# Non-parametric methodologies for reconstruction and estimation in nonlinear state-space models

Thi Tuyet Trang Chau

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

# THÈSE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
**C**OMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
*Mathèmatique et Sciences et Technologies de l'Information et de la Communication*
Spécialité : Mathématiques et leurs Interactions
Par


# Thi Tuyet Trang CHAU


## Non-parametric methodologies for reconstruction and estimation in nonlinear state-space models

**Thèse présentée et soutenue à** RENNES, le 26/2/2019
**Unité de recherche : UMR CNRS 6625 Institut de recherche mathématique de Rennes (IRMAR)**

**Rapporteurs avant soutenance :**

Marc BOCQUET, Professeur, CEREA École des Ponts ParisTech
Fredrik LINDSTEN, Professeur, Université d'Uppsala

**Composition du jury :**

Président :
Philippe NAVEAU, Dr, CNRS-LSCE Paris

Examinateur :
François LE GLAND, Dr, Inria Rennes

Dir. de thèse : **Valérie MONBET**, Professeur, IRMAR-INRIA Université de Rennes 1

Co-dir. de thèse : **Pierre AILLIOT**, Maître de conférences, LMBA Université de Brest

Invité(s)

# Acknowledgements

Since I was a child, I have dreamed of stepping on new lands in the world, learning and making great things. And France, the first abroad country I have experienced, gives me numerous opportunities and challenges. More than three years passed, the first achievement I have got is this dissertation. It has been resulting from not only my efforts but also supports of many people whom I would like to send all my best wishes and all my honest thanks to.

I first wish to express my deepest gratitude to my supervisors, Prof. Valérie Monbet and Dr. Pierre Ailliot. These two persons always guide me towards bright roads, give me valuable discussions and encourage me at the stages I was feeling depressed. During my PhD journey, they have taken a role of both dear teachers and close friends sharing me meaningful advices in research and life.

I want to deeply thank Dr. Pierre Tandeo, Dr. Juan Ruiz, Prof. François Le Gland, Dr. Anne Cuzol and Dr. Pierre Navaro who provide me various ideas and a lot of interesting feedback on my research. The thesis would not have been completed without their contributions.

I much appreciate Prof. Marc Bocquet and Prof. Fredrik Lindsten for reviewing my thesis manuscript. Their comments and remarks are indeed useful for its improvement. It is also my pleasure for giving my thanks to Prof. François Le Gland and Prof. Philippe Naveau who accepted to be jury members in my defense.

I am very grateful to Institute of Mathematical Research of Rennes (IRMAR) and Center of Henri Lebesgue (CHL) and the staff therein. A special thank is sent to Xhenxila Lachambre, Hélène Rousseaux, Elodie Cottrell, Marie-Aude Verger, Florian Rogowski and Chantal Halet for all their administrative supports.

Besides, I would like to thank Dr. Marion Gehlen and Dr. Frederic Chevallier for offering the current Postdoc position and facilitating the completion of the last phase of my PhD.

Last but not least, I wish to give all my heart to my parents, my sisters and my friends in Vietnam, France, Netherlands, Sweden, and Poland. All of them are on my side whenever I

need. Their love, encouragements, and supports are motivations for me in the past, the present and the future.

# Contents

# Acronyms

**AnDA** Analog Data Assimilation

**BS** Backward Simulation

**CPF** Conditional Particle Filter

**CPS** Conditional Particle Smoother

**DA** Data Assimilation

**EnKF** Ensemble Kalman Filter

**EnKS** Ensemble Kalman Smoother

**EKF** Extended Kalman Filter

**EM** Expectation-Maximization

**KF** Kalman Filter

**KS** Kalman Smother

**LCR** Local Constant Regression

**LLR** Local Linear Regression

**npSEM** non-parametric Stochastic Expectation-Maximization

**ODE** Ordinary Differential Equation

**PMCMC** Particle Markov Chain Monte Carlo

**PF** Particle Filter

**SEM** Stochastic Expectation-Maximization

**SSM** State-Space Model

# General introduction

## Motivations and problem statements

Thanks to the development of technological and computational sciences, both quantity and quality of data have been increasing in the last decades. This thesis was motivated by applications of data analysis in environment, climatology and oceanography. In these fields, the exponential growth in remote-sensing, in-situ or model-run data availability expected to continue in the future creates many new opportunities, needs and challenges. In particular, the environmental data are typically available with a complex spatio-temporal sampling, on irregular grids, and subject to observational errors due to the complexity of collecting data, modeling imperfection, etc.

State-space models (SSMs) [37, 51, 76, 114] is one popular approach for analyzing data with observational errors. In particular, they are at the heart of sequential data assimilation techniques in oceanography and meteorology. A general SSM consists of a dynamical model, which describes the physical evolution of the phenomenon of interest, and an observation model which models the relation between the (noisy) observations and the (true) state. Many difficulties arise when working with SSMs and in this thesis we focus on the following challenges (see Figure 1 for an illustration of these challenges).

i. **State reconstruction when the dynamical model is known and the parameters are known**

   Filtering and smoothing (so-called sequential data assimilation in geosciences) are standard approaches to recursively compute probability distributions of the state conditional on a sequence of observations. Within these assimilation frameworks, the dynamical model is used to propagate state estimates from a past time to latter times. The forecasts are then corrected by taking into account the available observations.

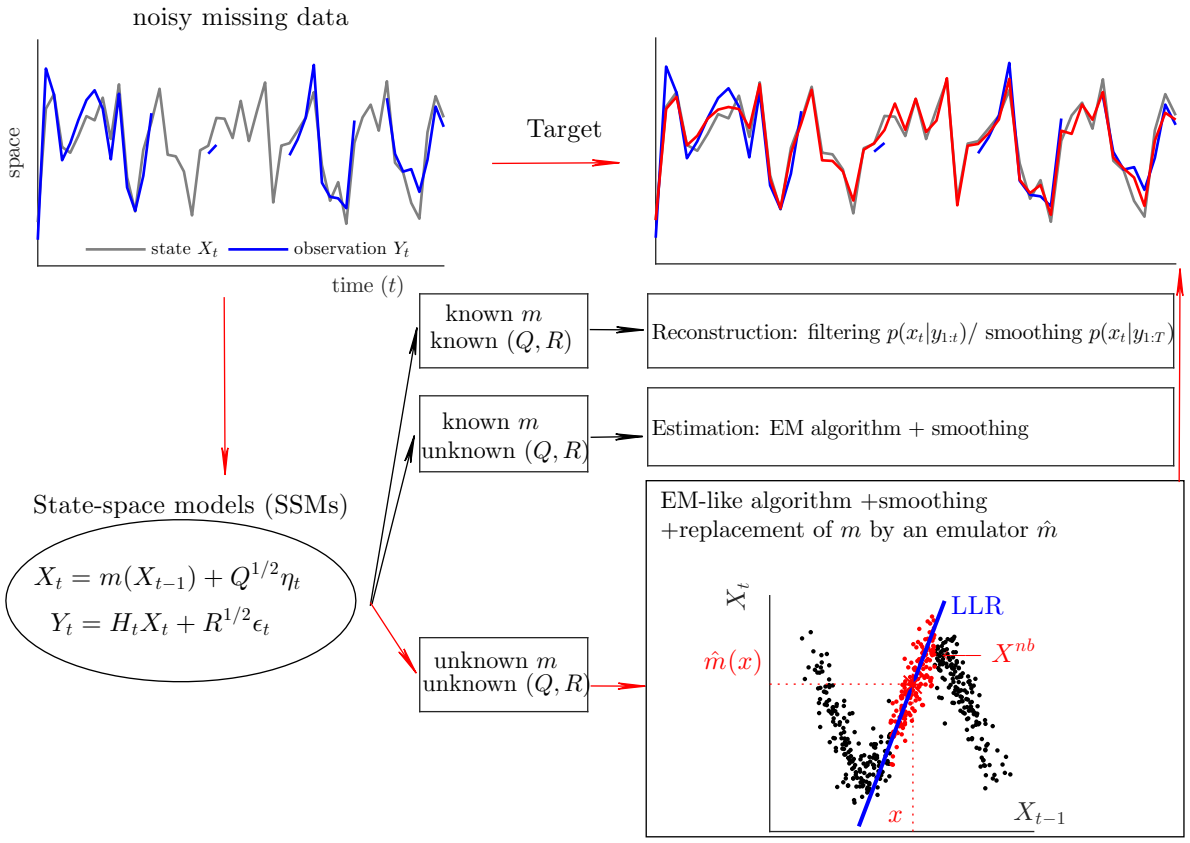


**Figure 1** – Illustration of statistical inference problems in SSMs addressed in the thesis. $X^{nb}$ denotes a set of neighbors of $x$ which are used to estimate $m(x)$ by local linear regression (LLR) method.

For linear Gaussian models, the Kalman recursions [51, 74, 85, 129, 134] can be used to correctly analyse the filtering and smoothing distributions. When state-space models are nonlinear, as it is the typical case for real applications, these distributions do not admit any closed form. Simulation-based methods are instead implemented. Ensemble Kalman-based approaches (see e.g. in [15, 56, 58]) are the most used assimilation approaches in practice due to their efficiency in approximating the filtering and smoothing distributions of high dimensional problems (only few simulations (members) of the dynamical model are run). Notwithstanding, the approximations do not converge to the true conditional distributions for (highly) nonlinear situations [93]. In statistical and signal processing communities, particle filters and smoothers are used as flexible and powerful tools to reconstruct the state in nonlinear and/or non-Gaussian models. Many algorithms have been proposed in the literature [23, 46, 49, 69].

ii. **Parameter estimation when the dynamical model is specified with unknown parameters**

The accuracy of the results obtained when reconstructing the physical variables from the observed data using SSMs does not only depend on the assimilation methods but is also related to the static parameters involved in the modeling of the physical processes and error noises. In practice, it is often difficult to specify reasonable values for these unknown parameters. This is due to the diversity of observation sources, the effect of physical terms and model complexity, or numerical failures [50, 182]. Therefore parameter estimation (or system identification) is one important preliminary task before running data assimilation algorithms.

Usual statistical approaches for parameter estimation consist of Bayesian and maximum likelihood estimation. The Bayesian approaches [4, 86, 102, 147, 148, 165] aim at simulating the joint distribution of the state and the parameter but that may be impractical for problems in high dimensional SSMs (e.g. error covariance inference). An alternative is to implement the maximum likelihood estimation approaches including Expectation-Maximization (EM) algorithm [42] and its variants [28, 41, 44, 110].

iii. **State reconstruction and parameter estimation when the dynamical model is, practically, unspecified as a parametric model**

In geosciences applications, the dynamical model is generally specified using differential equations derived from the physics and solved using numerical schemes. The numerical forecast model has to be run for each time step of the assimilation process. That usually leads to high computational cost in practice, for instance when the time increment between two successive state variables in the evolution model is large or only several components of the system are of the interest but the whole model must be run. Moreover, chaotic behaviours and model complexity can be reasons for inaccurate numerical approximations. Besides, various sources of uncertainties (unknown physical parameters, state noise covariances, forcing terms) may cause large bias between forecasts and observations. In such situations, the assimilation process may be inconsistent.

Nowadays, a huge amount of datasets recorded from satellite, situ or numerical simulations is available. The existence of such data promotes the development of data-driven models which are able to well describe the dynamics of the state. The combinations of the non-

parametric models and standard filtering and smoothing algorithms were first proposed in [95, 155].

Three main contributions of this thesis to these three challenges are listed below.

## Main contributions

i. **State reconstruction when the dynamical model is known and the parameters are known**

Recently, [4, 99, 102, 171] have developed conditional particle smoothers which allow to efficiently approximate the smoothing distribution with only few particles. In the thesis we investigate Conditional Particle Filter-Backward Simulation (CPF-BS) smoother presented in [101, 102, 171] and further discussed in [30]. We will show on several toy models that, at the same computational cost, the CPF-BS algorithm gives better results than standard particle-based smoothing algorithms.

ii. **Parameter estimation when the dynamical model is specified with unknown parameters**

When using the EM algorithms, the parameters are updated iteratively by maximizing a likelihood function defined consisting of smoothing distributions. Nevertheless, the smoothing distributions are intractable in nonlinear SSMs. In the works of [5, 86, 98, 116, 141, 149], it was proposed to combine the standard particle samplers, which permit to approximate the smoothing distributions, with the EM machinery. But this usually leads to a huge computational cost. In the thesis, we explore the combination of the CPF-BS sampler and EM algorithms, and show that this approach better performs than the combination of the stochastic EnKS and EM algorithm which is commonly used in real applications (see [30]).

iii. **State reconstruction and parameter estimation when the dynamical model is, practically, unspecified as a parametric model**

Inspired by the works of [95, 155], this thesis targets on investigating non-parametric methods for reconstruction of the state and the dynamical model using only observed data, in circumstances where the dynamical model is not specified. Two situations are considered. In the first situation, a learning dataset simulated from the state process with

no observation error is assumed to be available (as in [95, 155]). Based on these data, the dynamical model can be estimated by a non-parametric method (such as local regression [35, 38, 60], see Figure 1 for an illustration). In practice, such "perfect" observations of the state, with no observational error, are typically not available. In the second situation, only a sequence of the process with observational errors is available. This increases estimation errors if a non-parametric estimate is learned directly on this noisy data. To handle this problem, the thesis introduces a novel non-parametric algorithm which combines a non-parametric estimate of the dynamical model, a low-cost CPF-BS smoother and an EM-like algorithm. The performances of the proposed approach in terms of noise error reduction, missing-data imputation, parameter estimation and model comparison are illustrated on toy examples and wind data produced by Météo France.

## Plan of the thesis

Chapter 1 introduces fundamental materials and illustrates the issues tackled in the thesis. The concepts of SSMs and toy examples are first presented. Given a set of observations and a model with known parameters, filtering and smoothing methods used to compute the hidden state are reminded. We synthesize and analyze the advantages and drawbacks of different methods including Kalman recursions, some of their extensions and particle-based recursions. In the sequel, we summarize existing EM algorithms used to handle inference problems of SSMs with unknown parameters. The efficiency of parameter estimation of the EM algorithms combined with the particle-based filters and smoothers in nonlinear models is emphasized. In order to develop non-parametric algorithms, we review popular local regression methods used to construct non-parametric estimates of the dynamical model. Finally, we present the key ideas of implementation of these non-parametric emulators in the proposed algorithms.

In Chapter 2, we present non-parametric filtering algorithms for estimating the filtering distributions in nonlinear SSM models. Here local linear regression (LLR) is used to provide non-parametric estimates of the dynamical model. They are then combined with different filters including extended Kalman filter (EKF), ensemble Kalman filter (EnKF), bootstrap and optimal particle filters (PF). The main contribution of this chapter is the section of numerical results. Lots of experiments are run to compare the proposed approaches with the classical approaches, the proposed approaches with the non-parametric approaches using LCR estimates, and the proposed approaches within different filtering schemes. In summary, this chapter extends the

previous works [95, 154, 155] in: (1) pointing out that LLR gives better numerical forecast than LCR in filtering , (2) providing new combinations of LLR forecast emulator with EKF and optimal PF algorithms, (3) comparing all the mentioned approaches in different scenarios.

In DA applications of geosciences, the most favorite tools used to infer the state of the system from the observations are EnKF, EnKS and their extensions. Chapter 3 presents an alternative approach, CPF-BS smoother. This smoother allows to explore efficiently the latent space and simulate quickly relevant trajectories of the state conditionally on the observations. Numerical illustrations of the CPF-BS algorithms to simulate the state of toy models are provided that would help the readers to understand its smoothing process easily. Moreover, we propose to combine the CPF-BS smoother with an original stochastic EM (SEM) algorithm in order to estimate the unknown parameters and the hidden state. We show that this algorithm provides, with reasonable computational cost, accurate estimations of the static parameters and the state in highly nonlinear SSMs, where the application of an EM algorithm in conjunction with EnKS is limited.

The main contribution of this thesis is presented in Chapter 4. Novel non-parametric algorithms are invented to address two problems. Firstly, we aim at estimating the unknown parameters and inferring the hidden state given a sequence of observations and a "perfect" learning dataset (a simulated sequence of the state process without taking into account observational errors). Given the learning data, LLR is used to construct an estimate of the dynamical model. Based on Chapter 3, we propose to combine the statistical emulator with the low-cost CPF-BS smoother. This non-parametric smoother is then used to generate realizations of the state in an SEM algorithm. Nevertheless, such "perfect" data do rarely exist in reality but noisy data which are derived from the observation process. Consequently, estimating the dynamical model on the noisy data easily leads to increase bias and variance and may give bad effects on inference results. To deal with this issue, we now develop an SEM-like algorithm for estimating the dynamics and indentifying unknown parameters. Finally, different abilities of the novel method such as noise error reduction, missing-data imputation and parameter estimation are illustrated on toy models.

Chapter 5 presents two potential applications of the proposed non-parametric algorithms. Firstly, a non-parametric filtering algorithm is applied for model selection and model comparison given a set of observations and existing model runs. The performance of the proposed approach is compared to the one of the classical approach on toy models with different forcing parameterizations.

This work belongs partly to ECOS-SUD project in collaboration between France and Argentina ($2018 - 2020$). Then, we introduce an application of the npSEM algorithm for imputing noisy missing data. Wind data produced by Météo France is considered. Imputation results of the non-parametric SEM algorithm on the data are compared to the ones of regular regression methods.

At last, Chapter 6 recapitulates contributions of the thesis and introduces several topics for further research.

## Publications

This thesis is mostly contributed to the following submitted and preprint papers.

1. T.T.T. Chau, P. Ailliot, V. Monbet, P. Tandeo. *Simulation-based methods for uncertainty estimation in nonlinear state-space models*, submitted.

2. T.T.T. Chau, P. Ailliot, V. Monbet. *A novel non-parametric algorithm for reconstruction and estimation in nonlinear time series with observational error*, in revision.

3. T.T.T. Chau, P. Ailliot, V. Monbet, P. Tandeo. *Non-parametric filtering algorithms*, preprint.

4. T.T.T. Chau, J. Ruiz, P. Ailliot, P. Tandeo, V. Monbet. *An application of analog data assimilation methods in model comparison and model selection without specifying an explicit physical system*, preprint.

In addition, Python libraries for numerical experiments in the thesis are also developed.

1. *npSEM*, `https://github.com/tchau218/npSEM`.

2. *parEM*, `https://github.com/tchau218/parEM`.

Chapter 1

# Statistical inference in state-space models

State-space models (SSMs) [37,51] belong to an important class of time series models. Generally, an SSM consists of a dynamical model representing the evolution of the hidden state and an observation model describing the relation between the state and the measurements. Thanks to the diversity and simplicity in use, their frameworks have been applied in various areas such as statistics, economics and environmental sciences [5,24,59,118,160]. Numerous practical problems include estimating the latent state and relevant parameters given a sequence of observations and a parametric SSM. In the scope of this chapter, we first aim at providing definitions and several examples of the SSMs and reviewing classical methods used to tackle these usual inference problems. All is presented in Section 1.1. Then Section 1.2 generally introduces non-parametric SSMs, their main issues addressed in the thesis and materials used in the novel methodologies proposed in the next chapters.

## 1.1 Inference in parametric state-space models

### 1.1.1 State-space models

#### 1.1.1.1 Definitions

Let $(X_t)_{t=0:T}$ and $(Y_t)_{t=1:T}$ denote the hidden state and observation processes on a coupled space $(\mathcal{X}, \mathcal{Y})$. For each time step $t = 1 : T$, a general state-space model (SSM) is defined by

$$\begin{cases} X_t = \mathcal{M}_\theta\left(X_{t-1},\ \eta_t\right), & [hidden] \quad (1.1a) \\ Y_t = \mathcal{H}_\theta\left(X_t,\ \epsilon_t\right), & [observed] \quad (1.1b) \end{cases}$$

where $(\eta_t, \epsilon_t)$ represent stochastic noise processes and $\theta \in \Theta$ is a vector of static parameters involved in the model.

In Eq. (1.1), the dynamical model (1.1a) characterizes the evolution of the state. $\mathcal{M}_\theta$ is a function mapping the state from time $(t-1)$ to $(t)$. The model error noises $(\eta_t)_{t=1:T}$ include errors derived from modeling or parametrization imperfection, forcing terms, etc. They are assumed to have identical independent distributions with zeros means and $(Q_t)_{t=1:T}$ covariance matrices. Here $Q_t$ stands for model covariance which may vary in time or depend on the state value. The observation model (1.1b) formulates the relation between the state and the observation processes. The function $\mathcal{H}_\theta$ describes how well the observations capture the true state. For instance, in case of missing data, the transformation function is the mapping of the full state to a smaller space containing only its observed components. $(\epsilon_t)_{t=1:T}$ model for errors in data recording procedure, devices or observation formulation. These observational error noises are assumed to be independently distributed with zeros means and $(R_t)_{t=1:T}$ covariances and independent from the state. The size of observational covariances depends on the dimension of the observation at each particular time step. Note that the notation of error covariances $(Q_t, R_t)$ hereafter is substituted to $(Q, R)$ for the sake of presentation simplification if their values are time-constant.

Given an initial state distribution $p_\theta(x_0)$, a probabilistic description of the SSM (1.1) can be defined by

- $p_\theta(x_t|x_{t-1})$: Markov kernel (transition distribution of the hidden state process $(X_t)_t$) which depends on both the dynamical model $\mathcal{M}_\theta$ and the distribution of the model error $\eta_t$,

- $p_\theta(y_t|x_t)$: likelihood (observation distribution of the process $(Y_t)_t$ conditional on the state $X_t = x_t$) which is a function of the observation model $\mathcal{H}_\theta$ and the distribution of the observational error $\epsilon_t$.

The conditional dependence among the state variables and between the state and the observations is also illustrated by the following Directed Acyclic Graph (DAG).

$$\begin{array}{ccccccccc} \cdots & \rightarrow & X_{t-1} & \rightarrow & X_t & \rightarrow & X_{t+1} & \rightarrow & \cdots \\ & & \downarrow & & \downarrow & & \downarrow & & \\ \cdots & & Y_{t-1} & & Y_t & & Y_{t+1} & & \cdots \end{array}$$

#### 1.1.1.2 Examples

In this section, several examples of the SSM (1.1) are given. The dynamical and observational functions can be linear or nonlinear. To facilitate the presentation, model errors $(\eta_t)_t$ and observational errors $(\epsilon_t)_t$ are assumed to have additive Gaussian distributions which are the most usual cases considered in numerous applications.

a. **Linear state-space models**

Linear SSMs provide interesting properties for analyzing lots of problems in statistics, finance, signal processing, meteorology, etc [16, 19, 85, 86, 98, 116]. For instance, joint distributions and optimization problems relevant to the state and parameters in the models usually admit explicit expressions and/or analytic solutions. In the literature, a simple form of a linear model is presented as follows

$$\begin{cases} X_t = M_t X_{t-1} + \eta_t, \\ Y_t = H_t X_t + \epsilon_t, \end{cases} \tag{1.2}$$

where $(X_t, Y_t) \in \mathbb{R}^{d_{X_t}} \times \mathbb{R}^{d_{Y_t}}$, $M_t$ and $H_t$ are matrices in $\mathbb{R}^{d_{X_t}} \times \mathbb{R}^{d_{X_t}}$ and $\mathbb{R}^{d_{Y_t}} \times \mathbb{R}^{d_{X_t}}$, $\eta_t \sim \mathcal{N}(0, Q_t)$ and $\epsilon_t \sim \mathcal{N}(0, R_t)$. An illustration of this model is on Figure 1.1. In the thesis, we also use this type of model for verifying and comparing the results of several algorithms.

b. **Nonlinear state-space models**

Nonlinear SSMs have been considered in various applications. They are typically formulated by

$$\begin{cases} X_t = m(X_{t-1}) + \eta_t, \\ Y_t = h(X_t) + \epsilon_t, \end{cases} \tag{1.3}$$

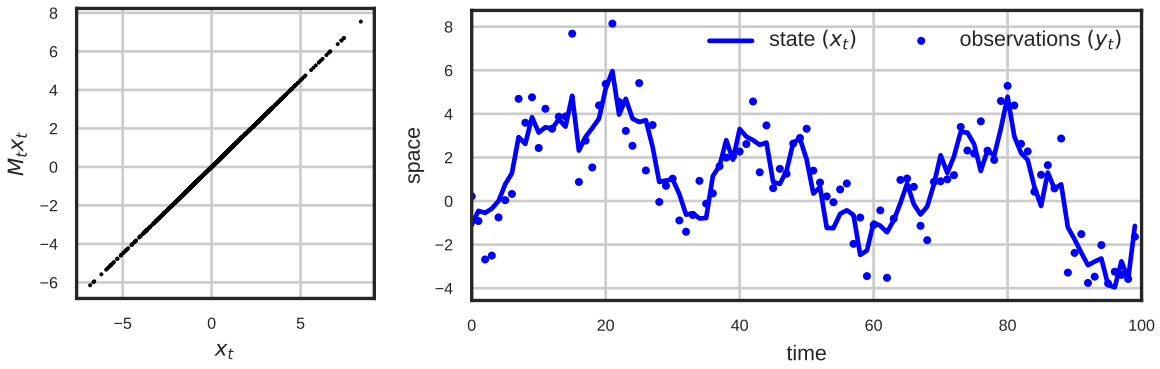


**Figure 1.1** – Scatter plot (left panel) of the dynamical model with respect to the state, and time series plot (right panel) of the state and observations simulated from a univariate linear SSM (1.2) where model coefficients $M_t = 0.9, H_t = 1$ and error variances $Q = R = 1$.

where $(X_t, Y_t) \in \mathbb{R}^{d_{X_t}} \times \mathbb{R}^{d_{Y_t}}$, $m$ and/or $h$ is a nonlinear function, $\eta_t \sim \mathcal{N}(0, Q_t)$ and $\epsilon_t \sim \mathcal{N}(0, R_t)$. Some examples of nonlinear models are presented below.

The first nonlinear model introduced is the sinus model (1.4) ( [117], see Figure 1.2 for illustrations of the model and time series of the state and observations). This nonlinear model is simple and univariate so that it facilitates the illustration of numerical results.

$$\begin{cases} X_t = \sin(3X_{t-1}) + \eta_t, \\ Y_t = X_t + \epsilon_t. \end{cases} \tag{1.4}$$

Here $(X_t, Y_t) \in \mathbb{R} \times \mathbb{R}$, $\eta_t \sim \mathcal{N}(0, Q)$ and $\epsilon_t \sim \mathcal{N}(0, R)$.

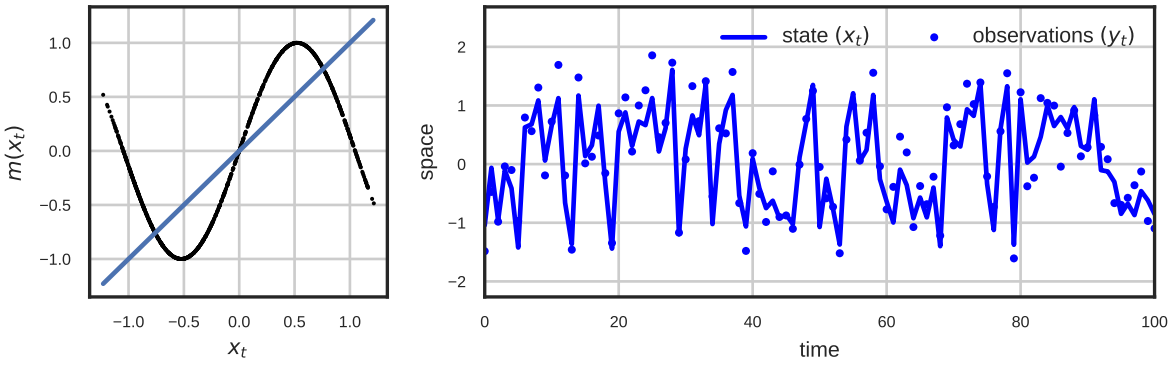


**Figure 1.2** – Scatter plot (left panel) of the dynamical model with respect to the state (the line represents an identity model), and time series plot (right panel) of the state and observations simulated from a sinus model (1.4) with error variances $Q = R = 0.1$.

A highly nonlinear system considered widely in the literature [48,69,88,89,141] to perform numerical illustrations of statistical inference problems is Kitagawa model (1.5). Both $m$

and $h$ in the SSM context are nonlinear and defined as follows

$$\begin{cases} X_t = 0.5X_{t-1} + 25\frac{X_{t-1}}{1+X_{t-1}^2} + 8\cos 1.2t + \ \eta_t, \\ Y_t = 0.05X_t^2 + \ \epsilon_t \end{cases} \tag{1.5}$$

where $(X_t, Y_t) \in \mathbb{R} \times \mathbb{R}$, $\eta_t \sim \mathcal{N}(0, Q)$ and $\epsilon_t \sim \mathcal{N}(0, R)$. This univariate nonlinear model is chosen because of its interesting properties. With the cos-term its transition $p(x_t|x_{t-1})$ can be multimodal distribution whose mean admits different values conditionally on a fixed value of $x_{t-1}$ (shown on the left panel of Figure 1.3),

$$p(x_t|x_{t-1} = x) = \mathcal{N}\left(x_t;\ 0.5x + \frac{x}{1+x^2} + 8\cos 1.2t, Q\right).$$

Moreover, the observation function is quadratic that would make confusion about whether the state is in the positive or negative space. If the observational error variance $R$ is large (here $R = 10$), we can also generate unreliable observations which probably provide the incorrect information of the state (see the right panel of Figure 1.3).

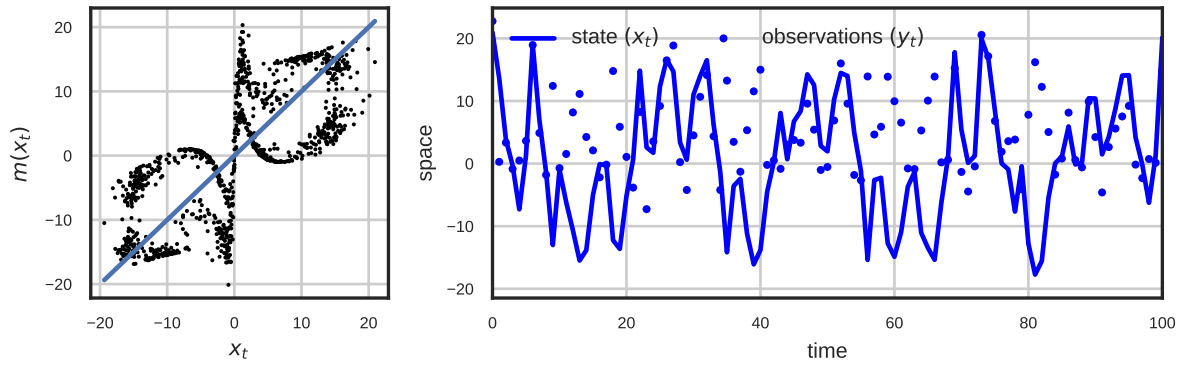


**Figure 1.3** – Scatter plot (left panel) of the dynamical model with respect to the state (the line represents an identity model), and time series plot (right panel) of the state and observations simulated from a Kitagawa model (1.5) with error variances $Q = 1$ and $R = 10$.

The more complicated model considered is the three-dimensional Lorenz 63 (L63, [107]) model which is nonlinear, non-periodic and chaotic (see left panel of Figure 1.4). This is one of the typical toy models used in data assimilation (DA) community. Additionally, the L63 model is used to mimic the atmospheric convection [50, 95, 153]. An example of

the L63 SSM is defined by

$$\begin{cases} X_t = m(X_{t-1}) + \eta_t, \\ Y_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} X_t + \epsilon_t \end{cases} \tag{1.6}$$

where $(X_t, Y_t) \in \mathbb{R}^3 \times \mathbb{R}^2$, $\eta_t \sim \mathcal{N}(0, Q)$, $\epsilon_t \sim \mathcal{N}(0, R)$ and $m$ is numerically approximated by integrating the following system of ordinary differential equations (ODEs) for $x \in \mathbb{R}^3$

$$\begin{cases} z_0 = x \\ \frac{dz_\tau}{d\tau} = g(z_\tau), \quad \tau \in [0, dt], \\ m(x) = z_{dt} \end{cases} \tag{1.7}$$

where $g(z) = [10(z(2) - z(1)), z(1)(28 - z(3)) - z(2), z(1)z(2) - 8/3z(3)]^\top$ for all $z \in \mathbb{R}^3$. The model time increment $dt$ in the above system indicates the level of model nonlinearity. The larger $dt$ the more nonlinear model. On the right panel of Figure 1.4, time series of the true state and the observations of the L63 model with $dt = 0.08$ (respect to 6-hour time step in the atmosphere) are shown.

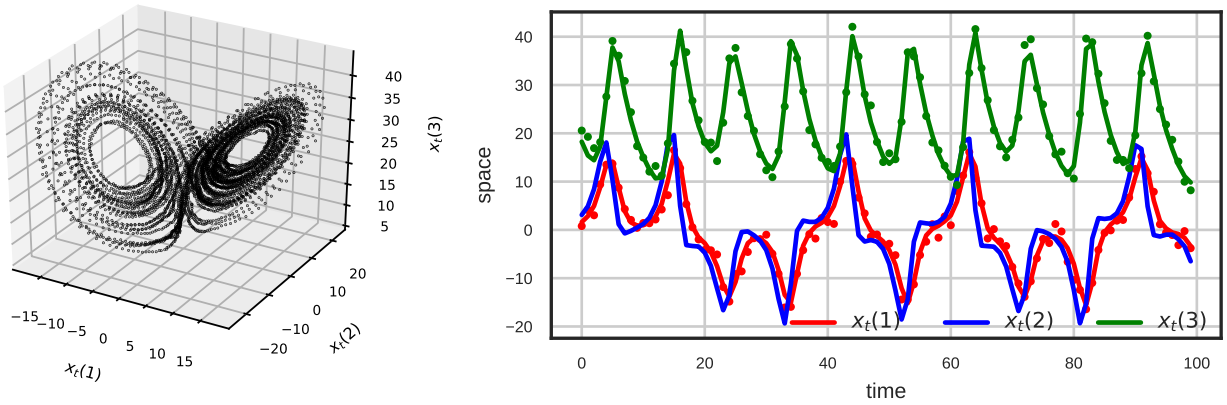


**Figure 1.4** – 3D-Scatter plot (left panel) of the dynamical model with respect to the state, and time series plot (right panel) of the state (lines) and observations (points) simulated from a L63 model (1.6) with error covariances $Q = 0.01I_3$ and $R = 2I_2$. The second component (blue) of the state is unobserved.

### 1.1.2 Filtering and smoothing in state-space models

Given a SSM (1.1) with fixed parameter $\theta \in \Theta$ and a sequence $y_{1:T} = (y_1, y_2, \cdots, y_T)$ of the observation process $(Y_t)_t$, we consider methodologies to infer the state sequence $x_{0:T} = (x_0, x_1, \cdots, x_T)$ of the hidden state process $(X_t)_t$. We focus on classical filtering and smoothing methods which enable to evaluate the corresponding distributions of the state conditional on the observations recursively. In various references of state-space analysis (see [11, 24, 49, 56, 68, 139, 172] for a few), the objectives of these methods are described as follows.

- **Filtering**: compute the marginal distribution (or its joint distribution) of the state given a part of the observation sequence

$$p_\theta(x_t|y_{1:t}) = \frac{p_\theta(y_t|x_t)\ p_\theta(x_t|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})} = \int \frac{p_\theta(y_t|x_t)\ p(x_t|x_{t-1})}{p_\theta(y_t|y_{1:t-1})}\ p_\theta(x_{t-1}|y_{1:t-1})dx_{t-1} \quad (1.8)$$

  where $p(x_t|y_{1:t-1})$ is the so-called prediction distribution or forecast distribution, and $p(y_t|y_{1:t-1})$ is the marginal likelihood or the normalization of the numerator. According to the above recursion, the filtering scheme combines two common steps:

  - *Forecast step* is to propagate the previous filtering distribution with a kernel associated according to the dynamical model (1.1a) (e.g. $p_\theta(x_t|x_{t-1})$).
  - *Correction step* is to assimilate the available observations using the information of the observation model (1.1b) (e.g. $p_\theta(y_t|x_t)$).

- **Smoothing**: compute the marginal distribution (or its joint distribution) of the state given all observations,

$$\begin{aligned} p_\theta\left(x_t|y_{1:T}\right) &= \int p_\theta\left(x_t|x_{t+1}, y_{1:T}\right)\ p_\theta\left(x_{t+1}|y_{1:T}\right) dx_{t+1} \\ &= p_\theta\left(x_t|y_{1:t}\right) \int \frac{p_\theta\left(x_{t+1}|x_t\right)\ p_\theta\left(x_{t+1}|y_{1:T}\right)}{p_\theta\left(x_{t+1}|y_{1:t}\right)}\ dx_{t+1}. \end{aligned} \quad (1.9)$$

  Smoothing is known as the reanalysis of the state given the filtering outputs. Dissimilar to filtering, computing the smoothing distributions of the state is carried out both forward and backward in time. To simulate or estimate the state at an instant time, the reverse phase purposes to adjust the future smoothed state (including future observed information) or its distribution and the filtering outputs (including the past and present observed

information). Therefore, the smoother generally provides better point estimation or simulation of the state than the filter.

There are different methods to compute the filtering and smoothing distributions. In this section, we focus on reviewing popular Kalman-based and particle-based approaches. The Kalman class consists of the original Kalman filter and smoother and their regular extensions (e.g. extended and ensemble Kalman recursions). The latter one contains Sequential Monte Carlo (SMC) methods and their combinations within Markov Chain Monte Carlo (MCMC) contexts.

#### 1.1.2.1 Kalman-based methods

Kalman filter (KF) and smoother (KS) [34,51,74,129,134] are optimal tools in sense of providing the exact filtering and smoothing distributions of linear Gaussian models (1.2). Given all dynamical and observational operators $(M_t, H_t)$ and error covariances $(Q_t, R_t)$, the conditional distributions appearing in the decomposition formulas (1.8) and (1.9) are Gaussian distributions with explicit means and covariances.

Let us denote the forecast [resp. analysis] mean and covariance as $x_t^f$ [resp. $x_t^a$] and $P_t^f$ [resp. $P_t^a$]. To derive the filtering distribution $p_\theta(x_t|y_{1:t})$ with the recursion (1.8), KF first computes the forecast distribution, $p_\theta(x_t|y_{1:t-1}) = \mathcal{N}\left(x_t;\ x_t^f, P_t^f\right)$. Then the correction step of KF using a Kalman gain $K_t$ (a solution of an optimization problem balancing the forecast and the current observation) provides analysis mean and covariance of the filtering distribution. Precisely, KF results $p_\theta(x_t|y_{1:t}) = \mathcal{N}\left(x_t;\ x_t^a, P_t^a\right)$. Expressions of these quantities are presented in Algorithm 1.

As mentioned in the previous section, the smoothing distribution $p_\theta(x_t|y_{1:T})$ can be calculated by the forward-backward recursion (1.9). Under linear Gaussian assumption, KS provides the exact smoothing distribution, $p_\theta(x_t|y_{1:T}) = \mathcal{N}\left(x_t;\ x_t^s, P_t^s\right)$, where $x_t^s$ and $P_t^s$ are denoted as its mean and covariance respectively. Given outputs of the KF, the smoothing distribution is computed by using the Rauch-Tung-Striebel (RTS) formulation (1.12) (see [129]) presented in Algorithm 2. The details of Algorithm 2 and their properties are presented in the literature [74,129,134] and a recent review [24]. Applications of Kalman filters and smoothers in navigation and meteorological DA can be found in [20,39,67].

In order to estimate the filtering and smoothing distributions for nonlinear models (1.3), Kalman-like methods including Extended Kalman filter (EKF) and smoother (EKS) were developed.

**Algorithm 1: Kalman filter (KF)**

- Initialization: set $x_0^a$, $P_0^a$.
- For $t = 1 : T$,
  + **Forecasting**: propagate the previous analysis distribution with

$$\begin{aligned} x_t^f &= M_t x_{t-1}^a, \\ P_t^f &= M_t P_{t-1}^a M_t^\top + Q_t, \end{aligned} \tag{1.10}$$

  + **Correcting**: adjust the forecast with the available observation $y_t$,

$$\begin{aligned} \tilde{y}_t &= y_t - H_t x_t^f, \\ K_t &= P_t^f H_t^\top \left( H_t P_t^f H_t^\top + R_t \right)^{-1}, \\ x_t^a &= x_t^f + K_t \tilde{y}_t, \\ P_t^a &= (I - K_t H_t) P_t^f, \end{aligned} \tag{1.11}$$

  end.

**Algorithm 2: Kalman smoother (KS)**

For $t = T - 1 : 0$,

$$\begin{aligned} J_t &= P_t^a M_{t+1}^\top \left( P_t^f \right)^{-1}, \\ x_t^s &= x_t^a + J_t \left( x_{t+1}^s - x_{t+1}^f \right), \\ P_t^s &= P_t^a + J_t \left( P_{t+1}^s - P_{t+1}^f \right) J_t^\top, \end{aligned} \tag{1.12}$$

end.

The filtering and smoothing schemes are almost the same as in the Kalman algorithms except two points. The forecast/analysis mean estimates in the first formulas of (1.10) and (1.11) are computed using the nonlinear functions $(m, h)$ instead of linear operators $(M_t, H_t)$. Besides, the state transition and observation matrices used in error covariances propagation are locally approximated by their Jacobians

$$M_t = \nabla m\left(x_{t-1}^a\right), \; H_t = \nabla h\left(x_t^f\right).$$

However, running the extended Kalman recursions suffers from computational issues. The algorithms require to compute the model Jacobian $(M_t)_t$ of the ODE system at each time step and huge storage of full forecast covariances $(P_t^f)_t$ in high dimensional models is compulsory as usual.

An alternative to handle these drawbacks is based on Monte Carlo or ensemble-based methods. They include ensemble Kalman filter (EnKF), ensemble Kalman smoother (EnKS) and their variants [14,15,24,56,58,108]. The ensemble-based algorithms implement ensembles of size $N$ to approximate the filtering and the smoothing distributions sequentially. In the EnKF algorithm, all members in the previous ensemble are propagated one by one using the transition kernel for every time step. The covariance matrix $P_t^f$ is then approximated by empirical covariance of the forecast ensemble. The analysis step uses this estimate of $P_t^f$ to compute the Kalman gain and correct each forecasted member. Note that the filtering distributions are not implied directly by using Gaussian assumption with the analysis means and covariances but they are described through ensembles. The EnKS algorithm is run with the ensembles derived from the forward filter. Similar to the previous smoothers, the EnKS uses RTS scheme (1.12) to adjust the analysis ensembles by taking into count both forward and backward observed information. In Eq. (1.12), the product of the analysis covariance and the transpose of the transition matrix is approximated by the empirical cross-covariance of the analysis ensemble at time $(t)$ and the forecast ensemble at time $(t+1)$. The details of the ensemble-based methods are presented in numerous references [24, 56, 58, 108]. In practice, small ensemble size ($N \leq 100$) is typically chosen for approximating the filtering and smoothing distributions. As a result, EnKF, EnKS and their extensions are usually applied in real inference problems, especially geosciences DA [2, 24, 55, 57, 96, 132, 169].

From a practical point of view, the extended and ensemble Kalman-based methods are favorite tools for DA in nonlinear inverse problems. Nevertheless, there exist several issues. For

instance, the Gaussianity of the prediction distributions $\{p_\theta(x_t|y_{1:t-1})\}_t$ assumed to interpret the recursions (1.8) and (1.9) may not be held because of the effects of model nonlinearity. In [93] (see in [54] for numerical illustration), the authors proved that approximations of nonlinear filtering and smoothing distributions derived from these methods do not converge to the Bayesian distributions. Due to that fact, we investigate particle-based methods (SMC and variants) in the thesis.

#### 1.1.2.2 Particle-based methods

This section gives an overview of some regular particle filters (PFs) and smoothers (PSs) which are usually used to treat inference problems for nonlinear SSMs (1.3). Combinations of these SMC samplers and MCMC schemes are then mentioned. Note that Gaussian assumptions of error noises can also be relaxed when working with these particle methods.

a. **Particle filters (PFs)**

Particle filters [23, 47, 48] have been proposed to compute approximations of the filtering distribution $p_\theta(x_t|y_{1:t})$ by a system of particles and their respective weights. A general PF algorithm is run based on a recursion of a joint distribution $p(x_{0:t}|y_{1:t})$ similar to the recursion (1.8) by using Monte Carlo and sequential importance sampling techniques. An approximation of filtering distribution is then deduced as a marginal of the approximation of this joint distribution over variables $x_{0:t-1}$.

Let us denote a system of particles and their corresponding weights $\{x_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1:N}$ which approximates the joint filtering distribution $p_\theta(x_{0:t-1}|y_{1:t-1})$ at time $(t-1)$. The next step of the algorithm consists in generating the new samples $\{x_{0:t}^{(i)}\}_{i=1:N}$ with a proposal kernel $\pi_\theta(x_t|x_{0:t-1}, y_{1:t})$. The correction step computes the corresponding weights $\{w_t^{(i)}\}_{i=1:N}$ of the particles according to the formula

$$W\left(x_{0:t}\right) = \frac{p_\theta\left(x_{0:t}|y_{1:t}\right)}{\pi_\theta(x_t|x_{0:t-1}, y_{1:t})} \overset{(1.8)}{\propto} \frac{p_\theta\left(y_t|x_t\right) p_\theta(x_t|x_{t-1})}{\pi_\theta(x_t|x_{0:t-1}, y_{1:t})} \, p_\theta\left(x_{0:t-1}|y_{1:t-1}\right). \tag{1.13}$$

The entire algorithm is presented in Algorithm 3. Here the resampling step is added in order to reduce impoverishment, a usual problem met in PF algorithms. A systematic resampling method (see others in [45, 77]) can be used to reselect potential particles in $\{x_{0:t-1}^{(i)}\}_{i=1:N_f}$. In this step the filter duplicates particles with large weights and discards particles with small weights.

**Algorithm 3: Particle Filter (PF)**

- Initialization:
  - + Sample $\{x_0^{(i)}\}_{i=1:N_f} \sim p_\theta(x_0)$.
  - + Set initial weights $w_0^{(i)} = 1/N, \forall i = 1:N$.
- For $t = 1:T$,
  - + **Resampling**: draw indices $\{I_t^i\}_{i=1:N}$ with respect to weights $\{w_{t-1}^{(i)}\}_{i=1:N}$.
  - + **Forecasting**: sample new particle,

$$x_t^{(i)} \sim \pi_\theta\left(x_t | x_{0:t-1}^{(I_t^i)}, y_{1:t}\right), \forall i = 1:N.$$

  - + **Weighting**: compute $\tilde{w}_t^{(i)} = W\left(x_{0:t-1}^{(I_t^i)}, x_t^{(i)}\right)$ by using Eq. (1.13) then normalize the weight, $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{i=1}^{N} \tilde{w}_t^{(i)}}, \forall i = 1:N$.

end for.

The above algorithm is referred to as an auxiliary PF algorithm. If the proposal distribution is chosen as

$$\pi_\theta(x_t | x_{0:t-1}, y_{1:t}) = p_\theta(x_t | x_{t-1}) \tag{1.14}$$

we get a simple filter so-called bootstrap PF. It is the quite usual choice for numerical experiments in statistics and applications [4, 49, 70, 95, 124, 169] and it is used in the thesis for most of numerical illustrations. Another popular proposal kernel is

$$\pi_\theta(x_t | x_{0:t-1}, y_{1:t}) = p_\theta(x_t | x_{t-1}, y_t) \tag{1.15}$$

leading to an optimal PF. With this choice, variance of the importance weight (1.13) conditional on $x_{0:t-1}$ and $y_{1:t}$ are constant and the particles are pushed towards the observations. That may be helpful for filtering in high dimensional models where the bootstrap filter easily degenerates. Details of the discussion on the choice of the proposal kernel $\pi_\theta$ can be found in [23, 48, 123, 145].

By using the PF algorithm (3), $p_\theta(x_{0:t}|y_{1:t})$ is approximated by

$$\widehat{p}_\theta\left(x_{0:t}|y_{1:t}\right) = \sum_{i=1}^{N} \delta_{x_{0:t}^{(i)}}\left(x_{0:t}\right) w_t^{(i)} \tag{1.16}$$

where $\delta$ is dirac distribution of $x$. Asymptotic properties of this estimator were given in [32, 40]. In last decades, PFs have been used to handle various inference problems in

statistics, oceanography, informatic technology, biology, economics, mechanical learning, etc [5, 70, 92, 115, 138, 158].

b. **Particle smoothers**

One simple way to compute the smoothing distribution $p_\theta(x_t|y_{1:T})$ as well as its joint distribution $p_\theta(x_{0:T}|y_{1:T})$ is based on the complete run of a PF. At the final time step, the system of particles and weights $\{x_{0:T}^{(i)}, w_t^{(i)}\}_{i=1:N}$ approximates the joint distribution $p_\theta(x_{0:T}|y_{1:T})$. Hence a smoothing distribution at an instant time $t$ can be depicted by $\{x_t^{(i)}, w_T^{(i)}\}_{i=1:N}$. However, this naive approach gets degeneracy issues when the number of observations ($T$) is large. The resampling step in the filter may lead to poor samples which contain lots of particles sharing the same values.

Forward filter-backward smoother (FFBS) based on the recursion (1.9) is presented in [17, 46] to reduce degeneracy in estimating $p_\theta(x_t|y_{1:T})$. After running a forward filter, the backward pass aims at re-weighting the particles $\{x_t^{(i)}\}_{i=1:N}$ by

$$w_t^{s,i} = w_t^{(i)} \sum_{j=1}^{N} \frac{p_\theta\left(x_{t+1}^{(j)}|x_t^{(i)}\right) w_{t+1}^{s,j}}{\sum_{i=1}^{N} p_\theta\left(x_{t+1}^{(j)}|x_t^{(i)}\right) w_t^{(i)}} \tag{1.17}$$

then the smoothing distribution is approximated by

$$\widehat{p}_\theta\left(x_t|y_{1:T}\right) = \sum_{i=1}^{N} \delta_{x_t^{(i)}}\left(x_t\right) w_t^{s,i} \tag{1.18}$$

In inference problems (e.g. parameter estimation) involving the joint smoothing distribution $p_\theta\left(x_{0:T}|y_{1:T}\right)$, backward simulation (BS) proposed by [69] is considered as a natural technique to simulate realizations of the state given the (forward) filter outputs. The sampler works based on the decomposition

$$p_\theta\left(x_{0:T}|y_{1:T}\right) = p_\theta\left(x_T|y_{1:T}\right) \prod_{t=0}^{T-1} p_\theta\left(x_t|x_{t+1}, y_{1:t}\right), \tag{1.19}$$

where the so-called backward kernel is defined as

$$p_\theta(x_t|x_{t+1}, y_{1:t}) \propto p_\theta\left(x_{t+1}|x_t\right)\ p_\theta\left(x_t|y_{1:t}\right). \tag{1.20}$$

Given the particles $(x_t^{(i)})_{t=0:T}^{i=1:N}$ and the weights $(w_t^{(i)})_{t=0:T}^{i=1:N}$ of the PF algorithm, the smoothing trajectories can be sequentially drawn from an estimate of the backward kernel (1.20). Other smoothers can be found in the recent reviews [64, 86]. The complete algorithm of BS and details are presented in the next chapters.

c. **Particle Gibbs samplers**

Particle Gibbs samplers, a branch of Particle Markov Chain Monte Carlo (PMCMC) approaches, are combinations of SMC and MCMC methods. These samplers permit to iteratively simulate realizations of sophisticated or high dimensional distributions (e.g. the nonlinear distribution $p_\theta(x_{0:T}|y_{1:T})$).

In the particle Gibbs samplers, one trajectory $X^* = (x_0^*, x_1^*, \cdots, x_T^*) \in \mathcal{X}^{T+1}$, so-called conditioning trajectory, is set as a prior. It is replaced for one of particle paths, e.g. $x_{0:T}^{(N)}$, and joint with other particle paths in an SMC-like scheme. After every iteration, the conditioning is updated by one of the particle paths generated from the SMC-like sampler. The procedure is then repeated, and this leads to construct Markov kernels leaving an invariant distribution which is exactly the smoothing distribution $p_\theta(x_{0:T}|y_{1:T})$. The most interesting property of the particle Gibbs samplers is that, given an arbitrary conditioning path, these samplers, with a fixed number of particles and a significantly large number of iterations, generate realizations distributed according to the smoothing distribution. For instance, as illustrated in the literature of Bayesian inference [4, 98, 99, 101, 102, 150], such approaches using a low fixed number of particles $(5 - 10^2)$ gives similar results as standard PSs using many particles $(10^2 - 10^6)$.

Conditional particle filter (CPF) is the first particle Gibbs sampler (also named as conditional SMC sampler) appeared in a discussion of Andrieu et al. [4]. This sampler is based on PF only so that the CPF approach typically gets path degeneracy as usual. $N$ trajectories often share the same ancestors when the length of the observation sequence is large. Moreover, in the CPF, the $N^{th}$-path is frozen for the conditioning while the other paths are broken due to resampling. Both issues may lead to generate the same realizations of the state and hence provide a poor approximation of the smoothing distribution. Such a problem is called slow mixing. In [99, 100, 102], Lindsten et al. proposed a new sampler, Conditional Particle Filtering-Ancestor Sampling (CPF-AS). The CPF-AS algorithm is almost the same as the CPF. Instead of fixing the conditioning path at $N^{th}$ position, the authors proposed to resample each of its indices $(I_t^N)_t$ given the current conditioning

particle and the observations up to time $t$. This strategy permits to renew the ancestral link of the conditioning and hence improves the mixing. Sequentially, CPF-AS produces a better approximation of the filtering distribution than the one obtained by the CPF. The advancement of the CPF-AS in practice is numerically illustrated in [30, 99, 102]. Theoretical behaviors of the CPF also hold for the CPF-AS. They were studied in [99]. Applications of CPF and CPF-AS for simulation of the smoothing distribution can be found in [4, 150].

However, the degeneracy problem in the CPF-AS sampler may still cause a poor approximation of $p_\theta(x_{0:T}|y_{1:T})$ (except a short sequence of observations is considered). In [101, 102], backward simulation was proposed to be combined with the particle Gibbs sampler, leading to Conditional Particle Filter-Backward Simulation (CPF-BS) smoother (see in Chapter 3 for more details).

### 1.1.3 Parameter estimation

In real applications, parameters ($\theta \in \Theta$) in SSM (1.1) are usually unspecified and using incorrect values of $\theta$ may lead to bad reconstruction results. This is illustrated on Figure 1.5 for reconstruction of the state in the sinus model (1.4) with $\theta = (Q, R)$. Smoothing with the right parameter values provides a good approximation of the true trajectory (left panel) whereas smoothing with wrong parameter values gets large biases and variances in simulating the state distributions (right panel). Therefore, identifying a reasonable value of $\theta$ before filtering or smoothing is necessary. A nice explanation of the problem was also given in [10]. In this section, we give a review of off-line likelihood-based methods which are widely used for parameter estimation given a sequence of observations $y_{1:T}$ and the model (1.1).

Likelihood-based methods for parameter identification include Bayesian inference and maximum likelihood estimation. These methods can be found in recent reviews [86, 152].

- **Bayesian inference**

  The Bayesian approach aims at inferring an arbitrary parameter by simulating from the joint distribution of the state and the parameter. Additionally, it is able to describe the shape of parameter distribution which might be multi-modal. But the Bayesian approaches still have some drawbacks. First, a very large number of iterations is required to get good approximations of the parameter distributions if a standard MCMC method (see e.g. [4, 86, 102]) is used. In [147, 148, 165] the authors proposed Bayesian approaches combined

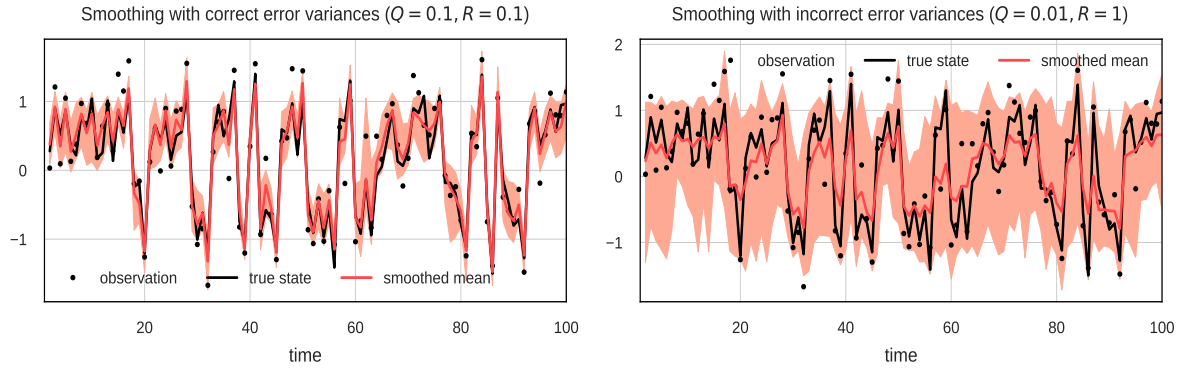


**Figure 1.5** – Impact of values of paramater $\theta = (Q, R)$ on smoothing distributions for the sinus model (1.4). The true state and observations have been simulated with the true value $\theta^* = (0.1, 0.1)$. The mean of the smoothing distributions are computed using a standard particle smoother [46] with 100 particles. Results obtained with the true parameter values $\theta^* = (0.1, 0.1)$ (left panel) and wrong parameter values $\tilde{\theta} = (0.01, 1)$ (right panel) are shown.

with EnKF algorithms and obtained approximations of the parameter distributions with a low number of members and iterations. However, simulating the distributions in high-dimensional SSMs is sometimes impractical. For example, it is difficult to simulate directly the full model covariance $Q$ which involves a lot of parameters if the latent state has values in a high dimensional space. To simplify the problem, $Q$ is typically supposed to have a predefined form, such as the multiplication of a scalar and a given matrix, and only the scale factor is estimated.

- **Maximum likelihood estimation**

  There are two major approaches in the statistical literature to maximize numerically the likelihood of models with latent variables: gradient ascent and Expectation-Maximization (EM) algorithms. Between these two approaches, the gradient ascent seems less efficient in several circumstances, for instance, *gradient ascent algorithms can be numerically unstable as they require to scale carefully the components of the score vector* as that stated in [86]. The EM approach is more favoured when considering complicated models such as the ones used in DA. The first EM algorithm was suggested by [42]. Various variants of the EM algorithm were proposed in the literature (see e.g. [28, 50, 86, 98, 110, 121, 126, 141, 152, 156, 164] and references therein). The common idea of these algorithms is to run an iterative procedure where an auxiliary quantity (1.21) which depends on the smoothing distribution

is maximized at each iteration until a convergence criterion is reached.

$$\mathbb{E}_{\theta'}\left[\ln\ p_\theta\left(x_{0:T}, y_{1:T}\right)\right] \triangleq \int \ln p_\theta\left(x_{0:T}, y_{1:T}\right)\ p_{\theta'}\left(x_{0:T}|y_{1:T}\right) dx_{0:T} \tag{1.21}$$

where $\theta'$ denotes a given value of the unknown parameter $\theta$.

Starting from an initial parameter an iteration of the EM algorithm has two main steps:

- **E-step**: compute the smoothing distribution $p_{\theta'}\left(x_{0:T}|y_{1:T}\right)$ given the observations $y_{1:T}$ and the parameter value $\theta'$, and deduce the auxiliary quantity (1.21),
- **M-step**: update the parameter value by maximizing the function (1.21) of $\theta$.

It can be shown that this procedure leads to increase the likelihood function $p_\theta(y_{1:T})$ at each iteration and gives a sequence of parameter values which converges to a local maximum of the likelihood.

For linear models, e.g. (1.2), the EM algorithm combined with Kalman smoothing (KS-EM, [143]) has been the dominant approach to estimate parameters. In the case of nonlinear and/or non-Gaussian models, e.g. (1.3), the expectation (1.21) under the distribution $p_{\theta'}(x_{0:T}|y_{1:T})$ is usually intractable and the EM algorithm cannot work in such situation. An alternative, originally proposed in [28] and [29], is to use a Monte Carlo approximation of (1.21) or stochastic versions of the EM [41].

To implement such procedures, it is necessary to generate samples of the smoothing distribution. A classical alternative in many applications consists in using the EnKS algorithm [58] leading to the EnKS-EM algorithm [50, 126, 156]. However, the EnKS approximation does not converge to the exact distribution $p_\theta(x_{0:T}|y_{1:T})$ for nonlinear state-space models [93]. In the literature [86,89,116,121,141], standard or approximate particle smoothing methods are generally used. Nevertheless, they demand a huge amount of particles to get a good approximation of the target probability distribution.

As mentioned in the previous section, conditional particle smoothers (CPF, CPF-AS, CPF-BS) [4,99,101,102,150] are able to simulate the smoothing distribution with a fixed number of particles. These samplers can be promising tools when combined with the iterative EM machines. In [98,102,149], the authors proposed to use a CPF-AS sampler within EM-like algorithms. However CPF-AS suffers from degeneracy (the particle set reduces to a very few effective particles) and consequently the estimated parameters of CPF-AS have bias

and/or large variance. In Chapter 3, we investigate the combination of CPF-BS and EM algorithms.

## 1.2 Inference in non-parametric state-space models

### 1.2.1 Non-parametric state-space models

Non-parametric SSMs are SSMs (1.3) where the dynamical model and/or the observation model are unknown and approximated by non-parametric estimators. An example of such models is the case of wind data recorded at five stations located in the North-West region of France (see on Figure 1.6). Due to failures of the collection process, instrumentals, and model formulation, the data may include observational errors and lots of gaps (e.g. at Brignonan). In order to infer

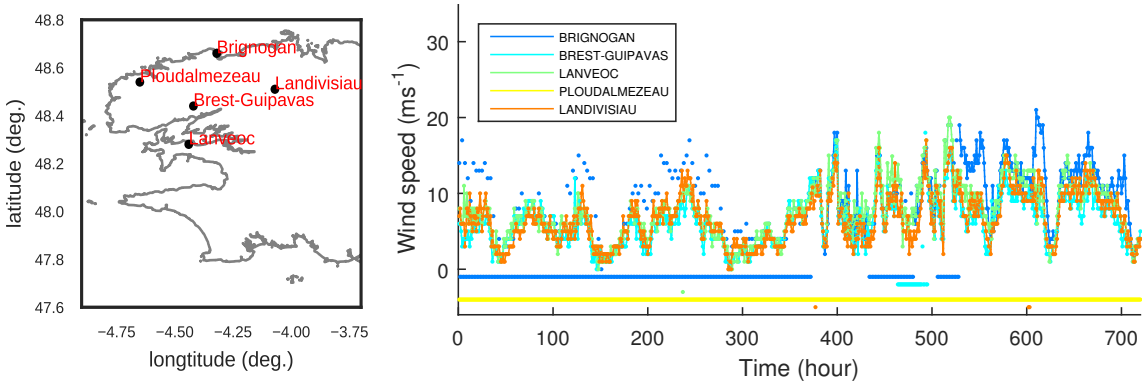


**Figure 1.6** – An illustration of wind data with gaps recorded at five stations in the North-West of France (produced by Météo France). Left panel: location map of the stations, right panel: time series of wind speed where the missing entries are shown by negative values.

the state given the noisy missing data, we can define the following non-parametric SSM model

$$\begin{cases} X_t = m(X_{t-1}) + \eta_t, \\ Y_t = H_t X_t + \epsilon_t, \end{cases} \tag{1.22}$$

where $(X_t)_t$ is the hidden state process of the wind system which we would like to retrieve, $(Y_t)_t$ stands for the observation process represented by the observed data $y_{1:T}$, and the error noises has Gaussian distributions $\mathcal{N}(0, Q_t)$ and $\mathcal{N}(0, R_t)$ respectively. In the above model, $m, Q_t, R_t$ are unknown and the adaptive observation operator $H_t$ describes the situation where some state components can be missing. For instance, if all components in the state variable are observed at time $t$, $H_t$ is an identity matrix $I_d$ ($d$ is the fixed dimension of the state variable $X_t$), and if only

the first component is observed $H_t$ is set by the first row vector of $I_d$. The size of observational error covariance ($R_t$) depends on the dimension of the observation ($Y_t$).

When working with such unknown models, classical approaches often use a simpler parametric model to replace $m$. However, it is generally difficult to identify an appropriate parametric model which can reproduce all the complexity of the phenomenon of interest. Nowadays, there exists a huge amount of historical datasets recorded using remote and in-situ sensors or obtained through numerical simulations and this promotes the development of data-driven approaches. Non-parametric SSMs were first appeared and analyzed in oceanographical DA [95, 154, 155]. In next section, we first present local regression methods (so-called analog methods in real applications, e.g. meteorological prediction [7, 8, 78, 175, 183]) learned on a historical dataset, which is used to estimate the dynamics. These non-parametric emulators are combined within the proposed algorithms to solve inference problems for non-parametric SSMs with unknown dynamics in the next chapters.

### 1.2.2 Data-driven forecast emulators in non-parametric state-space models

Suppose that a sequence $x_{0:T}$ of the state process $(X_t)_t$ in (1.3) is available. This section involves in presenting non-parametric estimates of $m$ at a given point $x$ (transition mean $E(X_t|X_{t-1} = x)$) and present several sampling methods for the transition kernels.

#### 1.2.2.1 Local regression for $m$ estimation

a. **Local constant method**

Local constant regression (LCR), known as Nadaraya-Watson kernel regression (NW), has been used to approximate the value of $m$ at a given $x$. In the literature [62], an estimate of $m$ is expressed by

$$\widehat{m}(x) = \sum_{t=1}^{T} \frac{x_t \, \mathcal{K}_h \left(x_{t-1} - x\right)}{\sum_{t=1}^{T} \mathcal{K}_h \left(x_{t-1} - x\right)}. \tag{1.23}$$

where $\mathcal{K}_h(u)$ is a chosen kernel with a bandwidth $h$. In practice, the method is applied in lots of areas because of its simplicity. For instance, Rajagopalan [128] resampled the vector of Utah daily weather variables conditionally on the data of the previous day. In [175] the author recommended using analog forecast learned on a 30-year historical dataset to simulate European daily mean temperature, see other application in [7]. Though this method is quite attractive in forecasting, it still gives a poor estimation of the model $m$

in some situations. Successors estimated by this emulator are always held in the range of the learning data. It is unable to correctly capture outliers and/or extreme values which often occur in natural phenomena.

b. **Local polynomial regression**

Local polynomial regression (LPR) proposed in [60, 61] is an alternative. The idea is to approximate the dynamical model $m$ by Taylor's expansion (1.24),

$$m\left(x'\right) \approx m(x) + \sum_{j=1}^{p} \frac{\nabla^j m(x)}{j!}(x'-x)^j \triangleq M_{0,x} + \sum_{j=1}^{p} M_{j,x}(x'-x)^j \tag{1.24}$$

where $\{\nabla^j m(x)\}_{j=1:p}$ are derivatives at point $x$ and $x'$ lives in a neighborhood of $x$. In order to obtain estimates of $m(x)$ and its derivatives, the coefficients $\{M_{j,x}\}_{j=0:p}$ are computed by minimizing the following least square error

$$\left\{ \sum_{t=1}^{T} \left\| x_t - \left( M_{0,x} + \sum_{j=1}^{p} M_{j,x}(x_{t-1}-x)^j \right) \right\|^2 W_h\left(x_{t-1}-x\right) \right\} \tag{1.25}$$

In formula (1.25), $W_h$ is a normalized weight function given by a smoothing kernel $\mathcal{K}_h$ with a bandwidth $h$. Such a kernel makes the role of choosing neighbors around $x$ such that the local estimate of $m$ is more precise.

In the cases where the dynamical function $m$ is approximated by the first-order of the Taylor's expansion (1.24), LPR method is referred to as Local Linear Regression (LLR) [61]. The method is also widely used in forecasting because of its simplicity (only two parameters required to be estimated) and efficiency (compared to LCR). For instance, Fan et al. [63] implemented LLR to estimate coefficients adapting for data of CD4 cells (vitals in the immune system). LLR was also used to fit wind power data in [122]. Generally, an estimate of the dynamical function $m$ is obtained by solving the least square problem (1.25) with respect to LLR coefficients ($M_{0,x}$ and $M_{1,x}$). It yields

$$\widehat{m}(x) = \widehat{M}_{0,x} = \sum_{t=1}^{T} x_t W_h\left(x_{t-1}-x\right) - \widehat{M}_{1,x} \sum_{t=1}^{T} \left(x_{t-1}-x\right) W_h\left(x_{t-1}-x\right), \tag{1.26}$$

where an estimate of the gradient $\nabla m(x)$ is

$$\widehat{M}_{1,x} = -\left[\sum_{t=1}^{T} x_{t-1}x_{t-1}^{\top}W_h\left(x_{t-1}-x\right) - \sum_{t=1}^{T} x_{t-1}W_h\left(x_{t-1}-x\right)\sum_{t=1}^{T} x_{t-1}^{\top}W_h\left(x_{t-1}-x\right)\right]^{-1} \times \left[\sum_{t=1}^{T} x_t x_{t-1}^{\top}W_h\left(x_{t-1}-x\right) - \sum_{t=1}^{T} x_t W_h\left(x_{t-1}-x\right)\sum_{t=1}^{T} x_{t-1}^{\top}W_h\left(x_{t-1}-x\right)\right]. \tag{1.27}$$

A comparison of LCR and LLR on the univariate model (1.4) is shown on Figure 1.7. LCR method gives a large bias estimate of the dynamical model, especially in its tails, when the learning data is not informative enough. Thanks to estimation ability of the slope, LLR permits to retrieve reasonable estimates in such poor situations. Asymptotic behaviors of LCR and LLR estimates related to these numerical results can be found in [38, 60, 103].

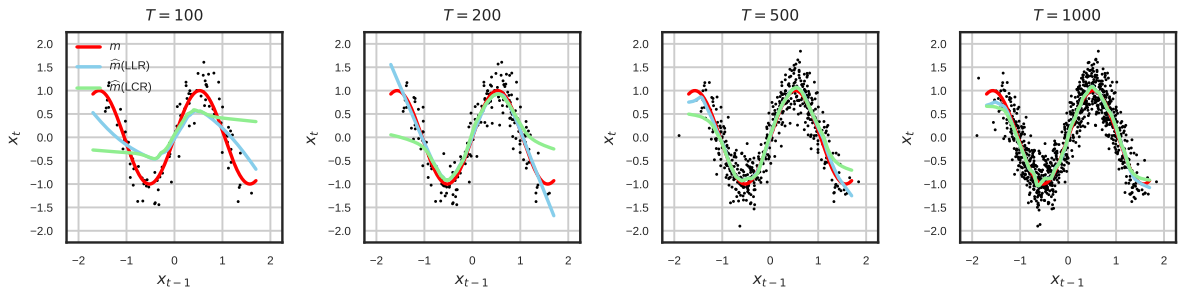


**Figure 1.7** – Comparison of LCR and LLR methods in estimation of the dynamical model $m$ on learning sequences of the state process $\{X_t\}_t$ of the sinus SSM (1.4) with $Q = R = 0.1$. The length of the learning data $T$ varies in $[100, 1000]$ from left to right. Scattered points stand for the relation between two successive values in the learning sequences.

#### 1.2.2.2 Kernel and bandwidth selection

The choice of kernel $\mathcal{K}_h$ and its bandwidth $h$ is very important in model estimation [61, 75, 144, 159]. The Epanechnikov and tricube kernels are the most applicable since both of them have compact supports which help to avoid learning the points far away from $x$. Following the work of [35], the tricube kernel (1.28) is more preferable in holding the derivative properties at kernel boundaries.

$$\mathcal{K}_h(x) = \frac{70}{81}\left(1 - \frac{\|x\|^3}{h^3}\right)^3 \mathbf{1}\left(\|x\| \leqslant h\right). \tag{1.28}$$

By using this kernel, the bandwidth $h$ is chosen as the radius of the compact support of the learning data $x_{0:T}$. When the model is nonlinear, using total points in the given data is useless. This work may easily increase the bias of estimates, moreover, require large storage space for computing regression coefficients. An alternative was proposed in [7, 60, 91, 119, 162] where the regression coefficients are learned on $n$-nearest neighborhoods and $h$ is thence set as the radius of every $x$'s neighborhood adaptively. Note that if number of nearest neighbors $n$ is large the bias of LLR estimates may be high, by contrast, if few neighbors are taken the variance associated with the estimates is large. A popular method to compute an optimal value of $n$ is normally based on a grid-search. The best number of neighbors is chosen on such a way that a loss function, e.g. root of mean square error (RMSE), between the true forecasts and their estimates reaches extreme values.

#### 1.2.2.3 Sampling methods

In many applications, not only the dynamical model $m$ but also the distribution of the model noise $\{\eta_t\}_t$ is of the interest. When the noise distribution is known the transition kernels $\{p(x_t|x_{t-1})\}_t$ can be deduced, consequently. Here we consider two situations of the model noise distribution: satisfying Gaussian assumption (as well as other parametric family assumptions) and otherwise. The Gaussian case is the most usual case in practice (e.g. in meteorological DA). With this assumption, the transition kernels have Gaussian distributions with means and covariances dependent on $m$ and $Q$ (if other parametric family distributions are considered, the kernels are identified with their certain parameters). In the case that these quantities or relevant static parameters are unknown, they are usually estimated by using an optimization algorithm (e.g. EM algorithm). In a particular case, covariance $Q$ depends on each value $x$, it can be estimated by

$$\widehat{Q}(x) = \sum_{t=1}^{T} \left[x_t - \widehat{m}\left(x_{t-1}\right)\right] \left[x_t - \widehat{m}\left(x_{t-1}\right)\right]^\top W_h\left(x_{t-1} - x\right). \tag{1.29}$$

where $\widehat{m}$ is an estimate of $m$ (see Eq. (1.26)). Other estimation methods can be found in [31,61,177]. By contrast, if the Gaussian assumption is unreliable we can use resampling methods [6,91,135] such as local bootstrap to generate the transition distributions. Briefly, to sample from the transition kernel conditionally on the value $x$, the residuals $(x_t - \widehat{m}(x_{t-1}))$ are resampled with respect to the local weights $\{W_h(x_{t-1} - x)\}_t$. Then a forecast sample is defined as a collection of the resampled residuals taking into account the deterministic estimate value $\widehat{m}(x)$ of the model.

### 1.2.3 Discussion

In practice, historical datasets recorded using remote and in-situ sensors usually take into account observational errors. A simple approach to estimate $m$ would consist in computing the non-parametric estimate $\widehat{m}$ based on the sequence $y_{1:T}$ instead of a sequence of the process $\{X_t\}$ but this is not satisfactory since the conditional distributions of $X_t$ given $X_{t-1} = x_{t-1}$ and $Y_t$ given $Y_{t-1} = y_{t-1}$ do not coincide. This is illustrated on Figure 1.8 obtained using a nonlinear univariate SSM defined in (1.4). The left plot shows a scatter plot of the true state $(X_{t-1}, X_t)$ and a non-parametric estimate $\widehat{m}$ obtained using LLR which is reasonably close to $m$. The right plot shows a scatter-plot of the observed sequence $(Y_{t-1}, Y_t)$. Note that $Y_t$ is obtained by adding a random noise to $X_t$ and this has the effect of blurring the scatter plot by moving the points both horizontally and vertically. The blue curve shows a non-parametric estimate of $E[Y_t|Y_{t-1}]$ obtained using LLR, which is a biased estimate of $m$. In a regression context, it is well known from the literature on errors-in-variables models that observational errors in covariates lead, in most cases, to a bias towards zero of the estimator of the regression function (see [26]). One of the classical approach to reduce the bias is to introduce instrumental variables which help to get information about the observational error. This approach has been adapted for linear first-order autoregressive models in [111] and further studied in [94]. Besides, [26] gave an overview of different methods to build consistent estimators in the context of regression. Among them, we notice the local polynomial regression and the Bayesian method for non-parametric estimation. But, as far as we know, they are not generalized for time series estimation.

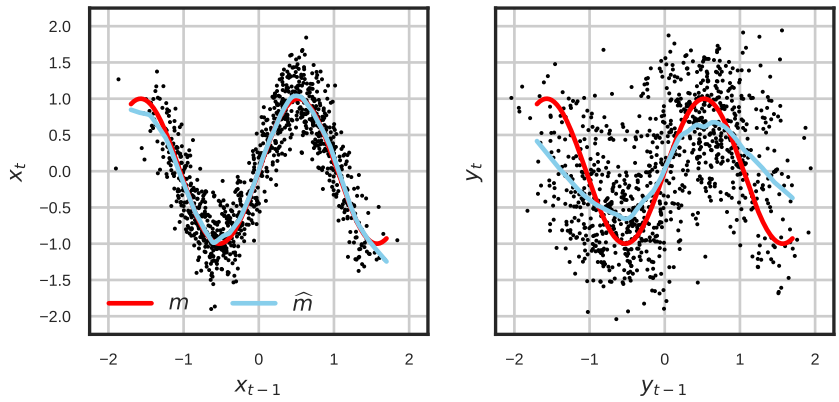


**Figure 1.8** – Scatter plots of $(X_{t-1}, X_t)$ (left) and $(Y_{t-1}, Y_t)$ (right) for the sinus SSM (1.4) with $Q = R = 0.1$. The blue curves represent for estimates of the conditional means obtained using LLR and the red curves represent for the true $m$ functions.

In the thesis, we target to develop novel methods for both model reconstruction and parameter estimation given the noisy data.

Chapter 2

# Non-parametric filtering in nonlinear state-space models.

In this chapter, we present non-parametric filtering algorithms for reconstruction of the hidden state in nonlinear SSMs given a historical dataset and an observation sequence derived from simulations of the state process and the observation process, respectively. The proposed algorithms consist in combining an LLR estimate of the dynamical model, learned on the historical dataset, within the classical filtering schemes. Numerical experiments are the main contribution of this chapter for comparisons in terms of reconstruction quality and computational costs between

- the classical approaches using the true dynamical model (see e.g. in [11,23,24,54,56,169]) and the proposed approaches using non-parametric estimates of the model,
- the non-parametric approaches using LCR estimates of the model (presented in [95,155]) and the proposed approaches using LLR estimates of the model,
- the proposed approaches using LLR estimates of the model within different filtering schemes (EKF, EnKF, bootstrap and optimal PF).

## 2.1 Introduction

Sequential data assimilation (DA) methods [3,11,24,120,169] are extensively used to approximate the state of environmental systems from noisy (partial) observations in geosciences. These methods are formulated underlying SSMs.

In numerous DA problems, nonlinear SSMs as (1.3) consisting of a nonlinear dynamical model and a linear observation model are usually considered. And the hidden state therein is

often estimated by running one of the classical filters (EKF, EnKF, and PF) given a sequence of the observations and a model. EKF [83] is used when the dynamical model is locally close to Gaussian linear model. This filter permits to compute filtering mean and covariance at a low computational cost. When the model is nonlinear or the dimension of the state is high (e.g. in geoscience DA), EnKF [58, 108] is the most suitable tool for point estimattion of the state. It typically requires a few members to approximate the filtering distribution. However, the EnKF approximation does not converge to the Bayesian filtering distribution [93]. PF algorithms [23,47,47,48] are alternatives for filtering in nonlinear (non-Gaussian) SSMs. Despite the need of lots of particles for converging the PF algorithms are very efficient in inference problems where the simulation of conditional distributions of the state is necessary.

A key feature of these classical DA algorithms is that they repeat the integration of an explicitly known ODE system of the dynamic. Particularly, such a numerical forecast model is intensively expensive in EnKF and PF algorithms since it is run for each member/ particles. Nowadays, a large amount of observations allows replacing the numerical model by non-parametric estimates. This substitution may have several advantages in reducing the computational cost and providing a better description of the real dynamics. Moreover, the non-parametric approaches are more flexible in local or regional DA problems where only some components of the state variable involved in an ODE system are of the interests and parametric estimates can focus on the chosen components. In [95,154,155], Tandeo et al. have recently proposed the combination of different local regression emulators (LCR, LLR) [35,61,63,71] (so-called analog emulators in geociences [7,8,78,175,183]) and DA algorithms (EnKF, EnKS and bootstrap PF). These novel methods have been applied to reconstruct the state of oceanographical systems in [59,153].

Since LLR generally performs the model estimation better than the LCR (see in Section 1.2.2.1), we propose to combine the LLR method with the filtering algorithms. As an extension of the works [95,154,155], this chapter introduces the non-parametric EKF and optimal PF algorithms. Furthermore, numerous numerical experiments will be carried out to compare the reconstruction performances and computational costs of the mentioned methods.

The chapter is organized in four sections. Section 2.2 introduces the proposed non-parametric EKF and PF algorithms using LLR estimate of the dynamics. The non-parametric EnKF proposed in [95, 154, 155] is reminded. We also list some advantages and disadvantages of these non-parametric approaches. In Section 2.3, numerical result are illustrated on the L63 model (1.6). Section 2.4 finally includes conclusions and perspectives.

## 2.2 Non-parametric filtering algorithms

Let us assume that a sequence $y_{1:t} = \{y_1, y_2, \cdots, y_t\}$ of the observation process $\{Y_t\}_t$ and a learning sequence of the state process $\{X_t\}_t$ are given. This section aims at introducing non-parametric filtering algorithms to estimate $\{p(x_t|y_{1:t})\}_t$, the conditional distributions of the hidden state given the observations up to time $t$ for the nonlinear SSM (1.3) where the dynamical model $m$ is unknown or analytically intractable, and the observation model is linear $h(x_t) = H_t x_t$.

Firstly, the non-parametric EKF algorithm is presented in Section 2.2.1. Here we discuss the use of LLR in estimating both the model $m$ and its first derivative function. Next, we remind the combination of LLR forecast emulator and the EnKF algorithm (proposed in [95, 155]) in Section 2.2.2. In a sequel, Section 2.2.3 introduces the combination of LLR forecast emulator and the PF algorithm using the bootstrap proposal kernel (1.14) or the optimal kernel (1.15).

### 2.2.1 Extended Kalman filter

The Extended Kalman filter (EKF) (see in [74, 83]) is known as an extension version of the KF (Algorithm 1) for estimating the filtering distributions $\{p(x_t|y_{1:t})\}_t$ of nonlinear Gaussian models (1.3) whose two first model derivatives can be approximated locally. When conditional distributions of the state given the observations (e.g. $p(x_t|y_{1:t-1})$) are Gaussian, EKF recursively computes approximations of the filtering mean $\mathbb{E}(X_t|y_{1:t})$ and covariance $\mathbb{V}(X_t|y_{1:t})$, denoted by $x_t^a$ and $P_t^a$. Sequentially, the filtering distribution is approximated by

$$\widehat{p}(x_t|y_{1:t}) = \mathcal{N}(x_t;\ x_t^a, P_t^a). \tag{2.1}$$

A classical EKF algorithm runs based on a two-step procedure (forecasting and correcting). In the forecast step, the forecast mean $x_t^f$ is computed as a propagation of the corrected mean $x_{t-1}^a$ via $m$ and the forecast covariance $P_t^f$ is obtained by a linear approximation of Eq. (1.10). This is summarized in the following scheme.

$$\begin{cases} x_t^f = m(x_{t-1}^a), \\ P_t^f = \nabla m(x_{t-1}^a) P_{t-1}^a \nabla m^\top(x_{t-1}^a) + Q_t. \end{cases} \tag{2.2}$$

Then, the correction step is the same as Eq. (1.11) in the KF algorithm.

When the dynamical function $m$ and its gradient function $\nabla m$ are unknown or intractable, we propose to substitute LLR estimates (Eq. 1.26 and Eq. 1.27) for $m$ and $\nabla m$ in the forecast step (2.2) of the classical EKF algorithm, leading to Algorithm 4.

**Algorithm 4: Extended Kalman Filter (EKF) with LLR forecasting**

- Initialization: set $x_0^a$, $P_0^a$.
- For $t = 1 : T$,
  + **Forecasting**: propagate the previous analysis mean and covariance

$$\begin{aligned} x_t^f &= \widehat{m}\left(x_{t-1}^a\right), \\ P_t^f &= \widehat{M}_{1,x_{t-1}^a} P_{t-1}^a \widehat{M}_{1,x_{t-1}^a}^\top + Q_t, \end{aligned} \tag{2.3}$$

  where $\widehat{m}(x)$ and $\widehat{M}_{1,x}$ are LLR estimates (see Eq. 1.26 and Eq. 1.27) of the dynamical function and its gradient at any values of $x$.
  + **Correcting**: adjust the forecast with the available observation $y_t$

$$\begin{aligned} \tilde{y}_t &= y_t - H_t x_t^f, \\ K_t &= P_t^f H_t^\top \left(H_t P_t^f H_t^\top + R_t\right)^{-1}, \\ x_t^a &= x_t^f + K_t \tilde{y}_t, \\ P_t^a &= (I - K_t H_t) P_t^f, \end{aligned} \tag{2.4}$$

  end.

One typical advantage of the algorithm is that it quickly computes approximations the filtering distributions in low-dimensional nonlinear Gaussian models. For high dimension problems, the extended Kalman recursions require huge storage for the full covariances $(P_t^f, P_t^a)_t$ as usual and EnKF may be more efficient.

### 2.2.2 Ensemble Kalman filter

Ensemble Kalman filter (EnKF) (see the origin and its variants in [14, 15, 24, 56, 81]) is a Monte Carlo approximation of the KF which enables to handle high dimensional filtering problems. For each time step, an ensemble of size $N$, denoted by $\{x_t^{a,(i)}\}_{i=1:N}$, is run and an estimate of the filtering distribution is deduced as follows

$$\widehat{p}\left(x_t | y_{1:t}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_t^{a,(i)}}\left(x_t\right). \tag{2.5}$$

This ensemble-based method does not require local linearity of the filtering distributions or a huge amount of members for convergence. Additionally, EnKF uses the $N$-ensemble to compute

an empirical covariance of the forecast covariance that avoids using huge computational memory as in the EKF. Therefore EnKF is often used in practical DA problems [3, 56, 80, 120, 180].

In the classical EnKF algorithm, the forecast step generates new members, denoted as $\{x_t^{f,(i)}\}_{i=1:N}$, by using the transition kernel $p(x_t|x_{t-1}^{a,(i)})$. The correction step then obtains the analysis $x_t^{a,(i)}$ by minimizing the error between the forecast and the observation via a Kalman gain. In the combination with LLR method, the estimate (1.26) of $m$ is used to construct the transition kernel in the forecast step. The details are described in Algorithm 5. Note that this algorithm is the stochastic EnKF algorithm and it is recently more often to use the deterministic EnKF algorithm (see in [137]).

**Algorithm 5: Ensemble Kalman Filter (EnKF) with LLR forecasting**

- Initialization: sample the first ensemble, $\{x_0^{a,(i)}\}_{i=1:N} \sim p_0(x)$.
- For $t = 1 : T$,
  + **Forecasting**: propagate the previous ensemble by Eq. (2.6) and deduce its empirical covariance $\widehat{P}_t^f$,

$$x_t^{f,(i)} \sim \mathcal{N}\left(\widehat{m}\left(x_{t-1}^{a,(i)}\right),\ Q_t\right), \tag{2.6}$$

  where $\widehat{m}$ is the LLR estimate (see Eq. 1.26) of the dynamical function $m$ at each member value.
  + **Correcting**: adjust the forecast with the available observation $y_t$ for each member

$$\begin{aligned} \tilde{y}_t &= y_t + \epsilon_t^{(i)} - H_t x_t^{f,(i)}, \\ K_t &= \widehat{P}_t^f H_t^\top \left(H_t \widehat{P}_t^f H_t^\top + R_t\right)^{-1}, \\ x_t^{a,(i)} &= x_t^{f,(i)} + K_t \tilde{y}_t, \end{aligned} \tag{2.7}$$

  where $\epsilon_t^{(i)} \sim \mathcal{N}(0, R_t)$, $\forall i = 1 : N$.
  end.

Algorithm 5 can be found in the pioneering works [95, 155]. It was numerically demonstrated that RMSEs between the true state and ensemble mean derived from Algorithm 5 tend to the ones derived from the classical EnKF when the length of the learning sequence is large enough. Recently, the novel method has been applied in DA applications [59, 153]. Nevertheless, the EnKF, similarly as the Kalman-based recursions, provides samples whose distribution does not converge to the Bayes filtering distribution if the model $m$ is nonlinear [93].

### 2.2.3 Particle filter

Particle Filter (PF) [23, 47, 47, 48] is an alternative to compute the filtering distributions $\{p(x_t|y_{1:t})\}_t$ for nonlinear (non-Gaussian) SSMs. Generally, a PF algorithm was built based on both Monte

Carlo and sequential importance resampling techniques. $N$ particles and their respectively normalized weights $\{x_t^{(i)}, w_t^{(i)}\}_{i=1:N}$ are run on a (1.8)-like recursion and provide an empirical approximation of the filtering distribution,

$$\widehat{p}\left(x_t|y_{1:t}\right) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}\left(x_t\right) \tag{2.8}$$

The details of the classical PF algorithm are presented in Algorithm 3 (Chapter 1 [Section 1.1.2.2]).

The combination of LLR emulator and PFs is given in Algorithm 6. Note that the optimal proposal kernel in Algorithm 6 is given under an explicit form as the observation model is linear and Gaussian.

**Algorithm 6: Particle Filter (PF) with LLR forecasting**

- Initialization:
  - + Sample $\{x_0^{(i)}\}_{i=1:N_f} \sim\ p(x_0)$.
  - + Set initial weights $w_0^{(i)} = 1/N, \forall i = 1:N$.
- For $t = 1:T$,
  - + **Resampling**: draw indices $\{I_t^i\}_{i=1:N}$ with respect to weights $\{w_{t-1}^{(i)}\}_{i=1:N}$.
  - + **Forecasting**: sample new particle, for all $i = 1:N$

$$x_t^{(i)} \sim \left[\begin{array}{ll} \mathcal{N}\left(\widehat{m}\left(x_{t-1}^{(I_t^i)}\right),\ Q_t\right), & \text{[bootstrap]} \\ \mathcal{N}\left(\Sigma_t\left[Q_t^{-1}\widehat{m}\left(x_{t-1}^{(I_t^i)}\right) + H_t^\top R_t^{-1} y_t\right],\ \Sigma_t\right), & \text{[optimal]} \end{array}\right.$$

    where $\widehat{m}$ is the LLR estimate (see Eq. 1.26) of the dynamical function at each particle and $\Sigma_t = \left(Q_t^{-1} + H_t^\top R_t^{-1} H_t\right)^{-1}$.
  - + **Weighting**: compute importance weight

$$w_t^{(i)} \sim \left[\begin{array}{ll} \mathcal{N}\left(y_t; x_t^{(i)},\ R_t\right), & \text{[bootstrap]} \\ \mathcal{N}\left(y_t; \widehat{m}\left(x_{t-1}^{(i)}\right), H_t Q_t H_t^\top + R_t\right), & \text{[optimal]} \end{array}\right.$$

    then calculate the corresponding normalized weight $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{i=1}^{N} \tilde{w}_t^{(i)}}$, $\forall i = 1:N$.

end.

In the next section, numerical comparisons of the classical and the non-parametric approaches (EKF, EnKF, and PF) are illustrated on a toy model. For the non-parametric approaches, the filtering algorithms are combined with LCR and LLR forecasting emulators. Note that an

optimal number of neighbors used in regression methods is computed based on the learning data before filtering.

## 2.3 Numerical results on Lorenz 63

Experiments in this section are run on the L63 model (1.6) with error noise covariances $Q = I_3$ and $R = 2I_2$. The dynamical model $m$ defined in the ODE system (1.7) is solved by running a Runge-Kutta scheme (see in [22]) with a fixed model time increment $dt = 0.08$ (except in the experiment where the reconstruction quality of the filtering algorithms are compared with different values of $dt$). Given the model (1.6), a $T$-learning sequence $x_{0:T}$ of the state process $\{X_t\}_t$ and a sequence $y'_{1:T'}$ of the observation process $\{Y_t\}_t$ (only the first and the third components of the state are observed) are simulated. On Figure 2.2, an illustration of a first part of the simulated state and observation sequences is shown.

In section 2.3.1, we first compare LLR and LCR methods in estimating $m$ given the $T$-learning data. Then, section 2.3.2 will illustrate experiments for state reconstruction in L63 model (1.6) given the observations and the learning data. The filtering schemes (EKF, EnKF, bootstrap PF and optimal PF) are combined with the true forecast model or its estimates denoted by $\widehat{m}$(LCR) and $\widehat{m}$(LLR). As mentioned, LCR cannot estimate the model Jacobian matrix so the method is not combined with the EKF algorithm.

### 2.3.1 Comparison of LCR and LLR methods for estimation of the dynamical model

In the first experiment, a grid-search algorithm is run on the $T$-learning data for identifying an optimal number of neighbors ($n$) used in LCR and LLR methods (cross-validation is used to avoid over-fitting). This is illustrated on the left panel of Figure 2.1. RMSEs (2.9) between the true state and the forecast values derived from LLR and LCR are computed for $n \in [10, 200]$, respectively.

$$RMSE(forecast) = \sqrt{\frac{\sum_{t=1}^{T} \|x_t - \widehat{m}\,(x_{t-1})\|^2}{T}} \tag{2.9}$$

The forecast error of the true dynamical model is also displayed. LLR typically needs more neighbors than LCR because it has more parameters to be estimated. But LLR almost gives smaller RMSEs than the ones of LCR. Moreover LLR errors is closer to the true error (between

$x_t$ and $m(x_{t-1})$) when $n$ is larger than 50. Note that if the length of the data ($T$) is large enough the true forecast error is equal to square root of trace of model error covariance $\sqrt{Tr(Q)} = \sqrt{3}$.

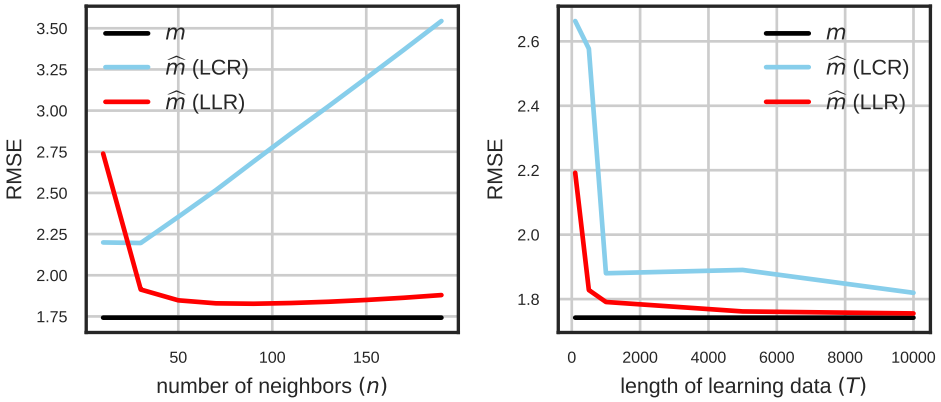


**Figure 2.1** – Comparison of RMSEs (2.9) of LCR and LLR on the L63 model (1.6) with $dt = 0.08, Q = I_3, R = 2I_2$. Left panel: RMSEs are computed on a learning sequence (length $T = 10^3$) with respect to the number of neighbors ($n$). Right panel: RMSEs are computed on a testing sequence (length $T' = 10^3$) with respect to the length of learning sequences ($T$) on which non-parametric estimates $\widehat{m}$ of the dynamical function $m$ is computed.

The advantage of LLR compared to LCR is also illustrated on the right panel of Figure 2.1. Learning sequences $\{x_{0:T}\}$ of the state process are simulated with different length $T \in [10^2, 10^5]$ and another testing sequence $x'_{0:T'}$ is generated with fixed $T' = 10^3$. Here LCR and LLR estimates are learned on each $T$-learning data and RMSEs (2.9) between $x'_t$ and $\widehat{m}(x'_{t-1})$ are computed with respect to $T$. Note that, for each learning data and each non-parametric estimation method, an $n$-grid search (as shown on the left panel of Figure 2.1) is run in order to retrieve a reasonable choice of a number of necessary neighbors before forecast. LCR errors are almost $[0.05 - 0.1]$ larger than LLR errors. As expected, when $T$ is large enough the LLR error tends to the true forecast error and converges quicker than the LCR error.

### 2.3.2 Comparison of classical and non-parametric filtering algorithms

We now compare the state reconstruction quality of different filtering algorithms in both classical and non-parametric setting. Given a learning sequence $x_{0:T}$ of the state process (for non-parametric approaches) and a testing sequence $y'_{1:T'}$ of the observation process (the length of the testing data $T'$ is fixed to $10^3$), filtering algorithms are run to approximate $p(x'_{0:T'}|y'_{1:T'})$. The main scores using to compare the efficiency of these algorithms consist of RMSEs (2.10)

between their mean estimates $\widehat{x}'_t$ of the filtering distribution and the true state $x'_t$

$$RMSE(filtering) = \sqrt{\frac{\sum_{t=1}^{T'} \|x'_t - \widehat{x}'_t\|^2}{T'}}, \tag{2.10}$$

and log-likelihood defined by

$$\begin{aligned} l(y'_{1:T'}) = \ln p\left(y'_{1:T'}\right) &= \ln p(y'_1) \prod_{t=2}^{T'} p\left(y'_t | y'_{1:t-1}\right) \\ &= \ln \int p\left(y'_1 | x'_1\right) p(x'_1) dx'_1 + \sum_{t=2}^{T'} \ln \int p\left(y'_t | x'_t\right) p(x'_t | y'_{1:t-1}) dx'_t \end{aligned} \tag{2.11}$$

where $p(x'_1)$ is the distribution propagated from the initial step and $p(x'_t|y'_{1:t-1})$ is the forecast distribution for other time steps.

By using EKF algorithms, the log-likelihood (2.11) is estimated by

$$\widehat{l}(y'_{1:T'}) = \sum_{t=1}^{T'} \ln \mathcal{N}\left(y'_t; H_t x'^f_t, H_t P'^f_t H_t^\top + R_t\right), \tag{2.12}$$

as $(x'^f_t, P'^f_t)$ are mean and covariance computed in the forecast step (see in Algorithm 4 for details). In cases of using EnKF or PF algorithms, the log-likelihood (2.11) is approximated by

$$\widehat{l}(y'_{1:T'}) = \sum_{t=1}^{T'} \ln \sum_{i=1}^{N} \frac{1}{N} \mathcal{N}\left(y'_t; H_t x'^{(i)}_t, R_t\right). \tag{2.13}$$

Here $\{x'^{(i)}_t\}_{i=1:N}$ is a sample generated after forecasting in these algorithms (see in Algorithms 5 and 6). The log-likelihood score is considered since it permits to assess the empirical distributions derived from the filtering algorithms while the RMSE score is used to compare their means only. Furthermore, an application of the non-parametric filtering algorithms in computing model evidence based on Eq. (2.13) is introduced in Chapter 5.

#### 2.3.2.1 State reconstruction performance of non-parametric filtering algorithms

In the first experiment of this section, EKF, EnKF, and PF algorithms combined with $\widehat{m}(LLR)$ estimate are run. $N = 10^3$ members/particles are used in EnKF and PF algorithms. On Figure 2.2, means (lines) and 95% confidence intervals (CIs, filled areas) of empirical filtering distributions are displayed for each component (results of the optimal PF are not shown on the

figure because those are similar to the ones of the bootstrap PF). The reconstruction quality of these algorithms are also compared in terms of RMSE and coverage probability (CP, percentage of the state belonging to 95% CI) presented in Table 2.1. Generally, all three algorithms well approximate the filtering distributions. Their mean estimates are close to the true state and 95% CIs almost cover the state sequence. Mean of CPs is in $[93\% - 94\%]$ which is close to the expected value 95%. Although the second component is unobserved, these filtering algorithms can provide its reasonable estimates (the means of empirical distributions are quite close to the true state and the CIs almost cover the state). The estimate bias [resp. CI] is larger [resp. wider] at several locations such as the bifurcations of the L63 model (around $t = 5, 50, 140$ for instance). When filtering at these locations, the transition distribution can be bi-modal and forecast values probably belong to two branches of the model. That leads to RMSEs [resp. CP] of the second component (with no observed information) larger [resp. smaller] than the ones corresponding to two other components. In a comparison among the mentioned filtering algorithms, EnKF and PF algorithms provide sample means and 95% CIs close to each other. As a result, RMSEs and CPs of these algorithms are similar. The EKF seems less effective than the others as the model with $dt = 0.08$ is nonlinear (see the second panel in the first row of Figure 2.4). The errors of EKF are approximately 0.06 larger than the errors of the other algorithms.

**Table 2.1** – Comparison of the reconstruction quality of non-parametric EKF, EnKF and PF algorithms on an observation sequence $y'_{1:T'}$ of the L63 model (1.6) with $dt = 0.08, Q = I_3, R = 2I_2$ and $T' = 10^3$ in terms of root of mean square error (RMSE) and coverage probability (CP). The non-parametric estimate $\widehat{m}$(LLR), learned on another state sequence with length $T = 10^3$, is used in these algorithms. The two scores are computed for each of the three components. EnKF and PF algorithms are run with $N = 10^3$ particles/realizations.

| Methods | | **EKF** | **EnKF** | **Bootstrap PF** | **Optimal PF** |
|---|---|---|---|---|---|
| 1st component | **RMSE** | 1.0281 | 1.0228 | 1.0261 | 1.0214 |
| | **CP** | 94.1% | 94.2% | 93.8% | 94.3% |
| 2nd component | **RMSE** | 1.7585 | 1.7509 | 1.7541 | 1.7475 |
| | **CP** | 94.7% | 94.5% | 92.7% | 93.7% |
| 3rd component | **RMSE** | 1.1147 | 1.1149 | 1.1076 | 1.1117 |
| | **CP** | 93% | 93.7% | 93.4% | 93.4% |

#### 2.3.2.2 Effect of the length of learning sequences ($T$) on state reconstruction of the non-parametric filtering algorithms

We now verify the convergence of the non-parametric filtering algorithms using LLR estimates $\widehat{m}$(LLR) of the dynamical model $m$. Several learning sequences with different length $T \in$

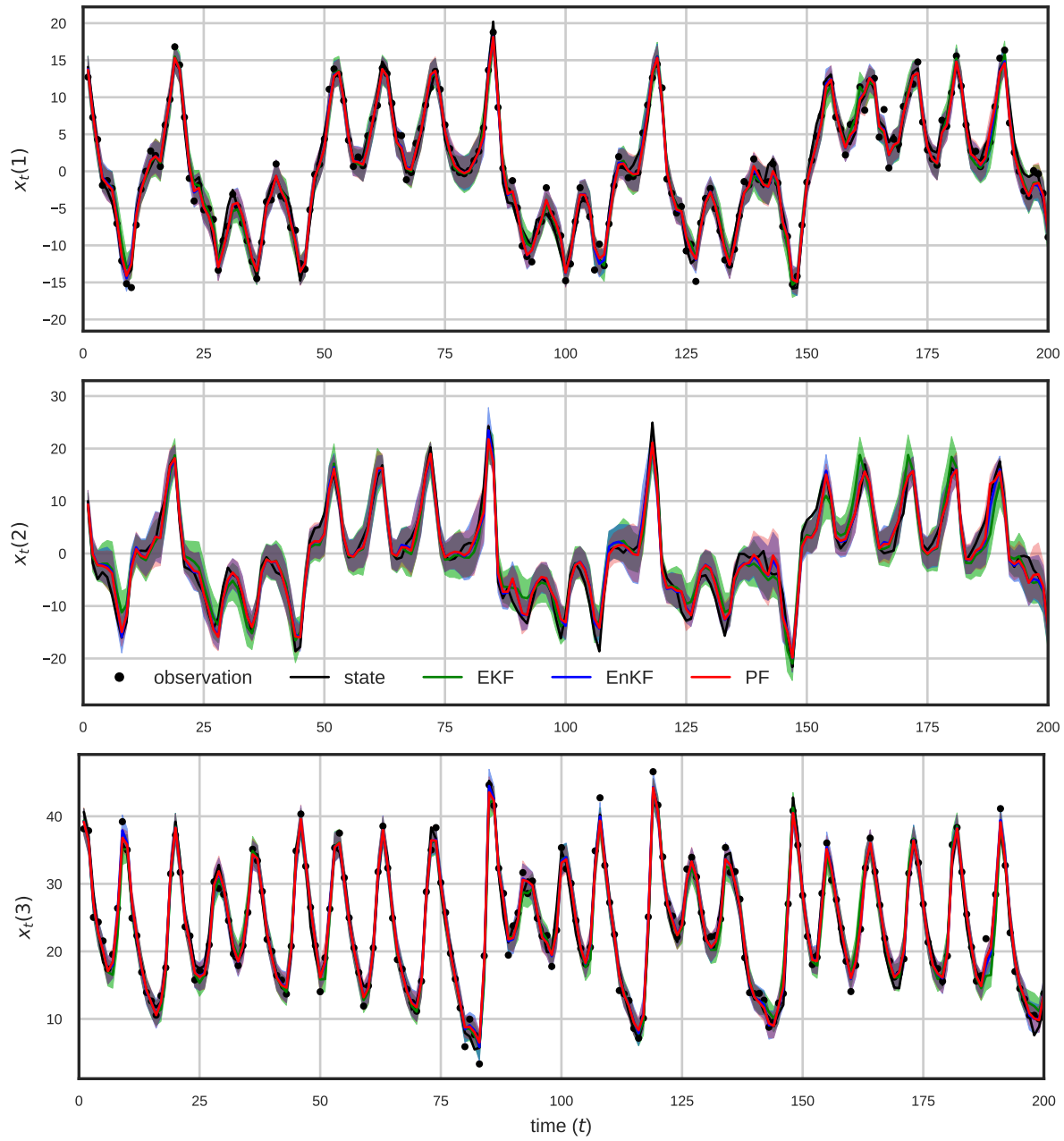


**Figure 2.2** – State reconstruction of non-parametric filtering algorithms on the L63 model (1.6) with $dt = 0.08, Q = I_3$, $R = 3I_2$. Time series of the state and observations simulated from the model are displayed by dark lines and points. Means (lines) and 95% CIs (filled areas) of filtering distributions are computed for each of three components (from top to bottom) by using non-parametric EKF, EnKF and bootstrap PF algorithms with $N = 10^3$ members/particles. These algorithms are combined with LLR forecast emulator learned on a learning sequence with length $T = 10^3$.

$[10^2, 10^5]$ are generated from the state process of the L63 model (1.6) with $dt = 0.08, Q = I_3, R = 2I_2$. The classical and non-parametric filtering algorithms are run on an observation

sequence with length $T' = 10^3$. We use $N = 10^3$ members/particles for the EnKF and PF algorithms. RMSE (2.10) and log-likelihood (2.11) are computed with respect to values of $T$. These results are illustrated on Figure 2.3.

At first glance, there is a visible improvement in both RMSE and log-likelihood estimate of the non-parametric EKF algorithm (solid line) compared to the classical EKF (dotted line). The combination of LLR and EKF provides much better reconstruction of the state due to the reason of computing Jacobian matrices $(M_t)_t$. In the classical algorithm, these matrices are computed depending on values of a particular state and time increment $dt$. When $dt$ is large, it leads to poor approximations of $(M_t)_t$. By contrast, LLR permits to estimate the local slopes based on only $n$ neighbors in the learning data. However, the EnKF and PF algorithms with a sufficiently large $N$ still give better scores than the EKF algorithms which subject to Gaussianity of all conditional distributions of the state in filtering (see Tables 2.2 and 2.3 for numerical values of RMSEs and log-likelihood estimates and Figure 2.4 for another experiment relevant to these above comments).

As displayed on the figure, the RMSEs and log-likelihood estimates derived from the non-parametric EnKF and PF algorithms using LLR estimate (solid line) tend to the scores of the classical ones when $T \geq 5 \times 10^3$. The results completely cohere with those illustrated on the right panel of Figure 2.1. Compared to the discrepancy between the scores of the non-parametric filtering algorithms at $T = 10^4$ and the ones of the classical algorithms, the discrepancy at small values of $T$ (for instance $T \in [5 \times 10^2, 2 \times 10^3]$) is not large. This may allow to run the non-parametric filtering algorithms without the need of a huge amount of the learning data. In summary, if the learning data is informative enough LLR estimate converges to the true model $m$. Consequentially, the non-parametric EnKF and PF algorithms give similar results as the classical algorithms.

#### 2.3.2.3 Effect of the sample size on state reconstruction of the non-parametric EnKF and PF algorithms

In this experiment, the reconstruction quality of the filtering algorithms using the dynamical model $m$ or its estimates ($\widehat{m}$(LCR) and $\widehat{m}$(LLR)) is compared in terms of RMSEs (2.10) and log-likelihood (2.11). They are shown in Table 2.2 and Table 2.3 respectively. Remember that the length of the learning sequence for LCR and LLR estimates is $T = 10^3$ and the number of observations is $T' = 10^3$. Here the scores corresponding to EnKF and PF algorithms are

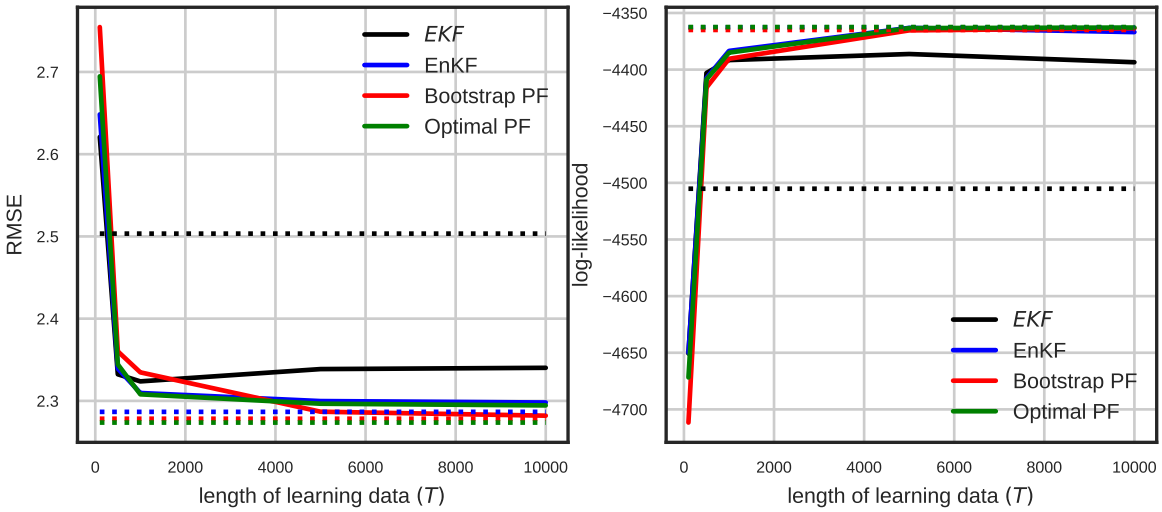


**Figure 2.3** – Comparison in state reconstruction quality (RMSE (2.10), log-likelihood (2.11)) of the classical filtering algorithms (dotted lines) using the true model ($m$) and non-parametric filtering algorithms (solid lines) using LLR estimate $\widehat{m}$(LLR) on L63 model(1.6) with $dt = 0.08, Q = I_3, R = 2I_2, T' = 10^3$ and $N = 10^3$ members/particles. In non-parametric algorithms, $\widehat{m}$(LLR) is estimated based on learning data with different length ($T$).

computed with respect to their sample size ($N$). Due to the stochastic nature of these filtering algorithms (derived from stochastic sampling for members/particles in the forecast step), each algorithm is repeated 10 times. Mean and standard deviation of the RMSEs and log-likelihood estimates of the algorithms are provided.

For all situations, the filtering algorithms using $\widehat{m}$(LCR) give approximately 25% larger RMSEs and 5% smaller log-likelihood values than the others. This relates to the bias in estimating the dynamical model (see Figure 2.1). As expected, the filtering algorithms using $\widehat{m}$(LLR) provide similar scores to the ones using the true dynamical model $m$. The EnKF algorithms quickly improve the scores from $N = 10$ to $N = 50$ and stabilize hereafter while PF algorithms seem to stabilize after $N = 500$. Especially, bootstrap PF algorithms with $N = 10$ give approximately 4 times greater [resp. smaller] than the errors [resp. log-likelihood estimates] of the EnKF. This is the practical well-known limitation of the bootstrap PF compared to the EKF and EnKF algorithms. The optimal PF algorithms work much better than the bootstrap in the cases using a low number of particles $N \in [10, 100]$. The reason is derived from taking into account the observed information in the proposal kernel, probably leading to force the forecast particles towards observations. When $N = 10$ the optimal PF is approximately 1.5 less effective than the EnKF, and as $N \geq 50$ the scores of the optimal PF algorithms are much closer to the ones of the EnKF but with slightly larger variance. In summary, point-estimation results of

the state of the EnKF algorithms seem to be not affected by the number of members ($N$). By contrast, it is extremely sensitive to the bootstrap PF results. Comparing to the sensitivity of $N$ to reconstruction results of the bootstrap PF algorithms, the one related to the optimal PF algorithms is significantly reduced.

**Table 2.2** – Comparison of RMSEs (2.10) between the estimated state and the true state on the L63 model (1.6) with $dt = 0.08, Q = I_3, R = 2I_2$ and $T' = 10^3$. Non-parametric model estimates of LCR or LLR methods are learned on a state sequence with $T = 10^3$. The estimated state is the mean of filtering distribution approximated by the filtering algorithms combined with different forecast models. For EnKF and PF algorithms, RMSEs mean and standard error of their 10 replications are shown with respect to sample size ($N$).

| Methods | Model | **number of members/ particles** ($N$) | | | | |
|---|---|---|---|---|---|---|
| | | 10 | 50 | 100 | 500 | 1000 |
| **EKF** | true | 2.5034 | | | | |
| | LCR | - | | | | |
| | LLR | 2.3202 | | | | |
| **EnKF** | true | 2.9065, 0.3234 | 2.3354, 0.0086 | 2.3122, 0.0152 | 2.2858, 0.0065 | 2.2788, 0.0045 |
| | LCR | 3.1432, 0.072 | 2.7113, 0.0175 | 2.6764, 0.0148 | 2.6438, 0.0049 | 2.6448, 0.0055 |
| | LLR | 2.7757, 0.1928 | 2.3660, 0.0202 | 2.3317, 0.0144 | 2.3138, 0.0044 | 2.3104, 0.0036 |
| **Bootstrap PF** | true | 11.4089, 1.0973 | 4.5604, 2.3551 | 2.7465, 0.6859 | 2.2862, 0.0052 | 2.2787, 0.0064 |
| | LCR | 12.9102, 0.9634 | 7.9979, 1.0873 | 6.5820, 1.8811 | 2.9478, 0.1249 | 2.7827, 0.0289 |
| | LLR | 11.2913, 2.4745 | 3.5288, 1.1366 | 2.6105, 0.2596 | 2.3417, 0.0082 | 2.3252, 0.0080 |
| **Optimal PF** | true | 4.3691, 0.8134 | 2.5543, 0.2995 | 2.3225, 0.0298 | 2.2747, 0.0054 | 2.2725, 0.0031 |
| | LCR | 6.0978, 0.6116 | 3.6095, 0.3500 | 3.0609, 0.1662 | 2.7456, 0.0188 | 2.7166, 0.0101 |
| | LLR | 4.6253, 0.7205 | 2.5376, 0.1085 | 2.4284, 0.1738 | 2.3230, 0.0050 | 2.3123, 0.0057 |

**Table 2.3** – Comparison of log-likelihood (2.11) computed by non-parametric filtering algorithms on the L63 model (1.6) with $dt = 0.08, Q = I_3, R = 2I_2$ and $T' = 10^3$. Non-parametric model estimates of LCR and LLR methods are learned a state sequence with $T = 10^3$. For EnKF and PF algorithms, log-likelihood mean and standard error of 10 replications of each algorithm are shown with respect to sample size ($N$).

| Methods | Model | **number of members/ particles** ($N$) | | | | |
|---|---|---|---|---|---|---|
| | | 10 | 50 | 100 | 500 | 1000 |
| **EKF** | true | -4505 | | | | |
| | LCR | - | | | | |
| | LLR | -4389 | | | | |
| **EnKF** | true | -4937, 243 | -4413, 9 | -4384, 10 | -4366, 3 | -4359, 2 |
| | LCR | -5206, 119 | 4664, 18 | 4620, 10 | -4585, 2 | -4580, 3 |
| | LLR | -4880, 202 | -4434, 11 | -4407, 8 | -4382.0, 2 | -4380, 2 |
| **Bootstrap PF** | true | -21494, 3016 | -6971, 3084 | -4778, 663 | -4370, 4 | -4365, 3 |
| | LCR | -26976, 3192 | -12460, 2037 | -9610, 2928 | -4808, 102 | -4673, 24 |
| | LLR | -21032, 6539 | -5459, 1216 | -4600, 253 | -4397.0, 5 | -4387, 5 |
| **Optimal PF** | true | -6905, 1155 | -4583, 224 | -4402, 24 | -4368, 3 | -4362, 2 |
| | LCR | -10127, 1227 | -5732, 481 | -4999, 184 | -4654, 16 | -4627, 11 |
| | LLR | -7117, 925 | -4561, 99 | -4480, 153 | -4390, 3 | -4384, 4 |

#### 2.3.2.4 Effect of nonlinearity of the dynamic on state reconstruction of the non-parametric algorithms

Let us now focus on the impact of model nonlinearity on reconstruction performances of the non-parametric EKF, EnKF and PF algorithms using LLR forecast emulator. This is displayed on Figure 2.4. In this experiment, nonlinearity level of the L63 model (1.6) with $Q = I_3, R = 2I_2$ is increased following up model time increment $dt \in [0.01, 0.2]$. For each $dt$, different learning and testing sequences with length $T = T' = 10^3$ are generated from the corresponding state and observation processes.

In the first row of Figure 2.4, scatter plots of the first components in two successive state variables $(X_{t-1}, X_t)$ derived from the dynamical models with different values of $dt$ are performed. Here one can see that the relation between $X_{t-1}$ and $X_t$ is almost linear for $dt = 0.01$ and it is highly nonlinear for $dt = 0.2$. In the last row, plots of RMSE (2.10) and log-likelihood (2.11), computed by the non-parametric filtering algorithms, as functions of $dt$ values are presented. For $dt \in [0.01, 0.1]$, the Kalman-based algorithms give similar scores as the PF algorithms. As $dt \geq 0.1$, the discrepancy is visible. The error [resp. log-likelihood] function of the EKF algorithm suddenly increases [decreases] at $dt = 0.1$. The score values are approximately 5.5 and $-7500$ at the final $dt$ values (not shown here). The EnKF algorithm also produces greater errors and lower likelihood values than the PF algorithms (percentages of the difference are approximately 20% and 15% at $dt = 0.2$). The difference increases with nonlinearity level of the model. As expected, the PF algorithms give the best reconstruction quality on such nonlinear cases.

## 2.4 Conclusions and Perspectives

In this chapter, we have presented non-parametric filtering algorithms for reconstruction of the hidden state in nonlinear SSMs given a historical dataset and an observational sequence derived from simulations of the state process and the observation process, respectively. The proposed algorithms consist in combining an LLR estimate of the dynamical model, learned on the historical dataset, with regular filtering schemes. Numerical experiments in this chapter allow to make comparisons in terms of reconstruction quality and computational cost of

- the classical approaches (see e.g in [11, 23, 24, 54, 56, 169]) and the proposed approaches,

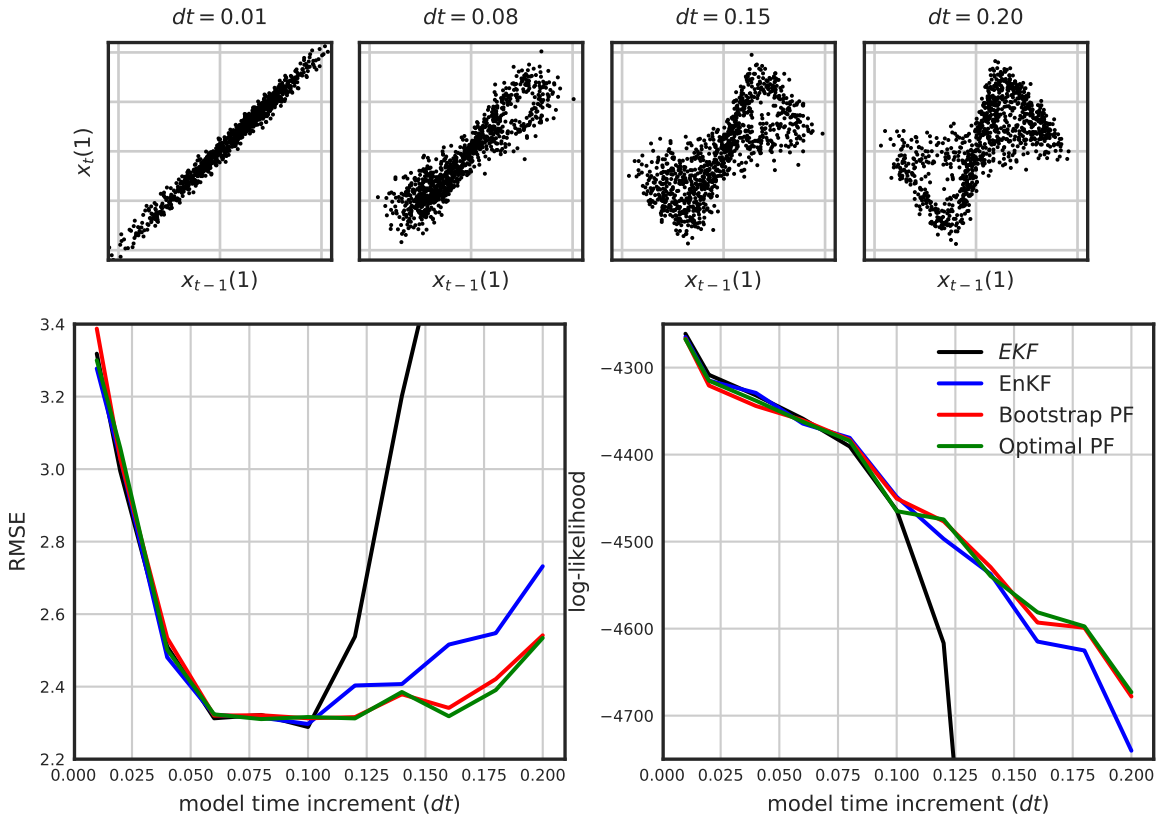


**Figure 2.4** – Comparison of the impact of model nonlinearlity in state reconstruction quality of different non-parametric filtering algorithms using LLR estimate on the L63 model (1.6) with $Q = I_3, R = 2I_2$. Learning data with length $T = 10^3$ and observation sequences with length $T' = 10^3$ are simulated from the model for every model time increment $dt \in [0.01, 0.2]$. First row: scatter plots of the first components values in two successive state variables $(X_{t-1}(1), X_t(1))$ with respect to $dt$, last row: plots of RMSE (2.10) and log-likelihood (2.11) computed by the filtering algorithms with respect to $dt$. EnKF and PF algorithms are run with $N = 10^3$ members/particles.

- the non-parametric approaches using LCR estimates for the model (see in [95, 155]) and the proposed approaches using LLR estimates for the model,
- the proposed approaches using LLR estimates for the model within different filtering scheme (EKF, EnKF, bootstrap and optimal PF).

All methods mentioned in this chapter are resumed in Diagram 2.5. There are 11 possible combinations (3 forecast models for each of the EnKF and PF algorithms, and only $m$ and $\widehat{m}$(LLR) for the EKF).

Compared to LCR, LLR generally gives better approximation of the dynamical model. Moreover, it permits to estimate the model gradient. The non-parametric filtering algorithms using LLR (**2c, 3c**) provide better estimation of the state in terms of RMSE and log-likelihood

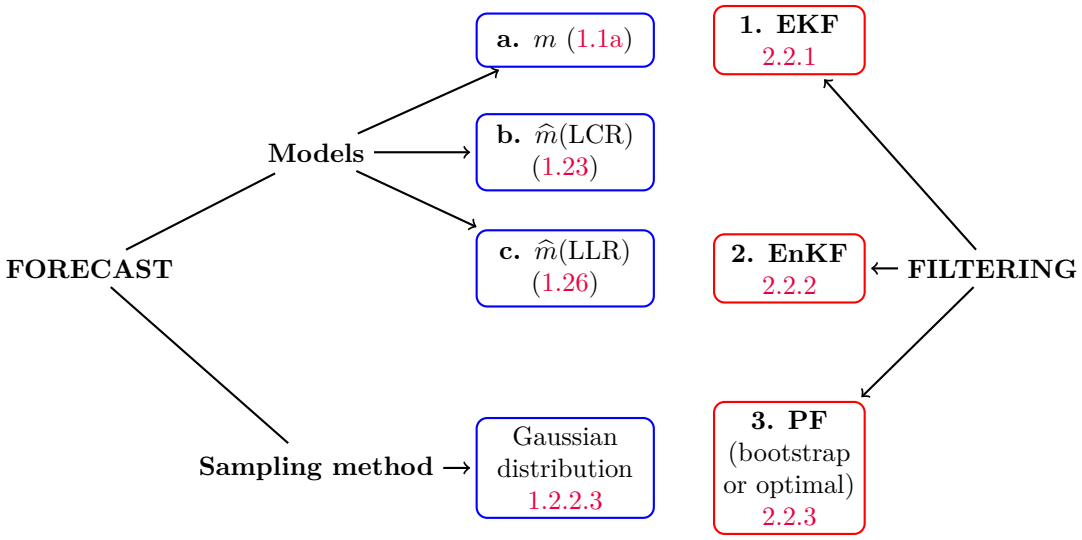


**Figure 2.5** – Diagram of forecast models and filtering methods introduced in the thesis.

scores than the ones using LCR (**2b, 3b**). If the learning data is informative enough the algorithms (**2c, 3c**) give similar results as the classical algorithms (**2a, 3a**). Especially, we have found that the non-parametric EKF algorithm (**1c**) better estimates the filtering distributions than the classical EKF (**1a**). That is due to the estimation of the first derivative of the model function in (**1c**) based on the local neighborhoods of the state only and independent from model time increment.

Among the different filtering schemes (**1c, 2c, 3c**) using LLR estimates for the model, we now propose several options which probably suit for different DA problems. First of all, the EKF algorithm (**1c**) has the lowest cost and it should be used if the SSM (1.3) is approximately a local linear model, and the conditional distributions of the state and the observations used in filtering satisfy Gaussian assumption. Otherwise, the choice between the EnKF algorithm (**2c**) and the PF algorithms (**3c**) depends on which objectives (point estimate of the state or its distributions) one wishes to obtain and how much computational resource is available. If the model is highly nonlinear and low-dimensional, the distributions of the state are required to be simulated and the computational resource is large enough, the PF algorithm (**3c**) with bootstrap proposal kernel (1.14) should be chosen. Otherwise, either (**2c**) or (**3c**) with optimal proposal kernel (1.15) using a few members/particles is appropriate to infer the state.

The future works related to this topic consist in combining such non-parametric forecast emulators with smoothing and parameter estimation algorithms, considering the cases where model covariance $Q$ is adaptive to the state values and then relaxing the Gaussian assumption of

noise distributions. Furthermore, we wish to implement the proposed methods in meteorological applications such as data assimilation, model change detection, and missing-data imputation. Last but not least, asymptotic properties of the non-parametric approaches need to be studied.

Chapter 3

# A particle-based method for maximum likelihood estimation in nonlinear state-space models

Data assimilation methods aim at estimating the state of a system by combining observations with a physical model. When sequential data assimilation is considered, the joint distribution of the latent state and the observations is described mathematically using an SSM, and filtering or smoothing algorithms are used to approximate the conditional distribution of the state given the observations. The most popular algorithms in the data assimilation community are based on the Ensemble Kalman Filter and Smoother (EnKF/EnKS) and their extensions. In this chapter, we investigate an alternative approach where a Conditional Particle Filter (CPF) is combined with Backward Simulation (BS). This allows to explore efficiently the latent space and simulate quickly relevant trajectories of the state conditionally to the observations. We also tackle the problem of parameter estimation. Indeed, the models generally involve statistical parameters in the physical models and/or in the stochastic models for the errors. These parameters impact the results of the data assimilation algorithm and there is a need for an efficient method to estimate them. Expectation-Maximization (EM) is the most classical algorithm in the statistical literature to estimate the parameters in models with latent variables. It consists in updating sequentially the parameters by maximizing a likelihood function where the state is approximated using a smoothing algorithm. In this chapter, we propose an original Stochastic Expectation-Maximization (SEM) algorithm combined with the CPF-BS smoother to estimate the statistical parameters. We show on several toy models that this algorithm provides, with reasonable

computational cost, accurate estimations of the statistical parameters and the state in highly nonlinear SSMs, where the application of EM algorithms using EnKS is limited.

## 3.1 Introduction

Data assimilation (DA) has been applied in various fields such as oceanography, meteorology or navigation [11, 24, 68, 79, 174] to reconstruct dynamical processes given observations. When sequential DA is used, an SSM is considered. It is defined sequentially for $t = 1 : T$ by Eq. (1.1) where $(X_t, Y_t)$ belong to the state and observation spaces $(\mathcal{X}, \mathcal{Y})$ and $(\eta_t, \epsilon_t)$ are independent noise sequences with zero means and covariance matrices denoted respectively $Q$ and $R$. The functions $\mathcal{M}$ and $\mathcal{H}$ describe respectively the evolution of the state $(X_t)$ and the transformation between the state and the observations $(Y_t)$. We denote $(x_t, y_t)$ instant values of the variables $(X_t, Y_t)$ and $\theta \in \Theta$ the vector of parameters. For instance, $\theta$ may contain physical parameters in the models $(\mathcal{M}_\theta, \mathcal{H}_\theta)$ and error covariances $(Q, R)$.

Given a fixed vector $\theta$ and $T$ measurements $y_{1:T} = (y_1, ..., y_T)$, DA schemes relate to compute filtering distributions $\{p_\theta(x_t|y_{1:t})\}_{t=1:T}$ or smoothing distributions $\{p_\theta(x_t|y_{1:T})\}_{t=1:T}$. However, it is often difficult to identify a reasonable value of $\theta$. This is due to the diversity of observation sources, the effect of physical terms and model complexity, or numerical failures [50, 182]. And incorrect values of $\theta$ may lead to bad reconstruction results. This is illustrated on Figure. 3.1 using the L63 model (see Eq. 1.6 for a formal definition). Smoothing with true parameter value provides a good approximation of the true trajectory (left panel) whereas the trajectory obtained with wrong parameter value is noisy and biased (right panel). This illustration emphasizes the role of parameter estimation in a DA context. A nice explanation of the problem is also given in [10].

One common approach to estimate parameters in DA community is based on empirical innovation statistics in method of moments [10, 112, 181, 182] whose formulas were first given in [43]. Although these methods permit an adaptive estimation of the error covariances $Q$ and $R$, physical parameters of nonlinear dynamical models are difficult to estimate with this approach. An alternative is to implement likelihood-based methods. A recent review, including Bayes inference and maximum likelihood estimation, can be found in [86]. The Bayesian approach aims to infer an arbitrary parameter by simulating from the joint distribution of the state and the parameter. Additionally, it is able to describe the shape of parameter distribution which might be multi-modal. But the Bayesian approaches still have some drawbacks. First, a very

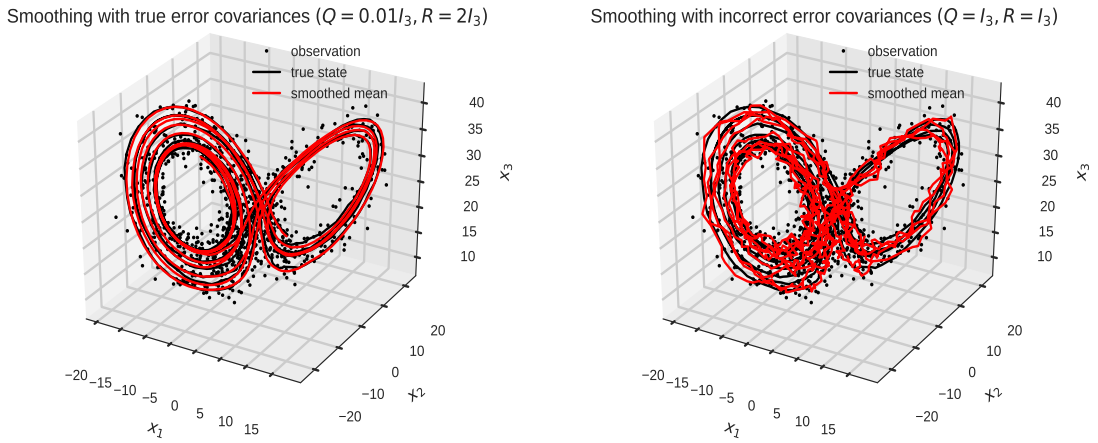


**Figure 3.1** – Impact of parameter values on smoothing distributions for the L63 model (1.6). The true state (black curve) and observations (black points) have been simulated with $\theta = (Q, R) = (0.01I_3, 2I_3)$. The mean of the smoothing distributions (read curve) are computed using a standard particle smoother [46] with 100 particles. Results are obtained with the true parameter values $\theta^* = (0.01I_3, 2I_3)$ (left panel) and wrong parameter values $\tilde{\theta} = (I_3, I_3)$ (right panel).

large number of iterations is required to get good approximations of the parameter distributions if a standard Markov Chain Monte Carlo (MCMC) method (see e.g. [4, 86, 102]) is used. In DA community, [147, 148, 165] proposed Bayesian approaches combined with EnKF algorithms and obtained approximations of the parameter distributions with a low number of members and iterations. However, simulating the distributions in high-dimensional SSMs is sometimes impractical. For example, it is difficult to simulate directly the full model covariance $Q$ which involves a lot of parameters if the latent state has values in a high dimensional space. To simplify the problem, $Q$ is typically supposed to have a predefined form, such as the multiplication of a scalar and a given matrix, and only the scale factor is estimated. In the thesis, we hence focus on maximum likelihood estimation.

There are two major approaches in the statistical literature to maximize numerically the likelihood in models with latent variables: Gradient ascent and Expectation-Maximization (EM) algorithms. As stated in [86] *gradient ascent algorithms can be numerically unstable as they require to scale carefully the components of the score vector* and thence the EM approach is generally favored when considering complicated models such as the ones used in DA. The first EM algorithm was suggested by [42]. Various variants of the EM algorithm were proposed in the statistical literature (see e.g. [28, 86, 98, 110, 141] and references therein) and in the DA

community (see [50, 106, 126, 152, 156, 164]). The common idea of these algorithms is to run an iterative procedure where an auxiliary quantity which depends on the smoothing distribution is maximized at each iteration until a convergence criterion is reached.

Within the EM machinery, the challenging issue is generally to compute the joint smoothing distribution $p_\theta(x_{0:T}|y_{1:T})$ of the latent state given an entire sequence of observations, where $x_{0:T} = (x_0, x_1, \cdots, x_T)$. For a linear Gaussian model (e.g. model 1.2), the Kalman smoother (KS, see Algorithm 1 and Algorithm 2) [143] based on Rauch-Tung-Streibel (RTS) provides an exact solution to this problem. The difficulty arises when the model is nonlinear (e.g. model 1.3) and the state does not take its values in a finite state space. In such situations the smoothing distribution is intractable. To tackle this issue, simulation-based methods were proposed. In DA, the ensemble Kalman smoother (EnKS) [24, 55, 58] and it variants [12, 13, 15] are the most favoured choices. By implementing the best linear unbiased estimate strategy, this method is able to approximate the smoothing distribution using only a few simulations of the physical model (members) at each time step. Unfortunately the approximation does not converge to the exact distribution $p_\theta(x_{0:T}|y_{1:T})$ for nonlinear SSMs [93]. Particle smoothers have been proposed as an alternative in [17,23,46,48,69]. However, they demand a huge amount of particles (and thus to run the physical models many times) to get a good approximation of the target probability distribution. Since 2010, conditional particle smoothers (CPSs) [99, 101, 102, 150], pioneered by [4], have been developed as other strategies to simulate the smoothing distribution. Contrary to the more usual smoothing samplers discussed above, CPSs simulate realizations using an iterative algorithm. At each iteration, one conditioning trajectory is plugged in a standard particle smoothing scheme. It helps the sampler to explore interesting parts of the state space with few particles. After a sufficient number of iterations, the algorithm provides samples approximately distributed according to the joint smoothing distribution.

In the DA community, EM algorithms have been generally used in conjunction with EnKS (EnKS-EM algorithm). Recent contributions [50, 126, 156] implement this approach using $20 - 100$ members and concentrate on estimating the initial state distribution and error covariances. In the statistical community, the combination of standard or approximate particle smoothers (PSs) with a large number of particles and EM algorithms (PS-EM) [86, 89, 116, 121, 141] is preferred. The number of particles is typically in the range $10^2 - 10^6$ which would lead to unrealistic computational time for usual DA problems (the number of particles corresponds to the number of time that the physical model needs to be run at each time step). In [98],

the author proposed to use a CPS algorithm, named Conditional particle filtering-Ancestor sampling (CPF-AS, [100]), within a stochastic EM algorithm (CPF-AS-SEM). The authors showed that the method can estimate $Q$ and $R$ using only 15 particles for univariate SSMs. However CPF-AS suffers from degeneracy (the particle set reduces to a very few effective particles) and consequently, the estimated parameters of CPF-AS-SEM have bias and/or large variance. In the present chapter, we propose to combine another CPS, referred to as Conditional particle filtering-Backward Simulation (CPF-BS, [102]), with the stochastic EM scheme. The novel proposed maximum likelihood estimate method, abbreviated as CPF-BS-SEM, aims at estimating the parameters with few particles and thus reasonable computational costs for DA. In this chapter we show that our approach has better performances than the EM algorithms combined with standard PS [141], CPF-AS [98] and EnKS [50]. Numerical illustrations are compared in terms of estimation quality and computational cost on highly nonlinear models.

The chapter is organized as follows. In Section 3.2, we introduce the main methods used in the chapter, including smoothing with the CPF-BS smoother and maximum likelihood estimation using CPF-BS-SEM. Section 3.3 is devoted to numerical experiments and Section 3.4 contains conclusions.

## 3.2 Methods

In this section, we first introduce the conditional particle smoother which is the key ingredient of the proposed method. This smoother is based on conditional particle filtering (CPF) which is described in Section 3.2.1.1. Standard particle filtering algorithm is also reminded and its performance is compared to the one of CPF. Section 3.2.1.2 presents iterative smoothing schemes which are the combinations of CPF and ancestor tracking algorithms. We also analyze benefits and drawbacks of these filters/smoothers. Then an iterative smoothing sampler based on CPF-BS is provided as an alternative to the CPF smoothers and their theoretical properties are quickly discussed in Section 3.2.1.3. Finally, the combination of CPF-BS with the EM machinery for maximum likelihood estimation is presented in Section 3.2.2.

### 3.2.1 Smoothing using conditional particle-based methods

#### 3.2.1.1 Particle Filtering (PF) and Conditional Particle Filtering (CPF)

In the SSM defined by (1.3), the latent state $(x_t)_{t=0:T}$ is derived from a Markov process defined by its prior distribution $p_\theta(x_0)$ and transition kernel $p_\theta(x_t|x_{t-1})$. Let us remind that the observations $(y_t)_{t=1:T}$ are conditionally independent given the state process and $p_\theta(y_t|x_t)$ denotes the conditional distribution of $y_t$ given $x_t$. The transition kernel $p_\theta(x_t|x_{t-1})$ depends on both the dynamical model $m$ and the distribution of the model error $\eta_t$ whereas the conditional observation distribution $p_\theta(y_t|x_t)$ is a function of the observation model $h$ and the distribution of the observation error $\epsilon_t$. In this section we discuss algorithms to approximate the filtering distribution $p_\theta(x_t|y_{1:t})$ which represents the conditional distribution of the state at time $t$ given the observations up to time $t$. For linear Gaussian models, the filtering distributions are Gaussian distributions whose means and covariances can be computed using the Kalman recursions. When SSMs are nonlinear, as it is the typical case for DA applications, the filtering distributions do not admit a closed form and particle filtering (PF) methods have been proposed to compute approximations of these quantities [23,47,48]. The general PF algorithm is based on the following relation between the filtering distributions at time $t-1$ and $t$

$$p_\theta(x_{0:t}|y_{1:t}) = \frac{p_\theta(y_t|x_t)\ p_\theta(x_t|x_{t-1})}{p_\theta(y_t|y_{1:t-1})}\ p_\theta(x_{0:t-1}|y_{1:t-1}) \tag{3.1}$$

where $p_\theta(y_t|y_{1:t-1})$ is the normalization term of $p_\theta(x_{0:t}|y_{1:t})$. Note that if we are able to compute the joint filtering distribution $p_\theta(x_{0:t}|y_{1:t})$ then it is possible to deduce the marginal filtering distribution $p_\theta(x_t|y_{1:t})$ by integrating over all variables $x_{0:t-1}$.

PF runs with $N_f$ particles to approximate $p_\theta(x_{0:t}|y_{1:t})$ recursively in time. Let us suppose that the filtering process has been done up to time $t-1$. Since PF is based on importance sampling, we now have a system of particles and their corresponding weights $\{x_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1:N_f}$ which approximates the joint filtering distribution $p_\theta(x_{0:t-1}|y_{1:t-1})$. The next step of the algorithm consists in deriving an approximation

$$\hat{p}_\theta\left(x_{0:t}|y_{1:t}\right) = \sum_{i=1}^{N_f} \delta_{x_{0:t}^{(i)}}\left(x_{0:t}\right) w_t^{(i)} \tag{3.2}$$

of $p_\theta(x_{0:t}|y_{1:t})$ based on Eq. (3.1). It is carried out in three main steps (see left panel of Figure 3.2 for an illustration):

- **Resampling.** Systematic resampling method (see in [45, 77] for a discussion on different resampling methods) can be used to reselect potential particles in $\{x_{0:t-1}^{(i)}\}_{i=1:N_f}$. In this step the filter duplicates particles with large weights and removes particles with small weights.
- **Forecasting.** It consists in propagating the particles from time $t-1$ to time $t$ with a proposal kernel $\pi_\theta(x_t|x_{0:t-1}, y_{1:t})$.
- **Weighting.** Importance weights $\{w_t^{(i)}\}_{i=1:N_f}$ of the particles $\{x_{0:t}^{(i)}\}_{i=1:N_f}$ are computed according to the formula

$$W\left(x_{0:t}\right) = \frac{p_\theta\left(x_{0:t}|y_{1:t}\right)}{\pi_\theta(x_t|x_{0:t-1}, y_{1:t})} \overset{(3.1)}{\propto} \frac{p_\theta\left(y_t|x_t\right) p_\theta(x_t|x_{t-1})}{\pi_\theta(x_t|x_{0:t-1}, y_{1:t})} \, p_\theta\left(x_{0:t-1}|y_{1:t-1}\right). \tag{3.3}$$

The entire algorithm of PF is presented in Algorithm 3 and reminded in Algorithm 7 for another comparing objective hereafter. $\{I_t^i\}_{t=1:T}^{i=1:N_f}$ in these algorithms are used to store the particle's indices across time steps in order to be able to reconstruct trajectories. These variables are key ingredients in some of the smoothing algorithms presented later.

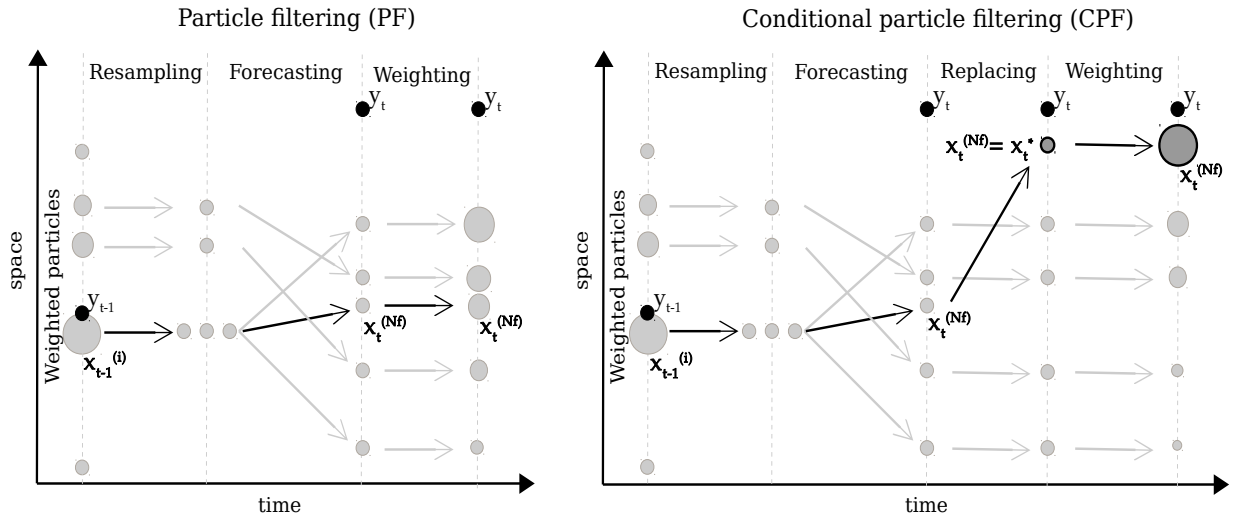


**Figure 3.2** – Comparison of PF and CPF schemes using $N_f = 5$ particles (light gray points) in time window $[t-1, t]$ on the SSM (1.3). The observation model is the identity function. The main difference is shown on black quivers as CPF replaces the particle $x_t^{(N_f)}$ with conditioning particle $x_t^*$ (dark gray point).

Note that, in a general PF algorithm, particles can be propagated according to any proposal distribution $\pi_\theta$. If we choose $\pi_\theta(x_t|x_{0:t-1}, y_{1:t}) = p_\theta(x_t|x_{t-1})\, p_\theta(x_{0:t-1}|y_{1:t-1})$ (see [23, 48, 123, 145]

or Chapter 1 [Section 1.1.2.2] for discussions on the choice of $\pi_\theta$), the importance weight function (3.3) can be simplified as $W(x_{0:t}) \propto p_\theta(y_t|x_t)$. With this choice, which is referred to as bootstrap filter in the literature, the forecast step consists in sampling according to the dynamical model $m$. It is the favorite choice for testing experiments [23, 95, 124, 169] and it is hence used in this chapter for numerical illustrations.

Conditional particle filtering (CPF) was introduced the first time by [4] and then discussed by many authors [99, 101, 102, 150]. The main difference with PF consists in plugging a conditioning trajectory $X^* = (x_0^*, \cdots, x_T^*) \in \mathcal{X}^{T+1}$ into a regular filtering scheme. In practice, CPF works in an iterative environment where the conditioning trajectory $X^*$ is updated at each iteration. This is further discussed in the next section. In this section, we assume that $X^*$ is given. Due to the conditioning, CPF algorithm differs from the PF algorithm in adding a replacing step between the forecasting and weighting steps. In this step, one of the particles is replaced by one conditioning element of the trajectory $X^*$. It is possible to set this conditioning particle as the particle number $N_f$ and this leads to updating the position of the particles at time $t$ according to

$$x_t^{(i)} = \begin{cases} x_t^{(i)} \sim \pi_\theta(x_t|x_{0:t-1}^{(I_t^i)}, y_{1:t}), & \forall i = 1 : N_f - 1 \\ x_t^*, \quad i = N_f. \end{cases} \tag{3.4}$$

Similarly to the PF, the reset sample $\{x_t^{(i)}\}_{i=1:N_f}$ is next weighted according to Eq. (3.3). In Algorithm 7 we present the differences between PF and CPF algorithms. The additional ingredients of CPF are highlighted using a gray color.

The general principle of the CPF algorithm is also presented on Figure 3.2. CPF does a selection between particles sampled from the proposal kernel $\pi_\theta$ and the conditioning particle. We can imagine two opposite situations. If the conditioning particle is "bad" (i.e. far from the true state) then the filtering procedure will not select it for the next time step by weighting and resampling. But if conditioning particle is "good" (i.e. close to the true state) then it will have a high weight and it will be duplicated and propagated at the next time step. This ensures that if a "good" sequence is used as conditioning trajectory, then the CPF algorithm will explore the state space in the neighborhood of this trajectory and thus, hopefully, an interesting part of the state space. This is also illustrated on Figure 3.3 which has been drawn using the Kitagawa SSM (given in Eq. 1.5). This univariate model was chosen because it is known that it is difficult to compute accurate approximations of the filtering distribution: the forecasting distribution

**Algorithm 7: Particle Filtering (PF)/*Conditional Particle Filtering (CPF)*.**

- Initialization:
  - + Sample $\{x_0^{(i)}\}_{i=1:N_f} \sim p_\theta(x_0)$.
  - + Set initial weights $w_0^{(i)} = 1/N_f, \forall i = 1 : N_f$.
- For $t = 1 : T$,
  - + **Resampling**: draw indices $\{I_t^i\}_{i=1:N}$ with respect to weights $\{w_{t-1}^{(i)}\}_{i=1:N}$.
  - + **Forecasting**: sample new particle

$$x_t^{(i)} \sim \pi_\theta \left( x_t | x_{0:t-1}^{(I_t^i)}, y_{1:t} \right), \forall i = 1 : N_f.$$

  - + ***Replacing*** *(only for CPF): set* $x_t^{(N_f)} = x_t^*$ *and* $I_t^{N_f} = N_f$.
  - + **Weighting**: compute $\tilde{w}_t^{(i)} = W\left(x_{0:t-1}^{(I_t^i)}, x_t^{(i)}\right)$ by using Eq. (3.3) then calculate its normalized weight $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{i=1}^{N_f} \tilde{w}_t^{(i)}}$, $\forall i = 1 : N_f$.

  end for.

$p_\theta(x_t|x_{t-1})$ can be bimodal due to the cos-term and the observation operator is quadratic. In addition, we use a large value of $R$ to get unreliable observations. On the left panel of the figure, around time $t = 17$, PF starts to simulate trajectories which are far away from the true state. All the particles are close to 0 and the dynamical model provides unstable and inaccurate forecasts. At the same time, the observation $y_t$ is unreliable and cannot help to correct the forecasts. It leads to a bad approximation of the filtering distribution since time $t = 18$: the forecast distributions remain far from the true state and the filter gives bad results. CPF gives better results thanks to a good conditioning trajectory which helps to generate relevant forecasts (see right panel of Figure 3.3).

When the number of particles $N_f$ is big, the effect of the conditioning particles becomes negligible and the PF and CPF algorithms give similar results. However, running a particle filter with a large number of particles is generally computationally impossible for DA problems. Algorithms which can provide a good approximation of the filtering distributions using only a few particles (typically in the range $10 - 100$) are needed. An alternative strategy to PF/CPF with a large number of particles, based on iterating the CPF algorithm with a low number of particles, is discussed in the next section.

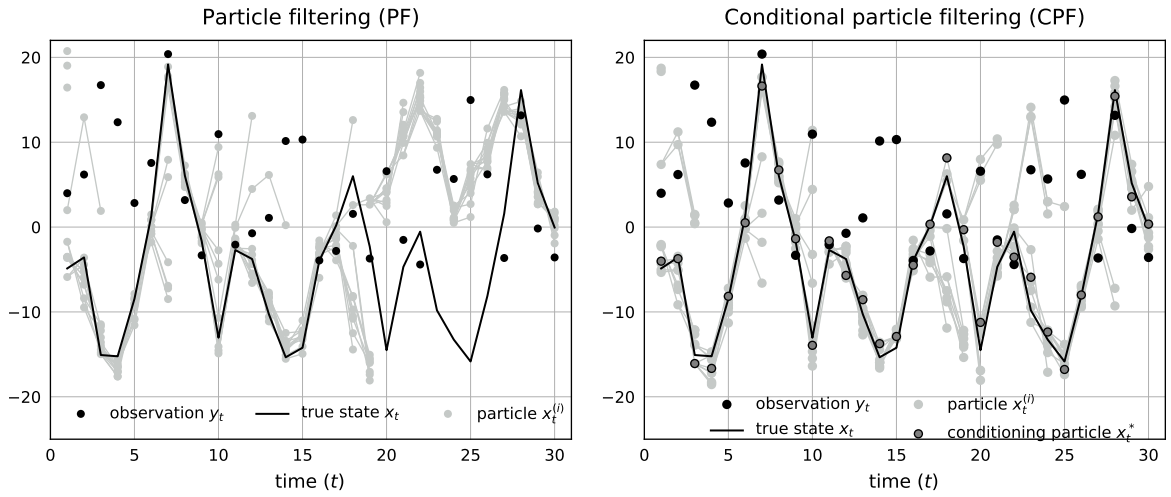


**Figure 3.3** – Comparisons of PF and CPF performances with 10 particles on the Kitagawa model (1.5), where $T = 30, (Q, R) = (1, 10)$. Conditioning particles (dark gray points) are supposed to live around to the true state trajectory (black curve). Gray lines are the links among particles which have the same ancestor.

#### 3.2.1.2 Smoothing with conditional particle filters

A key input to the CPF algorithm is the conditioning particles of the given trajectory $X^*$. As discussed in the previous Section, the "good" conditioning particles must be "close" to the true state in order to help the algorithm simulates interesting particles in the forecast step with reasonable computational costs. Remark also that the distribution of the particles simulated by running one iteration of the CPF depends on the distribution of the conditioning trajectory $X^*$. The distribution of $X^*$ must be chosen in such a way that the output of the CPF is precisely the smoothing distribution that we are targeting. One solution to this problem can be found in [4] (see a summary in Theorem 3.2.1): if $X^*$ is simulated according to the smoothing distribution then running the CPF algorithm with this conditioning trajectory will provide other sequences distributed according to the smoothing distributions. A more interesting result for the applications states that if the conditioning trajectory is "bad", then iterating the CPF algorithm after a certain number of iterations will provide "good" sequences for $X^*$ which are distributed approximately according to the smoothing distribution. At each iteration the conditioning trajectory $X^*$ is updated using one of the trajectories simulated by the CPF algorithm at the previous iteration. The corresponding procedure is described more precisely below.

Running the CPF algorithm (Algorithm 7) until the final time step $T$ gives a set of particles, weights, and indices which define an empirical distribution on $\mathcal{X}^{T+1}$,

$$\hat{p}_\theta\left(x_{0:T}|y_{1:T}\right) = \sum_{i=1}^{N_f} \delta_{x_{0:T}^{(i)}}\left(x_{0:T}\right) w_T^{(i)} \tag{3.5}$$

where $x_{0:T}^{(i)}$ is one particle path (realization) taken among particles (eg. one continuous gray link over all time steps on Figure 3.3), $w_T^{(i)}$ is its corresponding weight and $i$ is an index of its particle at the final time step. The simulation of one trajectory according to Eq. (3.5), is based on sampling its final particle with respect to the final weights $(w_T^{(i)})_{i=1:N_f}$ such that

$$p(x_{0:T}^s = x_{0:T}^{(i)}) \propto w_T^{(i)}. \tag{3.6}$$

Then, given the final particle, eg. $x_T^s = x_T^{(i)}$, the rest of the path is obtained by tracing the ancestors (parent, grandparent, etc) of the particle $x_T^{(i)}$. The information on the genealogy of the particles is stored in the indices $(I_t^i)_{t=1:T}^{i=1:N_f}$ since $I_t^i$ is the index of the parent of $x_t^{(i)}$. The technique is named ancestor tracking (also presented in statistical literature of standard PF such as [49]). It is illustrated on Figure 3.4. Given $i = 1$, the parent of particle $x_4^{(1)}$ is the particle $x_3^{(I_4^1)} = x_3^{(3)}$, its grandparent is the particle $x_2^{(I_3^3)} = x_2^{(3)}$ and its highest ancestor is $x_1^{(I_2^3)} = x_1^{(2)}$. At the end, we obtain one realization $x_{1:4}^s = x_{1:4}^{(1)} = (x_1^{(2)}, x_2^{(3)}, x_3^{(3)}, x_4^{(1)})$.

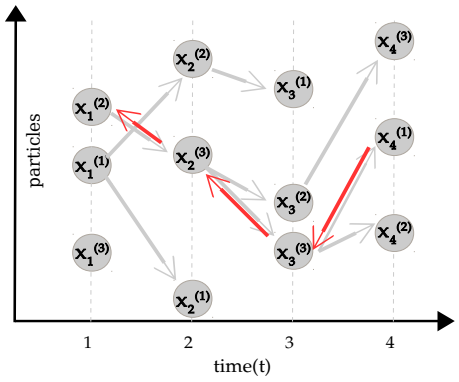


**Figure 3.4** – An example of ancestor tracking one smoothing trajectory (backward quiver) based on ancestral links of filtering particles (forward quivers). Particles (gray balls) are assumed to be obtained by a filtering algorithm with $T = 4$ and $N_f = 3$.

In practice the following procedure can be implemented to generate a path $x_{0:T}^s = x_{0:T}^{(J_T)} = (x_0^{(J_0)}, x_1^{(J_1)}, \cdots, x_T^{(J_T)})$ according to Eq. (3.5)

- For $t = T$, draw index $J_T$ with $p(J_T = i) \propto w_T^{(i)}$ and set $x_T^s = x_T^{(J_T)}$.

- For $t < T$, set index $J_t = I_{t+1}^{J_{t+1}}$ and $x_t^s = x_t^{(J_t)}$.

Finally the iterative smoothing algorithm using CPF can be described as follows,

**Algorithm 8: Smoothing with Conditional Particle Filtering (CPF).**

- Run CPF (Algorithm 7) given $X^*$ and observations $y_{1:T}$, with fixed parameter $\theta$ and $N_f$ particles.
- Run ancestor tracking procedure $N_s$ times to simulate $N_s$ trajectories according to Eq. (3.5).
- Update the conditioning particle $X^*$ with one of these trajectories.

According to Theorem 3.2.1 given in [4], this algorithm will generate trajectories which are approximatively distributed according to the smoothing distribution after a certain number of iterations, even if a low number of particles is used at each iteration. However, in practice running one iteration of the CPF algorithm leads to generating trajectories which are generally almost identical to the conditioning particle [102, 150]. The main reason for this is the so-called degeneracy issue: all the particles present at the final time step $T$ share the same ancestors after a few generations. This is illustrated on Figure 3.4: all the particles present at time $t = 4$ have the same grandparent at time $t = 2$. This is also visible on the left panel of Figure 3.5. The resampling makes disappear many particles whereas other particles have many children. As a consequence, all 10 particles at the final time step $T = 30$ have the same ancestors for $t < 20$. This degeneracy issue clearly favors the conditioning particle which is warranted to survive and reproduce at each time step. When iterating the CPF algorithm, the next conditioning sequence is thus very likely to be identical to the previous conditioning sequence, except maybe for the last time steps. This leads to an algorithm which has a poor mixing and lots of iterations are needed before converging to the smoothing distribution.

To improve the mixing, [99, 101, 102] proposed to modify the replacing step of Algorithm 7 as follows. After setting the final particle $x_t^{(N_f)} = x_t^* \in X^*$ to the conditioning particle, the index of its parent $I_t^{(N_f)}$ is drawn following Bayes' rule

$$p_\theta(I_t^{N_f} = i | x_t^*, y_{1:t}) \propto p_\theta(x_t^* | x_{t-1}^{(i)})\ w_{t-1}^{(i)}. \tag{3.7}$$

Resampling $I_t^{N_f}$ helps to break the conditioning trajectory $X^*$ into pieces so that the algorithm is less likely to simulate trajectories which are close to $X^*$. The different steps of a smoother using this algorithm referred to as Conditional Particle Filtering-Ancestor Sampling (CPF-AS) algorithm are given below.

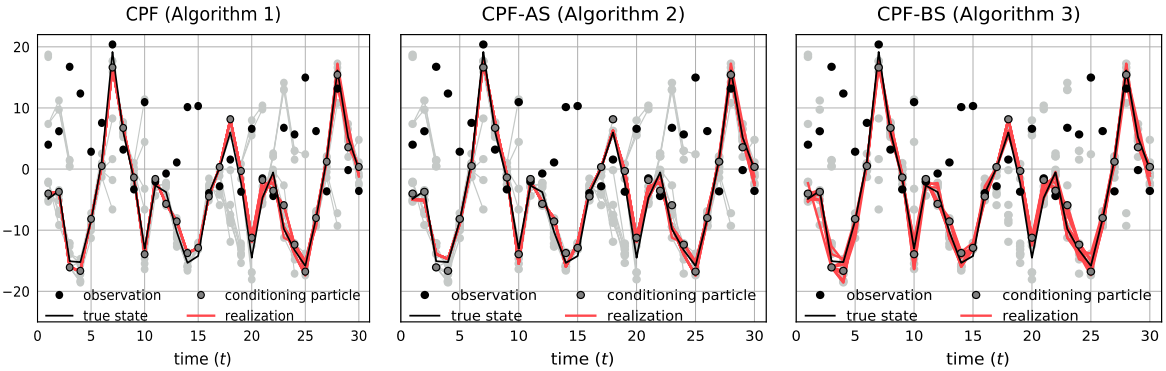


**Figure 3.5** – Comparison for simulating $N_s = 10$ realizations by using CPF smoother (Algorithm 8), CPF-AS smoother (Algorithm 9) (both based on particle genealogy- light gray links) and CPF-BS smoother (Algorithm 10) (based on backward kernel 3.10) given the same forward filtering pattern with $N_f = 10$ particles (light gray points). The experiment is run on the Kitagawa model (1.5) where $T = 30$ and $(Q, R) = (1, 10)$.

**Algorithm 9: Smoothing with Conditional Particle Filtering-Ancestor Sampling (CPF-AS).**

- Run CPF (Algorithm 7) wherein indices of conditional particles $(I_t^{N_f})_{t=1:T}$ are resampled with the rule (3.7), given $X^*$ and observations $y_{1:T}$, with fixed parameter $\theta$ and $N_f$ particles.
- Run ancestor tracking procedure $N_s$ times to get $N_s$ trajectories among particles of the CPF-AS algorithm.
- Update the conditioning particle $X^*$ with one of these trajectories.

In the above-mentioned references, it is shown empirically that this algorithm is efficient to simulate trajectories of the smoothing distribution with only $5 - 20$ particles. It is also proven that it has the same good theoretical properties (see Theorem 3.2.1) as the original CPF algorithm and that running enough iterations of the CPF-AS algorithm, starting from any conditioning particle $X^*$, permits to generate trajectories which are approximately distributed according to the smoothing distribution.

The comparison of the left and middle panels of Figure 3.5 shows that resampling the indices permits to obtain ancestor tracks which are different from the conditioning particles. However, like CPF smoother (Algorithm 8), tracking ancestral paths in the CPF-AS smoother (Algorithm 9) still suffers from the degeneracy problem mentioned above. It implies that the $N_s$ trajectories simulated at one iteration of the CPF-AS generally coincide, except for the last time steps, and thus give a poor description of the smoothing distribution. This is illustrated on Figure 3.5: all the trajectories simulated with the CPF-AS coincide for $t < 20$ and thus cannot describe the spread of the smoothing distribution. In practice, many particles which are

simulated with the physical model in the forecast step are forgotten when running the ancestor tracking and it leads to waste information and computing resources for DA applications. In the next section, we present conditional particle smoother wherein ancestor tracking is replaced by backward simulation in order to better use the information contained in the particles.

#### 3.2.1.3 Smoothing with Conditional particle filter-Backward simulation (CPF-BS)

Backward simulation (BS) was first proposed in the statistical literature in association with the regular particle filter [46, 49, 69]. Recently BS was combined with conditional smoothers [101, 102, 171]. In the framework of these smoothers, the smoothing distribution $p_\theta(x_{0:T}|y_{1:T})$ is decomposed as

$$p_\theta\left(x_{0:T}|y_{1:T}\right) = p_\theta\left(x_T|y_{1:T}\right) \prod_{t=0}^{T-1} p_\theta\left(x_t|x_{t+1}, y_{1:t}\right), \tag{3.8}$$

where

$$p_\theta(x_t|x_{t+1}, y_{1:t}) \propto p_\theta\left(x_{t+1}|x_t\right)\ p_\theta\left(x_t|y_{1:t}\right) \tag{3.9}$$

is the so-called backward kernel. Given the particles $(x_t^{(i)})_{t=0:T}^{i=1:N_f}$ and the weights $(w_t^{(i)})_{t=0:T}^{i=1:N_f}$ of the CPF algorithm (Algorithm 7) we obtain an estimate (3.2) of the filtering distribution $p_\theta\left(x_t|y_{1:t}\right)$. By plugging this estimate in (3.9), we deduce the following estimate of the backward kernel

$$\hat{p}_\theta\left(x_t|x_{t+1}, y_{1:t}\right) \propto \sum_{i=1}^{N_f} p_\theta(x_{t+1}|x_t^{(i)}) w_t^{(i)} \delta_{x_t^{(i)}}\left(x_t\right) \tag{3.10}$$

Using the relation (3.8) and the estimate (3.10), one smoothing trajectory $x_{0:T}^s = x_{0:T}^{J_{0:T}} = (x_0^{(J_0)}, x_1^{(J_1)}, \cdots, x_{T-1}^{(J_{T-1})}, x_T^{(J_T)})$ can be simulated recursively backward in time as follows.

- For $t = T$, draw $J_T$ with $p(J_T = i) \propto w_T^{(i)}$.
- For $t < T$,
  + Compute weights $w_t^{s,(i)} = p_\theta(x_{t+1}^{(J_{t+1})}|x_t^{(i)})\ w_t^{(i)}$ using (3.10), for all $i = 1 : N_f$.
  + Sample $J_t$ with $p(J_t = i) \propto w_t^{s,(i)}$.

  end for

To draw $N_s$ distinct realizations we just need to repeat $N_s$ times the procedure. The performance of BS given outputs of one run of the CPF algorithm is displayed on Figure 3.5 and the complete smoother using CPF-BS is described below (Algorithm 10).

Figure 3.6 illustrates how the iterative CPF-BS smoother works and performs on the Kitagawa model. The smoothing procedure is initialized with a "bad" conditioning trajectory ($x_t^* = 0$ for

**Algorithm 10: Smoothing with Conditional Particle Filtering-Backward Simulation (CPF-BS).**

- Run CPF (Algorithm 7) given $(X^*, Y)$ with $N_f$ particles and fixed parameter $\theta$.
- Run BS procedure $N_s$ times provided the forward filtering outputs to sample $N_s$ trajectories.
- Update the conditioning trajectory $X^*$ with one of these trajectories.

$t \in \{0, ..., T\}$). This impacts on the quality of the simulated trajectories which are far from the true state at the first iteration. Similar issues usually occur when running regular particle smoothers (such as Particle Filtering-Backward Simulation, PF-BS, see [46, 69]) with a small number of particles. The conditioning trajectory is then updated and it helps to drive the particles to interesting parts of the state space. After only 3 iterations, the simulated trajectories stay close to the true trajectory. Note that only 10 particles are used at each iteration.

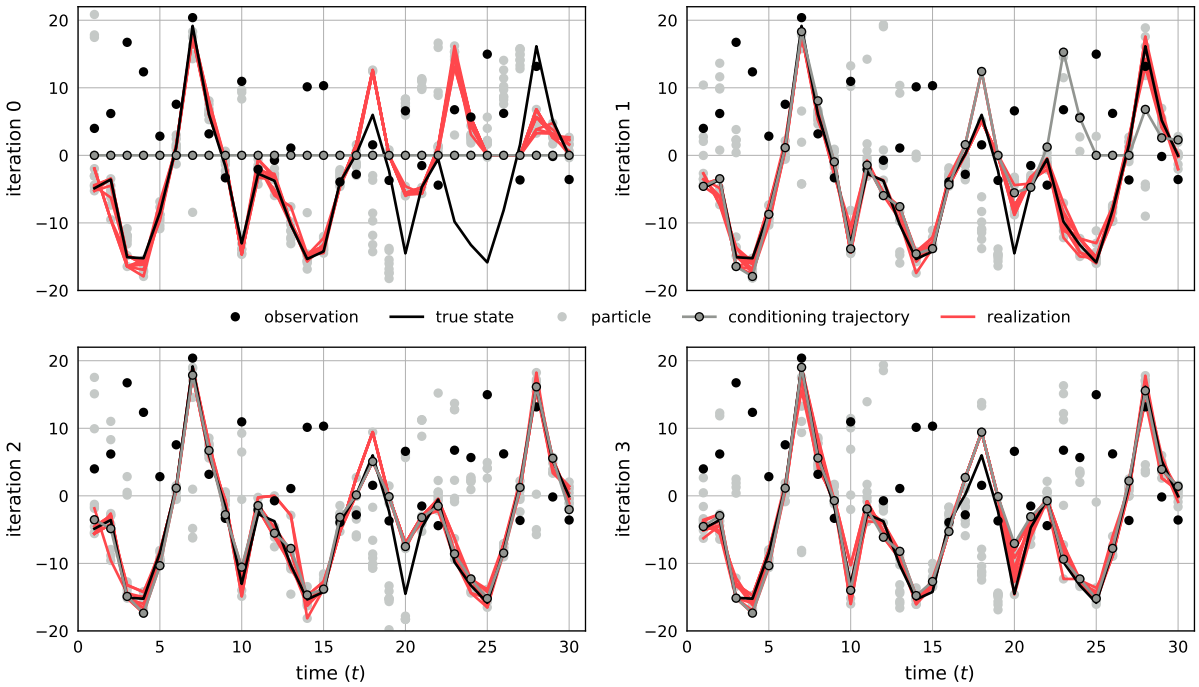


**Figure 3.6** – Performance of an iterative CPF-BS smoother (Algorithm 10) with $N_f = 10$ particles in simulating $N_s = 10$ realizations. The experiment is on the Kitagawa model (1.5) where $(Q, R) = (1, 10), T = 30$. The smoother given a zero-initial conditioning ($X^* = \mathbf{0} \in \mathbb{R}^T$) is run within 3 iterations. For each iteration the conditioning trajectory $X^*$ is one of realizations obtained from the previous.

Algorithms 8, 9 and 10 generate a new conditioning trajectory at each iteration and this defines a Markov kernel on $\mathcal{X}^{T+1}$ since the conditioning trajectory obtained at one iteration only depends on the conditioning particle at the previous iteration. Theorem 3.2.1 shows that

these Markov kernels have interesting theoretical properties (see also [33] for more results). This theorem was first proven for the CPF smoother in [4]. These results were then extended to CPF-BS in [101] and to CPF-AS in [99] with some extensions to solving inverse problems in non-Markovian models.

**Theorem 3.2.1** *For any number of particles ($N_f \geq 2$) and a fixed parameter $\theta \in \Theta$,*

*i. Markov kernel $\mathcal{K}_\theta$ defined by one of conditional smoothers (CPF: Algorithm 8, CPF-AS: Algorithm 9 and CPF-BS: Algorithm 10) leaves the invariant smoothing distribution $p_\theta(x_{0:T}|y_{1:T})$. That is, for all $X^* \in \mathcal{X}^{T+1}$ and $A \subset \mathcal{X}^{T+1}$,*

$$p_\theta(A|y_{1:T}) = \int \mathcal{K}_\theta(X^*, A)\ p_\theta(X^*|y_{1:T})\ dX^* \tag{3.11}$$

*where $\mathcal{K}_\theta(X^*, A) = \mathbb{E}_{\theta,X^*}\left[\mathbb{1}_A(x_{0:T}^{J_{0:T}})\right]$, and $x_{0:T}^{J_{0:T}} = \{x_0^{(J_0)}, \cdots, x_T^{(J_T)}\}$.*

*ii. The kernel $\mathcal{K}_\theta$ has $p_\theta$- irreducible and aperiodic. It hence converges to $p_\theta(x_{0:T}|y_{1:T})$ for any starting point $X^*$. Consequently,*

$$\|\mathcal{K}_\theta^r(X^*, \cdot) - p_\theta(\cdot|y_{1:T})\|_{TV} \overset{r\to\infty}{\longrightarrow}^{\text{as}} 0. \tag{3.12}$$

*where $\|\cdot\|_{TV}$ is the total variation norm.*

**Proof 3.2.1** *Theorem 3.2.1 in this chapter was proved corresponding to Theorem 5 in [4] for CPF (Algorithm 8), Theorem 1 and Theorem 2 in [99] for CPF-AS (Algorithm 9), and Theorem 1 in [101] for CPF-BS (Algorithm 10).*

The second property of this theorem implies that running the algorithm with any initial conditioning trajectory will permit to simulate samples distributed approximately according to the smoothing distribution after a sufficient number of iterations. However, in practice, the choice of a good initial trajectory is very important, in particular when the considered state space is complex (high nonlinearity, partly observed components,...). If we set an initial conditioning trajectory far from the truth, then lots of iterations are needed before exploring a space relevant to the true state. In such situations, it may be useful to provide an estimate of the true state using an alternative method (e.g. running another smoothing algorithm such as EnKS).

Despite sharing the same theoretical properties as the CPF and CPF-AS smoothers, we will show in Section 3.3 that CPF-BS algorithm gives better results in practice. This is due to its

ability to avoid the degeneracy problem and hence provide better descriptions of the smoothing distribution. At first glance, the computational cost of the backward technique seems to be higher than the one of ancestor tracking. Nevertheless, for DA applications, the computational complexity mainly comes from the numerical model which is used to propagate the $N_f$ particles in the forecast step. In addition, the transition probability in the backward kernel (3.10) can be computed by reusing the forecast information and does not require extra runs of the physical model. The computational cost of the CPF-BS algorithm is thus similar to the ones of CPF or CPF-AS algorithms and grows linearly with $N_f$.

Recently the CPF-BS with few particles $(5-20)$ has been used to sample $\theta$ and simulate the latent state in a Bayesian framework [99,101,102,150]. In the next section, we propose to use the CPF-BS smoother to perform maximum likelihood estimation which is the main contribution of this chapter.

### 3.2.2 Maximum likelihood estimate using CPF-BS

In this section, we discuss the estimation of the unknown parameter $\theta$ given a sequence of measurements $y_{1:T}$ of the SSM (1.1). The inference will be based on maximizing the incomplete likelihood of the observations,

$$L(\theta) = p_\theta(y_{1:T}) = \int p_\theta\left(x_{0:T}, y_{1:T}\right) dx_{0:T}. \tag{3.13}$$

The EM algorithm is the most classical numerical method to maximize the likelihood function in models with latent variables [28, 42]. It works following the auxiliary function

$$G(\theta, \theta') = \mathbb{E}_{\theta'}\left[\ln\ p_\theta\left(x_{0:T}, y_{1:T}\right)\right] \tag{3.14}$$

$$= \int \ln p_\theta\left(x_{0:T}, y_{1:T}\right)\ p_{\theta'}\left(x_{0:T}|y_{1:T}\right) dx_{0:T} \tag{3.15}$$

Due to Markovian assumption of the SSM (1.1) and independence properties of noises $(\epsilon_t, \eta_t)$ and the initial state $x_0$, the complete likelihood $p_\theta(x_{0:T}, y_{1:T})$ which appears in (3.14) can be decomposed as

$$p_\theta(x_{0:T}, y_{1:T}) = p_\theta\left(x_0\right) \prod_{t=1}^{T} p_\theta\left(x_t|x_{t-1}\right) \prod_{t=1}^{T} p_\theta\left(y_t|x_t\right). \tag{3.16}$$

The auxiliary function $G(.|\theta')$ is typically much simpler to optimize than the incomplete likelihood function and the EM algorithm consists in maximizing iteratively this function. Starting from an initial parameter $\theta_0$ an iteration $r$ of the EM algorithm has two main steps:

- **E-step**: compute the auxiliary quantity $G(\theta, \theta_{r-1})$,
- **M-step**: compute $\theta_r = \arg\max_\theta G(\theta, \theta_{r-1})$.

It can be shown that it leads to increasing the likelihood function at each iteration and gives a sequence which converges to a local maximum of $L$.

The EM algorithm combined with Kalman smoothing (KS-EM, [143]) has been the dominant approach to estimate parameters in linear Gaussian models. In nonlinear and/or non-Gaussian models, the expectation (3.14) under the distribution $p_{\theta'}(x_{0:T}|y_{1:T})$ is generally intractable and the EM algorithm cannot work in such situation. An alternative, originally proposed in [28, 29, 170], is to use a Monte Carlo approximation of (3.14)

$$\hat{G}(\theta, \theta') = \frac{1}{N_s} \sum_{j=1}^{N_s} \ln p_\theta \left( x_{0:T}^j, y_{1:T} \right), \tag{3.17}$$

where $(x_{0:T}^j)_{j=1,\dots,N_s}$ are $N_s$ trajectories simulated according to the smoothing distribution $p_{\theta'}(x_{0:T}|y_{1:T})$. This algorithm is generally named Stochastic EM (SEM) algorithm in the literature.

To implement such a procedure it is necessary to generate samples of the smoothing distribution. In the literature [86, 89, 116, 121, 141], standard or approximate particle smoothing methods are generally used. As discussed, it is generally computationally intractable for DA applications. A classical alternative in DA consists in using the EnKS algorithm [58] leading to the EnKS-EM algorithm [50, 156]. Note that this procedure does not necessarily lead to increasing the likelihood function at each iteration and may not converge. Here we explore alternative procedures based on the smoothers introduced in the previous section.

[98] proposed to use the CPF-AS smoother in an SEM-like algorithm. Here we present its original SEM version, leading to the CPF-AS-SEM algorithm. Given an initial parameter $\hat{\theta}_0$ and the first conditioning $X_0^*$, the algorithm is summed up as follows

- **E-step**:

i. Draw $N_s$ realizations by using the CPF-AS smoother (Algorithm 9) once with fixed parameter $\hat{\theta}_{r-1}$, the conditioning $X^*_{r-1}$ and the given observations $y_{1:T}$, wherein $X^*_r$ is new conditioning trajectory obtained after updating.

ii. Compute the quantity $\hat{G}(\theta, \hat{\theta}_{r-1})$ via Eq. (3.16) and Eq. (3.17).

- **M-step**: Compute $\hat{\theta}_r = \arg\max_{\theta} \hat{G}(\theta, \hat{\theta}_{r-1})$,

For each iteration $r$, $N_s$ smoothing trajectories are sampled given the previous conditioning trajectory $X^*_{r-1}$. It creates some (stochastic) dependence between the successive steps of the algorithms. This leads to such algorithm slightly different from regular EM algorithms. In [98] the author applied a similar algorithm to univariate models. Numerical results showed that this approach can give reasonable estimates with only few particles. Unfortunately, the degeneracy issue in the CPF-AS sampler may lead to estimates with some bias and large variance.

As discussed in the previous section, the CPF-BS smoother (Algorithm 10) outperforms the CPF-AS in producing better descriptions of the smoothing distribution. We hence propose a new method, CPF-BS-SEM, as an alternative to the CPF-AS-SEM for parameter estimation. The complete algorithm of the CPF-BS-SEM is presented as

**Algorithm 11: Stochastic EM algorithm using Conditional Particle Filtering-Backward Simulation (CPF-BS-SEM).**

- Initial setting: $\hat{\theta}_0$, $X^*_0$.
- For iteration $r \geq 1$,
  + **E-step**:
    i. Simulate $N_s$ samples by running CPF-BS smoother (Algorithm 10) once with fixed parameter $\hat{\theta}_{r-1}$, the conditioning $X^*_{r-1}$ and the given observations $y_{1:T}$, wherein $X^*_r$ is new conditioning trajectory obtained after updating.
    ii. Compute the quantity $\hat{G}(\theta, \hat{\theta}_{r-1})$ via Eq. (3.16) and Eq. (3.17).
  + **M-step**: compute $\hat{\theta}_r = \arg\max_{\theta} \hat{G}(\theta, \hat{\theta}_{r-1})$.
  end for.

The **E-step** of this algorithm permits to get several samples at the same computational cost that the one of CPF-AS-SEM which suffers from degeneracy. That is expected to give better estimates of the quantity $G$ in Eq. (3.17). Depending on the complexity of the SSM, the analytical or numerical procedure may be applied in the **M-step** to maximize $\hat{G}$. For Gaussian SSMs, the explicit expressions of estimators can be obtained directly as in the following example.

Such models have popularly been considered in DA context and are thus used to validate the algorithms in this chapter.

**Example**: Estimate parameter $\theta = \{Q, R\}$ in a Gaussian model

$$\begin{cases} x_t = m(x_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, Q) \\ y_t = h(x_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, R)\,. \end{cases} \tag{3.18}$$

where $m$ and $h$ can be linear or nonlinear functions.

Through Eq. (3.16) and (3.17), an estimate of the function $G$ of this Gaussian model is expressed by

$$\begin{aligned} \hat{G}(\theta, \hat{\theta}_{r-1}) = &-\frac{T}{2}\ln|Q| - \frac{T}{2}\ln|R| + C \\ &- \frac{1}{2N_s}\sum_{t=1}^{T}\sum_{j=1}^{N_s}\left[x_t^j - m\left(x_{t-1}^j\right)\right]^\top Q^{-1}\left[x_t^j - m\left(x_{t-1}^j\right)\right] \\ &- \frac{1}{2N_s}\sum_{t=1}^{T}\sum_{j=1}^{N_s}\left[y_t - h\left(x_t^j\right)\right]^\top R^{-1}\left[y_t - h\left(x_t^j\right)\right] \end{aligned} \tag{3.19}$$

where $C$ is independent to $\theta$ and $(x_t^j)_{t=0:T}^{j=1:N_s}$ are sampled from the CPF-BS smoother with respect to $\hat{\theta}_{r-1}$. Hence, an analytical expression of the estimator $\hat{\theta}_r = \{\hat{Q}_r, \hat{R}_r\}$ of $\theta$ which maximizes (3.19) is

$$\begin{aligned} \hat{Q}_r &= \frac{1}{TN_s}\sum_{t=1}^{T}\sum_{j=1}^{N_s}\left[x_t^j - m\left(x_{t-1}^j\right)\right]\left[x_t^j - m\left(x_{t-1}^j\right)\right]^\top, \\ \hat{R}_r &= \frac{1}{TN_s}\sum_{t=1}^{T}\sum_{j=1}^{N_s}\left[y_t - h\left(x_t^j\right)\right]\left[y_t - h\left(x_t^j\right)\right]^\top. \end{aligned} \tag{3.20}$$

Different strategies have been proposed in the literature for choosing the number $N_s$ of simulated trajectories in the **E-step**. If $N_s$ is large, then the law of large numbers implies that $\hat{G}$ is a good approximation of $G$ and the SEM algorithm is close to the EM algorithm. It is generally not possible to run the SEM algorithm with a large value of $N_s$. In such situation, it has been proposed to increase the value of $N_s$ at each iteration of the EM (Monte Carlo EM algorithm, MCEM, see [28, 170]) or to reuse the smoothing trajectories simulated in the previous iterations (stochastic approximation EM algorithm, SAEM, see [41, 90, 149]). It permits to decrease the variance of the estimates obtained with the SEM algorithms. For DA applications, it is generally computationally infeasible to increase significantly the value of $N_s$ but the SAEM strategy could

be explored. In the thesis, we only consider the combination of SEM and CPF-BS to facilitate the reading.

## 3.3 Numerical illustrations

Now we aim at validating the CPF-BS-SEM algorithm and comparing it with other EM algorithms including CPF-AS-SEM, PF-BS-SEM and EnKS-EM (such algorithms are presented in the mentioned references: [98,141] and [50] respectively). This is done through numerical experiments on three SSMs. A univariate linear Gaussian model (1.2) is first considered. For this model, the KS-EM algorithm can be run to provide an exact numerical approximation to the MLE and check the accuracy of the estimates derived from the SEM algorithms. Next more complicated nonlinear models (Kitagawa (1.5) and L63 (1.6)) are considered. We focus on showing the comparisons in terms of parameter and state estimation of the CPF-BS-SEM and CPF-AS-SEM algorithms with few particles on these highly nonlinear models, where we also point out the inefficiency of the EnKS-EM algorithm.

### 3.3.1 Linear model

A linear Gaussian SSM is defined as in Eq. (1.2) where $(x_t, y_t)_{t=1:T} \in \mathbb{R} \times \mathbb{R}$, $(M_t = A, H_t = 1)$ and noise variances $(Q, R)$ are constant. Let us denote $\theta = (A, Q, R)$ the vector of unknown parameters. Implementations of stochastic version of the EM algorithms for this model are discussed in [86, 98, 116]. A sequence of measurements $y_{1:T}$ is obtained by running (1.2) with true parameter value $\theta^* = (0.9, 1, 1)$ and $T = 100$ (shown on Figure 3.9). We set up the initial conditioning trajectory $X_0^*$ (only for the CPF-BS-SEM and CPF-AS-SEM algorithms) as the constant sequence equal to 0 (the same choice is done for the models considered in Sections 3.3.2 and 3.3.3) and the initial parameter $\hat{\theta}_0$ is sampled from a uniform distribution $\mathcal{U}([0.5,\ 1.5]^3)$.

For the first experiment, the CPF-BS-SEM and CPF-AS-SEM algorithms are run with $N_f = N_s = 10$ particles/realizations. Since the considered algorithms are stochastic, each of them is run 100 times to show the estimators distributions. Note that in the **M-step**, the coefficient $A$ can be easily computed using Eq. (3.19) before computing estimates of $(Q, R)$ with Eq. (3.20). Figure 3.7 shows the distribution of the corresponding estimator of $\theta$ every 10 iterations. Because the model is linear and Gaussian, we can also run the KS-EM [143] algorithm to get an accurate approximation of the true MLE of $\theta$. The estimate given by the KS-EM algorithm is shown on Figure 3.7. The differences with the true values of parameters are mainly due to the sampling

error of the MLE which is relatively important here because of the small sample size (only 100 observations to estimate 3 parameters). In the experiment, the CPF-BS-SEM and CPF-AS-SEM algorithms start to stabilize after only 10 iterations. Even with few particles, both algorithms provide estimates which have mean values close to the true MLE. As expected, CPF-BS-SEM is clearly better than CPF-AS-SEM in terms of variance.

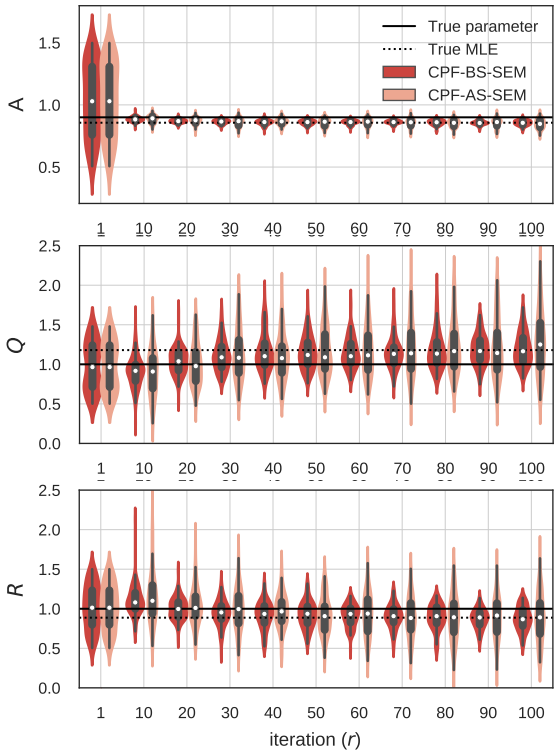


**Figure 3.7** – Comparison between CPF-BS-SEM and CPF-AS-SEM in estimating $\theta = (A, Q, R)$ for the linear Gaussian SSM model (1.2) with true parameter $\theta^* = (0.9, 1, 1)$ and $T = 100$. The results are obtained by running 100 repetitions of the two methods with 10 particles/realizations and 100 iterations. The empirical distribution of parameter estimates is represented every 10 iterations using one violin object with (black) quantile box and (white) median point inside. The true MLE (dotted line) is computed using KS-EM with $10^4$ iterations.

Then we compare the CPF-BS-SEM, CPF-AS-SEM and PF-BS-SEM algorithms varying the number of particles/realizations, $N_f = N_s \in \{10, 100, 1000\}$. The empirical distributions of the final estimators $\hat{\theta}_{100}$ obtained by the different algorithms are shown on Figure 3.8. The PF-BS-SEM algorithm with $N_f = N_s = 10$ or even $N_f = N_s = 100$ particles/realizations leads to estimates with a significant bias which is much bigger than the ones of other algorithms. It illustrates that the PF-BS-SEM algorithm based on the usual PF needs much more particles than the two other algorithms which use the idea of CPF. With $N_f = 1000$ particles, the PF-BS-SEM and CPF-BS-SEM give similar results since the effect of the conditioning trajectory

becomes negligible. Then comparing the performances of the CPF-BS-SEM and CPF-AS-SEM algorithms shows again that CPF-BS-SEM is better in terms of variance. The experiment was done on different $T$-sequences of measurements and similar results were obtained.

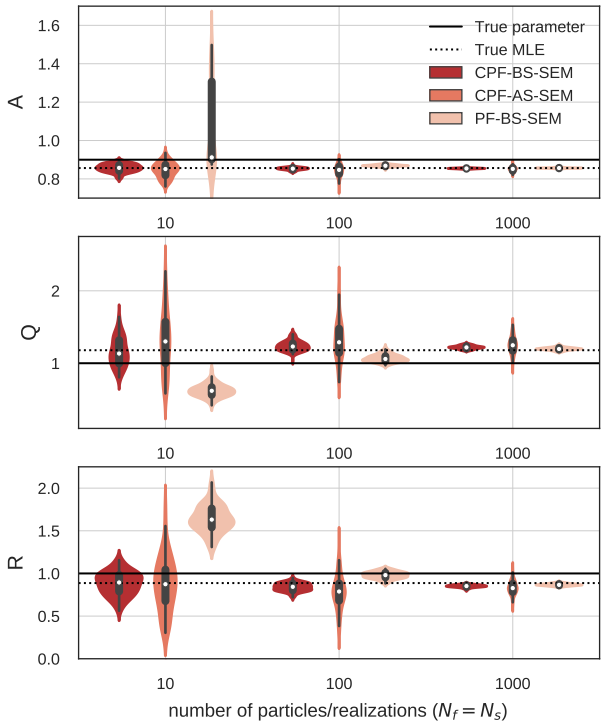


**Figure 3.8** – Comparison of the estimates of $\theta = (A, Q, R)$ at iteration 100 of CPF-BS-SEM, CPF-AS-SEM, and PF-BS-EM for the linear Gaussian SSM model (1.2) with true parameter $\theta^* = (0.9, 1, 1)$ and $T = 100$. These algorithms are run with different number of particles/trajectories ($N_f = N_s \in \{10, 100, 1000\}$). The true MLE (dotted line) is computed using KS-EM with $10^4$ iterations.

The reconstruction ability of the CPF-BS-SEM algorithm is displayed on Figure 3.9. 100 iterations of the algorithm is run once and the $N_s = 10$ trajectories simulated in each **E-step** of the last 10 iterations are stored. This produces 100 trajectories. Then empirical mean and 95% confidence interval (CI) of these 100-samples are computed and plotted on Figure 3.9. The root of mean square error (RMSE) between the smoothed mean and the true state is 0.6996 and the empirical coverage probability (percentage of the true states falling in the 95% CIs denoted CP hereafter) is 86%. In theory, the value should be close to 95%, here, the CPF-BS-SEM algorithm with non-large samples run on the short-fixed sequence of observations may give a smaller estimate of the score. An experiment to get the expected percentage is presented later (Table 3.1).

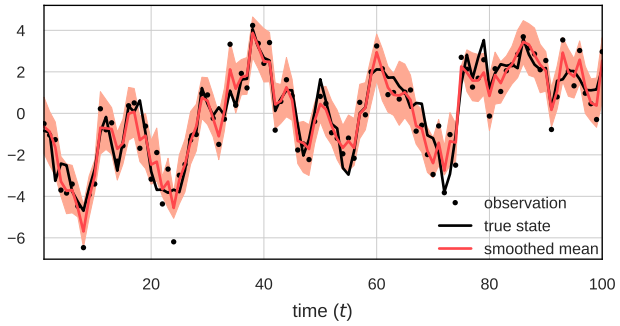


**Figure 3.9** – Reconstruction of the true state for the linear Gaussian SSM model (1.2) given $T = 100$ observations using the CPF-BS-SEM algorithm with 10 particles/realizations. Smoothed mean and 95% confidence interval are computed from realizations, which are simulated from last 10 iterations of the algorithm.

### 3.3.2 Kitagawa model

The algorithms are now applied on a highly nonlinear system widely considered in the literature to perform numerical illustrations on SSM [48, 69, 88, 89, 141]. Both $m$ and $h$ of the model are nonlinear and defined as in Eq. (1.5) where $(x_t, y_t)_{t=1:T} \in \mathbb{R} \times \mathbb{R}$. We denote $\theta = (Q, R)$ the unknown parameter. One sequence of $T = 100$ observations generated with true parameter value $\theta^* = (1, 10)$ is shown on Figure 3.12. Similar values are used in [69]. The large value of the observation variance $R$ leads to generate low quality observations and thus complicate the inference. Using only these 100 observations $y_{1:100}$, the target is to estimate $\theta$ and the true state $x_{1:100}$. The initial parameter value is simulated according to the uniform distribution $\hat{\theta}_0 \sim \mathcal{U}([1, 10]^2)$.

In this section, we only compare the CPF-BS-SEM and CPF-AS-SEM algorithms since PF-BS-SEM cannot work with a small number of particles (as shown in the linear case) and [98] also illustrated that CPF-AS-SEM using $N_f = 15$ particles outperforms PF-BS-SEM using $N_f = 1500$ particles and $N_s = 300$ realizations on the Kitagawa model. In the first experiment CPF-BS-SEM and CPF-AS-SEM are run with $N_f = N_s = 10$ particles/realizations. A comparison of the two methods in terms of estimates of log likelihood and parameter $\theta = (Q, R)$ is shown in Figure 3.10. Even with few particles the estimates obtained with the two methods seem to stabilize after 50 iterations and again the CPF-BS-SEM algorithm permits to reduce the variance of the estimates compared to the CPF-AS-SEM algorithm.

In the second experiment we run the two algorithms with fixed number of particles ($N_f = 10$) but different numbers of realizations ($N_s \in \{1, 5, 10\}$). Figure 3.11 displays the corresponding

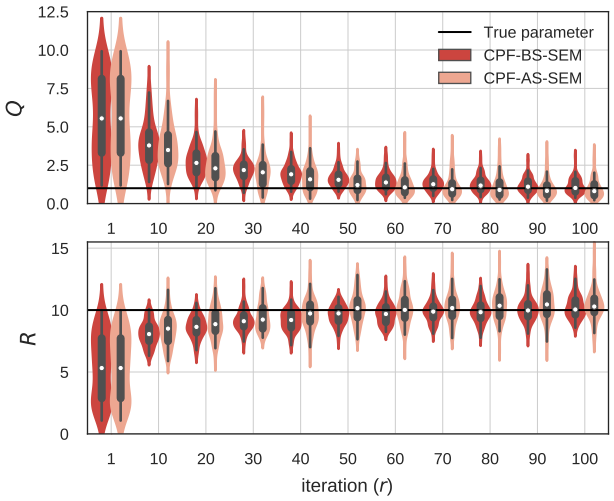


**Figure 3.10** – Comparison of the CPF-BS-SEM and CPF-AS-SEM algorithms on the Kitagawa model (1.5), where true parameter is $\theta^* = (1, 10)$ and number of observations is $T = 100$. The results are obtained by running 100 times of these methods with 10 particles/realizations and 100 iterations. The empirical distribution of parameter estimates is represented every 10 iterations using one violin object with (black) quantile box and (white) median point inside.

empirical distributions of $\hat{\theta}_{100}$. It shows that CPF-AS-SEM gives almost the same distributions of estimates as CPF-BS-SEM with $N_s = 1$. Moreover CPF-AS-SEM could not improve the estimate when we increase $N_s$ because of the degeneracy issue. CPF-BS-SEM with $N_s = 5$ and $N_s = 10$ gives better estimates in terms of bias and variance. In practice it seems useless to use a large value of $N_s$ when using BS given forward filtering information. Here CPF-BS-SEM with $N_s = 5$ has similar performance as CPF-BS-SEM with $N_s = 10$ (see also [69, 102, 150]).

Figure 3.12 shows the results obtained when reconstructing the latent space using the CPF-BS-SEM algorithm (using the same approach than for the linear model, based on storing the sequences simulated in the last 10 iterations of the algorithm). The mean of the empirical smoothing distribution seems to be close to the true state. The width of the confidence intervals varies in time and is larger (eg. at $t \in [85, 90]$) when the true state is more difficult to retrieve from the observations. The RMSE and the empirical CP with respect to the empirical smoothing distribution are 2.2478 and 84%.

### 3.3.3 Lorenz 63 model

In this section we consider the L63 model Eq. (1.6) where only the first and last components are observed. The dynamical model $m$ is related to the [107] model defined through the ODE system (1.7),

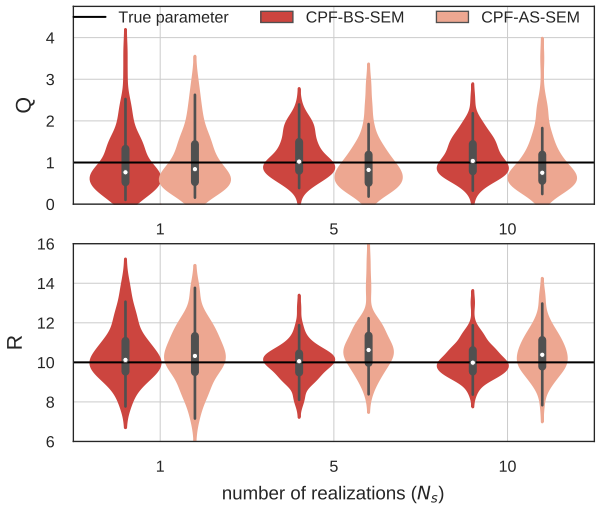


**Figure 3.11** – Comparison of the estimates of $\theta = (Q, R)$ at iteration 100 of the CPF-BS-SEM and CPF-AS-SEM algorithm on the Kitagawa model (1.5), where true parameter is $\theta^* = (1, 10)$ and number of observations is $T = 100$. The algorithms are run with fixed number of particles ($N_f = 10$) and different number of trajectories ($N_s \in \{1, 5, 10\}$).

In order to compute $m(x_{t-1})$, we run a Runge-Kutta scheme (order 5) to integrate the system (1.7) on the time interval $[0, dt]$ with initial condition $x_{t-1}$. The value of $dt$ affects the nonlinearity of the dynamical model $m$ (see top panels of Figure 2.4). For the sake of simplifying illustrations, error covariances are assumed to be diagonal. More precisely we assume that $Q = \sigma_Q^2 I_3$ and $R = \sigma_R^2 I_2$ and the unknown parameter to be estimated is $\theta = (\sigma_Q^2, \sigma_R^2) \in \mathbb{R}^+ \times \mathbb{R}^+$. Note that an analytical solution can be derived for the **M-step** of the EM algorithm in this constrained model. It leads to the following expression for updating the parameters in the iteration $r$ of the EM algorithm

$$\hat{\theta}_r = \left(\hat{\sigma}^2_{Q,r}, \hat{\sigma}^2_{R,r}\right) = \left(\frac{\mathrm{Tr}[\hat{Q}_r]}{3}, \frac{\mathrm{Tr}[\hat{R}_r]}{2}\right) \tag{3.21}$$

where $\hat{Q}_r$ and $\hat{R}_r$ come from Eq. (3.20). The initial parameter value of the EM algorithm is drawn using a uniform distribution $\hat{\theta}_0 \sim \mathcal{U}([0.5, 2] \times [1, 4])$.

For the first experiment we simulate $T = 100$ observations of the L63 model (1.6) with the model time step $dt = 0.15$ (it corresponds to around 20 loops of the L63 system) and true parameter $\theta^* = (1, 2)$ (shown on Figure 3.15). The CPF-BS-SEM and CPF-AS-SEM algorithms are compared on Figure 3.13. With only $N_f = N_s = 20$ particles/realizations, these two methods provide reasonable estimates of the parameters. The comparison has been done in different scenarios, with varying true parameter values $\theta^*$, and similar results were obtained. A lower

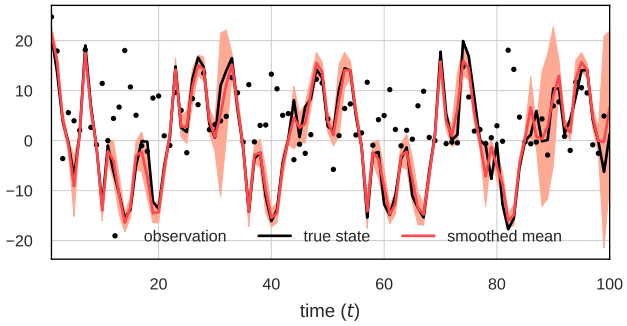


**Figure 3.12** – Reconstruction of the true state using CPF-BS-SEM with 10 particles/realizations on the Kitagawa model (1.5) given $T = 100$ observations. Smoothed means and 95% confidence intervals of all realizations simulated from the last 10 iterations of the algorithm are presented.

number of particles and realizations (eg. $N_f = N_s = 10$) can be used in these SEM algorithms but more iterations are needed (eg. 200) to obtain appropriate conditioning trajectories.

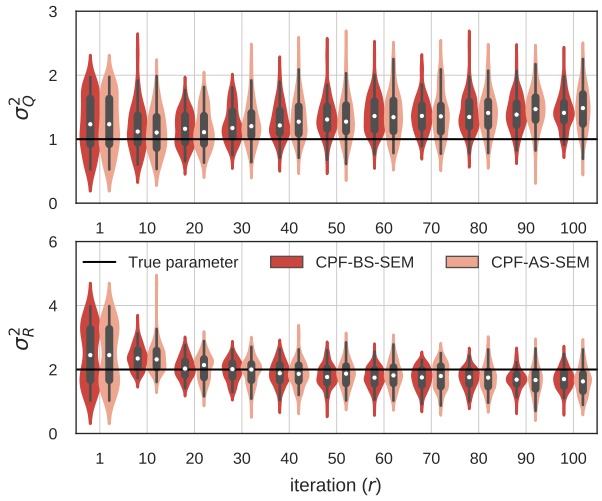


**Figure 3.13** – Comparison between CPF-BS-SEM and CPF-AS-SEM on the L63 model (1.6) with model time step $dt = 0.15$, true parameter $\theta^* = (1, 2)$ and $T = 100$ observations. Results obtained by running 100 repetitions of these methods with 20 particles/realizations and 100 iterations. The empirical distribution of parameter estimates is represented every 10 iterations using one violin object with (black) quantile box and (white) median point inside.

In the second experiment, we also compare the results obtained with the ones of the EnKS-EM algorithm. The EnKS-EM algorithm with a low number of $N$ of members often gets numerical issues when computing empirical covariances. Values of $N$ in the range $[20, 1000]$ has been chosen in lots of DA schemes using EnKS [50, 58, 95, 126, 156, 163, 164]. We have chosen to run the three algorithms with 20 members/particles to have comparable computational costs. The experiment is run on different simulated sequences of length $T = 100$, where the model

time step in (1.7) varies $dt \in \{0.01, 0.08, 0.15\}$. According to Figure 3.14, the CPF-BS-SEM algorithm gives better estimates compared to the CPF-AS-SEM and EnKS-EM algorithms. The bias and variance of the estimates obtained with the three algorithms increase with $dt$ representing the nonlinearity of the dynamic model. Note that the discrepancy increases quicker for the EnKS-EM algorithm. We found that it completely fails when $dt = 0.25$ whereas the CPF-BS-SEM and CPF-AS-SEM algorithms still give reasonable estimates (not shown; the Python library is available for such tests). This illustrates that the EnKS-EM algorithm is less robust to nonlinearities compared to the two algorithms based on the conditional particle filter.

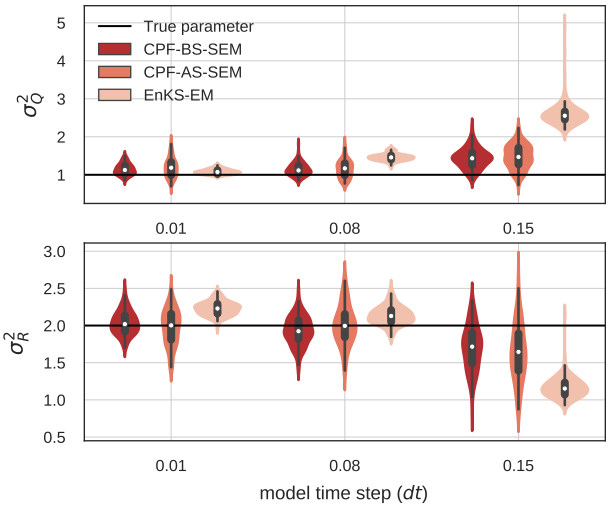


**Figure 3.14** – Comparison of the estimates of $\theta = (\sigma_Q^2, \sigma_R^2)$ for the CPF-BS-SEM, CPF-AS-SEM and EnKS-EM algorithms with 20 members/particles for the L63 models (1.6) with varying model time step $dt \in \{0.01, 0.08, 0.15\}$, true parameter $\theta^* = (1, 2)$ and number of observations is $T = 100$. Each empirical distribution of the estimates of $\theta$ is computed using 100 repetitions of each algorithm at the final iteration $r = 100$.

Figure 3.15 shows the results obtained when reconstructing the latent space using the CPF-BS-SEM algorithm (using the same approach as for the linear model, based on storing the sequences simulated in the last 10 iterations of the algorithm). The smoothed means of three variables are close to the true state and RMSEs for each component are $(0.8875, 1.0842, 1.2199)$. 95% CIs cover the true state components with respect to CPs $(87\%, 84\%, 88\%)$. Although the second variable $x_2$ is unobserved the algorithm provides a reasonable reconstruction of this component.

Finally, we perform a cross-validation exercise to check the out-of-sample reconstruction ability of the proposed method. Two sequences of the observations consisting of a learning sequence with length $T = 100$ and a test sequence with length $T' = 1000$ are simulated by

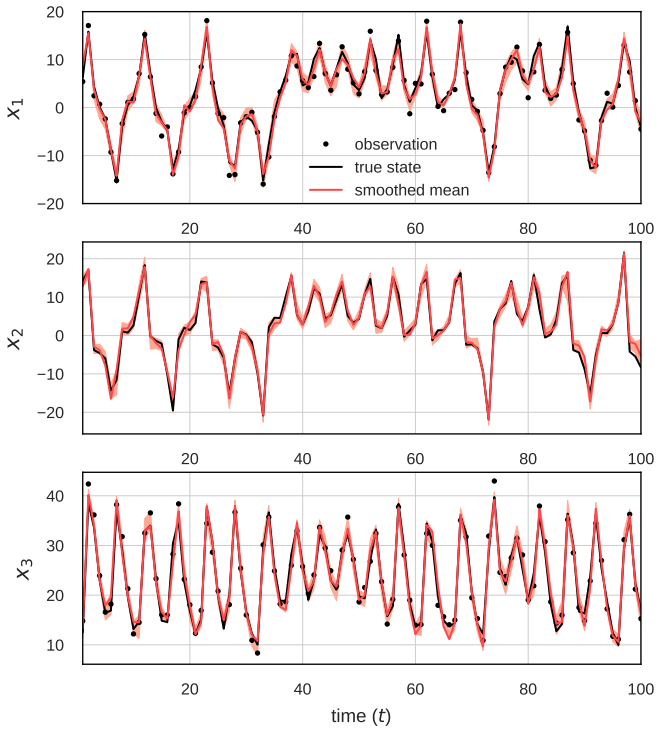


**Figure 3.15** – Reconstruction of the true state for the L63 model (1.6) with $dt = 0.15, T = 100$ by using the CPF-BS-SEM algorithm with 20 particles/realizations. Smoothed mean and 95% confidence interval of all realizations of the last 10 iterations of the algorithm are computed.

the L63 model (Eq. 1.6) with the true parameter values. Given the learning sequence, we first run the CPF-BS-SEM and the CPF-AS-SEM algorithms for estimating parameters. The mean values of the final estimates shown on Figure. 3.13 are computed. This provides point estimates of the unknown parameters. Then the CPF-BS and CPF-AS algorithms are run on another test sequence of observations with their corresponding estimated parameters. This provides an estimate of the smoothing distribution. Table 3.1 gives RMSEs and CPs for the unobserved component of all smoothing samples with respect to number of iterations in $\{5, 10, 50, 100\}$. As expected the CPs of the two algorithms tend to 95% when the number of samples is large enough. The CPF-BS smoother clearly outperforms the CPF-AS as it gets smaller RMSEs and larger CPs with a small number of iterations and thus less computational cost. Similar conclusions hold true when comparing the scores for the first and third components (results not shown here).

**Table 3.1** – Comparison of the reconstruction quality between the CPF-BS and CPF-AS smoothers on a test sequence in terms of root of mean square error (RMSE) and coverage probability (CP). The parameters are estimated on a sequence of length $T = 100$ (mean values of the final estimates shown on Figure 3.13). The CPF-BS and CPF-AS algorithms are run on a test sequence simulated using the L63 model (1.6) with $dt = 0.15, T' = 1000, \theta^* = (1, 2)$. The two scores are computed on the second component of the samples drawn from these smoothers with 20 particles/realizations.

| **number of iterations** | | 5 | 10 | 50 | 100 |
|---|---|---|---|---|---|
| **CPF-BS** | **RMSE** | 1.5310 | 1.2507 | 1.0098 | 0.9891 |
| | **CP** | 83.8% | 88.6% | 94.3% | 95.7% |
| **CPF-AS** | **RMSE** | 2.1595 | 1.5711 | 1.0125 | 0.9769 |
| | **CP** | 58.9% | 78.5% | 92.0% | 94.8% |

## 3.4 Conclusions

In this chapter, we show for SSMs with non-large dimension, CPF-BS and CPF-AS algorithms permit to simulate conditioning trajectories of the latent state given observations with a low number of particles ($5 - 20$, see also in [4, 99, 101, 102, 150]) compared to the standard particle smoother algorithms. That encourages to apply CPF-based smoothing algorithms in DA contexts. Compared to the EnKS, these algorithms permit to consider highly nonlinear and/or non-Gaussian SSMs. The CPF-BS sampler leads to a better description of the smoothing distribution at the same computational cost as the CPF-AS which only permits to generate one trajectory. Combined with EM methodology, it provides an efficient method to estimate the parameters such as error covariances. It also permits a better estimation of the uncertainty on the reconstructed trajectories in DA.

Chapter 4

# Reconstruction and estimation for non-parametric nonlinear state-space models

Inference problems such as state reconstruction, parameter identification and system control involving nonlinear state-space models with no close form nor any expression as a system of ODEs are clearly analytically intractable. To tackle the issues, an original algorithm combining sequential Monte Carlo method and non-parametric estimation with a stochastic Expectation-Maximization optimization algorithm is proposed. The algorithm allows to retrieve an estimation of the dynamical model, of the posterior distribution of the state and of the variance of the observation error from a noisy time series (or a time series observed with errors in measurements). In the chapter, we first motivate the objectives, then we describe the algorithm and an extensive simulation study illustrates results obtained.

## 4.1 Introduction

One of the classical problems in time series analysis consists in identifying a dynamical model from noisy data. Ignoring the noise in the inference procedure may lead to biased estimates for the dynamics, and this becomes more and more problematic when the signal-to-noise ratio increases.

**State-space models (SSMs)** provide a natural framework to study time series with observational noise in environment, economy, computer sciences, etc [51,118,160]. Some applications include data assimilation, system identification, model control, change detection, missing-data

imputation [5, 24, 59]. Here we recall a general SSM defined through the following equations,

$$\begin{cases} X_t = m\left(X_{t-1}\right) + \eta_t, \quad [hidden] & (4.1) \\ Y_t = H(X_t) + \epsilon_t, \qquad [observed]. & (4.2) \end{cases}$$

The latent process $\{X_t\}$ is a Markov chain whose transition kernel $p\left(x_t|x_{t-1}\right)$ depends on the deterministic model $m$ and the distribution of the white noise $\{\eta_t\}$. The observations $\{Y_t\}$ are assumed to be conditionally independent given the latent process. And the conditional probability distribution function of $Y_t$ given $X_t = x_t$, denoted by $p(y_t|x_t)$, describes the link between the latent space and the observations. It depends on the deterministic function $h$ and the distribution of the white noise sequence $\{\epsilon_t\}$, which is assumed to be independent of $\{\eta_t\}$. Throughout this chapter we assume that $H$ is known (typically $H(x) = x$) and that the white noise sequences have Gaussian distributions with $\eta_t \sim \mathcal{N}(0, Q)$ and $\epsilon_t \sim \mathcal{N}(0, R)$. The Gaussian assumption is a classical assumption for many applications but the proposed methodology is general enough to handle the non-Gaussian case. We assume that the covariance matrices $Q$ and $R$, which describe respectively the level of noise in the dynamics and in the observations, depend on an unknown parameter $\theta$.

In this chapter, we are interested in situations where the dynamical model $m$ is unknown or numerically intractable. To deal with this issue, a classical approach consists in using a simpler parametric model to replace $m$. However, it is generally difficult to find an appropriate parametric model which can reproduce all the complexity of the phenomenon of interest. In order to enhance the flexibility of the methodology and simplify the modeling procedure, we propose in this chapter to use a non-parametric approach to estimate $m$. Such **non-parametric SSMs** were originally proposed by [95, 154, 155] for data assimilation in oceanography or meteorology. In these application fields, a huge amount of historical datasets recorded using remote and in-situ sensors or obtained through numerical simulations is now available and this promotes the development of data-driven approaches. A non-parametric estimate $\hat{m}$ of $m$ was built using the available observations and the other quantities which appear in Eq. (4.1) (distribution of $\eta_t$ and conditional distribution $p(y_t|x_t)$) were assumed to be known). This non-parametric estimate was plugged into usual filtering and smoothing algorithms to reconstruct the latent space $X_{0:T} = (X_0, ..., X_T)$ given observations $y_{1:T} = (y_1, ..., y_T)$. It was checked, using numerical experiments on toy models, that replacing $m$ by $\hat{m}$ leads to similar results if the sample size used to estimate $m$ is large enough to ensure that $\hat{m}$ is "close enough" to $m$. Several non-

parametric estimates of $m$ were considered in [95]. The authors first used a nearest neighbors method, also known as the Nadaraya-Watson approach in statistics [60, 61] and analog method in meteorology [7, 175]. This is probably the most natural but better results were obtained with a slightly more sophisticated estimator known as local linear regression (LLR) [35, 38, 61]. Based on these results, LLR is also used in this work. Some applications to real data are discussed in [59, 153].

From a statistical point of view, the proposed model is semi-parametric with a parametric component for the white noise sequences whose distributions are described by a parameter $\theta$ and a non-parametric component for the dynamical model $m$. When working with such SSMs, we may have to tackle the different inference problems discussed below.

- **Reconstruction of the latent process (smoothing algorithms)**. Here we assume that the SSM is known, i.e. that model $m$ is known (or eventually replaced by an estimate $\hat{m}$) and that the parameter $\theta$ is known. The aim is to compute the conditional distribution of the latent state $X_{1:T}$ given observations $y_{1:T}$. Many algorithms have been proposed in the literature [23, 24, 46, 49, 69]. Recently, [4, 99, 102, 171] have developed conditional particle smoothers which are able to iteratively simulate the hidden state with few particles needed. In this work we propose to use the Conditional Particle Filter-Backward Simulation (CPF-BS) presented in [101, 102, 171] and further discussed in [30].

- **Parametric estimation.** Here we assume that the model $m$ is known but that the parameter $\theta$ is unknown. This leads to a classical parametric estimation problem where $\theta$ is estimated from the available observations $y_{1:T}$. In such situation, Expectation-Maximization (EM) algorithm and its variants are often used to perform maximum likelihood estimation [41, 42, 44, 86, 98, 110]. The E-step consists in computing the conditional distribution of the latent space given the observations and thus the EM algorithm needs to be combined with a smoothing algorithm [5, 50, 86, 141, 149, 151]. In [30] (corresponding to Chapter 3 in this thesis), it was proposed to use the CPF-BS algorithm in the E-step of the EM algorithm and found that the combination of CPF-BS algorithm and EM recursions leads to an efficient numerical procedure to estimate the parameters of nonlinear SSMs. This approach is also used in this chapter.

- **Non-parametric estimation of the dynamical model**. When the dynamical model $m$ is unknown and replaced by a non-parametric estimate, two situations may happen. In

the first one, a learning sequence of the process $\{X_t\}$ is available. In this situation, the non-parametric estimate $\hat{m}$ of $m$ can be based on this learning sequence and the parametric setting described above can then be used to estimate $\theta$ after replacing $m$ by $\hat{m}$. In practice, it means that we need "perfect" observations of the state (with no observational error) but this is generally not available. In the second situation, only a sequence $y_{1:T}$ of the process $\{Y_t\}$ with observational errors is available. This is a more usual situation in practice but it makes the estimation of $m$ more complicated.

In this chapter we mainly focus on the non-parametric estimation problem and discuss the estimation of $m$ and $\theta$ using a sequence $y_{1:T}$ with observational error. This is the more challenging problem in the problems listed above. A simple approach to estimate $m$ would consist in computing a non-parametric estimate $\hat{m}$ based on the sequence $y_{1:T}$ instead of a sequence of the process $\{X_t\}$ but this is not satisfactory since the conditional distributions of $X_t$ given $X_{t-1} = x_{t-1}$ and $Y_t$ given $Y_{t-1} = y_{t-1}$ do not coincide. This is illustrated on Figure 1.8 obtained using the nonlinear univariate SSM defined as in Eq. (1.4) where $\theta = (Q, R)$ is fixed by $(0.1, 0.1)$. The left plot shows a scatter plot of $(X_{t-1}, X_t)$ of the true state process and a non-parametric estimate $\hat{m}$, obtained using LLR, which is reasonably close to $m$. The right plot shows a scatter plot of $(Y_{t-1}, Y_t)$ of the observed sequence. Note that $Y_t$ is obtained by adding a random noise to $X_t$ and this has the effect of blurring the scatter plot by moving the points both horizontally and vertically. The $\hat{m}$-curve shows a non-parametric estimate of $E[Y_t|Y_{t-1}]$ obtained using LLR, which is a biased estimate of $m$. In a regression context, it is well known from the literature on errors-in-variables models that observational errors in covariates lead, in most cases, to a bias towards zero of the estimator of the regression function [26]. One of the classical approach to reduce the bias is to introduce instrumental variables which help to get information about the observational error. This approach has been adapted for linear first-order autoregressive models in [111] and further studied in [94]. Besides, [26] gave an overview of different methods to build consistent estimators in the context of regression. Among them, we notice the local polynomial regression and the Bayesian method for non-parametric estimation but, as far as we know, they are not generalized for time series.

In order to improve the estimate of $m$, we propose an original procedure where the non-parametric estimate $\hat{m}$ is updated at each iteration of the EM recursions using the smoothing trajectories simulated with the CPF-BS algorithm in the E-step. It permits to correct sequentially the estimation error and reduce the bias in the estimate of $m$. This method can be interpreted

as a generalization of the Bayesian approach of [26] for time series. This is the main contribution of this chapter. All the codes of the proposed algorithm used for numerical experiments in this chapter are available on `https://github.com/tchau218/npSEM`.

The chapter is organized as follows. In Section 4.2, we discuss the estimation of the parametric component using EM recursions. Then, in Section 4.3, we extend this algorithm to estimate both the parametric and non-parametric components in SSMs. In order to validate the proposed methodology, we perform some simulation experiments on toy models in Section 4.4. The chapter ends with some concluding remarks in Section 4.5.

## 4.2 Parametric estimation in state-space models

Let us now consider the problem of estimating the parametric part of the SSM (4.1). The aim is to estimate $\theta \in \Theta$ (an appropriate set of the unknown parameter) given a sequence $y_{1:T}$ of noisy observations. Here we assume that the true dynamical model $m$ is known or that an estimate $\hat{m}$ has already been fitted using other information, such as an observed sequence of the state. The notation $\mathfrak{M}$ stands for the true dynamical model $m$ if it is known, or for the prior estimate $\hat{m}$ otherwise.

The main idea of the algorithm is to iterate a two-step procedure. For each iteration $r \geq 1$, the first step (E-step) consists in computing $p\left(x_{0:T}|y_{1:T};\theta_{r-1}\right)$, a conditional distribution of the latent process $x_{0:T}$ given the observations $y_{1:T}$ and the previous parameter value $\theta_{r-1}$. The second step (M-step) consists in updating the parameter value by maximizing an intermediate function (4.3) obtained by integrating the complete likelihood function over the so-called smoothing distribution computed in the E-step.

$$\mathbb{E}_{p_r}\left[\ln p\left(X_{0:T}, y_{1:T};\theta\right)\right] \triangleq \int \ln p\left(x_{0:T}, y_{1:T};\theta\right) \times p(x_{0:T}|y_{1:T};\theta_{r-1})\, dx_{0:T} \tag{4.3}$$

For nonlinear (non-Gaussian) SSMs, $p\left(x_{0:T}|y_{1:T};\theta_{r-1}\right)$ does not have a tractable analytical expression. However, sequential Monte Carlo (SMC) algorithms [23,46,49,69] allow to generate sequences of this conditional distribution. They provide weighted samples $\{x_{0:T}^{(i)}, w_T^{(i)}\}_{i=1:N}$ which allow to approximate the posterior distribution using the empirical estimate

$$\hat{p}_r(dx_{0:T}|y_{1:T}) = \sum_{i=1}^{N} w_T^{(i)} \delta_{x_{0:T}^{(i)}}(dx_{0:T}), \tag{4.4}$$

According to the empirical distribution (4.4), one can obtain approximations of the expectation (4.3). That leads to the so-called Stochastic EM (SEM) algorithms.

One of the key points of an SEM algorithm is to compute an efficient approximation of Eq. (4.3). If $N$ is large the law of large numbers implies that $\mathbb{E}_{\widehat{p_r}}\left[\log p(X_{1:T}, y_{1:T}; \theta)\right]$ is a good approximation of the true expectation (4.3) and the SEM algorithm is close to EM algorithm. In order to save computational time, the number of simulated samples $N$ can be reduced by using other extensions of the SEM algorithm [41, 170] which are not presented in this chapter for simplifying the presentation. However, the SEM algorithms and their variants using standard particle approaches [5, 86, 116, 141] still suffer from another issue. To simulate good trajectories in the E-step, these particle smoothers are typically required a large number of particles (in a range $[10^2 - 10^6]$), and this has to be done at each iteration of the EM algorithms (see [64] for a recent review). Conditional SMC (CSMC) samplers, as known as combinations of SMC and Markov Chain Monte Carlo (MCMC) approaches, have been developed as alternatives. The first CSMC samplers, so-called Conditional Particle Filters (CPFs), were introduced in [4, 100] and they were used combined with EM algorithms in [98, 102, 149]. CPF algorithms simulate samples of $x_{0:T}$ conditionally on the current value of the parameter $\theta$ and the current value of the state sequence (referred to as the conditioning sequence). They allow to build a Markov chain which has the exact smoothing distribution $p(dx_{0:T}|y_{1:T}; \theta)$ as invariant distribution (see [30, 150] for numerical illustrations). Note that the convergence rate does not depend on the number of particles but on the number of iterations of the sampler.

Nevertheless, as many sequential smoothing algorithms, when the length $T$ of the observed sequence is large, CPF algorithms suffer from sample impoverishment. More precisely, at the end of the CPF, all the trajectories tend to share the same ancestors and the rate of convergence may be very slow. A way to reduce impoverishment and low mixing is to run a Backward Simulation algorithm after the CPF one. Backward simulation (BS), proposed initially in [69], is a natural technique to simulate the smoothing distribution given the (forward) filter outputs (see [17, 46, 102]). This leads to the Conditional Particle filter-Backward simulation (CPF-BS) sampler (see Algorithm 14 in Appendix). Recently, [30] proposed to use the CPF-BS smoothing algorithm in conjunction with the SEM algorithm. The authors showed that the method outperforms several existing EM algorithms in terms of both state reconstruction and parameter estimation, using low computational resources. All details can be found in Chapter 3. The SEM algorithm is reminded as follows.

**Algorithm 12: SEM algorithm for SSMs [SEM($\mathfrak{M}$)]**

**Initialization**: choose an initial parameter $\hat{\theta}_0$ and a conditioning trajectory.
For $r \geq 1$,
**(1) E-step**: generate $N$ trajectories $\{x_{0:T,r}^{(i)}\}_{i=1:N}$ by using CPF-BS algorithm (14) with the given conditioning sequence, the parameter value $\widehat{\theta}_{r-1}$, the dynamical model $\mathfrak{M}$ and the observations $y_{1:T}$, and deduce an empirical estimate $\widehat{p}_r$ of the smoothing distribution $p(x_{0:T}|y_{1:T};\widehat{\theta}_{r-1})$.
**(2) M-step**: compute an estimate of of $\theta$,

$$\widehat{\theta}_r = \arg\max_{\theta\in\mathbf{\Theta}} \mathbb{E}_{\widehat{p}_r}\left[\ln p(X_{0:T}, y_{1:T};\theta)\right]$$

end.

When the noises $(\eta_t, \epsilon_t)$ in the SSM (4.1) have Gaussian distributions with respect to unknown covariances $(Q, R)$, the following close form expressions can be derived as the estimators of $\theta = (Q, R)$ in the M-step.

$$\widehat{Q}_r = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\left[x_{t,r}^{(i)} - \mathfrak{M}\left(x_{t-1,r}^{(i)}\right)\right]\left[x_{t,r}^{(i)} - \mathfrak{M}\left(x_{t-1,r}^{(i)}\right)\right]^{\top}}{NT}, \tag{4.5}$$

$$\widehat{R}_r = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\left[y_t - H\left(x_{t,r}^{(i)}\right)\right]\left[y_t - H\left(x_{t,r}^{(i)}\right)\right]^{\top}}{NT}. \tag{4.6}$$

## 4.3 Non-parametric estimation in state-space models

In the previous section, it is assumed that the dynamical model $m$ is known or that an estimate $\hat{m}$ is available. The last case may happen, for example, when the evolution model (4.1) is observed without observational error on some time intervals and thus observations of the process $\{X_t\}$ are available. When no parametric model is available for $m$, a non-parametric estimate of $m$ can be built. Here, we focus on Local Linear Regression (LLR), but other non-parametric estimation methods can be easily plugged into the methodology described in this section (see Chapter 1 [Section 1.2.2.1] for details). In [35], the authors discussed the practical implementation of LLR and presented several interesting case studies. The asymptotic theory of these estimators was described in [136] (see also [61] for time series). The idea of LLR is to locally approximate $m$ by the first-order Taylor's expansion, $m(x') \approx m(x) + \nabla m(x)(x' - x)$, for any $x'$ in a neighborhood of $x$. In practice, the intercept $m(x)$ and the slope $\nabla m(x)$ are estimated by minimizing a weighted mean square error. The weights are defined relating to a kernel. Here we choose to

use the tricube kernel (Eq. 1.28) as in [35]. This kernel used in many implementations of LLR has a compact support and it is smooth at its boundary. One of the advantages of considering such a bounded kernel is that it reduces the computation of the points out of the support. For the tricube kernel, it is usual to fix a bandwidth $h$ equal to the half width of the kernel support. An alternative is to perform LLR on the $k$-nearest neighborhood of $x$ (see [7, 91, 119, 162]). In this case, the support is defined as the smallest rectangular area which contains the $k$ nearest neighbors. It leads to an adaptive way of defining the bandwidth $h = h_x$ because the support depends on the location $x$.

For the example presented in the general introduction of the thesis, the LLR estimate leads to the $\widehat{m}$-curve of the left panel of Figure. 1.8 when the time series $\{X_t\}$ is observed without observational error. As in parametric problems, ignoring the observational error causes inconsistent estimation of $m$ as it is illustrated on the right panel of Figure. 1.8 where the $\widehat{m}$-curve corresponds to the LLR estimate based on a sequence of the noisy process $\{Y_t\}$. We now propose to adapt the SEM algorithm (Algorithm 12) introduced in the previous section to better estimate $m$ in the case where $m$ is unknown and the only available observed data is a sequence $y_{1:T}$ of the process $\{Y_t\}$.

The key idea of the algorithm is to update a non-parametric estimate of $m$ at each iteration of the SEM algorithm using the smoothing trajectories simulated in the E-step. It permits to reduce sequentially the bias induced by the observation noise. More details are given in Algorithm 13.

$p(x_{0:T}, y_{1:T}; \theta, \widehat{m}_{r-1})$ denotes the complete likelihood function where $\widehat{m}_{r-1}$ is substituted to $m$. Algorithm 13 looks similar to Algorithm 12. The main difference between the two algorithms is that in the second one, at iteration $r$, $\mathfrak{M}$ is an approximation of $m$ defined using a non-parametric estimate based on the current sequences $\{\tilde{x}^{(i)}_{0:T,r}\}_{i=1:N}$ of the state $\{X_t\}$. This is illustrated on Figure. 4.1 using the toy model (1.4). At the first iteration, an initial parameter value $\hat{\theta}_0$ is chosen and a non-parametric estimate $\widehat{m}_0$ of $m$ is computed based on the observed sequence $y_{1:T}$. Then, in the E-step, a smoothing algorithm is run. This produces smoothed trajectories with less observation noise than in the original sequence. In the M-step, the parameter value $\theta$ and the non-parametric estimate of $m$ are updated by fitting the SSM using the smoothed trajectories as possible trajectories for the true state. To simplify the illustration, the LLR estimation $\widehat{m}_r$ of $m$ is learned based on one simulated trajectory denoted by $\tilde{x}_{0:T,r}$. As shown on Figure. 4.1, the distribution of $\tilde{x}_{0:T,r}$ is closer and closer to the one of $X_{0:T}$ (see

**Algorithm 13: SEM-like algorithm for non-parametric SSMs [npSEM]**

**Initialization**: choose an initial parameter $\hat{\theta}_0$, set the first learning sequence $\tilde{x}_{1:T,0} = y_{1:T}$ and a conditioning trajectory, and compute the corresponding LLR estimate $\widehat{m}_0$ on $\tilde{x}_{1:T,0}$.
For $r \geq 1$,
**(1) E-step**: generate $N$ trajectories $\left\{\tilde{x}^{(i)}_{0:T,r}\right\}_{i=1:N}$ by using CPF-BS algorithm (14) with the given conditioning sequence, the parameter value $\widehat{\theta}_{r-1}$, the dynamical model $\mathfrak{M} = \widehat{m}_{r-1}$ and the observations $y_{1:T}$, and deduce an empirical estimate $\widehat{p}_r$ of the smoothing distribution $p(x_{0:T}|y_{1:T}; \widehat{\theta}_{r-1})$.
**(2) M-step**:

i. compute an estimate of $\theta$,

$$\widehat{\theta}_r = \arg\max_{\theta\in\Theta} \mathbb{E}_{\widehat{p}_r}\left[\ln p(X_{0:T}, y_{1:T}; \theta, \widehat{m}_{r-1})\right].$$

ii. compute a LLR estimate $\widehat{m}_r$ of $m$ with $\left\{\tilde{x}^{(i)}_{0:T,r}\right\}_{i=1:N}$.

end.

Figure. 1.8) when $r$ increases and this permits to reduce the bias in the non-parametric estimate of $m$. The spirit of this algorithm is close to the one of the iterative global/local estimation (IGLE) algorithm of [176] for estimation of mixture models with mixing proportions depending on covariates.

Algorithm 13 is referred to a SEM-like algorithm because in the M-step the parameter $\theta$ is estimated as in Algorithm 12. But we have no warranty that the E-step leads to an increase of a likelihood function, and the M-step is composed of a likelihood maximization for $\theta$ and a *"data update"* for $m$ which cannot be written as a solution of an optimization problem.

At each iteration, the LLR estimate of $m$ is updated and a bandwidth has to be chosen for the kernel. This is done using cross-validation to minimize a mean square error, and a different optimal bandwidth is searched on a grid at each iteration. Note also that the time series $\{\tilde{x}^{(i)}_{0:T,r}\}_{i=1:N}$ are used both as a learning set to estimate $m$ at iteration $r$ and for the propagation step of the CPF-BS smoother in Algorithm 13. We found, using numerical experiments, that it leads to over-fitting. In order to reduce over-fitting, at each iteration $r$ and for each time $t$, $\widehat{m}_r(\tilde{x}^{(i)}_{t-1,r})$ is estimated using LLR based on the subsample $\tilde{x}^{(i)}_{0:(\ell-t),(t+\ell):T,r}$ where the sequence $\tilde{x}^{(i)}_{(\ell-t+1):(\ell+t-1)}$ is removed from the learning sequence. The lag $\ell$ is chosen as a priori.

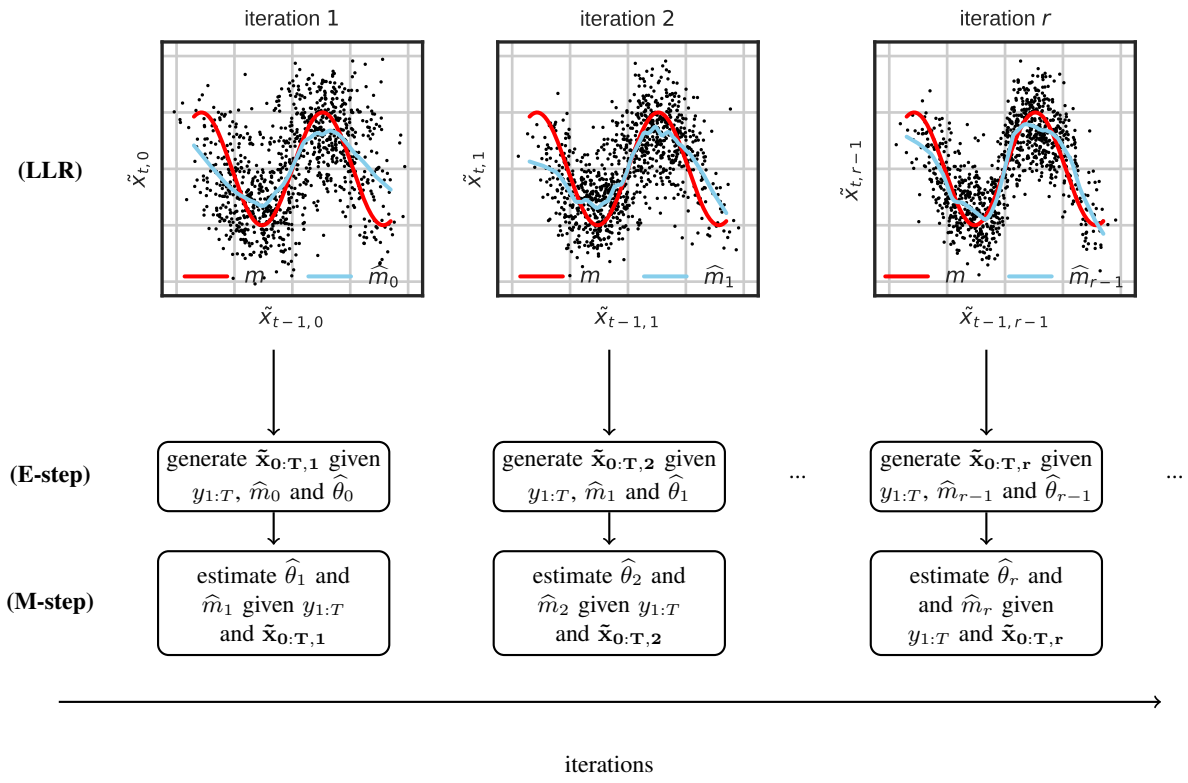


**Figure 4.1** – An illustration of Algorithm 13 (npSEM) on the sinus model (1.4). For each iteration, the LLR estimate $(\widehat{m}_r)_{r\geq 0}$ of the dynamical model $m$ is learned on the smoothed samples generated from the previous iteration ($\tilde{x}_{1:T,0} = Y_{1:T}$ for the first iteration).

## 4.4 Simulation results

A simulation study is now executed in order to explore some properties of the proposed algorithms and the performances of the proposed estimates. Two different toy models are considered. The first one is the sinus model (1.4). It is a univariate model which allows to plot the dynamical function $m$ and its estimates. The second model is the L63 model defined as in Eq. (1.6).

For each example, an observation sequence $y_{1:T}$ of length $T = 1000$ is simulated. Then, the SEM and SEM-like algorithms are run to estimate $Q$, $R$ and $m$ (if it is unknown). The SEM is run with both $\mathfrak{M} = m$ and $\mathfrak{M} = \widehat{m}$. In the CPF-BS algorithms used in the SEM and SEM-like algorithms, the numbers of particles for the filtering step and realizations for the backward simulation step are fixed to $N_f = 10$ and $N_s = 5$ (see Appendix 4.5).

The initial conditioning sequence for the CPF-BS has to be chosen carefully to help the algorithm converge quickly to the target posterior distribution. For that, an Ensemble Kalman

Smoother (EnKS, [58]) is run with 20 members. It provides an approximation of the mean of the smoothing distribution which is used as the first conditioning trajectory.

The convergence of the SEM and SEM-like algorithms is illustrated by plotting the evolution of the estimates of $Q$ and $R$ (or mean of their diagonal values if they are matrices) with respect to the iteration number. In the SEM-like algorithm, we also expect to improve the estimation of $m$ from one iteration to the next one. In order to illustrate that, the following likelihood ratio statistics (see in [61]) is considered.

$$\mathbf{T}_r = -2 \ln \frac{L_0}{L_r} \tag{4.7}$$

as $L_0 = \prod_{t=1}^{T} p(x_t|x_{t-1})$ is the Markovian likelihood of a trajectory $x_{0:T}$ of the latent state computed with the true dynamical model $(m, Q)$. Remark that in unidimensional Gaussian cases, this likelihood depends only on forecast error (Eq. 2.9) and the variance $Q$. Similarly, the likelihood $L_r$ with respect to the SEM or SEM-like algorithm is computed using the same expression where the true dynamical model with $(m, Q)$ is replaced by their estimates at the iteration $r$ of the algorithm. If the fitted dynamical model is close to the true one then $\mathbf{T}_r$ is close to 0, whereas negative values for $\mathbf{T}_r$ indicate a large discrepancy between the two dynamical models.

Finally, the estimated parameters are plugged into the CPF-BS algorithm to infer a latent state time series $x'_{0:T}$ given an observed sequence $y'_{1:T}$. The second lines of Tables 4.1 and 4.2 report the reconstruction errors between smoothed state time series and the true state time series for the same observations $y'_{1:T}$. More precisely, the smoothing is performed using the CPF-BS algorithm for a fixed $\mathfrak{M}$ with $N_f = 10$ , $N_s = 5$ and 100 iterations. The conditional mean $E(X_t|y'_{1:T})$ is approximated by the empirical mean $\hat{x}'_t$ computed with the output time series of the CPF-BS and the error is measured using the root mean square error

$$RMSE(smoothing) = \sqrt{\frac{\sum_{t=1}^{T} \|x'_t - \hat{x}'_t\|^2}{T}}. \tag{4.8}$$

### 4.4.1 Sinus model

We first consider again the sinus model (1.4) with true parameter value $\theta^* = (Q^*, R^*) = (0.1, 0.1)$. A simulated time series is shown on Figure 1.2 where the full line represents a

realization of the state $\{X_t\}$ and the dots represent the corresponding observations of the $\{Y_t\}$ process.

On Figure. 4.2, error variances $\hat{Q}_r$ and $\hat{R}_r$ estimated by SEM($m$), SEM($\hat{m}$) and SEM-like algorithms are displayed for the first 200 iterations respectively. Here SEM($m$) [resp. SEM($\hat{m}$)] denotes the SEM algorithm of Section 4.2 when the true dynamical model $m$ is known [resp. replaced by a non-parametric estimate $\hat{m}$ obtained using a sequence of the state $x_{0:T}$ and LLR]. Since the length of the time series is large ($T = 1000$), it is expected that the estimates from SEM($m$) and SEM($\hat{m}$) are close to each other and to the true values of $Q$ and $R$. The small difference observed between the two curves as well as their erratic decrease is due to the randomness of the Monte Carlo steps in the algorithms. The SEM-like algorithm converges slower but this algorithm enables to retrieve the variance of the observational error and separate the noise associated with the observations and the noise coming from the stochastic dynamical system.

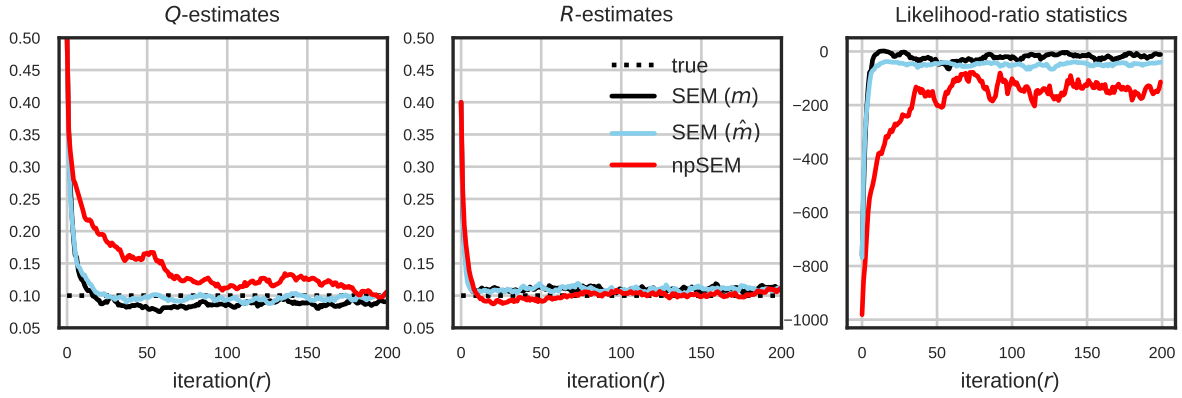


**Figure 4.2** – Comparison of the estimated parameters of SEM and npSEM algorithms on the sinus model (1.4). The left (resp. middle) panel shows the evolution of the Q (resp. R) estimates with respect to the iteration number of these algorithms. The right panel shows the evolution of the likelihood-ratio statistic (4.7).

The likelihood ratio shown on Figure 4.2 permits to assess the ability of the proposed algorithms in estimating the dynamical model (4.1). The values of the likelihood ratio associated to the SEM($\hat{m}$) algorithm stabilize after 10 iterations. It shows that if a sequence of the true state is available, then LLR gives a good estimate of $m$ and the SEM($\hat{m}$) algorithm gives an estimate of $Q$ close to the true value (shown in the first panel of Figure 4.2). The statistic associated to the SEM-like algorithm corresponds, at step 0, to the discrepancy between $m$ and the biased estimation of $m$ shown on the last panel of Figure 1.8. The increase of the likelihood-ratio statistics shows that the SEM-like algorithm allows to efficiently update the estimate $m$

and reduce the effect of the observational error and hence obtain a reasonable estimate of $Q$. The model reconstruction and the decrease of observational error are also illustrated on Figure 4.3 through scatter plots of a couple of variables at consecutive time steps $(t-1, t)$ at iteration 0 (observation, left panel) and at the last iteration (right panel) of the SEM-liked algorithm. In the middle, the scatter plot of a realization of the true model is shown. We can see that the SEM-like algorithm correctly retrieves the dynamical model from the noisy observations.

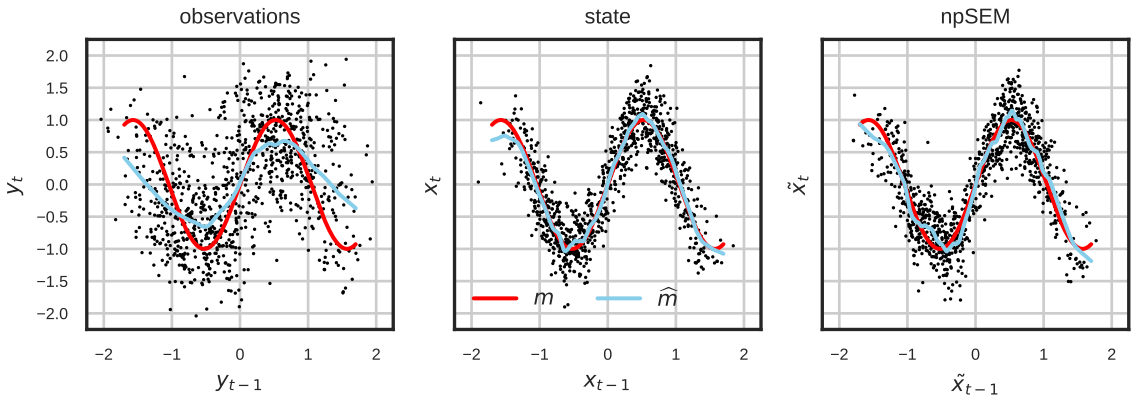


**Figure 4.3** – Scatter-plots of $(Y_{t-1}, Y_t)$ (left), $(X_{t-1}, X_t)$ (middle) and $\left(\widetilde{X}_{t-1}, \widetilde{X}_t\right)$ for the SSM defined by Eq. (1.4). $\widetilde{X}_t$ stands for one of realizations generated at the final iteration of the npSEM algorithm. The $\hat{m}$-curves show estimates of the conditional mean function $m$ obtained using LLR.

Table 4.1 reports RMSE. The first column corresponds to the true model and the forecasting error is close to $\sqrt{Q^*} = 0.3162$ which is expected. In the second column, $m$ is estimated using LLR estimate based on observations of the true state $x_{0:T}$. The forecasting errors of the first and the second columns are similar and this confirms that LLR provides a good estimation of $m$ in this situation. In the third column, $m$ is estimated using LLR based on observations of the noisy state $y_{1:T}$. The large error highlights the bias of this estimate. In the two next columns, $Q$ and $R$ are estimated by the SEM algorithms. The fourth column reports the RMSE when $m$ is known (SEM($m$) algorithm), the fifth one for $m$ is estimated by LLR without observational error (SEM($\hat{m}$) algorithm). This column should be compared to the first and second one respectively, and they show the extra error made when $Q$ and $R$ are unknown and estimated before running the smoothing algorithm to reconstruct the latent state (the forecast errors on the first line are the same). The errors are really close to each other. It shows again that the SEM algorithm is able to retrieve accurate estimates of $Q$ and $R$ and that the LLR estimate is close enough to the true $m$ to lead to a similar approximation of the smoothing mean $E(X_t|y_{1:T})$. Finally,

**Table 4.1** – RMSEs (Eqs. 2.9 and 4.8) for forecasting and smoothing of a state sequence of model (1.4). The parameters are estimated on a sequence of length $T = 1000$. The smoothing algorithms are run with 10 particles. $\theta^*$ denotes the true values of the parameters. $X, Y$ and $\widetilde{X}$ represent sequences generated from the true state process $\{X_t\}$, the observation process $\{Y_t\}$ and the npSEM algorithm, respectively.

| | $(m, \theta^*)$ | $(\hat{m}_X, \theta^*)$ | $(\hat{m}_Y, \theta^*)$ | $(m, \hat{\theta}_{SEM(m)})$ | $(\hat{m}_X, \hat{\theta}_{SEM(\hat{m})})$ | $(\hat{m}_{\widetilde{X}}, \hat{\theta}_{npSEM})$ |
|---|---|---|---|---|---|---|
| $RMSE(forecast)$ | 0.3072 | 0.3109 | 0.4528 | 0.3072 | 0.3109 | 0.3294 |
| $RMSE(smoothing)$ | 0.2520 | 0.2523 | 0.3041 | 0.2581 | 0.2553 | 0.2597 |

the most interesting column is the last one where $m$ is estimated using the npSEM algorithm. The forecast [resp. smoothing] RMSE is about 6% [resp. 3%] greater than the error made with the true values of $Q$, $R$ and $m$. This is a great improvement compared to the third column and this shows that the npSEM algorithm has efficiently reduced the bias in the estimation of the dynamical model due to the observational errors.

### 4.4.2 Lorenz 63 model

In real applications (see [65,84,105,166] for a few), dynamical systems are often multidimensional and observations can be missing. To reproduce such situations, results hereafter are given for an L63 model with randomly missing observations. The considered L63 SSM on $\mathbb{R}^3$ is defined as

$$\begin{cases} X_t = m(X_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, Q) \\ Y_t = H_t X_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, R_t), \end{cases} \tag{4.9}$$

True values of error covariances in the above model are fixed by $(Q^*, R_t^*) = (\sigma_Q^{2,*} I_3, \sigma_{R_t}^{2,*} I_{d_{Y_t}}) = (I_3, 2I_{d_{Y_t}})$ where $I_d$ and $I_{d_{Y_t}}$ denote the identity matrices with dimension in $d$ and $d_{Y_t} \in \{1, 2, 3\}$ (corresponding to the number of components observed at time $t$). The dynamical function $m$ at any value $x$ in $\mathbb{R}^3$ is computed by integrating the ODE system (1.7). For each time $t$, Eq. (1.7) is integrated by running a Runge-Kutta scheme (order 5). The value of $dt$ is fixed to 0.08. In the experiments, the length of the observed time series is $T = 1000$.

In Eq. (4.9), the measurement operator $H_t$ and the covariance $R_t$ depend on the time in order to take into account situations where some of the components of $X_t$ are not observed. For instance, if the full state is observed the $H_t = I_3$ and $I_{d_{Y_t}} = I_3$ whereas if only the first component is observed then $H_t = [1, 0, 0]$ and $I_{d_{Y_t}} = I_1$. In the experiments, 10% of the observations, chosen randomly with a uniform distribution among the $T$ times and the 3 components, are set to missing values. An example of the simulated time series is shown on Figure. 4.4. Given the

observed sequence, the SEM and SEM-like algorithms are run in order to estimate the parameter $\theta = (Q, \sigma^2_{R_t})$ and the model $m$ (if it is unknown).

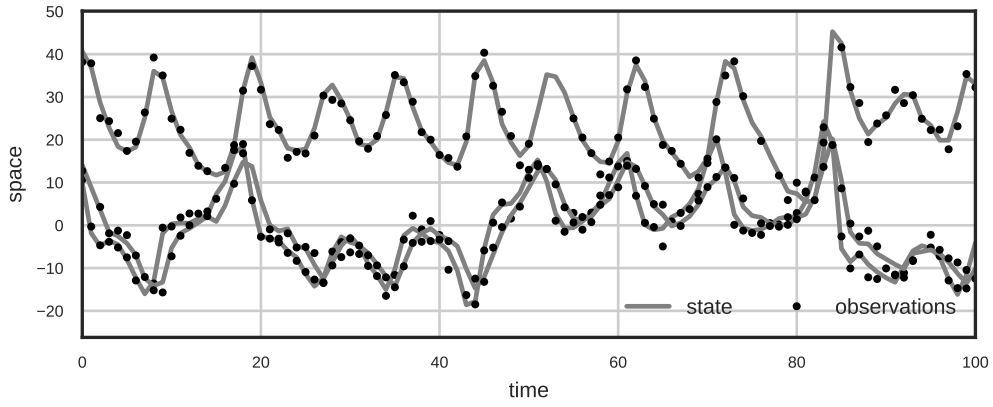


**Figure 4.4** – Time series of the state and observations simulated from the L63 model (4.9). 10% of the observations are set as missing values (e.g. shown in time interval $[50, 60]$).

Figure. 4.5 illustrates the convergence of the SEM and SEM-like algorithms for the L63 model (4.9). As for the previous example, the SEM($m$) and SEM($\widehat{m}$) have similar behaviours because the LLR well estimates $m$ when a long enough time series of the state $\{X_t\}$ is available. The rate of convergence of the npSEM algorithm seems to be close to the one of the other SEM algorithms but, after 500 iterations, $\sigma^2_Q$ is over-estimated and $\sigma^2_{R_t}$ is under-estimated. Remark however that the ratio of $\sigma^2_Q$ and $\sigma^2_{R_t}$, which is a key quantity of SSMs since it describes the relative weights of the dynamical model and the observation in the filters, is well estimated. We also found empirically that these biases seem to decrease with the percentage of missing values (not shown). Note that if the number of iterations is increased then the results do not change significantly. The covariance $Q$ measures implicitly the confidence which we can have in the state model. So its over-estimation is probably linked to the estimation error on $m$. According to Figure. 4.5, the difference of the likelihood-ratio statistics (4.7) between SEM($\widehat{m}$) and npSEM algorithms shows that the two fitted dynamical models are close to each other. This is also illustrated on Figure. 4.6 where scatter plots of successive variables are shown for the three components of the L63 model. The scatter plots of the first column correspond to the state observed with the observational error. The ones of the middle column are realizations of the true state model. And the third column displays a simulation of the dynamical model at the end of the npSEM algorithm. These plots show that the npSEM algorithm efficiently filters the observational error because the right plot is much closer to the middle one than the left plot.

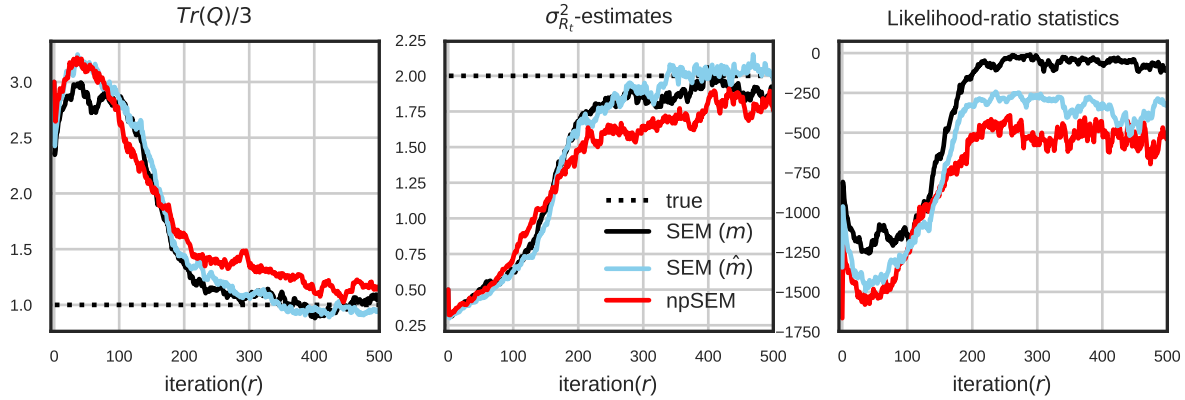


**Figure 4.5** – Comparison of the estimated parameters of SEM and npSEM algorithms on the L63 model (4.9). The left (resp. middle) panel shows the evolution of the trace of $Q$ (resp. $R_t$) estimates with respect to the iteration number of the EM algorithm. The right panel shows the evolution of the likelihood-ratio statistics (4.7).

Table 4.2 reports the RMSE mentioned in the introduction of this section. In the first column, the RMSE of smoothing is bigger than the one of forecasting. This may seem surprising but this can be explained by the presence of missing data. Table 4.2 shows that substituting $m$ by a non-parametric estimate such as LLR increases the forecasting error of about 7% and the smoothing error to less than 3% (comparing columns 1 [resp.2] and column 4 [resp.5]). We retrieve that smoothing, which incorporates the available observations to the forecast in the filtering procedure, is less sensitive to the prediction error linked to the non-parametric estimation. Furthermore, this substitution seems to have no impact on the estimation of the parameters $Q$ and $R_t$ since the errors of columns 2 and 5 are similar. Now, if the state is observed with observational errors and missing data, the RMSEs (last column) are increased of almost 20% for the forecasting and 10% for the smoothing compared to the case where the model is completely known (column 1). The increase of RMSE is composed of two different additional errors: the one due to the observational error and the one due to the presence of missing data which leads to a smaller learning dataset. However, to estimate the state time series (last column) we use no information of the model except the one contained in the observation time series. So we can conclude that the estimator performs reasonably well and clearly improves the naive estimator learned on the raw observations (column 3 of Table 4.2).

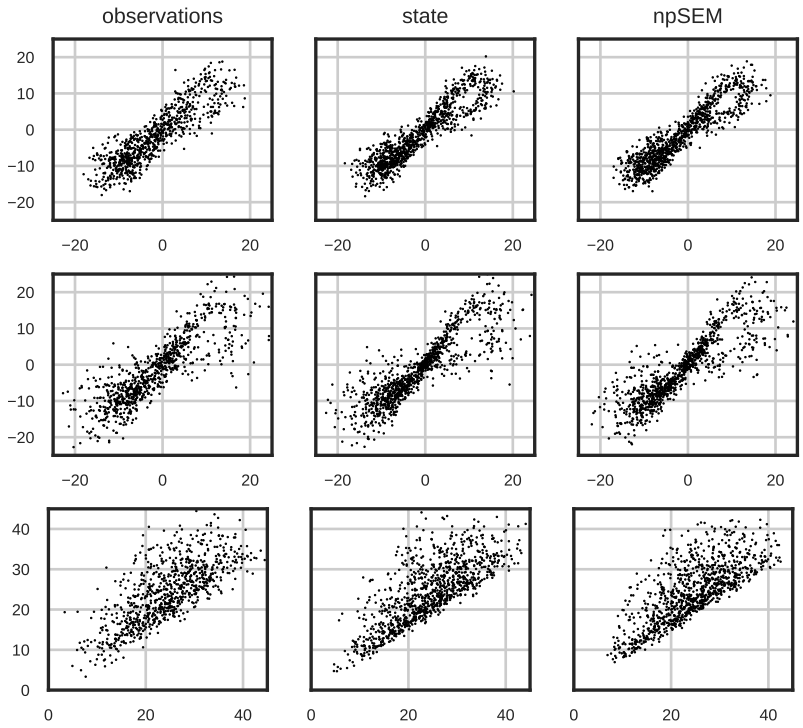


**Figure 4.6** – Scatter plots of $(Y_{t-1}, Y_t)$ (left), $(X_{t-1}, X_t)$ (middle) and $\left(\widetilde{X}_{t-1}, \widetilde{X}_t\right)$ (right) for the L63 model defined by (4.9). $\{\widetilde{X}_t\}$ stands for one of realizations generated at the final iteration of the npSEM algorithm.

## 4.5 Conclusions and perspectives

In this chapter, we introduce non-parametric approaches for SSMs. The proposed methodology permits to analyze time series with observational errors without specifying a dynamical model. We show, through numerical experiments on toy models, that it permits to successfully estimate the dynamical model and reconstruct the latent space from noisy observations.

The theoretical properties of the proposed algorithm need to be investigated. On the modeling aspect, we plan to relax the assumption of a constant covariance error $Q$ for the

**Table 4.2** – RMSEs (Eqs. 2.9 and 4.8) for forecasting and smoothing of a state sequence of the L63 model (4.9). The parameters are estimated on a sequence of length $T = 1000$. The smoothing algorithms are run with 10 particles. $\theta^*$ denotes the true values of the parameters. $X, Y$ and $\widetilde{X}$ represent to sequences generated from the true state process $\{X_t\}$, the observation process $\{Y_t\}$ and the npSEM algorithm, respectively.

| | $(m, \theta^*)$ | $(\hat{m}_X, \theta^*)$ | $(\hat{m}_Y, \theta^*)$ | $(m, \hat{\theta}_{SEM(m)})$ | $(\hat{m}_X, \hat{\theta}_{SEM(\hat{m})})$ | $(\hat{m}_{\widetilde{X}}, \hat{\theta}_{npSEM})$ |
|---|---|---|---|---|---|---|
| $RMSE(forecast)$ | 1.0013 | 1.0701 | 1.4201 | 1.0013 | 1.0701 | 1.2562 |
| $RMSE(smoothing)$ | 1.0210 | 1.0322 | 1.2227 | 1.0225 | 1.0492 | 1.1163 |

dynamical model and consider models where $Q$ varies in time to handle, for example, heteroscedastic time series. Non-parametric approaches could also be developed to estimate $Q$ in this context.

## Appendix

Note that in Algorithm 14, transition kernels have conditional means which are defined corresponding to model $\mathfrak{M}$ ($\mathfrak{M} = m$ for using the true evolution model (4.1), $\mathfrak{M} = \hat{m}$ or $\mathfrak{M} = \hat{m}_r$ for a LLR estimate of $m$).

**Algorithm 14: Smoothing with Conditional Particle Filter-Backward Simulation (CPF-BS)**

**Inputs**: conditioning trajectory $X^* = x^*_{0:T}$, observations $y_{1:T}$ and fixed parameter $\theta$.

**1**. Run **CPF** algorithm with the inputs given to obtain a system of $N_f$ particles and their weights $(x_t^{(i)}, w_t^{(i)})_{t=0:T}^{i=1:N_f}$.

- Initialization:
  + Sample $\{x_0^{(i)}\}_{i=1:N_f} \sim\ p_\theta(x_0)$ and set $x_0^{(N_f)} = x_0^*$.
  + Set initial weights $w_0^{(i)} = 1/N_f, \forall i = 1 : N_f$.
- For $t = 1 : T$,
  + Resample indices $\{I_t^i\}_{i=1:N_f}$ of potential particles with respect to the previous weights $(w_{t-1}^{(i)})_{i=1:N_f}$.
  + Propagate new particle

$$x_t^{(i)} \sim p_\theta\left(x_t | x_{t-1}^{(I_t^i)}\right), \forall i = 1 : N_f.$$

  + Replace for the conditioning particle, $x_t^{(N_f)} = x_t^*$ and $I_t^{N_f} = N_f$.
  + Compute the weight

$$w_t^{(i)} = \frac{p_\theta(y_t | x_t^{(i)})}{\sum_{i=1}^{N_f} p_\theta(y_t | x_t^{(i)})}, \forall i = 1 : N_f$$

  end for.

**2**. Repeat the following **BS** algorithm using the outputs of the **CPF** algorithm to gets $N_s$ trajectories $\{\tilde{x}^j_{0:T}\}_{j=1:N_s}$.

- For $t = T$, draw $\tilde{x}_T^j$ following the discrete distribution $p(\tilde{x}_T^j = x_T^{(i)}) = w_T^{(i)}$.
- For $t < T$,
  + Calculate smoothing weights

$$\tilde{w}_t^{(i)} = \frac{p_\theta(\tilde{x}_{t+1}^j | x_t^{(i)})\ w_t^{(i)}}{\sum_{j=1}^{N_f} p_\theta(\tilde{x}_{t+1}^j | x_t^{(i)})\ w_t^{(i)}},\ \forall i = 1 : N_f.$$

  + Draw $\tilde{x}_t^j$ with respect to $p(\tilde{x}_t^j = x_t^{(i)}) = \tilde{w}_t^{(i)}$.
  end for

**3**. Update the new conditioning trajectory $X^*$ by sampling uniformly from $N_s$ trajectories.

**Outputs**: realizations describing the smoothing distribution $p_\theta(x_{0:T}|y_{1:T})$.

Chapter 5

# Applications of non-parametric algorithms

This chapter presents two applications of the non-parametric algorithms. In Section 5.1, we aim at using a non-parametric filtering algorithm for model selection/ model comparison given a set of observations and existing model runs. The performance of the proposed approach is compared to the one of the classical approach on L63 models with different forcing parameterizations. This section belongs to a part of ECOS-SUD project in collaboration between France and Argentina (2018 – 2020). The second application is then introduced in Section 5.2. Here we propose to apply the npSEM algorithm to impute noisy missing data in reality. Wind data shown in Section 1.2 (produced by Météo France) is reconsidered. Imputation results of the npSEM algorithm on the data are compared to the ones of regular regression methods.

## 5.1 Model selection and model comparison using a non-parametric filtering algorithm

### 5.1.1 General context

Model selection or model comparison [21, 131] generally aims at determining or detecting one model in a set of different models which well describes a sequence of observations. Some of applications can be found in [72, 73] for climate attribution and detection, [52, 53] for Bayesian model selection of subsurface flow models and [27] for studies of the discrimination between phenomenological models of the glacial-interglacial cycle. Generally, models in these applications can have different configurations (parameterization choices in physical systems, forcing terms, model error noises, etc) or different formulations. For instance, different L63 models, where $m$

is defined in Eq. (5.1) for $x \in \mathbb{R}^3$, $z_\tau \in \mathbb{R}^3$, $\lambda \in [-8, 8]$ and $\tau \in [0, dt]$,

$$\begin{cases} z_0 = x, \\ \frac{dz_\tau}{d\tau} = \lambda F + g(z_\tau), \quad \text{as } F = \left[\cos\frac{7\pi}{9}, \sin\frac{7\pi}{9}, 0\right]^\top, \\ m(x) = z_{dt}, \end{cases} \tag{5.1}$$

can be constructed with different values of forcing parameter $\lambda$. When $\lambda = 0$, the system of ODEs (5.1) is the classical one (1.7) which represents a physical model in the classical world. In the case that $\lambda \neq 0$, the L63 model including a forcing term represents a modified model in the climate change world. On Figure 5.1, we show two trajectories (curves) derived from two L63 dynamical models with respect to $\lambda = \lambda_0 = 0$ (correct) and $\lambda = \lambda_1 = 8$ (incorrect), and one set of observations (points) derived from a L63 SSM associated to the correct model. According to the figure, the state is located on the right wing rather than the left due to the fact that the forcing term with the incorrect parameter $\lambda_1 = 8$ shifts the L63 trajectory to the direction of 140°.

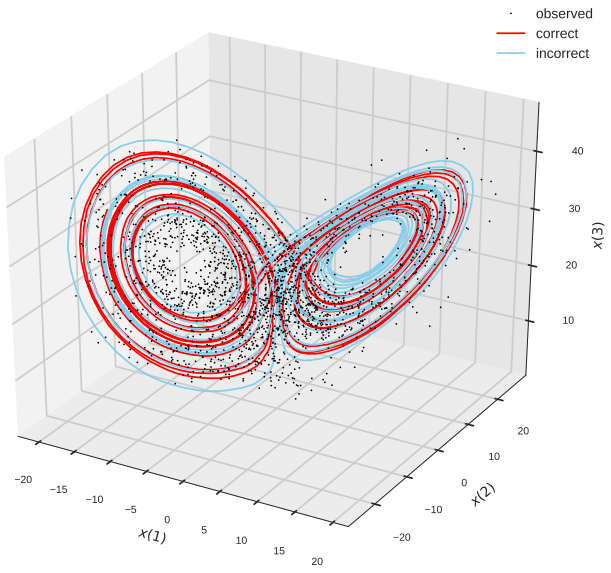


**Figure 5.1** – Simulated trajectories derived from the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$, $dt = 0.01, Q = 0.001I_3, R = 2I_3$. Correct state and observation sequences are generated from the correct model with forcing parameter $\lambda_0 = 0$, and incorrect state sequence is generated with the incorrect model with forcing parameter $\lambda_1 = 8$.

In the next section, we introduce methods based on model evidence for detecting the best model associated to a given sequence of observations.

### 5.1.2 Methods

Let $\{\mathcal{M}_i\}_{i=1:L}$ denote a finite set of different models . Given a sequence of observations $y_{1:T}$, the classical model selection or model comparison is carried out based on computing model evidence (ME) estimates (see e.g. in [5, 27]), $\ln p\left(y_{1:T}|\mathcal{M}_i\right)$ which is the global log-likelihood function of the observations $y_{1:T}$ conditional on the model $\mathcal{M}_i$ (see Eq. 2.11 for its interpretation). If $\mathcal{M}_i = m$, one obtains the exact ME. Although the standard ME approach based on global log-likelihood computation is widely applied in model selection and model comparison, it may not enable to study properties of the model at specific positions such as boundaries, stationary points or extreme values.

Alternatively, [25] proposed to compute a local log-likelihood function on every time interval $[t+1:t+K] \subseteq [1,T]$, named as contextual model evidence (CME). For $i = 1:L$ and $t = 1:T-K$, CME is defined by

$$\begin{aligned} l_i(t,K) &= \ln p\left(y_{t+1:t+K}|y_{1:t};\mathcal{M}_i\right) = \sum_{s=t+1}^{t+K} \ln p\left(y_s|y_{1:s-1};\mathcal{M}_i\right) \\ &= \sum_{s=t+1}^{t+K} \ln \int p\left(y_s|x_s\right)\ p\left(x_s|y_{1:s-1};\mathcal{M}_i\right) dx_s, \end{aligned} \tag{5.2}$$

where $K$ is the length of evidencing window. In [25], one classical filtering algorithm (e.g. EnKF) is run to

- generate $N$-sample $\{x_s^j\}_{j=1:N} \sim p(x_s|y_{1:s-1};\mathcal{M}_i)$ (forecast distribution according to each model $\mathcal{M}_i$),
- compute the CME estimate of $l_i(t,K)$ (5.2) whose integral is approximated by mean of all marginal log-likelihood $\{p\left(y_s|x_s^{(j)}\right)\}_{j=1:N}$ given the forecast sample (see Chapter 2 [Section 2.3.2] for details).

The authors showed that this approach allows to compare the likelihood of different model candidates and then assign the most appropriate model candidate describing a given observed sequence, and compared model evidence performances of the classical EnKF algorithm and its variants.

In this section, we consider the situations where several datasets derived from model simulation exist. State-of-the-art model-run datasets are available through CMIP5 project `https://cmip.llnl.gov/cmip5/`. Given a set of observations, the objectives consist in developing low-cost

methods for intercomparison of the datasets and analyzing some of the factors (e.g. error noises) involved in model identification quality. In such cases, running the numerical forecast models is not useful due to wasting the available datasets and computational resources for integration of systems of ODEs. Here we propose to use a non-parametric EnKF algorithm (so-called analog EnKF (AnEnKF) algorithm in [95, 153, 155]) for computing CME estimates. It is the classical EnKF in combination with LLR estimates of the models learned on the corresponding datasets presented in Algorithm 5. Note that other non-parametric filtering algorithms can be used depending on different scenarios of models (e.g. different nonlinearity level, Gaussian or non-Gaussian assumption) and computational resources (see the discussion in Chapter 2 [Section 2.4]). Here, we focus on validating the non-parametric approach in model evidence estimation and comparing performances of the classical and the proposed approaches in model selection and model comparison. Criteria for assessing these methods consist of

- estimates of CME $l_i(t, K)$ (5.2): the larger CME estimates the better model describing observations in the evidencing window $[t+1, t+K]$,

- estimates of average CME (5.3): the larger average CME estimate the better model describing a given sequence of observations,

$$\bar{l}_i(K) = \frac{\sum_{t=1}^{T-K} l_i(t, K)}{T-K}, \quad \forall i = 0 : L, \tag{5.3}$$

- estimates of pair-wise CME difference (5.4) (log-likelihood ratio): if $D_{i,j}$ is positive $\mathcal{M}_i$ better matches observations on the evidencing window $[t+1, t+K]$ than $\mathcal{M}_j$, and vice versa,

$$D_{i,j}(t, K) = l_i(t, K) - l_j(t, K), \quad \forall i = 0 : L, j = 0 : L, \tag{5.4}$$

- percentage of $\mathcal{M}_i$-identification (5.5): if $p_{0,i}(K) > 50\%$ [resp. $p_{0,i}(K) < 50\%$] one obtains more [resp. less] evidence on the model $\mathcal{M}_i$, and if $p_{0,i}(K) = 50\%$ one needs more conditions (e.g. larger $K$) to select an appropriate model for the given observed sequence,

$$p_{0,i}(K) = \frac{\sum_{t=1}^{T-K} \mathbf{1}\left(D_{0,i}(t, K) \geq 0\right)}{T-K} 100\%, \quad \forall i = 0 : L. \tag{5.5}$$

In the next section, results of model selection and model comparison on L63 models are presented using the above criteria.

### 5.1.3 Results

Let us consider a L63 SSM (Eq. 4.9) where the dynamical model $m$ is defined as in Eq. (5.1) with $dt = 0.1, \lambda = 0$ (see in [25]) and the observation transformation operator is $H_t = I_3$. For each experiments, a sequence of $10^3$ observations (see top panel of Figure 5.2 for a time series plot of the state and observations derived from the model) and $L$ learning data with length $T$ derived from $L$ models $\{\mathcal{M}_i\}_{i=1:L}$ with different values of forcing parameters $\{\lambda_i\}_{i=0:L}$ are given. The EnKF algorithm using numerical model $\{\mathcal{M}_i\}_{i=0:L}$ and the AnEnKF algorithm (Algorithm 5) using LLR estimates (1.26) of the models learned on the given datasets are run with 100 members in order to compute estimates of CME (5.2).

In the first experiment two L63 models $\{\mathcal{M}_i\}_{i=0:1}$ with $\lambda_0 = 0$ (correct) and $\lambda_1 = 8$ (incorrect) are considered. The correct [resp. incorrect] learning data with length $T = 10^4$ and the observed sequence with length $10^3$ are simulated from the correct [resp. incorrect] model. CME (local log-likelihood) of these two models is estimated with evidencing window size $K = 1$. On the bottom panel of Figure 5.2, time series of the CME estimates computed by the classical and the non-parametric EnKF algorithms are plotted. As expected, the CME estimates in the incorrect model are almost smaller than in the correct one for both classical and non-parametric methods. Low peaks more frequently occur in CME time series of the incorrect model. These peaks seem to correspond to sensitive positions (e.g. bifurcations at time steps in $[475, 480]$ and $[575, 600]$) of the L63 trajectory where the forcing term of the incorrect model easily changes the direction of the state trajectory. Sensitivity of state position to CME values is shown later.

With the same experiment as the previous one, a comparison of the classical and non-parametric algorithms in computing CME difference $D_{0,1}(t, 1)$ (Eq. 5.4) as a function of state position is displayed on Figure 5.3. Here the CME difference values are shown with respect to the first and third components. Obviously, the classical and non-parametric approaches give similar estimates of the CME difference values. A number of cells with positive $D_{0,1}(t, 1)$ values seems to be larger than the one with negative values, the correct model can hence be identified even with only one observation ($K = 1$) in this experiment. Correct model identification seems to be facilitated at lower bounds of L63 attractors which are the sensitive positions discussed in the

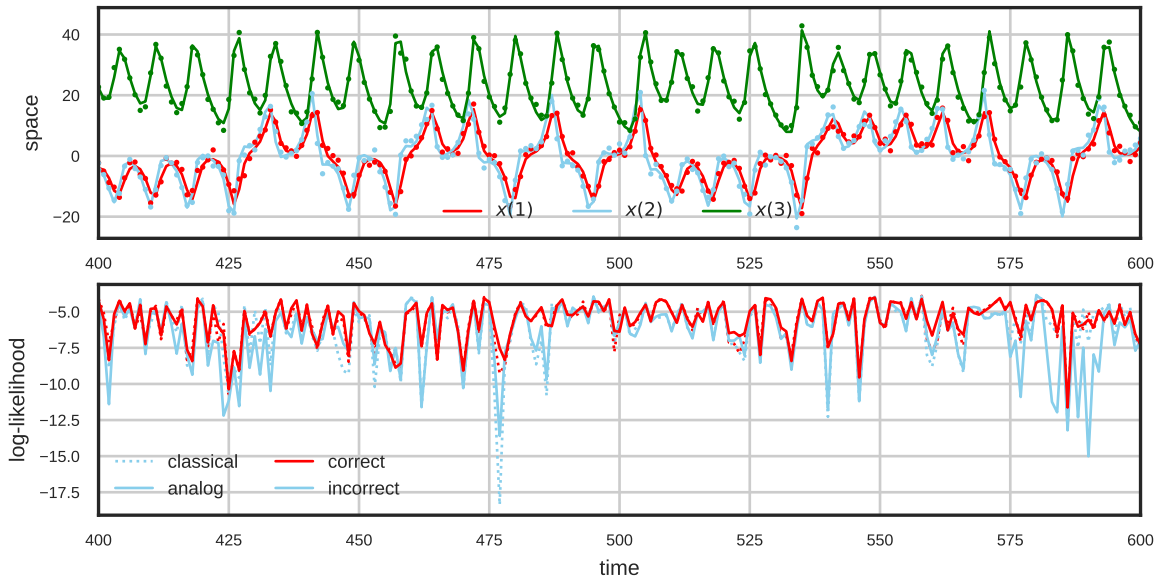


**Figure 5.2** – Top: time series plot of a segment of the state and observed sequences simulated from the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$, $dt = 0.1, \lambda = 0, Q = 0.001 I_3, R = 2I_3$. Bottom: time series plot of CME estimates of $l_i(t, 1)$ (Eq. 5.2) derived from the classical and the non-parametric (analog) algorithms for both correct model ($\lambda_0 = 0$) and incorrect model ($\lambda_1 = 8$).

previous experiment. In summary, this experiment based on computing the local CME estimates or local CME difference ($K = 1$) allows to survey the impacts of model evidence on each of state positions, especially on fixed points or extreme points of the models. Moreover, these scores can be the promising metrics in order to compare the skill of different model candidates and select the better model capturing each of the observations.

For the second experiment, the sensitivity of model identification to the amount of learning data used in the non-parametric approach is explored. Different learning sequences with length $T \in [10^2, 5 \times 10^4]$ are simulated from the correct and incorrect L63 models. 10 repetitions of the AnEnKF algorithm, where the non-parametric emulator is estimated based on each of the given datasets, are carried out. Means and 95% CIs of percentage of the correct identification $p_{0,1}(1)$ (Eq. 5.5) computed by the algorithms are presented in Table 5.1. Results obtained by the classical algorithm are also shown in order to compare the ones obtained by the non-parametric algorithms. In this table, the percentage values derived from the non-parametric algorithms increase when $T$ increases as expected (see Figures 2.1 and 2.3 and the involved comments for an explanation). For $T \geq 5 \times 10^3$, the non-parametric approach seems to stabilize and gives slightly higher confidence in identification of the correct model than the classical approach. This may occur in circumstances of deterministic [resp. approximately deterministic] dynamical models

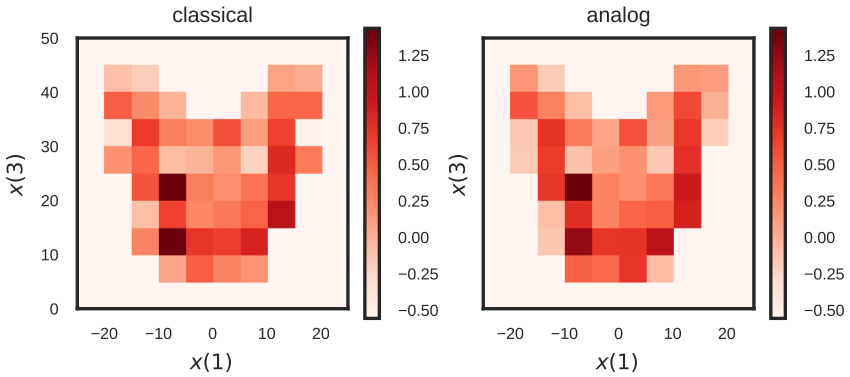


**Figure 5.3** – Comparison of the classical and non-parametric algorithms in computing CME difference $D_{0,1}(t,1)$ (Eq. 5.4) with respect to state position (the illustration is for the first and third components) on the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$, $dt = 0.1, \lambda = 0, Q = 0.001 I_3, R = 2 I_3$. Two models are considered with correct $\lambda_0 = 0$ and incorrect $\lambda_1 = 8$.

with null [resp. insignificant] error covariance (here $Q = 0.001 I_3$), the most usual scenarios considered in the classical DA applications. Non-parametric estimates of these models in the non-parametric filtering algorithms probably produce larger forecast variance, widen the support of the forecast samples and enable to better cover the observations. The effects of error noises on model comparison or selection are also studied in the next experiments.

**Table 5.1** – Sensitivity of the model identification with respect to length of learning data ($T$) used to estimate the dynamical model $m$ in the non-parametric algorithms on the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$, $dt = 0.1, \lambda = 0$, $Q = 0.001 I_3$ and $R = 2 I_3$. Correct [resp. incorrect] learning data with length $T \in [10^2 - 10^5]$ and the observed sequence with length $T' = 10^3$ are simulated from the correct [resp. incorrect] model with $\lambda_0 = 0$ [resp. $\lambda_1 = 8$]. Means and 95% confidence intervals (CI) of the correct model identification percentage $p_{0,1}(1)$ (Eq. 5.5) are computed for each of the algorithms using 10 repetitions.

| Approaches | | $T = 10^2$ | $T = 5 \times 10^2$ | $T = 10^3$ | $T = 5 \times 10^3$ | $T = 10^4$ | $T = 5 \times 10^4$ |
|---|---|---|---|---|---|---|---|
| **Classical** | Mean (%) | 64.43 | | | | | |
| | CIs (%) | [63.17, 65.79] | | | | | |
| **Analog** | Mean (%) | 54.94 | 61.76 | 63.98 | 66.88 | 67.19 | **67.17** |
| | CIs (%) | [53.59, 56.41] | [60.46, 62.66] | [62.92, 64.7] | [66.37, 68.0] | [65.89, 68.88] | [**66.49**, **67.82**] |

In this experiment, one $10^3$-observed sequence and two different $10^4$-learning sequences (with correct $\lambda_0 = 0$ and incorrect $\lambda_1 = 8$) are generated from each of L63 models with error noise covariances $Q$ and $R$ varying. Means and 95% CIs of percentage of the correct identification $p_{0,1}(1)$ (Eq. 5.5) computed by the classical and the proposed algorithms are presented in Table 5.2. Generally, the correct model identification is more efficient when model error and observational error are small. Compared to the scores of the classical approaches, the ones of the non-parametric approaches are 3% better when model error covariance $Q$ is

insignificant ($Q = 0.001I_3$). For other cases the classical approach better performs the model identification because the forecast phase includes model noises and there is no effect on biased estimates. However, it consumes the computational cost due to multiple runs of numerical forecast models as mentioned. In situations where observational error covariance $R$ is not large, the observations close to the state and CME estimates (likelihood) strongly depend on the observations holding behaviors of the correct model. As a result, the correct model is identified easily as the discrepancy among the two given models is significant ($\lambda_0 = 0$ and $\lambda_1 = 8$). The percentage almost reaches to 100% when $Q = 0.001I_3$ and $R = 0.01I_3$ even with evidencing window size $K = 1$.

**Table 5.2** – Sensitivity of the model identification to error noise covariances $(Q, R)$ of the classical and the analog EnKF algorithms on the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$ and $dt = 0.1$. The correct [resp. incorrect] learning data with length $T = 10^4$ and the observed sequence with length $T' = 10^3$ are simulated from the correct [resp. incorrect] model with $\lambda_0 = 0$ [resp. $\lambda_1 = 8$]. $Q$ [resp. $R$] is fixed to $0.001I_3$ [resp. $2I_3$] if the value of $R$ [resp. $Q$] varies. Means and 95% confidence intervals (CI) of the correct model identification percentage $p_{0,1}(1)$ (Eq. 5.5) are computed for each of the algorithms using 10 repetitions.

| Approaches | | $Q = 0.001I_3$ | $Q = 0.01I_3$ | $Q = 0.1I_3$ | $Q = I_3$ | $Q = 10I_3$ |
|---|---|---|---|---|---|---|
| **Classical** | Mean (%) | 64.43 | 63.12 | 61.19 | 57.14 | 50.65 |
| | CIs (%) | [63.17, 65.79] | [61.81, 63.84] | [59.52, 62.44] | [55.26, 59.6] | [49.11, 51.61] |
| **Analog** | Mean (%) | **67.19** | 62.64 | 59.81 | 55.54 | 50.76 |
| | CIs (%) | [**65.89**, **68.88**] | [61.88, 63.52] | [58.79, 60.95] | [53.98, 57.68] | [48.65, 52.86] |

| Approaches | | $R = 0.01I_3$ | $R = 0.1I_3$ | $R = 2I_3$ | $R = 10I_3$ | $R = 20I_3$ |
|---|---|---|---|---|---|---|
| **Classical** | Mean (%) | 99.69 | 88.96 | 64.43 | 57.39 | 54.67 |
| | CIs (%) | [99.6, 99.78] | [88.15, 89.44] | [63.17, 65.79] | [56.28, 58.69] | [51.77, 56.44] |
| **Analog** | Mean (%) | **99.99** | 89.94 | 67.19 | 57.78 | 54.42 |
| | CIs (%) | [**99.92**, **100.0**] | [89.51, 90.42] | [65.89, 68.88] | [56.75, 58.75] | [52.65, 56.71] |

On Figure 5.4, we illustrate a strategy to select one good model among several models with $\lambda_i \in [-8, 8]$. A $10^3$-sequence of observations derived from the L63 model with correct parameter $\lambda_0 = 0$ is given. Several $10^4$-learning datasets are simulated with respect to $\lambda_i$. The classical and the proposed algorithms are run to compute estimates of the average CME ($\bar{l}_i(1)$, Eq. 5.3) for each value of the forcing parameter. According to the figure, the highest average CME estimates correspond to models close to the correct model. The larger absolute values of the difference between $\lambda_0$ and $\lambda_i$, the lower model evidence. The non-parametric approach gives similar average CME pattern than the one of the classical approach but the average CME function is 0.1 higher because of forecast noise production (analyzed as in the experiments above). Through this

experiment, we can deduce that computing the average CME estimates permits to return good model indicated to the whole sequence of observations.

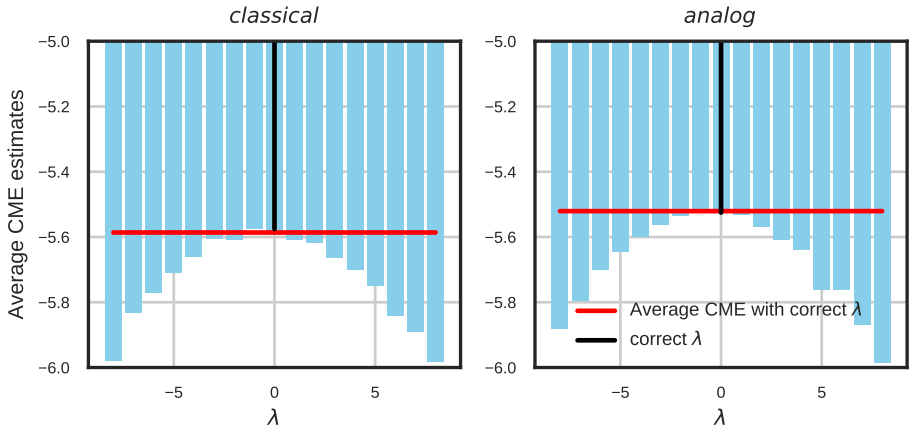


**Figure 5.4** – Comparison of average CME estimates ($\bar{l}_i(1)$, Eq. 5.3) of the classical and non-parametric approaches on the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$, $dt = 0.1$, $Q = 0.001 I_3$ and $R = 2I_3$, $\lambda_0 = 0$ (correct) and $\{\lambda_i\}_i \in [-8, 8]$.

Finally, the sensitivity of model identification of the non-parametric filtering algorithm to the discrepancy of learning datasets (different values of $\lambda$) and the evidencing window size ($K$) is illustrated on Figure 5.5. Percentage of correct model identification ($p_{0,i}(K)$, Eq. 5.5) is computed with respect to each of the given datasets and evidencing window length. With extremely small $K$ and non-large discrepancy of the learning dataset, the percentage is approximately 50% which says no confidence in comparing model description skill of these datasets and there is the need of more observations for model identification. When $K$ increases the percentage of correct identification increases. Moreover, the scores tend to 100% if $|\lambda_0 - \lambda_i| \geq 6$ and $K = 100$ (only 100 observations are used to estimate CME).

## 5.2 Data imputation using non-parametric stochastic Expectation-Maximization algorithm. An application to wind data

### 5.2.1 General context

Data imputation is the recovery process of missing values existing in data. Its applications can be found in numerous fields (see [18, 59, 82, 104, 140, 146, 161] for a few). Particularly in meteorology, data recorded from in-situ sensors are usually missing because of failures of recording devices, complex interaction or accidental variation of meteorological variables (e.g.

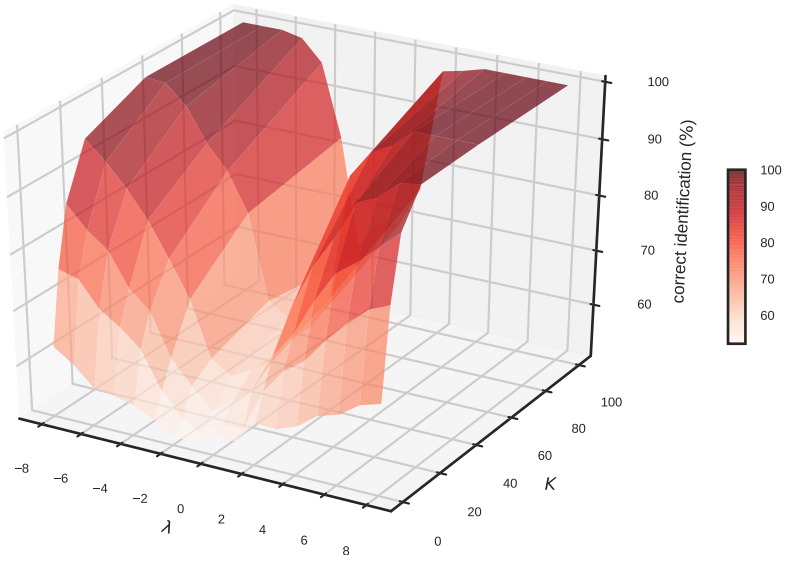


**Figure 5.5** – Sensitivity of the model identification ($p_{0,i}$, Eq. 5.5) of the non-parametric filtering algorithm with respect to values of forcing parameter $\lambda$ ($\lambda_i \in [-8, 8]$)) and window size ($K \in [1, 100]$) on the L63 SSM (Eq. 4.9) with $m$ defined in Eq. (5.1), $H_t = I_3$, $dt = 0.1, \lambda = 0$, $Q = 0.001 I_3$ and $R = 2 I_3$.

extreme values), etc. An example of data with missing values is the wind data introduced in Chapter 1 [Section 1.2] (see Figure 1.6 for an illustration). There exist different situations of gaps in this data consisting of long [resp. short] gaps at one station (e.g. Ploudalmezeau [resp. Brest-Guipavas]), simultaneous gaps at several stations (e.g. Brignogan and Ploudalmezeau), etc. Such missing data may provide incomplete information of variables, increase bias in statistical inference and reduce the accuracy of conclusions in data analysis. Therefore, developing an efficient tool for reconstructing gaps becomes one of the important tasks to handle such problems.

### 5.2.2 Methods

There are numerous methods for missing-data imputation (see in [65, 84, 105, 125, 140, 166, 175] for instance). One method is regression which has been used in imputing meteorological data. Missing value of one station at a current time ($t$) can be computed based on complete data of the others at time $(t-1, t, t+1)$ since they are closely correlated. A general regression model is defined below.

$$Y_t^i = \tilde{m}\left(Y_{t-1}, \{Y_t^j\}_{j \neq i}, Y_{t+1}\right) + \tilde{\epsilon}_t^i, \quad i, j = 1 : d_{Y_t} \tag{5.6}$$

where $\tilde{m}$ denotes a regressive function of variables and $\{\tilde{\epsilon}_t^i\}_i$ denote Gaussian error noises. In Eq. (5.6), $i$ indicates the location of one missing component (corresponding to data at one

station) of the observation $Y_t$ and $j$ indicates its non-missing component. Given the model and an observed data $y_{1:T}$, different (ridge, logistic or support vector) regression approaches can be used to estimate missing values. However, neither regression nor other imputation approaches (e.g.interpolation, multiple imputations) [105] can well treat the observational errors derived from modeling imperfection and errors of measurement process or instrumentals. This problem is so-called errors-in-variables [26,94,111] leading to substantial bias in reconstruction of missing data.

Following the numerical results and their analysis presented in Chapter 4, we propose to apply the npSEM algorithm (Algorithm 13) to impute gaps and reduce observational errors in the wind data. Here an SSM (e.g. Eq. 1.22) formulating a state dynamic and a transformation between the state and incomplete observations is considered instead of a regression model (e.g. Eq. 5.6). In the SSM, the dynamical model $m$ of the state is unspecified and the observational transformation is set by $H_t = I_{d_{Y_t} \times d}$ ($I_{d_{Y_t} \times d}$ is a modified identity matrix). The model error covariance is assumed to be time-invariant and has full form while the observational error covariance is assumed to be adaptive in time, $R_t = \sigma^2_{R_t} I_{d_{Y_t}}$ ($\sigma^2_{R_t}$ is a scalar value in $\mathbb{R}^+$). Given the observations $y_{1:T}$, the objectives consist in reconstructing the dynamical model $m$, estimating the model error covariance and observational error covariance scalar $(Q, \sigma^2_{R_t})$ and computing the smoothed estimates of missing values. Results of the proposed algorithm on the wind data are presented in the next section.

### 5.2.3 Results

Let us take an extract of wind data in January of 2010 recorded at five stations of North-West of Brittany, France (see location map on the left panel of Figure 1.6) as an observed data ($Y_{1:T}$) with length $T = 744$. 10% artificial gaps are created from the data for validation. Note that the observed data takes into account both wind speed ($U_{1:T}$) and wind direction ($\Phi_{1:T}$) by the transformation (5.7) of the polar coordinates $(U_t, \Phi_t)_t$. Such a combination of these two variables is often considered in meteorology because data at each station is now redefined on $\mathbb{R}^2$ that is, therefore, easier to deal with than the couple $(U_t, \Phi_t)$ defined on $\mathbb{R}^+ \times [0, 2\pi]$ (see in [1,109,127] and references therein).

$$Y_t = [U_t \cos(\Phi_t), U_t \sin(\Phi_t)], \quad \forall t = 1 : T \tag{5.7}$$

where $Y_t \in \mathbb{R}^{d_{Y_t}}$.

Given the data with artificial gaps, the npSEM algorithm is run on a SSM model (1.22) including a dynamical model of the state process $\{X_t\}_t$ in $\mathbb{R}^d$ with dimension $d = 10$ and an observation model of the observation process $\{Y_t\}_t$ in $\mathbb{R}^{d_{Y_t}}$ with adaptive dimension $d_{Y_t} \leq d$ ($\forall t = 1 : T$). Error noise covariances are initially set equal to $Q_0 = 4I_d$ and $R_{t,0} = 2I_{d_{Y_t}}$. The first learning data is the observed data. Given the first learning data and the observations, a non-parametric EnKS algorithm (Algorithm 5 in conjunction with a backward smoothing algorithm based on RTS scheme (1.12), see in [58]) is run with 20 members, then the mean of samples derived from the algorithm is set for the first conditioning trajectory $X_0^*$. Number of particles and number of realizations are $N_f = 10$ and $N_s = 5$ respectively. The npSEM algorithm is run until 100 iterations. To compare the performance of the npSEM algorithm, the regular (ridge) regression methods (LR, LLR) based on the regression model (5.6) are considered. Here RMSEs are computed between the true values of the artificial gaps and the corresponding estimates derived from each method.

In Figure 5.6, performances of parameter estimation, error noise reduction and state reconstruction are shown. Through the first row of the figure, one can see that the npSEM algorithm allows to estimate the full model covariance matrix. Variance estimates of the model error noise (shown on diagonal of each panel) are significantly reduced. The highest estimate value of the variance is at wind state of Brignonan station (for both directions denoted as locations 1 and 6 on the panels). This may be due to the position of this station close to the coast leading to the strong volatility of wind speed (as shown in the time series plot of Figure 1.6), and the existence of lots of gaps. The algorithm also enables to reproduce the data correlation between the stations. The covariance entries admit different values instead of zeros as in the first setting. This may also be helpful for inference of the state of systems with correlated variables. Observational scalar parameter estimates are also computed. The estimate function (bottom left panel) seems to stabilize at iteration 100.

The RMSEs between the true values of the artificial gaps and the mean of smoothed samples derived from the npSEM algorithm are displayed on the bottom right panel of Figure 5.6. The score is chosen since the information of the true state is unavailable in real applications. As shown, the RMSE function decreases as expected. In Table 5.3, the imputation performance of the npSEM algorithm is also compared to the one of the regression methods. Both standard and ridge LR/ LLR are considered based on the regression model (5.6). Regarding the table, the standard regression methods completely fail due to ill-condition (lots of gaps but a short

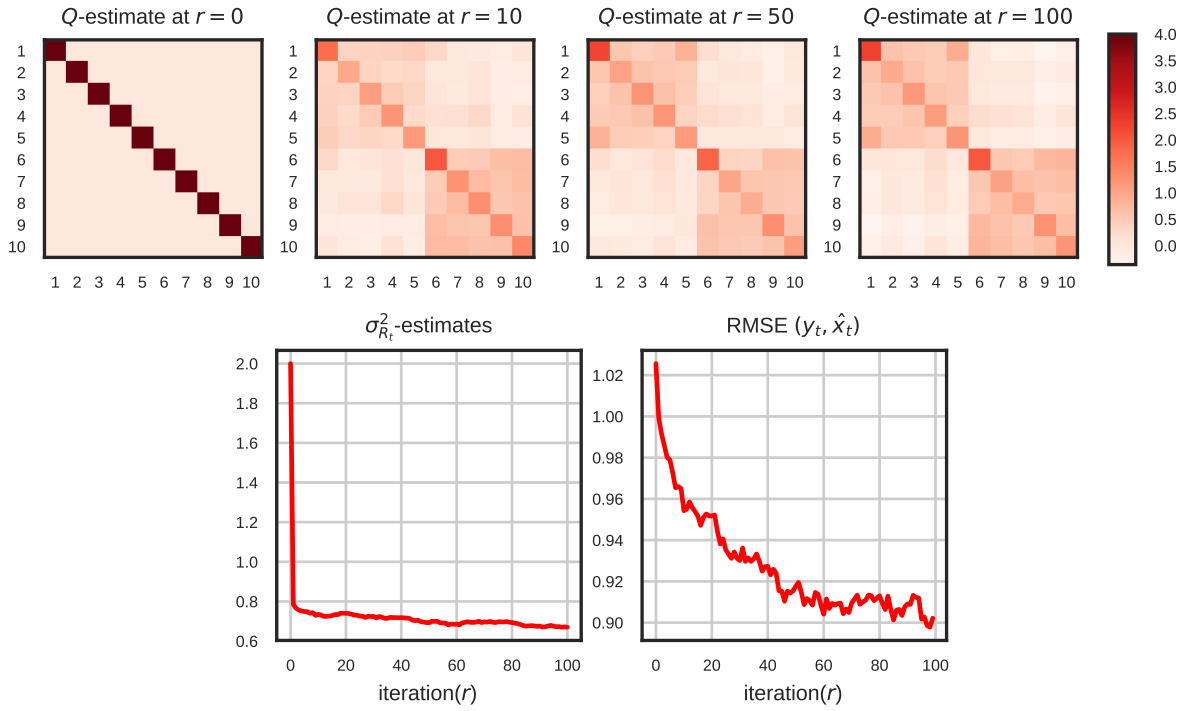


**Figure 5.6** – Parameter estimation and state reconstruction of npSEM algorithm on wind data (see Figure 1.6 for its time series plot). RMSE (Eq. 4.8) is computed between the observation $y_t$ and smoothed mean $\hat{x}_t$ derived from the algorithm.

sequence of the data). Ridge regression decreases much more errors. Nevertheless, the npSEM algorithm improves the ridge regression methods. Its RMSE is approximately 1.5 and 2 less proportional than the ones of ridge LLR and ridge LR.

**Table 5.3** – RMSEs between true values of artificial gaps and imputed data derived from different imputation methods based on regression model and state-space model. For npSEM algorithm, imputed data are the means of smoothed samples of last 10 iterations.

| Models | **Regression** (Eq. 5.6) | | | | **SSM** (Eq. 1.22) |
|---|---|---|---|---|---|
| Methods | **LR** | **Ridge LR** | **LLR** | **Ridge LLR** | **npSEM** |
| **RMSE** | 7.512 | 1.7127 | 6.6385 | 1.4286 | **0.9065** |

Abilities of the npSEM algorithm in model reconstruction and error noise reduction of the wind data are illustrated on Figure 5.7. The relations between two successive variables in the original observed data (first row) and in the corrected data (last row) derived from the npSEM algorithm are compared. For all stations, the corrected data have smaller variances than the observed data, especially in the tails. Here the proposed algorithm can reduce the observational errors in the data or at least calibrate ratios of the model errors and the observational errors

such that these stochastic terms enable to compensate uncertainties in model approximation and inferring tasks (filtering, smoothing and estimating). Using the npSEM algorithm may permit to better reconstruct a dynamical model for the wind data than using standard regression methods learned on this raw data directly.

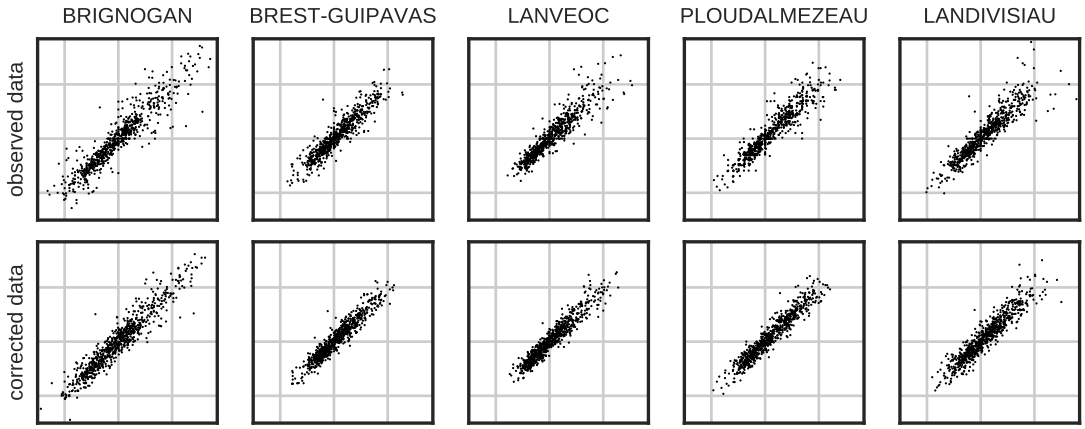


**Figure 5.7** – Scatter plot of two successive variables in the observed data and the corrected data derived from npSEM algorithm. The results are shown for the data recorded corresponding to West-East direction.

## 5.3 Conclusions and Perspectives

In Section 5.1, we show that the non-parametric filtering algorithm permits to compute model evidence given a sequence of observations and different model-run sequences of the state. We propose different metrics to compare these datasets and select the best model-run data describing the observations. Compared to the performance of the classical filtering algorithm, the proposed algorithm avoids rerunning numerical forecast models which are expensive in DA. In the future works, we would like to apply the non-parametric filtering algorithm on model evidence on modified models such as models including smooth forcing functions and model covariance with varying values, and high dimensional models but only few variables in the models considered. Climate change attribution and detection based on estimating model evidence with non-parametric filtering algorithms will be also investigated.

In Section 5.2, we propose to use the npSEM algorithm for missing-noisy data imputation on wind data produced by Météo France. Several illustrations of the performances of the algorithm in terms of parameter estimation, model reconstruction, data imputation, and error reduction are displayed. Here the wind data seems to follow approximately a linear model so that a Kalman

algorithm can be run to get similar results but with a low cost. Therefore the future works include detecting data, which derived from nonlinear dynamics, for exhaustive applications of the npSEM algorithm.

Chapter 6

# Conclusions and Perspectives

## 6.1 Conclusions

Statistical inference tools including standard particle-based filters/smoothers, and SEM machines are used to infer the state of nonlinear systems and relevant parameters from noisy observed data. However, the regular tools using a huge amount of particles require expensive computations in practice. As one of the contributions of the thesis, we detected and then illustrated the efficiency of CPFs/CPSs, and of their combinations with SEM algorithms in state and parameter estimation with a low computational cost.

In the classical approaches, forecast models are derived from an explicit dynamical model or its numerical approximations. For real applications in geosciences, numerical forecast models have to be run for each time step. That usually leads to the need of high computational resources, for instance when the time increment between two successive state variables in the evolution model is large or only several components of the system are of interest but the whole model must be run. In the thesis, we proposed novel non-parametric methods where the dynamical model is replaced by a non-parametric estimate, learned on a historical dataset recorded from satellites, in-situ sensors or numerical simulations of physical systems.

As the first step, we explored different combinations of non-parametric emulators (LC and LLR) and filtering schemes (EKF, EnKF, and PF), leading to non-parametric filtering algorithms. Through numerical experiments, we found that results derived from the non-parametric filtering algorithms converge to those of the classical filtering algorithms if the learning data is informative enough.

The breakthroughs of this thesis consist in developing the non-parametric SEM algorithms. These algorithms were proposed to reconstruct unspecified dynamical models, estimate unknown parameters, and infer the hidden state for each of the two following conditions:

1. Perfect observations: a learning dataset simulated from the state process with no observation error is assumed to be available.

2. Imperfect observations: only the noisy data taking into account observational errors is given.

In the first situation, we combined the non-parametric emulator, learned on the perfect data, with the CPS smoother and the original SEM algorithm. However, such "perfect" observations of the state, with no observational error, are typically not available. In the second situation, only a sequence of the process with observational errors is available. This increases estimation errors if a non-parametric estimate is learned directly on this noisy data. To handle this problem, the thesis introduced a novel non-parametric algorithm which combines the non-parametric emulator, the low-cost CPS smoother with an SEM-like algorithm wherein the smoothed samples generated from the algorithm in the current iteration are set as learning data for the latter iteration.

Finally, the potential abilities of the proposed approaches in terms of model selection/model comparison noise error reduction, missing-data imputation and parameter estimation were illustrated on toy examples and wind data produced by Météo France.

## 6.2 Perspectives

The thesis closes at this stage but opens lots of interesting subjects for future research in statistics and various applications. Some of them are discussed as follows.

First of all, we plan to apply the proposed non-parametric algorithms consisting of non-parametric filtering (smoothing) algorithms and non-parametric SEM algorithms to small dimensional DA problems (e.g. tracking and system control) or local/regional applications in geosciences (e.g. missing-data imputation, model selection, and climate change detection). An example is the case study of climate change phenomena in Europe instead of the global climate change, or an analysis of the seasonal variability of several factors (e.g. temperature, rainfall, and wind) instead of the whole weather system. Here we target to emphasize the state-of-art abilities (model estimation, state, and parameter inference) of the proposed algorithms without using an explicit formulation of the dynamical system. In such contexts, only some components of the state variable involved in the systems of ODEs are of interests and parametric estimates might focus on the chosen components. Then, the use of expensive numerical forecast models derived from integrating the systems of ODEs of the whole world/space of a physical phenomenon in the

classical algorithms is no longer necessary. Furthermore, we expect to extend the application fields of these novel methods.

Up to the present time, the proposed non-parametric methods have been constructed as the (partial) combination of standard regression forecast emulators (analog emulators) and particle-based samplers. In practice, working with these tools often suffers from curse of dimensionality in high dimensional state spaces (e.g. see in [9,66,87,130,133,168]). For the regression approaches in machine learning, the problem is so-called "large $d$, small $T$" (e.g. the number of parameters in slopes and intercepts of variables estimated by LLR are much larger than the number of necessary analogs in learning data). Regarding the particle-based samplers such as bootstrap PF/PS and bootstrap CPF/CPS, they typically meet degeneracy and sample impoverishment (the amount of values in particle sets decreases when the dimension of the state and/or the length of the observation sequence increases). There exist numerous solutions proposed to deal with the curse of dimensionality problem in regression methods (e.g principle components analysis and classical scaling [167,173,178]) and particle-based methods (e.g. proposal kernel improvements, localization and block sampling [97,113,124,142,168,169]). As an expectation for applications of the non-parametric inference algorithms in high dimensional problems, we continue to develop these algorithms in conjunction with the mentioned solutions.

We also pay attention to other research topics related to the proposed filtering (smoothing) and SEM algorithms. One potential topic consists in considering the cases of adaptive model covariances which depend on the state values in heteroskedastic time series [51,157,179]. These scenarios often occur in meteorological models (e.g. wind intensity and rainfall). Besides, it is possible to relax the Gaussian assumption of error noises, that help to detect all abilities of the particle-based methods for statistical inference in nonlinear non-Gaussian models. For instance, in the case of extreme values in climatology, GEV distribution is more frequently used to describe the extreme phenomena than the Gaussian distribution [36]. Discussions of potential approaches for handling these two problems are summarized in Chapter 1 [Section 1.2.2.3].

Theoretical studies are very important in further developments of the proposed methods. In lots of references of local regression methods (e.g. [35,60,61]), of CPFs/CPSs (e.g. [4,99,101,102]), and of SEM machines (e.g. [41, 44, 149, 170]), their asymptotic behaviors were considered and proven. In future research, we are going to explore properties of the proposed approaches using the combination of the theoretical behaviors of the mentioned materials.

# List of Figures

# List of Tables

# Bibliography


[1] Pierre Ailliot, Julie Bessac, Valérie Monbet, and Françoise Pene. Non-homogeneous hidden markov-switching models for wind time series. *Journal of Statistical Planning and Inference*, 160:75–88, 2015.

[2] Idrissa Amour and Tuomo Kauranne. A variational ensemble kalman filtering method for data assimilation using 2d and 3d version of coherens model. *International Journal for Numerical Methods in Fluids*, 83(6):544–558, 2017.

[3] Jeffrey L Anderson. An ensemble adjustment kalman filter for data assimilation. *Monthly weather review*, 129(12):2884–2903, 2001.

[4] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[5] Christophe Andrieu, Arnaud Doucet, Sumeetpal S Singh, and Vladislav B Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, 2004.

[6] Germán Aneiros-Pérez, Ricardo Cao, and Juan M Vilar-Fernández. Functional methods for time series prediction: a nonparametric approach. *Journal of Forecasting*, 30(4):377–392, 2011.

[7] Mohammad Bannayan and Gerrit Hoogenboom. Weather analogue: a tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach. *Environmental Modelling & Software*, 23(6):703–713, 2008.

[8] TP Barnett and RW Preisendorfer. Multifield analog prediction of short-term climate fluctuations using a climate state vector. *Journal of the Atmospheric Sciences*, 35(10):1771–1787, 1978.

[9] Thomas Bengtsson, Peter Bickel, Bo Li, et al. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

[10] Tyrus Berry and Timothy Sauer. Adaptive ensemble kalman filtering of non-linear systems. *Tellus A: Dynamic Meteorology and Oceanography*, 65(1):20331, 2013.

[11] Laurent Bertino, Geir Evensen, and Hans Wackernagel. Sequential data assimilation techniques in oceanography. *International Statistical Review*, 71(2):223–241, 2003.

[12] M. Bocquet and P. Sakov. Joint state and parameter estimation with an iterative ensemble Kalman smoother. *Nonlin. Processes Geophys.*, 20:803–818, 2013.

[13] M. Bocquet and P. Sakov. An iterative ensemble Kalman smoother. *Q. J. R. Meteorol. Soc.*, 140:1521–1535, 2014.

[14] Marc Bocquet. Ensemble kalman filtering without the intrinsic need for inflation. *Nonlinear Processes in Geophysics*, 18(5):735–750, 2011.

[15] Marc Bocquet and Pavel Sakov. Combining inflation-free and iterative ensemble kalman filters for strongly nonlinear systems. *Nonlinear Processes in Geophysics*, 19(3):383–399, 2012.

[16] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[17] Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61, 2010.

[18] Andrew Briggs, Taane Clark, Jane Wolstenholme, and Philip Clarke. Missing.... presumed at random: cost-analysis of incomplete data. *Health economics*, 12(5):377–392, 2003.

[19] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2013.

[20] Robert Grover Brown, Patrick YC Hwang, et al. *Introduction to random signals and applied Kalman filtering*, volume 3. Wiley New York, 1992.

[21] Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.

[22] John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[23] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[24] Alberto Carrassi, Marc Bocquet, Laurent Bertino, and Geir Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 0(0):e535, 2018.

[25] Alberto Carrassi, Marc Bocquet, Alexis Hannart, and Michael Ghil. Estimating model evidence using data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):866–880, 2017.

[26] Raymond J Carroll, David Ruppert, Ciprian M Crainiceanu, and Leonard A Stefanski. *Measurement error in nonlinear models: a modern perspective*. Chapman and Hall/CRC, 2006.

[27] Jake Carson, Michel Crucifix, Simon Preston, and Richard D Wilkinson. Bayesian model selection for the glacial–interglacial cycle. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(1):25–54, 2018.

[28] Gilles Celeux, Didier Chauveau, and Jean Diebolt. On Stochastic Versions of the EM Algorithm. Research Report RR-2514, INRIA, 1995.

[29] KS Chan and Johannes Ledolter. Monte carlo em estimation for time series models involving counts. *Journal of the American Statistical Association*, 90(429):242–252, 1995.

[30] Thi Tuyet Trang Chau, Pierre Ailliot, Valérie Monbet, and Pierre Tandeo. An efficient particle-based method for maximum likelihood estimation in nonlinear state-space models. *arXiv preprint arXiv:1804.07483*, 2018.

[31] Lu-Hung Chen, Ming-Yen Cheng, and Liang Peng. Conditional variance estimation in heteroscedastic regression models. *Journal of Statistical Planning and Inference*, 139(2):236–245, 2009.

[32] Nicolas Chopin et al. Central limit theorem for sequential monte carlo methods and its application to bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

[33] Nicolas Chopin, Sumeetpal S Singh, et al. On particle gibbs sampling. *Bernoulli*, 21(3):1855–1883, 2015.

[34] Charles K Chui, Guanrong Chen, et al. *Kalman filtering*. Springer, 2017.

[35] William S Cleveland and Susan J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.

[36] Stuart Coles, Joanna Bawa, Lesley Trenner, and Pat Dorazio. *An introduction to statistical modeling of extreme values*, volume 208. Springer, 2001.

[37] Jacques JF Commandeur and Siem Jan Koopman. *An introduction to state space time series analysis*. Oxford University Press, 2007.

[38] Jan G De Gooijer and Dawit Zerom. On conditional density estimation. *Statistica Neerlandica*, 57(2):159–176, 2003.

[39] Dick P Dee. Simplification of the kalman filter for meteorological data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 117(498):365–384, 1991.

[40] Pierre Del Moral. Feynman-kac formulae. In *Feynman-Kac Formulae*, pages 47–93. Springer, 2004.

[41] Bernard Delyon, Marc Lavielle, and Eric Moulines. Convergence of a stochastic approximation version of the em algorithm. *Annals of statistics*, pages 94–128, 1999.

[42] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[43] Gérald Desroziers, Loic Berre, Bernard Chapnik, and Paul Poli. Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3385–3396, 2005.

[44] J Diebolt, E Ip, and Ingram Olkin. A stochastic em algorithm for approximating the maximum likelihood estimate. *Markov chain Monte Carlo in practice. Chapman and Hall, Dordrect, The Netherlands*, 1996.

[45] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[46] Randal Douc, Aurelien Garivier, Eric Moulines, and Jimmy Olsson. On the forward filtering backward smoothing particle approximations of the smoothing distribution in general state spaces models. *arXiv preprint arXiv:0904.0316*, 2009.

[47] Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[48] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[49] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[50] Denis Dreano, Pierre Tandeo, Manuel Pulido, Boujemaa Ait-El-Fquih, Thierry Chonavel, and Ibrahim Hoteit. Estimating model-error covariances in nonlinear state-space models using kalman smoothing and the expectation–maximization algorithm. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1877–1885, 2017.

[51] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

[52] Ahmed H Elsheikh, Ibrahim Hoteit, and Mary F Wheeler. Efficient bayesian inference of subsurface flow models using nested sampling and sparse polynomial chaos surrogates. *Computer Methods in Applied Mechanics and Engineering*, 269:515–537, 2014.

[53] Ahmed H Elsheikh, Mary F Wheeler, and Ibrahim Hoteit. Hybrid nested sampling algorithm for bayesian model selection applied to inverse subsurface flow problems. *Journal of Computational Physics*, 258:319–337, 2014.

[54] Geir Evensen. Using the extended kalman filter with a multilayer quasi-geostrophic ocean model. *Journal of Geophysical Research: Oceans*, 97(C11):17905–17924, 1992.

[55] Geir Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

[56] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[57] Geir Evensen. Analysis of iterative ensemble smoothers for solving inverse problems. *Computational Geosciences*, 22(3):885–908, 2018.

[58] Geir Evensen and Peter Jan Van Leeuwen. An ensemble kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

[59] Ronan Fablet, Phi Huynh Viet, and Redouane Lguensat. Data-driven models for the spatio-temporal interpolation of satellite-derived sst fields. *IEEE Transactions on Computational Imaging*, 3(4):647–657, 2017.

[60] Jianqing Fan. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Routledge, 2018.

[61] Jianqing Fan and Qiwei Yao. *Nonlinear time series: nonparametric and parametric methods*. Springer Science & Business Media, 2008.

[62] Jianqing Fan, Qiwei Yao, and Howell Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206, 1996.

[63] Jianqing Fan and J-T Zhang. Two-step estimation of functional linear models with applications to longitudinal data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):303–322, 2000.

[64] Paul Fearnhead and Hans R Künsch. Particle filters and data assimilation. *Annual Review of Statistics and Its Application*, 5:421–449, 2018.

[65] Gláucia Tatiana Ferrari and Vitor Ozaki. Missing data imputation of climate datasets: Implications to modeling extreme drought events. *Revista Brasileira de Meteorologia*, 29(1):21–28, 2014.

[66] Jerome H Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

[67] G Galanis, P Louka, P Katsafados, G Kallos, and I Pytharoulis. Applications of kalman filters based on non-linear functions to numerical weather predictions. *Ann. Geophys*, 24:1–10, 2006.

[68] Michael Ghil and Paola Malanotte-Rizzoli. Data assimilation in meteorology and oceanography. In *Advances in geophysics*, volume 33, pages 141–266. Elsevier, 1991.

[69] Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

[70] AH Abdul Hafez. Depth estimation using particle filters for image-based visual servoing. *Journal of Control Engineering and Applied Informatics*, 18(2):48–56, 2016.

[71] Peter Hall, Rodney CL Wolff, and Qiwei Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical association*, 94(445):154–163, 1999.

[72] A Hannart, J Pearl, FEL Otto, P Naveau, and M Ghil. Causal counterfactual theory for the attribution of weather and climate-related events. *Bulletin of the American Meteorological Society*, 97(1):99–110, 2016.

[73] Alexis Hannart, Alberto Carrassi, Marc Bocquet, Michael Ghil, Philippe Naveau, Manuel Pulido, Juan Ruiz, and Pierre Tandeo. Dada: data assimilation for the detection and attribution of weather and climate-related events. *Climatic Change*, 136(2):155–174, 2016.

[74] Simon Haykin. *Kalman filtering and neural networks*, volume 47. John Wiley & Sons, 2004.

[75] Nils-Bastian Heidenreich, Anja Schindler, and Stefan Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA Advances in Statistical Analysis*, 97(4):403–433, Oct 2013.

[76] Diederich Hinrichsen and Anthony J Pritchard. *Mathematical systems theory I: modelling, state space analysis, stability and robustness*, volume 48. Springer Berlin, 2005.

[77] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.

[78] Pascal Horton, Michel Jaboyedoff, and Charles Obled. Global optimization of an analog method by means of genetic algorithms. *Monthly Weather Review*, 145(4):1275–1294, 2017.

[79] Ibrahim Hoteit, Dinh-Tuan Pham, George Triantafyllou, and Gerasimos Korres. A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Monthly Weather Review*, 136(1):317–334, 2008.

[80] Peter L Houtekamer and Herschel L Mitchell. Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, 126(3):796–811, 1998.

[81] PL Houtekamer and Fuqing Zhang. Review of the ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 144(12):4489–4532, 2016.

[82] José M Jerez, Ignacio Molina, Pedro J García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial intelligence in medicine*, 50(2):105–115, 2010.

[83] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

[84] WL Junger and A Ponce De Leon. Imputation of missing data in time series for air pollutants. *Atmospheric Environment*, 102:96–104, 2015.

[85] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[86] Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan Maciejowski, Nicolas Chopin, et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351, 2015.

[87] Eamonn Keogh and Abdullah Mueen. Curse of dimensionality. In *Encyclopedia of Machine Learning and Data Mining*, pages 314–315. Springer, 2017.

[88] Genshiro Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, pages 1203–1215, 1998.

[89] Juho Kokkala, Arno Solin, and Simo Särkkä. Expectation maximization based parameter estimation by sigma-point and particle smoothing. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8. IEEE, 2014.

[90] Estelle Kuhn and Marc Lavielle. Coupling a stochastic approximation version of em with an mcmc procedure. *ESAIM: Probability and Statistics*, 8:115–131, 2004.

[91] Upmanu Lall and Ashish Sharma. A nearest neighbor bootstrap for resampling hydrologic time series. *Water Resources Research*, 32(3):679–693, 1996.

[92] Bernard Lamien, Helcio Rangel Barreto Orlande, and Guillermo Enrique Eliçabe. Particle filter and approximation error model for state estimation in hyperthermia. *Journal of Heat Transfer*, 139(1):012001, 2017.

[93] François Le Gland, Valérie Monbet, and Vu Duc Tran. Large sample asymptotics for the ensemble kalman filter. In Dan Crisan and Boris Rozovskii, editors, *Handbook on Nonlinear Filtering*, chapter 22, pages 598–631. Oxford University Press, Oxford, 2009.

[94] Nayoung Lee, Hyungsik Roger Moon, and Qiankun Zhou. Many ivs estimation of dynamic panel regression models with measurement error. *Journal of Econometrics*, 200(2):251–259, 2017.

[95] Redouane Lguensat, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet. The analog data assimilation. *Monthly Weather Review*, 145(10):4093–4107, 2017.

[96] Liangping Li, Ryan Puzel, and Arden Davis. Data assimilation in groundwater modelling: ensemble kalman filter versus ensemble smoothers. *Hydrological Processes*, 32(13):2020–2029, 2018.

[97] Tiancheng Li, Shudong Sun, Tariq Pervez Sattar, and Juan Manuel Corchado. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with applications*, 41(8):3944–3954, 2014.

[98] Fredrik Lindsten. An efficient stochastic approximation em algorithm using conditional particle filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6274–6278. IEEE, 2013.

[99] Fredrik Lindsten, Michael I Jordan, and Thomas B Schön. Particle gibbs with ancestor sampling. *The Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

[100] Fredrik Lindsten, Thomas Schön, and Michael I Jordan. Ancestor sampling for particle gibbs. In *Advances in Neural Information Processing Systems*, pages 2591–2599, 2012.

[101] Fredrik Lindsten and Thomas B Schön. On the use of backward simulation in particle markov chain monte carlo methods. *arXiv preprint arXiv:1110.2873*, 2011.

[102] Fredrik Lindsten, Thomas B Schön, et al. Backward simulation methods for monte carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143, 2013.

[103] X CHEN O LINTON and PM ROBINSON. The estimation of conditional densities. *Asymptotics in Statistics and Probability: Papers in Honor of George Gregory Roussas*, page 71, 2000.

[104] Roderick JA Little. Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, 6(3):287–296, 1988.

[105] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 333. John Wiley & Sons, 2014.

[106] Yun Liu, J-M Haussaire, Marc Bocquet, Yelva Roustan, Olivier Saunier, and Anne Mathieu. Uncertainty quantification of pollutant source retrieval: comparison of bayesian methods with application to the chernobyl and fukushima daiichi accidental releases of radionuclides. *Quarterly Journal of the Royal Meteorological Society*, 143(708):2886–2901, 2017.

[107] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[108] Jan Mandel. A brief tutorial on the ensemble kalman filter. *arXiv preprint arXiv:0901.3725*, 2009.

[109] Kantilal Varichand Mardia. *Statistics of directional data*. Academic press, 2014.

[110] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[111] Erik Meijer, Laura Spierdijk, and Tom Wansbeek. Measurement error in the linear dynamic panel data model. In *ISS-2012 Proceedings Volume On Longitudinal Data Analysis Subject to Measurement Errors, Missing Values, and/or Outliers*, pages 77–92. Springer, 2013.

[112] Takemasa Miyoshi. The gaussian approach to adaptive covariance inflation and its implementation with the local ensemble transform kalman filter. *Monthly Weather Review*, 139(5):1519–1535, 2011.

[113] Christian Naesseth, Fredrik Lindsten, and Thomas Schon. Nested sequential monte carlo methods. In *International Conference on Machine Learning*, pages 1292–1301, 2015.

[114] Katsuhiko Ogata. *Discrete-time control systems*, volume 2. Prentice Hall Englewood Cliffs, NJ, 1995.

[115] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39. Springer, 2004.

[116] Jimmy Olsson, Olivier Cappé, Randal Douc, Eric Moulines, et al. Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

[117] Nicolas Papadakis, Étienne Mémin, Anne Cuzol, and Nicolas Gengembre. Data assimilation with the weighted ensemble kalman filter. *Tellus A: Dynamic Meteorology and Oceanography*, 62(5):673–697, 2010.

[118] Charles M Paulsen, Richard A Hinrichsen, and Timothy R Fisher. Measure twice, estimate once: Pacific salmon population viability analysis for highly variable populations. *Transactions of the American Fisheries Society*, 136(2):346–364, 2007.

[119] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[120] Dinh Tuan Pham, Jacques Verron, and Marie Christine Roubaud. A singular evolutive extended kalman filter for data assimilation in oceanography. *Journal of Marine systems*, 16(3-4):323–340, 1998.

[121] Umberto Picchini and Adeline Samson. Coupling stochastic em and approximate bayesian computation for parameter inference in state-space models. *Computational Statistics*, 33(1):179–212, 2018.

[122] Pierre Pinson, Henrik Aa Nielsen, Henrik Madsen, and Torben S Nielsen. Local linear regression with adaptive orthogonal fitting for the wind power application. *Statistics and Computing*, 18(1):59–71, 2008.

[123] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[124] Jonathan Poterjoy. A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review*, 144(1):59–76, 2016.

[125] Rossella Lo Presti, Emanuele Barca, and Giuseppe Passarella. A methodology for treating missing data applied to daily rainfall data in the candelaro river basin (italy). *Environmental monitoring and assessment*, 160(1-4):1, 2010.

[126] Manuel Pulido, Pierre Tandeo, Marc Bocquet, Alberto Carrassi, and Magdalena Lucini. Stochastic parameterization identification using ensemble kalman filtering combined with maximum likelihood methods. *Tellus A: Dynamic Meteorology and Oceanography*, 70(1):1–17, 2018.

[127] Xu Qin, She J Zhang, and Dong X Yan. A new circular distribution and its application to wind data. *Journal of Mathematics Research*, 2(3):12, 2010.

[128] Balaji Rajagopalan and Upmanu Lall. A k-nearest-neighbor simulator for daily precipitation and other weather variables. *Water resources research*, 35(10):3089–3101, 1999.

[129] Herbert E Rauch, CT Striebel, and F Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[130] Patrick Rebeschini, Ramon Van Handel, et al. Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866, 2015.

[131] Sebastian Reich and Colin Cotter. *Probabilistic forecasting and Bayesian data assimilation.* Cambridge University Press, 2015.

[132] Rolf H Reichle, Dennis B McLaughlin, and Dara Entekhabi. Hydrologic data assimilation with the ensemble kalman filter. *Monthly Weather Review*, 130(1):103–114, 2002.

[133] James M Robins and Ya'acov Ritov. Toward a curse of dimensionality appropriate (coda) asymptotic theory for semi-parametric models. *Statistics in medicine*, 16(3):285–319, 1997.

[134] Guillermo Rodriguez. Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics. *IEEE Journal on Robotics and Automation*, 3(6):624–639, 1987.

[135] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[136] David Ruppert and Matthew P Wand. Multivariate locally weighted least squares regression. *The annals of statistics*, pages 1346–1370, 1994.

[137] Pavel Sakov and Peter R Oke. A deterministic formulation of the ensemble kalman filter: an alternative to ensemble square root filters. *Tellus A: Dynamic Meteorology and Oceanography*, 60(2):361–371, 2008.

[138] DJ Salmond and H Birch. A particle filter for track-before-detect. In *American Control Conference, 2001. Proceedings of the 2001*, volume 5, pages 3755–3760. IEEE, 2001.

[139] Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

[140] Mohammad-Taghi Sattari, Ali Rezazadeh-Joudi, and Andrew Kusiak. Assessment of different methods for estimation of missing data in precipitation studies. *Hydrology Research*, 48(4):1032–1044, 2017.

[141] Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

[142] François Septier and Gareth W Peters. An overview of recent advances in monte-carlo methods for bayesian filtering in high-dimensional spaces. In *Theoretical Aspects of Spatial-Temporal Modeling*, pages 31–61. Springer, 2015.

[143] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

[144] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[145] Chris Snyder. Particle filters, the "optimal" proposal and high-dimensional systems. In *Proceedings of the ECMWF Seminar on Data Assimilation for atmosphere and ocean*, pages 1–10, 2011.

[146] Matthew Stephens and Paul Scheet. Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *The American Journal of Human Genetics*, 76(3):449–462, 2005.

[147] Jonathan R Stroud and Thomas Bengtsson. Sequential state and variance estimation within the ensemble kalman filter. *Monthly Weather Review*, 135(9):3194–3208, 2007.

[148] Jonathan R Stroud, Matthias Katzfuss, and Christopher K Wikle. A bayesian adaptive ensemble kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, 146(1):373–386, 2018.

[149] Andreas Svensson and Fredrik Lindsten. Learning dynamical systems with particle stochastic approximation em. *arXiv preprint arXiv:1806.09548*, 2018.

[150] Andreas Svensson, Thomas B. Schön, and Manon Kok. Nonlinear state space smoothing using the conditional particle filter**this work was supported by the project probabilistic modelling of dynamical systems (contract number: 621-2013-5524) and cadics, a linnaeus center, both funded by the swedish research council (vr). *IFAC-PapersOnLine*, 48(28):975 – 980, 2015. 17th IFAC Symposium on System Identification SYSID 2015.

[151] Andreas Svensson and Thomas B Schön. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

[152] Pierre Tandeo, Pierre Ailliot, Marc Bocquet, Alberto Carrassi, Takemasa Miyoshi, Manuel Pulido, and Yicun Zhen. Joint estimation of model and observation error covariance matrices in data assimilation: a review. *arXiv preprint arXiv:1807.11221*, 2018.

[153] Pierre Tandeo, Pierre Ailliot, Bertrand Chapron, Redouane Lguensat, and Ronan Fablet. The analog data assimilation: application to 20 years of altimetric data. In *CI 2015: 5th International Workshop on Climate Informatics*, pages 1–2, 2015.

[154] Pierre Tandeo, Pierre Ailliot, Ronan Fablet, Juan Ruiz, François Rousseau, and Bertrand Chapron. The analog ensemble kalman filter and smoother. In *CI 2014: 4th International Workshop on Climate Informatics*, 2014.

[155] Pierre Tandeo, Pierre Ailliot, Juan Ruiz, Alexis Hannart, Bertrand Chapron, Anne Cuzol, Valérie Monbet, Robert Easton, and Ronan Fablet. Combining analog method and

ensemble data assimilation: application to the lorenz-63 chaotic system. In *Machine Learning and Data Mining Approaches to Climate Science*, pages 3–12. Springer, 2015.

[156] Pierre Tandeo, Manuel Pulido, and François Lott. Offline parameter estimation using enkf and maximum likelihood error covariance estimates: Application to a subgrid-scale orography parametrization. *Quarterly Journal of the Royal Meteorological Society*, 141(687):383–395, 2015.

[157] James W Taylor, Patrick E McSharry, Roberto Buizza, et al. Wind power density forecasting using ensemble predictions and time series models. *IEEE Transactions on Energy Conversion*, 24(3):775, 2009.

[158] Francisco Curado Teixeira, João Quintas, Pramod Maurya, and António Pascoal. Robust particle filter formulations with application to terrain-aided navigation. *International Journal of Adaptive Control and Signal Processing*, 31(4):608–651, 2017.

[159] George R Terrell and David W Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[160] J Timmer. Modeling noisy time series: physiological tremor. *International Journal of Bifurcation and Chaos*, 8(07):1505–1516, 1998.

[161] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[162] Concepción Crespo Turrado, María del Carmen Meizoso López, Fernando Sánchez Lasheras, Benigno Antonio Rodríguez Gómez, José Luis Calvo Rollé, and Francisco Javier de Cos Juez. Missing data imputation of solar radiation data under different atmospheric conditions. *Sensors*, 14(11):20382–20399, 2014.

[163] Genta Ueno, Tomoyuki Higuchi, Takashi Kagimoto, and Naoki Hirose. Maximum likelihood estimation of error covariances in ensemble-based filters and its application to a coupled atmosphere–ocean model. *Quarterly Journal of the Royal Meteorological Society*, 136(650):1316–1343, 2010.

[164] Genta Ueno and Nagatomo Nakamura. Iterative algorithm for maximum-likelihood estimation of the observation-error covariance matrix for ensemble-based filters. *Quarterly Journal of the Royal Meteorological Society*, 140(678):295–315, 2014.

[165] Genta Ueno and Nagatomo Nakamura. Bayesian estimation of the observation-error covariance matrix in ensemble-based filters. *Quarterly Journal of the Royal Meteorological Society*, 142(698):2055–2080, 2016.

[166] Stef Van Buuren. *Flexible imputation of missing data*. Chapman and Hall/CRC, 2018.

[167] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.

[168] Peter Jan van Leeuwen. Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999, 2010.

[169] Peter Jan Van Leeuwen. Nonlinear data assimilation for high-dimensional systems. In *Nonlinear Data Assimilation*, pages 1–73. Springer, 2015.

[170] Greg CG Wei and Martin A Tanner. 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, 1990.

[171] Nick Whiteley. Discussion on particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):306–307, 2010.

[172] Christopher K Wikle and L Mark Berliner. A bayesian tutorial for data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):1–16, 2007.

[173] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[174] Daniel B Work, Sébastien Blandin, Olli-Pekka Tossavainen, Benedetto Piccoli, and Alexandre M Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010(1):1–35, 2010.

[175] Pascal Yiou. Anawege: a weather generator based on analogues of atmospheric circulation. *Geoscientific Model Development*, 7(2):531–543, 2014.

[176] Derek S Young and David R Hunter. Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics & Data Analysis*, 54(10):2253–2266, 2010.

[177] K Yu and MC Jones. Likelihood-based local linear estimation of the conditional variance function. *Journal of the American Statistical Association*, 99(465):139–144, 2004.

[178] Ming Yuan, Ali Ekici, Zhaosong Lu, and Renato Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):329–346, 2007.

[179] Fadhilah Yusof and Ibrahim Lawal Kane. Volatility modeling of rainfall time series. *Theoretical and applied climatology*, 113(1-2):247–258, 2013.

[180] Fuqing Zhang, Chris Snyder, and Juanzhen Sun. Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble kalman filter. *Monthly Weather Review*, 132(5):1238–1253, 2004.

[181] Yicun Zhen and John Harlim. Adaptive error covariances estimation methods for ensemble kalman filters. *Journal of computational physics*, 294:619–638, 2015.

[182] Mengbin Zhu, Peter J Van Leeuwen, and Weimin Zhang. Estimating model error covariances using particle filters. *Quarterly Journal of the Royal Meteorological Society*, 2017.

[183] Eduardo Zorita and Hans Von Storch. The analog method as a simple statistical downscaling technique: comparison with more complicated methods. *Journal of climate*, 12(8):2474–2489, 1999.

# Résumé

## Thèse: Méthodologies non-paramétriques pour la reconstruction et l'estimation dans les modèles d'états non linéaires

Thi Tuyet Trang CHAU

# Motivations

Grâce au développement des sciences technologiques et informatiques, la quantité et la qualité nombre de données a augmenté au cours des dernières décennies. Cette thèse a été motivée par les applications d'analyse de données en environnement, climatologie et océanographie. Dans ces domaines, les exponentielles croissance de la disponibilité des données obtenues par télédétection, in situ ou par modèle devrait se poursuivre dans les années à venir. L'avenir crée de nombreuses opportunités, besoins et défis. En particulier, l'environnement données sont généralement disponibles avec un échantillonnage spatio-temporel complexe, sur des grilles irrégulières, et sujet à des erreurs d'observation dues à la complexité de la collecte des données, de la modélisation des imperfections,etc.

Les modèles d'espaces d'état (SSM) [8,16,22,32] sont une approche populaire pour analyser des données avec erreurs d'observation. En particulier, ils sont au cœur des technologies d'assimilation séquentielle des données notamment en océanographie et en météorologie. Les SSM sont consitués d'un modèle dynamique, qui décrit l'évolution physique du phénomène d'intérêt, et un modèle d'observation qui modélise la relation entre les observations (bruitées) et l'état (vrai). De nombreuses difficultés surviennent quand on travaille avec les SSM et dans cette thèse nous nous concentrons sur les défis suivants (voir la Figure 1 pour une illustration de ces défis).

i. **Reconstruction d'état lorsque le modèle dynamique est connu et les paramètres sont connus**

   Le filtrage et le lissage (assimilation séquentielle de données en géosciences) sont des approches usuelles pour estimer récursivement les distributions de probabilité de l'état conditionnellement à une séquence d'observations. Dans le cadre de l'assimilation, le modèle dynamique est utilisé pour propager des estimations de l'état d'un temps passé à des temps plus récents. Les prévisions sont alors corrigées en tenant compte des observations disponibles.

   Pour les modèles linéaires gaussiens, les récurrences de Kalman [16, 21, 23, 34, 35] peuvent être utilisées pour analyser correctement les distributions de filtrage et de lissage. Quand les modèles d'espace états sont non linéaires, comme c'est le cas typique des applications réelles, ces distributions n'admettent pas d'expression explicite. Des méthodes basées sur la simulation sont implémentées. Les approches basées sur le filtre de Kalman d'ensemble (voir par exemple dans [3, 17, 18])) sont les approches d'assimilation les plus utilisées en pratique en raison de leur efficacité à approcher les distributions de

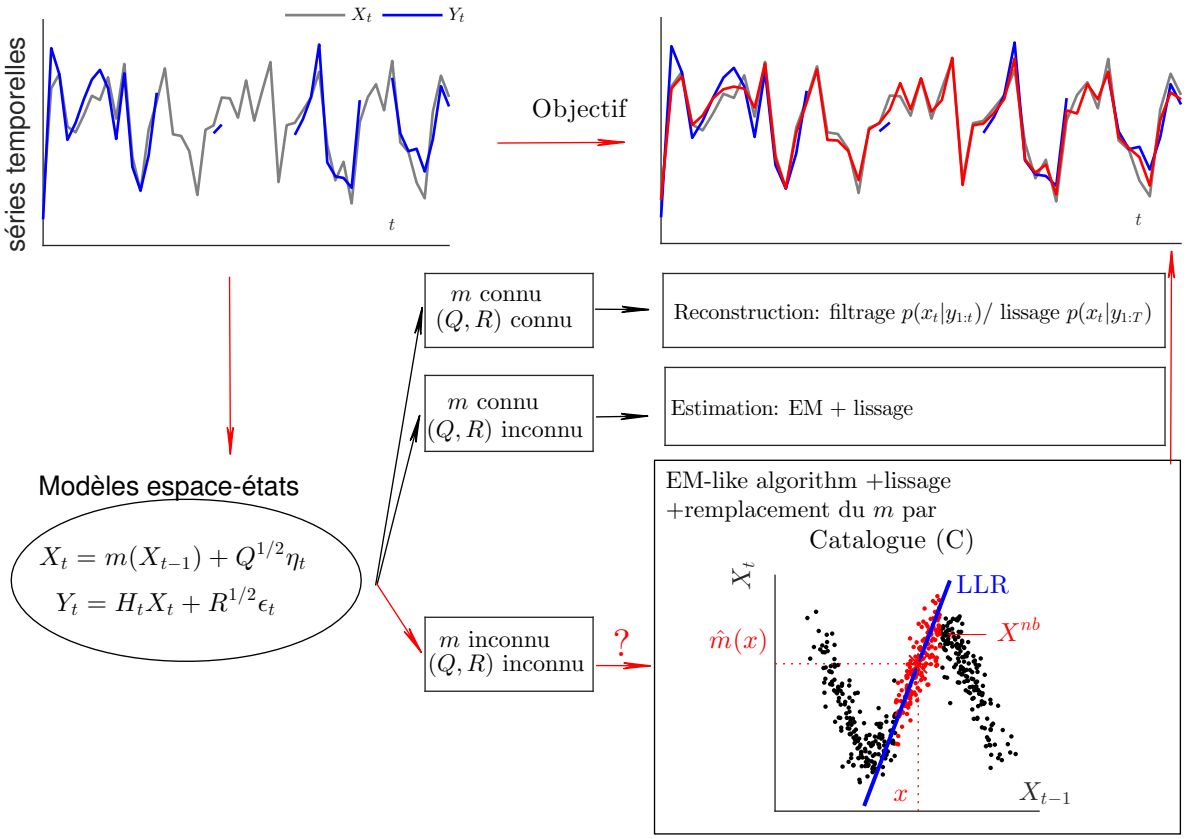


**Figure 1:** Illustration des problèmes d'inférence statistique dans la thèse.

filtrage et de lissage de problèmes de grande dimension (seulement quelques simulations (membres) du modèle dynamique sont exécutées). Néanmoins, les approximations ne convergent pas vers la vraie distribution conditionnelle pour des situations (hautement) non linéaires [25]. En statistique et traitement du signal, les filtres à particules sont utilisés comme outils puissants et flexibles pour reconstruire l'état dans des modèles non linéaires et / ou non gaussiens. De nombreux algorithmes ont été proposés dans la littérature [4, 13, 14, 20].

ii. **Estimation des paramètres lorsque le modèle dynamique est spécifié avec paramètres inconnus**

La précision des résultats obtenus quand on reconstruit des variables physiques, à partir données observées, à l'aide de SSM ne dépend pas uniquement des méthodes d'assimilation, mais est liées aux paramètres statiques des erreurs. En pratique, il est souvent difficile de spécifier des valeurs raisonnables pour ces paramètres inconnus. Ceci est dû à la diversité des sources d'observation, à l'effet des termes physiques et complexité du modèle ou à des échecs numériques [15, 44]. Par conséquent, l'estimation des paramètres (ou identification du système) est une tâche préliminaire importante avant de réaliser l'assimilation de données.

Les approches statistiques habituelles pour l'estimation des paramètres sont basées sur des méthodes bayésienne et ou du maximum de vraisemblance. Les approches bayésiennes [1, 24, 30, 37, 38, 42] visent à simuler la répartition conjointe de l'état et du paramètre, mais cela n'est pas toujours possible pour des SSM de grande dimension (par exemple, inférence d'erreur de covariance). Une alternative consiste à mettre en œuvre des approches d'estimation par maximum de vraisemblance, notamment via l'algorithme (EM) [11] et

ses variantes [5, 10, 12, 31].

iii. **Reconstruction d'état et estimation des paramètres lorsque le modèle dynamique est inconnu**

Dans les applications en géosciences, le modèle dynamique est généralement spécifié par des équations dérivées de la physique et résolues à l'aide de schémas numériques. Le modèle numérique de prévision doit être exécuté pour chaque étape du processus d'assimilation ce qui conduit à des coûts de calcul élevés dans la pratique. De plus, les comportements chaotiques et la complexité du modèle peuvent être des raisons d'approximations numériques inexactes. En outre, diverses sources d'incertitude (paramètres physiques inconnus, variance du bruit d'état, forçagess) peuvent entraîner un biais important entre les prévisions et les observations. Dans de telles situations, le processus d'assimilation peut être incohérent.

De nos jours, une énorme quantité de jeux de données enregistrés sur des satellites, in situ ou extraits de simulations numériques est disponible. L'existence de telles données favorise le développement de modèles basés sur les données, capables de bien décrire la dynamique de l'état. Les combinaisons d'approches non paramétriques avec des algorithmes standard de filtrage et de lissage ont été proposés pour la première fois dans [26, 41].

Trois contributions principales de cette thèse à ces trois défis sont énumérées ci-dessous.

# Principales contributions

i. **Reconstruction d'état lorsque le modèle dynamique est connu et les paramètres sont connus**

Récemment [1, 28, 30, 43] ont mis au point des filtres à particules conditionnels qui permettent d'approcher efficacement la distribution de lissage avec seulement quelques particules. Dans la thèse nous étudions l'algorihme de lissage conditional particle filter – backward smoother (CPF-BS) présenté dans [29, 30, 43] et discuté plus en détail dans [6]. Nous allons montrer sur plusieurs modèles de jouets que, pour un coût de calcul équivalent, l'algorithme CPF-BS donne de meilleurs résultats que les algorithmes particulaires de lissage usuels.

ii. **Estimation des paramètres lorsque le modèle dynamique est spécifié avec paramètre inconnus** Lors de l'utilisation des algorithmes EM, les paramètres sont mis à jour de manière itérative en maximisant une fonction de vraisemblance définie à l'aide de la distribution de lissage. Néanmoins, la distribution de lissage n'a pas d'expression explicite dans es SSM non linéaires. Dans les articles de [2, 24, 27, 33, 36, 39], il a été proposé de combiner les échantillonneurs standard de particules, qui permettent d'approcher la distribution de lissage, avec des méthodes EM. Mais cela conduit généralement à un énorme coût de calcul. Dans la thèse, nous explorons la combinaison de l'échantillonneur CPF-BS et d'algorithmes EM, et nous montrons que cette approche est plus performante que la combinaison des algorithmes stochastiques EnKS et EM couramment utilisés dans les applications réelles (voir [6]).

iii. **Reconstruction d'état et estimation des paramètres lorsque le modèle dynamique est non spécifié**

Inspirée des travaux de [26,41], cette thèse se concentre sur les méthodes non paramétriques pour la reconstruction de l'état et du modèle dynamique en utilisant uniquement les données observées dans des situations où le modèle dynamique n'est pas spécifié. Deux situations sont considérées. Dans le premier cas, on suppose qu'un jeu de données d'apprentissage simulé à partir du processus d'état sans erreur d'observation est disponible (comme dans [26, 41]). Sur la base de ces données, le modèle dynamique peu être estimé par une méthode non paramétrique (telle que la régression locale [7, 9, 19]. En pratique, de telles observations "parfaites" de l'état, sans erreur d'observation, ne sont généralement pas disponibles. Dans la seconde situation, seule une séquence du processus avec des erreurs d'observation est disponible. Cela augmente les erreurs d'estimation si l'estimation non paramétrique est apprise directement sur ces données bruitées. Pour gérer ce problème, la thèse introduit un nouvel algorithme non paramétrique qui combine un une estimation non-paramétrique du modèle dynamique, un lisseur CPF-BS à faible coût et un algorithme de type EM. Les performances de l'approche proposée en termes de réduction des erreurs de bruit, d'imputation de données manquantes, d'estimation des paramètres et de comparaison de modèles sont illustrés à l'aide d'exemples de jouets et les données de vent produites par Météo France.

# Plan de la thèse

Le chapitre 1 présente les éléments fondamentaux et illustre les problèmes abordés dans la thèse. Les concepts des SSM et les exemples jouets sont d'abord introduits. À partir d'un ensemble d'observations et d'un modèle avec des paramètres connus, des méthodes de filtrage et de lissage permettant de calculer l'état caché sont rappelés. Nous synthétisons et analysons les avantages et les inconvénients de différentes méthodes y compris les filtres de Kalman, certaines de leurs extensions et les filtres à base de particules. Dans la suite, nous résumons les algorithmes EM existants utilisés pour traiter les problèmes d'inférence de SSM avec paramètres inconnus. L'efficacité de l'estimation des paramètres par des algorithmes EM combinés avec des filtres particulaires est discutée. l'accent est mis sur les filtres à base de particules et les lissoirs dans les modèles non linéaires. Avec l'objectif de développer des algorithmes non paramétriques, nous passons en revue les méthodes de régression linéaires locales (LLR) calssiques utilisées pour estimer le modèle dynamique. Enfin, nous présentons les idées clés de l'implémentation de ces émulateurs non paramétriques dans les algorithmes proposés.

Dans le chapitre 2, nous présentons des algorithmes de filtrage non paramétriques permettant d'estimer des distribution de filtrage dans les modèles SSM non linéaires. Ici, la régression linéaire locale (LLR) est de nouveau utilisée pour fournir des estimations non paramétriques du modèle dynamique. Elles sont ensuite combinées avec différents filtres tels que le filtre de Kalman étendu (EKF), le filtre de Kalman d'ensemble (EnKF), le bootstrap et le filtres particulaire (PF). La contribution principale de ce chapitre est la section des résultats numériques. De nombreuses expériences sont menées pour comparer les approches proposées avec les approches classiques, les approches proposées avec les approches non paramétriques utilisant des estimations par plus proches voisins, et les approches proposées dans différents schémas de filtrage. En résumé, ce chapitre étend les travaux précédents [26, 40, 41] en: (1) soulignant que la LLR donne une meilleure prédiction numérique que les méthodes de plus proches voisins classiques, (2) fournissant de nouvelles combinaisons d'émulateur no paramétriques avec les filtres de Kalman étendu et des filtres particulaires, (3) comparant toutes les approches mentionnées dans différents scénarios

Dans les applications d'assimliation de données en géosciences, les outils les plus utilisés pour déduire l'état du système des observations sont EnKF, EnKS et leurs extensions. Au chapitre 3, nous étudions une approche alternative, le CPF-BS. Ce lisseur permet d'explorer efficacement le l'espace d'état et de simuler rapidement des trajectoires pertinentes de l'état conditionnellement aux observations. Des illustrations numériques des algorithmes CPF-BS sur les modèles de jouets sont proposées de façon à aider les lecteurs à comprendre le processus de lissage facilement. En outre, nous proposons de combiner le lisseur CPF-BS avec un algorithme stochastique EM (SEM) original afin d'estimer les paramètres inconnus et l'état caché. Nous montrons sur plusieurs problèmes jouets que cet algorithme fournit, avec un coût de calcul raisonnable, des estimations précisesdes paramètres statiques et de l'état dans les SSM hautement non linéaires, où l'application d'un algorithme EM en conjonction avec EnKS est limité.

La contribution principale de cette thèse est présentée au chapitre 4. De nouveaux algorithmes non paramétriques sont développés pour résoudre deux problèmes. Tout d'abord, notre objectif est d'estimer les paramètres des lois des erreurs et de reconstruire l'état caché étant donné une séquence d'observations et un ensemble d'apprentissage "parfait" (une séquence simulée du processus d'état sans erreur d'observation). Sachant les données d'apprentissage, la LLR est utilisée pour construire une estimation du modèle dynamique. Sur la base du chapitre 3, nous proposons de combiner l'émulateur statistique avec le lisseur CPF-BS à faible coût. Ce lisseur non paramétrique est utilisé pour générer des réalisations de l'état dans un algorithme SEM. Néanmoins, de telles données "parfaites" existent rarement dans la réalité. Les données dérivées du processus d'observation sont le plus souvent bruités. Et, l'estimation du modèle dynamique sur les données bruitées mènent facilement à une augmentation du biais et de la variance et peut avoir des effets néfastes sur les résultats de l'inférence. Pour traiter ce problème, nous développons maintenant un algorithme de type SEM pour reconstruire la dynamique et estimation de paramètres inconnus dans le cas où on ne dispose que d'observations bruitéss. Enfin, différentes performances de la nouvelle méthode telles que la réduction des erreurs de bruit, l'imputation des données manquantes et l'estimation des paramètres sont illustrées sur les modèles de jouets.

Le chapitre 5 présente deux applications des algorithmes non paramétriques proposés. Tout d'abord, un algorithme de filtrage non paramétrique est appliqué à la sélection et à la comparaison de modèles étant donné un ensemble d'observations et des modèles existants. La performance de l'approche proposée est comparée à celle de l'approche classique sur des modèles de jouets avec différents paramètres de forçage. Ensuite, nous introduisons une application de l'algorithme npSEM pour l'imputation de données manquantes. Les données de vent produites par Météo France sont considérés. Les résultats de l'algorithme SEM non paramétrique sur les données sont comparés à ceux de méthodes de régression.

Enfin, le chapitre 6 récapitule les contributions de la thèse et introduit plusieurs sujets de recherche ultérieure.

# References


[1] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[2] Christophe Andrieu, Arnaud Doucet, Sumeetpal S Singh, and Vladislav B Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438,

2004.

[3] Marc Bocquet and Pavel Sakov. Combining inflation-free and iterative ensemble kalman filters for strongly nonlinear systems. *Nonlinear Processes in Geophysics*, 19(3):383–399, 2012.

[4] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[5] Gilles Celeux, Didier Chauveau, and Jean Diebolt. On Stochastic Versions of the EM Algorithm. Research Report RR-2514, INRIA, 1995.

[6] Thi Tuyet Trang Chau, Pierre Ailliot, Valérie Monbet, and Pierre Tandeo. An efficient particle-based method for maximum likelihood estimation in nonlinear state-space models. *arXiv preprint arXiv:1804.07483*, 2018.

[7] William S Cleveland and Susan J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.

[8] Jacques JF Commandeur and Siem Jan Koopman. *An introduction to state space time series analysis*. Oxford University Press, 2007.

[9] Jan G De Gooijer and Dawit Zerom. On conditional density estimation. *Statistica Neerlandica*, 57(2):159–176, 2003.

[10] Bernard Delyon, Marc Lavielle, and Eric Moulines. Convergence of a stochastic approximation version of the em algorithm. *Annals of statistics*, pages 94–128, 1999.

[11] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[12] J Diebolt, E Ip, and Ingram Olkin. A stochastic em algorithm for approximating the maximum likelihood estimate. *Markov chain Monte Carlo in practice. Chapman and Hall, Dordrect, The Netherlands*, 1996.

[13] Randal Douc, Aurelien Garivier, Eric Moulines, and Jimmy Olsson. On the forward filtering backward smoothing particle approximations of the smoothing distribution in general state spaces models. *arXiv preprint arXiv:0904.0316*, 2009.

[14] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[15] Denis Dreano, Pierre Tandeo, Manuel Pulido, Boujemaa Ait-El-Fquih, Thierry Chonavel, and Ibrahim Hoteit. Estimating model-error covariances in nonlinear state-space models using kalman smoothing and the expectation–maximization algorithm. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1877–1885, 2017.

[16] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

[17] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[18] Geir Evensen and Peter Jan Van Leeuwen. An ensemble kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

[19] Jianqing Fan. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Routledge, 2018.

[20] Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

[21] Simon Haykin. *Kalman filtering and neural networks*, volume 47. John Wiley & Sons, 2004.

[22] Diederich Hinrichsen and Anthony J Pritchard. *Mathematical systems theory I: modelling, state space analysis, stability and robustness*, volume 48. Springer Berlin, 2005.

[23] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[24] Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan Maciejowski, Nicolas Chopin, et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351, 2015.

[25] François Le Gland, Valérie Monbet, and Vu Duc Tran. Large sample asymptotics for the ensemble kalman filter. In Dan Crisan and Boris Rozovskii, editors, *Handbook on Nonlinear Filtering*, chapter 22, pages 598–631. Oxford University Press, Oxford, 2009.

[26] Redouane Lguensat, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet. The analog data assimilation. *Monthly Weather Review*, 145(10):4093–4107, 2017.

[27] Fredrik Lindsten. An efficient stochastic approximation em algorithm using conditional particle filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6274–6278. IEEE, 2013.

[28] Fredrik Lindsten, Michael I Jordan, and Thomas B Schön. Particle gibbs with ancestor sampling. *The Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

[29] Fredrik Lindsten and Thomas B Schön. On the use of backward simulation in particle markov chain monte carlo methods. *arXiv preprint arXiv:1110.2873*, 2011.

[30] Fredrik Lindsten, Thomas B Schön, et al. Backward simulation methods for monte carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143, 2013.

[31] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[32] Katsuhiko Ogata. *Discrete-time control systems*, volume 2. Prentice Hall Englewood Cliffs, NJ, 1995.

[33] Jimmy Olsson, Olivier Cappé, Randal Douc, Eric Moulines, et al. Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

[34] Herbert E Rauch, CT Striebel, and F Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[35] Guillermo Rodriguez. Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics. *IEEE Journal on Robotics and Automation*, 3(6):624–639, 1987.

[36] Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

[37] Jonathan R Stroud and Thomas Bengtsson. Sequential state and variance estimation within the ensemble kalman filter. *Monthly Weather Review*, 135(9):3194–3208, 2007.

[38] Jonathan R Stroud, Matthias Katzfuss, and Christopher K Wikle. A bayesian adaptive ensemble kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, 146(1):373–386, 2018.

[39] Andreas Svensson and Fredrik Lindsten. Learning dynamical systems with particle stochastic approximation em. *arXiv preprint arXiv:1806.09548*, 2018.

[40] Pierre Tandeo, Pierre Ailliot, Ronan Fablet, Juan Ruiz, François Rousseau, and Bertrand Chapron. The analog ensemble kalman filter and smoother. In *CI 2014: 4th International Workshop on Climate Informatics*, 2014.

[41] Pierre Tandeo, Pierre Ailliot, Juan Ruiz, Alexis Hannart, Bertrand Chapron, Anne Cuzol, Valérie Monbet, Robert Easton, and Ronan Fablet. Combining analog method and ensemble data assimilation: application to the lorenz-63 chaotic system. In *Machine Learning and Data Mining Approaches to Climate Science*, pages 3–12. Springer, 2015.

[42] Genta Ueno and Nagatomo Nakamura. Bayesian estimation of the observation-error covariance matrix in ensemble-based filters. *Quarterly Journal of the Royal Meteorological Society*, 142(698):2055–2080, 2016.

[43] Nick Whiteley. Discussion on particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):306–307, 2010.

[44] Mengbin Zhu, Peter J Van Leeuwen, and Weimin Zhang. Estimating model error covariances using particle filters. *Quarterly Journal of the Royal Meteorological Society*, 2017.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

**Titre: Méthodologies non-paramétriques pour la reconstruction et l'estimation dans les modèles d'états non linéaires**

**Mots clés:** estimation non-paramétrique, les algorithmes EM, régression locale, conditional particle filtering, lissage, modèles statistiques spatio- temporels non-linéaires

**Resumé :** Le volume des données disponibles permettant de décrire l'environnement, en particulier l'atmosphère et les océans, s'est accru à un rythme exponentiel. Ces données regroupent des observations et des sorties de modèles numériques. Les observations (satellite, in situ, etc.) sont généralement précises mais sujettes à des erreurs de mesure et disponibles avec un échantillonnage spatio-temporel irrégulier qui rend leur exploitation directe difficile. L'amélioration de la compréhension des processus physiques associée à la plus grande capacité des ordinateurs ont permis des avancés importantes dans la qualité des modèles numériques. Les solutions obtenues ne sont cependant pas encore de qualité suffisante pour certaines applications et ces méthodes demeurent lourdes à mettre en oeuvre. Filtrage et lissage (les méthodes d'assimilation de données séquentielles en pratique) sont développés pour abonder ces problèmes. Ils sont généralement formalisées sous la forme d'un modèle espace-état, dans lequel on distingue le modèle dynamique qui décrit l'évolution du processus physique (état), et le modèle d'observation qui décrit le lien entre le processus physique et les observations disponibles.

Dans cette thèse, nous abordons trois problèmes liés à l'inférence statistique pour les modèles espace-états: reconstruction de l'état, estimation des paramètres et remplacement du modèle dynamique par un émulateur construit à partir de données. Pour le premier problème, nous introduirons tout d'abord un algorithme de lissage original qui combine les algorithmes Conditional Particle Filter (CPF) et Backward Simulation (BS). Cet algorithme CPF-BS permet une exploration efficace de l'état de la variable physique, en raffinant séquentiellement l'exploration autour des trajectoires qui respectent le mieux les contraintes du modèle dynamique et des observations. Nous montrerons sur plusieurs modèles jouets que, à temps de calcul égal, l'algorithme CPF-BS donne de meilleurs résultats que les autres CPF et l'algorithme EnKS stochastique qui est couramment utilisé dans les applications opérationnelles. Nous aborderons ensuite le problème de l'estimation des paramètres inconnus dans les modèles espace-état. L'algorithme le plus usuel en statistique pour estimer les paramètres d'un modèle espace-état est l'algorithme EM qui permet de calculer itérativement une approximation numérique des estimateurs du maximum de vraisemblance. Nous montrerons que les algorithmes EM et CPF-BS peuvent être combinés efficacement pour estimer les paramètres d'un modèle jouet. Pour certaines applications, le modèle dynamique est inconnu ou très coûteux à résoudre numériquement mais des observations ou des simulations sont disponibles. Il est alors possible de reconstruire l'état conditionnellement aux observations en utilisant des algorithmes de filtrage/lissage dans lesquels le modèle dynamique est remplacé par un émulateur statistique construit à partir des observations. Nous montrerons que les algorithmes EM et CPF-BS peuvent être adaptés dans ce cadre et permettent d'estimer de manière non-paramétrique le modèle dynamique de l'état à partir d'observations bruitées. Pour certaines applications, le modèle dynamique est inconnu ou très coûteux à résoudre numériquement mais des observations ou des simulations sont disponibles. Il est alors possible de reconstruire l'état conditionnellement aux observations en utilisant des algorithmes de filtrage/lissage dans lesquels le modèle dynamique est remplacé par un émulateur statistique construit à partir des observations. Nous montrerons que les algorithmes EM et CPF-BS peuvent être adaptés dans ce cadre et permettent d'estimer de manière non-paramétrique le modèle dynamique de l'état à partir d'observations bruitées. Enfin, les algorithmes proposés sont appliqués pour imputer les données de vent (produit par Météo France).

**Title: Non-parametric methodologies for reconstruction and estimation in nonlinear state-space models**

**Keywords:** non-parametric estimation, EM algorithms, local regression, conditional particle filtering, smoothing, nonlinear state-space models

**Abstract :** The amount of both observational and model-simulated data within the environmental, climate and ocean sciences has grown at an accelerating rate. Observational (e.g. satellite, in-situ...) data are generally accurate but still subject to observational errors and available with a complicated spatio-temporal sampling. Increasing computer power and understandings of physical processes have permitted to advance in models accuracy and resolution but purely model driven solutions may still not be accurate enough. Filtering and smoothing (or sequential data assimilation methods) have developed to tackle the issues. Their contexts are usually formalized under the form of a space-state model including the dynamical model which describes the evolution of the physical process (state), and the observation model which describes the link between the physical process and the available observations.

In this thesis, we tackle three problems related to statistical inference for nonlinear state-space models: state reconstruction, parameter estimation and replacement of the dynamic model by an emulator constructed from data. For the first problem, we will introduce an original smoothing algorithm which combines the Conditional Particle Filter (CPF) and Backward Simulation (BS) algorithms. This CPF-BS algorithm allows for efficient exploration of the state of the physical variable, sequentially refining exploration around trajectories which best meet the constraints of the dynamic model and observations. We will show on several toy models that, at the same computation time, the CPF-BS algorithm gives better results than the other CPF algorithms and the stochastic EnKS algorithm which is commonly used in real applications. We will then discuss the problem of estimating unknown parameters in state-space models. The most common statistical algorithm for estimating the parameters of a space-state model is based on EM algorithm, which makes it possible to iteratively compute a numerical approximation of the maximum likelihood estimators. We will show that the EM and CPF-BS algorithms can be combined to effectively estimate the parameters in toy models. In some applications, the dynamical model is unknown or very expensive to solve numerically but observations or simulations are available. It is thence possible to reconstruct the state conditionally to the observations by using filtering/smoothing algorithms in which the dynamical model is replaced by a statistical emulator constructed from the observations. We will show that the EM and CPF-BS algorithms can be adapted in this framework and allow to provide non-parametric estimation of the dynamic model of the state from noisy observations. Finally the proposed algorithms are applied to impute wind data (produced by Météo France).