Nested filtering methods for Bayesian inference in state space models

Sara Pérez Vieites

Advisor: Joaquín Míguez

Department of Signal Theory & Communications,
Universidad Carlos III de Madrid.

January, 2022

Index

Index

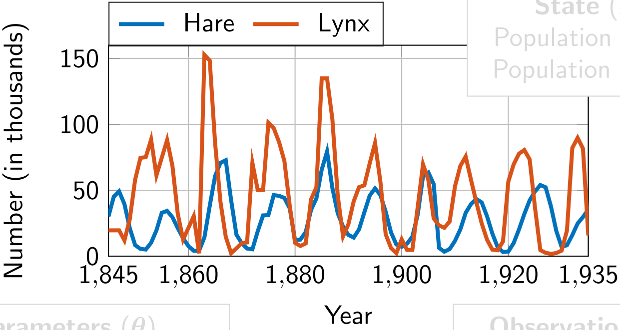
Introduction

We aim at **estimating the time evolution** of **dynamical systems** of different fields of science, such as:

- **Geophysics**. Prediction of the weather, ice sea changes, climate (i.e. fluid dynamics).
- **Biochemistry**. Prediction of the interactions and population of certain molecules.
- **Ecology**. Prediction of the population of prey and predator species in certain region.
- **Quantitative finance**. Evaluation/estimation of price options and risk.
- **Engineering**. Object/target tracking for applications such as surveillance or air traffic control.
- ...

Introduction

Study of hare-lynx interactions in a region of Canada.



State (x_t)
Population of lynx
Population of hare

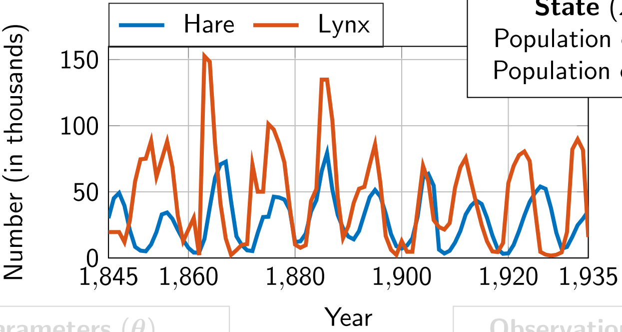
Parameters (θ)
Prey growth rate
Predator-prey encounters
Predator growth rate
Predator mortality rate

Observations (y_t)
Number of pelts sold
Sighting
etc

Introduction

Study of hare-lynx interactions in a region of Canada.

State (x_t)
Population of lynx
Population of hare

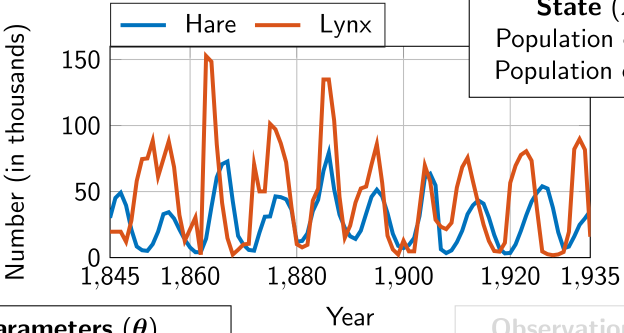


Parameters (θ)
Prey growth rate
Predator-prey encounters
Predator growth rate
Predator mortality rate

Observations (y_t)
Number of pelts sold
Sighting
etc

Introduction

Study of hare-lynx interactions in a region of Canada.



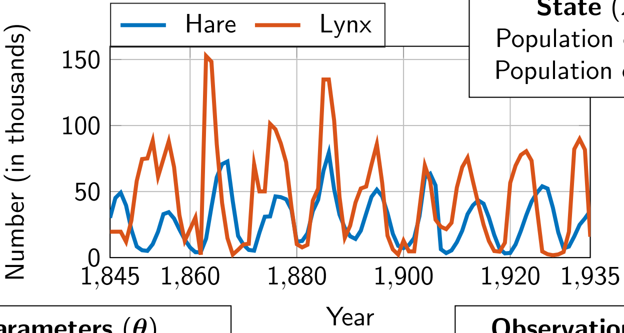
State $(\boldsymbol{x}_t)$
Population of lynx
Population of hare

Parameters (θ)
Prey growth rate
Predator-prey encounters
Predator growth rate
Predator mortality rate

Observations $(\boldsymbol{y}_t)$
Number of pelts sold
Sighting
etc

Introduction

Study of hare-lynx interactions in a region of Canada.



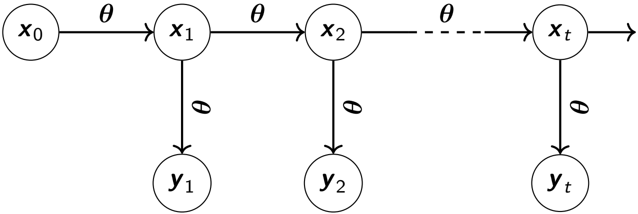
State ($\boldsymbol{x}_t$)
Population of lynx
Population of hare

Parameters (θ)
Prey growth rate
Predator-prey encounters
Predator growth rate
Predator mortality rate

Observations ($\boldsymbol{y}_t$)
Number of pelts sold
Sighting
etc

State-space model

These systems can be represented by **Markov state-space dynamical models**:



State-space model

These state-space systems can be written as

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) + \boldsymbol{v}_t, \\ \boldsymbol{y}_t &= \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{r}_t, \end{aligned}$$

- $\boldsymbol{f}$ and $\boldsymbol{g}$ are the state transition function and the observation function
- $\boldsymbol{v}_t$ and $\boldsymbol{r}_t$ are state and observation noises

In terms of a set of **relevant probability density functions (pdfs)**:

- Prior pdfs: $\theta \sim p(\theta)$ and $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$
- Transition pdf of the state: $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \theta)$
- Conditional pdf of the observation: $\boldsymbol{y}_t \sim p(\boldsymbol{y}_t | \boldsymbol{x}_t, \theta)$

State-space model

These state-space systems can be written as

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) + \boldsymbol{v}_t, \\ \boldsymbol{y}_t &= \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{r}_t, \end{aligned}$$

- $\boldsymbol{f}$ and $\boldsymbol{g}$ are the state transition function and the observation function
- $\boldsymbol{v}_t$ and $\boldsymbol{r}_t$ are state and observation noises

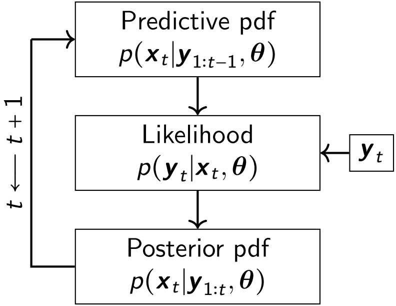
In terms of a set of **relevant probability density functions (pdfs)**:

- Prior pdfs: $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ and $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$
- Transition pdf of the state: $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta})$
- Conditional pdf of the observation: $\boldsymbol{y}_t \sim p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})$

State estimation

⟶ We are interested in the Bayesian estimation of the state variables, this is the **posterior density function of the state** $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$.

Classical filtering methods:



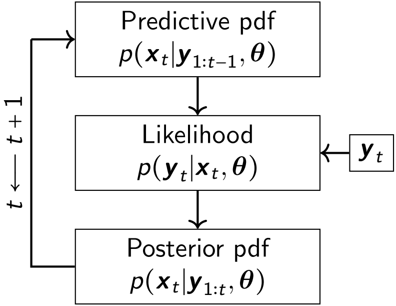
- They assume $\boldsymbol{\theta}$ is known.
- The **predictive pdf** can be computed with the conditional pdf $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ (given by the state-space model).
- The **likelihood** is given by the state-space model.

⟶ Usually $\boldsymbol{\theta}$ is not given.

State estimation

⟶ We are interested in the Bayesian estimation of the state variables, this is the **posterior density function of the state** $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$.

Classical filtering methods:



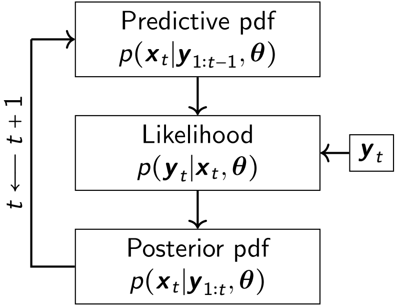
- They assume $\boldsymbol{\theta}$ is known.
- The **predictive pdf** can be computed with the conditional pdf $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ (given by the state-space model).
- The **likelihood** is given by the state-space model.

⟶ Usually $\boldsymbol{\theta}$ is not given.

State estimation

⟶ We are interested in the Bayesian estimation of the state variables, this is the **posterior density function of the state** $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t},\boldsymbol{\theta})$.

Classical filtering methods:

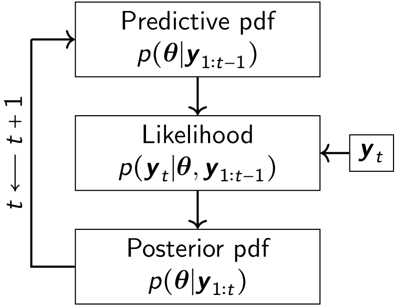


- They assume $\boldsymbol{\theta}$ is known.
- The **predictive pdf** can be computed with the conditional pdf $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{\theta})$ (given by the state-space model).
- The **likelihood** is given by the state-space model.

⟶ Usually $\boldsymbol{\theta}$ is not given.

Parameter estimation

Applying the **same principles** to parameter estimation

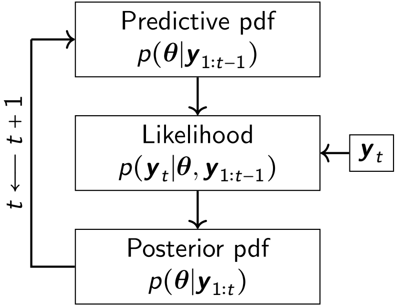


- The **likelihood** is NOT described by the state-space model.
- Neither the likelihood nor the posterior distribution of θ can be computed directly.

$\longrightarrow$ Several approaches have been proposed to solve this problem.

Parameter estimation

Applying the **same principles** to parameter estimation



- The **likelihood** is NOT described by the state-space model.
- Neither the likelihood nor the posterior distribution of $\boldsymbol{\theta}$ can be computed directly.

$\longrightarrow$ Several approaches have been proposed to solve this problem.

State of the Art Methods

Some **methods for parameter and state estimation** can be classified as

- $\longrightarrow$ **State augmentation methods with artificial dynamics**. They use an extended state vector $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{\theta}_t]^\top$.
- $\longrightarrow$ **Particle learning (PL) techniques**. It is a sampling-resampling scheme that depends only on a set of finite-dimensional statistics.
- $\longrightarrow$ **Recursive maximum likelihood (RML) methods**. They are well-principled but they provide only output point estimates.

State of the Art Methods

During the past years, there have been advances leading to methods that

- aim at calculating the **posterior probability distribution of the unknown variables and parameters** of the models.
- can be applied to a broad class of models.
- are **well-principled probabilistic methods** with theoretical guarantees.

State of the Art Methods

Some examples are:

- **sequential Monte Carlo square (SMC^2)**[1] ⟶ batch technique
- **particle Markov chain Monte Carlo (PMCMC)**[2] ⟶ batch
- **nested particle filter (NPF)**[3]: two intertwined layers of Monte Carlo methods (one for the state tracking and the other for the parameter estimation)

⟶ it is a recursive technique.
⟶ The computational cost becomes prohibitive in high-dimensional problems.

[1] Chopin, Jacob, and Papaspiliopoulos, "SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates".

[2] Andrieu, Doucet, and Holenstein, "Particle Markov chain Monte Carlo methods".

[3] Crisan and Miguez, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models".

State of the Art Methods

Some examples are:

- **sequential Monte Carlo square (SMC^2)**[1] ⟶ batch technique
- **particle Markov chain Monte Carlo (PMCMC)**[2] ⟶ batch
- **nested particle filter (NPF)**[3]: two intertwined layers of Monte Carlo methods (one for the state tracking and the other for the parameter estimation)

  ⟶ it is a recursive technique.

  ⟶ The computational cost becomes prohibitive in high-dimensional problems.

---

[1] Chopin, Jacob, and Papaspiliopoulos, "SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates".

[2] Andrieu, Doucet, and Holenstein, "Particle Markov chain Monte Carlo methods".

[3] Crisan and Miguez, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models".

State of the Art Methods

Some examples are:

- **sequential Monte Carlo square (SMC^2)**[1] $\longrightarrow$ batch technique
- **particle Markov chain Monte Carlo (PMCMC)**[2] $\longrightarrow$ batch
- **nested particle filter (NPF)**[3]: two intertwined layers of Monte Carlo methods (one for the state tracking and the other for the parameter estimation)

  $\longrightarrow$ it is a recursive technique.
  $\longrightarrow$ The computational cost becomes prohibitive in high-dimensional problems.

---

[1] Chopin, Jacob, and Papaspiliopoulos, "SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates".

[2] Andrieu, Doucet, and Holenstein, "Particle Markov chain Monte Carlo methods".

[3] Crisan and Miguez, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models".

Summary

The state-of-the-art methods have one or more of the following **issues**:

- Lack of theoretical guarantees.
- Restricted to very specific models.
- Estimation error not quantified.
- Batch technique.
- Prohibitive computational cost for high-dimensional problems.

Objectives

We propose a **generalised methodology** that estimate the **joint posterior probability distribution of the parameters and the state** that

⟶ works **recursively**,

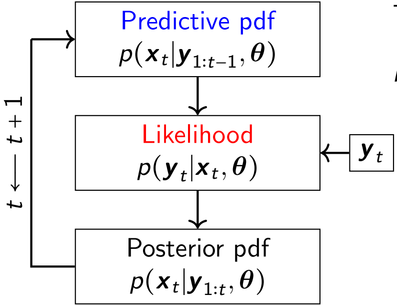
⟶ uses the **nested structure** of the NPF and

⟶ reduces the **computational cost**.

Index

Optimal filter

We assume $\boldsymbol{\theta}$ and the previous post. pdf $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ are known.

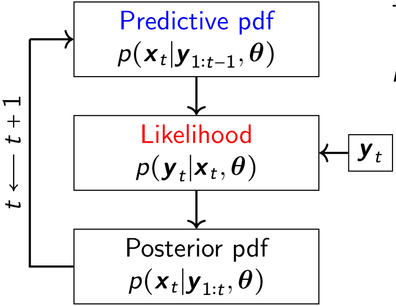


The **posterior pdf** can be written as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$$

Optimal filter

We assume $\boldsymbol{\theta}$ and the previous post. pdf $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ are known.

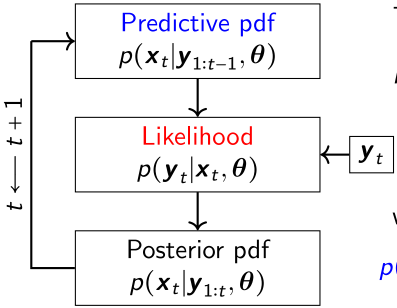


The **posterior pdf** can be written as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$$

Optimal filter

We assume $\boldsymbol{\theta}$ and the previous post. pdf $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ are known.



The **posterior pdf** can be written as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$$

where

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) = \int \underbrace{p(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{x}_{t-1})}_{\text{Transition pdf}} \overbrace{p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})}^{\text{Previous post. pdf}} d\boldsymbol{x}_{t-1}$$

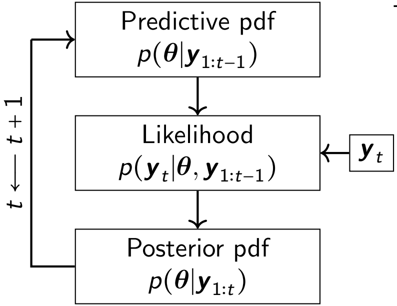
Model inference

We aim at computing the joint posterior pdf $p(\boldsymbol{\theta}, \boldsymbol{x}_t | \boldsymbol{y}_{1:t})$, that can be written as

$$p(\boldsymbol{\theta}, \boldsymbol{x}_t | \boldsymbol{y}_{1:t}) = \underbrace{p(\boldsymbol{x}_t | \boldsymbol{\theta}, \boldsymbol{y}_{1:t})}_{2^{nd} \text{ layer}} \underbrace{p(\boldsymbol{\theta} | \boldsymbol{y}_{1:t})}_{1^{st} \text{ layer}}$$

$\longrightarrow$ The **key difficulty** in this class of models is the Bayesian estimation of the parameter vector $\boldsymbol{\theta}$.

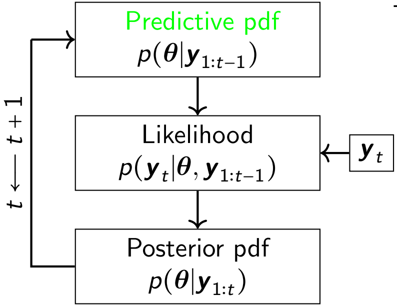
1^{st} layer of inference



The **posterior pdf** can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

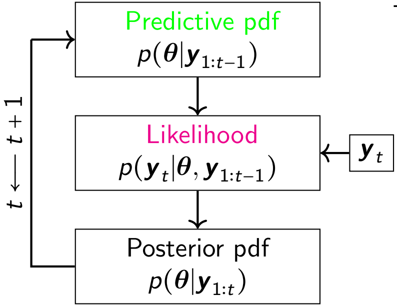
1^{st} layer of inference



The **posterior pdf** can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

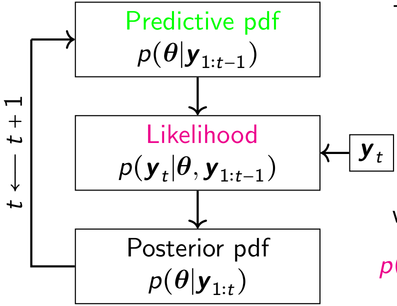
1^{st} layer of inference



The **posterior pdf** can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

1^{st} layer of inference



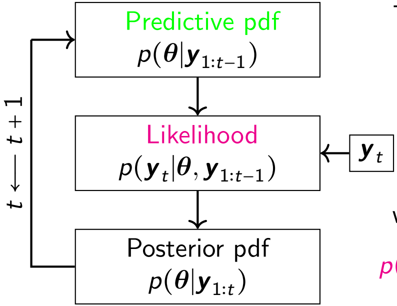
The **posterior pdf** can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

where

$$p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t$$

1^{st} layer of inference



The **posterior pdf** can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})\,p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

where

$$p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1}) = \int \underbrace{p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})}_{\text{Likelihood } (2^{nd} \text{ layer})} \overbrace{p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})}^{\text{Pred. pdf } (2^{nd} \text{ layer})} d\boldsymbol{x}_t$$

Model inference

1st layer

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})}_{\text{Pred. pdf of } \boldsymbol{\theta}}$$

$$\underbrace{p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})}_{\text{Likelihood of } \boldsymbol{\theta}} = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t$$

2nd layer

$$\underbrace{p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})}_{\text{Pred. pdf of } \boldsymbol{x}} = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{\theta})p(\boldsymbol{x}_{t-1}|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\underbrace{p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})}_{\text{Likelihood of } \boldsymbol{x}}$$

$$\underbrace{p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t})}_{\text{Post. pdf of } \boldsymbol{x}} \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})$$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})}_{\text{Post. pdf of } \boldsymbol{\theta}} \propto p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

Index

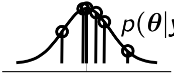
Nested hybrid filter

- We introduce the use of **different types of Monte Carlo methods** in the **first layer** of the algorithm (SMC or SQMC).
- **Gaussian methods** are applied in the **second layer** (EKFs, EnKFs, etc).
- We obtain theoretical guarantees on the **convergence**.

Nested hybrid filter

SMC (N samples)

1st layer



$p(\theta|\mathbf{y}_{1:t-1}) \longrightarrow \{\theta^i_{t-1}\}$

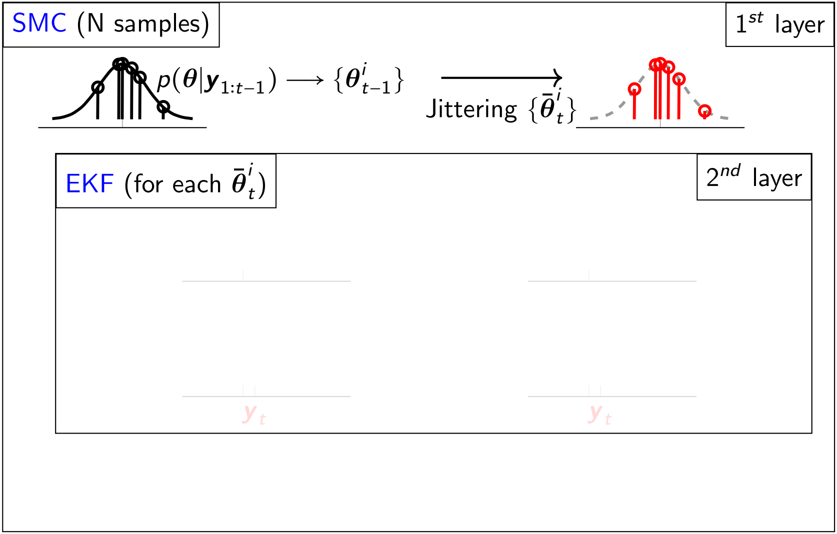
EKF (for each $\bar{\theta}^i_t$)

2nd layer

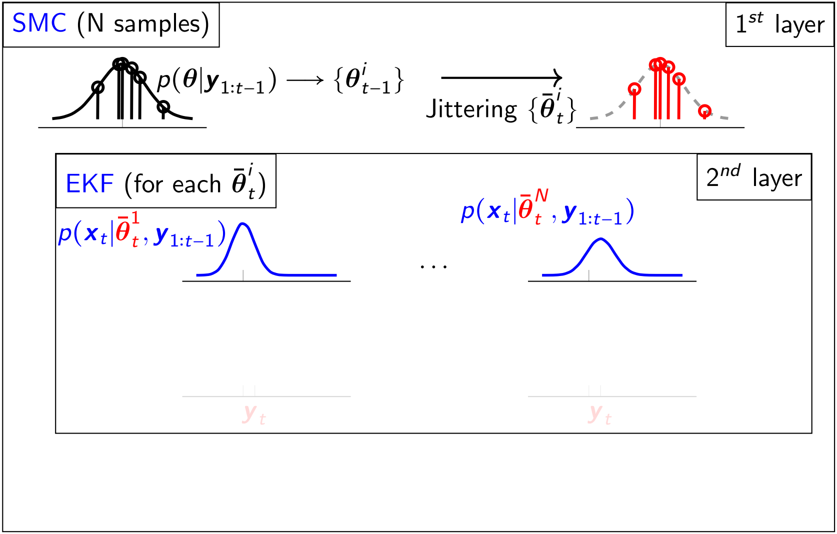
y_t

y_t

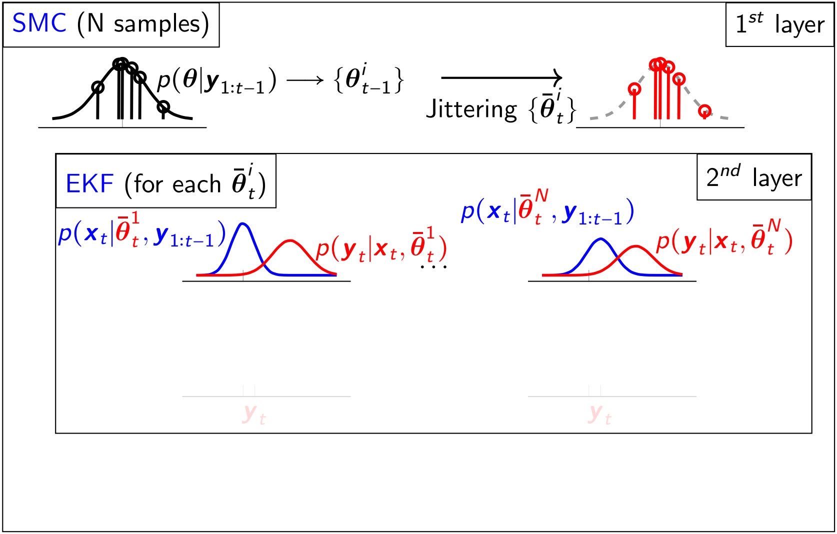
Nested hybrid filter



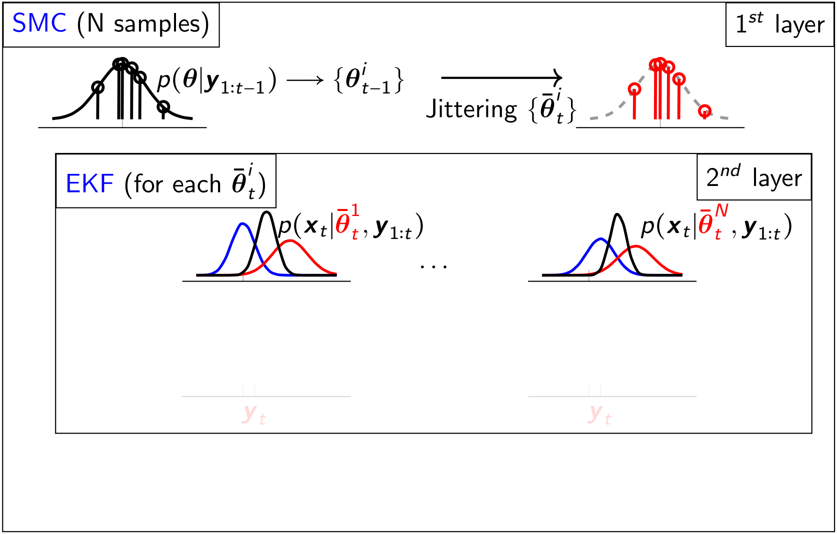
Nested hybrid filter



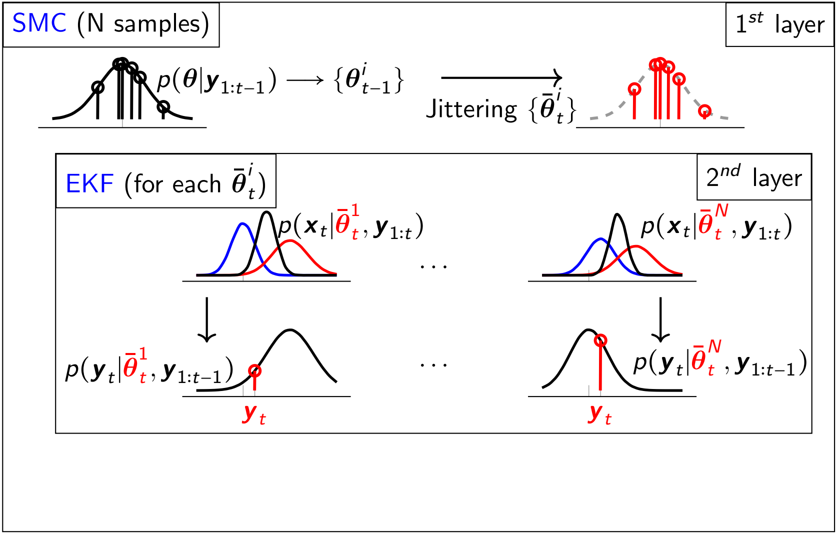
Nested hybrid filter



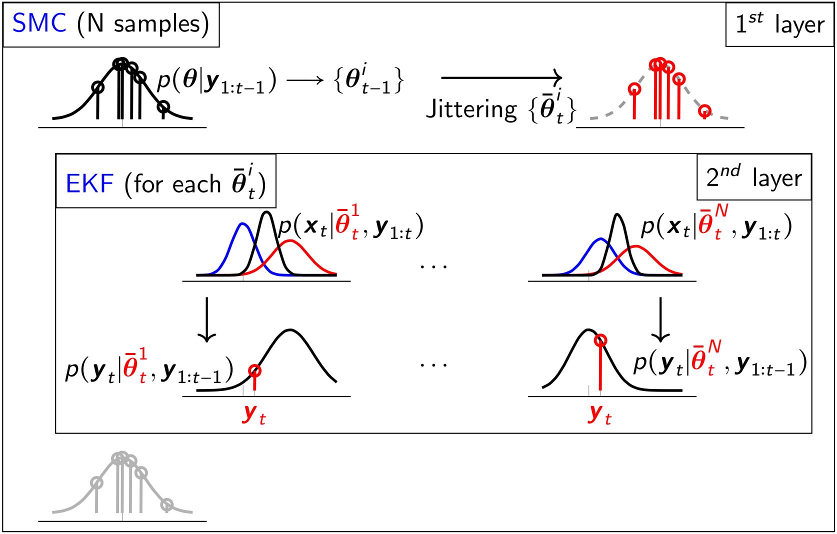
Nested hybrid filter



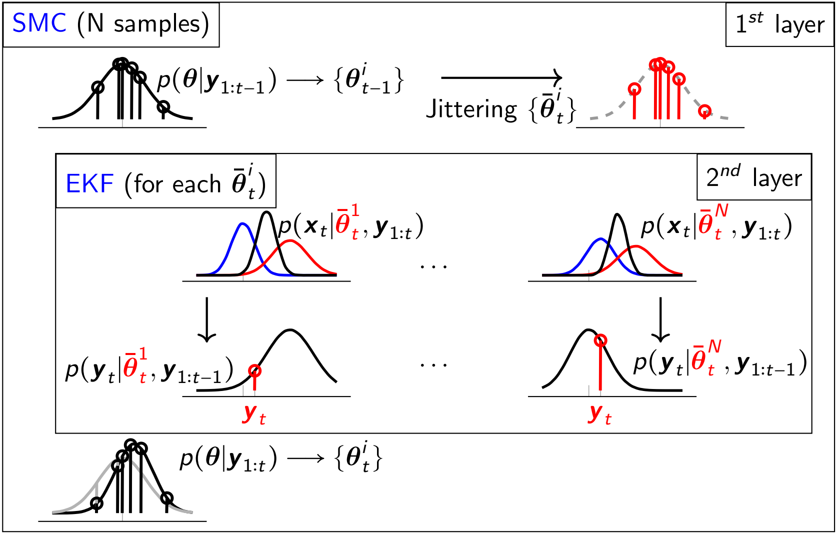
Nested hybrid filter



Nested hybrid filter



Nested hybrid filter



Convergence Theorem

The sequence of posterior probability measures of the unknown parameters, $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$, $t \geq 1$, can be constructed recursively starting from a prior $p(\boldsymbol{\theta})$ as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto u_t(\boldsymbol{\theta}) \star p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

where $u_t(\boldsymbol{\theta}) = p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})$.

A.1. The estimator $\hat{u}_t(\theta)$ is random and can be written as

$$\hat{u}_t(\theta) = u_t(\theta) + b_t(\theta) + m_t(\theta),$$

where $u_t(\theta) := p(\boldsymbol{y}_t|\theta, \boldsymbol{y}_{1:t-1})$ is the **true likelihood**, $m_t(\theta)$ is a zero-mean **random variable** with finite variance and $b_t(\theta)$ is a deterministic and bounded **bias function**.

Convergence Theorem

The sequence of posterior probability measures of the unknown parameters, $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$, $t \geq 1$, can be constructed recursively starting from a prior $p(\boldsymbol{\theta})$ as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto u_t(\boldsymbol{\theta}) \star p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

where $u_t(\boldsymbol{\theta}) = p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})$.

A.1. The estimator $\hat{u}_t(\boldsymbol{\theta})$ is random and can be written as

$$\hat{u}_t(\boldsymbol{\theta}) = u_t(\boldsymbol{\theta}) + b_t(\boldsymbol{\theta}) + m_t(\boldsymbol{\theta}),$$

where $u_t(\boldsymbol{\theta}) := p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})$ is the **true likelihood**, $m_t(\boldsymbol{\theta})$ is a zero-mean **random variable** with finite variance and $b_t(\boldsymbol{\theta})$ is a deterministic and bounded **bias function**.

Convergence Theorem

Theorem 1

Let the sequence of observations $y_{1:t_o}$ be arbitrary but fixed, with $t_o < \infty$, and choose an arbitrary function $h \in B(D)$. Let $p^N(d\boldsymbol{\theta}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_t^i}(d\boldsymbol{\theta})$ be the random probability measure in the parameter space generated by the nested filter. If A.1 holds and under regularity conditions, then

$$\left\| \int h(\boldsymbol{\theta}) p^N(d\boldsymbol{\theta}|\mathbf{y}_{1:t}) - \int h(\boldsymbol{\theta}) \bar{p}(\boldsymbol{\theta}|\mathbf{y}_{1:t}) d\boldsymbol{\theta} \right\|_p \leq \frac{c_t \|h\|_\infty}{\sqrt{N}},$$

for $t = 0, 1, \ldots, t_o$, where $\{c_t\}_{0 \leq t \leq t_o}$ is a sequence of constants independent of N. □

If, instead of the true likelihood $u_t(\theta)$, we use another biased function $\bar{u}_t(\theta) \neq u_t(\theta)$ to update the posterior probability measure $p(\theta|\mathbf{y}_{1:t})$, then we obtain the new sequence of measures

$$\bar{p}(\theta|\mathbf{y}_{1:t}) \propto \bar{u}_t(\theta) \star \bar{p}(\theta|\mathbf{y}_{1:t-1}), \quad t = 1, 2, \ldots$$

Convergence Theorem

Theorem 1

Let the sequence of observations $y_{1:t_o}$ be arbitrary but fixed, with $t_o < \infty$, and choose an arbitrary function $h \in B(D)$. Let $p^N(d\boldsymbol{\theta}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{\theta}_t^i}(d\boldsymbol{\theta})$ be the random probability measure in the parameter space generated by the nested filter. If A.1 holds and under regularity conditions, then

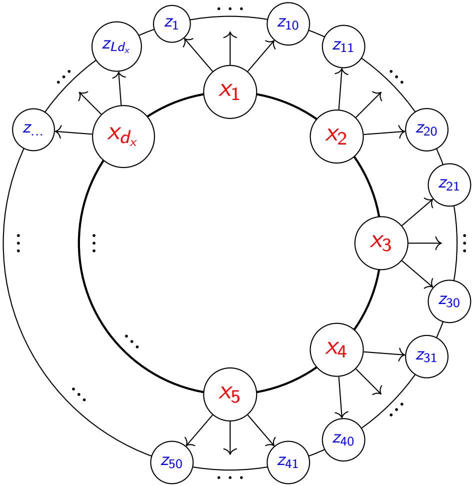
$$\left\| \int h(\boldsymbol{\theta}) p^N(d\boldsymbol{\theta}|\mathbf{y}_{1:t}) - \int h(\boldsymbol{\theta}) \bar{p}(\boldsymbol{\theta}|\mathbf{y}_{1:t}) d\boldsymbol{\theta} \right\|_p \le \frac{c_t \|h\|_\infty}{\sqrt{N}},$$

for $t = 0, 1, \ldots, t_o$, where $\{c_t\}_{0\le t\le t_o}$ is a sequence of constants independent of N. □

If, instead of the true likelihood $u_t(\boldsymbol{\theta})$, we use another biased function $\bar{u}_t(\boldsymbol{\theta}) \neq u_t(\boldsymbol{\theta})$ to update the posterior probability measure $p(\boldsymbol{\theta}|\mathbf{y}_{1:t})$, then we obtain the new sequence of measures

$$\bar{p}(\boldsymbol{\theta}|\mathbf{y}_{1:t}) \propto \bar{u}_t(\boldsymbol{\theta}) \star \bar{p}(\boldsymbol{\theta}|\mathbf{y}_{1:t-1}), \quad t = 1, 2, \ldots$$

A stochastic Lorenz 96 model



A stochastic Lorenz 96 model

We consider a stochastic two-scale Lorenz 96 model that is described by

- the **slow state** variable, $\boldsymbol{x} = [x_1, \ldots, x_{d_x}]^\top$,
- the **fast state** variable, $\boldsymbol{z} = [z_1, \ldots, z_{d_z}]^\top$, with $d_z = Ld_x$,
- the **static parameters** $\boldsymbol{\alpha} = [F, C, B, H]^\top$ and
- for $j = 1, \ldots, d_x$ and $l = 1, \ldots, d_z$, the following **SDEs (in continuous time)**

$$dx_j = \overbrace{\left[-x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F - \frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} z_l \right]}^{\boldsymbol{f}_{1,j}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\alpha})} d\tau + \sigma_x dv_j,$$

$$dz_l = \underbrace{\left[-CBz_{l+1}(z_{l+2} - z_{l-1}) - Cz_l + \frac{CF}{B} + \frac{HC}{B} x_{\lfloor (l-1)L \rfloor} \right]}_{\boldsymbol{f}_{2,l}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\alpha})} d\tau + \sigma_z dw_l,$$

A Stochastic Lorenz 96 Model

- Applying a discretization method with a step Δ, we obtain the **discrete-time version**

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \Delta \boldsymbol{f}_1(\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\alpha}) + \sigma_{\boldsymbol{x}} \boldsymbol{v}_t,$$
$$\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \Delta \boldsymbol{f}_2(\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\alpha}) + \sigma_{\boldsymbol{z}} \boldsymbol{w}_t$$

- The **observations** are written as

$$\boldsymbol{y}_t = \begin{bmatrix} x_{K,tm} \\ x_{2K,tm} \\ \vdots \\ x_{d_y K,tm} \end{bmatrix} + \boldsymbol{r}_t, \quad (1)$$

$\longrightarrow$ This model is used to **generate the ground truth values** for the slow variables $\{\boldsymbol{x}_t\}_{t\geq 0}$, and the synthetic observations, $\{\boldsymbol{y}_t\}_{t\geq 0}$.

A Stochastic Lorenz 96 Model

- Applying a discretization method with a step Δ, we obtain the **discrete-time version**

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \Delta \boldsymbol{f}_1(\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\alpha}) + \sigma_{\boldsymbol{x}} \boldsymbol{v}_t,$$
$$\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \Delta \boldsymbol{f}_2(\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\alpha}) + \sigma_{\boldsymbol{z}} \boldsymbol{w}_t$$

- The **observations** are written as

$$\boldsymbol{y}_t = \begin{bmatrix} x_{K,tm} \\ x_{2K,tm} \\ \vdots \\ x_{d_y K,tm} \end{bmatrix} + \boldsymbol{r}_t, \quad (1)$$

$\longrightarrow$ This model is used to **generate the ground truth values** for the slow variables $\{\boldsymbol{x}_t\}_{t\geq 0}$, and the synthetic observations, $\{\boldsymbol{y}_t\}_{t\geq 0}$.

A Stochastic Lorenz 96 Model

- Applying a discretization method with a step Δ, we obtain the **discrete-time version**

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \Delta \boldsymbol{f}_1(\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\alpha}) + \sigma_{\boldsymbol{x}} \boldsymbol{v}_t,$$
$$\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \Delta \boldsymbol{f}_2(\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{\alpha}) + \sigma_{\boldsymbol{z}} \boldsymbol{w}_t$$

- The **observations** are written as

$$\boldsymbol{y}_t = \begin{bmatrix} x_{K,tm} \\ x_{2K,tm} \\ \vdots \\ x_{d_y K,tm} \end{bmatrix} + \boldsymbol{r}_t, \quad (1)$$

$\longrightarrow$ This model is used to **generate the ground truth values** for the slow variables $\{\boldsymbol{x}_t\}_{t\geq 0}$, and the synthetic observations, $\{\boldsymbol{y}_t\}_{t\geq 0}$.

A Stochastic Lorenz 96 Model

For the **forecast model**, we use instead the differential equation

$$\dot{x}_j = f_j(\boldsymbol{x}, \boldsymbol{\theta}) = -x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F - \frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} z_l \qquad (2)$$

where

- function $\ell(x_j, \mathrm{a})$ is a polynomial in x_j of degree 2, for the coupling term $\frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} \bar{z}_l$.
- $\mathrm{a} = [\mathrm{a}_1, \mathrm{a}_2]^\top$ is a (constant) parameter vector,
- $\boldsymbol{\theta} = [F, \mathrm{a}^\top]^\top$ contains all the parameters.

A Stochastic Lorenz 96 Model

For the **forecast model**, we use instead the differential equation

$$\dot{x}_j = f_j(\boldsymbol{x}, \boldsymbol{\theta}) = -x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F - \ell(x_j, \mathrm{a}) \quad (2)$$

where

- function $\ell(x_j, \mathrm{a})$ is a polynomial in x_j of degree 2, for the coupling term $\frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} \bar{z}_l$.
- $\mathrm{a} = [\mathrm{a}_1, \mathrm{a}_2]^\top$ is a (constant) parameter vector,
- $\theta = [F, \mathrm{a}^\top]^\top$ contains all the parameters.

A Stochastic Lorenz 96 Model

For the **forecast model**, we use instead the differential equation

$$\dot{x}_j = f_j(\boldsymbol{x}, \boldsymbol{\theta}) = -x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F - \ell(x_j, \mathrm{a}) \quad (2)$$

where

- function $\ell(x_j, \mathrm{a})$ is a polynomial in x_j of degree 2, for the coupling term $\frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} \bar{z}_l$.
- $\mathrm{a} = [\mathrm{a}_1, \mathrm{a}_2]^\top$ is a (constant) parameter vector,
- $\boldsymbol{\theta} = [F, \mathrm{a}^\top]^\top$ contains all the parameters.

Numerical results

We have **implemented four different algorithms** following the **nested hybrid methodology**:

		2nd layer	
		Extended Kalman filter (EKF)	Ensemble Kalman filter (EnKF)
1st layer	**Sequential Monte Carlo (SMC)**	**SMC-EKF**	SMC-EnKF
	Sequential quasi-Monte Carlo (SQMC)	SQMC-EKF	SQMC-EnKF

Numerical results

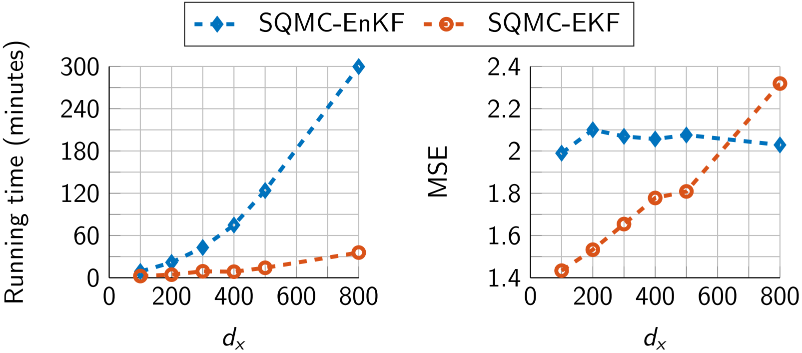
Algorithm	Running time (minutes)	MSE
NHF: SQMC-EKF	2.16	0.46
NHF: SMC-EKF	2.27	0.49
NHF: SQMC-EnKF	6.83	0.62
NHF: SMC-EnKF	7.12	0.95
NPF ($N = M = 800$)	17.96	11.91

$\longrightarrow$ In the four cases of NHF, the accuracy and the running time are improved in comparison to the NPF.

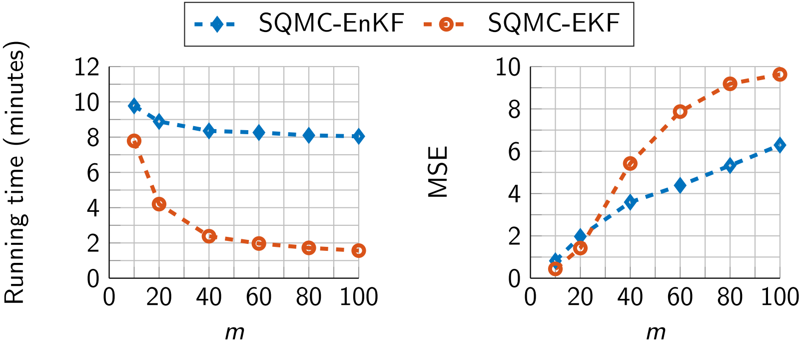
$\longrightarrow$ The least error and running times are obtained with the NHFs that use the EKF in the second layer.

$\longrightarrow$ Using the same samples N, the SQMC reduces slightly the running time and the error compared to the SMC.

Numerical results



Numerical results



⟶ As the **gap between observations** m increases, less data points are effectively available for the estimation task.

Summary of NHFs

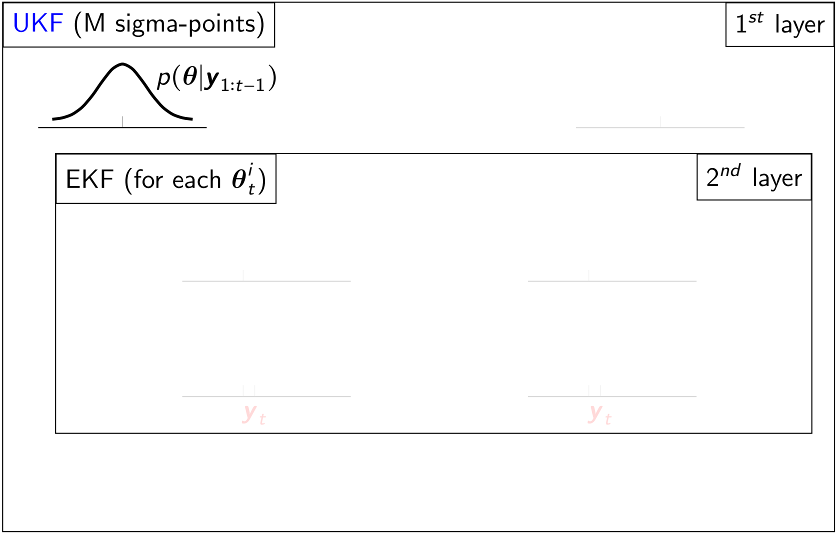
- We introduce the **nested hybrid filters (NHFs)**, that use **Monte Carlo**-based methods in the first layer and **Gaussian** methods in the second layer.
- The algorithm **converges to a well defined limit distribution**.
- We have **implemented four algorithms** (SQMC-EKF, SQMC-EnKF, SMC-EKF and SMC-EnKF) that **outperform the NPF**.
- The **selection of the filtering techniques** in each layer depends on the specific problem.

Index

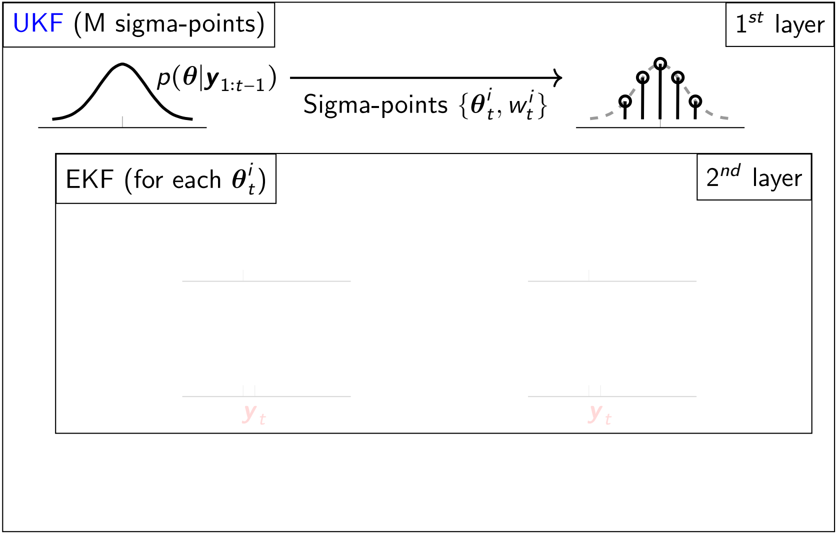
Nested Gaussian filter

- We introduce the use of **deterministic sampling techniques** in the **first layer** of the algorithm, such as the cubature Kalman filter (CKF) and the unscented Kalman filter (UKF).
- We keep applying **Gaussian methods** in the **second layer** of the algorithm.
- We describe how the algorithms can work **sequentially and recursively**.

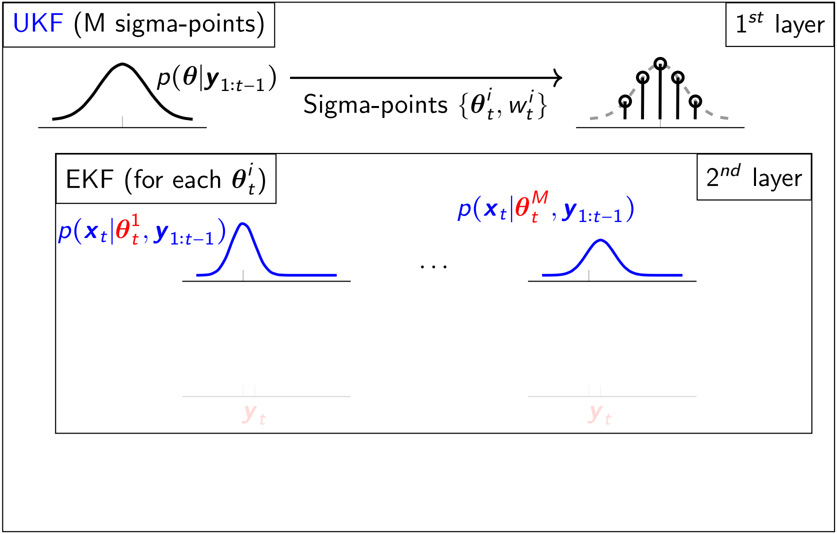
Nested Gaussian filter



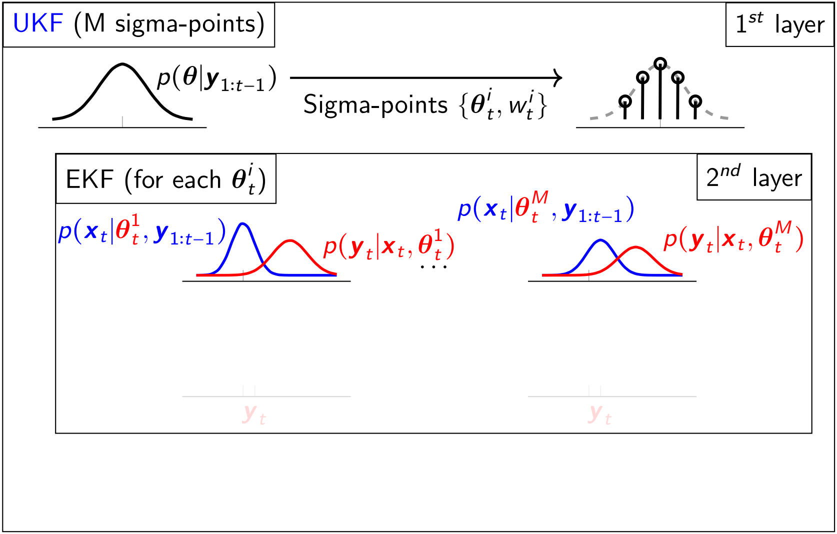
Nested Gaussian filter



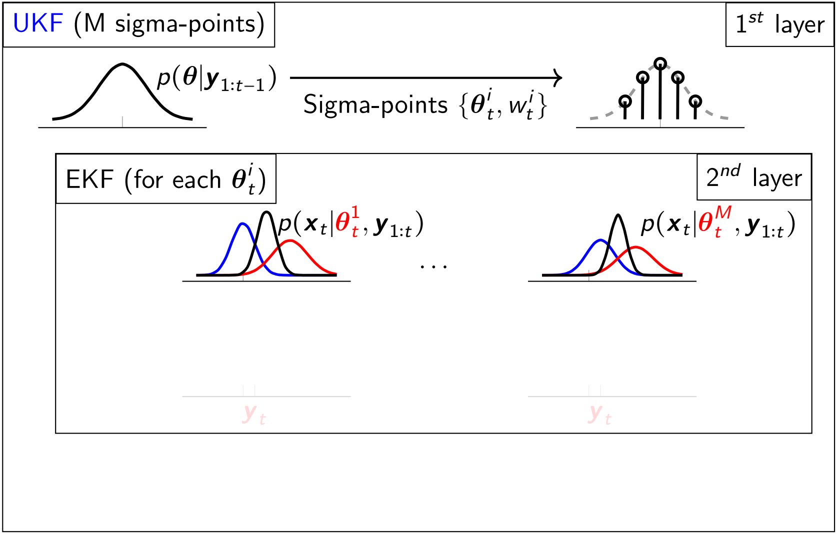
Nested Gaussian filter



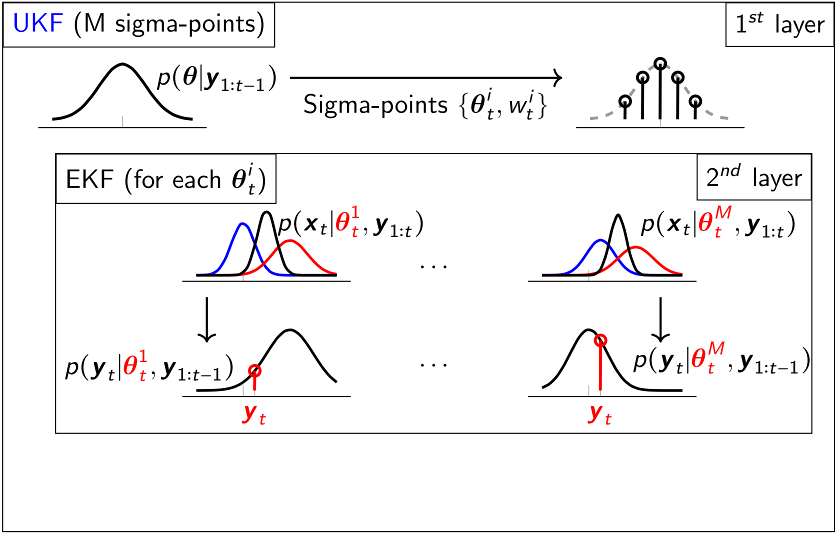
Nested Gaussian filter



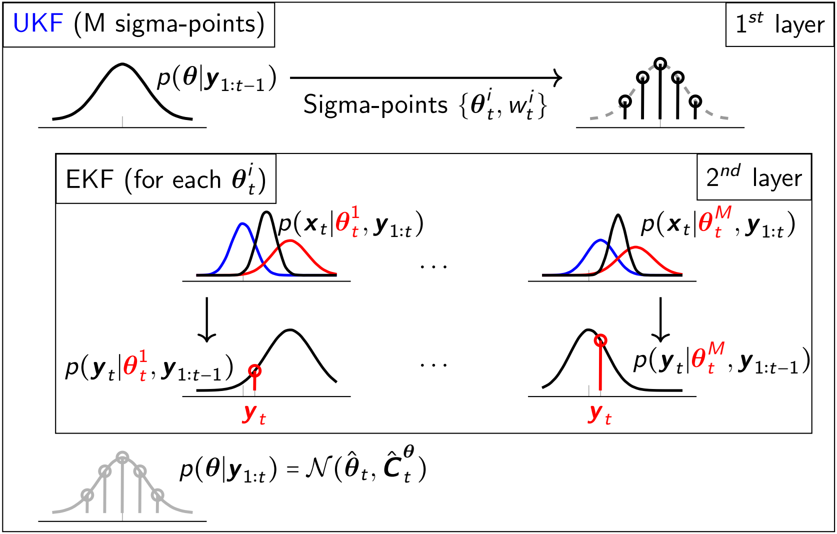
Nested Gaussian filter



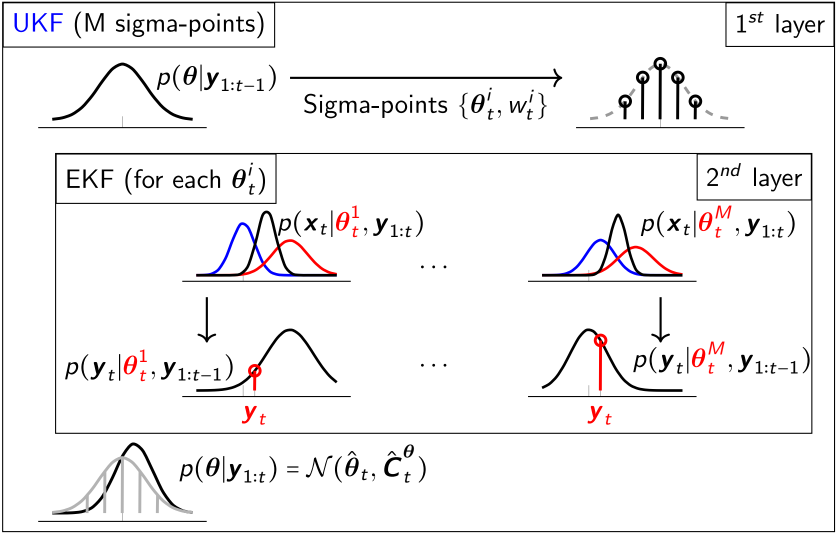
Nested Gaussian filter



Nested Gaussian filter



Nested Gaussian filter



Recursivity of NGF

$\longrightarrow$ This filter is **not recursive**.

- As every time step t the **sigma-points θ_t^i are recalculated**, the computations of the second layer need to **start from scratch**.
- In order to make it **recursive** we approximate

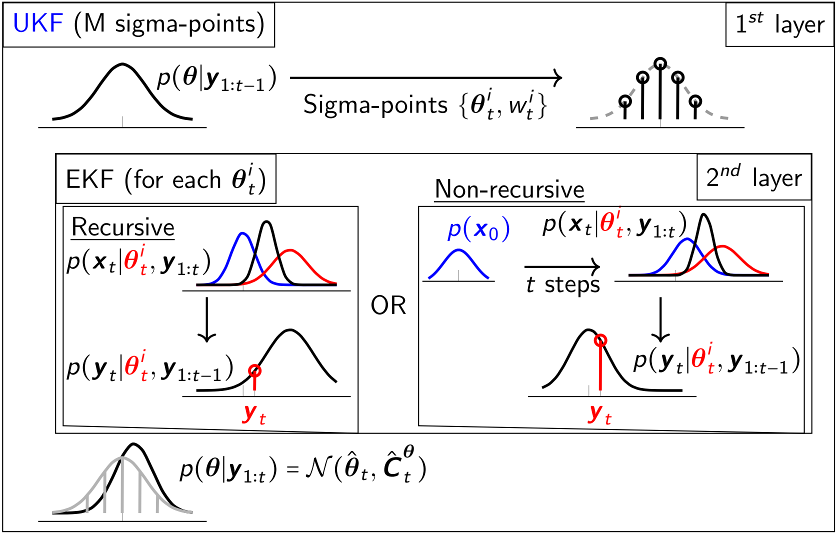
$$p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^i) \approx p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i).$$

Recursive NGF

Every time step the norm $\| \boldsymbol{\theta}_t^i - \boldsymbol{\theta}_{t-1}^i \|_p$ is computed and compared against a prescribed relative **threshold** $\lambda > 0$.

- If $\| \boldsymbol{\theta}_t^i - \boldsymbol{\theta}_{t-1}^i \|_p < \lambda \| \boldsymbol{\theta}_{t-1}^i \|_p$,
  we assume $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^i) \approx p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$.
- If $\| \boldsymbol{\theta}_t^i - \boldsymbol{\theta}_{t-1}^i \|_p > \lambda \| \boldsymbol{\theta}_{t-1}^i \|_p$,
  we need to compute the pdf $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^i)$ from the prior $p(\boldsymbol{x}_0)$.

Nested Gaussian filter

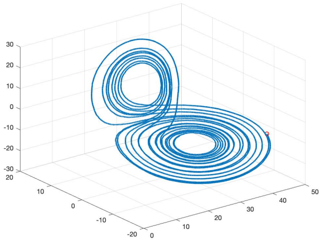


The Lorenz 63 model

We consider a stochastic Lorenz 63 model, whose dynamics are described by

- the **state variables** $\boldsymbol{x}_t$ with dimension $d_x = 3$,
- the **static parameters** $\boldsymbol{\theta} = [S, R, B]^\top$ and
- the following **SDEs**

$$dx_1 = [-S(x_1 - x_2)]d\tau + \sigma dv_1,$$
$$dx_2 = [Rx_1 - x_2 - x_1x_3]d\tau + \sigma dv_2,$$
$$dx_3 = [x_1x_2 - Bx_3]d\tau + \sigma dv_3,$$



The Lorenz 63 model

- Applying a discretization method with step Δ, we obtain

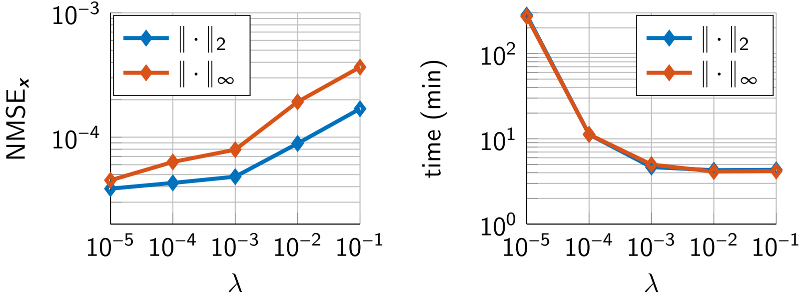
$$x_{1,t+1} = x_{1,t} - \Delta S(x_{1,t} - x_{2,t}) + \sqrt{\Delta}\sigma v_{1,t},$$
$$x_{2,t+1} = x_{2,t} + \Delta[(R - x_{3,t})x_{1,t} - x_{2,t}] + \sqrt{\Delta}\sigma v_{2,t},$$
$$x_{3,t+1} = x_{3,t} + \Delta(x_{1,t}x_{2,t} - Bx_{3,t}) + \sqrt{\Delta}\sigma v_{3,t},$$

- We assume linear observations of the form

$$\boldsymbol{y}_t = k_o \begin{bmatrix} x_{1,t} \\ x_{3,t} \end{bmatrix} + \boldsymbol{r}_t,$$

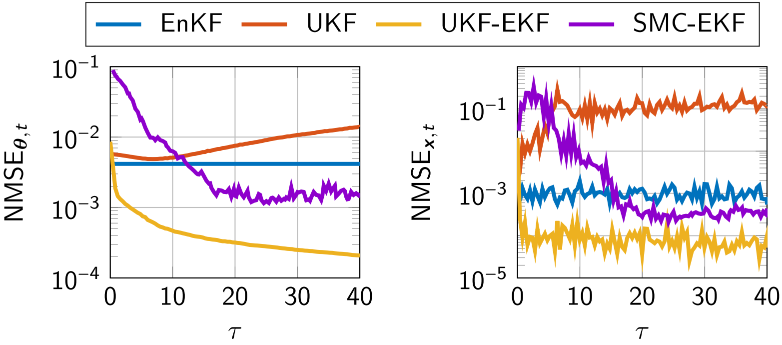
where k_o is a fixed known parameter and $\boldsymbol{r}_t \sim \mathcal{N}(\boldsymbol{r}_t | \boldsymbol{0}, \sigma_y^2 \boldsymbol{I}_2)$.

Numerical results



⟶ We observe that below $\lambda = 10^{-3}$ there is almost no improvement in the error of the state.

Numerical results



Summary of NGFs

- We introduce the **nested Gaussian filters (NGFs)**, that use **deterministic sampling** methods in the first layer and **Gaussian** methods in the second layer.
- We have introduced and assessed the values of a **relative threshold** $\lambda > 0$ that enables the algorithm to work **recursively**.
- We have **implemented a UKF-EKF** and compare it to other algorithms.

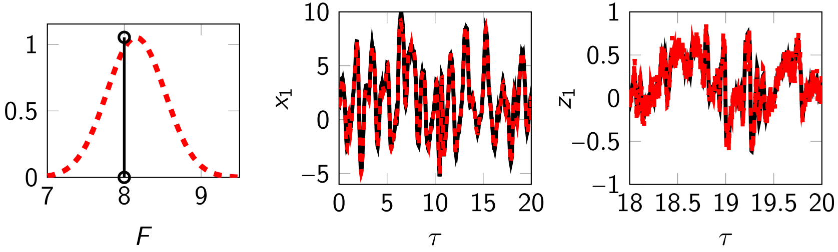
Index

Conclusions

- We have introduced a **generalized nested methodology** that estimates the full posterior distribution of the **static parameters** and **state dynamical variables**.
- This **probabilistic methodology** admits **different types of filtering techniques in each layer**, leading to the nested hybrids filters, the nested Gaussian filters and the NPF.
- We keep the algorithm working **recursively** by applying the **jittering** or by using a **distance dependant on the parameter space**.
- We have proved, under general assumptions, that **the family of nested hybrid filters converges** to a possibly **biased** version of the **posterior distribution of the parameters**.
- The **use of Gaussian filters** in the nested methodology admits **fast implementations** and are well suited to **high-dimensional systems**.

Conclusions

- **Further generalization** of the nested filters to estimate heterogeneous **multi-scale systems**.
- **Three layers of computation** for the static parameters and the two sets of state variables.



Future research

- Further research in the extension of the nested methodology to general **multi-scale state-space models**.
- Further study of the NGFs by introducing **cubature rules** in the first layer of the algorithm.
- Introduction to **recursive maximum likelihood methods** in the first layer of the algorithm.
- Extension of the **convergence analysis**.
- **Application** of the methodology to **large-scale models**.

List of publications

- Sara Pérez-Vieites and Joaquín Míguez. "Kalman-based nested hybrid filters for recursive inference in state-space models". *2020 28th European Signal Processing Conference (EUSIPCO)*, 2468-2472.
- Sara Pérez-Vieites and Joaquín Míguez. "A nested hybrid filter for parameter estimation and state tracking in homogeneous multi-scale models". *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 1-8.
- Sara Pérez-Vieites, Inés Pérez Mariño and Joaquín Míguez. "Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems". *Physical Review E*, 98 (6), 063305.
- Sara Pérez-Vieites and Joaquín Míguez. "Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models". *Signal Processing*, 189, 108295.

Thank you!