Solving nonlinear inference problems using sequential Monte Carlo

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

UPPSALA
UNIVERSITET

Thomas Schön
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:

Thomas B. Schön, Fredrik Lindsten, Johan Dahlin, Johan Wågberg, Christian A. Naesseth, Andreas Svensson and Liang Dai. **Sequential Monte Carlo methods for system identification**. In *Proceedings of the 17th IFAC Symposium on System Identification (SYSID)*, Beijing, China, October 2015.

UPPSALA UNIVERSITET

Introduction

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

$$
\begin{aligned}
x_{t+1} \,|\, x_t &\sim f_\theta(x_{t+1} \,|\, x_t, u_t), \\
y_t \,|\, x_t &\sim g_\theta(y_t \,|\, x_t, u_t), \\
x_1 &\sim \mu_\theta(x_1), \\
(\theta &\sim \pi(\theta)).
\end{aligned}
\qquad
\begin{aligned}
x_{t+1} &= a_\theta(x_t, u_t) + v_{\theta,t}, \\
y_t &= c_\theta(x_t, u_t) + e_{\theta,t}, \\
x_1 &\sim \mu_\theta(x_1), \\
(\theta &\sim \pi(\theta)).
\end{aligned}
$$

Identifying the nonlinear SSM: Find θ 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.

UPPSALA UNIVERSITET

Two commonly used problem formulations

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

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

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 $x_{1:T}$,

$$p_\theta(y_{1:T}) = \int p_\theta(x_{1:T}, y_{1:T}) \mathrm{d}x_{1:T}.$$

We are averaging $p_\theta(x_{1:T}, y_{1:T})$ over all possible state sequences.

Equivalently we have

$$p_\theta(y_{1:T}) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{1:t-1}) = \prod_{t=1}^{T} \int g_\theta(y_t \mid x_t) \underbrace{p_\theta(x_t \mid y_{1:t-1})}_{\textbf{key challenge}} \mathrm{d}x_t.$$

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 provide approximate solutions to **integration** problems where there is a **sequential structure** present.

The particle filter and the particle smoother maintain empirical approximations

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

Converge to the true distributions as $N \longrightarrow \infty$.

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, $\widetilde{x}_{1:T} \sim p(x_{1:T} \mid y_{1:T})$.

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

UPPSALA UNIVERSITET

Identification strategies – overview

Marginalization Deal with $x_{1:T}$ by marginalizing (integrating) them out and view θ 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 θ.

- Frequentistic formulation: Expectation Maximization (EM).
- Bayesian formulation: the Gibbs sampler.

Only data augmentation strategies in this talk.

UPPSALA UNIVERSITET

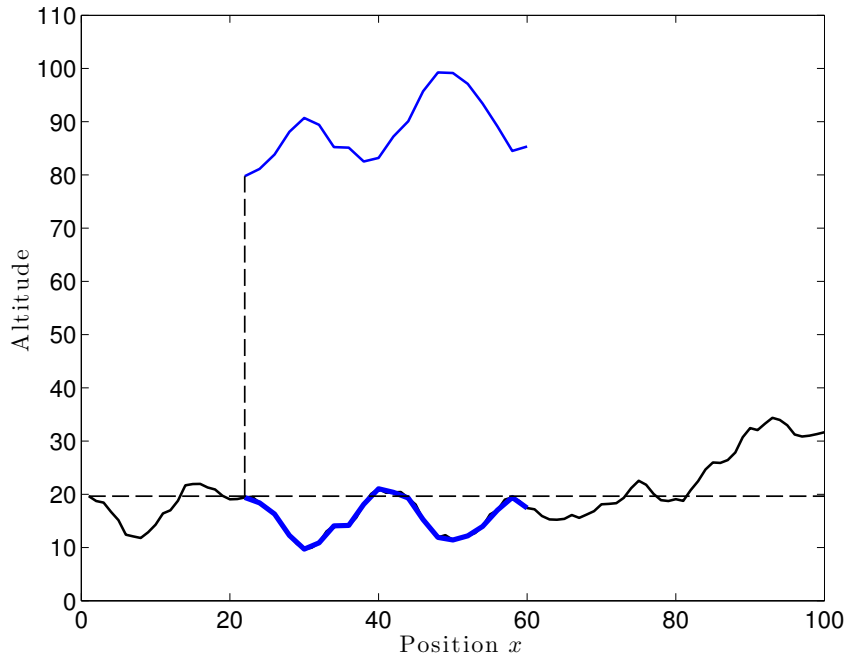
Outline

1. Problem formulation
2. Identification strategies for nonlinear SSMs
3. **Sequential Monte Carlo (SMC)**
4. Data augmentation
   a) Expectation maximization (EM)
   b) Gibbs sampling "identification without optimization"
5. Snapshots of current research
   a) The Gaussian process SSM and regularization
   b) The nonlinear SSM is just a special case...
   c) SMC in high dimensions
   d) Coupling of particle filters

Particle filter – introductory example (I/III)

Consider a toy 1D localization problem.

Data



Model

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 in many applications, see e.g.

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.

Particle filter – introductory example (II/III)

Task: Find the state x_t based on a set of measurements $y_{1:t} \triangleq \{y_1, \ldots, y_t\}$ by computing the filter PDF $p(x_t \mid y_{1:t})$.

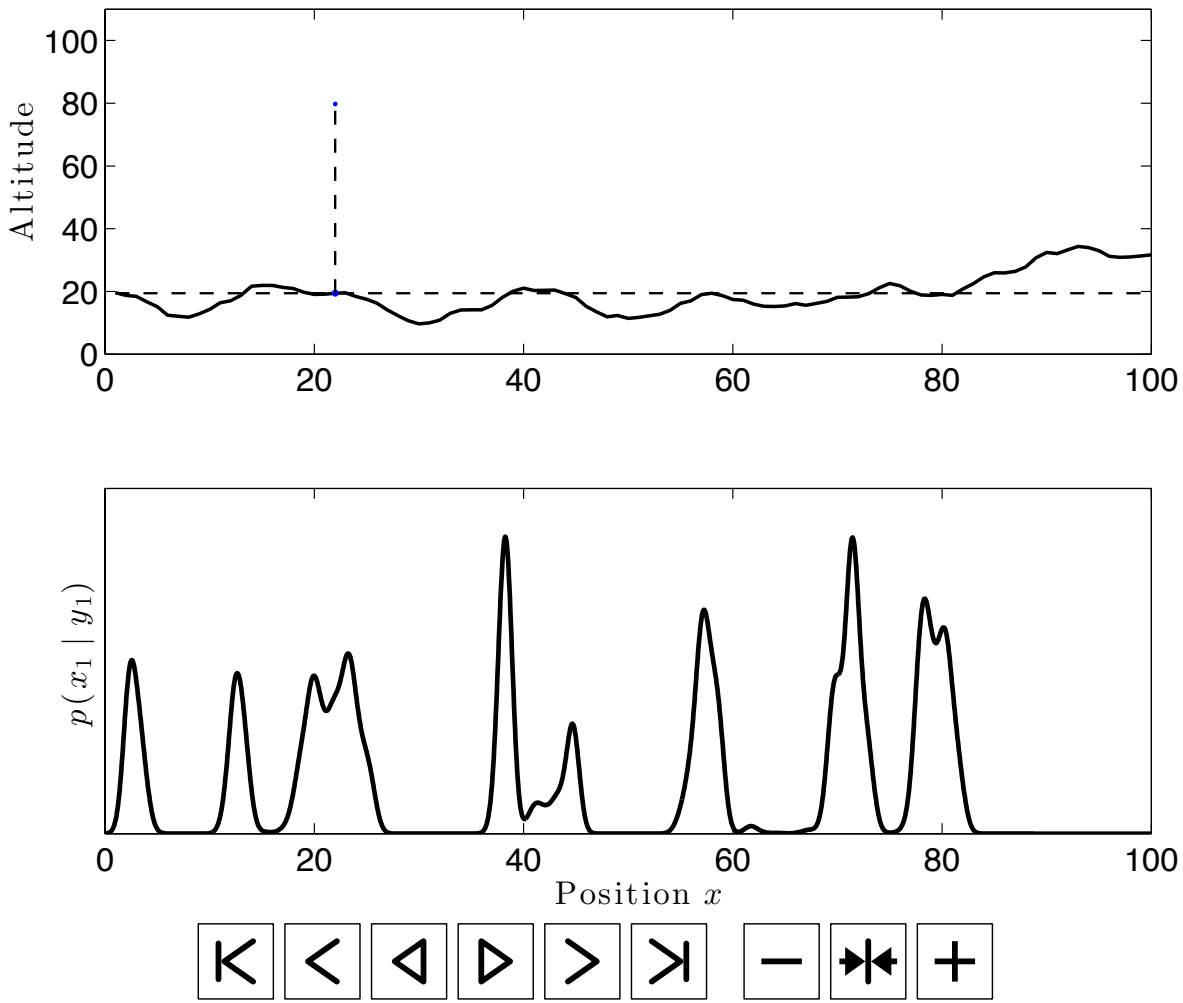
The particle filter (PF) maintains an approximation according to

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

that converge to the true filtering distribution as $N \to \infty$.

"Think of each particle as one simulation of the system state. Keep the ones that best explains the measurements."

Particle filter – introductory example (III/III)



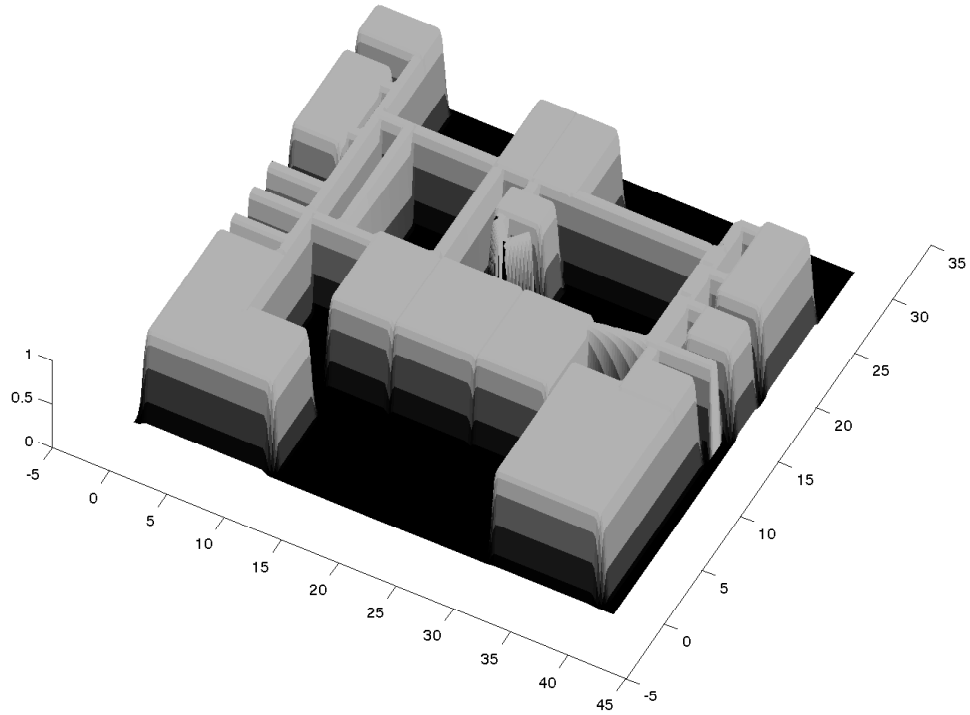
Highlights two **key capabilities** of the PF:

1. Automatically handles an unknown and dynamically changing number of hypotheses.
2. Work with nonlinear/non-Gaussian models.

UPPSALA UNIVERSITET

Example – 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.



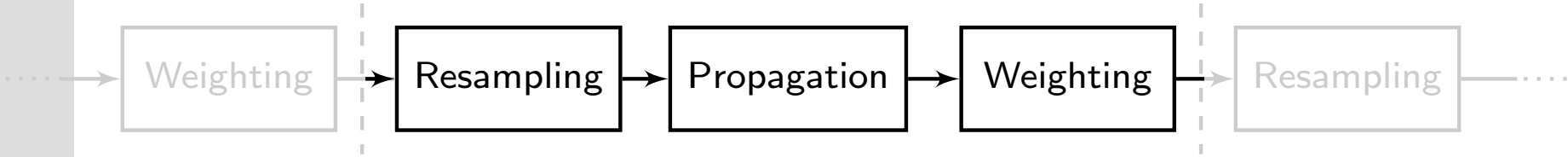
UPPSALA
UNIVERSITET

Example – indoor localization

Show movie

Johan Kihlberg, Simon Tegelid, Manon Kok and Thomas B. Schön. **Map aided indoor positioning using particle filters.** *Reglermöte (Swedish Control Conference)*, Linköping, Sweden, June 2014.

Sequential Monte Carlo – particle filter



SMC = resampling + sequential importance sampling

1. **Resampling:** $\mathbb{P}\left(a_t^i = j\right) = \bar{w}_{t-1}^j / \sum_l \bar{w}_{t-1}^l$.
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.

Use of random numbers in the particle filter

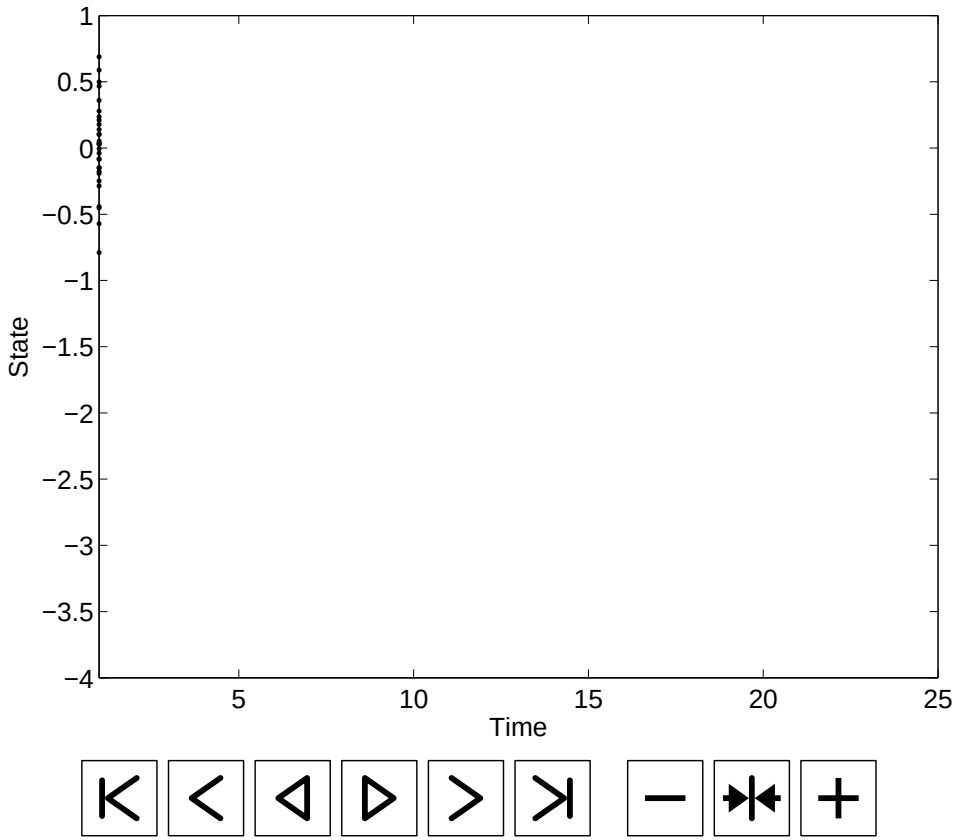
Random numbers are used

1. to initialize the particles,
2. to resample and
3. to propagate them.

The weighting step does not require any new random numbers, it is just a function of already existing random numbers.

We can reason about and make use of the probability distribution from which the particle filter generates one realisation each time it is executed.

The particle system degenerates (illustration)



Clearly motivates the need for **particle smoothers**.

Self-contained introduction to particle smoothing using BS and AS

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.

UPPSALA UNIVERSITET

Outline

1. Problem formulation
2. Identification strategies for nonlinear SSMs
3. Sequential Monte Carlo (SMC)
4. **Data augmentation**
   a) Expectation maximization (EM)
   b) Gibbs sampling "identification without optimization"
5. Snapshots of current research
   a) The Gaussian process SSM and regularization
   b) The nonlinear SSM is just a special case...
   c) SMC in high dimensions
   d) Coupling of particle filters

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 θ.

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 θ.

UPPSALA UNIVERSITET

Data augmentation – EM

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

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

The expectation maximization algorithm is an iterative approach to compute ML estimates of unknown parameters (θ) in probabilistic models involving latent variables (e.g. 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 $p_\theta(y_{1:T})$,

$$p_\theta(x_{1:T}, y_{1:T}) = p_\theta(x_{1:T} \mid y_{1:T}) p_\theta(y_{1:T}).$$

Data augmentation – EM

EM works by iteratively computing

$$\mathcal{Q}(\theta, \theta_k) = \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta_k}(x_{1:T} \mid y_{1:T}) \mathrm{d}x_{1:T}$$

and then maximizing $\mathcal{Q}(\theta, \theta_k)$ w.r.t. θ.

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 $p_{\theta_k}(x_{1:T} \mid y_{1:T})$.

UPPSALA UNIVERSITET

Using EM and particle smoothing together

Algorithm 1 EM for identifying nonlinear dynamical systems

1. **Initialise:** Set $k = 1$ and choose an initial θ_1.
2. **While** not converged **do:**

(a) **Expectation (E) step:** Compute

$$\mathcal{Q}(\theta, \theta_k) = \int \log p_\theta(x_{1:T}, y_{1:T}) \underbrace{p_{\theta_k}(x_{1:T} \mid y_{1:T})}_{} \mathrm{d}x_{1:T}$$

using **sequential Monte Carlo** (particle smoother).

(b) **Maximization (M) step:** Compute $\theta_{k+1} = \arg\max_{\theta \in \Theta} \mathcal{Q}(\theta, \theta_k)$

(c) $k \leftarrow k + 1$

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

UPPSALA UNIVERSITET

Data augmentation – Gibbs sampling

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 $p(\theta, x_{1:T} \,|\, y_{1:T})$.

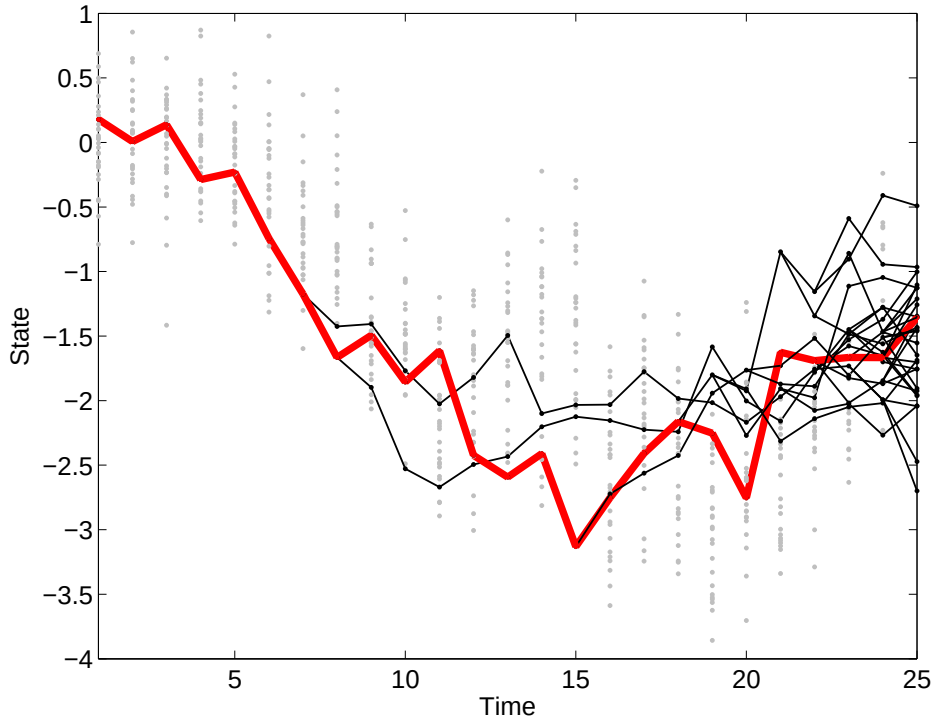
A (blocked) example:

- 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 $\mathbb{P}\left(x^{\star}_{1:T} = x^{i}_{1:T}\right) \propto w^{i}_{T}$ we get, $x^{\star}_{1:T} \overset{\text{approx.}}{\sim} p(x_{1:T} \,|\, \theta, y_{1:T})$.



UPPSALA UNIVERSITET

Problems and a solution

Problems with this approach,

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

To get around these problems,

Use a conditional particle filter. One pre-specified reference trajectory 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.

UPPSALA UNIVERSITET

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 sampler 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 **without** introducing any systematic errors!

UPPSALA UNIVERSITET

Three SMC samplers

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)

   N. Whiteley, **Discussion on Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72(3), 306–307, 2010.

   Fredrik Lindsten and Thomas B. Schön. **On the use of backward simulation in the particle Gibbs sampler**. *Proceedings of the 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.

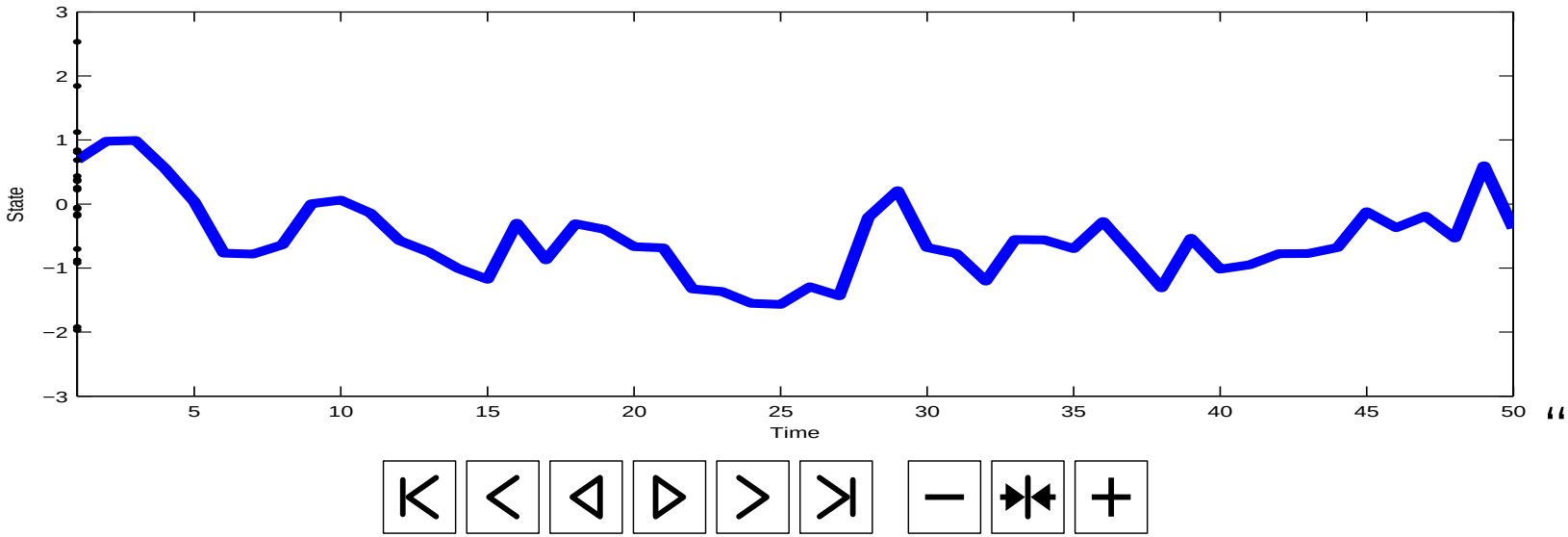
3. **CPF with ancestor sampling (CPF-AS)**

   Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Particle Gibbs with ancestor sampling**. *Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

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^{th} particle deterministically: $x_t^N = x'_t$.



CPF causes us to degenerate to the something that is very **similar** to the reference trajectory, resulting in **slow mixing**.

UPPSALA UNIVERSITET

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.

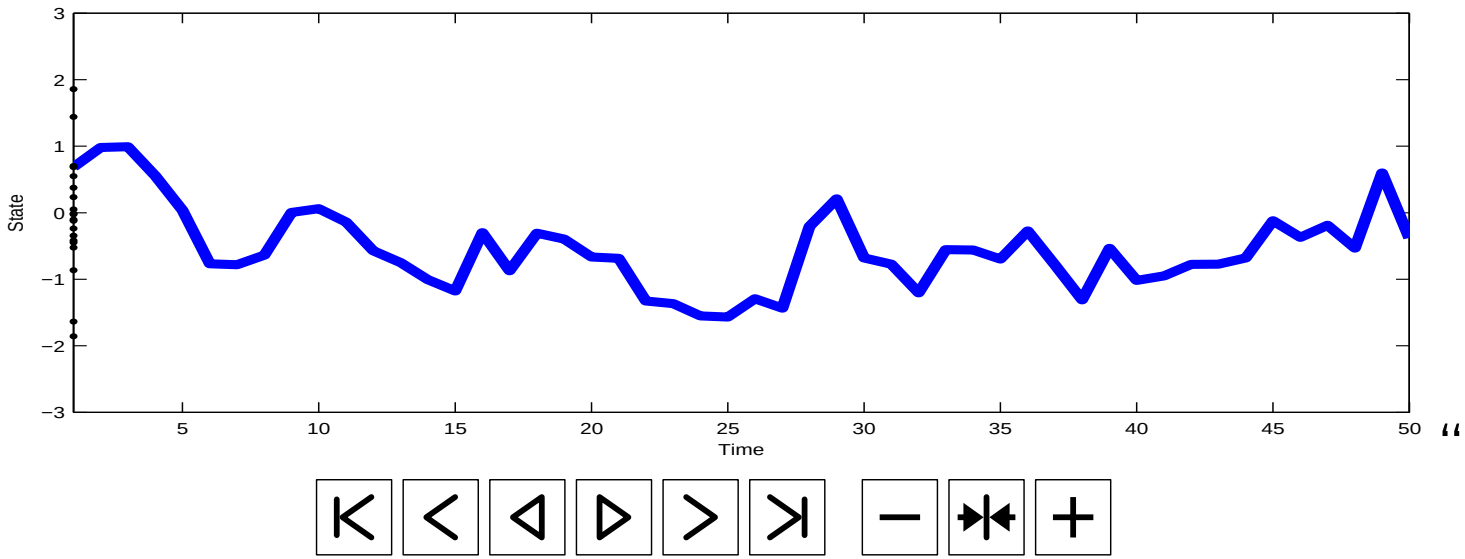
Implication: Ancestor sampling opens up for inference in a wider class of models, e.g. non-Markovian SSMs, PGMs and BNP models.

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*.

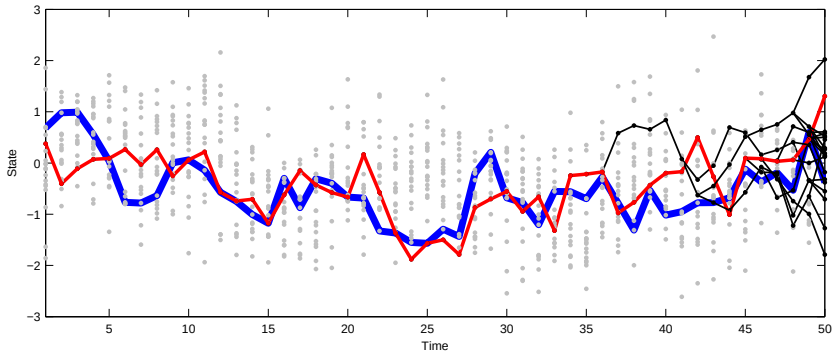
- At each time t, sample $N-1$ particles in the standard way.
- Set the N^{th} particle deterministically: $x_t^N = x'_t$.
- Generate an artificial history for x_t^N by ancestor sampling.



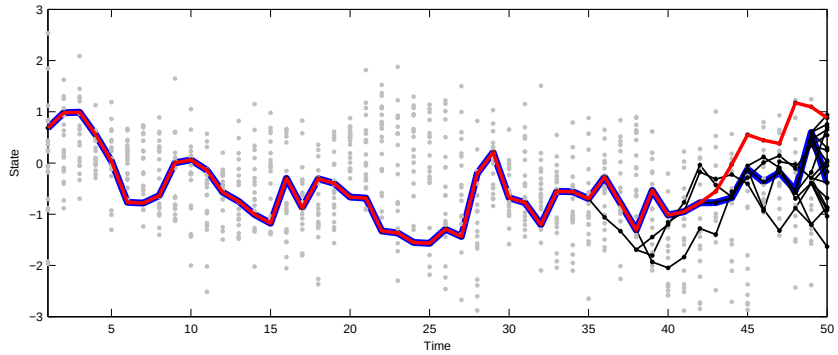
CPF-AS causes us to degenerate to something that is very **different** from the reference trajectory, resulting in **better mixing**.

PGAS vs PG

PGAS



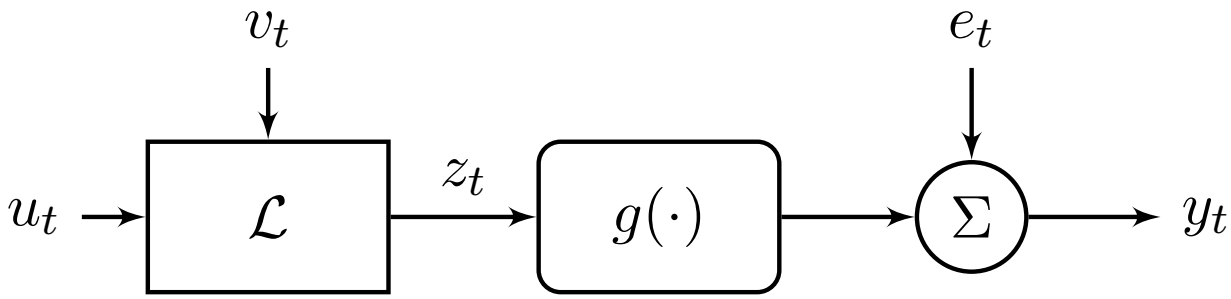
PG



Red: Old reference trajectory.

Blue: New reference trajectory.

Example – semiparametric Wiener model



Parametric LGSS and a nonparametric static nonlinearity:

$$x_{t+1} = \underbrace{\begin{pmatrix} A & B \end{pmatrix}}_{\Gamma} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + v_t, \qquad v_t \sim \mathcal{N}(0, Q),$$

$$z_t = C x_t.$$

$$y_t = g(z_t) + e_t, \qquad e_t \sim \mathcal{N}(0, R).$$

Example – semiparametric Wiener model

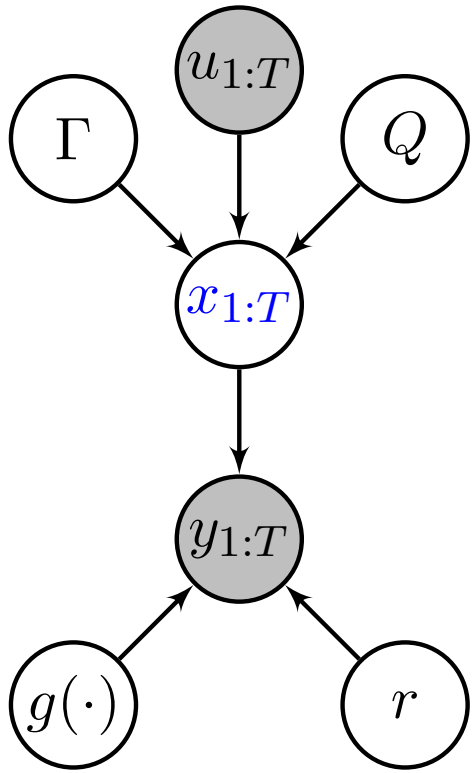
"Parameters": $\theta = \{A, B, Q, g(\,\cdot\,), r\}$.

Bayesian model specified by priors

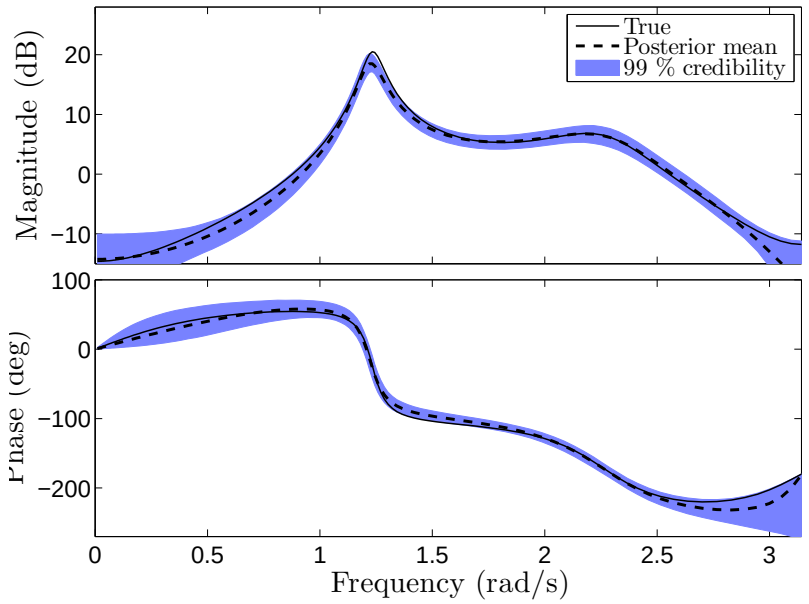
- Conjugate priors for $\Gamma = [A\ B]$, Q and r,
  - $p(\Gamma, Q) =$ Matrix-normal inverse-Wishart
  - $p(r) =$ inverse-Wishart
- Gaussian process prior on $g(\,\cdot\,)$,

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

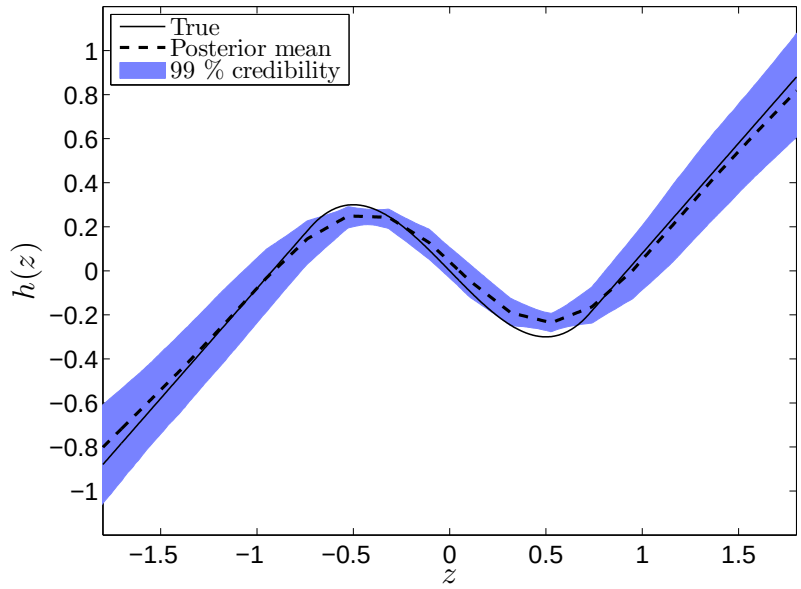
Inference using PGAS with $N = 15$ particles. $T = 1\,000$ measurements. We ran $15\,000$ MCMC iterations and discarded $5\,000$ as burn-in.



Example – semiparametric Wiener model



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

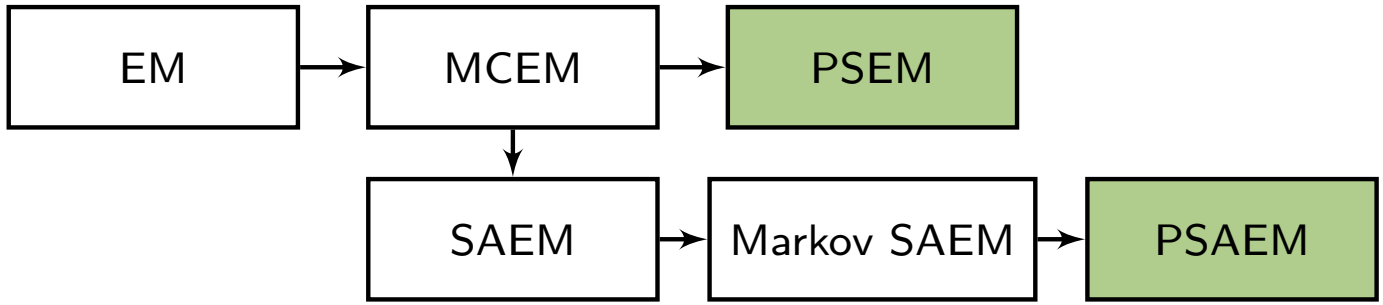


Static nonlinearity (non-monotonic), estimated mean (dashed black), true (black) and the 99% credibility intervals (blue).

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

Look at the algorithms!!

Combined ML and Bayesian approach



In stochastic approximation EM (SAEM) $\mathcal{Q}(\theta, \theta_k)$ is replaced by

$$\widehat{\mathcal{Q}}_k(\theta) = (1-\gamma_k)\widehat{\mathcal{Q}}_{k-1}(\theta) + \gamma_k \log p_\theta(x_{1:T}[k], y_{1:T}),$$

where $x_{1:T}[k]$ denotes a sample from $p_{\theta_k}(x_{1:T} \mid y_{1:T})$.

F. Lindsten, **An efficient stochastic approximation EM algorithm using conditional particle filters**. In *Proceedings of the Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

UPPSALA UNIVERSITET

Outline

1. Problem formulation
2. Identification strategies for nonlinear SSMs
3. Sequential Monte Carlo (SMC)
4. Data augmentation
   a) Expectation maximization (EM)
   b) Gibbs sampling "identification without optimization"
5. **Snapshots of current research**
   a) The Gaussian process SSM and regularization
   b) The nonlinear SSM is just a special case...
   c) SMC in high dimensions
   d) Coupling of particle filters

UPPSALA UNIVERSITET

Gaussian Process nonlinear state space model

Consider the Gaussian Process SSM (GP-SSM):

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

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

We can now find the posterior distribution

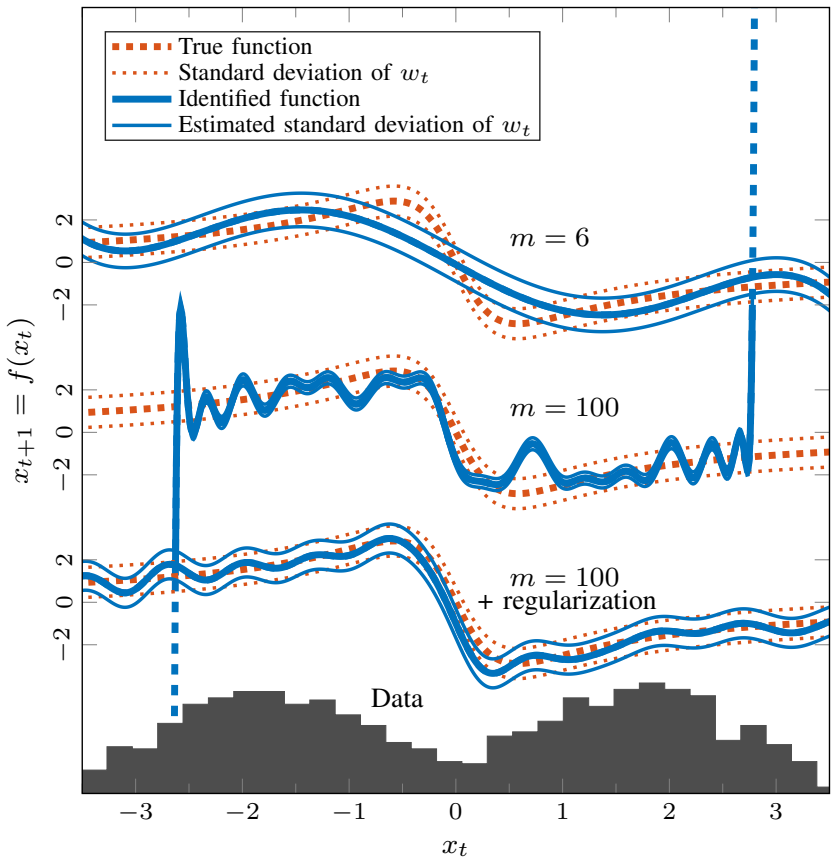
$$p(f, g, Q, R, \theta \,|\, y_{1:T}),$$

by making use of new MCMC algorithms.

Andreas Svensson, Arno Solin, Simo Särkkä and Thomas B. Schön, **Computationally efficient Bayesian learning of Gaussian process state space models**. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Cadiz, Spain, May, 2016.

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

Gaussian Process nonlinear state space model



This gives us a **flexible** nonparametric model where the GP prior on f takes on the role of a regularizer. This provides a data-driven way of **tuning** the model complexity.

Toy example:

$$x_{t+1} = -10\frac{x_t}{1+3x_t^2} + v_t,$$

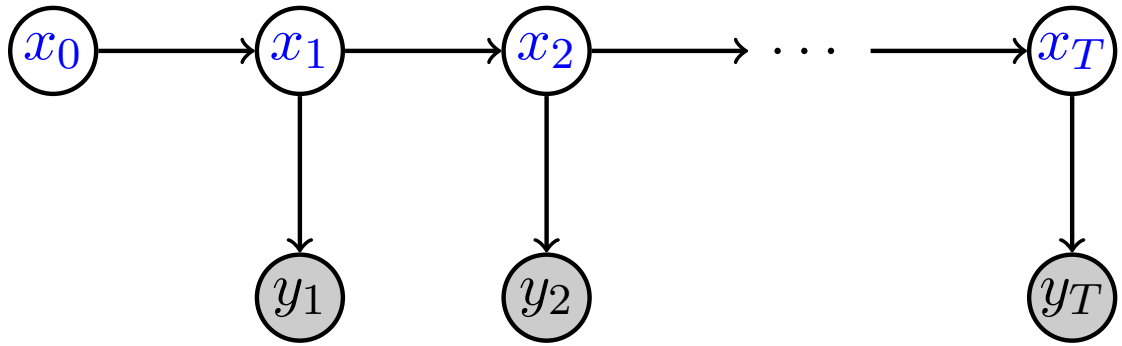
$$y_t = x_t + e_t.$$

Andreas Svensson and Thomas B. Schön. **A flexible state space model for learning nonlinear dynamical systems**, *Preprint on ArXiv:1603.05486*, 2016.

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,

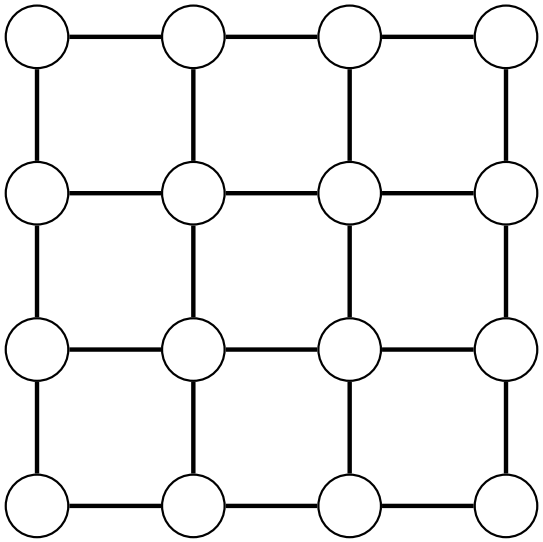
1. a set of **vertices** $\mathcal{V}$ (nodes) represents the random variables
2. a set of **edges** $\mathcal{E}$ containing elements $(i, j) \in \mathcal{E}$ connecting a pair of nodes $(i, j) \in \mathcal{V}$

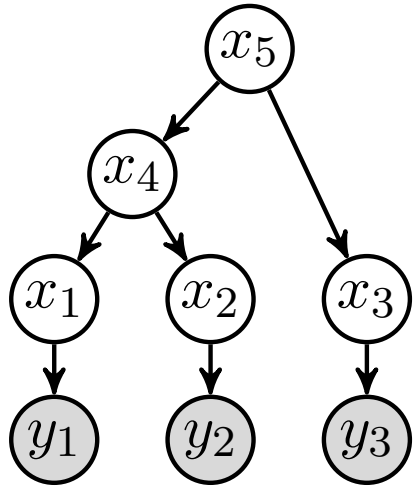


$$p(x_{0:T}, y_{1:T}) = p(x_0) \prod_{t=1}^{N} p(x_t \mid x_{t-1}) \prod_{t=1}^{N} p(y_t \mid x_t).$$

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.





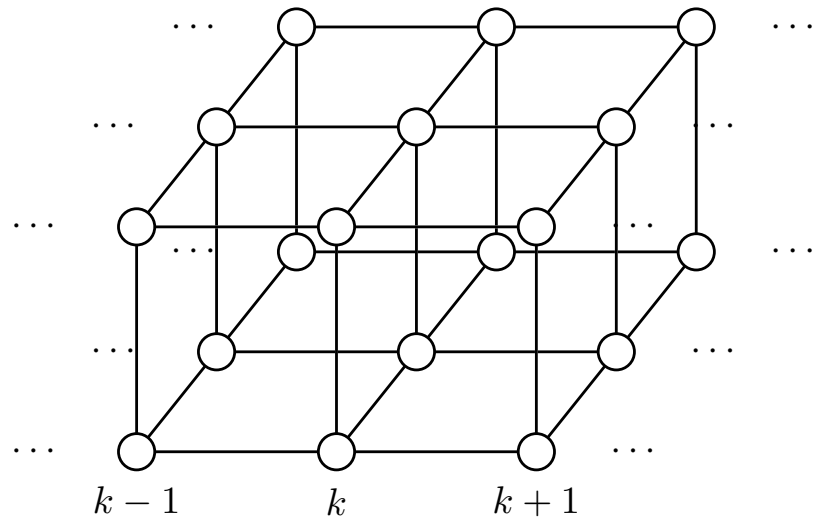
Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön, **Sequential Monte Carlo methods for graphical models**. *Advances in Neural Information Processing Systems (NIPS)* 27, Montreal, Canada, December, 2014.

Fredrik Lindsten, Adam M. Johansen, Christian A. Naesseth, Bonnie Kirkpatrick, Thomas B. Schön, John Aston and Alexandre Bouchard-Côté. **Divide-and-Conquer with Sequential Monte Carlo**. *arXiv:1406.4993*, June 2015.

SMC in high dimensions

The bootstrap PF suffers from weight collapse in high-dimensional settings.

This degeneracy can be reduced by using so-called **fully adapted** proposals.



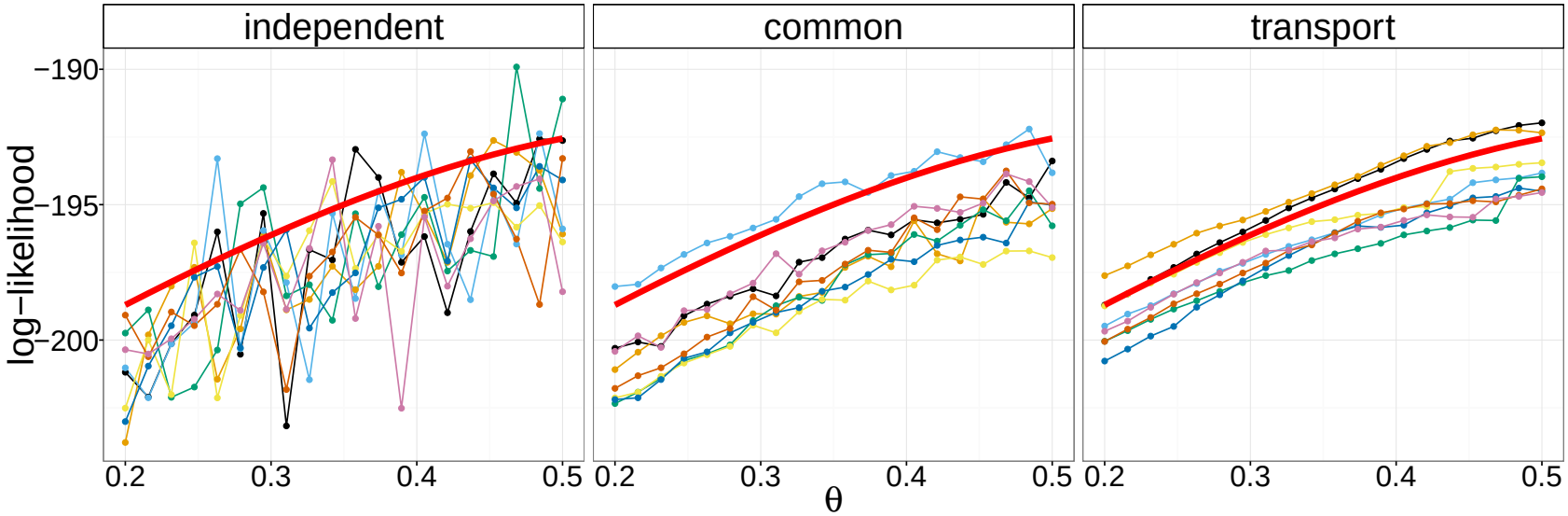
We can mimic the efficient fully adapted proposals for arbitrary latent spaces and structures in high-dimensional models.

Approximations the proposal distribution and use a **nested coupling** of multiple SMC samplers and backward simulators.

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön, **Nested sequential Monte Carlo**. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, July, 2015.

Coupling of particle filters

New **coupled resampling** schemes can be used to improve a variety of particle-based algorithms.



Log-likelihood estimates obtained by particle filters, in a hidden AR(1).

New particle smoother, easy to parallelize and with analysis.

Pierre Jacob, Fredrik Lindsten, Thomas B. Schön, **Coupling of particle filters**. On *arXiv* soon.

UPPSALA UNIVERSITET

New project – ASSEMBLE

Vision – hands on: "Develop the equivalent of cvx for inference and learning problems". Create a **market place** for inference algorithms, especially SMC and MCMC.

Develop a formally defined **probabilistic modeling language** and a model compiler.

> **Probabilistic programming** makes use of computer programs to represent probabilistic models.

Creates a clear **separation** between the model and the inference methods, encouraging model based thinking. Potential to automate inference!

Time frame: 5 years, starting 1 July 2016. Industry collaborators.

Swedish Foundation for Strategic Research.

UPPSALA UNIVERSITET

Conclusion

	Marginalization	Data augmentation
ML	Direct optimization	Expectation Maximization
Bayesian	Metropolis Hastings	Gibbs sampling

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 $p(x_{1:T} \mid y_{1:T})$.
3. Simulate from the smoothing pdf, $\widetilde{x}_{1:T} \sim p(x_{1:T} \mid y_{1:T})$.

Fast moving research area offering lots of opportunities!

UPPSALA UNIVERSITET

SMC convergence in one slide...

Let $\varphi : \mathsf{X} \mapsto \mathbb{R}$ be some test function of interest. The expectation

$$\mathrm{E}_\theta \left[\varphi(x_t) \,|\, y_{1:t}\right] = \int \varphi(x_t) p_\theta(x_t \,|\, y_{1:t}) \mathrm{d}x_t,$$

can be estimated by the particle filter

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

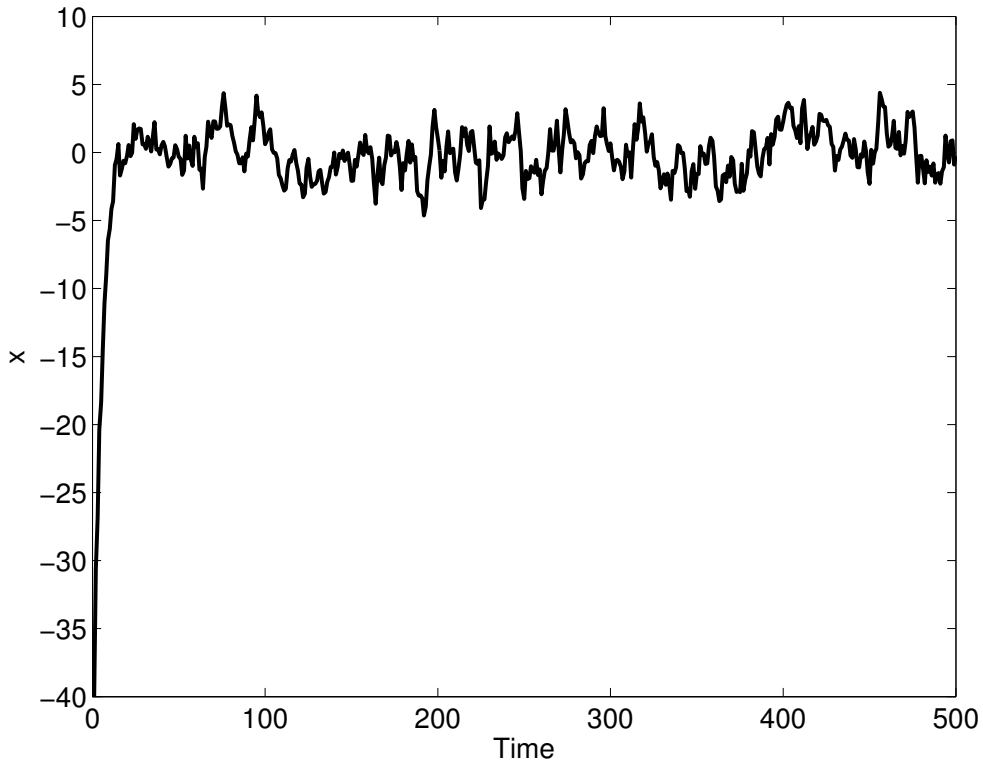
The **CLT** governing the convergence of this estimator states

$$\sqrt{N}\left(\widehat{\varphi}_t^N - \mathrm{E}_\theta \left[\varphi(x_t) \,|\, y_{1:t}\right]\right) \xrightarrow{\mathrm{d}} \mathcal{N}(0, \sigma_t^2(\varphi)).$$

The **likelihood estimate** $\widehat{p}_\theta(y_{1:t}) = \prod_{s=1}^t \left\{ \frac{1}{N} \sum_{i=1}^N \bar{w}_s^i \right\}$ from the PF is **unbiased**, $\mathrm{E}_{\psi_\theta}\left[\widehat{p}_\theta(y_{1:t})\right] = p_\theta(y_{1:t})$ for any value of N and there are **CLTs available** as well.

Micro: MCMC – AR(1) example (I/II)

One realisation from $x[k+1] = 0.8x[k] + v[k]$ where $v[k] \sim \mathcal{N}(0,1)$. Initialise in $x[0] = -40$.

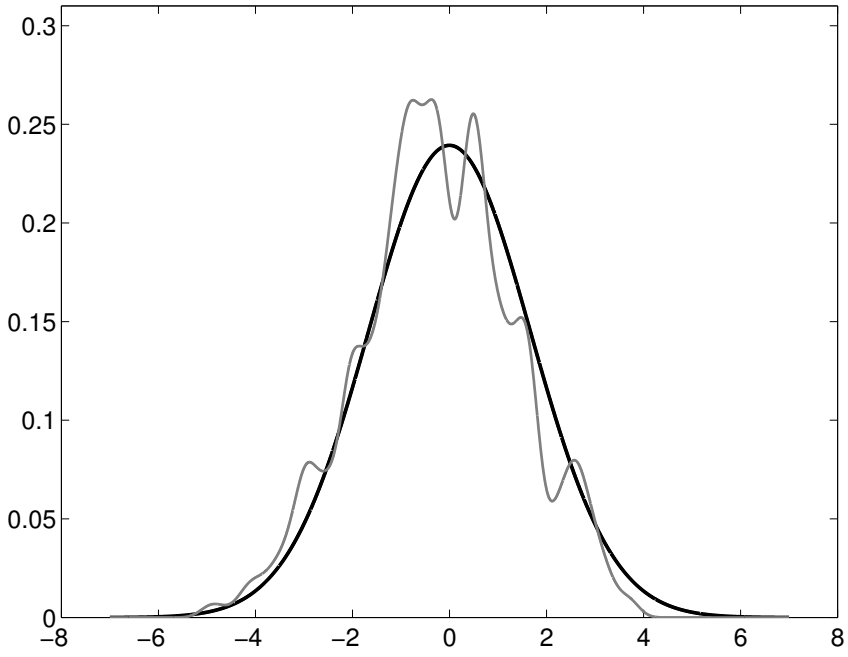


This will eventually generate samples from the following **stationary distribution**:

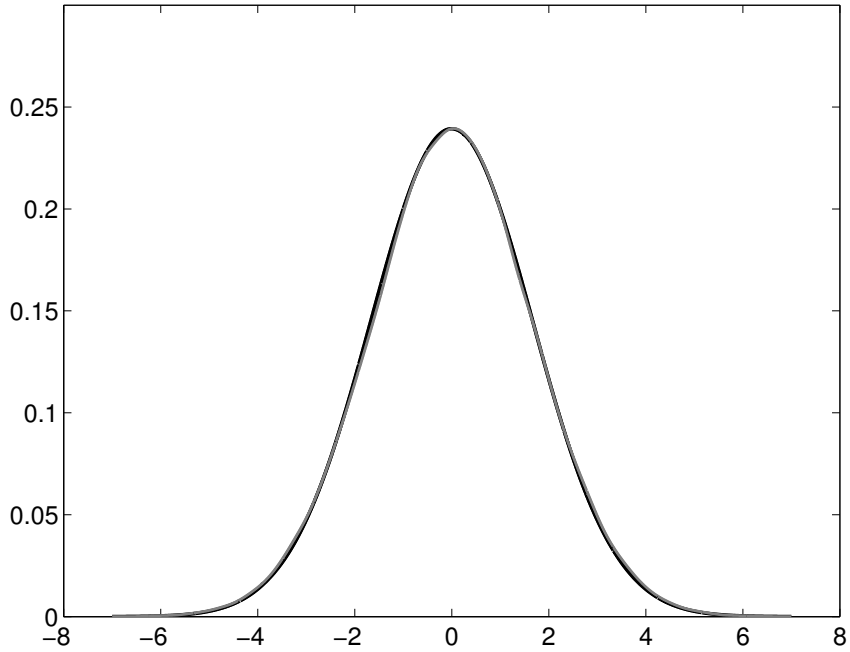
$$\pi^{\mathsf{s}}(x) = \mathcal{N}\left(x \,\middle|\, 0, \frac{1}{1-0.8^2}\right)$$

as $t \to \infty$.

Micro: MCMC – AR(1) example (II/II)



$1\,000$ samples



$100\,000$ samples

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 $1\,000$ 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.