Learning dynamical systems using particle filters

UPPSALA UNIVERSITET

Thomas Schön

Division of Systems and Control
Department of Information Technology
Uppsala University

Joint work with (alphabetical order): **Roger Frigola** (University of Cambridge), **Fredrik Gustafsson** (Linköping University), **Michael I. Jordan** (UC Berkeley), **Johan Kihlberg** (Semcon), **Fredrik Lindsten** (Linköping University), **Lennart Ljung** (Linköping University), **Brett Ninness** (University of Newcastle, Australia), **Per Johan Nordlund** (Saab), **Carl E. Rasmussen** (University of Cambridge), **Simon Tegelid** (Xdin), **Adrian Wills** (MRA, Newcastle, Australia).

Dynamical systems are everywhere!

Some of the dynamical systems we have been working with,

We first have to learn the models. Then we can use them.

Outline

1. Probabilistic models of dynamical systems
2. State inference
3. Sequential Monte Carlo (SMC), the particle filter
    a) Key idea
    b) indoor localization example
    c) UAV localization example
4. Learning dynamical models
    a) Maximum Likelihood (ML) identification (very brief)
    b) Bayesian identification ((P)MCMC)
    c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to **both** the maximum likelihood and the Bayesian approaches.

Outline

1. Probabilistic models of dynamical systems
2. State inference
3. Sequential Monte Carlo (SMC), the particle filter
    a) Key idea
    b) indoor localization example
    c) UAV localization example
4. Learning dynamical models
    a) Maximum Likelihood (ML) identification (very brief)
    b) Bayesian identification ((P)MCMC)
    c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to **both** the maximum likelihood and the Bayesian approaches.

Probabilistic models of dynamical systems

Basic representation: Two discrete-time stochastic processes,

- $\{x_t\}_{t \geq 1}$ representing the state of the system.
- $\{y_t\}_{t \geq 1}$ representing the measurements from the sensors.

The probabilistic model is described using two (f and g) probability density functions (PDFs):

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_\theta(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim g_\theta(y_t \mid x_t). \end{aligned}$$

Model = PDF

This type of model is referred to as a **state space model (SSM)** or a **hidden Markov model (HMM)**.

State inference in dynamical systems (I/III)

Aim: Compute a probabilistic representation of our knowledge of the state, based on information that is present in the measurements.

The **filtering PDF**

$$p(x_t \mid y_{1:t}),$$

provides a representation of the uncertainty about the state at time t, given all the measurements up to time t.

The obvious question is now, how do we compute this object?

$$\begin{aligned} p(x_t \mid y_{1:t}) = p(x_t \mid y_t, y_{1:t-1}) &= \frac{p(y_t \mid x_t, y_{1:t-1}) p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \\ &= \frac{g(y_t \mid x_t) p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}. \end{aligned}$$

State inference in dynamical systems (II/III)

Apparently we need an expression also for the prediction PDF

$$p(x_t \mid y_{1:t-1}).$$

Using marginalization we have

$$\begin{aligned} p(x_t \mid y_{1:t-1}) &= \int p(x_t, x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1} \\ &= \int \underbrace{p(x_t \mid x_{t-1}, y_{1:t-1})}_{f(x_t \mid x_{t-1})} p(x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1}. \end{aligned}$$

Hence, the prediction PDF is given by

$$p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1}.$$

State inference in dynamical systems (III/III)

We have now showed that for the nonlinear SSM

$$x_{t+1} \mid x_t \sim f(x_{t+1} \mid x_t),$$
$$y_t \mid x_t \sim g_\theta(y_t \mid x_t).$$

the uncertain information that we have about the state is captured by the filtering PDF, which we compute sequentially using a **measurement update**

$$p(x_t \mid y_{1:t}) = \frac{\overbrace{g(y_t \mid x_t)}^{\text{measurement model}} \quad \overbrace{p(x_t \mid y_{1:t-1})}^{\text{prediction PDF}}}{p(y_t \mid y_{1:t-1})}$$

and a **time update**

$$p(x_t \mid y_{1:t-1}) = \int \underbrace{f(x_t \mid x_{t-1})}_{\text{dynamical model}} \underbrace{p(x_{t-1} \mid y_{1:t-1})}_{\text{filtering PDF}} \mathrm{d}x_{t-1}.$$

State inference – simple special case

Consider the following special case (Linear Gaussian State Space (LGSS) model)

$$
\begin{aligned}
x_{t+1} &= Ax_t + Bu_t + v_t, & v_t &\sim \mathcal{N}(0, Q_t),\\
y_t &= Cx_t + Du_t + e_t, & e_t &\sim \mathcal{N}(0, R_t).
\end{aligned}
$$

or, equivalently,

$$
\begin{aligned}
x_{t+1} \mid x_t \sim f(x_{t+1} \mid x_t) &= \mathcal{N}(x_{t+1} \mid Ax_t + Bu_t, Q_t),\\
y_t \mid x_t \sim g(y_t \mid x_t) &= \mathcal{N}(y_t \mid Cx_t + Du_t, R_t).
\end{aligned}
$$

It is now straightforward to show that the solution to the time update and measurement update equations is given by the **Kalman filter**,

$$
\begin{aligned}
p(x_t \mid y_{1:t}) &= \mathcal{N}(x_t \mid \widehat{x}_{t|t}, P_{t|t}),\\
p(x_{t+1} \mid y_{1:t}) &= \mathcal{N}(x_{t+1} \mid \widehat{x}_{t+1|t}, P_{t+1|t}).
\end{aligned}
$$

State inference – interesting case

Obvious question: what do we do in an interesting case, for example when we have a nonlinear model with non-Gaussian noise?

1. Need a general representation of the filtering PDF
2. Try to solve the equations

$$p(x_t \mid y_{1:t}) = \frac{g(y_t \mid x_t) p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})},$$

$$p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1},$$

as accurately as possible.

Outline

1. Probabilistic models of dynamical systems
2. State inference
3. **Sequential Monte Carlo (SMC), the particle filter**
    a) **Key idea**
    b) **indoor localization example**
    c) **UAV localization example**
4. Learning dynamical models
    a) Maximum Likelihood (ML) identification (very brief)
    b) Bayesian identification ((P)MCMC)
    c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to **both** the maximum likelihood and the Bayesian approaches.

The particle filter

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

$$
\begin{aligned}
x_{t+1} \mid x_t &\sim f_t(x_{t+1} \mid x_t), \\
y_t \mid x_t &\sim g_t(y_t \mid x_t), \\
x_1 &\sim \mu(x_1).
\end{aligned}
$$

The particle filter maintains an empirical distribution made up of N samples (particles) $\{x_t^i\}_{i=1}^N$ and corresponding weights $\{w_t^i\}_{i=1}^N$

$$\widehat{p}^N(x_t \mid y_{1:t}) = \sum_{i=1}^N w_t^i \delta_{x_t^i}(x_t).$$

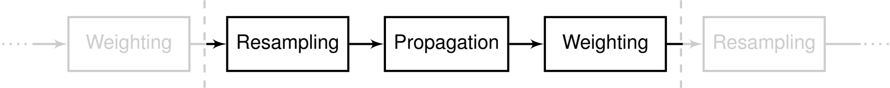
“Think of each particle as one simulation of the system state. Only keep the good ones.”

The particle filter - Very brief history

The particle filter has been around for roughly **20** years.

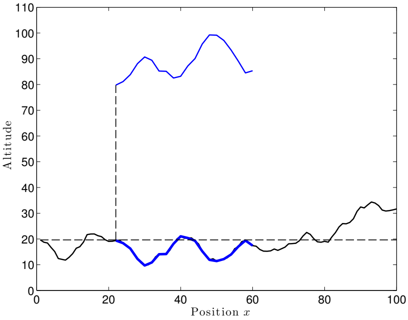
The use of particle methods for nonlinear system identification started to take off some **5** years ago.

Now this is a very active problem (and solution) within many fields.



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:

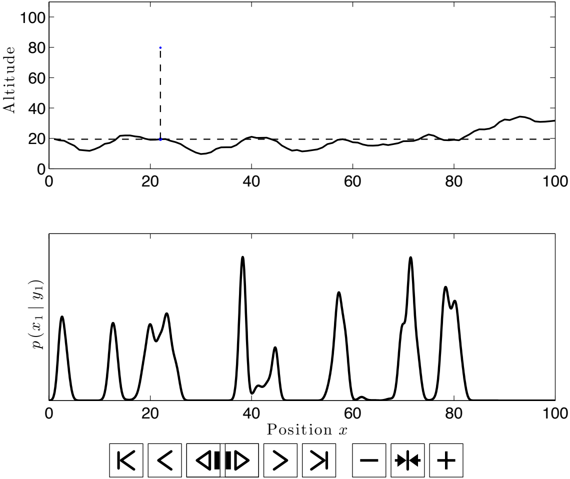
$$y_t = h(x_t) + e_t.$$

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. Details are available in,

Thomas Schön, Fredrik Gustafsson, and Per-Johan Nordlund. **Marginalized particle filters for mixed linear/nonlinear state-space models**. *IEEE Transactions on Signal Processing*, 53(7):2279-2289, July 2005.

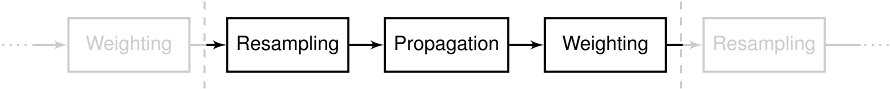
The particle filter – toy problem



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.

The particle filter



1. **Resampling:** $\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N \to \{\tilde{x}_{t-1}^i, 1/N\}_{i=1}^N$.
2. **Propagation:** $x_t^i \sim q_t(x_t \mid \tilde{x}_{t-1}^i)$.
3. **Weighting:** $w_t^i = W_t(x_t^i, y_t)$.

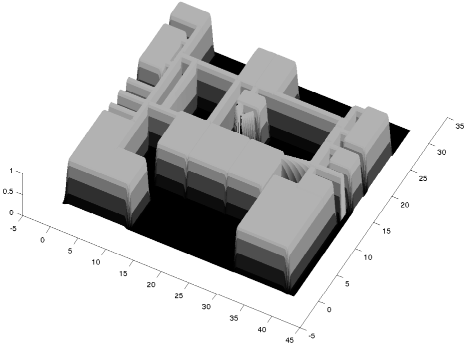
The result is a new weighted set of particles $\{x_t^i, w_t^i\}_{i=1}^N$ targeting $p(x_t \mid y_{1:t})$.

A systematic way of obtaining approximations that converge

Xiao-Li Hu, Thomas B. Schön and Lennart Ljung. **A basic convergence result for particle filtering**. *IEEE Transactions on Signal Processing*, 56(4):1337-1348, April 2008.

Example 1 – indoor localization

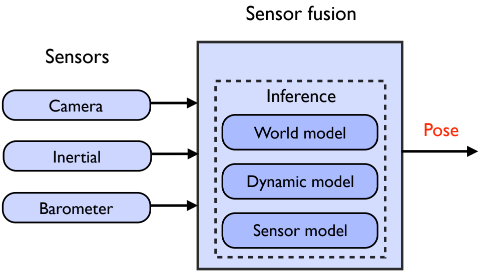
Aim: Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.



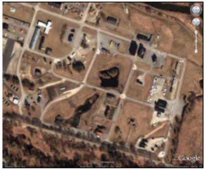
Show movie

Example 2 – UAV localization (I/III)

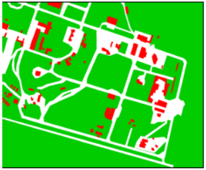
Aim: Compute the position and orientation of a helicopter by exploiting the information present in Google maps images of the operational area.



Example 2 – UAV localization (II/III)



Map over the operational environment obtained from Google Earth.



Manually classified map with grass, asphalt and houses as pre-specified classes.

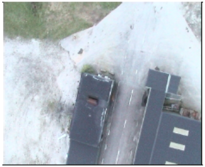
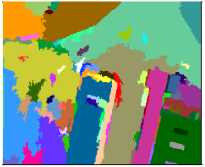
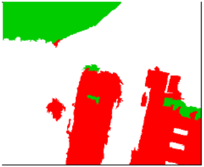


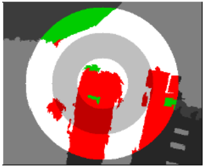
Image from on-board camera



Extracted superpixels

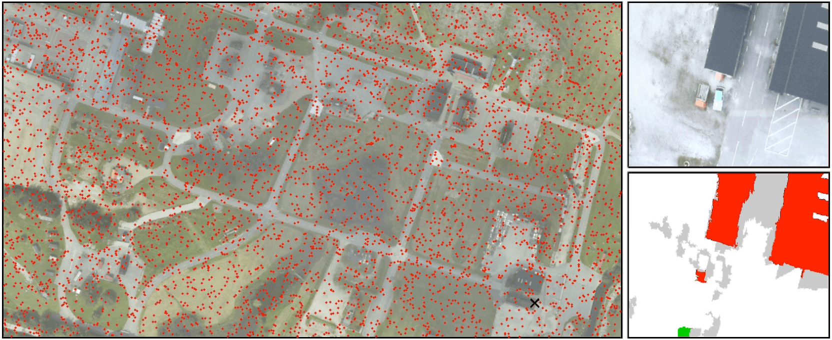


Superpixels classified as grass, asphalt or house



Three circular regions used for computing class histograms

Example 2 – UAV localization (III/III)



Show movie

Fredrik Lindsten, Jonas Callmer, Henrik Ohlsson, David Törnqvist, Thomas B. Schön, Fredrik Gustafsson, **Geo-referencing for UAV Navigation using Environmental Classification**. *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010.

Outline

1. Probabilistic models of dynamical systems
2. State inference
3. Sequential Monte Carlo (SMC), the particle filter
    a) Key idea
    b) indoor localization example
    c) UAV localization example
4. **Learning dynamical models**
    a) Maximum Likelihood (ML) identification (very brief)
    b) Bayesian identification ((P)MCMC)
    c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to **both** the maximum likelihood and the Bayesian approaches.

Learning nonlinear dynamical systems from data

A state space model (SSM) consists of a Markov process $\{x_t\}_{t\geq 1}$ and a measurement process $\{y_t\}_{t\geq 1}$, related according to

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_t(x_{t+1} \mid x_t), \\ y_t \mid x_t &\sim g_t(y_t \mid x_t), \\ x_1 &\sim \mu(x_1). \end{aligned} \qquad \begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t), \\ y_t \mid x_t &\sim g_{\theta,t}(y_t \mid x_t), \\ x_1 &\sim \mu_\theta(x_1). \end{aligned}$$

We observe

$$y_{1:T} \triangleq \{y_1, \ldots, y_T\},$$

(leaving the latent variables $x_{1:T}$ unobserved).

Identification problem: Find f, g, μ (or θ) based on $y_{1:T}$.

Strategies for inference in latent variable models

Alternate between updating θ and updating $x_{1:T}$.

Frequentists:

- Find $\widehat{\theta}_{\text{ML}} = \arg\max_{\theta} p_{\theta}(y_{1:T})$.
- Use e.g. the expectation maximization (EM) algorithm.

Bayesians:

- Find $p(\theta \mid y_{1:T})$.
- Use e.g. Gibbs or Metropolis-Hastings sampling.

Maximum likelihood – nonlinear SSMs

Maximum likelihood (ML) amounts to solving,

$$\widehat{\theta}^{\text{ML}} = \arg\max_{\theta} \log p_{\theta}(y_{1:T}) = \arg\max_{\theta} \sum_{t=1}^{T} \log p_{\theta}(y_t \mid y_{1:t-1}),$$

where

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t), \\ y_t \mid x_t &\sim g_{\theta,t}(y_t \mid x_t), \\ x_1 &\sim \mu_{\theta}(x_1). \end{aligned}$$

Can be solved by combining the Expectation Maximization (EM) algorithm with a particle smoother.

Thomas B. Schön, Adrian Wills and Brett Ninness. **System Identification of Nonlinear State-Space Models**. *Automatica*, 47(1):39-49, January 2011.

Problem formulation – Bayesian

Consider a Bayesian SSM (θ is now a random variable with a prior density $p(\theta)$)

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t), \\ y_t \mid x_t &\sim g_{\theta,t}(y_t \mid x_t), \\ x_1 &\sim \mu_\theta(x_1), \\ \theta &\sim p(\theta). \end{aligned}$$

Identification problem: Compute the posterior $p(\theta, x_{1:T} \mid y_{1:T})$, or one of its marginals.

The **key challenge** is that there is no closed form expression available for the posterior.

Fredrik Lindsten and Thomas B. Schön, **Backward simulation methods for Monte Carlo statistical inference**, *Foundations and Trends in Machine Learning*, 6(1):1-143, 2013.

Gibbs sampler for SSMs

Markov chain Monte Carlo (MCMC) methods allow us to generate samples from a target distribution by simulating a Markov chain.

Gibbs sampling (blocked) for SSMs amounts to iterating

- Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$,
- Draw $x_{1:T}[m] \sim p(x_{1:T} \mid \theta[m], y_{1:T})$.

The above procedure results in a Markov chain,

$$\{\theta[m], x_{1:T}[m]\}_{m \geq 1}$$

with $p(\theta, x_{1:T} \mid y_T)$ as its stationary distribution!

Gibbs sampler for a general SSM

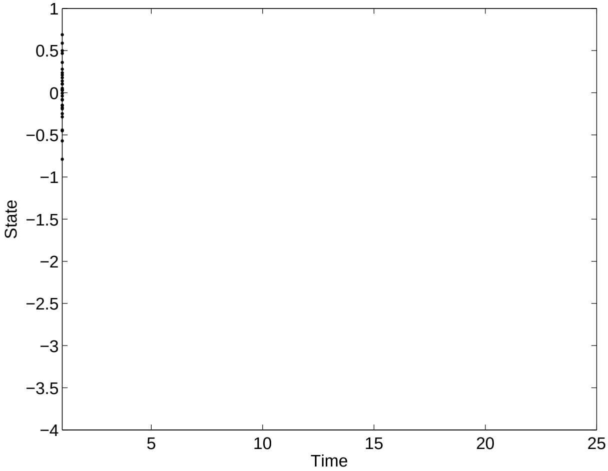
What would a Gibbs sampler for a general nonlinear/non-Gaussian SSM look like?

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

Problem: $p(x_{1:T} \mid \theta[m], y_{1:T})$ not available!

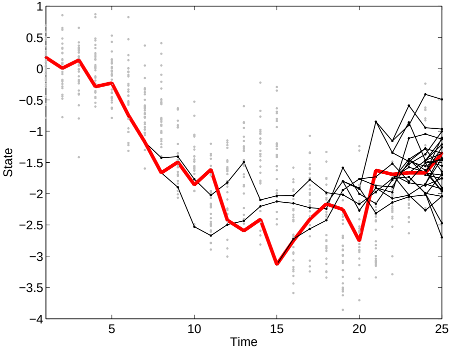
Idea: Approximate $p(x_{1:T} \mid \theta[m], y_{1:T})$ using a sequential Monte Carlo method!

Sampling based on the PF (I/II)



Sampling based on the PF (II/II)

With $P(x'_{1:T} = x^i_{1:T}) \propto w^i_T$ we get, $x'_{1:T} \overset{\text{approx.}}{\sim} p(x_{1:T} \mid \theta, y_{1:T})$.



Problems

Problems with this approach,

- Based on a PF $\Rightarrow$ approximate sample.
- Does not leave $p(\theta, x_{1:T} \mid y_{1:T})$ 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 path is retained throughout the sampler.

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.

Particle Markov Chain Monte Carlo (PMCMC)

The idea underlying **PMCMC** 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.

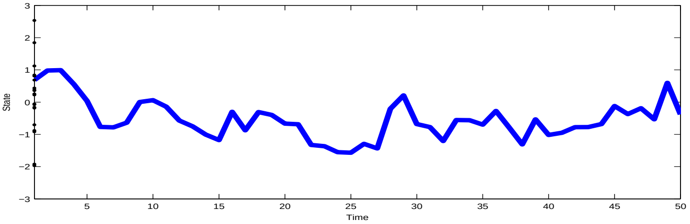
This Markov kernel is then used in a standard MCMC algorithm (e.g. Gibbs, resulting in the **Particle Gibbs (PG)**).

Three SMC samplers leaving $p(x_{1:T} \mid \theta, y_{1:T})$ invariant:

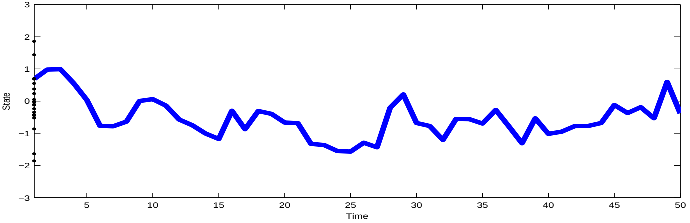
1. Conditional particle filter (CPF)
   Christophe Andrieu, Arnaud Doucet and Roman Holenstein, **Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72:269-342, 2010.
2. CPF with backward simulation (CPF-BS)
   Fredrik Lindsten and Thomas B. Schön. **On the use of backward simulation in the particle Gibbs sampler**. *Proc. of the 37th Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.
3. **CPF with ancestor sampling (CPF-AS)**
   Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Ancestor sampling for particle Gibbs**, *Advances in Neural Information Processing Systems (NIPS) 25*, Lake Tahoe, NV, US, December, 2012.

CPF vs. CPF-AS

CPF



CPF-AS



Conditional PF with ancestor sampling

Theorem

For any $N \geq 2$*, the procedure;*

(i) *Run CPF-AS*$(x^{\star}_{1:T})$*;*

(ii) *Sample* $P(x'_{1:T} = x^i_{1:T}) \propto w^i_T$*;*

defines a Markov kernel on X^T *which leaves* $p(x_{1:T} \mid \theta, y_{1:T})$ *invariant.*

Three additional reasons for using CPF-AS:

1. Significantly improves the mixing compared to CPF.
2. The computational complexity is linear in N.
3. Opens up for non-Markovian models.

Particle Gibbs with ancestor sampling

Bayesian identification: Gibbs + CPF-AS = PG-AS

Algorithm PG-AS: Particle Gibbs with ancestor sampling

```
1. Initialize: Set $\{\theta[0], x_{1:T}[0]\}$ arbitrarily.
2. For $m \geq 1$, iterate:
    (a) Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$.
    (b) Run CPF-AS$(x_{1:T}[m-1])$, targeting $p(x_{1:T} \mid \theta[m], y_{1:T})$.
    (c) Sample with $P(x_{1:T}[m] = x_{1:T}{}^{i}) \propto w_T^i$.
```

For any number of particles $N \geq 2$, the Markov chain $\{\theta[m], x_{1:T}[m]\}_{m\geq 1}$ has stationary distribution $p(\theta, x_{1:T} \mid y_{1:T})$.

Toy example – stochastic volatility (I/II)

Consider the stochastic volatility model,

$$\begin{aligned} x_{t+1} &= 0.9x_t + w_t, & w_t &\sim \mathcal{N}(0, \theta), \\ y_t &= e_t \exp\left(\frac{1}{2}x_t\right), & e_t &\sim \mathcal{N}(0, 1). \end{aligned}$$

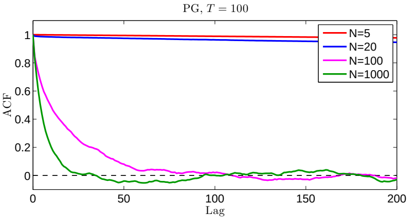
Let us study the ACF for the estimation error, $\widehat{\theta} - \mathrm{E}\left[\theta \mid y_{1:T}\right]$

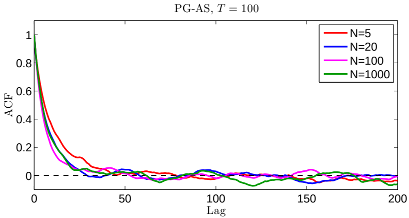
Toy example – stochastic volatility (I/II)

Consider the stochastic volatility model,

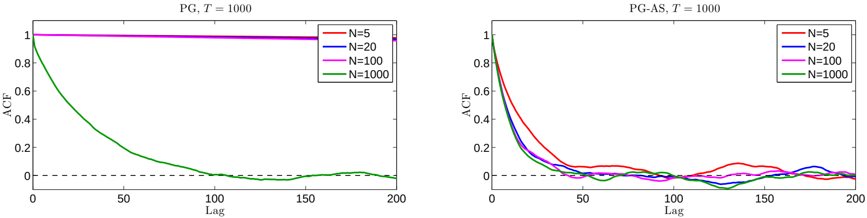
$$
\begin{aligned}
x_{t+1} &= 0.9 x_t + w_t, & w_t &\sim \mathcal{N}(0, \theta),\\
y_t &= e_t \exp\left(\frac{1}{2} x_t\right), & e_t &\sim \mathcal{N}(0, 1).
\end{aligned}
$$

Let us study the ACF for the estimation error, $\widehat{\theta} - \mathrm{E}\left[\theta \mid y_{1:T}\right]$





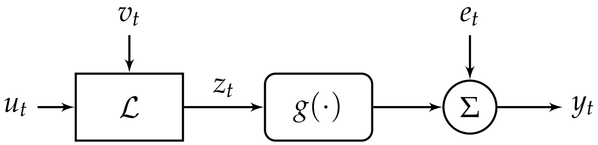
Toy example – stochastic volatility (II/II)



Some observations:

- We want the ACF to decay to zero as rapidly as possible (indicates good mixing in the PG sampler).
- Note the superior mixing of PG-AS compared to PG-CPF (already for just $N = 5$ particles!).

Example – semiparametric Wiener model (I/III)



Parametric LGSS and a nonparametric static nonlinearity:

$$
\begin{aligned}
x_{t+1} &= \underbrace{\begin{pmatrix} A & B \end{pmatrix}}_{\Gamma} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + v_t, & v_t &\sim \mathcal{N}(0, Q), \\
z_t &= Cx_t. \\
y_t &= g(z_t) + e_t, & e_t &\sim \mathcal{N}(0, R).
\end{aligned}
$$

Example – semiparametric Wiener model (II/III)

Everything is learned from the data, by introducing the possibility to switch specific model components on and off.

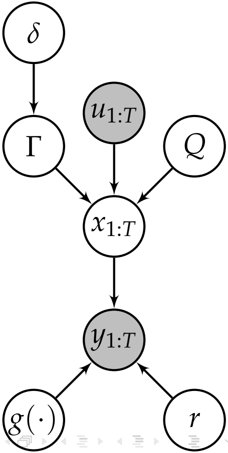
“Parameters”: $\theta = \{A, B, Q, \delta, g(\cdot), r\}$.

Bayesian model specified by priors

- Sparseness prior (ARD) on $\Gamma = [A \; B]$,
- Inverse-Wishart prior on Q and r
- Gaussian process prior on $g(\cdot)$,

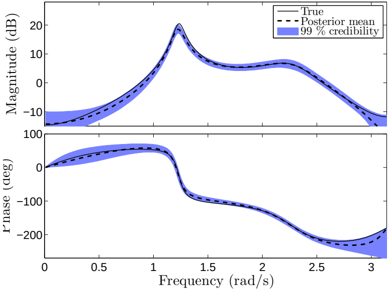
$$g(\cdot) \sim \mathcal{GP}(z, k(z, z')).$$

Inference using PG-AS with $N = 15$ particles. $T = 1\,000$ measurements. We ran $15\,000$ MCMC iterations and discarded $5\,000$ as burn-in.

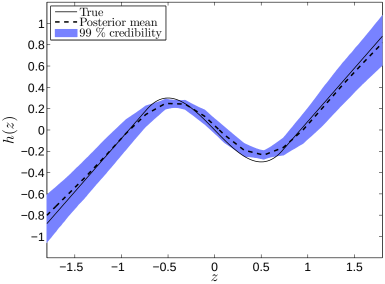


Example – semiparametric Wiener model (III/III)

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).

Fredrik Lindsten, Thomas B. Schön and Michael I. Jordan. **Bayesian semiparametric Wiener system identification**. *Automatica*, 49(7): 2053-2063, July 2013.

Conclusions

- Probabilistic models of dynamical systems.
- Sequential Monte Carlo introduced via the particle filter.
- EM-PS for ML learning in nonlinear SSMs.
- PG-AS for Bayesian learning in nonlinear SSMs.
- The conditional particle filter defines a kernel on X^T leaving $p_\theta(x_{1:T} \mid y_{1:T})$ invariant.

There is a lot of interesting research that remains to be done!!

- We are working on a book project,

  Thomas B. Schön and Fredrik Lindsten, **Computational learning in dynamical systems**, 2013.

  Send me an e-mail if you are interested in a draft.
- PhD course: `user.it.uu.se/~thosc112/CIDS.html`

Some references

Forthcoming book (includes all material used in this seminar)

Thomas B. Schön and Fredrik Lindsten, **Computational learning in dynamical systems**, 2013.

Novel introduction of PMCMC (very nice paper!)

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.

Self-contained introduction to BS and AS (not limited to SSMs)

Fredrik Lindsten and Thomas B. Schön, **Backward simulation methods for Monte Carlo statistical inference**, *Foundations and Trends in Machine Learning*, 6(1):1-143, 2013.

PG-AS (and the Wiener identification example)

Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Ancestor sampling for particle Gibbs**, *Advances in Neural Information Processing Systems (NIPS) 25*, Lake Tahoe, NV, US, December, 2012.

Fredrik Lindsten, Thomas B. Schön and Michael I. Jordan. **Bayesian semiparametric Wiener system identification**. *Automatica*, 2013, 49(7): 2053-2063.

ML identification of nonlinear SSMs (and Wiener example)

Thomas B. Schön, Adrian Wills and Brett Ninness. **System Identification of Nonlinear State-Space Models**. *Automatica*, 47(1):39-49, January 2011.

Adrian Wills, Thomas B. Schön, Lennart Ljung and Brett Ninness. **Identification of Hammerstein-Wiener Models**. *Automatica*, 49(1): 70-81, January 2013.

Bayesian inference using Gaussian processes

Roger Frigola, Fredrik Lindsten, Thomas B. Schön and Carl E. Rasmussen, **Bayesian inference and learning in Gaussian process state-space models with particle MCMC**. In *Advances in Neural Information Processing Systems (NIPS) 26*, Lake Tahoe, NV, USA, December 2013. (accepted for publication)