niño. *Geophysical Research Letters* 36(19).

Hijazi, Saddam, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. 2020. Data-driven pod-galerkin reduced order model for turbulent flows. *Journal of Computational Physics* 416:109513.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hol, Jeroen D, Thomas B Schon, and Fredrik Gustafsson. 2006. On resampling algorithms for particle filters. In *2006 ieee nonlinear statistical signal processing workshop*, 79–82. IEEE.

Holmes, P., J. L. Lumley, and G. Berkooz. 1996. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge.

Hornik, Kurt. 1993. Some new results on neural network approximation. *Neural networks* 6(8):1069–1072.

Houtekamer, Peter L, and Herschel L Mitchell. 1998. Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review* 126(3):796–811.

Hyndman, Rob J, and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International journal of forecasting* 22(4):679–688.

Ide, Kayo, Leonid Kuznetsov, and Christopher KRT Jones. 2002. Lagrangian data assimilation for point vortex systems*. *Journal of Turbulence* 3(053).

Jaeger, Herbert. 2007. Echo state network. *scholarpedia* 2(9):2330.

Johnson, Alicia A, Galin L Jones, Ronald C Neath, et al. 2013. Component-wise markov chain monte carlo: Uniform and geometric ergodicity under mixing and composition. *Statistical Science* 28(3):360–375.

Kalman, Rudolph E, and Richard S Bucy. 1961. New results in linear filtering and prediction theory. *Journal of basic engineering* 83(1):95–108.

Kalman, Rudolph Emil. 1960. A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82(1):35–45.

Kalnay, Eugenia. 2003. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press.

Katzfuss, Matthias, Jonathan R Stroud, and Christopher K Wikle. 2020. Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. *Journal of the American Statistical Association* 115(530):866–885.

Kelly, David, Andrew J Majda, and Xin T Tong. 2015. Concrete ensemble kalman filters with rigorous catastrophic filter divergence. *Proceedings of the National Academy of Sciences* 112(34):10589–10594.

Kim, Hye-Mi, Peter J Webster, and Judith A Curry. 2012. Seasonal prediction skill of ecmwf system 4 and ncep cfsv2 retrospective forecast for the northern hemisphere winter. *Climate Dynamics* 39(12):2957–2973.

Kitagawa, Genshiro. 1996. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics* 5(1): 1–25.

Kitagawa, Genshiro, and Seisho Sato. 2001. Monte carlo smoothing and self-organising state-space model. In *Sequential monte carlo methods in practice*, 177–195. Springer.

Kleeman, Richard. 2011. Information theory and dynamical system predictability. *Entropy* 13(3):612–649.

Kohavi, Ron, David H Wolpert, et al. 1996. Bias plus variance decomposition for zero-one loss functions. In *Icml*, vol. 96, 275–83.

Kokkala, Juho, Arno Solin, and Simo Särkkä. 2014. Expectation maximization based parameter estimation by sigma-point and particle smoothing. In *17th international conference on information fusion (fusion)*, 1–8. IEEE.

Kondrashov, Dmitri, Mickaël D Chekroun, and Michael Ghil. 2015. Data-driven non-Markovian closure models. *Physica D: Nonlinear Phenomena* 297:33–55.

Kristensen, Niels Rode, Henrik Madsen, and Sten Bay Jørgensen. 2004. Parameter estimation in stochastic grey-box models. *Automatica* 40(2):225–237.

Kullback, Solomon. 1987. Letter to the editor: The kullback-leibler distance. *AMERICAN STATISTICIAN*.

Kullback, Solomon, and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22(1):79–86.

Kwasniok, F. 1996. The reduction of complex dynamical systems using principal interaction patterns. *Physica D: Nonlinear Phenomena* 92(1-2):28–60.

Lahoz, William, Boris Khattatov, and Richard Ménard. 2010. Data assimilation and information. In *Data assimilation*, 3–12. Springer.

Lainscsek, Claudia. 2012. A class of lorenz-like systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22(1):013126.

Lau, William K-M, and Duane E Waliser. 2011. *Intraseasonal variability in the atmosphere-ocean climate system*. Springer Science & Business Media.

Law, Kody, Andrew Stuart, and Kostas Zygalakis. 2015. Data assimilation. *Cham, Switzerland: Springer* 214.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Lee, Y, and AJ Majda. 2017. Multiscale data assimilation and prediction using clustered particle filters. *J Comput Phys*.

Lermusiaux, Pierre FJ. 1999. Data assimilation via error subspace statistical estimation. part ii: Middle atlantic bight shelfbreak front simulations and esse validation. *Monthly Weather Review* 127(7):1408–1432.

Leutbecher, Martin, and Tim N Palmer. 2008. Ensemble forecasting. *Journal of computational physics* 227(7):3515–3539.

Lin, Kevin K, and Fei Lu. 2021. Data-driven model reduction, wiener projections, and the koopman-mori-zwanzig formalism. *Journal of Computational Physics* 424: 109864.

Liptser, Robert S, and Albert N Shiryaev. 2013. *Statistics of random processes ii: Applications*, vol. 6. Springer Science & Business Media.

Liu, Jun S, and Rong Chen. 1998. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association* 93(443):1032–1044.

Lorenz, Edward N. 1963. Deterministic nonperiodic flow. *Journal of the atmospheric sciences* 20(2):130–141.

———. 1996. Predictability: A problem partly solved. In *Proc. seminar on predictability*, vol. 1.

Lucarini, Valerio, Francesco Ragone, and Frank Lunkeit. 2017. Predicting climate change using response theory: Global averages and spatial patterns. *Journal of Statistical Physics* 166(3-4):1036–1064.

Majda, Andrew. 2003. *Introduction to pdes and waves for the atmosphere and ocean*, vol. 9. American Mathematical Soc.

Majda, Andrew, Rafail V Abramov, and Marcus J Grote. 2005. *Information theory and stochastics for multiscale nonlinear systems*, vol. 25. American Mathematical Soc.

Majda, Andrew, Richard Kleeman, David Cai, et al. 2002. A mathematical framework for quantifying predictability through relative entropy. *Methods and Applications of Analysis* 9(3):425–444.

Majda, Andrew J. 2012. Challenges in climate science and contemporary applied mathematics. *Communications on Pure and Applied Mathematics* 65(7):920–948.

———. 2015. Statistical energy conservation principle for inhomogeneous turbulent dynamical systems. *Proceedings of the National Academy of Sciences* 112(29): 8937–8941.

———. 2016. *Introduction to turbulent dynamical systems in complex systems*. Springer.

Majda, Andrew J, and Michal Branicki. 2012. Lessons in uncertainty quantification for turbulent dynamical systems. *Discrete & Continuous Dynamical Systems-A* 32(9): 3133–3221.

Majda, Andrew J, and Nan Chen. 2018. Model error, information barriers, state estimation and prediction in complex multiscale systems. *Entropy* 20(9):644.

Majda, Andrew J, Christian Franzke, and Daan Crommelin. 2009. Normal forms for reduced stochastic climate models. *Proceedings of the National Academy of Sciences* 106(10):3649–3653.

Majda, Andrew J, Christian Franzke, and Boualem Khouider. 2008. An applied mathematics perspective on stochastic modelling for climate. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 366(1875):2427–2453.

Majda, Andrew J, Boris Gershgorin, and Yuan Yuan. 2010. Low-frequency climate response and fluctuation–dissipation theorems: Theory and practice. *Journal of Atmospheric Sciences* 67(4):1186–1201.

Majda, Andrew J, and Ian Grooms. 2014. New perspectives on superparameterization for geophysical turbulence. *Journal of Computational Physics* 271:60–77.

Majda, Andrew J, and Marcus J Grote. 2009. Mathematical test models for superparametrization in anisotropic turbulence. *Proceedings of the National Academy of Sciences* 106(14):5470–5474.

Majda, Andrew J, and John Harlim. 2012a. *Filtering complex turbulent systems*. Cambridge University Press.

———. 2012b. Physics constrained nonlinear regression models for time series. *Nonlinearity* 26(1):201.

Majda, Andrew J, Ilya Timofeyev, and Eric Vanden Eijnden. 1999. Models for stochastic climate prediction. *Proceedings of the National Academy of Sciences* 96(26): 14687–14691.

Majda, Andrew J, Ilya Timofeyev, and Eric Vanden Eijnden. 2001. A mathematical framework for stochastic climate models. *Communications on Pure and Applied Mathematics* 54(8):891–974.

Majda, Andrew J, and Yuan Yuan. 2012. Fundamental limitations of ad hoc linear and quadratic multi-level regression models for physical systems. *Discrete and Continuous Dynamical Systems B* 17(4):1333–1363.

Manneville, Paul, and Yves Pomeau. 1979. Intermittency and the lorenz model. *Physics Letters A* 75(1-2):1–2.

Moffatt, HK. 2021. Extreme events in turbulent flow. *Journal of Fluid Mechanics* 914.

Mohamad, Mustafa A, and Themistoklis P Sapsis. 2015. Probabilistic description of extreme events in intermittently unstable dynamical systems excited by correlated stochastic processes. *SIAM/ASA Journal on Uncertainty Quantification* 3(1):709–736.

Moosavi, Azam, Razvan Stefanescu, and Adrian Sandu. 2015. Efficient construction of local parametric reduced order models using machine learning techniques. *arXiv preprint arXiv:1511.02909*.

Mou, Changhong, Birgul Koc, Omer San, Leo G Rebholz, and Traian Iliescu. 2021a. Data-driven variational multiscale reduced order models. *Computer Methods in Applied Mechanics and Engineering* 373:113470.

Mou, Changhong, Honghu Liu, David R Wells, and Traian Iliescu. 2020. Data-driven correction reduced order models for the quasi-geostrophic equations: A numerical investigation. *International Journal of Computational Fluid Dynamics* 34(2): 147–159.

Mou, Changhong, Zhu Wang, David R Wells, Xuping Xie, and Traian Iliescu. 2021b. Reduced order models for the quasi-geostrophic equations: A brief survey. *Fluids* 6(1):16.

Myung, In Jae. 2003. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology* 47(1):90–100.

Noack, Bernd R, Marek Morzynski, and Gilead Tadmor. 2011. *Reduced-order modelling for flow control*, vol. 528. Springer Science & Business Media.

Nummiaro, Katja, Esther Koller-Meier, and Luc Van Gool. 2003. An adaptive color-based particle filter. *Image and vision computing* 21(1):99–110.

Olsson, Jimmy, Olivier Cappé, Randal Douc, Eric Moulines, et al. 2008. Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli* 14(1):155–179.

Olsson, Jimmy, Johan Westerborn, et al. 2017. Efficient particle-based online smoothing in general hidden markov models: the paris algorithm. *Bernoulli* 23(3): 1951–1996.

Ottobre, Michela, Natesh S Pillai, Frank J Pinski, Andrew M Stuart, et al. 2016. A function space hmc algorithm with second order langevin diffusion limit. *Bernoulli* 22(1):60–106.

Palmer, Tim. 2019. The ecmwf ensemble prediction system: Looking back (more than) 25 years and projecting forward 25 years. *Quarterly Journal of the Royal Meteorological Society* 145:12–24.

Palmer, Tim N. 2001. A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models. *Quarterly Journal of the Royal Meteorological Society* 127(572): 279–304.

Palmer, TN. 1993. A nonlinear dynamical perspective on climate change. *Weather* 48(10):314–326.

Papaspiliopoulos, Omiros, Gareth O Roberts, and Osnat Stramer. 2013. Data augmentation for diffusions. *Journal of Computational and Graphical Statistics* 22(3): 665–688.

Pathak, Jaideep, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. 2018. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters* 120(2):024102.

Pawar, Suraj, Shady E Ahmed, Omer San, and Adil Rasheed. 2020. Data-driven recovery of hidden physics in reduced order modeling of fluid flows. *Physics of Fluids* 32(3):036602.

Peherstorfer, Benjamin, and Karen Willcox. 2015. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering* 291:21–41.

Pelt, Daniël M, and James A Sethian. 2018. A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences* 115(2):254–259.

Phillips, Thomas J, Gerald L Potter, David L Williamson, Richard T Cederwall, James S Boyle, Michael Fiorino, Justin J Hnilo, Jerry G Olson, Shaocheng Xie, and J John Yio. 2004. Evaluating parameterizations in general circulation models: Climate simulation meets weather prediction. *Bulletin of the American Meteorological Society* 85(12):1903–1916.

Plant, RS, and George C Craig. 2008. A stochastic parameterization for deep convection based on equilibrium statistics. *Journal of the Atmospheric Sciences* 65(1): 87–105.

Ragone, Francesco, Valerio Lucarini, and Frank Lunkeit. 2016. A new framework for climate sensitivity and prediction: a modelling perspective. *Climate Dynamics* 46(5-6):1459–1471.

Raissi, Maziar, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378:686–707.

Rasp, Stephan, and Nils Thuerey. 2021. Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for weatherbench. *Journal of Advances in Modeling Earth Systems* 13(2):e2020MS002405.

Rauch, Herbert E, CT Striebel, and F Tung. 1965. Maximum likelihood estimates of linear dynamic systems. *AIAA journal* 3(8):1445–1450.

Rotstayn, Leon D. 2000. On the "tuning" of autoconversion parameterizations in climate models. *Journal of Geophysical Research: Atmospheres* 105(D12):15495–15507.

Rowley, C. W., I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. 2009. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics* 641:115–127.

Salmon, Rick. 1998. *Lectures on geophysical fluid dynamics*. Oxford University Press.

San, Omer, and Romit Maulik. 2018. Extreme learning machine for reduced order modeling of turbulent geophysical flows. *Physical Review E* 97(4):042322.

Särkkä, Simo. 2013. *Bayesian filtering and smoothing*, vol. 3. Cambridge University Press.

Scher, S, and G Messori. Ensemble methods for neural network-based weather forecasts. *Journal of Advances in Modeling Earth Systems* e2020MS002331.

Scher, Sebastian. 2018. Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters* 45(22):12–616.

Scher, Sebastian, and Gabriele Messori. 2018. Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society* 144(717):2830–2841.

Schmid, P. J. 2010. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics* 656:5–28.

Sheard, Sarah A, and Ali Mostashari. 2009. Principles of complex systems for systems engineering. *Systems Engineering* 12(4):295–311.

Simonoff, Jeffrey S. 2012. *Smoothing methods in statistics*. Springer Science & Business Media.

Smarra, Francesco, Achin Jain, Tullio De Rubeis, Dario Ambrosini, Alessandro D'Innocenzo, and Rahul Mangharam. 2018. Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied energy* 226:1252–1272.

Sonoda, Sho, and Noboru Murata. 2017. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis* 43(2):233–268.

Strogatz, Steven H. 2018. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press.

Taira, Kunihiko, Maziar S Hemati, Steven L Brunton, Yiyang Sun, Karthik Duraisamy, Shervin Bagheri, Scott TM Dawson, and Chi-An Yeh. 2020. Modal analysis of fluid flows: Applications and outlook. *AIAA journal* 58(3):998–1022.

Tanner, Martin A, and Wing Hung Wong. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association* 82(398):528–540.

Tao, Wei-Kuo, Jiun-Dar Chern, Robert Atlas, David Randall, Marat Khairoutdinov, Jui-Lin Li, Duane E Waliser, Arthur Hou, Xin Lin, Christa Peters-Lidard, et al. 2009. A multiscale modeling system: Developments, applications, and critical issues. *Bulletin of the American Meteorological Society* 90(4):515–534.

Taylor, Karl E. 2001. Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research: Atmospheres* 106(D7):7183–7192.

Tomizawa, Futo, and Yohei Sawada. 2020. Combining ensemble kalman filter and reservoir computing to predict spatio-temporal chaotic systems from imperfect observations and models. *Geoscientific Model Development Discussions* 1–33.

Tong, Xin T, Mathias Morzfeld, and Youssef M Marzouk. 2020. Mala-within-gibbs samplers for high-dimensional distributions with sparse conditional structure. *SIAM Journal on Scientific Computing* 42(3):A1765–A1788.

Toth, Zoltan, and Eugenia Kalnay. 1997. Ensemble forecasting at ncep and the breeding method. *Monthly Weather Review* 125(12):3297–3319.

Trenberth, Kevin E, John T Fasullo, and Theodore G Shepherd. 2015. Attribution of climate extreme events. *Nature Climate Change* 5(8):725–730.

Vallis, Geoffrey K. 2017. *Atmospheric and oceanic fluid dynamics*. Cambridge University Press.

Vlachas, Pantelis R, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. 2018. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474(2213):20170844.

Wan, Zhong Yi, and Themistoklis P Sapsis. 2017. Reduced-space gaussian process regression for data-driven probabilistic forecast of chaotic dynamical systems. *Physica D: Nonlinear Phenomena* 345:40–55.

Wang, Xiong, and Guanrong Chen. 2013. A gallery of lorenz-like and chen-like attractors. *International Journal of Bifurcation and Chaos* 23(04):1330011.

Wei, Greg CG, and Martin A Tanner. 1990. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association* 85(411):699–704.

Weyn, Jonathan A, Dale R Durran, and Rich Caruana. 2020. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems* 12(9):e2020MS002109.

Wikner, Alexander, Jaideep Pathak, Brian R Hunt, Istvan Szunyogh, Michelle Girvan, and Edward Ott. 2021. Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31(5):053114.

Wilcox, David C. 1988. Multiscale model for turbulent flows. *AIAA journal* 26(11): 1311–1320.

Wilks, Daniel S. 2005. Effects of stochastic parametrizations in the lorenz'96 system. *Quarterly Journal of the Royal Meteorological Society* 131(606):389–407.

Xie, Xuping, Muhammad Mohebujjaman, Leo G Rebholz, and Traian Iliescu. 2018. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing* 40(3):B834–B857.

Xu, Qin. 2007. Measuring information content from observations for data assimilation: Relative entropy versus shannon entropy difference. *Tellus A: Dynamic Meteorology and Oceanography* 59(2):198–209.

Yan, Xin, and Xiaogang Su. 2009. *Linear regression analysis: theory and computing*. World Scientific.