



An algorithm for non-parametric estimation in state-space models

Thi Tuyet Trang Chau, Pierre Ailliot, Valérie Monbet

► To cite this version:

Thi Tuyet Trang Chau, Pierre Ailliot, Valérie Monbet. An algorithm for non-parametric estimation in state-space models. Computational Statistics and Data Analysis, 2021, 153, pp.107062. 10.1016/j.csda.2020.107062 . hal-02975833

HAL Id: hal-02975833

<https://hal.science/hal-02975833>

Submitted on 4 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



An algorithm for non-parametric estimation in state–space models

Thi Tuyet Trang Chau^{a,*}, Pierre Ailliot^b, Valérie Monbet^a

^a IRMAR-INRIA, University of Rennes, Rennes, France

^b Univ Brest, CNRS, LMBA - UMR 6205, Brest, France

ARTICLE INFO

Article history:

Received 19 September 2018

Received in revised form 7 June 2020

Accepted 4 August 2020

Available online 13 August 2020

Keywords:

State–space models

Non-parametric statistics

SEM algorithm

Local linear regression

Conditional particle filter

ABSTRACT

State–space models are ubiquitous in the statistical literature since they provide a flexible and interpretable framework for analyzing many time series. In most practical applications, the state–space model is specified through a parametric model. However, the specification of such a parametric model may require an important modeling effort or may lead to models which are not flexible enough to reproduce all the complexity of the phenomenon of interest. In such situations, an appealing alternative consists in inferring the state–space model directly from the data using a non-parametric framework. The recent developments of powerful simulation techniques have permitted to improve the statistical inference for parametric state–space models. It is proposed to combine two of these techniques, namely the Stochastic Expectation–Maximization (SEM) algorithm and Sequential Monte Carlo (SMC) approaches, for non-parametric estimation in state–space models. The performance of the proposed algorithm is assessed through simulations on toy models and an application to environmental data is discussed.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

State–space models (SSMs) provide a natural framework to study time series with observational noise in environment, economy, computer sciences, etc. They have a wide range of applications in data assimilation, system identification, model control, change detection, missing-data imputation (see e.g. [Durbin and Koopman, 2012](#)). The general SSM which is considered in this paper is defined through the following equations,

$$\begin{cases} X_t = m(X_{t-1}, Z_t) + \eta_t, & [\text{hidden}] \quad (\text{a}) \\ Y_t = H_t(X_t) + \epsilon_t, & [\text{observed}] \quad (\text{b}). \end{cases} \quad (1)$$

The dynamical model m describes the time evolution of the latent process $\{X_t\}$. It may depend on some covariates (or control) denoted as $\{Z_t\}$. The operator H_t links the latent state to the observations $\{Y_t\}$. The random sequences $\{\eta_t\}$ and $\{\epsilon_t\}$ model respectively the random components in the dynamical model and the observational error. Throughout this paper, we make the classical assumptions that H_t is known (typically $H_t(x) = x$) and that $\{\eta_t\}$ and $\{\epsilon_t\}$ are independent sequences of Gaussian distributions such that $\eta_t \stackrel{iid}{\sim} \mathcal{N}(0, Q_t(\theta))$ and $\epsilon_t \stackrel{iid}{\sim} \mathcal{N}(0, R_t(\theta))$ where θ denotes the parameters involved in the parameterization of the covariance matrices.

In this paper, 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

* Correspondence to: LSCE, CEA Saclay, 91191 Gif-sur-Yvette cedex, France.

E-mail addresses: trang.chau@lsce.ipsl.fr (T.T.T. Chau), pierre.ailliot@univ-brest.fr (P. Ailliot), valerie.monbet@univ-rennes1.fr (V. Monbet).

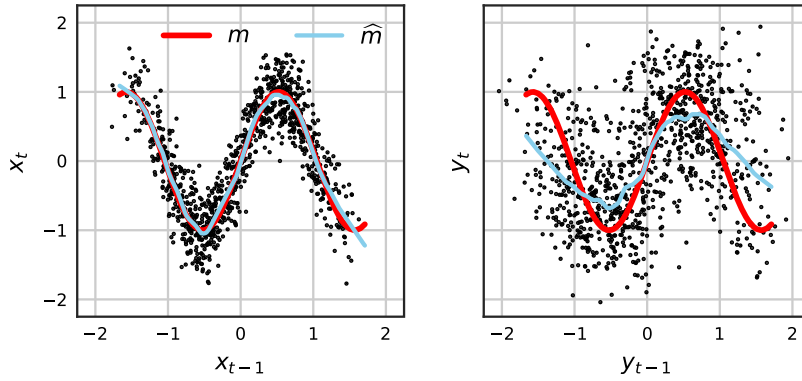


Fig. 1. Scatter plots of (x_{t-1}, x_t) (left) and (y_{t-1}, y_t) (right) for a sequence simulated with the SSM defined by (2) and corresponding LLR estimate \hat{m} of m .

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, non-parametric approaches have been proposed to estimate m .

Such **non-parametric SSMs** were originally introduced in Tandeo et al. (2015) and Lguensat et al. (2017) for data assimilation in oceanography and meteorology. In these application fields, a huge amount of historical data sets recorded using remote and in situ sensors or obtained through numerical simulations are now available and this promotes the development of data-driven approaches. It was proposed to build a non-parametric estimate \hat{m} of m using the available observations and plug this non-parametric estimate into usual filtering and smoothing algorithms to reconstruct the latent space $X_{1:T} = (X_1, \dots, X_T)$ given observations $y_{1:T} = (y_1, \dots, y_T)$. Numerical experiments on toy models show 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 . Some applications to real data are discussed in Fablet et al. (2017b).

Various non-parametric estimation methods have been considered to build surrogate nonlinear dynamical models in oceanography and meteorology. The more natural one is probably the nearest neighbors method known as the Nadaraya-Watson approach in statistics (Fan and Yao, 2008) and analog methods in meteorology (Yiou, 2014). In Lguensat et al. (2017), better results were obtained with a slightly more sophisticated estimator known as local linear regression (LLR) in statistics (Cleveland and Devlin, 1988) and constructed analogs in meteorology (Tippett and DelSole, 2013). More recently, it has been proposed to use other machine learning (ML) tools such as deep learning (see Bocquet et al., 2019) or sparse regression (see Brunton et al., 2017) to better handle high dimensional data and mimic the behavior of numerical methods used to approximate the solutions of physical models.

In the above mentioned references, it is generally assumed that a sequence $x_{1:T} = (x_1, \dots, x_T)$ of “perfect” observations with no observational error is available to estimate the dynamical model m . However, in practical applications, only a sequence $y_{1:T}$ of the process $\{Y_t\}$ with observational errors is given to fit the model. The main contribution of this paper is to propose a method to build non-parametric estimate of m in this context. A simple approach would consist in “forgetting” the observational errors and computing directly a non-parametric estimate based on the sequence $y_{1:T}$ but this may lead to biased estimates. This is illustrated in Fig. 1 obtained with the toy SSM defined as

$$\begin{cases} X_t = \sin(3X_{t-1}) + \eta_t, & \eta_t \stackrel{iid}{\sim} \mathcal{N}(0, Q) \\ Y_t = X_t + \epsilon_t, & \epsilon_t \stackrel{iid}{\sim} \mathcal{N}(0, R) \end{cases} \quad (2)$$

with $Q = R = 0.1$. The left plot shows the scatter plot (x_{t-1}, x_t) for a simulated sequence (x_1, \dots, x_T) of the latent process $\{X_t\}$ and the corresponding non-parametric estimate \hat{m} based on this sample which is reasonably close to m . The right plot shows the scatter plot (y_{t-1}, y_t) of the corresponding sequence with observation noise. 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. It leads to a biased estimate of m when the non-parametric estimation method is computed on this sequence. 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 Carroll et al., 2006). One of the classical approaches to reduce the bias is to introduce instrumental variables which help to get information about the observational error. This approach has been adapted to linear first order auto-regressive models in Meijer et al. (2013) and further studied in Lee et al. (2017). Carroll et al. (2006) give 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 have not been generalized for time series.

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 θ and a non-parametric component for the dynamical

model m . The Expectation–Maximization (EM) algorithm and its variants (see e.g. [Dempster et al., 1977](#)) are often used to fit parametric models with a latent component. The E-step of the EM algorithm consists in computing the smoothing distribution, i.e. the conditional distributions of the latent space given the observations. The smoothing distributions can generally not be computed analytically in nonlinear state–space models. However, the recent development of powerful simulation techniques, known as Sequential Monte Carlo (SMC) methods, permits to produce nowadays accurate approximations of the smoothing distributions at a reasonable computational cost and this allows to perform statistical inference in nonlinear state–space models (see [Kantas et al., 2015](#), for a recent review). In [Chau et al. \(2018\)](#), it was proposed to use the Conditioning Particle Filter-Backward Simulation (CPF-BS) algorithm presented for instance in [Lindsten et al. \(2013\)](#) in the E-step of the EM algorithm. It was found, through numerical experiments, that the combination of CPF-BS algorithm and EM recursions leads to an efficient numerical procedure to estimate the parameters of parametric SSMs.

This paper discusses an extension of the algorithms proposed in [Chau et al. \(2018\)](#) to non-parametric SSMs where the non-parametric estimate of m is updated at each iteration of the EM recursions using the trajectories simulated from the smoothing algorithms. It permits to correct sequentially the bias in the estimate of m due to observational errors. This method can also be interpreted as a generalization of the Bayesian approach of [Carroll et al. \(2006\)](#) for time series.

The paper is organized into the following sections. The estimation of the parametric component using EM recursions is introduced in Section 2. Then this algorithm is extended to estimate both the parametric and non-parametric components in Section 3. Simulation results obtained on a toy model (Lorenz-63) are presented in Section 4. Then, Section 5 discusses an application to oceanographic data, where the algorithms are used to impute missing wave data given noisy observations. The paper ends with some concluding remarks in Section 6. All the codes of the proposed approach used for numerical experiments in this paper are available on <https://github.com/tchau218/npSEM>.

2. SEM algorithm for parametric estimation in SSMs

In this section, it is assumed that the dynamical model m is known or that a surrogate model has already been fitted and the estimation of the unknown parameter θ , given a sequence $y_{1:T}$ of noisy observations, is discussed. The notation \mathfrak{M} stands for the true dynamical model m if it is known, or for the surrogate model otherwise. Remark that the covariate Z_t which appears in (1a) is omitted in Sections 2–4 for the sake of simplification.

The EM algorithm is probably the most usual algorithm to perform maximum likelihood estimation in models with latent variables including SSMs. It is an iterative algorithm where, at each iteration $r \geq 1$, the parameter value $\hat{\theta}_{r-1}$ is updated through the following steps.

- **E-step:** compute the smoothing distribution $p(x_{0:T}|y_{1:T}; \hat{\theta}_{r-1})$ defined as the conditional distribution of the latent sequence $X_{0:T}$ given the sequence of observations $y_{1:T}$ and the current parameter value $\hat{\theta}_{r-1}$.
- **M-step:** maximize the intermediate function

$$I(\theta|\hat{\theta}_{r-1}, \mathfrak{M}) \triangleq \int \log p(x_{0:T}, y_{1:T}; \theta, \mathfrak{M}) p(x_{0:T}|y_{1:T}; \hat{\theta}_{r-1}) dx_{0:T} \quad (3)$$

obtained by integrating the complete log-likelihood function

$$\log p(x_{0:T}, y_{1:T}; \theta, \mathfrak{M}) = \log p(x_0) + \sum_{t=1}^T \log p(x_t|x_{t-1}; Q_t(\theta), \mathfrak{M}) + \sum_{t=1}^T \log p(y_t|x_t; R_t(\theta)) \quad (4)$$

over the smoothing distribution computed in the E-step where the initial distribution $p(x_0)$ is assumed to be known. $p(x_t|x_{t-1}; Q_t(\theta), \mathfrak{M})$ and $p(y_t|x_t; R_t(\theta))$ denote respectively the transition kernel of the Markov process $\{X_t\}$ defined by (1a) and the conditional probability distribution function of Y_t given $X_t = x_t$ associated to (1b).

Finally, the parameter value is updated as

$$\hat{\theta}_r = \arg \max_{\theta} I(\theta|\hat{\theta}_{r-1}, \mathfrak{M}). \quad (5)$$

For nonlinear (non-Gaussian) SSMs, the smoothing distributions do not have any tractable analytical expression. However, sequential Monte Carlo (SMC) algorithms (see [Cappé et al., 2007](#); [Godsill et al., 2004](#), for instance) allow to generate sequences of these conditional distributions. They provide samples (particles) $\{\tilde{x}_{0:T,r}^{(i)}\}_{i=1:N}$ to approximate the smoothing distribution $p(\cdot|y_{1:T}; \hat{\theta}_{r-1})$ with the empirical distribution

$$\hat{p}_r(dx_{0:T}|y_{1:T}) = \frac{1}{N} \sum_{i=1}^N \delta_{\tilde{x}_{0:T,r}^{(i)}}(dx_{0:T}), \quad (6)$$

where δ_x denotes the point mass at x . Replacing the true smoothing distribution by this empirical distribution in (3) to estimate the intermediate function I of the EM algorithm leads to the so-called Stochastic EM (SEM) algorithm.

One of the key points of SEM algorithms is to compute an approximation of the intermediate function I at a reasonable computational cost. If the number of particles N is large, then the law of large numbers implies that

$$\widehat{I}(\theta|\widehat{\theta}_{r-1}, \mathfrak{M}) \triangleq \frac{1}{N} \sum_{i=1}^N \log p(\tilde{x}_{0:T,r}^{(i)}; y_{1:T}; \theta, \mathfrak{M})$$

is a good approximation of $I(\theta|\widehat{\theta}_{r-1}, \mathfrak{M})$ and the SEM algorithm is close to the EM algorithm. Various extensions of the SEM algorithm have been proposed to reduce the size of simulated samples N and save computational time (Wei and Tanner, 1990; Delyon et al., 1999). The SEM algorithms and their variants using particle filters (see Kantas et al., 2015) suffer from another issue. In order to get samples which approximately follow the smoothing distribution a large number of particles are typically required. Since the smoothing algorithm has to be run at each iteration of the EM algorithm, this may lead to prohibitive computational costs (see Fearnhead and Künsch, 2018, for a recent review).

Kalman-based algorithms, such as the Ensemble Kalman Smoother (EnKS), are traditionally used in the data assimilation community since they generally provide good approximations of the smoothing distributions with a low number of particles (see Carrassi et al., 2018). However, they are based on Gaussian approximations which may not be suitable for nonlinear SSMs.

Conditional SMC (CSMC) samplers, which are based on combinations of SMC and Markov Chain Monte Carlo (MCMC) approaches, have been developed as alternatives to particle filters and Kalman-based algorithms. The first CSMC samplers, called Conditional Particle Filters (CPFs), have been introduced by Andrieu et al. (2010) and Lindsten et al. (2012) and were combined with the EM algorithm in Lindsten et al. (2013). The CPF algorithms simulate samples of $x_{0:T}$ conditionally on the current value of the parameter θ and a trajectory in the latent state-space referred to as the conditioning sequence. The conditioning sequence is updated sequentially and this builds a Markov chain which has the exact smoothing distribution $p(dx_{0:T}|y_{1:T}; \theta)$ as an invariant distribution (see Svensson et al., 2015; Chau et al., 2018, for numerical illustrations).

Nevertheless, as many sequential smoothing algorithms, when the length T of the observed sequence is large, the CPF algorithms suffer from “sample impoverishment” with all the trajectories sharing the same ancestors. A way to reduce impoverishment is to run a Backward Simulation (BS) algorithm after the CPF one. Backward simulation, proposed initially in Godsill et al. (2004), is a natural technique to simulate the smoothing distribution given the (forward) filter outputs (see Lindsten et al., 2013). This leads to the Conditional Particle Filter-Backward Simulation (CPF-BS) sampler (see Algorithm 3 in Appendix A). Recently, Chau et al. (2018) proposed to use the CPF-BS smoothing algorithm in conjunction with the SEM algorithm (see Algorithm 1). The authors found experimentally that the method outperforms several existing EM algorithms in terms of both state reconstruction and parameter estimation, using low computational resources.

Algorithm 1: SEM algorithm for parametric SSMs [SEM(\mathfrak{M})]

Initialization: choose an initial parameter value $\widehat{\theta}_0$ and an initial conditioning sequence $x_{0:T}^*$.

For $r \geq 1$,

(1) E-step: generate N trajectories $\{\tilde{x}_{0:T,r}^{(i)}\}_{i=1:N}$ of the smoothing distribution using the CPF-BS algorithm (3) with parameter value $\widehat{\theta}_{r-1}$, dynamical model \mathfrak{M} , conditioning sequence $x_{0:T}^* = \tilde{x}_{0:T,r-1}^{(1)}$, and observations $y_{1:T}$,

(2) M-step: update the parameter value

$$\widehat{\theta}_r = \arg \max_{\theta} \widehat{I}(\theta|\widehat{\theta}_{r-1}, \mathfrak{M}),$$

end.

3. Non-parametric estimation in SSMs

In the previous section, it is assumed that the true dynamical model m is known or that a surrogate model is available but this may be unrealistic for some applications. In this section, the joint estimation of θ and non-parametric estimation of m from a sequence $y_{1:T}$ with observational error is discussed.

Following the numerical results presented in Lguensat et al. (2017), Local Linear Regression (LLR) is used to build a non-parametric estimate of m . The idea of LLR is to locally approximate m by a 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 where the weights are defined using a kernel. In this study the tricube kernel is used as in Cleveland and Devlin (1988). This kernel has a compact support and is smooth at its boundary. Throughout the paper, LLR is performed based on the k -nearest neighborhood of x (Altman, 1992). In this case, the support of the kernel is defined as the smallest rectangular area which contains the k nearest neighbors and the kernel bandwidth adapts to the density of points in the neighborhood of x .

As mentioned in the introduction, applying the LLR estimation method on a sequence $y_{1:T}$ with observational errors leads to a biased estimate for m (see Fig. 1). In order to reduce sequentially the bias induced by the observation noise, it is proposed in Algorithm 2 to update the non-parametric estimate of m at each iteration of the SEM algorithm. Algorithm 2 is similar to Algorithm 1, with the exception that the non-parametric estimate \widehat{m}_r of the dynamical model is updated at each iteration r of the EM algorithm using LLR on the trajectories of the smoothing distribution simulated in the E-step.

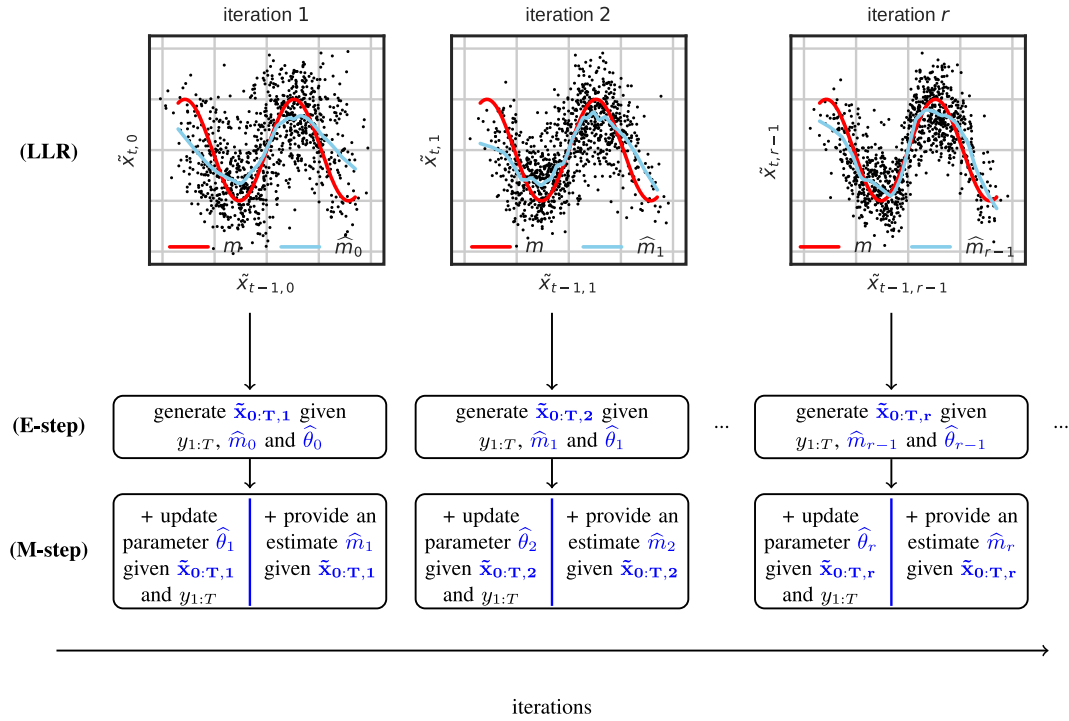


Fig. 2. An illustration of Algorithm 2 (npSEM) on the sinus SSM (2).

This estimate is then used in the smoothing algorithm and in the complete log-likelihood function (4) which appears in the definition of the intermediate function of the EM algorithm.

Algorithm 2: SEM-like algorithm for non-parametric SSMs [npSEM]

Initialization: choose an initial parameter value $\hat{\theta}_0$, an estimate of the dynamical model \hat{m}_0 , and a conditioning sequence $\tilde{x}_{0:T}^*$.

For $r \geq 1$,

(1) E-step: generate N trajectories $\{\tilde{x}_{0:T,r}^{(i)}\}_{i=1:N}$ using a sequential Monte Carlo smoothing algorithm with parameter value $\hat{\theta}_{r-1}$, dynamical model \hat{m}_{r-1} , conditioning sequence $\tilde{x}_{0:T}^* = \tilde{x}_{0:T,r-1}^{(1)}$ and observations $y_{1:T}$,

(2) M-step:

- i. Parameter update: compute the parameter value

$$\hat{\theta}_r = \arg \max_{\theta} \hat{I}(\theta | \hat{\theta}_{r-1}, \hat{m}_{r-1}),$$

- ii. Catalog update: compute an LLR estimate \hat{m}_r of m based on the 'updated catalog' $\{\tilde{x}_{0:T,r}^{(i)}\}_{i=1:N}$,

end.

The name "SEM-like" algorithm is used to highlight that the proposed algorithm shares similarities with the SEM algorithm for parametric estimation in SSMs. Remark that it is not an SEM algorithm because the M-step is not only composed of a likelihood maximization for θ , as in the usual EM algorithm, but also of a 'catalog update' for m . This is illustrated in Fig. 2 using the sinus model (2). At each iteration, the estimate $(\hat{m}_r)_{r \geq 0}$ of the dynamical model m is updated using LLR on the sample of the smoothing distribution simulated in the E-step. It allows to correct sequentially the bias in the non-parametric estimate of m which is induced by the observation error. Such EM-like and SEM-like algorithms have been proposed in the literature for fitting mixture models (Zhang, 2002; Benaglia et al., 2009) with non-parametric estimates of the distributions in the different components of the mixture. The spirit of Algorithm 2 is also close to the one of the iterative global/local estimation (IGLE) algorithm of Young and Hunter (2010) for estimation of mixture models with mixing proportions depending on covariates.

Remark that the smoothing sample $\{\tilde{x}_{t,r-1}^{(i)}\}_{i \in \{1, \dots, N\}}$ at time t depends on the observation y_t at the same time, and that over-fitting may occur if this sample is used to build the non-parametric estimate \hat{m}_{r-1} and propagate the particles at time t in the smoothing algorithm at iteration r of the EM algorithm. This over-fitting was confirmed using numerical experiments. To tackle this issue, at each iteration r and for each time t , $\hat{m}_r(\tilde{x}_{t-1,r}^{(i)})$ is estimated using LLR based on the subsamples $\{\tilde{x}_{0:(t-\ell) \cup (t+\ell):T,r-1}^{(i)}\}_{i \in \{1, \dots, N\}}$ where the smoothing sequences $\{\tilde{x}_{(t-\ell+1):(t+\ell-1),r-1}^{(i)}\}_{i \in \{1, \dots, N\}}$ are removed from the learning sequence. The lag ℓ is chosen as a priori such that the correlation between $Y_{t-\ell}$ and Y_t is low. At each iteration, the LLR estimate of m is updated and the number k of nearest neighbors needs to be chosen. Cross-validation technique is used to select an optimal value of k .

The numerical complexity of Algorithm 2 is mainly linked to the nearest neighbor search which has to be performed for the LLR estimation. At each iteration r , the nearest neighbor search is repeated for each discrete time $t \in \{1, \dots, T\}$ and for each particle $i \in \{1, \dots, N\}$, thus NT searches are performed. Furthermore, when the catalog is updated the cross-validation needs to be performed to update the optimal number of neighbors and this adds more nearest neighbor searches. The nearest neighbor search is carried out by using the *introsselect* algorithm (Musser, 1997) which has a complexity of $O(M)$ in the best cases and $O(M \log M)$ in the worst cases. Here, M denotes the size of the learning data set. For the univariate SSM (2), 56.5 min of CPU time are necessary to run 100 iterations of the npSEM algorithm ($N = 10, T = 1000$) using Python on a computer with a 3-GHz CPU and 128-GB of RAM.

4. Simulation results

In the previous sections, the sinus SSM (2) was used as an illustrative example. Many simulation experiments were performed using this model and the obtained results were generally satisfactory. Some results are reported in a Supplementary Material document for the sake of brevity. This section focuses on the more challenging 3-dimensional Lorenz model (see in Lorenz, 1963) which is one of the favorite toy models in data assimilation since it is a sophisticated (nonlinear, non-periodic, chaotic) but low-dimensional dynamical system (Lguensat et al., 2017; Bocquet et al., 2019). The considered Lorenz-63 (L63) SSM on \mathbb{R}^3 is defined as

$$\begin{cases} X_t = m(X_{t-1}) + \eta_t, & \eta_t \stackrel{iid}{\sim} \mathcal{N}(0, Q) \\ Y_t = X_t + \epsilon_t, & \epsilon_t \stackrel{iid}{\sim} \mathcal{N}(0, R). \end{cases} \quad (7)$$

Covariance matrices in the above model are assumed to be diagonal and proportional to the identity matrix I_3 of dimension 3, such that $Q = \sigma_Q^2 I_3$ and $R = \sigma_R^2 I_3$ with true parameter values $\sigma_Q^2 = 1$ and $\sigma_R^2 = 4$. The dynamical model m at any value x in \mathbb{R}^3 is computed by integrating the following differential system

$$\begin{cases} z(0) = x \\ \frac{dz(\tau)}{d\tau} = g(z(\tau)), & \tau \in [0, dt] \\ m(x) = z(dt) \end{cases} \quad (8)$$

where $g(z) = (10(z_2 - z_1), z_1(28 - z_3) - z_2, z_1 z_2 - 8/3 z_3)$, $\forall z = (z_1, z_2, z_3) \in \mathbb{R}^3$. For each time t , the system of ordinary differential equations (8) is integrated by running a Runge-Kutta scheme (order 5). The value of dt is fixed to 0.08 which corresponds to a 6-hour time step in the observation of atmospheric data and was considered in the works of Dreano et al. (2017) and Lguensat et al. (2017).

Given an observed sequence, the eight algorithms listed below are run and compared.

- ‘CPF-BS update’ corresponds to the npSEM algorithm 2.
- ‘CPF-BS no update’ corresponds to the npSEM algorithm 2 where the ‘catalog update’ step is not performed. In this algorithm, only the value of the parameter θ is updated at each iteration but the non-parametric estimate of m is not updated. Hence it also corresponds to the SEM(\mathfrak{M}) algorithm 1 with \mathfrak{M} obtained using LLR on the observed sequence $y_{1:T}$.
- ‘CPF-BS perfect’ corresponds to the SEM(\mathfrak{M}) algorithm 1 with \mathfrak{M} obtained using LLR on a perfect sequence. Remark that in order to run this algorithm, a realization of the true state $x_{1:T}$ (‘perfect catalog’) needs to be available, which is generally not the case for real applications.
- ‘CPF-BS true m ’ corresponds to the SEM(\mathfrak{M}) algorithm 1 with $\mathfrak{M} = m$. In this algorithm, the true dynamical model is assumed to be known.
- ‘EnKS update’, ‘EnKS no update’, ‘EnKS perfect’ and ‘EnKS true m ’ are the same algorithms as defined above except that the smoother used in the E-step is the EnKS instead of the CPF-BS.

Hereafter, the parametric estimations obtained from the eight algorithms are compared. Then the ability of the algorithms to reconstruct the state is evaluated with respect to the length of the learning time series and with respect to dt which is related to the strength of the nonlinearities in the observed sequence. Unless stated otherwise, observed sequences $y_{1:T}$ of length $T = 1000$ are simulated and 150 iterations of the different algorithms are run. In order to evaluate

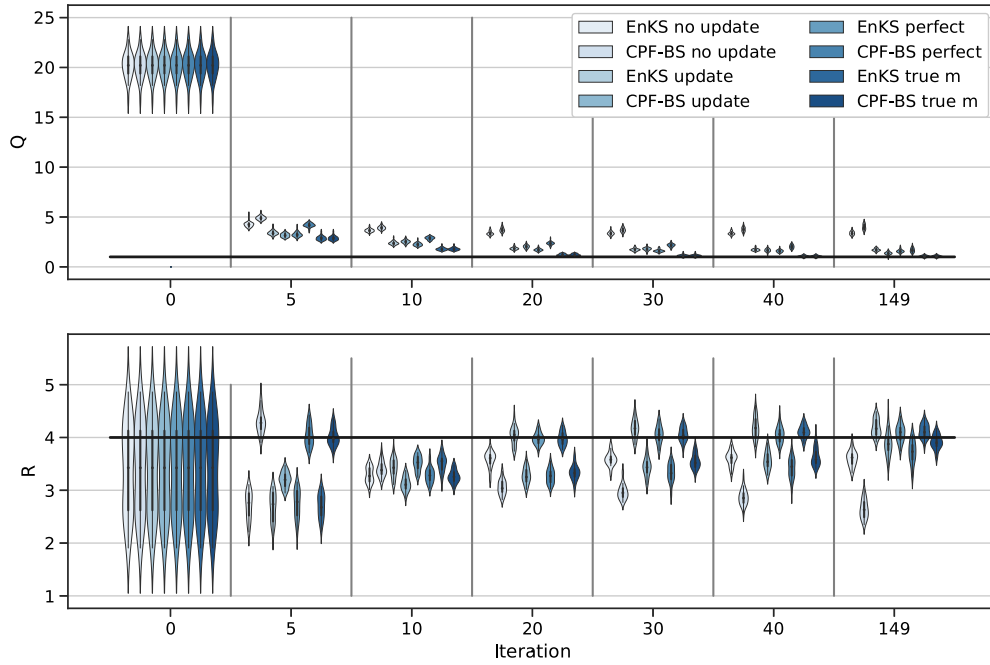


Fig. 3. Distribution of the estimation in Q and R with respect to the iteration number for the L63 model (7). The horizontal black lines materialize the true values. The vertical gray lines separate the iteration blocks. In each block the performance of the eight algorithms is displayed. Results are obtained with $T = 1000$.

the variability of the estimators, each experiment is repeated on 30 independent sequences. The number of members in the EnKS algorithm is set equal to 20. This is a classical value used in the literature (Dreano et al., 2017). The number of particles N_f of the conditional particle filter is fixed equal to $N_f = 10$ and the number of realizations N_s for the backward simulation step is fixed equal to $N_s = 5$ (see Appendix A). These values have been chosen empirically. The lag $l = 5$ is used in the npSEM algorithms. Inspired by the application, the EM algorithm coupled with a Kalman Smoother is run to initialize the algorithms (see Section 5 for more details). In practice, 100 iterations of this algorithm are performed. Then, the estimate of θ is set as the initial parameter value $\hat{\theta}_0$ of all the eight algorithms, and the mean of the smoothing distribution is taken as the initial conditioning sequence $x_{0:T}^*$ in the algorithms combined with the CPF-BS. The initial estimate \hat{m}_0 of m for the 'EnKS update' and 'CPF-BS update' algorithms is also obtained by using LLR on the mean of the smoothing distribution derived from the EM algorithm.

In Fig. 3, the distributions of the estimates of the parameters σ_Q^2 and σ_R^2 , which correspond to the diagonal coefficients of the covariance matrices Q and R , are represented with respect to the iteration number using violin plots. In general, the biases and variances of the estimates obtained using the eight algorithms have been significantly reduced after a few iterations. This is expected since the initial parameters are obtained using the EM algorithm coupled with a Kalman Smoother which approximates the L63 model by a linear Gaussian model. The 'EnKS no update' and 'CPF-BS no update' algorithms (1st and 2nd violin boxes of each block of 8) lead to the worst estimates. These algorithms based on a noisy version of the true state use a poor surrogate model in the forecast steps. In the 'EnKS update' and 'CPF-BS update' algorithms (3rd and 4th violin boxes), the catalog used to estimate the dynamic is updated at each iteration. It permits to iteratively reduce the observation errors in the catalog and reduce the estimation error. They hence provide estimates close to the ones obtained with the 'EnKS perfect', the 'CPF-BS perfect' and the two SEM algorithms using the true L63 model. The CPF-BS algorithm is expected to better capture the nonlinearities in the model compared to the EnKS algorithm and it is thus not surprising that the 'CPF-BS update' algorithm slightly outperforms the 'EnKS update' algorithm.

In order to measure the global performance of the methodology, the reconstruction error whose computation is described hereafter is considered. First, validation sequences of $x_{1:T'}$ and $y_{1:T'}$ ($T' = 1000$) are generated with the true model. These validation sequences are independent from the learning time series. At each iteration of the SEM and npSEM algorithms, the associated smoothing algorithm is run on the validation sequence $y_{1:T'}$ using the current estimates of m , Q and R . Then, the sample mean \bar{x}_t of the smoothed particles over the last 10 EM iterations is computed as an estimate of the conditional expectation of the latent state given the sequence of observations $E[X_t|y_{1:T'}]$. Finally, the Root of Mean

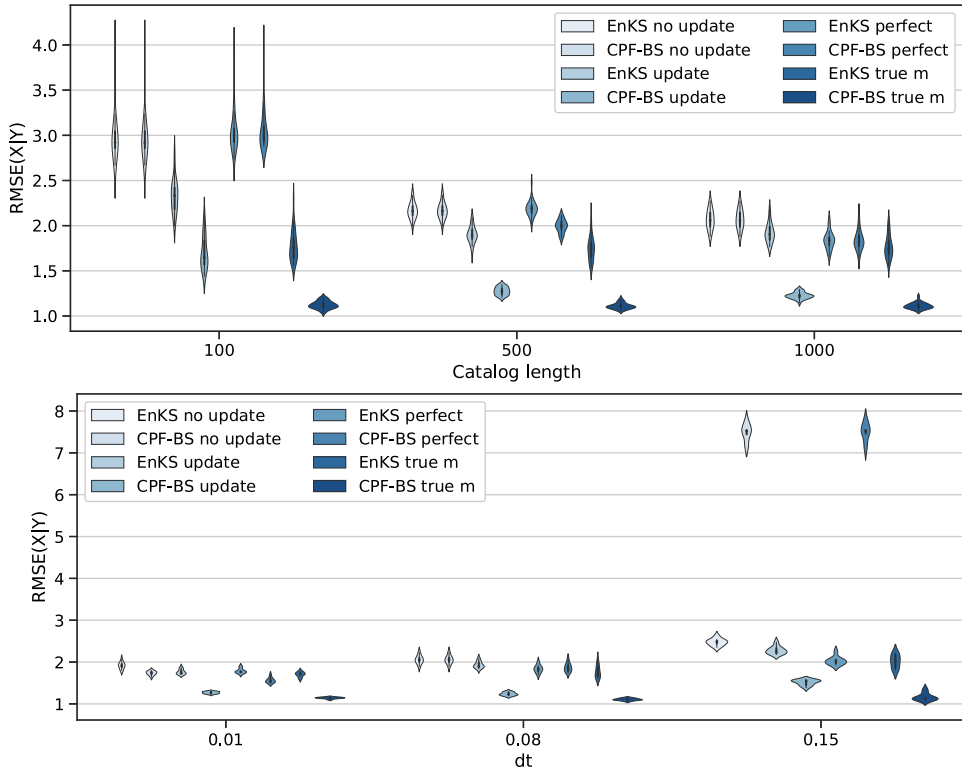


Fig. 4. Reconstruction errors ($RMSE(X|Y)$) of the SEM and npSEM algorithms with respect to different catalog length (T) and nonlinearity degree (dt) for the L63 model (7).

Square Error (RMSE),

$$RMSE(X|Y) = \sqrt{\frac{1}{T'} \sum_{t=1}^{T'} (\bar{x}_t - x_t)^2}, \quad (9)$$

is used to assess the reconstruction skill of the algorithms.

The distributions of the reconstruction errors $RMSE(X|Y)$ for the eight algorithms are shown in Fig. 4 for different lengths $T \in \{100, 500, 1000\}$ of the learning sequences. As expected, the reconstruction errors decrease when T increases and the algorithms do not seem to suffer too much from a degradation of the mixing properties of the smoothing algorithms. Again, the algorithms with catalog updates clearly outperform the algorithms with no update and the algorithms based on the CPF-BS algorithms outperform the ones based on the EnKS algorithms. It is also noteworthy that the algorithms with catalog updates clearly outperform the algorithms which use the perfect catalog $x_{1:T}$, and lead to results close to those obtained with the true model m despite being calibrated using only a noisy sequence $y_{1:T}$. In order to investigate the robustness of the proposed methodology to highly nonlinear dynamics, the bottom panel of Fig. 4 shows the reconstruction error as a function of $dt \in \{0.01, 0.08, 0.15\}$. Nonlinearities increase with dt and thus it is not surprising that the reconstruction errors generally increase with dt . Remark however that the 'CPF-BS update' algorithm again performs well even when the nonlinearities are strong.

Fig. 5 illustrates the ability of the 'CPF-BS update' Algorithm 2 to reconstruct the dynamics of the three components of the L63 model. From left to right, the scatter plots correspond to successive values at time $t - 1$ and t of the observed sequence, the true state and a realization simulated at the last iteration of the algorithm. As for the sinus model used in the introduction, the comparison of the left and middle panel shows that the observation noise significantly blurs the true dynamic. The proposed algorithm efficiently reduces the noise and the scatter plots corresponding to the catalog in the last iteration of the 'CPF-BS update' algorithm (right panel) look very similar to the ones of the true dynamic (middle panel). It suggests that proposed methodology is successful in estimating the true dynamical model m . This could be useful in applications where this model is of interest and, for example, be used to build or validate a surrogate parametric model. This is also confirmed by the time series displayed in Fig. 6 which shows that the true state x_t is generally close to the smoothing mean \bar{x}_t and generally lies in the 95% prediction interval.

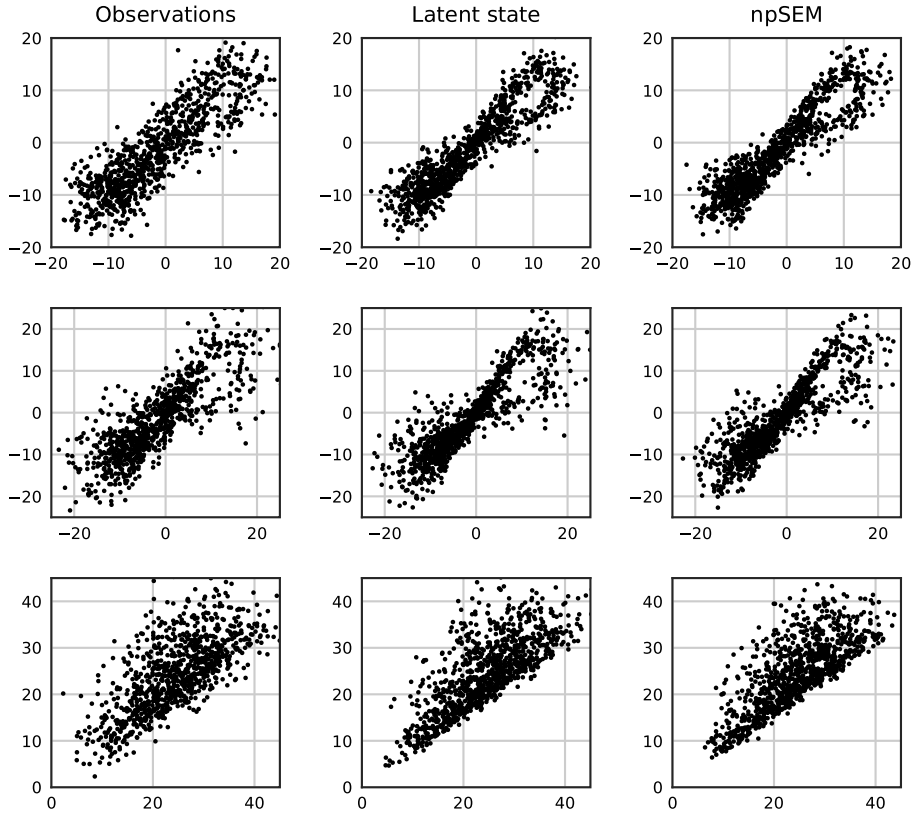


Fig. 5. Scatter-plots of (y_{t-1}, y_t) (left), (x_{t-1}, x_t) (middle) and $(\tilde{x}_{t-1}, \tilde{x}_t)$ (right) for each of the three components of the L63 model defined by (7). $\{\tilde{x}_t\}$ stands for one of realizations generated at the final iteration of the 'CPF-BS update' algorithm.

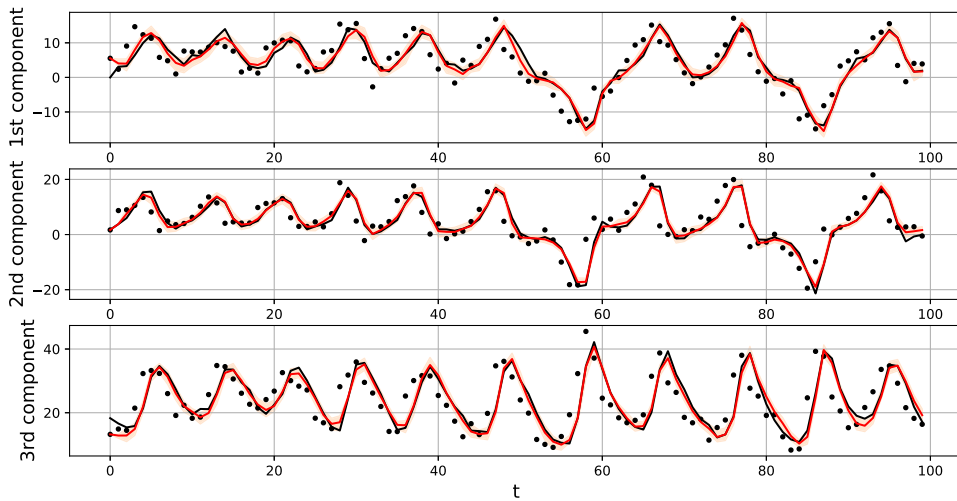


Fig. 6. Time series of L63 components (black lines) with noisy observations (black dots) and smoothing distribution of the 'CPF-BS update' algorithm. Empirical means (red lines) and 95% prediction intervals (red areas) are computed using the smoothing trajectories generated in the last 10 iterations of the 'CPF-BS update' algorithm. On this sequence the coverage probability is approximately 85% for each component. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

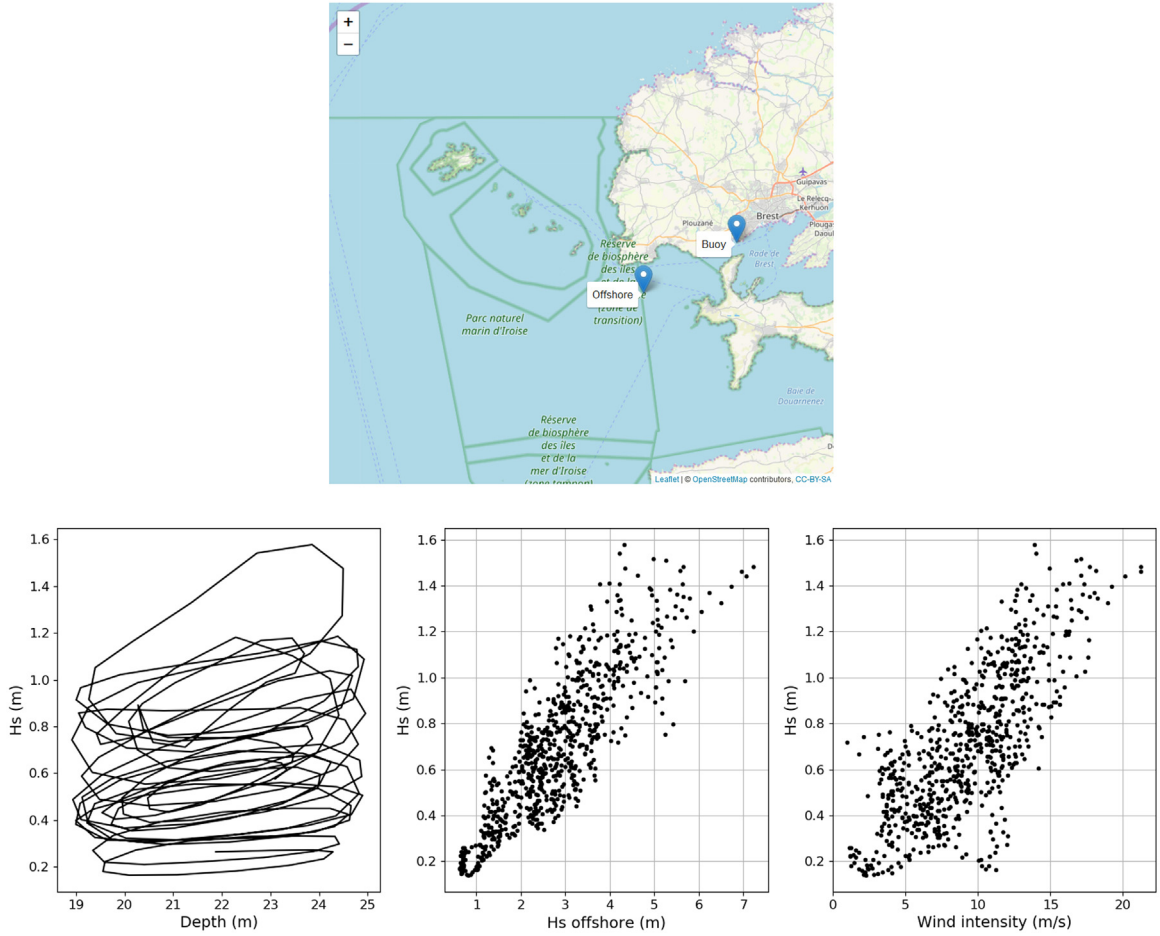


Fig. 7. Top: map with buoy and offshore location indicated. Bottom: bivariate time series of (H_s, D) , (left panel), scatter plots of (H_s^{off}, H_s) (middle panel) and (H_s, U) in January 2016, at St Anne du Porzic (France).

5. Real case study

Data imputation is a recurrent problem in many fields and particularly in oceanography since measurements are generally complicated to make over the ocean (Fablet et al., 2017b). In this section, the proposed methodology is illustrated using data from the HOMERE data set. HOMERE is an hourly hindcast data set based on the WAVEWATCH III model (version 4.11) on an unstructured spatial grid covering the English Channel and Bay of Biscay over the period 1994–2016 (Bouidière et al., 2013). Our particular interest is the significant wave height H_s . It is usually defined as 4 times the standard deviation of the surface elevation and was intended to mathematically express the height estimated by a “trained observer”. We focus on the point with geographical coordinate $(48.3563^\circ\text{N}, 4.5508^\circ\text{W})$ located in the entrance of the bay of Brest (see top panel of Fig. 7). At this location, where a buoy is located, wave conditions are influenced by the water depth D which depends on the tide, local wind speed U which creates wind sea and offshore wave conditions since the waves generally propagate to the east from the Atlantic Ocean inside the bay. Hereafter H_s^{off} denotes the significant wave height at the location shown in Fig. 7 and $Z_t = (D_t, U_t, H_{s,t}^{\text{off}}) \in \mathbb{R}^3$ the values of the covariates at time t . At the bottom of Fig. 7, the pairwise relations between H_s and its covariates are shown. The tide induces cycles where H_s grows with the depth and the link between H_s and H_s^{off} is linear in mean but exhibits heteroscedasticity.

In the numerical experiments, the H_s from the hindcast data set at the buoy location is supposed to be the true state which we wish to reconstruct. As usual in the literature, a logarithm transform is applied to $H_{s,t}$ (O’Carroll, 1984) and the result is denoted as X_t . In order to mimic the behavior of noisy observations recorded at the buoy location, an artificial time series $\{Y_t\}$ is simulated by adding a white noise to $\{X_t\}$ according to the observation equation (1b). Different levels of observation noise $\sqrt{R} \in \{0.1, 0.2, 0.5\}$ are considered since it might impact the performance of the reconstruction. Some gaps corresponding to missing data are also created in the time series $\{Y_t\}$. In Fig. 9, the time series $\{X_t\}$ (plain

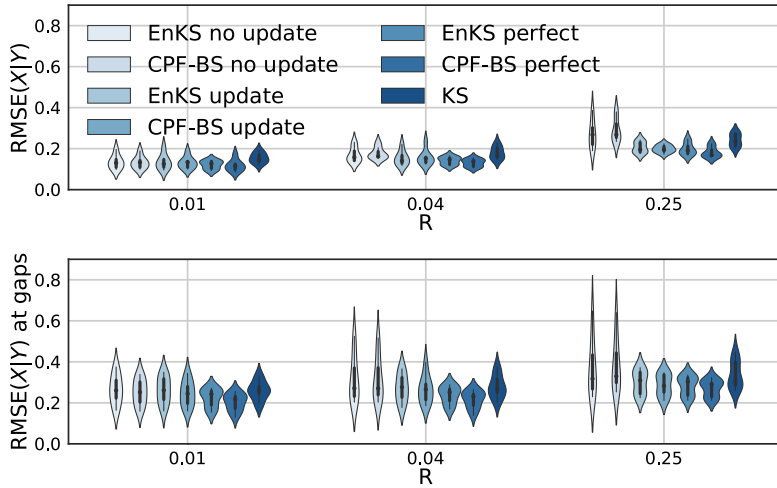
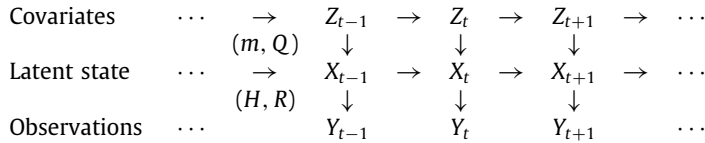


Fig. 8. Reconstruction errors computed on the validation data of log transformed H_s in January 2016 at St Anne du Porzic (France) with respect to different values of observation variance R . Top: $RMSE(X|Y)$ computed between the reconstructed time series and the true state, bottom: $RMSE(X|Y)$ computed at the time steps where data are missing (gaps). Each violin box is obtained from results of 10 repetitions of each algorithm run on different learning sequences, which are randomly sampled in the January months of the period 1994–2015.

lines), $\{Y_t\}$ (dots) and $\{Z_t\}$ (bottom plots) during 10 days in January 2016 are shown with missing values around the 2nd and the 7th of January. Our goal is to impute the missing values as well as reconstruct the time series $\{X_t\}$ from the observed sequence $\{Y_t\}$ and the covariate sequence $\{Z_t\}$. The considered meteorological time series are non-stationary with important seasonal and eventually inter-annual components. A pre-processing step can be applied to the data in order to remove these components. In this work, the inter-annual components (related e.g. to climate change) are neglected and seasonal components can be taken into account by fitting different models separately for the 12 calendar months. Here, we focus only on the January month.

We consider the state–space model (1) whose structure is summarized by the directed graph below.



The dynamical operator m is unknown but it can be estimated using the non-parametric methodology introduced above where the covariates are used in the nearest neighbor search step. In order to evaluate the global performance of the algorithm, a reconstruction error is computed as in the previous section. More precisely, the time series is split into two parts where the first one (data in January for the period 1994–2015) is used to fit the model and the second one (data in January of 2016) is used as a validation sequence to compute the reconstruction error (9). In the experiments, the performances of the algorithms introduced in the previous sections are compared with the ones obtained with a linear Gaussian state–space model, which is defined by (1) where $m(X_{t-1}, Z_t) = \alpha X_{t-1} + \beta$ with α and β real parameters, and $H(x) = x$. The unknown parameters are estimated using the EM algorithm combined with the Kalman smoother (KS) which provides analytic and exact solutions for the smoothing distributions. This approach is one of the most usual approaches to model time series with observation noise.

Fig. 8 shows the reconstruction errors $RMSE(X|Y)$. On the top panel, the error is estimated using the whole time series (reconstruction error) whereas on the bottom plot the error is computed only based on the time step where the data are missing (imputation error). Let us first notice that the algorithms based on CPF-BS only slightly outperform the ones based on EnKS. This may be a sign that, conditionally to the covariates, the nonlinearities in the dynamics of H_s dynamic are not strong. As expected, the reconstruction errors and the imputation errors tend to increase when the variance of the noise increases and the catalog update helps to reduce the reconstruction error, especially when the variance of the observation noise is large.

The imputation error is slightly stronger than the reconstruction error. It is expected because the values are missing over quite large time intervals (about 24 h). In the middle of these gaps, there is less information on H_s and the reconstruction error increases. It is clearly visible in Fig. 9 which shows a times series of 10 days of H_s as well as those of the covariates. On the two plots, the dots are the observations. The red line and shaded areas materialize respectively the empirical mean and a 95% prediction interval of the smoothing distributions. It shows that the npSEM algorithm clearly

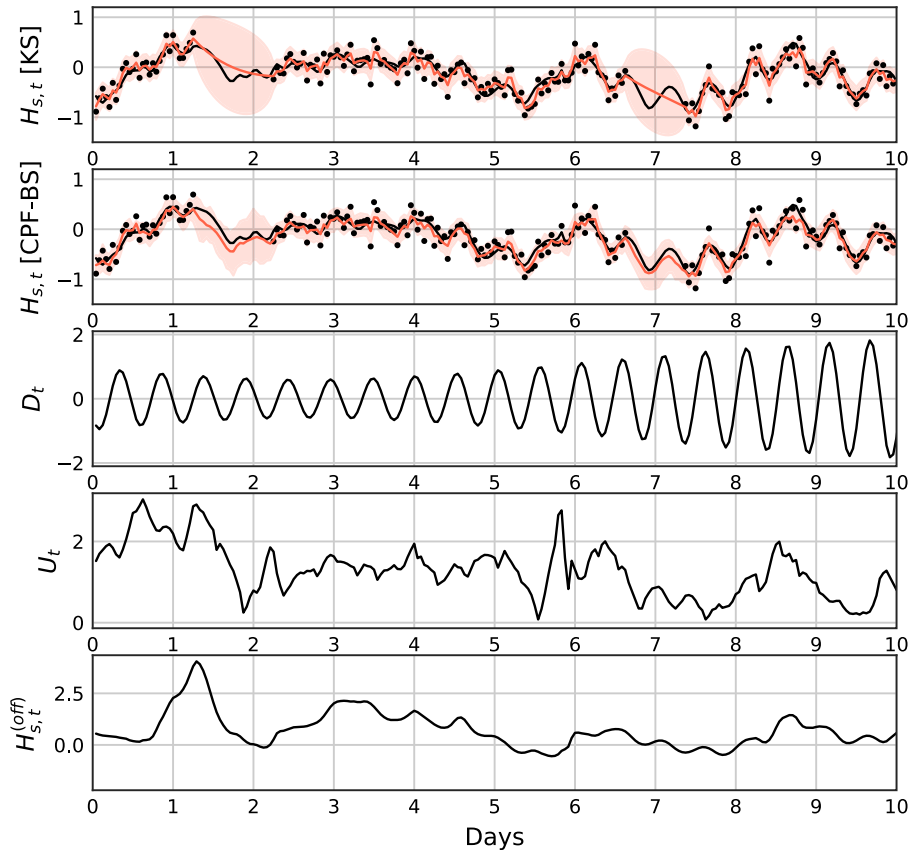


Fig. 9. Log transformed H_s (2 top panels) and covariates (other panels) in January 2016 at St Anne du Porzic (France). The observations with noise $R = 0.04$ are represented by the dots. The reconstruction of the state is represented by the red plain line for the linear SSM (1st panel) and ‘CPF-BS update’ algorithm (2nd panel). The shaded areas represent a 95% prediction interval. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

outperforms the results obtained with the EM algorithm coupled with Kalman smoother in terms of bias and variance. The introduction of covariates in the npSEM algorithm allows to reproduce the tide cycle and include the information brought by the wind and offshore wind conditions even in long gaps with missing data when no information on the H_s condition at the buoy location is available.

6. Conclusions and perspectives

This paper introduces an npSEM algorithm for non-parametric estimation in SSMs. Numerical experiments on toy models and oceanographic data show that it permits to successfully reconstruct the latent state-space from noisy observations and estimate the latent dynamic m .

The proposed methodology has only been validated on low dimensional time series and more works have to be done in order to handle higher dimensional problems since both the particle filters and the nearest-neighbors estimation methods suffer from the curse of dimensionality. Combining recent advances in the particle filters for higher dimensional systems (see e.g. [Beskos et al., 2017](#)) with advanced machine learning approaches for estimating m (see e.g. [Bocquet et al., 2019](#); [Fablet et al., 2017a](#)) may allow to tackle higher dimensional problems with the additional advantage of leading to a reduction of computational costs if the machine learning tool is efficiently implemented.

The proposed approach is based on the assumption that the noise sequences are Gaussian, which may be restrictive for practical applications. In order to give preliminary ideas about the performance of the algorithms when the model is mis-specified some additional numerical experiments are detailed in the Supplementary Material. Data are generated from an SSM with Student noises and the algorithms of the paper are used to reconstruct the state time series. It shows that the reconstruction error slightly increases when the tails of the noise are heavier than the ones of a Gaussian distribution, but the algorithms seem to be pretty robust.

Appendix A

Algorithm 3: Smoothing with Conditional Particle Filter-Backward Simulation (CPF-BS)

Inputs: conditioning trajectory $X^* = x_{0:T}^*$, observations $y_{1:T}$ and fixed parameter θ .

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

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

+ 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 $\{\tilde{x}_{0:T}^j\}_{j=1:N_s}$.

• 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 $p_\theta(x_{0:T} | y_{1:T})$.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.csda.2020.107062>.

References

- Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *Amer. Statist.* 46 (3), 175–185.
- Andrieu, C., Doucet, A., Holenstein, R., 2010. Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 72 (3), 269–342.
- Benaglia, T., Chauveau, D., Hunter, D.R., 2009. An EM-like algorithm for semi-and nonparametric estimation in multivariate mixtures. *J. Comput. Graph. Statist.* 18 (2), 505–526.
- Beskos, A., Crisan, D., Jasra, A., Kamatani, K., Zhou, Y., 2017. A stable particle filter for a class of high-dimensional state-space models. *Adv. Appl. Probab.* 49 (1), 24–48.
- Bocquet, M., Brajard, J., Carrassi, A., Bertino, L., 2019. Data assimilation as a deep learning tool to infer ODE representations of dynamical models. *Nonlinear Process. Geophys. Discuss.* 2019, 1–29.
- Boudière, E., Maisondieu, C., Ardhuin, F., Accensi, M., Pineau-Guillou, L., Lepesqueur, J., 2013. A suitable metocean hindcast database for the design of Marine energy converters. *Int. J. Mar. Energy* 3, e40–e52.
- Brunton, S.L., Brunton, B.W., Proctor, J.L., Kaiser, E., Kutz, J.N., 2017. Chaos as an intermittently forced linear system. *Nat. Commun.* 8 (1), 19.
- Cappé, O., Godsill, S.J., Moulines, E., 2007. An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. IEEE* 95 (5), 899–924.
- Carrassi, A., Bocquet, M., Bertino, L., Evensen, G., 2018. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdiscip. Rev. Clim. Change* e535. <http://dx.doi.org/10.1002/wcc.535>.

- Carroll, R.J., Ruppert, D., Crainiceanu, C.M., Stefanski, L.A., 2006. *Measurement Error in Nonlinear Models: A Modern Perspective*. Chapman and Hall/CRC.
- Chau, T.T.T., Ailliot, P., Monbet, V., Tandeo, P., 2018. An efficient particle-based method for maximum likelihood estimation in nonlinear state-space models. *arXiv preprint arXiv:1804.07483*.
- Cleveland, W.S., Devlin, S.J., 1988. Locally weighted regression: an approach to regression analysis by local fitting. *J. Am. Statist. Assoc.* 83 (403), 596–610.
- Delyon, B., Lavielle, M., Moulines, E., 1999. Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.* 94–128.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. Ser. B Methodol.* 1–38.
- Dreano, D., Tandeo, P., Pulido, M., Ait-El-Fquih, B., Chonavel, T., Hoteit, I., 2017. Estimating model-error covariances in nonlinear state-space models using Kalman smoothing and the expectation-maximization algorithm. *Q. J. R. Meteorol. Soc.* 143 (705), 1877–1885.
- Durbin, J., Koopman, S.J., 2012. *Time Series Analysis by State Space Methods*, Vol. 38. OUP Oxford.
- Fablet, R., Ouala, S., Herzet, C., 2017a. Bilinear residual Neural Network for the identification and forecasting of dynamical systems. *arXiv preprint arXiv:1712.07003*.
- Fablet, R., Viet, P.H., Lguensat, R., 2017b. Data-driven models for the spatio-temporal interpolation of satellite-derived SST fields. *IEEE Trans. Comput. Imag.* 3 (4), 647–657.
- Fan, J., Yao, Q., 2008. *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer Science & Business Media.
- Fearnhead, P., Künsch, H.R., 2018. Particle filters and data assimilation. *Annu. Rev. Stat. Appl.* 5, 421–449.
- Godsill, S.J., Doucet, A., West, M., 2004. Monte Carlo smoothing for nonlinear time series. *J. Am. Statist. Assoc.* 99 (465), 156–168.
- Kantas, N., Doucet, A., Singh, S.S., Maciejowski, J., Chopin, N., et al., 2015. On particle methods for parameter estimation in state-space models. *Statist. Sci.* 30 (3), 328–351.
- Lee, N., Moon, H.R., Zhou, Q., 2017. Many IVs estimation of dynamic panel regression models with measurement error. *J. Econometrics* 200 (2), 251–259.
- Lguensat, R., Tandeo, P., Ailliot, P., Pulido, M., Fablet, R., 2017. The analog data assimilation. *Mon. Weather Rev.* 145 (10), 4093–4107.
- Lindsten, F., Schön, T., Jordan, M.I., 2012. Ancestor sampling for particle gibbs. In: *Advances in Neural Information Processing Systems*. pp. 2591–2599.
- Lindsten, F., Schön, T.B., et al., 2013. Backward simulation methods for Monte Carlo statistical inference. *Found. Trends[®] Mach. Learn.* 6 (1), 1–143.
- Lorenz, E.N., 1963. Deterministic nonperiodic flow. *J. Atmos. Sci.* 20 (2), 130–141.
- Meijer, E., Spierdijk, L., Wansbeek, T., 2013. 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*. Springer, pp. 77–92.
- Musser, D.R., 1997. Introspective sorting and selection algorithms. *Softw. - Pract. Exp.* 27 (8), 983–993.
- O'Carroll, F., 1984. Weather modelling for offshore operations. *J. R. Statist. Soc.: Ser. D Statist.* 33 (1), 161–169.
- Svensson, A., Schön, T.B., Kok, M., 2015. Nonlinear state space smoothing using the conditional particle filter. *IFAC-PapersOnLine* 48 (28), 975–980. <http://dx.doi.org/10.1016/j.ifacol.2015.12.257>, URL: <http://www.sciencedirect.com/science/article/pii/S2405896315028815>. 17th IFAC Symposium on System Identification SYSID 2015.
- Tandeo, P., Ailliot, P., Ruiz, J., Hannart, A., Chapron, B., Cuzol, A., Monbet, V., Easton, R., Fablet, R., 2015. Combining analog method and ensemble data assimilation: application to the Lorenz-63 chaotic system. In: *Machine Learning and Data Mining Approaches to Climate Science*. Springer, pp. 3–12.
- Tippett, M.K., DelSole, T., 2013. Constructed analogs and linear regression. *Mon. Weather Rev.* 141 (7), 2519–2525.
- Wei, G.C., Tanner, M.A., 1990. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Am. Statist. Assoc.* 85 (411), 699–704.
- Yiou, P., 2014. Anawege: a weather generator based on analogues of atmospheric circulation. *Geosci. Model Dev.* 7 (2), 531–543.
- Young, D.S., Hunter, D.R., 2010. Mixtures of regressions with predictor-dependent mixing proportions. *Comput. Statist. Data Anal.* 54 (10), 2253–2266.
- Zhang, B., 2002. An EM algorithm for a semiparametric finite mixture model. *J. Stat. Comput. Simul.* 72 (10), 791–802.