A nested hybrid filter for parameter estimation and state tracking in homogeneous multi-scale models

Sara Pérez-Vieites, Joaquín Míguez

Department of Signal Theory & Communications,
Universidad Carlos III de Madrid.
spvieites@tsc.uc3m.es

July 9, 2020

Introduction

We aim at tracking **homogeneous multi-scale systems**. They are of broad interest since they are found in many fields of science (biology, fluid dynamics, chemistry...).
The goal is to estimate the time-evolution of a system governed by

- processes with **different time-scales**,
- that may be **described by diverse laws**,
- and with **cross dependencies** among them

State of the Art Methods

We can find **theoretically-guaranteed solutions** for systems with unknown static parameters and dynamic state variables: a multi-scale problem with only two time scales:

- **Sequential Monte Carlo square (SMC^2)** [Chopin et al., 2011] or **particle Markov chain Monte Carlo (PMCMC)** [Andrieu et al., 2010] aim at computing the joint posterior probability distribution of all the unknown variables and parameters of the system. Unfortunately, they are batch techniques.
- **Nested particle filters (NPFs)** [Crisan et al., 2018] is a scheme with two intertwined layers of Monte Carlo methods that approximates the same distribution but in a recursive way. Then, it is better suited for long sequences of observations but the computational cost is prohibitive.
- **Nested hybrid filters (NHFs)** [S. Pérez-Vieites, 2017] introduce Gaussian filtering techniques in the second layer of the algorithm, reducing considerably the computational cost.

State-space Model

We are interested in systems described by multidimensional stochastic differential equations (SDEs):

$$d\boldsymbol{x} = f_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\theta})dt + g_{\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{\theta})dt + \sigma_x d\boldsymbol{v}, \quad (1)$$

$$d\boldsymbol{z} = f_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{\theta})dt + g_{\boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{\theta})dt + \sigma_z d\boldsymbol{w}, \quad (2)$$

where:

- t denotes time.
- $\boldsymbol{x}(t) \in \mathbb{R}^{d_x}$ are the slow states of the system.
- $\boldsymbol{z}(t) \in \mathbb{R}^{d_z}$ are the fast states of the system.
- $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$ are fixed vector of unknown parameters.
- $f_{\boldsymbol{x}}$, $f_{\boldsymbol{z}}$, $g_{\boldsymbol{x}}$ and $g_{\boldsymbol{z}}$ are possibly non-linear functions.
- σ_x, $\sigma_z > 0$ are known scale parameters that control the intensity of stochastic perturbations.
- $\boldsymbol{v}(t)$, $\boldsymbol{w}(t)$ are vectors of independent standard Wiener processes with dimension d_x and d_z.

Macro-micro Solver

In order to handle both time scales we apply a macro-micro solver [Weinan et al., 2005]. We use different integration steps: Δ_z for $\boldsymbol{z}$ and $\Delta_x \gg \Delta_z$ for $\boldsymbol{x}$:

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \Delta_x\left(f_{\boldsymbol{x}}(\boldsymbol{x}_{n-1}, \boldsymbol{\theta}) + g_{\boldsymbol{x}}(\bar{\boldsymbol{z}}_n, \boldsymbol{\theta})\right) + \sqrt{\Delta_x}\sigma_x \boldsymbol{v}_n, \quad (3)$$

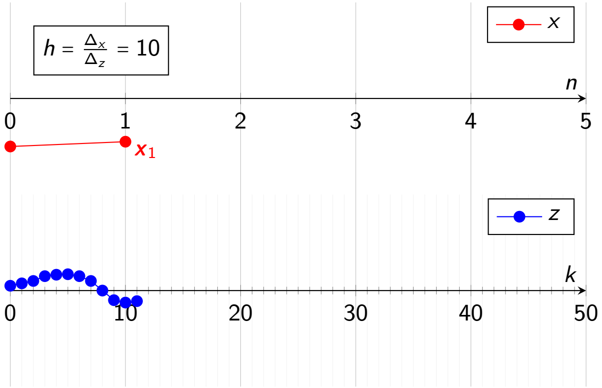
$$\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \Delta_z\left(f_{\boldsymbol{z}}(\boldsymbol{x}_{\lfloor \frac{k-1}{h} \rfloor}, \boldsymbol{\theta}) + g_{\boldsymbol{z}}(\boldsymbol{z}_{k-1}, \boldsymbol{\theta})\right) + \sqrt{\Delta_x}\sigma_x \boldsymbol{w}_k, \quad (4)$$

where

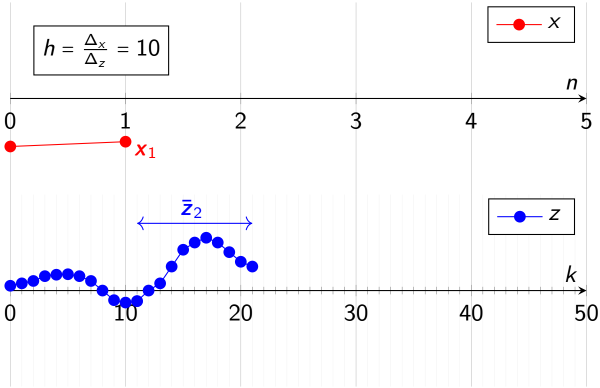
- $n \in \mathbb{N}$ denotes discrete time in the time scale of $\boldsymbol{x}$ ($\boldsymbol{x}_n \simeq \boldsymbol{x}(n\Delta_x)$),
- $k \in \mathbb{N}$ denotes discrete time in the time scale of $\boldsymbol{z}$ ($\boldsymbol{z}_k \simeq \boldsymbol{z}(k\Delta_z)$),
- $h = \frac{\Delta_x}{\Delta_z}$ is the ratio between the two time scales and

$$\bar{\boldsymbol{z}}_n = \frac{1}{h} \sum_{i=h(n-1)+1}^{hn} \boldsymbol{z}_i. \quad (5)$$

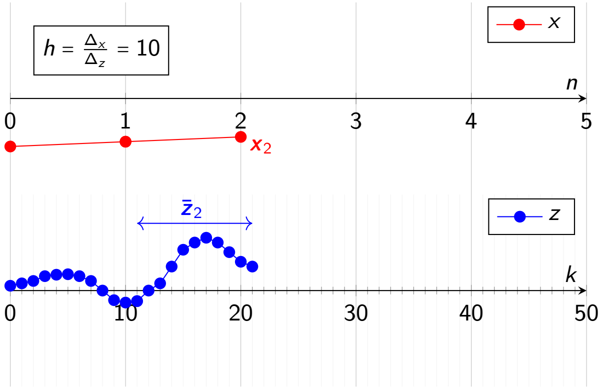
Macro-micro Solver



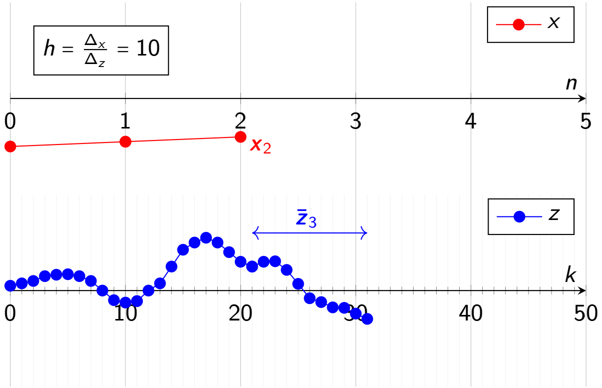
Macro-micro Solver



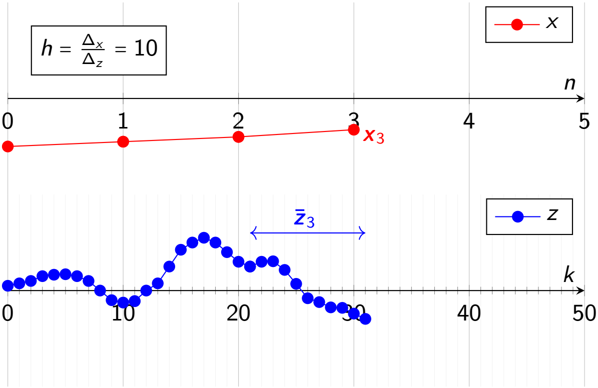
Macro-micro Solver



Macro-micro Solver



Macro-micro Solver



Model Inference

We aim at approximating the joint posterior probability density function (pdf) $p(\boldsymbol{\theta}, \mathbf{x}_{0:n}, \mathbf{z}_{hn}|\mathbf{y}_{1:n})$. Using the chain rule, we can factorize this pdf as

$$p(\mathbf{z}_{hn}, \mathbf{x}_n, \boldsymbol{\theta}|\mathbf{y}_{1:n}) = \underbrace{p(\mathbf{z}_{hn}|\mathbf{x}_{0:n}, \mathbf{y}_{1:n}, \boldsymbol{\theta})}_{3^{rd}\ layer} \underbrace{p(\mathbf{x}_{0:n}|\mathbf{y}_{1:n}, \boldsymbol{\theta})}_{2^{nd}\ layer} \underbrace{p(\boldsymbol{\theta}|\mathbf{y}_{1:n})}_{1^{st}\ layer}$$

→ Each of these pdf's can be handled in a **different layer of computation**.

General methodology

1^{st} layer

$$p(\theta|y_{1:n}) \propto \underbrace{p(y_n|\theta, y_{1:n-1})}_{\text{likelihood of } \theta} \underbrace{p(\theta|y_{1:n-1})}_{\text{posterior pdf at } n-1}$$

2^{nd} layer

$$p(y_n|\theta, y_{1:n-1}) = \int \overbrace{p(y_n|x_{0:n}, y_{1:n-1}, \theta)}^{\text{likelihood of } \theta \text{ and } x_{0:n}} \underbrace{p(x_{0:n}|y_{1:n-1}, \theta)}_{\text{joint predictive pdf of } x_{0:n}} dx_{0:n}$$

3^{rd} layer

$$p(y_n|x_{0:n}, y_{1:n-1}, \theta) = \int \overbrace{p(y_n|z_{hn}, x_n, \theta)}^{\text{likelihood of } \theta,\, x_n \text{ and } z_{hn}} \times \underbrace{p(z_{hn}|x_{0:n}, y_{1:n-1}, \theta)}_{\text{predictive pdf of } z_{hn}} dz_{hn}$$

General methodology

1st layer

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n}) \propto \underbrace{p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1})}_{\text{likelihood of } \boldsymbol{\theta}} \; \underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n-1})}_{\text{posterior pdf at } n-1}$$

2nd layer

$$p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1}) = \int \overbrace{p(\boldsymbol{y}_n|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta})}^{\text{likelihood of } \boldsymbol{\theta} \text{ and } \boldsymbol{x}_{0:n}} \; \underbrace{p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n-1}, \boldsymbol{\theta})}_{\text{joint predictive pdf of } \boldsymbol{x}_{0:n}} \; d\boldsymbol{x}_{0:n}$$

3rd layer

$$p(\boldsymbol{y}_n|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta}) = \int \overbrace{p(\boldsymbol{y}_n|\boldsymbol{z}_{hn}, \boldsymbol{x}_n, \boldsymbol{\theta})}^{\text{likelihood of } \boldsymbol{\theta},\ \boldsymbol{x}_n \text{ and } \boldsymbol{z}_{hn}} \times \underbrace{p(\boldsymbol{z}_{hn}|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta})}_{\text{predictive pdf of } \boldsymbol{z}_{hn}} \, d\boldsymbol{z}_{hn}$$

General methodology

1^{st} layer

$$\color{magenta}{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n})} \propto \underbrace{p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1})}_{\text{likelihood of } \boldsymbol{\theta}} \; \underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n-1})}_{\text{posterior pdf at } n-1}$$

2^{nd} layer

$$p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1}) = \int \overbrace{p(\boldsymbol{y}_n|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta})}^{\text{likelihood of } \boldsymbol{\theta} \text{ and } \boldsymbol{x}_{0:n}} \; \underbrace{p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n-1}, \boldsymbol{\theta})}_{\text{joint predictive pdf of } \boldsymbol{x}_{0:n}} \; d\boldsymbol{x}_{0:n}$$

3^{rd} layer

$$p(\boldsymbol{y}_n|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta}) = \int \overbrace{p(\boldsymbol{y}_n|\boldsymbol{z}_{hn}, \boldsymbol{x}_n, \boldsymbol{\theta})}^{\text{likelihood of } \boldsymbol{\theta},\, \boldsymbol{x}_n \text{ and } \boldsymbol{z}_{hn}}$$

$$\times \underbrace{p(\boldsymbol{z}_{hn}|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta})}_{\text{predictive pdf of } \boldsymbol{z}_{hn}} \, d\boldsymbol{z}_{hn}$$

Multi-scale nested filters: a particular implementation

SMC (N samples) — 1^{st} layer

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

  SMC (J samples) — 2^{nd} layer

    UKF ($L+1$ sigma-points) — 3^{rd} layer

    - Prediction: $\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, 0 \leq l \leq L \longrightarrow p(\boldsymbol{z}_{hn}|\boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

  - Prediction: $\bar{\boldsymbol{x}}_n^{i,j}, 1 \leq j \leq J \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

    UKF ($L+1$ sigma-points) — 3^{rd} layer

    - Likelihood of $\boldsymbol{\theta}$, $\boldsymbol{x}_{0:n}$ and $\boldsymbol{z}_{hn}$: $p(\boldsymbol{y}_n|\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, \bar{\boldsymbol{x}}_n^{i,j}, \bar{\boldsymbol{\theta}}_n^i)$
    - Update: $p(\boldsymbol{z}_{hn}|\bar{\boldsymbol{x}}_n^{i,j}, \boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

  - Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_{0:n}$: $p(\boldsymbol{y}_n|\bar{\boldsymbol{x}}_{0:n}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i) \longrightarrow$ weights
  - Resampling: $\{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\} \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

- Likelihood of $\boldsymbol{\theta}$: $p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1}) \longrightarrow$ weights
- Resampling: $\{\boldsymbol{\theta}_n^i, \{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

$p(\boldsymbol{z}_{hn}, \boldsymbol{x}_n, \boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

Multi-scale nested filters: a particular implementation

SMC (N samples) 1^{st} layer

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

SMC (J samples) 2^{nd} layer

UKF ($L+1$ sigma-points) 3^{rd} layer

- Prediction: $\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, 0 \leq l \leq L \longrightarrow p(\boldsymbol{z}_{hn}|\boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

- Prediction: $\bar{\boldsymbol{x}}_n^{i,j}, 1 \leq j \leq J \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

UKF ($L+1$ sigma-points) 3^{rd} layer

- Likelihood of $\boldsymbol{\theta}$, $\boldsymbol{x}_{0:n}$ and $\boldsymbol{z}_{hn}$: $p(\boldsymbol{y}_n|\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, \bar{\boldsymbol{x}}_n^{i,j}, \bar{\boldsymbol{\theta}}_n^i)$
- Update: $p(\boldsymbol{z}_{hn}|\bar{\boldsymbol{x}}_n^{i,j}, \boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

- Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_{0:n}$: $p(\boldsymbol{y}_n|\bar{\boldsymbol{x}}_{0:n}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i) \longrightarrow$ weights
- Resampling: $\{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\} \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

- Likelihood of $\boldsymbol{\theta}$: $p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1}) \longrightarrow$ weights
- Resampling: $\{\boldsymbol{\theta}_n^i, \{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

$p(\boldsymbol{z}_{hn}, \boldsymbol{x}_n, \boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

Multi-scale nested filters: a particular implementation

SMC (N samples) — 1^{st} layer

- Jittering: $\bar{\theta}_n^i \sim \kappa_N^{\theta_{n-1}^i}$, $1 \leq i \leq N \longrightarrow p(\theta | y_{1:n-1})$

  SMC (J samples) — 2^{nd} layer

  UKF ($L+1$ sigma-points) — 3^{rd} layer

  - Prediction: $\tilde{z}_{hn}^{i,j,l}, 0 \leq l \leq L \longrightarrow p(z_{hn} | x_{0:n-1}^{i,j}, y_{1:n-1}, \bar{\theta}_n^i)$

  - Prediction: $\bar{x}_n^{i,j}, 1 \leq j \leq J \longrightarrow p(x_{0:n} | y_{1:n-1}, \bar{\theta}_n^i)$

  UKF ($L+1$ sigma-points) — 3^{rd} layer

  - Likelihood of θ, $x_{0:n}$ and z_{hn}: $p(y_n | \tilde{z}_{hn}^{i,j,l}, \bar{x}_n^{i,j}, \bar{\theta}_n^i)$
  - Update: $p(z_{hn} | \bar{x}_n^{i,j}, x_{0:n-1}^{i,j}, y_{1:n}, \bar{\theta}_n^i)$

  - Likelihood of θ and $x_{0:n}$: $p(y_n | \bar{x}_{0:n}^{i,j}, y_{1:n-1}, \bar{\theta}_n^i) \longrightarrow$ weights
  - Resampling: $\{x_n^{i,j}, \{z_{hn}^{i,j,l}\}\} \longrightarrow p(x_{0:n} | y_{1:n}, \bar{\theta}_n^i)$

- Likelihood of θ: $p(y_n | \theta, y_{1:n-1}) \longrightarrow$ weights
- Resampling: $\{\theta_n^i, \{x_n^{i,j}, \{z_{hn}^{i,j,l}\}\}\} \longrightarrow p(\theta | y_{1:n})$

$p(z_{hn}, x_n, \theta | y_{1:n})$

Multi-scale nested filters: a particular implementation

SMC (N samples) — 1^{st} layer

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

  SMC (J samples) — 2^{nd} layer

  UKF ($L+1$ sigma-points) — 3^{rd} layer

  - Prediction: $\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, 0 \leq l \leq L \longrightarrow p(\boldsymbol{z}_{hn}|\boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

  - Prediction: $\bar{\boldsymbol{x}}_n^{i,j}, 1 \leq j \leq J \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

  UKF ($L+1$ sigma-points) — 3^{rd} layer

  - Likelihood of $\boldsymbol{\theta}$, $\boldsymbol{x}_{0:n}$ and $\boldsymbol{z}_{hn}$: $p(\boldsymbol{y}_n|\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, \bar{\boldsymbol{x}}_n^{i,j}, \bar{\boldsymbol{\theta}}_n^i)$
  - Update: $p(\boldsymbol{z}_{hn}|\bar{\boldsymbol{x}}_n^{i,j}, \boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

  - Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_{0:n}$: $p(\boldsymbol{y}_n|\bar{\boldsymbol{x}}_{0:n}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i) \longrightarrow$ weights
  - Resampling: $\{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\} \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

- Likelihood of $\boldsymbol{\theta}$: $p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1}) \longrightarrow$ weights
- Resampling: $\{\boldsymbol{\theta}_n^i, \{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

$p(\boldsymbol{z}_{hn}, \boldsymbol{x}_n, \boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

Multi-scale nested filters: a particular implementation

SMC (N samples) — 1^{st} layer

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

  SMC (J samples) — 2^{nd} layer

    UKF ($L+1$ sigma-points) — 3^{rd} layer

    - Prediction:$\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, 0 \leq l \leq L \longrightarrow p(\boldsymbol{z}_{hn}|\boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

  - Prediction: $\bar{\boldsymbol{x}}_n^{i,j}, 1 \leq j \leq J \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i)$

    UKF ($L+1$ sigma-points) — 3^{rd} layer

    - Likelihood of $\boldsymbol{\theta}$, $\boldsymbol{x}_{0:n}$ and $\boldsymbol{z}_{hn}$: $p(\boldsymbol{y}_n|\tilde{\boldsymbol{z}}_{hn}^{i,j,l}, \bar{\boldsymbol{x}}_n^{i,j}, \bar{\boldsymbol{\theta}}_n^i)$
    - Update: $p(\boldsymbol{z}_{hn}|\bar{\boldsymbol{x}}_n^{i,j}, \boldsymbol{x}_{0:n-1}^{i,j}, \boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

  - Likelihood of $\boldsymbol{\theta}$ and $\boldsymbol{x}_{0:n}$: $p(\boldsymbol{y}_n|\bar{\boldsymbol{x}}_{0:n}^{i,j}, \boldsymbol{y}_{1:n-1}, \bar{\boldsymbol{\theta}}_n^i) \longrightarrow$ weights
  - Resampling: $\{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\} \longrightarrow p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}, \bar{\boldsymbol{\theta}}_n^i)$

- Likelihood of $\boldsymbol{\theta}$: $p(\boldsymbol{y}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1:n-1}) \longrightarrow$ weights
- Resampling: $\{\boldsymbol{\theta}_n^i, \{\boldsymbol{x}_n^{i,j}, \{\boldsymbol{z}_{hn}^{i,j,l}\}\}\} \longrightarrow p(\boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

$p(\boldsymbol{z}_{hn}, \boldsymbol{x}_n, \boldsymbol{\theta}|\boldsymbol{y}_{1:n})$

Numerical results

We consider a stochastic two-scale Lorenz 96 model that is described, in continuous time, by the SDEs

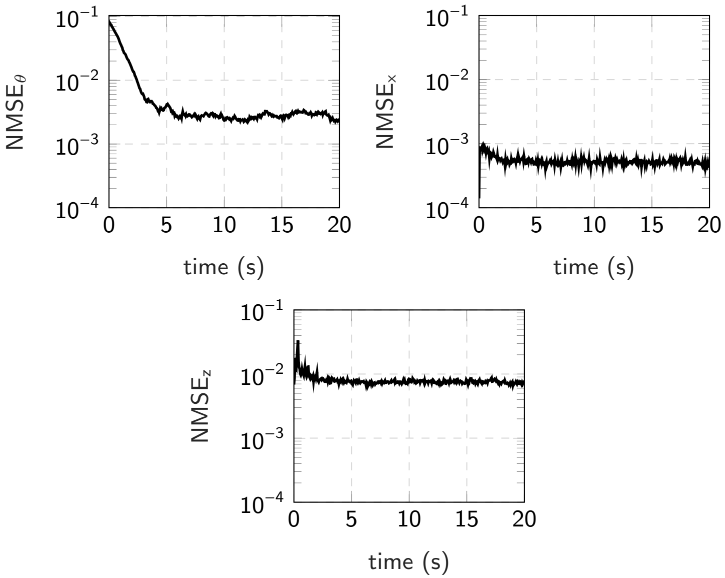
$$dx_j = \left[-x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F - \frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} z_l \right] dt + \sigma_x dv_j, \quad (6)$$

$$dz_l = \left[-CBz_{l+1}(z_{l+2} - z_{l-1}) - Cz_l + \frac{CF}{B} + \frac{HC}{B} x_{\lfloor (l-1)L \rfloor} \right] dt + \sigma_z dw_l, \quad (7)$$

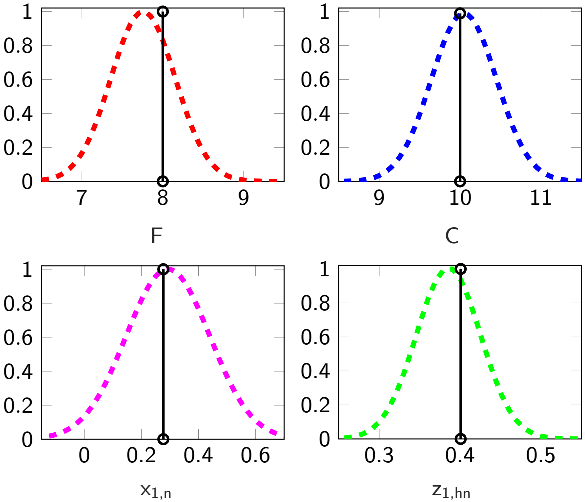
where

- the slow variables, $\boldsymbol{x}$, are a d_x-dimensional vector,
- the fast variables, $\boldsymbol{z}$, are d_z-dimensional $(d_z > d_x)$ and
- we assume static parameters H and B are known, while we need to estimate $\boldsymbol{\theta} = [F, C]^\top$.

NMSE



Approximate posterior density functions



Conclusions

- We have introduced a new recursive methodology for tracking the time evolution and evaluate any static parameters of homogeneous multi-scale systems.
- It is a nested multilayered structure that allows different computation schemes at each layer. Specifically, we have explored the use of sequential Monte Carlo in both first and second layers of the filter, and in the third layer, Gaussian filters (UKF).
- We have analyzed a dynamical system of 3 time-scales (static parameters, slow dynamic state variables and fast dynamic state variables), showing the average performance of the method in terms of estimation errors.

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010).
Particle Markov chain Monte Carlo methods.
Journal of the Royal Statistical Society B, 72:269–342.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2011).
SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates.
Arxiv preprint, arXiv:1101.1528 [stat.CO].

Crisan, D., Miguez, J., et al. (2018).
Nested particle filters for online parameter estimation in discrete-time state-space markov models.
Bernoulli, 24(4A):3039–3086.

S. Pérez-Vieites, I. P. Mariño, J. M. (2017).
A probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems.
arXiv:, 1708.03730 [stat.CO].

Weinan, E., Liu, D., and Vanden-Eijnden, E. (2005).
Analysis of multiscale methods for stochastic differential equations.