Using particle filters to identify nonlinear systems

“The particle filter provides a systematic way of exploring the state space”

Thomas Schönh
Division of Systems and Control
Department of Information Technology
Uppsala University.

Email: thomas.schon@it.uu.se,
www: user.it.uu.se/~thosc112

The talk is based on this paper:

Introduction

A state space model (SSM) consists of a Markov process \( \{x_t\}_{t \geq 1} \) that is indirectly observed via a measurement process \( \{y_t\}_{t \geq 1} \),

\[
\begin{align*}
  x_{t+1} \mid x_t & \sim f_\theta(x_{t+1} \mid x_t, u_t), \\
  y_t \mid x_t & \sim g_\theta(y_t \mid x_t, u_t), \\
  x_1 & \sim \mu_\theta(x_1), \\
  (\theta & \sim \pi(\theta)).
\end{align*}
\]

Identifying the nonlinear SSM: Find \( \theta \) based on \( y_{1:T} \triangleq \{y_1, y_2, \ldots, y_T\} \) (and \( u_{1:T} \)). Hence, the off-line problem.

One of the key challenges: The states \( x_{1:T} \) are unknown.

Aim of the talk: Reveal the structure of the system identification problem arising in nonlinear SSMs and highlight where SMC is used.
Two commonly used problem formulations

Maximum likelihood (ML) formulation – model the unknown parameters as a deterministic variable and solve

$$\hat{\theta}_{\text{ML}} = \arg \max_{\theta \in \Theta} p_\theta(y_{1:T}).$$

Bayesian formulation – model the unknown parameters as a random variable $\theta \sim \pi(\theta)$ and compute

$$p(\theta | y_{1:T}) = \frac{p(y_{1:T} | \theta) \pi(\theta)}{p(y_{1:T})} = \frac{p_\theta(y_{1:T}) \pi(\theta)}{p(y_{1:T})}.$$ 

The combination of ML and Bayes is probably more interesting than we think.
Central object – the likelihood

The likelihood is computed by marginalizing the joint density

\[
p_\theta(x_1:T, y_1:T) = \mu_\theta(x_1) \prod_{t=1}^{T} g_\theta(y_t \mid x_t) \prod_{t=1}^{T-1} f_\theta(x_{t+1} \mid x_t),
\]

w.r.t. the state sequence \(x_1:T\),

\[
p_\theta(y_1:T) = \int p_\theta(x_1:T, y_1:T) dx_1:T.
\]

We are averaging \(p_\theta(x_1:T, y_1:T)\) over all possible state sequences.

Equivalently we have

\[
p_\theta(y_1:T) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{1:t-1}) = \prod_{t=1}^{T} \int g_\theta(y_t \mid x_t) p_\theta(x_t \mid y_{1:t-1}) dx_t.
\]

key challenge
Sequential Monte Carlo

The need for computational methods, such as SMC, is tightly coupled to the intractability of the integrals on the previous slide.

SMC offers numerical approximations to state estimation problems. The particle filter and the particle smoother maintain empirical approximations

\[
\hat{p}_\theta(x_t | y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t), \quad \hat{p}_\theta(x_{1:t} | y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_{1:t}^i}(x_{1:t}).
\]

Converge to the true distributions as \( N \to \infty \).
Using SMC for nonlinear system identification

SMC can be used to approximately

1. Compute the likelihood and its derivatives.
2. Solve state smoothing problems, e.g. compute $p(x_{1:T} \mid y_{1:T})$.
3. Simulate from the smoothing pdf, $\tilde{x}_{1:T} \sim p(x_{1:T} \mid y_{1:T})$.

These three capabilities are key components in implementing various nonlinear system identification strategies.
Identification strategies – overview

**Marginalisation** Deal with $x_{1:T}$ by marginalizing (integrating) them out and view $\theta$ as the only unknown quantity.
- Frequentistic formulation: Prediction Error Method (PEM) and direct maximization of the likelihood.
- Bayesian formulation: the Metropolis Hastings sampler.

**Data augmentation** Deal with $x_{1:T}$ by treating them as auxiliary variables to be estimated along with $\theta$.
- Frequentistic formulation: Expectation Maximization (EM) algorithm.
- Bayesian formulation: the Gibbs sampler.

Only data augmentation strategies in this talk.
1. Problem formulation
2. Identification strategies for nonlinear SSMs
3. Sequential Monte Carlo (SMC)
4. Using SMC as a proposal mechanism within MCMC
5. Data augmentation
   a) Expectation maximization (EM)
   b) Gibbs sampling
6. Snapshots of current research
   a) The Gaussian process SSM and regularization
   b) The nonlinear SSM is just a special case...
Sequential Monte Carlo – particle filter

The particle filter provides an approximation \( p(x_{1:t} \mid y_{1:t}) \), when the state evolves according to an SSM,

\[
\begin{align*}
x_{t+1} \mid x_t &\sim f_{\theta}(x_{t+1} \mid x_t), \\
y_t \mid x_t &\sim g_{\theta}(y_t \mid x_t), \\
x_1 &\sim \mu_{\theta}(x_1).
\end{align*}
\]

The particle filter maintains an empirical distribution made up of \( N \) samples (particles) \( \{x^i_{1:t}\}_{i=1}^N \) and corresponding weights \( \{w^i_{1:t}\}_{i=1}^N \)

\[
\hat{p}(x_{1:t} \mid y_{1:t}) = \sum_{i=1}^N w^i_t \delta_{x^i_{1:t}}(x_{1:t}).
\]

“Think of each particle as one simulation of the system state. Keep the ones that best explains the measurements.”
The particle filter – toy problem

Consider a toy 1D localization problem.

Dynamic model:

\[ x_{t+1} = x_t + u_t + v_t, \]

where \( x_t \) denotes position, \( u_t \) denotes velocity (known), \( v_t \sim \mathcal{N}(0, 5) \) denotes an unknown disturbance.

Measurements:

\[ y_t = h(x_t) + e_t. \]

where \( h(\cdot) \) denotes the world model (here the terrain height) and \( e_t \sim \mathcal{N}(0, 1) \) denotes an unknown disturbance.

The same idea has been used for the Swedish fighter JAS 39 Gripen,

The particle filter – toy problem

Highlights two key capabilities of the PF:

1. Automatically handles an unknown and dynamically changing number of hypotheses.

2. Work with nonlinear/non-Gaussian models.
Aim: Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.
Sequential Monte Carlo – particle filter

SMC = resampling + sequential importance sampling

1. **Resampling:** \( \mathbb{P}(a_t^i = j) = \bar{w}_t^j / \sum_l \bar{w}_t^l \).

2. **Propagation:** \( x_t^i \sim f_\theta(x_t | x_{1:t-1}^{a_t^i}) \) and \( x_{1:t}^i = \{x_{1:t-1}^{a_t^i}, x_t^i\} \).

3. **Weighting:** \( \bar{w}_t^i = W_t(x_t^i) = g_\theta(y_t | x_t) \).

The **ancestor indices** \( \{a_t^i\}_{i=1}^N \) are very **useful** auxiliary variables! They make the stochasticity of the resampling step explicit.
Sequential Monte Carlo – particle filter

Let

\[ \mathbf{x}_t \triangleq \{ x_1^t, \ldots, x_N^t \}, \quad \mathbf{a}_t \triangleq \{ a_1^t, \ldots, a_N^t \} \]

denote all particles and ancestor indices generated at time \( t \).

The SMC algorithm generates a single realization of a collection of random variables

\[ \{ \mathbf{x}_{1:T}, \mathbf{a}_{2:T} \} \in X^{NT} \times \{ 1, \ldots, N \}^{N(T-1)} \]

distributed according to

\[
\psi(\mathbf{x}_{1:T}, \mathbf{a}_{2:T}) \triangleq \prod_{i=1}^{N} q_1(x_1^i) \prod_{t=2}^{T} \prod_{i=1}^{N} M_t(a_t^i, x_t^i),
\]

where

\[
M_t(a_t, x_t) = \frac{\bar{w}_{t-1}^{a_t}}{\sum_l \bar{w}_{t-1}^l} f_t(x_t \mid x_{1:t-1}^{a_t}).
\]
The particle system degenerates (illustration)

Clearly motivates the need for particle smoothers.

Self-contained introduction to particle smoothing using BS and AS

Exact approximation: $p(\theta \mid y_{1:T})$ is recovered exactly, despite the fact that we employ an SMC approximation of the likelihood using a finite number of particles $N$.

This is one of the members in the particle MCMC (PMCMC) family introduced by Christophe Andrieu, Arnaud Doucet and Roman Holenstein, Particle Markov chain Monte Carlo methods, *Journal of the Royal Statistical Society: Series B*, 72:269-342, 2010.

The idea underlying PMCMC is to make use of SMC algorithms to propose (simulate) state trajectories $x_{1:T}$. These state trajectories are then used within standard MCMC algorithms.

1. Particle Metropolis Hastings
2. Particle Gibbs
1. Problem formulation
2. Identification strategies for nonlinear SSMs
3. Sequential Monte Carlo (SMC)
4. Using SMC as a proposal mechanism within MCMC
5. **Data augmentation**
   a) Expectation maximization (EM)
   b) Gibbs sampling
6. Snapshots of current research
   a) The Gaussian process SSM and regularization
   b) The nonlinear SSM is just a special case...
Identification strategy – data augmentation

**Motivation:** If we had access to the complete likelihood

\[ p_\theta(x_{1:T}, y_{1:T}) = \mu_\theta(x_1) \prod_{t=1}^{T} g_\theta(y_t | x_t) \prod_{t=1}^{T-1} f_\theta(x_{t+1} | x_t) \]

the problem would be much easier.

**Key idea:** Treat the state sequence \( x_{1:T} \) as an *auxiliary variable* that is estimated together with \( \theta \).

The data augmentation strategy breaks the original problem into two new and closely linked problems.

Intuitively the data augmentation strategy amounts to iterating between updating \( x_{1:T} \) and \( \theta \).
Maximum likelihood (ML) formulation – model the unknown parameters as a deterministic variable and solve

\[ \hat{\theta}_{\text{ML}} = \arg \max_{\theta \in \Theta} p_{\theta}(y_{1:T}). \]

The expectation maximization algorithm is an iterative approach to compute ML estimates of unknown parameters (\( \theta \)) in probabilistic models involving latent variables (the state trajectory \( x_{1:T} \)).

Expectation maximization (EM) employs the complete likelihood \( p_{\theta}(x_{1:T}, y_{1:T}) \) as a substitute for the observed likelihood \( p_{\theta}(y_{1:T}) \),

\[ p_{\theta}(x_{1:T}, y_{1:T}) = p_{\theta}(x_{1:T} \mid y_{1:T}) p_{\theta}(y_{1:T}). \]
EM works by iteratively computing

\[
Q(\theta, \theta_k) = \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta_k}(x_{1:T} \mid y_{1:T}) \, dx_{1:T}
\]

and then maximizing \(Q(\theta, \theta_k)\) w.r.t. \(\theta\).

**Problem:** The E-step requires us to solve a smoothing problem, i.e. to compute an expectation under \(p_{\theta_k}(x_{1:T} \mid y_{1:T})\).

SMC is used to approximate the smoothing pdf \(p_{\theta_k}(x_{1:T} \mid y_{1:T})\).
Bayesian formulation – model the unknown parameters as a random variable $\theta \sim \pi(\theta)$ and compute

$$p(\theta | y_{1:T}) = \frac{p(y_{1:T} | \theta)\pi(\theta)}{p(y_{1:T})} = \frac{p_{\theta}(y_{1:T})\pi(\theta)}{p(y_{1:T})}. $$

Gibbs sampling amounts to sequentially sampling from conditionals of the target distribution $p(\theta, x_{1:T} | y_{1:T})$.

A (blocked) example:

- Draw $\theta[m] \sim p(\theta | x_{1:T}[m-1], y_{1:T})$; Ok!
- Draw $x_{1:T}[m] \sim p(x_{1:T} | \theta[m], y_{1:T})$. Hard!

SMC is used to simulate from the smoothing pdf $p(x_{1:T} | y_{1:T})$. 
Sampling based on SMC

With $\mathbb{P}(x_{1:T}^* = x_{1:T}^i) \propto w^i_T$ we get, $x_{1:T}^*$ \text{approx.} \sim p(x_{1:T} | \theta, y_{1:T})$. 

![Graph showing state versus time](image_url)
Problems and a solution

Problems with this approach,

- Based on a PF $\Rightarrow$ approximate sample.
- Does not leave $p(x_{1:T} | \theta, y_{1:T})$ invariant!
- Relies on large $N$ to be successful.
- A lot of wasted computations.

To get around these problems,

Use a conditional particle filter (CPF). One pre-specified reference trajectory is retained throughout the sampler.

Particle Gibbs (PG)

The idea underlying Particle Gibbs (PG) is to make use of a certain SMC sampler to construct a Markov kernel leaving the joint smoothing distribution \( p(x_{1:T} \mid \theta, y_{1:T}) \) invariant.

This Markov kernel is then used within a standard Gibbs that operates on a non-standard space.

SMC is used to build an MCMC kernel with \( p(x_{1:t} \mid \theta, y_{1:t}) \) as its stationary distribution \textbf{without} introducing any systematic errors!
Three SMC samplers leaving \( p(x_{1:T} | \theta, y_{1:T}) \) invariant:

1. **Conditional particle filter (CPF)**
   

2. **CPF with backward simulation (CPF-BS)**
   
   

3. **CPF with ancestor sampling (CPF-AS)**
   
Conditional particle filter (CPF)

Let $x'_{1:T} = (x'_1, \ldots, x'_T)$ be a fixed reference trajectory.

- At each time $t$, sample $N - 1$ particles in the standard way.
- Set the $N^\text{th}$ particle deterministically: $x^N_t = x'_t$.

CPF causes us to degenerate to the something that is very similar to the reference trajectory, resulting in slow mixing.
CPF vs. CPF-AS – motivation

BS is problematic for models with more intricate dependencies.

**Reason:** Requires complete trajectories of the latent variable in the backward sweep.

**Solution:** Modify the computation to achieve the same effect as BS, but **without** an explicit backwards sweep.

**Implication:** Ancestor sampling opens up for inference in a wider class of models, e.g. non-Markovian SSMs, PGMs and BNP models.

Ancestor sampling is conceptually similar to backward simulation, but instead of using separate forward and backward sweeps, we achieve the same effect in a **single forward sweep**.
CPF-AS – intuition

Let $x'_{1:T} = (x'_1, \ldots, x'_T)$ be a fixed reference trajectory.

- At each time $t$, sample $N - 1$ particles in the standard way.
- Set the $N^{th}$ particle deterministically: $x^N_t = x'_t$.
- Generate an artificial history for $x^N_t$ by ancestor sampling.

CPF-AS causes us to degenerate to something that is very different from the reference trajectory, resulting in better mixing.
Example – semiparametric Wiener model

\[
\begin{aligned}
    x_{t+1} &= \Gamma x_t + v_t, \\
    z_t &= C x_t, \\
    y_t &= g(z_t) + e_t,
\end{aligned}
\]

Parametric LGSS and a nonparametric static nonlinearity:

\[
\begin{aligned}
    x_{t+1} &= (A B) \begin{pmatrix} x_t \\ u_t \end{pmatrix} + v_t, \\
    v_t &\sim \mathcal{N}(0, Q), \\
    z_t &= C x_t, \\
    y_t &= g(z_t) + e_t, \\
    e_t &\sim \mathcal{N}(0, R).
\end{aligned}
\]
Example – semiparametric Wiener model

“Parameters”: \( \theta = \{ A, B, Q, g(\cdot), r \} \).

**Bayesian model** specified by priors

- Conjugate priors for \( \Gamma = [A B], Q \) and \( r \),
  - \( p(\Gamma, Q) = \text{Matrix-normal inverse-Wishart} \)
  - \( p(r) = \text{inverse-Wishart} \)
- Gaussian process prior on \( g(\cdot) \),
  \[
  g(\cdot) \sim \mathcal{GP}(z, k(z, z')).
  \]

**Inference** using PGAS with \( N = 15 \) particles. \( T = 1000 \) measurements. We ran 15 000 MCMC iterations and discarded 5 000 as burn-in.
Example – semiparametric Wiener model

Show movie

Bode diagram of the 4th-order linear system. Estimated mean (dashed black), true (solid black) and 99\% credibility intervals (blue).

Static nonlinearity (non-monotonic), estimated mean (dashed black), true (black) and the 99\% credibility intervals (blue).

Outline

1. Problem formulation
2. Identification strategies for nonlinear SSMs
3. Sequential Monte Carlo (SMC)
4. Using SMC as a proposal mechanism within MCMC
5. Data augmentation
   a) Expectation maximization (EM)
   b) Gibbs sampling
6. Snapshots of current research
   a) The Gaussian process SSM and regularization
   b) The nonlinear SSM is just a special case...
A nonparametric SSM based on GPs

Consider the Gaussian Process SSM (GP-SSM):

\[
\begin{align*}
    x_{t+1} &= f(x_t) + w_t, \quad \text{s.t.} \quad f(x) \sim \mathcal{GP}(0, \kappa_\theta, f(x, x')) \\
    y_t &= g(x_t) + e_t, \quad \text{s.t.} \quad g(x) \sim \mathcal{GP}(0, \kappa_\theta, g(x, x'))
\end{align*}
\]

The model functions \( f \) and \( g \) are assumed to be realizations from a Gaussian process prior and \( w_t \sim \mathcal{N}(0, Q), e_t \sim \mathcal{N}(0, R) \).

We can now find the posterior distribution

\[
p(f, g, Q, R, \theta \mid y_{1:T}),
\]

by making use of PGAS.

Regularizing nonlinear SSMs

Regularization allows us to tune the model complexity.

Place a GP-SSM prior on the nonlinear transition \( f(\cdot) \),

\[
    f(x) = \sum_{k=1}^{\infty} \omega^k \phi^k(x) \approx \sum_{k=1}^{m} \omega^k \phi^k(x)
\]

Regularize using a prior on the weights, e.g.,

\[
    p([\omega^1, \ldots, \omega^m]) = \mathcal{N}(0, P^{-1}).
\]

PSAEM

The nonlinear SSM is just a special case...

A **graphical model** is a probabilistic model where a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the conditional independency structure between random variables,

1. a set of **vertices** $\mathcal{V}$ (nodes) represents the random variables
2. a set of **edges** $\mathcal{E}$ containing elements $(i, j) \in \mathcal{E}$ connecting a pair of nodes $(i, j) \in \mathcal{V}$

\[
p(x_0:T, y_1:T) = p(x_0) \prod_{t=1}^{N} p(x_t | x_{t-1}) \prod_{t=1}^{N} p(y_t | x_t).
\]
The nonlinear SSM is just a special case...

Constructing an artificial sequence of intermediate target distributions for an SMC sampler is a powerful (and quite possibly underutilized) idea.


## Conclusion

<table>
<thead>
<tr>
<th>ML Bayesian</th>
<th>Marginalization</th>
<th>Data augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct optimization</td>
<td>Expectation Maximization</td>
<td></td>
</tr>
<tr>
<td>Metropolis Hastings</td>
<td>Gibbs sampling</td>
<td></td>
</tr>
</tbody>
</table>

SMC is used to realize all of these approaches for nonlinear SSMs.

SMC can be used to approximately

1. Compute the likelihood and its derivatives.
2. Solve state smoothing problems, e.g. compute $p(x_{1:T} | y_{1:T})$.
3. Simulate from the smoothing pdf, $\tilde{x}_{1:T} \sim p(x_{1:T} | y_{1:T})$.

---

1 day tutorial at ICASSP (Shanghai) on March 20, 2016,


A lot of interesting research that remains to be done!!
Open tenure track position in ML

If you are interested, send an e-mail to Thomas Schön
thomas.schon@it.uu.se

If you know someone else who might be interested feel free to help us spread this information as much as you can and want.

Deadline for applications: November 30, 2015. Details here:
www.uu.se/en/about-uu/join-us/details/?positionId=75817

2. I am also looking for new PhD students. Feel free to spread this information as well!! Topic: Nonlinear inference in system identification and machine learning.

Deadline for applications: December 20, 2015.
www.uu.se/en/about-uu/join-us/details/?positionId=78032
Let $\varphi : X \mapsto \mathbb{R}$ be some test function of interest. The expectation

$$E_\theta [\varphi(x_t) \mid y_{1:t}] = \int \varphi(x_t) p_\theta(x_t \mid y_{1:t}) \, dx_t,$$

can be estimated by the particle filter

$$\hat{\varphi}_t^N \triangleq \sum_{i=1}^{N} w_t^i \varphi(x_t^i).$$

The **CLT** governing the convergence of this estimator states

$$\sqrt{N} (\hat{\varphi}_t^N - E_\theta [\varphi(x_t) \mid y_{1:t}]) \xrightarrow{d} N(0, \sigma^2_t(\varphi)).$$

The **likelihood estimate** $\hat{p}_\theta(y_{1:t}) = \prod_{s=1}^{t} \left\{ \frac{1}{N} \sum_{i=1}^{N} \bar{w}_s^i \right\}$ from the PF is **unbiased**, $E_{\psi_\theta} [\hat{p}_\theta(y_{1:t})] = p_\theta(y_{1:t})$ for any value of $N$ and there are **CLTs available** as well.
One realisation from $x[k + 1] = 0.8x[k] + v[k]$ where $v[k] \sim \mathcal{N}(0, 1)$. Initialise in $x[0] = -40$. This will eventually generate samples from the following stationary distribution:

$$\pi^s(x) = \mathcal{N}\left(x \mid 0, \frac{1}{1 - 0.8^2}\right)$$

as $t \to \infty$. 
The true stationary distribution is showed in black and the empirical histogram obtained by simulating the Markov chain $x[k + 1] = 0.8x[k] + v[k]$ is plotted in gray.

The initial 1000 samples are discarded (burn-in).
Micro: MCMC

In the example, the Markov chain was fully specified and the stationary distribution could be expressed in closed form.

Not possible in the situations we are interested in, but we can (since 2010) find a Markov chain that has the target distribution (e.g. $p(\theta | y_{1:T})$) as its stationary distribution.

Two constructive ways of doing this are:

1. Metropolis Hastings (MH) algorithm
2. Gibbs sampling

Markov chain Monte Carlo (MCMC) methods allow us to generate samples from a target distribution by simulating a Markov chain which has the target distribution as its stationary distribution.