Nested filtering methods for Bayesian inference in state space models

Sara Pérez Vieites

CERI Systèmes Numériques,
IMT Nord Europe (Lille, France).

Joint work with
Joaquín Míguez (Universidad Carlos III de Madrid, Spain) and
Víctor Elvira (University of Edinburgh).

February, 2023

Index

Index

Introduction

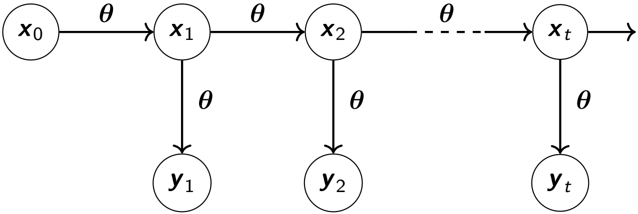
We aim at **estimating the time evolution** of **dynamical systems** of different fields of science, such as:

- **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.
- ...

$\longrightarrow$ There are plenty of applications where the estimation of a dynamical system is needed.

State-space model

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



State-space model

The state ($\boldsymbol{x}_t$), the observations ($\boldsymbol{y}_t$) and the parameters ($\boldsymbol{\theta}$) of these state-space systems are related following the **equations**

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

- $\boldsymbol{f}$ 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-space model

The state ($\boldsymbol{x}_t$), the observations ($\boldsymbol{y}_t$) and the parameters ($\boldsymbol{\theta}$) of these state-space systems are related following the **equations**

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

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

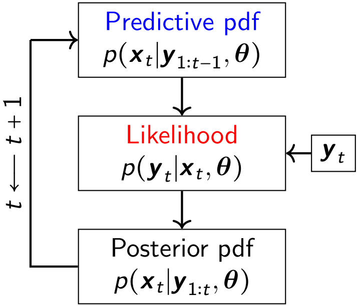
In terms of a set of **relevant probability density functions (pdfs)**:

- 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

$\longrightarrow$ Goal: **Bayesian estimation of the state variables**, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$.

Classical filtering methods assume $\boldsymbol{\theta}$ is known, and compute



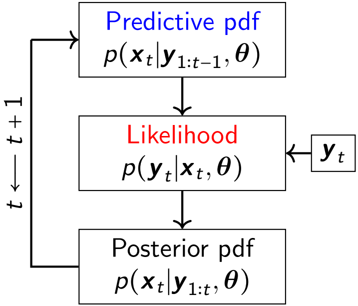
Both $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ and $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ are described by the model.

$\longrightarrow$ Usually $\boldsymbol{\theta}$ is not given and it needs to be estimated as well.
Therefore, we are interested in both parameter and state estimation.

State estimation

$\longrightarrow$ Goal: **Bayesian estimation of the state variables**, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$.

Classical filtering methods assume $\boldsymbol{\theta}$ is known, and compute



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

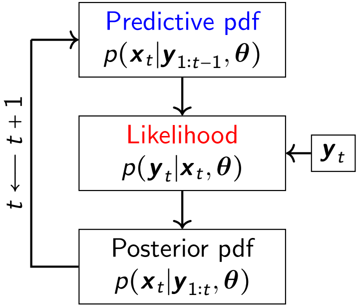
Both $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ and $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ are described by the model.

$\longrightarrow$ Usually $\boldsymbol{\theta}$ is not given and it needs to be estimated as well.
Therefore, we are interested in both parameter and state estimation.

State of the estimation

→ Goal: **Bayesian estimation of the state variables**, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t},\boldsymbol{\theta})$.

Classical filtering methods assume $\boldsymbol{\theta}$ is known, and compute



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

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

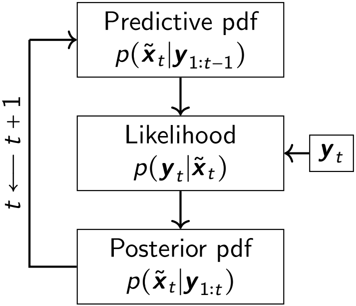
Both $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{\theta})$ and $p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})$ are described by the model.

→ Usually $\boldsymbol{\theta}$ is not given and it needs to be estimated as well.
Therefore, we are interested in both parameter and state estimation.

Index

State-of-the-art methods for parameter and state estimation

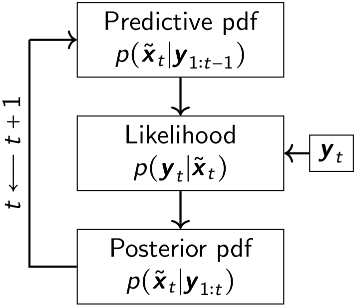
1. **State augmentation methods with artificial dynamics**



- Use of an extended state vector $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{\theta}_t]^\top$.
- Artificial dynamics are introduced in $\boldsymbol{\theta}$ to avoid degeneracy.
- Easy to apply but the artifical dynamics might introduce bias and the method lacks theoretical guarantees.

State-of-the-art methods for parameter and state estimation

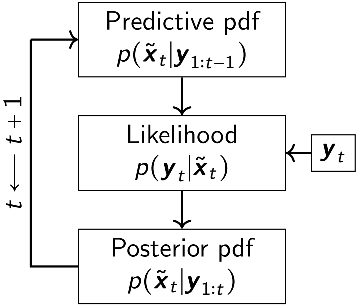
1. **State augmentation methods with artificial dynamics**



- Use of an extended state vector $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{\theta}_t]^\top$.
- Artificial dynamics are introduced in $\boldsymbol{\theta}$ to avoid degeneracy.
- Easy to apply but the artifical dynamics might introduce bias and the method lacks theoretical guarantees.

State-of-the-art methods for parameter and state estimation

1. **State augmentation methods with artificial dynamics**



- Use of an extended state vector $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{\theta}_t]^\top$.
- Artificial dynamics are introduced in $\boldsymbol{\theta}$ to avoid degeneracy.
- Easy to apply but the artifical dynamics might introduce bias and the method lacks theoretical guarantees.

State-of-the-art methods for parameter and state estimation

2. **Particle learning (PL) techniques**
   - It is a sampling-resampling scheme.
   - It depends only on a set of finite-dimensional statistics. In a Monte Carlo setting this means that the static parameters can be efficiently represented by sampling.
   - The posterior probability distribution of $\boldsymbol{\theta}$ conditional on the states $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t$ can be computed in closed form.
   - However, this approach is restricted to very specific models.

State-of-the-art methods for parameter and state estimation

2. **Particle learning (PL) techniques**
   - It is a sampling-resampling scheme.
   - It depends only on a set of finite-dimensional statistics. In a Monte Carlo setting this means that the static parameters can be efficiently represented by sampling.
   - The posterior probability distribution of $\boldsymbol{\theta}$ conditional on the states $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t$ can be computed in closed form.
   - However, this approach is restricted to very specific models.

State-of-the-art methods for parameter and state estimation

3. **Recursive maximum likelihood (RML) methods**

   - They enable the sequential processing of the observed data as they are collected.
   - They are well-principled.
   - They can be applied to a broad class of models.
   - However, they do not yield full posterior distributions of the unknowns and therefore, they only output point estimates instead.

State-of-the-art methods for parameter and state estimation

3. **Recursive maximum likelihood (RML) methods**
   - They enable the sequential processing of the observed data as they are collected.
   - They are well-principled.
   - They can be applied to a broad class of models.
   - However, they do not yield full posterior distributions of the unknowns and therefore, they only output point estimates instead.

State-of-the-art methods for parameter and state estimation

4. There have been advances leading to methods that
   - aim at calculating the **posterior probability distribution of the unknown variables and parameters** of the models and they can quantify the uncertainty or estimation error.
   - can be applied to a broad class of models.
   - are **well-principled probabilistic methods** with theoretical guarantees.

State-of-the-art methods for parameter and state estimation

4. There have been advances leading to methods that
   - aim at calculating the **posterior probability distribution of the unknown variables and parameters** of the models and they can quantify the uncertainty or estimation error.
   - can be applied to a broad class of models.
   - are **well-principled probabilistic methods** with theoretical guarantees.

State-of-the-art methods for parameter and state estimation

Some examples are:

- **particle Markov chain Monte Carlo (PMCMC)**[1]
- **sequential Monte Carlo square (SMC^2)**[2]

⟶ they are batch techniques: the whole sequence of observations has to to be re-processed from scratch.

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

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

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

State-of-the-art methods for parameter and state estimation

Some examples are:

- **particle Markov chain Monte Carlo (PMCMC)**[1]
- **sequential Monte Carlo square (SMC^2)**[2]

$\longrightarrow$ they are batch techniques: the whole sequence of observations has to to be re-processed from scratch.

$\longrightarrow$ The computational cost becomes prohibitive in high-dimensional problems.

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

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

Summary and Motivation

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 (it only provides point estimates).
- Batch technique (the whole sequence of observations have to be re-processed from scratch every time step).
- Prohibitive computational cost for high-dimensional problems.

$\longrightarrow$ We propose a **set of algorithms** that estimate the **joint posterior probability distribution of the parameters and the state**, while solving all the issues.

Index

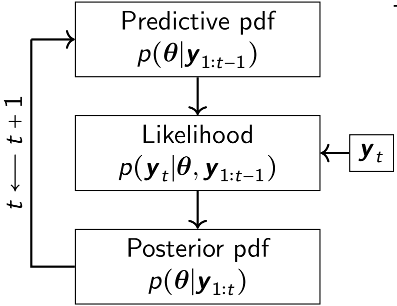
Model inference

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 $\boldsymbol{\theta}$.

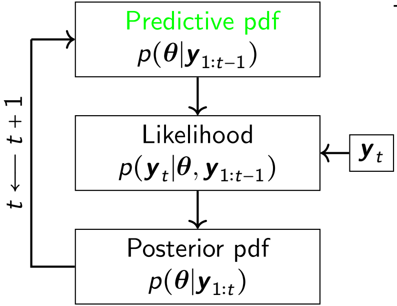
1^{st} layer of inference



The **posterior pdf** can be written as

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

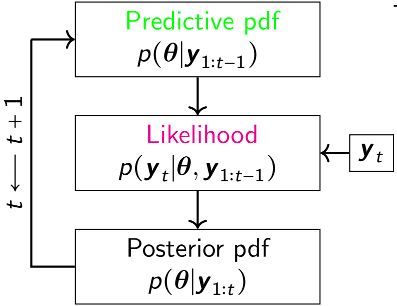
1^{st} layer of inference



The **posterior pdf** can be written as

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

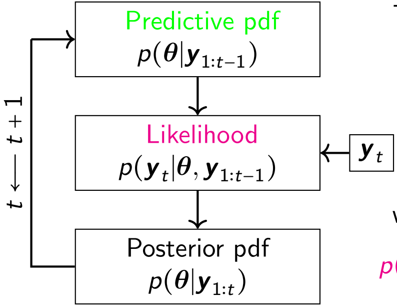
1^{st} layer of inference



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

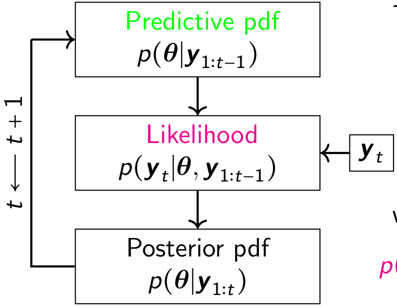
$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

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

1^{st} layer

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

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

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

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

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

2^{nd} layer

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

Family of nested filters

1. **Nested particle filters (NPFs)**[3].
   - Both layers ⟶ Sequential Monte Carlo (SMC) methods
2. **Nested hybrid filters (NHFs)**[4].
   - θ-layer ⟶ Monte Carlo-based methods (e.g., SMC or SQMC)
   - x-layer ⟶ Gaussian techniques (e.g., EKFs or EnKFs)
3. **Nested Gaussian filters (NGFs)**[5].
   - θ-layer ⟶ Deterministic sampling methods (e.g., UKF).
   - x-layer ⟶ Gaussian techniques (e.g., EKFs or EnKFs).

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

[4] Pérez-Vieites, Mariño, and Míguez, "Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems".

[5] Pérez-Vieites and Míguez, "Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models".

Family of nested filters

1. **Nested particle filters (NPFs)**[3].
   - Both layers ⟶ Sequential Monte Carlo (SMC) methods
2. **Nested hybrid filters (NHFs)**[4].
   - $\boldsymbol{\theta}$-layer ⟶ Monte Carlo-based methods (e.g., SMC or SQMC)
   - $\boldsymbol{x}$-layer ⟶ Gaussian techniques (e.g., EKFs or EnKFs)
3. **Nested Gaussian filters (NGFs)**[5].
   - $\boldsymbol{\theta}$-layer ⟶ Deterministic sampling methods (e.g., UKF).
   - $\boldsymbol{x}$-layer ⟶ Gaussian techniques (e.g., EKFs or EnKFs).

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

[4] Pérez-Vieites, Mariño, and Míguez, "Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems".

[5] Pérez-Vieites and Míguez, "Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models".

Family of nested filters

1. **Nested particle filters (NPFs)**[3].
   - Both layers ⟶ Sequential Monte Carlo (SMC) methods
2. **Nested hybrid filters (NHFs)**[4].
   - $\boldsymbol{\theta}$-layer ⟶ Monte Carlo-based methods (e.g., SMC or SQMC)
   - $\boldsymbol{x}$-layer ⟶ Gaussian techniques (e.g., EKFs or EnKFs)
3. **Nested Gaussian filters (NGFs)**[5].
   - $\boldsymbol{\theta}$-layer ⟶ Deterministic sampling methods (e.g., UKF).
   - $\boldsymbol{x}$-layer ⟶ Gaussian techniques (e.g., EKFs or EnKFs).

---

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

[4] Pérez-Vieites, Mariño, and Míguez, "Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems".

[5] Pérez-Vieites and Míguez, "Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models".

Index

Nested hybrid filter (NHF)

SMC (N samples)

1^{st} layer

- Jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta}_{t-1}^i), 1 \leq i \leq N \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$

- Likelihood of θ: weights $w_t^i \longrightarrow p(\boldsymbol{y}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample i)

2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$

- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \bar{\boldsymbol{\theta}}_t^i)$

- Update: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i})$

- Resampling: $\{\boldsymbol{\theta}_t^i, \boldsymbol{x}^i_{t|t,\boldsymbol{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\boldsymbol{\theta}_t^i}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$

Nested hybrid filter (NHF)

SMC (N samples) — 1^{st} layer

- Jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta}_{t-1}^i), 1 \le i \le N \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta}$: weights $w_t^i \longrightarrow p(\boldsymbol{y}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample i) — 2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\theta}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\theta}_t^i})$
- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \bar{\boldsymbol{\theta}}_t^i)$
- Update: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\bar{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\bar{\theta}_t^i})$

- Resampling: $\{\boldsymbol{\theta}_t^i, \boldsymbol{x}^i_{t|t,\theta_t^i}, \boldsymbol{C}^i_{t|t,\theta_t^i}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$

Nested hybrid filter (NHF)

SMC (N samples)

1^{st} layer

- Jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta}_{t-1}^i), 1 \leq i \leq N \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta}$: weights $w_t^i \longrightarrow p(\boldsymbol{y}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample i)

2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$
- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \bar{\boldsymbol{\theta}}_t^i)$
- Update: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i})$

- Resampling: $\{\boldsymbol{\theta}_t^i, \boldsymbol{x}^i_{t|t,\boldsymbol{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\boldsymbol{\theta}_t^i}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$

Nested hybrid filter (NHF)

SMC (N samples)

1^{st} layer

- Jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta}_{t-1}^i), 1 \leq i \leq N \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$

- Likelihood of $\boldsymbol{\theta}$: weights $w_t^i \longrightarrow p(\boldsymbol{y}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample i)

2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$

- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \bar{\boldsymbol{\theta}}_t^i)$

- Update: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i})$

- Resampling: $\{\boldsymbol{\theta}_t^i, \boldsymbol{x}^i_{t|t,\boldsymbol{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\boldsymbol{\theta}_t^i}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$

Nested hybrid filter (NHF)

SMC (N samples)

1^{st} layer

- Jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta}_{t-1}^i), 1 \leq i \leq N \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta}$: weights $w_t^i \longrightarrow p(\boldsymbol{y}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample i)

2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$
- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \bar{\boldsymbol{\theta}}_t^i)$
- Update: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i})$

- Resampling: $\{\theta_t^i, \boldsymbol{x}^i_{t|t,\theta_t^i}, \boldsymbol{C}^i_{t|t,\theta_t^i}\} \longrightarrow p(\theta|\boldsymbol{y}_{1:t})$

Nested hybrid filter (NHF)

SMC (N samples) — 1^{st} layer

- Jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta}_{t-1}^i), 1 \leq i \leq N \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta}$: weights $w_t^i \longrightarrow p(\boldsymbol{y}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample i) — 2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$
- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \bar{\boldsymbol{\theta}}_t^i)$
- Update: $p(\boldsymbol{x}_t|\bar{\boldsymbol{\theta}}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t,\bar{\boldsymbol{\theta}}_t^i})$

- Resampling: $\{\boldsymbol{\theta}_t^i, \boldsymbol{x}^i_{t|t,\boldsymbol{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\boldsymbol{\theta}_t^i}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$

Recursivity of NPF and NHF

- θ is static and **samples θ_{t-1}^i do not evolve**. After several resampling steps the filter would degenerate.
- It is convenient to have a procedure to generate a new set $\{\bar{\theta}_t^i\}_{1\le i\le N}$ which yields an approximation of $p(\theta|\mathbf{y}_{1:t-1})$.

$\longrightarrow$ The jittering step allows these filters to run **recursively**:

We mutate the particles $\theta_{t-1}^1, \ldots, \theta_{t-1}^N$ independently using a jittering kernel $\kappa_N(\theta|d\theta)$ and obtain $\bar{\theta}_t^1, \ldots, \bar{\theta}_t^N$.

Recursivity of NPF and NHF

- $\boldsymbol{\theta}$ is static and **samples θ^i_{t-1} do not evolve**. After several resampling steps the filter would degenerate.
- It is convenient to have a procedure to generate a new set $\{\bar{\boldsymbol{\theta}}^i_t\}_{1 \le i \le N}$ which yields an approximation of $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$.

$\longrightarrow$ The jittering step allows these filters to run **recursively**:

We mutate the particles $\boldsymbol{\theta}^1_{t-1}, \dots, \boldsymbol{\theta}^N_{t-1}$ independently using a jittering kernel $\kappa_N(\boldsymbol{\theta}|d\boldsymbol{\theta})$ and obtain $\bar{\boldsymbol{\theta}}^1_t, \dots, \bar{\boldsymbol{\theta}}^N_t$.

Convergence Theorem (NHF)

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 (NHF)

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 (NHF)

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 (NHF)

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

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

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

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

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

Nested Gaussian filter (NGF)

UKF (M sigma-points) | 1^{st} layer

- Generate sigma-points: $\{\theta_t^i, w_t^i\}, 0 \leq i \leq M-1 \longrightarrow p(\theta|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\theta \longrightarrow p(\boldsymbol{y}_t|\theta_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample sigma-point i) | 2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\theta_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\theta}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\theta}_t^i})$
- Likelihood of θ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \theta_t^i)$
- Update: $p(\boldsymbol{x}_t|\theta_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\theta_t^i}, \boldsymbol{C}^i_{t|t,\theta_t^i})$

- Compute $\hat{\theta}_t^i$ and $\hat{\boldsymbol{C}}_t^\theta \longrightarrow p(\theta|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\theta_t|\hat{\theta}_t^i, \hat{\boldsymbol{C}}_t^\theta)$

Nested Gaussian filter (NGF)

1^{st} layer

UKF (M sigma-points)

- Generate sigma-points: $\{\boldsymbol{\theta}_t^i, w_t^i\}, 0 \leq i \leq M-1 \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta} \longrightarrow p(\boldsymbol{y}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t-1})$

2^{nd} layer

EKF (per each sample sigma-point i)

- Prediction: $p(\boldsymbol{x}_t|\theta_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\theta}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\theta}_t^i})$
- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \theta_t^i)$
- Update: $p(\boldsymbol{x}_t|\theta_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\theta_t^i}, \boldsymbol{C}^i_{t|t,\theta_t^i})$

- Compute $\hat{\theta}_t^i$ and $\hat{\boldsymbol{C}}_t^\theta \longrightarrow p(\theta|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\theta_t|\hat{\theta}_t^i, \hat{\boldsymbol{C}}_t^\theta)$

Nested Gaussian filter (NGF)

UKF (M sigma-points) — 1^{st} layer

- Generate sigma-points: $\{\boldsymbol{\theta}_t^i, w_t^i\}, 0 \leq i \leq M-1 \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta} \longrightarrow p(\boldsymbol{y}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t-1})$

EKF (per each sample sigma-point i) — 2^{nd} layer

- Prediction: $p(\boldsymbol{x}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$
- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}_t^i)$
- Update: $p(\boldsymbol{x}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\boldsymbol{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\boldsymbol{\theta}_t^i})$

- Compute $\hat{\theta}_t^i$ and $\hat{C}_t^\theta \longrightarrow p(\theta|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\theta_t|\hat{\theta}_t^i, \hat{C}_t^\theta)$

Nested Gaussian filter (NGF)

UKF (M sigma-points) — 1^{st} layer

- Generate sigma-points: $\{\boldsymbol{\theta}_t^i, w_t^i\}, 0 \leq i \leq M-1 \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$
- Likelihood of $\boldsymbol{\theta} \longrightarrow p(\boldsymbol{y}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t-1})$

  EKF (per each sample sigma-point i) — 2^{nd} layer

  - Prediction: $p(\boldsymbol{x}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i}, \boldsymbol{C}^i_{t|t-1,\bar{\boldsymbol{\theta}}_t^i})$
  - Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}_t^i)$
  - Update: $p(\boldsymbol{x}_t|\boldsymbol{\theta}_t^i, \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{x}^i_{t|t,\boldsymbol{\theta}_t^i}, \boldsymbol{C}^i_{t|t,\boldsymbol{\theta}_t^i})$

- Compute $\hat{\boldsymbol{\theta}}_t^i$ and $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{\theta}_t|\hat{\boldsymbol{\theta}}_t^i, \hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}})$

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 $p(\boldsymbol{x}_0)$.

Index

Ongoing work

Focusing on an efficient use of the available computational resources.

- Reduction of the number of $\boldsymbol{\theta}$-samples when the filter converges [Accepted paper, ICASSP 2023][6].
- Adapting the number of samples of each layer online
  $\longrightarrow$ Further study of $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$.

[6] Pérez-Vieites and Elvira, "Adaptive Gaussian nested filter for parameter estimation and state tracking in dynamical systems".

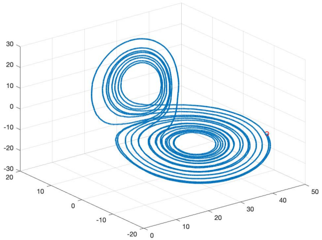
Index

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**

$$dx_1 = [-S(x_1 - x_2)]d\tau + \sigma dv_1,$$
$$dx_2 = [Rx_1 - x_2 - x_1 x_3]d\tau + \sigma dv_2,$$
$$dx_3 = [x_1 x_2 - Bx_3]d\tau + \sigma dv_3,$$



The Lorenz 63 model

- Applying a discretization method with step Δ, we obtain

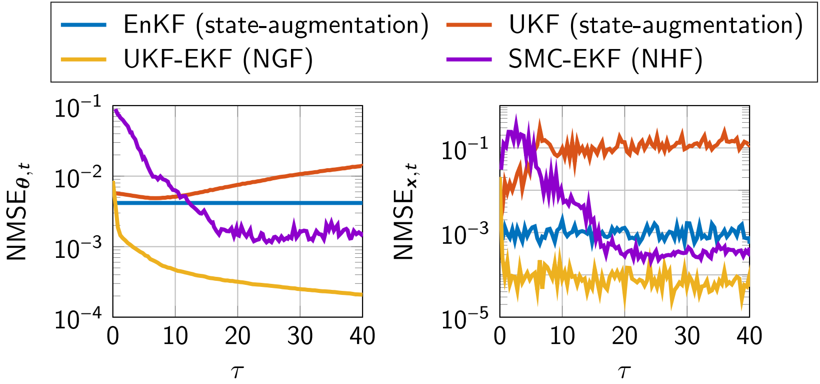
$$x_{1,t+1} = x_{1,t} - \Delta S(x_{1,t} - x_{2,t}) + \sqrt{\Delta}\sigma v_{1,t},$$
$$x_{2,t+1} = x_{2,t} + \Delta[(R - x_{3,t})x_{1,t} - x_{2,t}] + \sqrt{\Delta}\sigma v_{2,t},$$
$$x_{3,t+1} = x_{3,t} + \Delta(x_{1,t}x_{2,t} - Bx_{3,t}) + \sqrt{\Delta}\sigma v_{3,t},$$

- 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 [Signal Processing 2021][7]

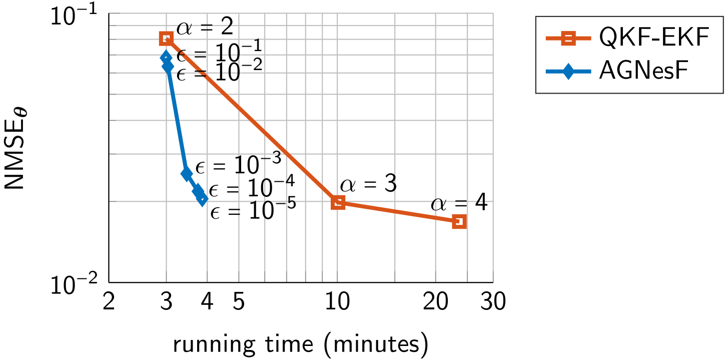


⟶ The nested schemes outperform the augmented-state methods.

⟶ The UKF-EKF is three times faster than SMC-EKF.

[7] Pérez-Vieites and Míguez, "Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models".

Numerical results [ICASSP 2023][8]



1. NGF: QKF-EKF with different number of points/samples, N_{θ} (the greater α, the greater N_{θ}.
2. Adaptive Gaussian nested filter (AGNesF).

[8] Pérez-Vieites and Elvira, "Adaptive Gaussian nested filter for parameter estimation and state tracking in dynamical systems".

Index

Conclusions

We have introduced a **generalized nested methodology**

1. that is flexible. It admits different types of filtering techniques in each layer, leading to a **set of algorithms**.
2. that works recursively.
3. with theoretical guarantees (under general assumptions).

Open to collaborate and discuss possible applications !

- Time-series problems with availability of relatively frequent observations / data
- e.g., remote sensing, energy, ecology, but not only

Conclusions

We have introduced a **generalized nested methodology**

1. that is flexible. It admits different types of filtering techniques in each layer, leading to a **set of algorithms**.
2. that works recursively.
3. with theoretical guarantees (under general assumptions).

Open to collaborate and discuss possible applications !

- Time-series problems with availability of relatively frequent observations / data
- e.g., remote sensing, energy, ecology, but not only

Thank you!

- Pérez-Vieites & Elvira (2023). *Adaptive Gaussian nested filter for parameter estimation and state tracking in dynamical systems*. In ICASSP 2023 [Accepted].
- Pérez-Vieites, Molina-Bulla & Míguez (2022). *Nested smoothing algorithms for inference and tracking of heterogeneous multi-scale state-space systems*. arXiv preprint arXiv:2204.07795.
- Pérez-Vieites & Míguez (2021). *Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models*. Signal Processing, 189, 108295.
- Pérez-Vieites & Míguez (2020). *A nested hybrid filter for parameter estimation and state tracking in homogeneous multi-scale models*. In FUSION 2020 (pp. 1-8). IEEE.
- Pérez-Vieites & Míguez (2020). *Kalman-based nested hybrid filters for recursive inference in state-space models*. In EUSIPCO 2020 (pp. 2468-2472). IEEE.
- Pérez-Vieites, Vilà-Vals, Bugallo, Míguez & Closas (2019). *Second Order Subspace Statistics for Adaptive State-Space Partitioning in Multiple Particle Filtering*. In CAMSAP 2019 (pp. 609-613). IEEE.
- Pérez-Vieites, Mariño & Míguez (2018). *Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems*. Physical Review E, 98(6), 063305.

sarapv.github.io

sara.perez-vieites@imt-nord-europe.fr