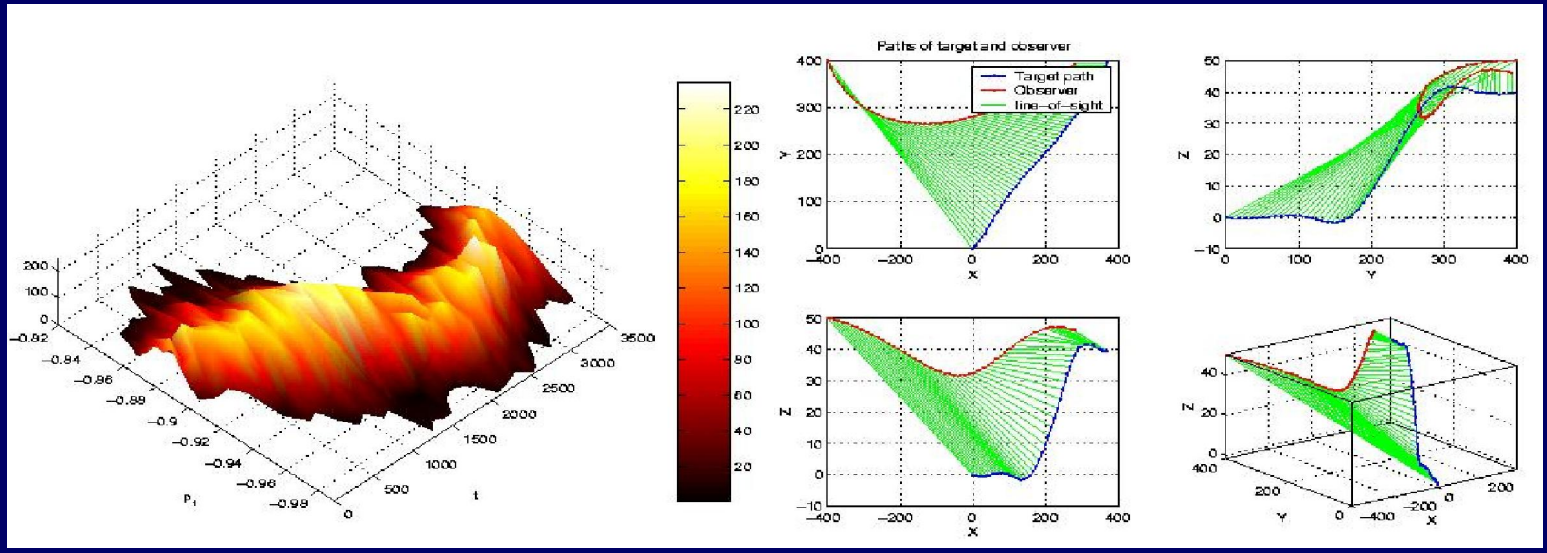
Sequential Monte Carlo Methods

Simon Godsill www-sigproc.eng.cam.ac.uk/~sjg

Cambridge, September, 2009



University of Cambridge

Overview

- We are faced with many problems involving large, sequentially evolving datasets:

  tracking, computer vision, speech and audio, robotics, financial time series, ...

- We wish to form models and algorithms for Bayesian sequential updating of probability distributions as data evolve

- Here we consider the Sequential Monte Carlo (SMC), or 'particle filtering' methodology (Gordon, Salmond and Smith IEE (93), Doucet, Godsill and Andrieu Stats. and Comp. (2000), Cappé, Godsill and Moulines Proc. IEEE (2007), and many more ...)

In many applications it is required to estimate a latent or 'hidden' process (the 'state' of the system) from noisy, convolved or non-linearly distorted observations. Since data also arrive sequentially in many applications it is therefore desirable (or essential) to estimate the hidden process on-line, in order to avoid memory storage of huge datasets and to make inferences and decisions in real time. Some typical applications from the engineering perspective include:

- Tracking for radar and sonar applications
- Real-time enhancement of speech and audio signals
- Sequence and channel estimation in digital communications channels
- Medical monitoring of patient eeg/ecg signals
- Image sequence tracking

In this tutorial we will consider sequential estimation in such applications.

Only when the system is linear and Gaussian can exact estimation be performed, using the classical Kalman filter. I will present a succinct derivation of the Kalman filter, based on Bayesian updating of probability models. In most applications, however, there are elements of non-Gaussianity and/or non-linearity which make analytical computations impossible. Here we must adopt numerical strategies. I will consider a powerful class of Monte Carlo filters, known generically as particle filters, which are well-adapted to general problems in this category.

Contents

Today:

- Bayes' Theorem
- Monte Carlo methods
- State space models, filtering and smoothing
- Kalman filter/ extended Kalman filter
- Monte Carlo filtering
- Sequential Monte Carlo (bootstrap)
- General Sequential Importance Sampling

Tomorrow:

Other exotica (auxiliary particle filter, smoothing, Rao-Blackwell, MCMC, etc...).

Bayes' Theorem and inference

Observations: $\mathbf{y}$

Quantity of interest: $\mathbf{x}$

Other parameters/unknowns: θ

Likelihood:

$$p(\,\mathbf{y}\,|\,\mathbf{x},\,\theta\,)$$

Joint posterior for all unknowns (Bayes' Theorem):

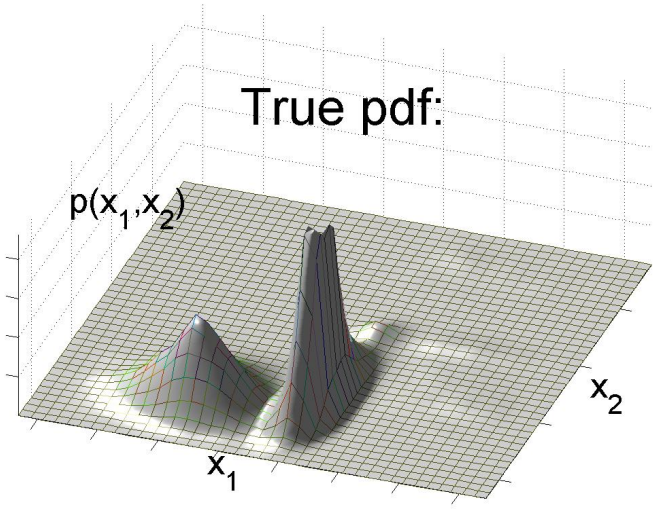
$$p(\,\mathbf{x},\,\theta\,|\,\mathbf{y}\,) \;=\; \frac{p(\,\mathbf{y}\,|\,\mathbf{x},\,\theta\,)\,p(\,\mathbf{x},\,\theta\,)}{p(\mathbf{y})}$$

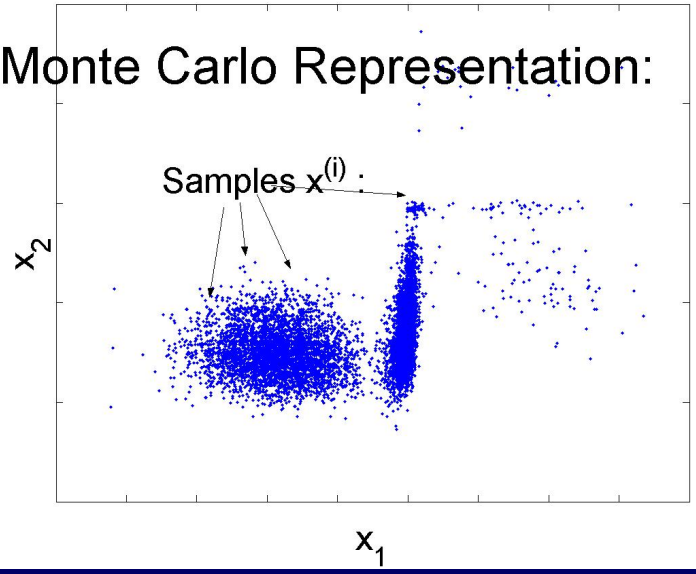
Marginal posterior for $\mathbf{x}$:

$$p(\,\mathbf{x}\,|\,\mathbf{y}\,) \;=\; \int_{\theta} p(\,\mathbf{x},\,\theta\,|\,\mathbf{y}\,)\,d\theta$$

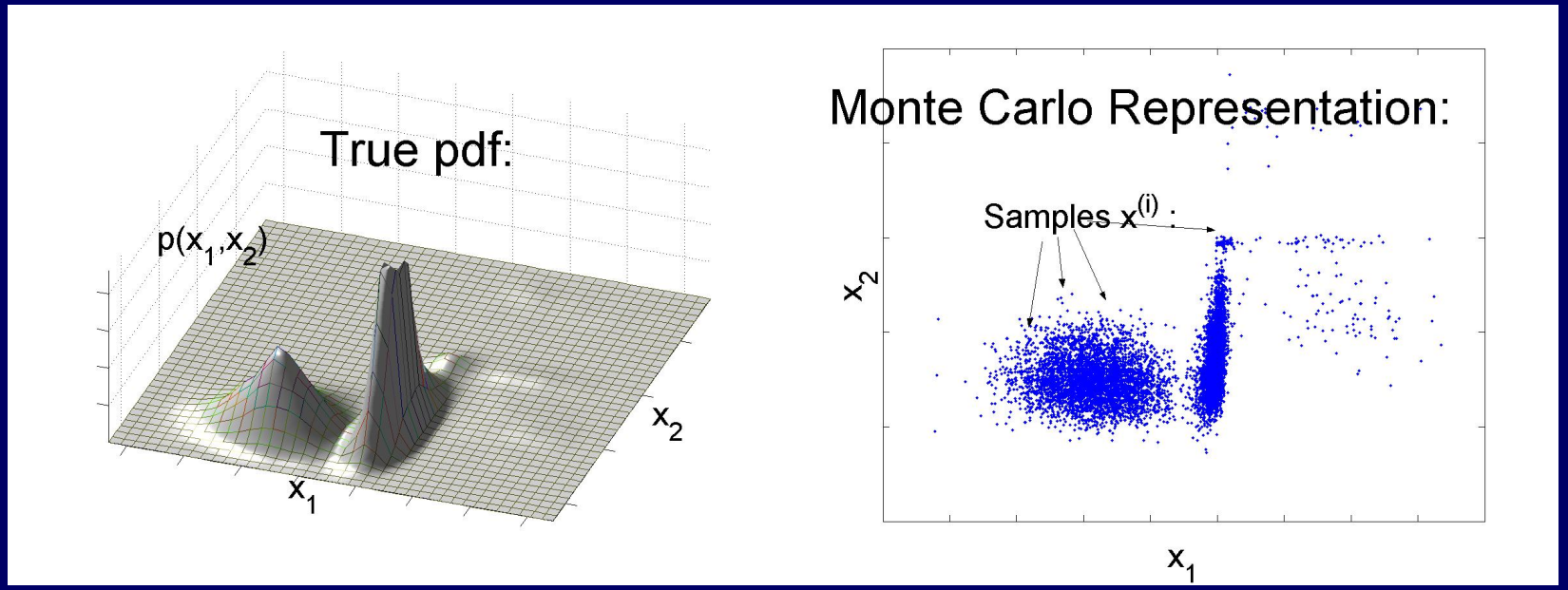
Marginal posterior is used for inference about the quantity of interest, $\mathbf{x}$.

Monte Carlo Methods





Monte Carlo Methods



In the Monte Carlo method, we are concerned here with estimating the properties of some highly complex probability distribution $p(x)$, e.g. expectations:

$$\mathbb{E}X = \int h(x)p(x)dx$$

where $h(.)$ is some useful function for estimation.

In cases where this cannot be achieved analytically the approximation problem can be tackled indirectly, as it is often possible to generate random samples from $p(x)$, i.e. by representing the distribution as a collection of random points:

$$x^{(i)},\ i = 1, ..., N, \text{ for large } N$$

In cases where this cannot be achieved analytically the approximation problem can be tackled indirectly, as it is often possible to generate random samples from $p(x)$, i.e. by representing the distribution as a collection of random points:

$$x^{(i)},\ i = 1, ..., N, \text{ for large } N$$

We can think of the Monte Carlo representation informally as:

$$p(x) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x)$$

In cases where this cannot be achieved analytically the approximation problem can be tackled indirectly, as it is often possible to generate random samples from $p(x)$, i.e. by representing the distribution as a collection of random points:

$$x^{(i)},\ i = 1, ..., N, \text{ for large } N$$

We can think of the Monte Carlo representation informally as:

$$p(x) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x)$$

Then the Monte Carlo expectation falls out easily as:

$$\mathbb{E}X = \int h(x)p(x)dx \approx \int h(x) \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x)dx = \frac{1}{N} \sum_{i=1}^{N} h(x^{(i)})$$

Alternatively, suppose we draw the random samples $x^{(i)}$ from a distribution $q(x)$ instead of $p(x)$. Now the expectation can be estimated using importance sampling:

Alternatively, suppose we draw the random samples $x^{(i)}$ from a distribution $q(x)$ instead of $p(x)$. Now the expectation can be estimated using importance sampling:

$$\mathbb{E}X = \int h(x)p(x)dx = \int h(x)\frac{q(x)p(x)}{q(x)}dx \approx \int h(x)\frac{p(x)}{q(x)}\frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x)dx$$

$$= \frac{1}{N}\sum_{i=1}^{N}\frac{p(x^{(i)})}{q(x^{(i)})}h(x^{(i)}) = \sum_{i=1}^{N}w^{(i)}h(x^{(i)})$$

Alternatively, suppose we draw the random samples $x^{(i)}$ from a distribution $q(x)$ instead of $p(x)$. Now the expectation can be estimated using importance sampling:

$$\mathbb{E}X = \int h(x)p(x)dx = \int h(x)\frac{q(x)p(x)}{q(x)}dx \approx \int h(x)\frac{p(x)}{q(x)}\frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x)dx$$

$$= \frac{1}{N}\sum_{i=1}^{N}\frac{p(x^{(i)})}{q(x^{(i)})}h(x^{(i)}) = \sum_{i=1}^{N}w^{(i)}h(x^{(i)})$$

where $w^{(i)} \propto \frac{p(x^{(i)})}{q(x^{(i)})}$ is the importance weight and we can think informally of $p(x)$ as

$$p(x) \approx \sum_{i=1}^{N}w^{(i)}\delta_{x^{(i)}}(x), \quad \sum_{i=1}^{N}w^{(i)} = 1$$

There are numerous versions of Monte Carlo samplers, including Markov chain Monte Carlo, simulated annealing, importance sampling, quasi-Monte Carlo, ...

There are numerous versions of Monte Carlo samplers, including Markov chain Monte Carlo, simulated annealing, importance sampling, quasi-Monte Carlo, ...

Here we limit attention to Sequential Monte Carlo methods, which are proving very successful for solving challenging state-space modelling problems.

Important trick: Automatic marginalisation using Monte Carlo:

If

$$(x, \theta) \sim p(x, \theta|y)$$

Important trick: Automatic marginalisation using Monte Carlo:

If

$$(x, \theta) \sim p(x, \theta|y)$$

then

$$x \sim p(x|y)$$

and

$$\theta \sim p(\theta|y)$$

Important trick: Automatic marginalisation using Monte Carlo:

If

$$(x, \theta) \sim p(x, \theta|y)$$

then

$$x \sim p(x|y)$$

and

$$\theta \sim p(\theta|y)$$

[Since

$$p(x, \theta|y) = p(x|y)p(\theta|x, y) = p(\theta|y)p(x|\theta|y)$$

]

State space models, filtering and smoothing

We will focus here on a broad and general class of models. Examples include:

- Hidden Markov models
- Most standard time series models: AR, MA, ARMA,...
- Special models from tracking, computer vision, finance, communications, bioinformatics, ...

First define the notations used. We will consider a very general class of time series models, the state space model. Almost all models of practical utility can be represented within this category, using a state vector of finite dimension. The sequential inference methods presented can readily be extended beyond the Markovian state space models given here, but for simplicity we retain the standard Markovian setup.

Note: from here on column vectors are denoted in standard typeface, e.g. x_t, and matrices are denoted by capitals, e.g. B. This avoids some cumbersome heavy typeface notations.

- Consider a time series with states x_t, $t \in \{0, 2, ..., T\}$.

Note: from here on column vectors are denoted in standard typeface, e.g. x_t, and matrices are denoted by capitals, e.g. B. This avoids some cumbersome heavy typeface notations.

- Consider a time series with states x_t, $t \in \{0, 2, ..., T\}$.
- The states evolve in time according to a probability model. Assume a Markov structure, i.e.

$$p(x_{t+1}|x_0, x_1, ..., x_t) = f(x_{t+1}|x_t) \tag{1}$$

Note: from here on column vectors are denoted in standard typeface, e.g. x_t, and matrices are denoted by capitals, e.g. B. This avoids some cumbersome heavy typeface notations.

- Consider a time series with states x_t, $t \in \{0, 2, ..., T\}$.
- The states evolve in time according to a probability model. Assume a Markov structure, i.e.

$$p(x_{t+1}|x_0, x_1, ..., x_t) = f(x_{t+1}|x_t) \tag{1}$$

- The states are 'partially' observed through a likelihood function for observations $\{y_t\}$ which are assumed independent given the states, i.e.

$$p(y_{t+1}|x_0, x_1, ..., x_t, x_{t+1}, y_0, y_1, ..., y_t) = g(y_{t+1}|x_{t+1}) \tag{2}$$

Summarise as a 'state space' or 'dynamical' model:

$$
\begin{aligned}
x_{t+1} &\sim f(x_{t+1}|x_t) && \text{State evolution density} \\
y_{t+1} &\sim g(y_{t+1}|x_{t+1}) && \text{Observation density}
\end{aligned}
\tag{3}
$$

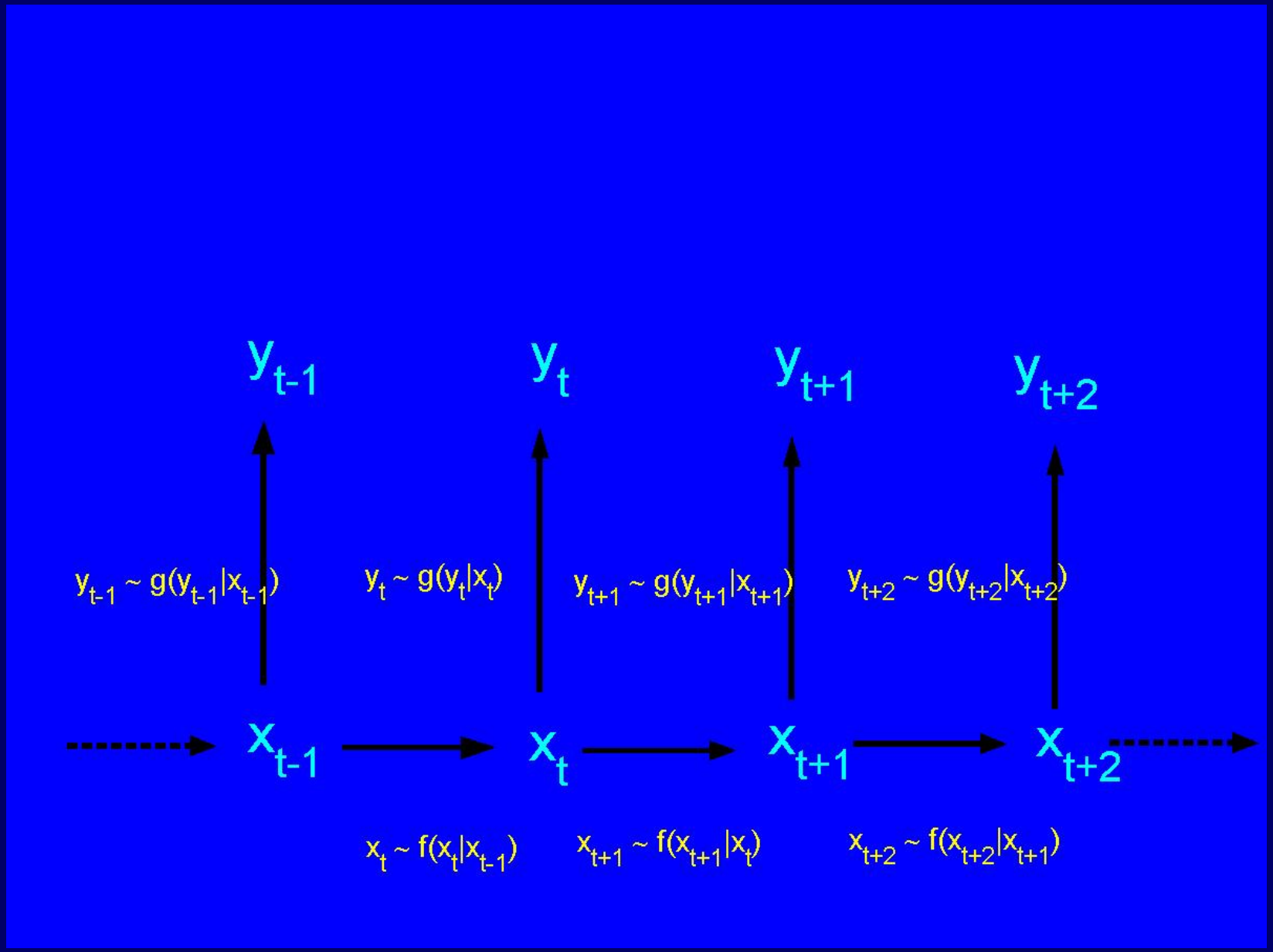
Summarise as a 'state space' or 'dynamical' model:

$$\begin{aligned} x_{t+1} &\sim f(x_{t+1}|x_t) && \text{State evolution density} \\ y_{t+1} &\sim g(y_{t+1}|x_{t+1}) && \text{Observation density} \end{aligned} \tag{3}$$

Joint density can be expressed using the probability chain rule:

$$p\left(x_{0:t}, y_{0:t}\right) = f(x_0)\prod_{i=1}^{t} f(x_t|x_{t-1}) \prod_{i=0}^{t} g(y_t|x_t)$$

where $f(x_0)$ is the distribution of the initial state, $x_{0:t} \stackrel{\triangle}{=} (x_0, ..., x_t)$ and $y_{0:t} \stackrel{\triangle}{=} (y_0, ..., y_t)$.


y_{t-1}
y_t
y_{t+1}
y_{t+2}
$y_{t-1} \sim g(y_{t-1}|x_{t-1})$
$y_t \sim g(y_t|x_t)$
$y_{t+1} \sim g(y_{t+1}|x_{t+1})$
$y_{t+2} \sim g(y_{t+2}|x_{t+2})$
x_{t-1}
x_t
x_{t+1}
x_{t+2}
$x_t \sim f(x_t|x_{t-1})$
$x_{t+1} \sim f(x_{t+1}|x_t)$
$x_{t+2} \sim f(x_{t+2}|x_{t+1})$

Example: linear AR model observed in noise

$$z_t = \sum_{i=1}^{P} a_i z_{t-i} + e_t$$

$$y_t = z_t + w_t$$

with e_t and w_t independently distributed as zero mean Gaussians with variance σ_e^2 and σ_w^2, respectively (fixed and known).

a_i are the AR coefficients, of order P, also assumed here to be fixed and known.

We observe the noisy signal y_t.

The only unknown here is the signal z_t.

Form state vector as:

$$x_t = [z_t, z_{t-1}, ..., z_{t-P+1}]^T \tag{4}$$

Then a state space model in terms of the signal values is obtained as:

$$x_t = Ax_{t-1} + \epsilon_t \tag{5}$$

$$y_t = Bx_t + w_t \tag{6}$$

where:

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_P \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$\Sigma_\epsilon = \begin{bmatrix} \sigma_e^2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Alternatively, in terms of state evolution and observation densities:

$$f(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}|Ax_t, \Sigma_\epsilon)$$
$$g(y_t|x_t) = \mathcal{N}(y_t|Bx_t, \sigma_w^2) \tag{7}$$

Alternatively, in terms of state evolution and observation densities:

$$f(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}|Ax_t, \Sigma_\epsilon)$$
$$g(y_t|x_t) = \mathcal{N}(y_t|Bx_t, \sigma_w^2) \tag{7}$$

This is an example of the linear Gaussian state space model, an important special case that is used extensively to construct algorithms in the nonlinear non-Gaussian case (extended Kalman filters, Rao-Blackwellised particle filters, ...).

Example: Non-linear Model:

$$
\begin{aligned}
x_t &= A(x_{t-1}) + v_t \\
&= \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + v_t \\
y_t &= B(x_t) + w_t \\
&= \frac{(x_t)^2}{20} + w_t
\end{aligned}
$$

where $v_t \sim \mathcal{N}(0, \sigma_v^2)$ and $w_t \sim \mathcal{N}(0, \sigma_w^2)$.

This may be expressed in terms of density functions as:

$$
\begin{aligned}
f(x_{t+1}|x_t) &= \mathcal{N}(x_{t+1}|A(x_t), \sigma_v^2) \\
g(y_t|x_t) &= \mathcal{N}(y_t|B(x_t), \sigma_w^2)
\end{aligned}
$$

Estimation tasks

Given observed data up to time t:

$$y_{0:t} \triangleq (y_0, ..., y_t)$$

Wish to infer the 'hidden states':

$$x_{0:t} \triangleq (x_0, ..., x_t)$$

Specifically:

- Filtering:

  Wish to estimate $p(x_t|y_{0:t})$ itself or expectations of the form

$$\overline{h} = \mathbb{E}h(x_t) = \int h(x_t)p(x_t|y_{0:t})dx_t$$

  e.g. $h(x_t) = x_t$ - posterior mean estimation (MMSE estimator)

Specifically:

- Filtering:

  Wish to estimate $p(x_t|y_{0:t})$ itself or expectations of the form

$$\overline{h} = \mathbb{E}h(x_t) = \int h(x_t)p(x_t|y_{0:t})dx_t$$

  e.g. $h(x_t) = x_t$ - posterior mean estimation (MMSE estimator)

- Smoothing ('fixed lag'):

$$p(x_{t-L}|y_{0:t})$$

Specifically:

- Filtering:

  Wish to estimate $p(x_t|y_{0:t})$ itself or expectations of the form

  $$\overline{h} = \mathbb{E}h(x_t) = \int h(x_t)p(x_t|y_{0:t})dx_t$$

  e.g. $h(x_t) = x_t$ - posterior mean estimation (MMSE estimator)

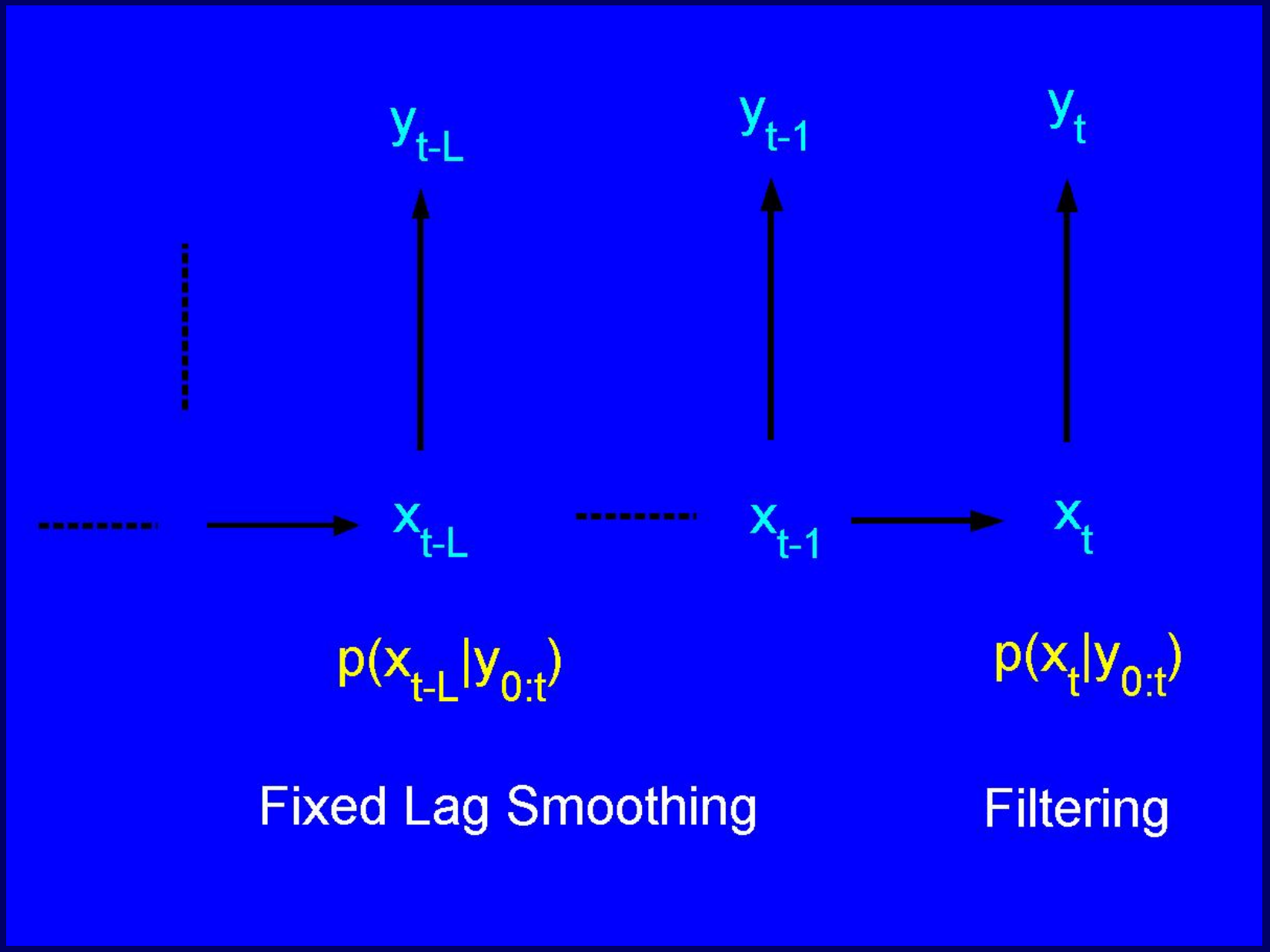
- Smoothing ('fixed lag'):

  $$p(x_{t-L}|y_{0:t})$$

- Smoothing ('fixed interval'):

  Estimate entire state sequence given all data:

  $$p(x_{0:T}|y_{0:T})$$


y_{t-L}
y_{t-1}
y_t
x_{t-L}
x_{t-1}
x_t
$p(x_{t-L}|y_{0:t})$
$p(x_t|y_{0:t})$
Fixed Lag Smoothing
Filtering

Filtering

At time t, Suppose we have $p(x_t|y_{0:t})$ but wish to find $p(x_{t+1}|y_{0:t+1})$. In principle we can use the filtering recursions:

Prediction step:

$$\begin{aligned} p(x_{t+1}|y_{0:t}) &= \int p(x_t, x_{t+1}|y_{0:t})dx_t \\ &= \int p(x_t|y_{0:t})p(x_{t+1}|x_t, y_{0:t})dx_t \\ &= \int p(x_t|y_{0:t})f(x_{t+1}|x_t)dx_t \end{aligned} \tag{8}$$

Filtering

At time t, Suppose we have $p(x_t|y_{0:t})$ but wish to find $p(x_{t+1}|y_{0:t+1})$. In principle we can use the filtering recursions:

Prediction step:

$$\begin{aligned} p(x_{t+1}|y_{0:t}) &= \int p(x_t, x_{t+1}|y_{0:t})dx_t \\ &= \int p(x_t|y_{0:t})p(x_{t+1}|x_t, y_{0:t})dx_t \\ &= \int p(x_t|y_{0:t})f(x_{t+1}|x_t)dx_t \end{aligned} \tag{8}$$

Correction step (Bayes' Theorem):

$$p(x_{t+1}|y_{0:t+1}) = \frac{g(y_{t+1}|x_{t+1})p(x_{t+1}|y_{0:t})}{p(y_{t+1}|y_{0:t})} \tag{9}$$

The sequential scheme is as follows:

Time	$t-1$		t		$t+1$	...
Data	y_{t-1}		y_t		y_{t+1}	
Filtering	$p(x_{t-1}\|y_{0:t-1})$		$p(x_t\|y_{0:t})$		$p(x_{t+1}\|y_{0:t+1})$	
Prediction		$p(x_t\|y_{0:t-1})$		$p(x_{t+1}\|y_{0:t})$		

However, in the general case the integral is intractable and approximations must be used. (x_t high-dimensional, $f()$, $g()$ non-Gaussian, ...)

Linear Gaussian models - the Kalman filter

[(Anderson and Moore (79), Harvey (89)]

- In cases where the state space model is linear and Gaussian, the classic Kalman filter can be applied. In this case we have:

$$f(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}|Ax_t, C)$$
$$g(y_t|x_t) = \mathcal{N}(y_t|Bx_t, D) \tag{10}$$

  where $\mathcal{N}(x|\mu, Q)$ is the Gaussian density function with mean vector μ and covariance matrix Q.

- We can write this equivalently as:

$$x_{t+1} = Ax_t + v_t \tag{11}$$

$$y_t = Bx_t + w_t \tag{12}$$

where v_t and w_t are zero mean Gaussian vectors with covariance matrices C and D, respectively. v_t and w_t are independent over time and also independent of one another (not strictly necessary)

- We can write this equivalently as:

$$x_{t+1} = Ax_t + v_t \tag{11}$$

$$y_t = Bx_t + w_t \tag{12}$$

where v_t and w_t are zero mean Gaussian vectors with covariance matrices C and D, respectively. v_t and w_t are independent over time and also independent of one another (not strictly necessary)

- We also require that the initial state be Gaussian distributed:

$$p(x_0) = \mathcal{N}(x_0|\mu_0, P_0)$$

- We first require $p(x_{t+1}|y_{0:t})$, the prediction step from the above filtering recursion:

$$p(x_{t+1}|y_{0:t}) = \int p(x_t|y_{0:t}) f(x_{t+1}|x_t) dx_t$$

- We first require $p(x_{t+1}|y_{0:t})$, the prediction step from the above filtering recursion:

$$p(x_{t+1}|y_{0:t}) = \int p(x_t|y_{0:t}) f(x_{t+1}|x_t) dx_t$$

- Suppose that we have already that at time t:

$$p(x_t|y_{0:t}) = \mathcal{N}(x_t|\mu_t, P_t)$$

- Now, from 11 we have

$$x_{t+1} = Ax_t + v_t$$

- Now, from 11 we have

$$x_{t+1} = Ax_t + v_t$$

Thus from standard transformation of variables theory (linear Gaussian case) we have:

$$x_t \sim \mathcal{N}(x_t|\mu_t, P_t)$$

$$x_{t+1} = Ax_t + v_t$$

- Now, from 11 we have

$$x_{t+1} = Ax_t + v_t$$

Thus from standard transformation of variables theory (linear Gaussian case) we have:

$$x_t \sim \mathcal{N}(x_t|\mu_t, P_t)$$

$$x_{t+1} = Ax_t + v_t$$

Therefore:

$$p(x_{t+1}|y_{0:t}) = \mathcal{N}(x_{t+1}|\mu_{t+1|t}, P_{t+1|t})$$

where:

$$\mu_{t+1|t} = A\mu_t, \; P_{t+1|t} = C + AP_tA^T \tag{13}$$

- Now, the correction step of the above filtering recursion is

$$p(x_{t+1}|y_{0:t+1}) = \frac{g(y_{t+1}|x_{t+1})p(x_{t+1}|y_{0:t})}{p(y_{t+1}|y_{0:t})} \tag{14}$$

- Now, the correction step of the above filtering recursion is

$$p(x_{t+1}|y_{0:t+1}) = \frac{g(y_{t+1}|x_{t+1})p(x_{t+1}|y_{0:t})}{p(y_{t+1}|y_{0:t})} \tag{14}$$

- Substituting the above Gaussian forms into the numerator gives:

$$\begin{aligned} p(x_{t+1}|y_{0:t+1}) &\propto \mathcal{N}(y_{t+1}|Bx_{t+1}, D)\mathcal{N}(x_{t+1}|\mu_{t+1|t}, P_{t+1|t}) \\ &\propto \exp(-\frac{1}{2}\{[y_{t+1} - Bx_{t+1}]^T D^{-1}[y_{t+1} - Bx_{t+1}]\}) \\ &\quad \times \exp(-\frac{1}{2}\{[x_{t+1} - \mu_{t+1|t}]^T P_{t+1|t}^{-1}[x_{t+1} - \mu_{t+1|t}]\}) \\ &\propto \exp(-\frac{1}{2}\{[x_{t+1} - \mu_{t+1}]^T P_{t+1}^{-1}[x_{t+1} - \mu_{t+1}]\}) \\ &= \mathcal{N}(x_{t+1}|\mu_{t+1}, P_{t+1}) \end{aligned}$$

where

$$\mu_{t+1} = P_{t+1}(B^T D^{-1} y_{t+1} + P^{-1}_{t+1|t}\mu_{t+1|t}),\ P_{t+1} = (B^T D^{-1} B + P_{t+1|t})^{-1}$$

where

$$\mu_{t+1} = P_{t+1}(B^T D^{-1} y_{t+1} + P^{-1}_{t+1|t}\mu_{t+1|t}),\ P_{t+1} = (B^T D^{-1} B + P_{t+1|t})^{-1}$$

- This expression can be rearranged using the matrix inversion lemma to give:

$$\mu_{t+1} = \mu_{t+1|t} + K_t(y_{t+1} - B\mu_{t+1|t}), \qquad \text{and} \qquad P_{t+1} = (I - K_t B)P_{t+1|t}$$

where

$$K_t = P_{t+1|t}B^T(BP_{t+1|t}B^T + D)^{-1}$$

- Hence the whole Kalman filtering recursion can be summarised as:

$$\mu_{t+1|t} = A\mu_t \tag{15}$$

$$P_{t+1|t} = C + AP_tA^T \tag{16}$$

$$\mu_{t+1} = \mu_{t+1|t} + K_t(y_{t+1} - B\mu_{t+1|t}) \tag{17}$$

$$P_{t+1} = (I - K_tB)P_{t+1|t} \tag{18}$$

$$K_t = P_{t+1|t}B^T(BP_{t+1|t}B^T + D)^{-1} \tag{19}$$

Things you can do with a Kalman filter

The Kalman filter is a fundamental tool for tracking and on-line estimation problems:

- Estimate the system state sequentially using $\hat{x}_t = \mu_t$
- Obtain an uncertainty measure about the state using $\text{var}(\hat{x}_t) = P_t$
- Recursive least squares. With $C = 0$ we have the same model and updating rules as used in the RLS algorithm - hence RLS is a special case of Kalman.
- Fixed-lag smoothing: augment the state with past states: $x'_t = [x_t\, x_{t-1}\, \ldots\, x_{t-p}]$
- Fixed interval smoothing: the Kalman smoother operates backwards in time, estimating recursively $p(x_t|y_{0:T})$, $t < T$.

Likelihood evaluation.

A key result is that the Kalman filter can sequentially evaluate the likelihood function, $p(y_{0:t})$. This is used for maximum likelihood or maximum a posteriori parameter estimation, and also for Bayesian model choice problems and the Rao-Blackwellised particle filter.

Likelihood evaluation.

A key result is that the Kalman filter can sequentially evaluate the likelihood function, $p(y_{0:t})$. This is used for maximum likelihood or maximum a posteriori parameter estimation, and also for Bayesian model choice problems and the Rao-Blackwellised particle filter.

To see how this works, start from the Kalman prediction step:

$$p(x_{t+1}|y_{0:t}) = \mathcal{N}(x_{t+1}|A\mu_t, C + AP_tA^T)$$

Likelihood evaluation.

A key result is that the Kalman filter can sequentially evaluate the likelihood function, $p(y_{0:t})$. This is used for maximum likelihood or maximum a posteriori parameter estimation, and also for Bayesian model choice problems and the Rao-Blackwellised particle filter.

To see how this works, start from the Kalman prediction step:

$$p(x_{t+1}|y_{0:t}) = \mathcal{N}(x_{t+1}|A\mu_t, C + AP_tA^T)$$

Now, equation 12 expresses y_{t+1} in terms of Bx_{t+1} plus a random Gaussian disturbance w_{t+1} with covariance matrix D:

$$y_{t+1} = Bx_{t+1} + w_{t+1}$$

Hence, again using transformation of Gaussian variables, we can obtain the conditional likelihood:

$$p(y_{t+1}|y_{0:t}) = \mathcal{N}(y_{t+1}|B\mu_{t+1|t}, D + BP_{t+1|t}B^T)$$

Finally, using the probability chain rule, we obtain the likelihood function:

$$p(y_{0:T}) = p(y_0) \prod_{t=0}^{T-1} p(y_{t+1}|y_{0:t})$$

Numerical methods - or things you can't do with the Kalman filter

- The Kalman filter is optimal only for the linear Gaussian model. In other cases the Kalman filter will give the best linear estimator in a mean-square error sense, but this may not be good enough for highly non-linear or non-Gaussian models
- There are numerous methods for dealing with more general models, all based on numerical approximations to the filtering recursions of equations 8 and 9, e.g. the Gaussian sum filter, unscented Kalman filter, etc...
- Here we will consider two important examples: the extended Kalman filter (EKF) and the Monte Carlo particle filter

The extended Kalman filter (EKF)

The extended Kalman filter is the classical method for estimating non-linear state-space systems.

- Consider the following non-linear state-space model, which is the non-linear equivalent to equations 11 and 12:

$$x_{t+1} = A(x_t) + v_t \tag{20}$$

$$y_t = B(x_t) + w_t \tag{21}$$

  where $A()$ and $B()$ are now non-linear functions.

- Perform a 1st order Taylor expansion of $A()$ and $B()$ around the points μ_t and $\mu_{t|t-1}$, respectively:

$$A(x_t) \approx A(\mu_t) + \left.\frac{\partial A(x_t)}{\partial x_t}\right|_{x_t=\mu_t} (x_t - \mu_t)$$

$$B(x_t) \approx B(\mu_{t|t-1}) + \left.\frac{\partial B(x_t)}{\partial x_t}\right|_{x_t=\mu_{t|t-1}} (x_t - \mu_{t|t-1})$$

- Substituting these approximations into the state-space model leads to a linearized set of equations which can be solved using the standard Kalman filter
- Limitatations - the approximation is still unimodal, hence for multimodal distributions the filter will fail
- Also the tracking performance and error covariance estimates will be sub-optimal

Monte Carlo Filtering

Consider numerical estimation of the following expectation:

$$\overline{h} = \mathbb{E}h(x_t) = \int h(x_t)p(x_t|y_{0:t})dx_t$$

- If the integral is intractable then we can resort to a Monte Carlo integration:

$$\widehat{\overline{h}} = 1/N \sum_{i=1}^{N} h(x_t^{(i)}), \quad \text{where} \;\; x_t^{(i)} \overset{\text{iid}}{\sim} p(x_t|y_{0:t}) \tag{22}$$

Monte Carlo Filtering

Consider numerical estimation of the following expectation:

$$\overline{h} = \mathbb{E}h(x_t) = \int h(x_t)p(x_t|y_{0:t})dx_t$$

- If the integral is intractable then we can resort to a Monte Carlo integration:

$$\widehat{\overline{h}} = 1/N \sum_{i=1}^{N} h(x_t^{(i)}), \quad \text{where} \quad x_t^{(i)} \overset{\text{iid}}{\sim} p(x_t|y_{0:t}) \tag{22}$$

- More generally, when we cannot sample directly from $p(x_t|y_{0:t})$, we can sample from another distribution $q(x_t)$ ('importance function') having the same support as $p(x_t|y_{0:t})$. So we make N random draws from $q()$ instead of $p()$:

$$x_t^{(i)} \sim q(x_t), \qquad i = 1, ..., N$$

- Now we have to make a correction to ensure that the expectation estimate is good. It turns out that the required correction is proportional to the ratio $p()/q()$, which is termed the importance weight:

$$w_t^{(i)} \propto \frac{p(x_t^{(i)}|y_{0:t})}{q(x_t^i)}$$

- Now we have to make a correction to ensure that the expectation estimate is good. It turns out that the required correction is proportional to the ratio $p()/q()$, which is termed the importance weight:

$$w_t^{(i)} \propto \frac{p(x_t^{(i)}|y_{0:t})}{q(x_t^i)}$$

- If we normalise the importance weights such that $\sum_{i=1}^{N} w_t^{(i)} = 1$ we can form an empirical approximation to the filtering density:

$$p(x_t|y_{0:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(x_t) \tag{23}$$

from which expectation estimates can be obtained as:

$$\widehat{\overline{h}} = \sum_{i=1}^{N} w^{(i)} h(x_t^{(i)}), \tag{24}$$

$$\text{where} \quad x_t^{(i)} \overset{\text{iid}}{\sim} q(x_t), \quad w_t^{(i)} \propto p(x_t^{(i)}|y_{0:t})/q(x_t^{(i)}), \quad \sum_{i=1}^{N} w_t^{(i)} = 1 \tag{25}$$

from which expectation estimates can be obtained as:

$$\widehat{\overline{h}} = \sum_{i=1}^{N} w^{(i)} h(x_t^{(i)}), \tag{24}$$

$$\text{where} \quad x_t^{(i)} \overset{\text{iid}}{\sim} q(x_t), \quad w_t^{(i)} \propto p(x_t^{(i)}|y_{0:t})/q(x_t^{(i)}), \quad \sum_{i=1}^{N} w_t^{(i)} = 1 \tag{25}$$

i.e.

$$\begin{aligned} \overline{h} = \mathbb{E}h(x_t) &= \int h(x_t) p(x_t|y_{0:t}) dx_t \\ &\approx \int h(x_t) \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(x_t) dx_t = \sum_{i=1}^{N} w^{(i)} h(x_t^{(i)}) \end{aligned}$$

- **Resampling** (this will prove important in the sequential setting). We now have the option of resampling the the particles so they have uniform weights:

$$\text{Set} \quad x'^{(i)}_t = x^{(i)}_t \quad \text{with probability} \quad w^{(i)}_t$$

and set $w'^{(i)}_{t+1} = 1/N$.

- **Resampling** (this will prove important in the sequential setting). We now have the option of resampling the the particles so they have uniform weights:

$$\text{Set} \quad x'^{(i)}_t = x^{(i)}_t \quad \text{with probability} \quad w^{(i)}_t$$

and set $w'^{(i)}_{t+1} = 1/N$.

While this is unnecessary in the static case, and would always increase the Monte Carlo variation of our estimators, it is a vital component of the sequential schemes which follow, limiting degeneracy of the importance weights over time. Note that resampling schemes can incorporate variance reduction strategies such as stratification in order to improve performance.

- We now have a means for approximating $p(x_t|y_{0:t})$ and also expectations of x_t.
- But, how do we adapt this to the sequential context? (Note that $p(x_t|y_{0:t})$ cannot in general be evaluated).

Sequential Monte Carlo (SMC) - the Particle filter

A generic solution involves repeated importance sampling/resampling sequentially through time (particle filter) (see e.g. Gordon et al. 1993 (IEE), Kitagawa (1993, J. Comp.Graph. Stats.), Doucet Godsill Andrieu 2000 (Stats. amd computing), Liu and Chen 1997 (JASA)).

The SMC scheme mimics the filtering recursions as follows:

Sequential Monte Carlo (SMC) - the Particle filter

A generic solution involves repeated importance sampling/resampling sequentially through time (particle filter) (see e.g. Gordon et al. 1993 (IEE), Kitagawa (1993, J. Comp.Graph. Stats.), Doucet Godsill Andrieu 2000 (Stats. amd computing), Liu and Chen 1997 (JASA)).

The SMC scheme mimics the filtering recursions as follows:

- Suppose we have available a collection of samples, or 'particles' drawn randomly from the filtering density at time t:

$$x_t^{(i)} \sim p(x_t|y_{0:t}), \qquad i = 1, ..., N \quad (N \text{ large})$$

  i.e.

$$p(x_t|y_{0:t}) \simeq \frac{1}{N} \sum_{i=1}^{N} \delta(x_t - x_t^{(i)})$$

- Substitute this into the prediction equation:

$$
\begin{aligned}
p(x_{t+1}|y_{0:t}) &= \int p(x_t|y_{0:t}) f(x_{t+1}|x_t) dx_t \\
&\approx \int \frac{1}{N} \sum_{i=1}^{N} \delta(x_t - x_t^{(i)}) f(x_{t+1}|x_t) dx_t \\
&= \frac{1}{N} \sum_{i=1}^{N} f(x_{t+1}|x_t^{(i)})
\end{aligned}
$$

- Substitute this into the prediction equation:

$$\begin{aligned} p(x_{t+1}|y_{0:t}) &= \int p(x_t|y_{0:t}) f(x_{t+1}|x_t) dx_t \\ &\approx \int \frac{1}{N} \sum_{i=1}^{N} \delta(x_t - x_t^{(i)}) f(x_{t+1}|x_t) dx_t \\ &= \frac{1}{N} \sum_{i=1}^{N} f(x_{t+1}|x_t^{(i)}) \end{aligned}$$

- Then perform the correction step using Bayes' theorem:

$$p(x_{t+1}|y_{0:t+1}) \approx \frac{1}{N} \frac{g(y_{t+1}|x_{t+1}) \sum_{i=1}^{N} f(x_{t+1}|x_t^{(i)}))}{p(y_{t+1}|y_{0:t})}$$

- SMC is a collection of methods for drawing random samples from the above Monte Carlo approximation to $p(x_{t+1}|y_{0:t+1})$, i.e. producing a new set of random draws:

$$x_{t+1}^{(i)} \sim p(x_{t+1}|y_{0:t+1}), \qquad i = 1, ..., N \quad (N \text{ large})$$

- SMC is a collection of methods for drawing random samples from the above Monte Carlo approximation to $p(x_{t+1}|y_{0:t+1})$, i.e. producing a new set of random draws:

$$x_{t+1}^{(i)} \sim p(x_{t+1}|y_{0:t+1}), \qquad i = 1, ..., N \quad (N \text{ large})$$

- These random samples can be obtained by many means:
  - Rejection sampling (slow but exact - requires an envelope function)
  - Importance sampling - most common procedure
  - MCMC - effective but slow [note that MCMC can be applied in conjunction with IS - then very effective]
  - Special schemes such as annealed importance sampling.

There are many variants on schemes to achieve this (Bootstrap filter (Gordon et al. 1993, Sequential Importance sampling, (Doucet Godsill Andrieu (2000), Liu and Chen (1997)), Auxiliary Particle filters (Pitt and Shephard (1998)), etc.

- Consider updating the filtering distribution from t to $t+1$:

Step 0:

$$p(x_t, x_{t+1}|y_{0:t}) = p(x_t|y_{0:t}) f(x_{t+1}|x_t)$$

Step 1:

$$p(x_{t+1}|y_{0:t}) = \int p(x_t, x_{t+1}|y_{0:t}) dx_t$$

Step 2:

$$p(x_{t+1}|y_{0:t+1}) = \frac{g(y_{t+1}|x_{t+1}) p(x_{t+1}|y_{0:t})}{p(y_{t+1}|y_{0:t})}$$

- We would like to mimic the three steps here by Monte Carlo operations.

- We would like to mimic the three steps here by Monte Carlo operations.
- Suppose we start off with many 'particles' drawn from the filtering distribution $p(x_t|y_{0:t})$. We label these particles as

$$x_t^{(i)}, \quad i = 1, 2, ..., N \ \text{ with } \ N >> 1$$

- We would like to mimic the three steps here by Monte Carlo operations.
- Suppose we start off with many 'particles' drawn from the filtering distribution $p(x_t|y_{0:t})$. We label these particles as

$$x_t^{(i)}, \quad i = 1, 2, ..., N \ \text{ with } \ N >> 1$$

- These can be used to plot histogram estimates of $p(x_t|y_{0:t})$, form Monte Carlo estimates of expectations, ..., in fact perform almost any inference procedure we care to choose, provided N is 'sufficiently' large.

- We can simulate **Step 0** above by taking each particle $x_t^{(i)}$ in turn and generating a new state from the state transition density according to:

$$x_{t+1}^{(i)} \sim f(x_{t+1}|x_t^{(i)})$$

- We can simulate **Step 0** above by taking each particle $x_t^{(i)}$ in turn and generating a new state from the state transition density according to:

$$x_{t+1}^{(i)} \sim f(x_{t+1}|x_t^{(i)})$$

- Each pair $(x_t^{(i)}, x_{t+1}^{(i)})$ is now a joint random sample from $p(x_t, x_{t+1}|y_{0:t})$.

- We can simulate **Step 0** above by taking each particle $x_t^{(i)}$ in turn and generating a new state from the state transition density according to:

$$x_{t+1}^{(i)} \sim f(x_{t+1}|x_t^{(i)})$$

- Each pair $(x_t^{(i)}, x_{t+1}^{(i)})$ is now a joint random sample from $p(x_t, x_{t+1}|y_{0:t})$.
- By construction, $x_{t+1}^{(i)}$ taken on its own is a random sample from the required marginal distribution $p(x_{t+1}|y_{0:t})$, (**Step 1**)

- We can simulate **Step 0** above by taking each particle $x_t^{(i)}$ in turn and generating a new state from the state transition density according to:

$$x_{t+1}^{(i)} \sim f(x_{t+1}|x_t^{(i)})$$

- Each pair $(x_t^{(i)}, x_{t+1}^{(i)})$ is now a joint random sample from $p(x_t, x_{t+1}|y_{0:t})$.
- By construction, $x_{t+1}^{(i)}$ taken on its own is a random sample from the required marginal distribution $p(x_{t+1}|y_{0:t})$, (**Step 1**)

- **Step 2**. We now have samples from $p(x_{t+1}|y_{0:t})$. Step 2 gives us the appropriate importance weight:

$$\begin{aligned} w_{t+1} &\propto \frac{p(x_{t+1}|y_{0:t+1})}{q(x_{t+1})} \\ &\propto \frac{\frac{g(y_{t+1}|x_{t+1})p(x_{t+1}|y_{0:t})}{p(y_{t+1}|y_{0:t})}}{p(x_{t+1}|y_{0:t})} \\ &\propto g(y_{t+1}|x_{t+1}) \end{aligned}$$

- We now have the option of
  1. retaining weighted particles, in which case the weights are accumulated over time as

$$w_{t+1} \propto w_t \, g(y_{t+1}|x_{t+1})$$

- We now have the option of

  1. retaining weighted particles, in which case the weights are accumulated over time as

$$w_{t+1} \propto w_t \, g(y_{t+1}|x_{t+1})$$

  Or:

  2. Resampling the particles so they have uniform weights:

$$\text{Set} \quad x'^{(i)}_{t+1} = x^{(i)}_{t+1} \quad \text{with probability} \quad w_t^{(i)}$$

  and set $w'^{(i)}_{t+1} = 1/N$.

- A basic algorithm with (optional) resampling at every time step, the 'Bootstrap Filter', is thus (Gordon, Salmond and Smith 1993, Kitagawa 1996:

For $t = 1, 2..., T$

For $i = 1, 2, ..., N$

$x_{t+1}^{(i)} \sim f(x_{t+1}^{(i)} | x_t^{(i)})$

$w_{t+1}^{(i)} \propto w_t^{(i)} \, g(y_{t+1} | x_{t+1}^{(i)})$

End

For $i = 1, 2, ..., N$

(Optional) Resample $x_{t+1}^{(i)}$ with probability $w_{t+1}^{(i)}$. Set $w_{t+1}^{(i)} = 1/N$

End

End

Example: standard nonlinear model

$$
\begin{aligned}
x_t &= A(x_{t-1}) + v_t \\
&= \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + v_t \\
y_t &= B(x_t) + w_t \\
&= \frac{(x_t)^2}{20} + w_t
\end{aligned}
$$

where $v_t \sim \mathcal{N}(0, \sigma_v^2)$ and $w_t \sim \mathcal{N}(0, \sigma_w^2)$.

This may be expressed in terms of density functions as:

$$
\begin{aligned}
f(x_{t+1}|x_t) &= \mathcal{N}(x_{t+1}|A(x_t), \sigma_v^2) \\
g(y_t|x_t) &= \mathcal{N}(y_t|B(x_t), \sigma_w^2)
\end{aligned}
$$

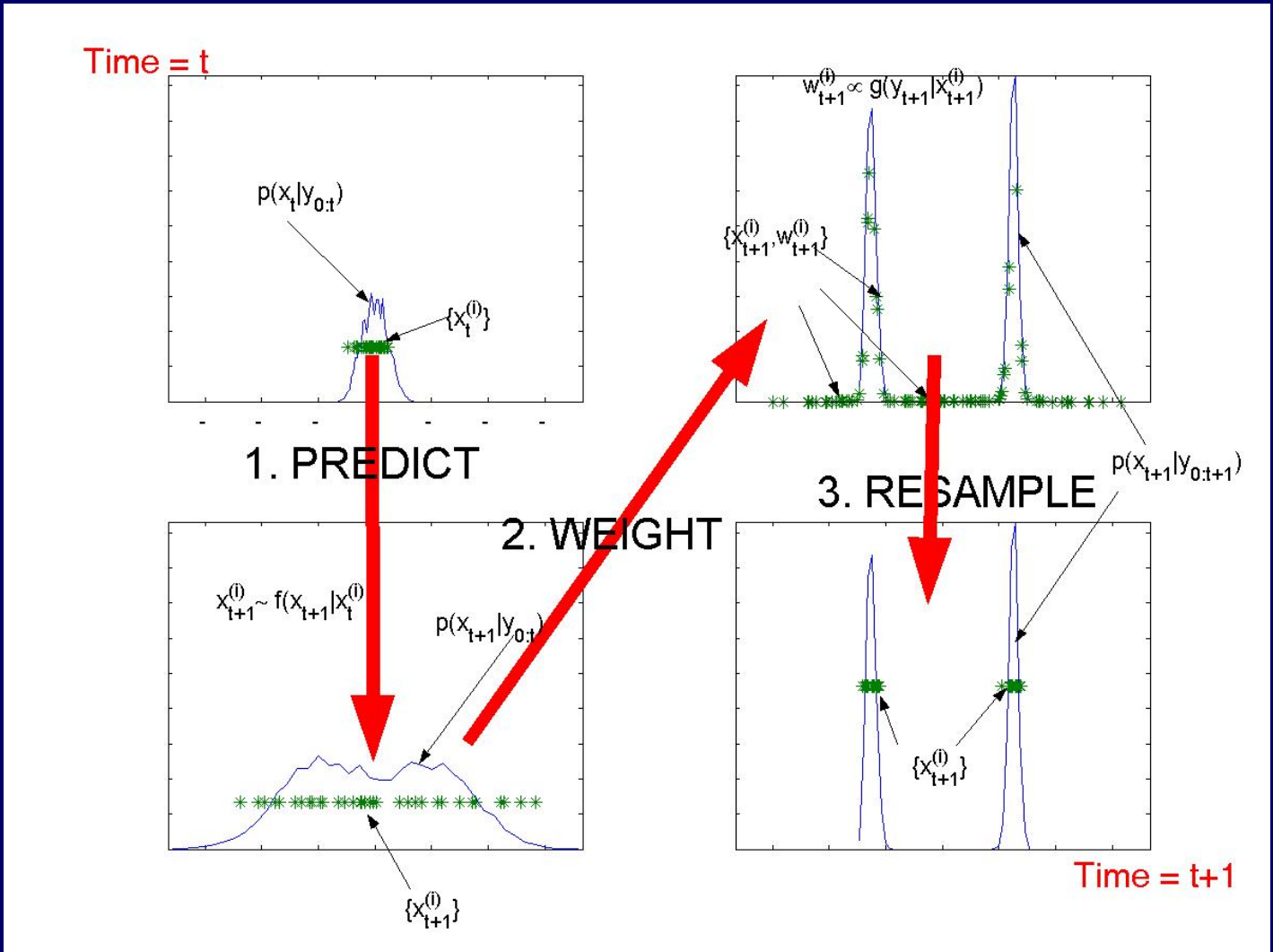


Figure 1: Bootstrap filter operation - nonlinear model

General Sequential Importance Sampling

We can do better in many cases than the basic bootstrap filter, by choosing a better importance function. Consider now the following modified updates:

Step 0':

$$q(x_t, x_{t+1}|y_{0:t+1}) = p(x_t|y_{0:t})q(x_{t+1}|x_t, y_{0:t+1})$$

Step 2':

$$p(x_t, x_{t+1}|y_{0:t+1}) = \frac{g(y_{t+1}|x_{t+1})f(x_{t+1}|x_t)p(x_t|y_{0:t})}{p(y_{t+1}|y_{0:t})}$$

We now consider $q(x_t, x_{t+1}|y_{0:t+1})$ to be an importance function for $p(x_t, x_{t+1}|y_{0:t+1})$. The importance weight for **Step 2'** is hence modified to:

$$w_{t+1}^{(i)} \propto w_t^{(i)} \frac{g(y_{t+1}|x_{t+1}^{(i)}) f(x_{t+1}^{(i)}|x_t^{(i)})}{q(x_{t+1}^{(i)}|x_t^{(i)})} \tag{26}$$

This is the general sequential importance (SIS) sampling method (Liu and Chen, 1998, Doucet, Godsill and Andrieu, 2000).

We now consider $q(x_t, x_{t+1}|y_{0:t+1})$ to be an importance function for $p(x_t, x_{t+1}|y_{0:t+1})$. The importance weight for **Step 2'** is hence modified to:

$$w_{t+1}^{(i)} \propto w_t^{(i)} \frac{g(y_{t+1}|x_{t+1}^{(i)}) f(x_{t+1}^{(i)}|x_t^{(i)})}{q(x_{t+1}^{(i)}|x_t^{(i)})} \tag{26}$$

This is the general sequential importance (SIS) sampling method (Liu and Chen, 1998, Doucet, Godsill and Andrieu, 2000).

Important special cases:

- $q(x_{t+1}|x_t) = f(x_{t+1}|x_t)$ - bootstrap filter (Gordon et al 1993, Kitagawa 1996) - 'prior' sampling
- $q(x_{t+1}|x_t) = p(x_{t+1}|x_t, y_{t+1})$ - sequential imputations Liu et al. 1995 - optimal importance function Doucet, Godsill and Andrieu 2000).

Repeated application over time (without resampling) leads to degeneracy of the weights - all the mass becomes concentrated on a few i - hence estimates are poor.

Repeated application over time (without resampling) leads to degeneracy of the weights - all the mass becomes concentrated on a few i - hence estimates are poor.

The resampling procedure (choosing $x_{t+1}^{(i)}$ with probability $w_{t+1}^{(i)}$) alleviates this - SIR (Gordon et al 1993).

eg. Measure degeneracy by estimating the variance of $w_{t+1}^{(i)}$ - since reduction in effective sample size is approximately (1+var($w_{t+1}^{(i)}$)).

A Basic Importance Sampling/Resampling Particle Filter

The first step initialises the initial states of the filter at $t = 0$:

$$x_0^{(i)} \sim p(x_0|y_0), \;\; i = 1, 2, ..., N$$

where it is assumed that this draw can be made easily (use MCMC or static IS if not).

Then, for t=0,1,2,...

- At time t, have

$$p(x_t|y_{0:t}) \simeq \hat{p}(x_t|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

- At time t, have

$$p(x_t|y_{0:t}) \simeq \hat{p}(x_t|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

- For $i = 1, ..., N$:

$$x_{t+1}^{(i)} \sim q(x_{t+1}|x_t^{(i)})$$

- At time t, have

$$p(x_t|y_{0:t}) \simeq \hat{p}(x_t|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

- For $i = 1, ..., N$:

$$x_{t+1}^{(i)} \sim q(x_{t+1}|x_t^{(i)})$$

Update the importance weight:

$$w_{t+1}^{(i)} \propto w_t^{(i)} \frac{g(y_{t+1}|x_{t+1}^{(i)}) f(x_{t+1}^{(i)}|x_t^{(i)})}{q(x_{t+1}^{(i)}|x_t^{(i)})}$$

- At time t, have

$$p(x_t|y_{0:t}) \simeq \hat{p}(x_t|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

- For $i = 1, ..., N$:

$$x_{t+1}^{(i)} \sim q(x_{t+1}|x_t^{(i)})$$

  Update the importance weight:

$$w_{t+1}^{(i)} \propto w_t^{(i)} \frac{g(y_{t+1}|x_{t+1}^{(i)}) f(x_{t+1}^{(i)}|x_t^{(i)})}{q(x_{t+1}^{(i)}|x_t^{(i)})}$$

- Optionally, resample $\{x_{t+1}^{(i)}\}$ N times with replacement using weights $w_{t+1}^{(i)}$, and then resetting $w_{t+1}^{(i)} = 1/N$.

The algorithm is now modified to:

For $t = 0, 2..., T$

For $i = 1, 2, ..., N$

$x_{t+1}^{(i)} \sim q(x_{t+1}^{(i)} | x_t^{(i)}, y_{0:t+1})$

$w_{t+1}^{(i)} \propto w_t^{(i)} \frac{g(y_{t+1}|x_{t+1}^{(i)}) f(x_{t+1}^{(i)}|x_t^{(i)})}{q(x_{t+1}^{(i)}|x_t^{(i)}, y_{0:t+1})}$

End

For $i = 1, 2, ..., N$

(Optional) Resample $x_{t+1}^{(i)}$ with probability $w_{t+1}^{(i)}$. Set $w_{t+1}^{(i)} = 1/N$

End

End

Conclusions

- Today we covered the basic SMC algorithm and its underlying concepts:
  - Bayesian Filtering
  - Kalman Filter
  - Bootstrap Filter and General SMC filter
- Tomorrow we cover other exotica, including:
  - General SMC (part II!)
  - Auxiliary particle filter
  - Rao-Blackwellised particle filter
  - Monte Carlo smoothing
  - MCMC particle filters
  - Applications.

Second part of the tutorial will cover:

- General SMC (part II!) and the Auxiliary particle filter
- Rao-Blackwellised particle filter
- Monte Carlo smoothing
- MCMC particle filters
- Applications.

[No time for details of Population Monte Carlo and Parameter estimation]

General Sequential Importance Sampling (Part II!)

Question: can we rationalise the resampling step? Why use it, and how should it ideally be used?

Answer: yes! Make it part of the importance sampling.

General Sequential Importance Sampling (Part II!)

Question: can we rationalise the resampling step? Why use it, and how should it ideally be used?

Answer: yes! Make it part of the importance sampling.

Consider the problem from the point of view of joint simulation from the smoothing density.

- 'Particle' approximation to smoothing density:

$$p(x_{0:t}|y_{0:t}) \simeq \hat{p}(x_{0:t}|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_{0:t} - x_{0:t}^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

  where $w_t^{(i)}$ is the 'weight' of particle $x_{0:t}^{(i)}$. Note that $w_t^{(i)}$ will be uniform at time steps where resampling has been carried out.

- Factorise smoothing density at $t+1$:

$$p(x_{0:t+1}|y_{0:t+1}) = p(x_{0:t}|y_{0:t}) \times \frac{g(y_{t+1}|x_{t+1})f(x_{t+1}|x_t)}{p(y_{t+1}|y_{0:t})}$$

- Factorise smoothing density at $t+1$:

$$p(x_{0:t+1}|y_{0:t+1}) = p(x_{0:t}|y_{0:t}) \times \frac{g(y_{t+1}|x_{t+1})f(x_{t+1}|x_t)}{p(y_{t+1}|y_{0:t})}$$

- Apply importance function $q(x_{0:t+1})$:

$$w_{t+1} \propto p(x_{0:t}|y_{0:t}) \times \frac{g(y_{t+1}|x_{t+1})f(x_{t+1}|x_t)}{q(x_{0:t+1})}$$

- In a sequential setting we assume that $\hat{p}(x_{0:t}|y_{0:t}) = p(x_{0:t}|y_{0:t})$ is the 'truth' (the accumulation of error can be analysed) and apply an importance function which fixes the past history of the new particles to be one of the current particles $x_{0:t}^{(i)}$. This corresponds to choosing an importance function of the form:

$$q(x_{0:t+1}) = q(x_{t+1}|x_{0:t}) \sum_{i=1}^{N} v_t^{(i)} \delta_{x_{0:t}^{(i)}}(x_{0:t})$$

- In a sequential setting we assume that $\hat{p}(x_{0:t}|y_{0:t}) = p(x_{0:t}|y_{0:t})$ is the 'truth' (the accumulation of error can be analysed) and apply an importance function which fixes the past history of the new particles to be one of the current particles $x_{0:t}^{(i)}$. This corresponds to choosing an importance function of the form:

$$q(x_{0:t+1}) = q(x_{t+1}|x_{0:t}) \sum_{i=1}^{N} v_t^{(i)} \delta_{x_{0:t}^{(i)}}(x_{0:t})$$

- Importance weights can then be computed as (see Godsill and Clapp (2001)):

$$w_{t+1}^{(j)} \propto \frac{w_t^{(j)}}{v_t^{(j)}} \times \frac{g(y_{t+1}|x_{t+1}^{(j)}) f(x_{t+1}^{(j)}|x_t^{(j)})}{q(x_{t+1}^{(j)}|x_{0:t}^{(j)})}$$

where $x_{0:t+1}^{(j)} \stackrel{\Delta}{=} (x_{0:t}^{(j)}, x_{t+1}^{(j)})$ is the jth particle drawn from the joint distribution $q(x_{0:t+1})$.

- Note the non-standard term here, $v_t^{(j)}$, which serves to boost or diminish the particle histories in the importance function. This weight v_t should intuitively reflect a particle's importance at $t+1$, whereas its original weight w_t reflects its importance at t.

- Note the non-standard term here, $v_t^{(j)}$, which serves to boost or diminish the particle histories in the importance function. This weight v_t should intuitively reflect a particle's importance at $t+1$, whereas its original weight w_t reflects its importance at t.
- Drawing from $q(x_{0:t})$ with weights v_t can be interpreted as the usual 'resampling' or 'selection' step found in particle filters. The resampling is performed first, but this is only a conceptual difference. It can be performed in a stratified or part-deterministic form to reduce estimator variance.

- Note the non-standard term here, $v_t^{(j)}$, which serves to boost or diminish the particle histories in the importance function. This weight v_t should intuitively reflect a particle's importance at $t+1$, whereas its original weight w_t reflects its importance at t.
- Drawing from $q(x_{0:t})$ with weights v_t can be interpreted as the usual 'resampling' or 'selection' step found in particle filters. The resampling is performed first, but this is only a conceptual difference. It can be performed in a stratified or part-deterministic form to reduce estimator variance.
- The algorithm can be written, for $t = 1, ...,\ j = 1, ...N$ as:
  1. Select $x_{0:t}^{(j)}$ with probability $v_t^{(j)}$
  2. $x_{t+1}^{(j)} \sim q(x_{t+1}|x_{0:t}^{(j)})$
  3. $w_{t+1}^{(j)} \propto \frac{w_t^{(j)}}{v_t^{(j)}} \times \frac{g(y_{t+1}|x_{t+1}^{(j)}) f(x_{t+1}^{(j)}|x_t^{(j)})}{q(x_{t+1}^{(j)}|x_{0:t}^{(j)})}$

- This formulation is of interest because it expresses several common variants on particle filtering in one framework. In particular:

- This formulation is of interest because it expresses several common variants on particle filtering in one framework. In particular:
  - $v_t^{(j)} = w_t^{(j)}$ - this corresponds to the standard SIR filtering method, e.g. the bootstrap filter of Gordon et al. (93). Particles are resampled according to their weight at time t.

- This formulation is of interest because it expresses several common variants on particle filtering in one framework. In particular:
  - $v_t^{(j)} = w_t^{(j)}$ - this corresponds to the standard SIR filtering method, e.g. the bootstrap filter of Gordon et al. (93). Particles are resampled according to their weight at time t.
  - $v_t^{(j)} = 1/N$ - filter with 'no resampling' ('sequential imputations'). In practice would implement in fully stratified form by simply selecting particles deterministically $1...N$.

– $v_t^{(j)} \propto g(y_{t+1}|\hat{x}_{t+1}^{(j)})$, where $\hat{x}_{t+1}^{(j)}$ is some (any!) 'estimate' of the new state. This is the auxiliary particle filter of Pitt and Shephard (1998). This makes use of the particle's importance at time $t+1$ in an efficient way.

– $v_t^{(j)} \propto g(y_{t+1}|\hat{x}_{t+1}^{(j)})$, where $\hat{x}_{t+1}^{(j)}$ is some (any!) 'estimate' of the new state. This is the auxiliary particle filter of Pitt and Shephard (1998). This makes use of the particle's importance at time $t+1$ in an efficient way.

– 'Fully adapted' version of auxiliary filter

$$v_t^{(j)} \propto p(y_{t+1}|x_t^{(j)}, y_{0:t}) = \int p(y_{t+1}, x_{t+1}|x_t^{(j)}, y_{0:t}) dx_{t+1}$$

- not generally computable.

– $v_t^{(j)} \propto g(y_{t+1}|\hat{x}_{t+1}^{(j)})$, where $\hat{x}_{t+1}^{(j)}$ is some (any!) 'estimate' of the new state. This is the auxiliary particle filter of Pitt and Shephard (1998). This makes use of the particle's importance at time $t+1$ in an efficient way.

– 'Fully adapted' version of auxiliary filter

$$v_t^{(j)} \propto p(y_{t+1}|x_t^{(j)}, y_{0:t}) = \int p(y_{t+1}, x_{t+1}|x_t^{(j)}, y_{0:t}) dx_{t+1}$$

- not generally computable.

– When coupled with an unknown parameter:

$$v_t^{(i)} \propto p(y_{t+1}|x_t^{(j)}, y_{0:t}, \theta) = \int p(y_{t+1}, x_{t+1}|\theta, x_t^{(j)}, y_{0:t}) dx_{t+1}$$

, and $q(x_{t+1}|x_t^{(j)}, \theta, y_{0:t+1})$ this becomes the 'reverse order sampling' of Polson et al (U. Chicago Tech. rep. 2008). Only applicable to models where 'Gibbs sampler is available'.

Smoothing with particle filters

[Work with Arnaud Doucet, Mike West and William Fong, see Godsill, Doucet and West JASA (2004), Fong, Godsill, Doucet and West (IEEE Tr. SP 2002)]

- It is possible to extend the particle framework to provide smoothing as well as filtering. Smoothing is very useful in problems where batch processing is required, or some 'lookahead' is allowable in the system.

Smoothing with particle filters

[Work with Arnaud Doucet, Mike West and William Fong, see Godsill, Doucet and West JASA (2004), Fong, Godsill, Doucet and West (IEEE Tr. SP 2002)]

- It is possible to extend the particle framework to provide smoothing as well as filtering. Smoothing is very useful in problems where batch processing is required, or some 'lookahead' is allowable in the system.
- First though, note that the standard filter also gives a smoothed output 'for free' [simply store the whole trajectory for each particle (see last section)]:

$$p(x_{0:t}|y_{0:t}) \simeq \hat{p}(x_{0:t}|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_{0:t} - x_{0:t}^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

Smoothing with particle filters

[Work with Arnaud Doucet, Mike West and William Fong, see Godsill, Doucet and West JASA (2004), Fong, Godsill, Doucet and West (IEEE Tr. SP 2002)]

- It is possible to extend the particle framework to provide smoothing as well as filtering. Smoothing is very useful in problems where batch processing is required, or some 'lookahead' is allowable in the system.
- First though, note that the standard filter also gives a smoothed output 'for free' [simply store the whole trajectory for each particle (see last section)]:

$$p(x_{0:t}|y_{0:t}) \simeq \hat{p}(x_{0:t}|y_{0:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(x_{0:t} - x_{0:t}^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

- We will consider the fixed interval problem ('batch' processing), i.e. estimation of:

$$\{x_0, x_1, x_2, \ldots, x_T\} \text{ from } \{y_0, y_1, y_2, \ldots, y_T\}$$

Fixed lag and other versions can be obtained by suitable modifications to the algorithms.

- First, assume that particle filtering has been done for $t = 1, 2, ..., T$, leading to

$$p(x_t|y_{0:t}) \simeq \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \;\; t = 0, 1, 2, ..., T$$

- First, assume that particle filtering has been done for $t = 1, 2, ..., T$, leading to

$$p(x_t|y_{0:t}) \simeq \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \;\; t = 0, 1, 2, ..., T$$

- Now factorise the smoothing density as follows:

$$p(x_{0:T}|y_{0:T}) = \prod_{t=0}^{T} p(x_t|x_{t+1:T}, y_{0:T})$$

where, by the assumptions of the Markov state-space model:

$$p(x_t|x_{t+1:T}, y_{0:T}) \propto p(x_t|y_{0:t}) f(x_{t+1}|x_t)$$

- First, assume that particle filtering has been done for $t = 1, 2, ..., T$, leading to

$$p(x_t|y_{0:t}) \simeq \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \;\; t = 0, 1, 2, ..., T$$

- Now factorise the smoothing density as follows:

$$p(x_{0:T}|y_{0:T}) = \prod_{t=0}^{T} p(x_t|x_{t+1:T}, y_{0:T})$$

where, by the assumptions of the Markov state-space model:

$$p(x_t|x_{t+1:T}, y_{0:T}) \propto p(x_t|y_{0:t}) f(x_{t+1}|x_t)$$

- This factorisation allows construction of an algorithm operating in the

reverse time direction $t = T, T-1, ..., 0$.

Algorithm: Particle smoother

- Draw $\widetilde{x}_T \sim p(x_T|y_{0:T})$
- For $t = T-1$ to 1:
- Calculate $w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\widetilde{x}_{t+1}|x_t^{(i)})$ for $i = 1, ..., N$
- Choose $\widetilde{x}_t = x_t^{(i)}$ with probability $w_{t|t+1}^{(i)}$
- End

reverse time direction $t = T, T-1, ..., 0$.

Algorithm: Particle smoother

- Draw $\widetilde{x}_T \sim p(x_T|y_{0:T})$
- For $t = T-1$ to 1:
- Calculate $w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\widetilde{x}_{t+1}|x_t^{(i)})$ for $i = 1, ..., N$
- Choose $\widetilde{x}_t = x_t^{(i)}$ with probability $w_{t|t+1}^{(i)}$
- End

The sequence

$$(\widetilde{x}_0, \widetilde{x}_1, \ldots, \widetilde{x}_T)$$

is then an (approximate) random draw from

$$p(x_{0:T}|y_{0:T}) = \prod_{t=0}^{T} p(x_t|x_{t+1:T}, y_{0:T})$$

Repeated application allows Monte Carlo estimation of the smoothed state sequence.

Repeated application allows Monte Carlo estimation of the smoothed state sequence.

Variants on the algorithm also allow MAP smoothing, see Godsill, Doucet and West 2001 (Ann. Inst. St. Math.)

Repeated application allows Monte Carlo estimation of the smoothed state sequence.

Variants on the algorithm also allow MAP smoothing, see Godsill, Doucet and West 2001 (Ann. Inst. St. Math.)

You can also do marginal smoothing using similar ideas (see Hurzeler and Kunsch (1997), Doucet, Godsill and Andrieu (2000).

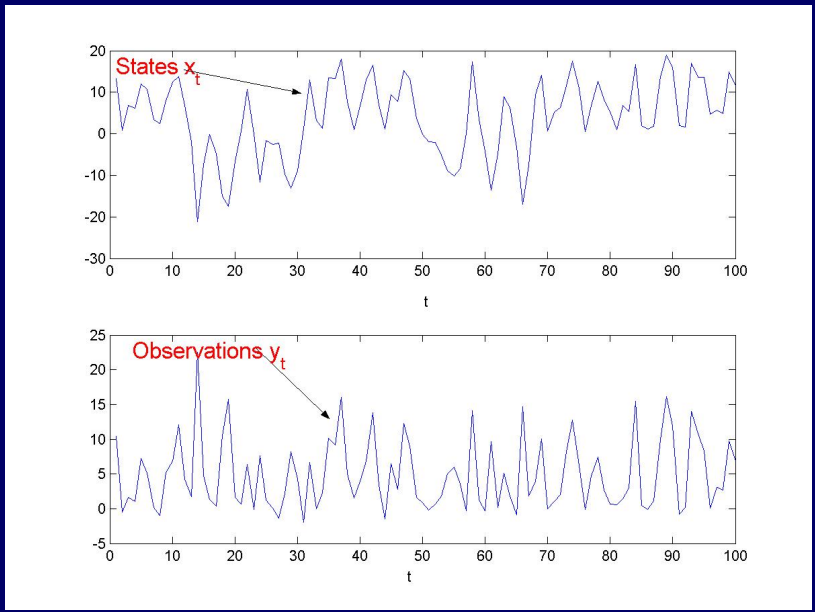
Repeated application allows Monte Carlo estimation of the smoothed state sequence.

Variants on the algorithm also allow MAP smoothing, see Godsill, Doucet and West 2001 (Ann. Inst. St. Math.)

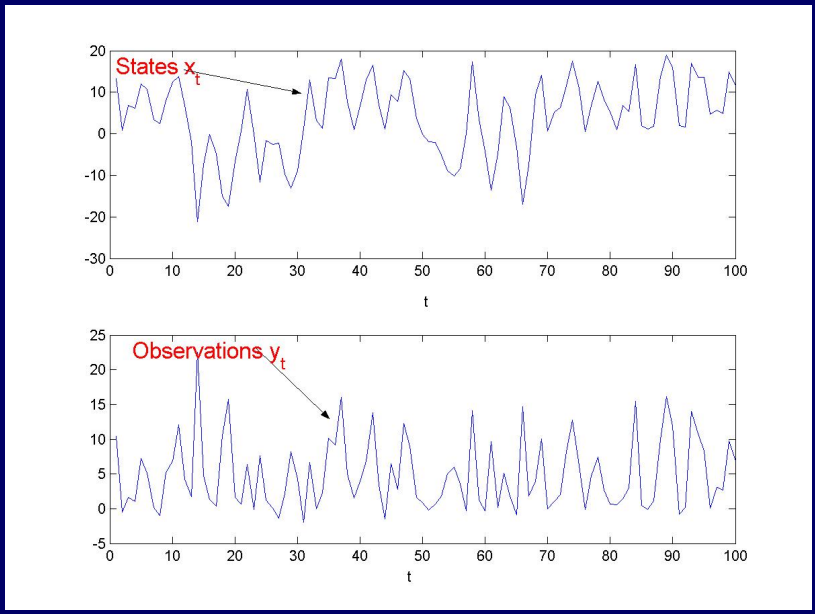
You can also do marginal smoothing using similar ideas (see Hurzeler and Kunsch (1997), Doucet, Godsill and Andrieu (2000).

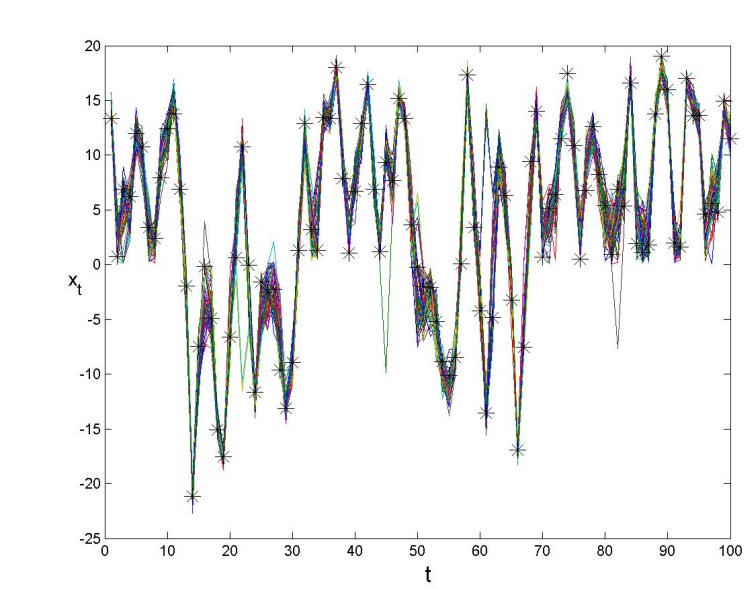
Finally, more recent developments in the area have focussed on the so-called 2-filter formula for smoothing - see recent papers by Briers, Doucet, De Freitas, Fearnhead ...

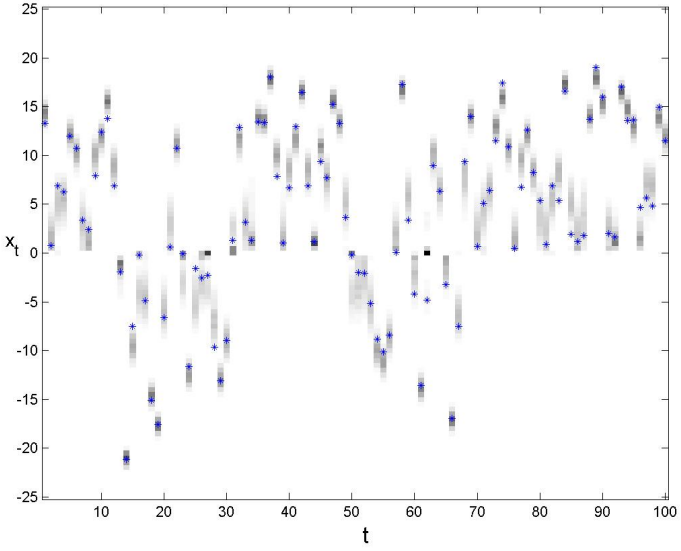
Example - the nonlinear model

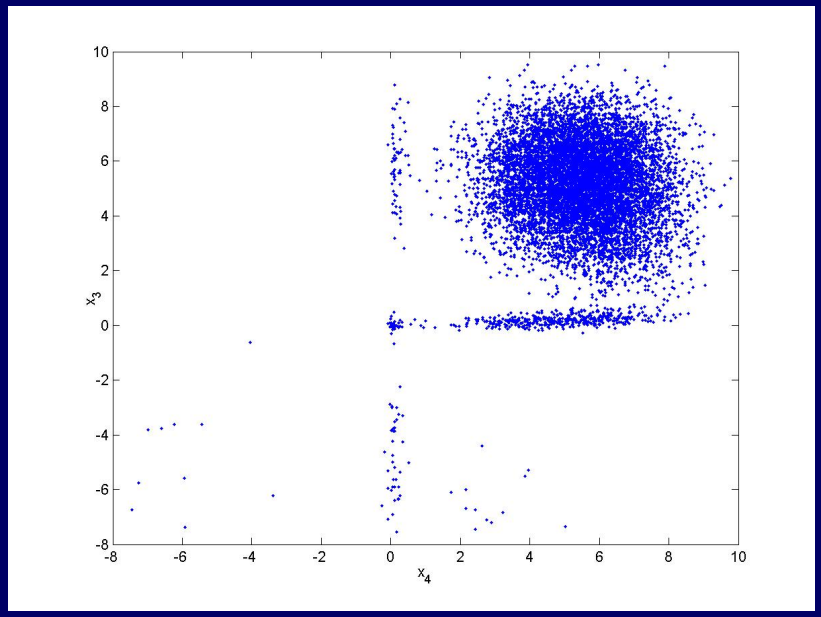


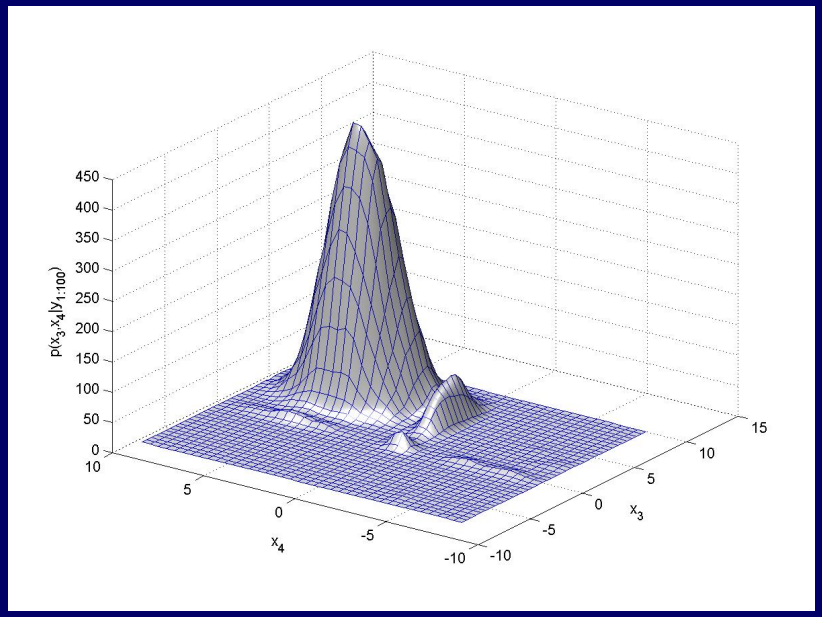
Example - the nonlinear model

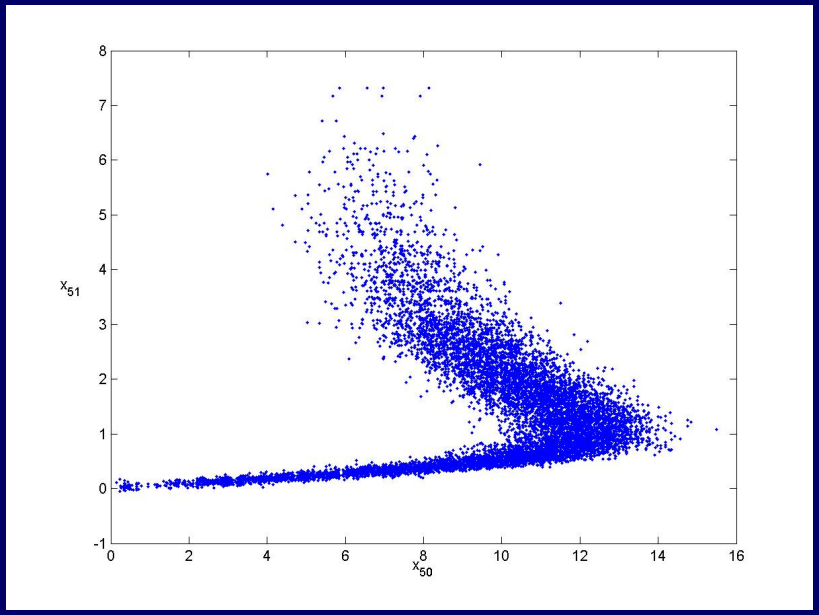


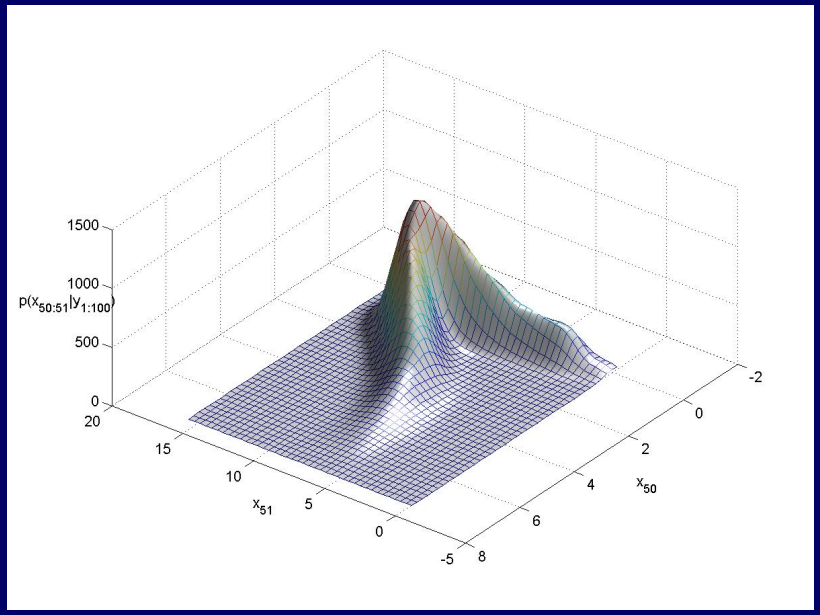





x_3
x_4


$p(x_3,x_4|y_{1:100})$
x_4
x_3


x_{51}
x_{50}


$p(x_{50:51}|y_{1:100})$
x_{51}
x_{50}

Example - a nonlinear TVAR model for non-stationary speech

[Fong, Godsill, Doucet, West (IEEE Tr. SP 2002)]

Signal process $\{z_t\}$ generated as standard Time-varying autoregression:

Example - a nonlinear TVAR model for non-stationary speech

[Fong, Godsill, Doucet, West (IEEE Tr. SP 2002)]

Signal process $\{z_t\}$ generated as standard Time-varying autoregression:

$$f(z_t|z_{t-1:t-P}, a_t, \sigma_{e_t}) = \mathcal{N}\left(\sum_{i=1}^{P} a_{t,i} z_{t-i}, \sigma_{e_t}^2\right)$$

$$g(y_t|x_t, \sigma_{v_t}) = \mathcal{N}\left(x_t, \sigma_{v_t}^2\right)$$

Example - a nonlinear TVAR model for non-stationary speech

[Fong, Godsill, Doucet, West (IEEE Tr. SP 2002)]

Signal process $\{z_t\}$ generated as standard Time-varying autoregression:

$$f(z_t|z_{t-1:t-P}, a_t, \sigma_{e_t}) = \mathcal{N}\left(\sum_{i=1}^{P} a_{t,i} z_{t-i}, \sigma_{e_t}^2\right)$$

$$g(y_t|x_t, \sigma_{v_t}) = \mathcal{N}\left(x_t, \sigma_{v_t}^2\right)$$

- $a_t = (a_{t,1}, a_{t,2}, ..., a_{t,P})$ is the P^{th} order AR coefficient vector
- $\sigma_{e_t}^2$ is the innovation variance at time t.
- $\sigma_{v_t}^2$ is the observation noise variance.
- a_t is assumed to evolve over time as a dynamical model. We choose a nonlinear parameterisation based on time-varying lattice coefficients

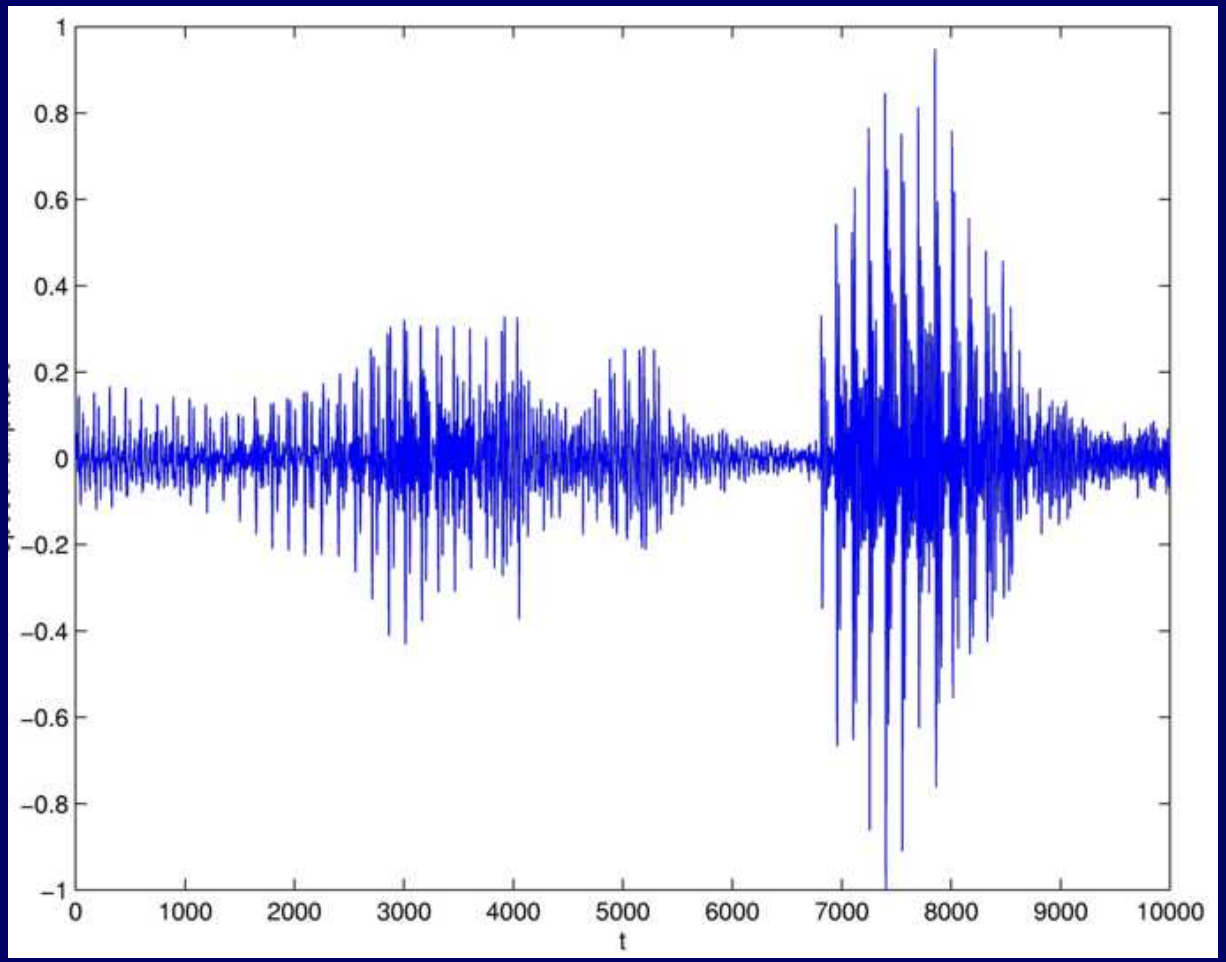


Figure 2: Speech data. 0.62s of a US male speaker saying the words '...rewarded by...'. Sample rate 16kHz, resolution 16-bit, from the TIMIT speech database

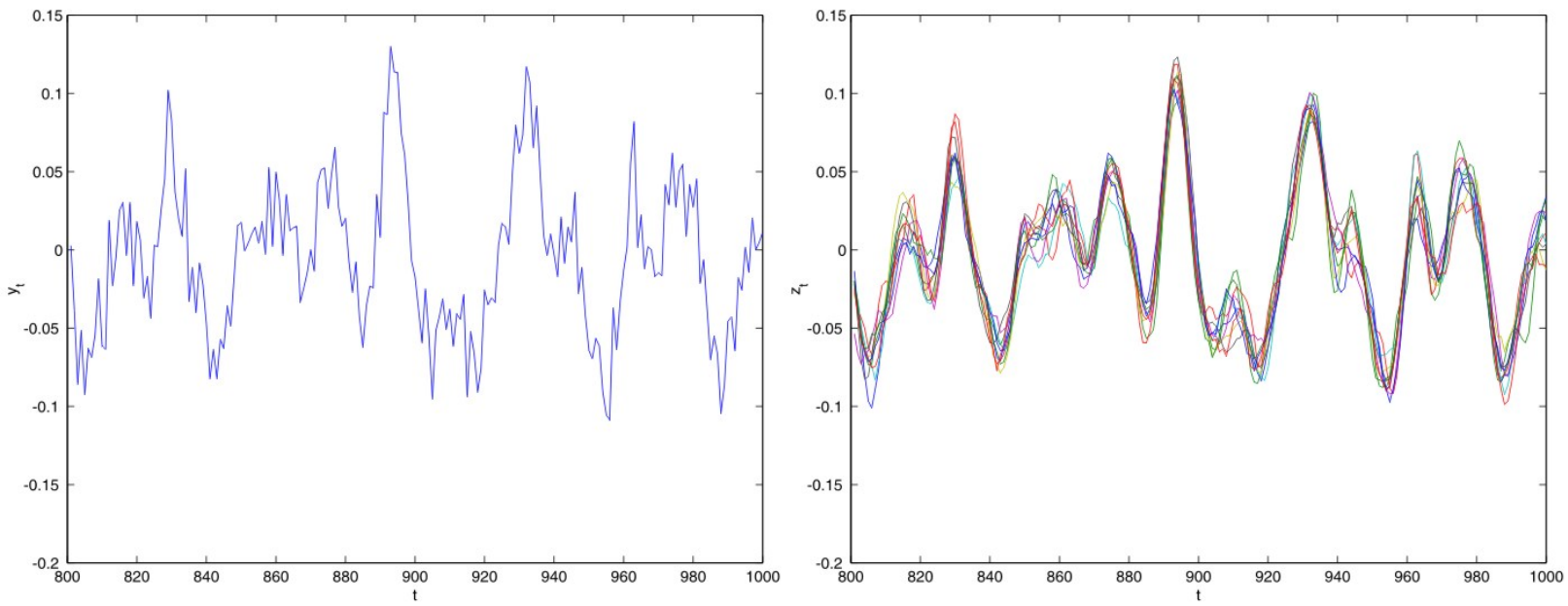


Figure 3: Noisy speech, t=801,...,1000, and smoothed realisations

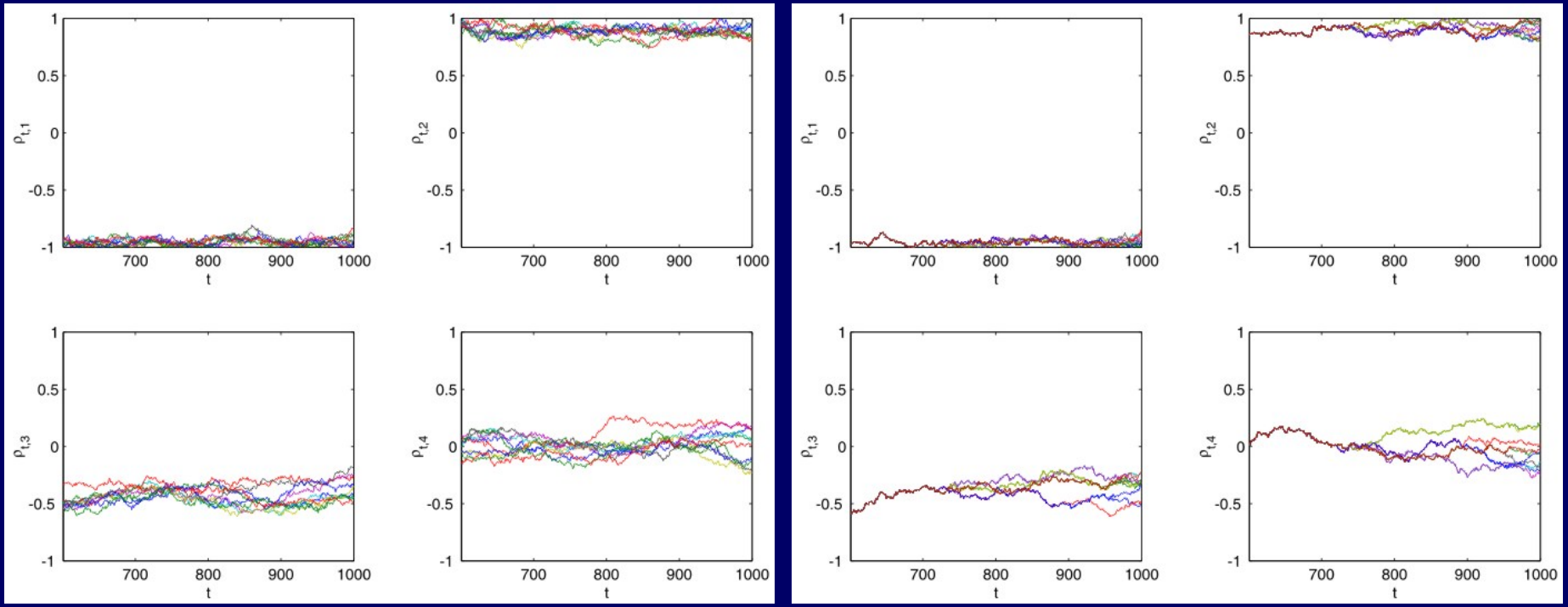


Figure 4: 10 realizations from the smoothing density for the TV-PARCOR coefficients (LHS) compared with standard trajectory-based method (RHS).

Rao-blackwellised filtering

- So far we have considered generic models without any significant constraints on their structure.

Rao-blackwellised filtering

- So far we have considered generic models without any significant constraints on their structure.
- What, though, if there is some analytic structure in the model, e.g. some component of the model is linear/Gaussian, or a Hidden Markov model (HMM)?

Rao-blackwellised filtering

- So far we have considered generic models without any significant constraints on their structure.
- What, though, if there is some analytic structure in the model, e.g. some component of the model is linear/Gaussian, or a Hidden Markov model (HMM)?
- One answer is to mix the analytic filtering results of the Kalman filter or HMM filter with the particle filter: 'Rao-Blackwellised', 'Mixture Kalman' or 'Marginalised' particle filter - see Doucet, Godsill and Andrieu (2000), Chen and Liu (2000), Schon, Gustafsson, Nordlund (2005).

Rao-blackwellised filtering

- So far we have considered generic models without any significant constraints on their structure.
- What, though, if there is some analytic structure in the model, e.g. some component of the model is linear/Gaussian, or a Hidden Markov model (HMM)?
- One answer is to mix the analytic filtering results of the Kalman filter or HMM filter with the particle filter: 'Rao-Blackwellised', 'Mixture Kalman' or 'Marginalised' particle filter - see Doucet, Godsill and Andrieu (2000), Chen and Liu (2000), Schon, Gustafsson, Nordlund (2005).
- Split the state vector into two parts x^L ('good cop') and x^N ('bad cop')

- The 'good' part of the state, x_t^L, can be written as a linear state space model:

$$x_t^L = A(x_t^N)x_{t-1}^L + u_t^L \ , \tag{27}$$

$$y_t = B(x_t^N)x_t^L + v_t^L \ . \tag{28}$$

  Here u_t^L and v_t^L are independent, zero-mean, Gaussian disturbances with covariances C_u and C_v, respectively. nonlinear state x_t^N.

- The 'good' part of the state, x_t^L, can be written as a linear state space model:

$$x_t^L = A(x_t^N)x_{t-1}^L + u_t^L \ , \tag{27}$$

$$y_t = B(x_t^N)x_t^L + v_t^L \ . \tag{28}$$

  Here u_t^L and v_t^L are independent, zero-mean, Gaussian disturbances with covariances C_u and C_v, respectively. nonlinear state x_t^N.

- Now the 'bad' nonlinear part of the state obeys a general dynamical model:

$$x_t^N \sim f(x_t^N | x_{0:t-1}^N), \ \ x_0^N \sim f(x_0^N) \ . \tag{29}$$

- The idea is now to marginalise the linear state sequence: the calculations can be done exactly and sequentially by the Kalman filter likelihood evaluation!

$$p(x_{0:t}^N|y_{0:t}) = \int p(x_{0:t}^L, x_{0:t}^N|y_{0:t})dx_{0:t}^L \ .$$

- The idea is now to marginalise the linear state sequence: the calculations can be done exactly and sequentially by the Kalman filter likelihood evaluation!

$$p(x_{0:t}^N|y_{0:t}) = \int p(x_{0:t}^L, x_{0:t}^N|y_{0:t})dx_{0:t}^L \ .$$

- Particle filtering is then run on the nonlinear state sequence only.
  - Prediction

$$p(x_{0:t}^N|y_{0:t-1}) = p(x_{0:t-1}^N|y_{0:t-1})f(x_t^N|x_{0:t-1}^N) \ .$$

  - Correction

$$p(x_{0:t}^N|y_{0:t}) = \frac{p(y_t|y_{0:t-1}, x_{0:t}^N)p(x_{0:t}^N|y_{0:t-1})}{p(y_t|y_{0:t-1})} \ ,$$

- $\{(x_{0:t}^{N,(i)}, \omega_t^{(i)})\}_{i=1,\ldots,N}$ denote the nonlinear state particles. Then, Rao-Blackwellised estimation scheme for x^L is obtained as a random Gaussian mixture approximation given by

$$p(x_t^L|y_{0:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} p(x_t^L|x_{0:t}^{N,(i)}, y_{0:t}) \ , \tag{30}$$

  where the conditional densities $p(x_t^L|x_{0:t}^{N,(i)}, y_{0:t})$ are Gaussian and computed using Kalman filtering recursions (likelihood evaluation).

- $\{(x_{0:t}^{N,(i)}, \omega_t^{(i)})\}_{i=1,\ldots,N}$ denote the nonlinear state particles. Then, Rao-Blackwellised estimation scheme for x^L is obtained as a random Gaussian mixture approximation given by

$$p(x_t^L|y_{0:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} p(x_t^L|x_{0:t}^{N,(i)}, y_{0:t}) \ , \tag{30}$$

  where the conditional densities $p(x_t^L|x_{0:t}^{N,(i)}, y_{0:t})$ are Gaussian and computed using Kalman filtering recursions (likelihood evaluation).

- Since $p(x_t^L|x_{0:t}^{N,(i)}, y_{0:t})$ depends on the particle $x_{0:t}^{N,(i)}$, the Kalman filter must be run for every particle - very expensive - performance/computation trade-off.

Example: Heavy-tailed noise modelling

Lombardi and Godsill (IEEE Tr SP 2006)

- One of the empirical features of noise in many audio sources (gramophone disks, ...) is heavy-tailed behaviour (clicks, crackles...).
- We will consider on-line estimation and signal extraction in heavy-tailed α-stable law noise models by means of sequential Monte Carlo methods.

α-Stable Distributions

- The α-stable family of distributions stems from a generalised general version of the central limit theorem in which the assumption of the finiteness of the variance is replaced by a much less restrictive one concerning the regular behaviour of the tails.
- The family is identified by means of the characteristic function

$$\phi(t) = \begin{cases} \exp\left\{i\delta t - \gamma^{\alpha}|t|^{\alpha}\left[1 - i\beta \mathrm{sgn}](t)\tan(\pi\alpha/2)\right]\right\} & \text{if } \alpha \neq 1 \\ \exp\left\{i\delta t - \gamma|t|\left[1 + i\beta\frac{2}{\pi}\mathrm{sgn}(t)\ln|t|\right]\right\} & \text{if } \alpha = 1 \end{cases} \tag{31}$$

  which depends on four parameters: $\alpha \in (0, 2]$, measuring the tail thickness (thicker tails for smaller values of the parameter), $\beta \in [-1, 1]$ determining the degree and sign of asymmetry, $\gamma > 0$ (scale) and $\delta \in \Re$ (location).

- Unfortunately, (31) can be inverted to yield a closed-form density function only for a very few cases (Gaussian, Cauchy, Levy).
- Estimation difficulties have thus hindered the use of α-stable distributions in applied work.
- Here we propose sequential Monte Carlo method for extraction of TVAR time series observed in symmetric α-stable noise ($\beta = 0$).
- By representing the symmetric stable law as a scale mixture of Gaussians, the Monte Carlo approach avoids any direct evaluations of the stable law pdf.

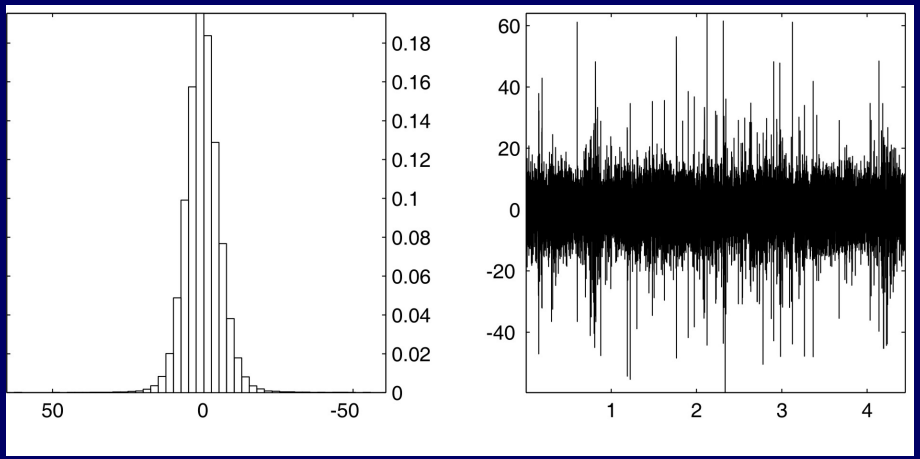


Figure 5: Waveform and histogram of noise from gramophone sound recording.

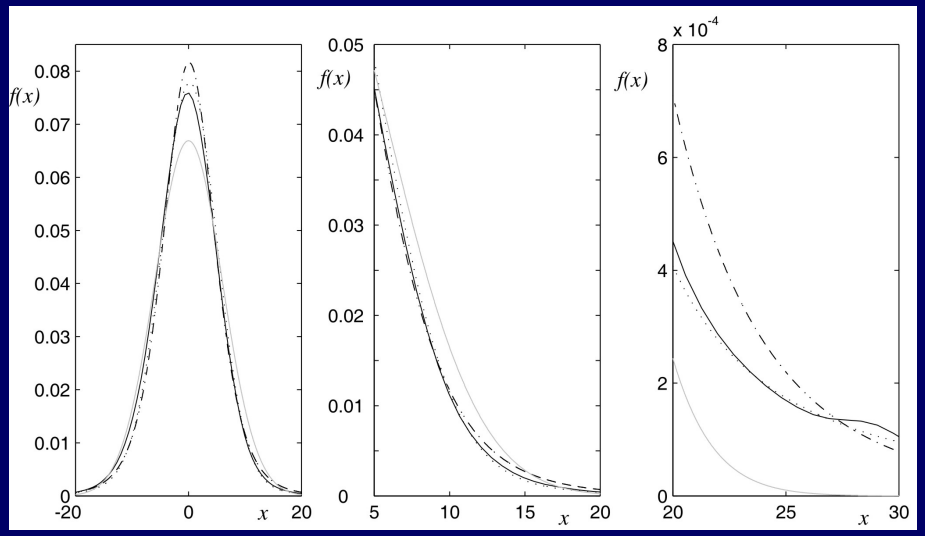


Figure 6: Kernel density (solid line), Gaussian fit (grey line), Student t fit (dashed line) and α-stable fit (dotted line).

- In the case of symmetric stable distributions we can exploit the fact that, if we have two random variables

$$X_1 \sim \mathrm{St}(\alpha_1, 0), \quad X_2 \sim \mathrm{St}(\alpha_2, 1),$$

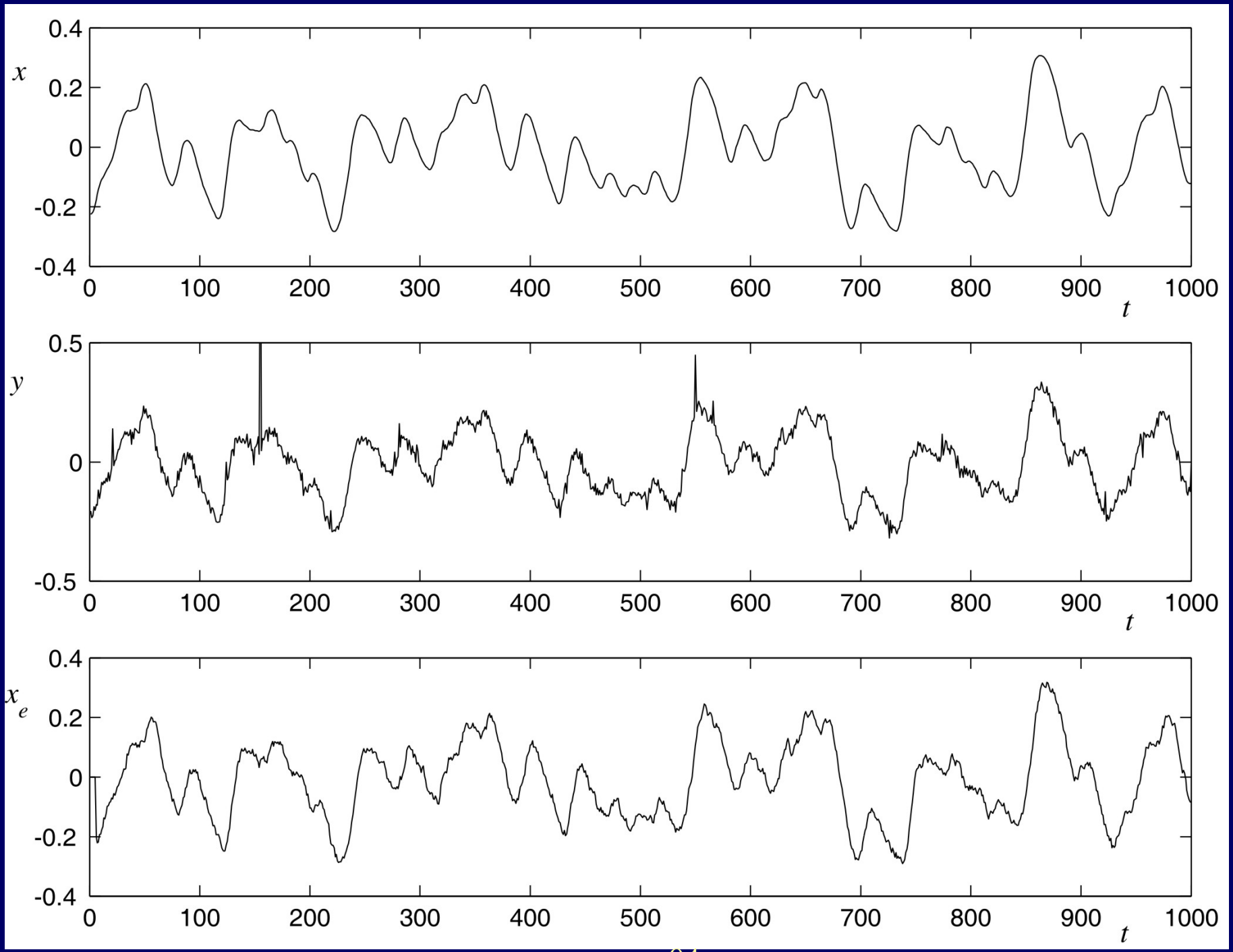
then the distribution of their product is

$$X_1 X_2 \sim \mathrm{St}(\alpha_1 \alpha_2, 0).$$

- It follows that a symmetric α-stable random variable can be thought of as the product of a Gaussian X_1 and a perfectly skewed stable random variable:

$$X_2 \sim \mathrm{St}\left(\tfrac{\alpha}{2}, 1\right).$$

- Hence a conditionally Gaussian particle filter may be implemented, allowing Rao-Blackwellisation (Kalman filter) - see Vermaak, Andrieu, Doucet and Godsill (IEEE SA, 2002) for the Gaussian noise case.


x
0.4
0.2
0
-0.2
-0.4
0
100
200
300
400
500
600
700
800
900
1000
t
y
0.5
0
-0.5
0
100
200
300
400
500
600
700
800
900
1000
t
x_e
0.4
0.2
0
-0.2
-0.4
0
100
200
300
400
500
600
700
800
900
1000
t

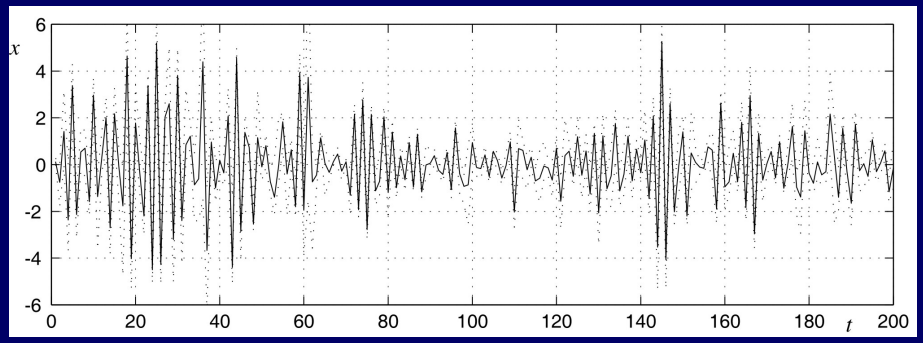


Figure 7: Filtered signal (solid line) with 95% quantile bands (dotted lines), stable noise, $\alpha = 1.4$.

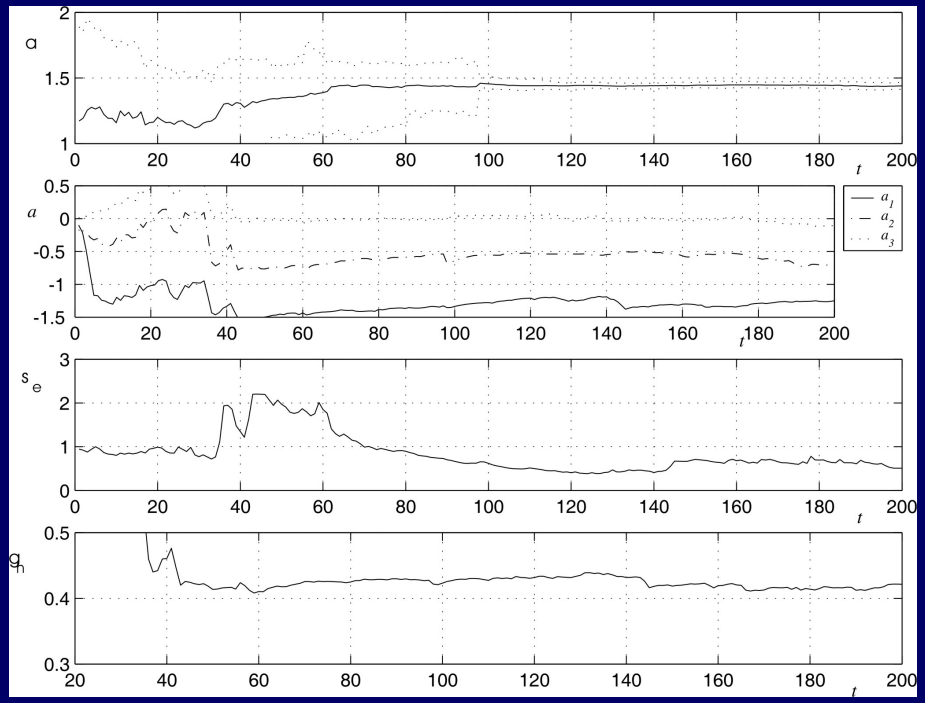


Figure 8: Estimated parameters of the model, stable noise.

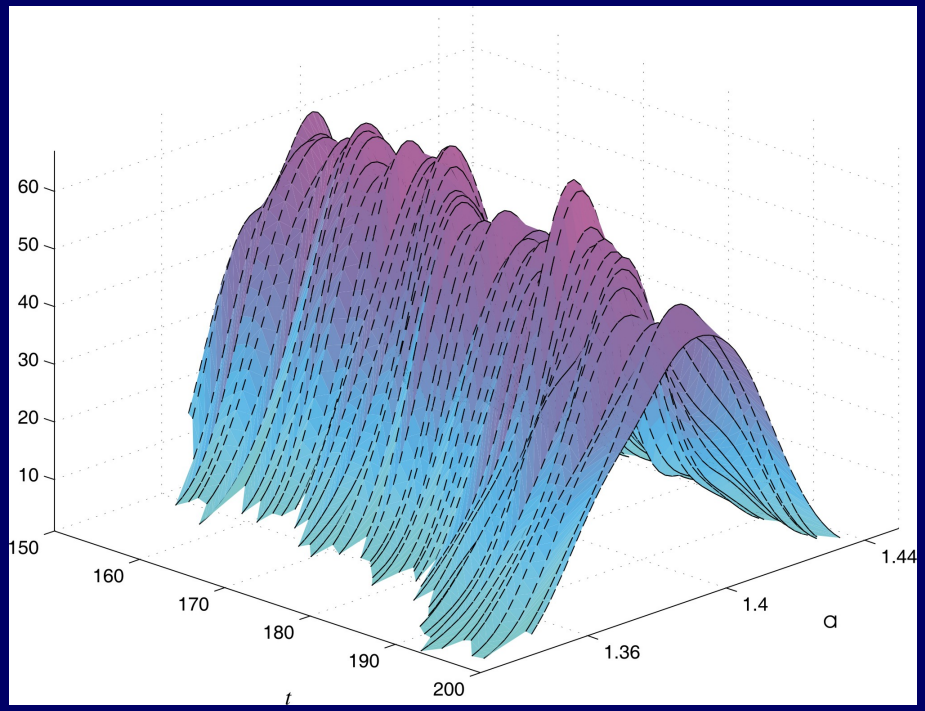


Figure 9: Kernel smoothed posterior densities of α in for $t = 150, \ldots, 200$.

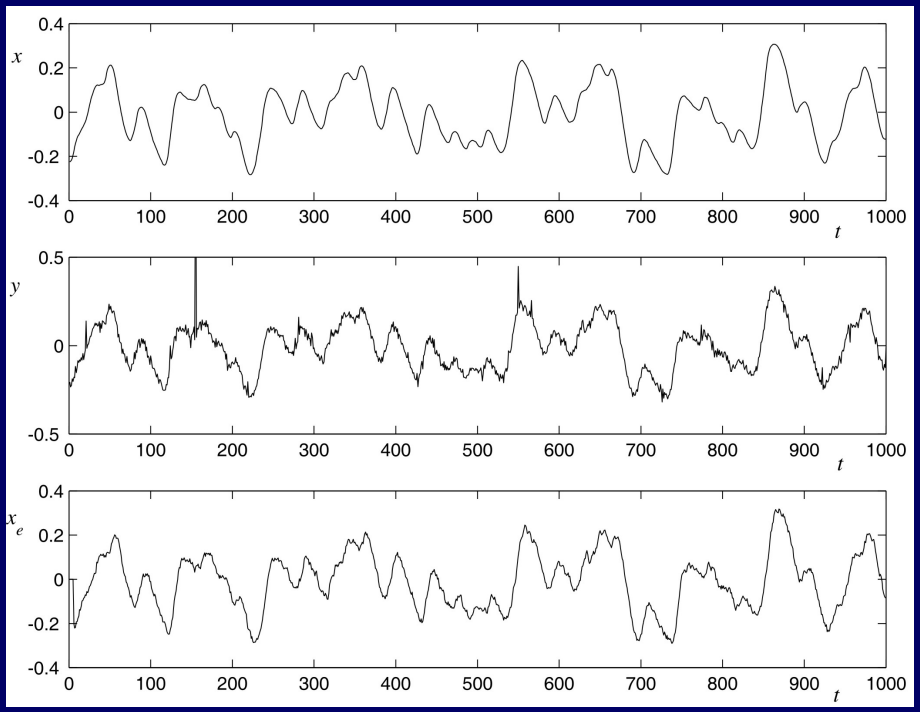


Figure 10: Excerpt of clean, noisy and reconstructed signal for Boards of Canada's "Music is Math".

Non-linear distortion correction

[Work with William Fong (2002)]

Model the observation process as a Gaussian-limited dependent process:

$$y_t = \mathcal{F}(z_t) \tag{32}$$

with $\mathcal{F}(\cdot)$ being a general distortion function.

The inverse process,

$$\mathcal{F}^{-1}(y_t) = \{z_t; \mathcal{F}(z_t) = y_t\} \tag{33}$$

is a one-to-many mapping and provides a range of possible values that the input might have taken. In terms of probability distributions,

$$p(y_t|z_t) = \delta(y_t - \mathcal{F}(z_t))$$

As before, $\{z_t\}$ is modelled as a TVAR process, this time using the standard linear random walk paramerisation of the coeffs.

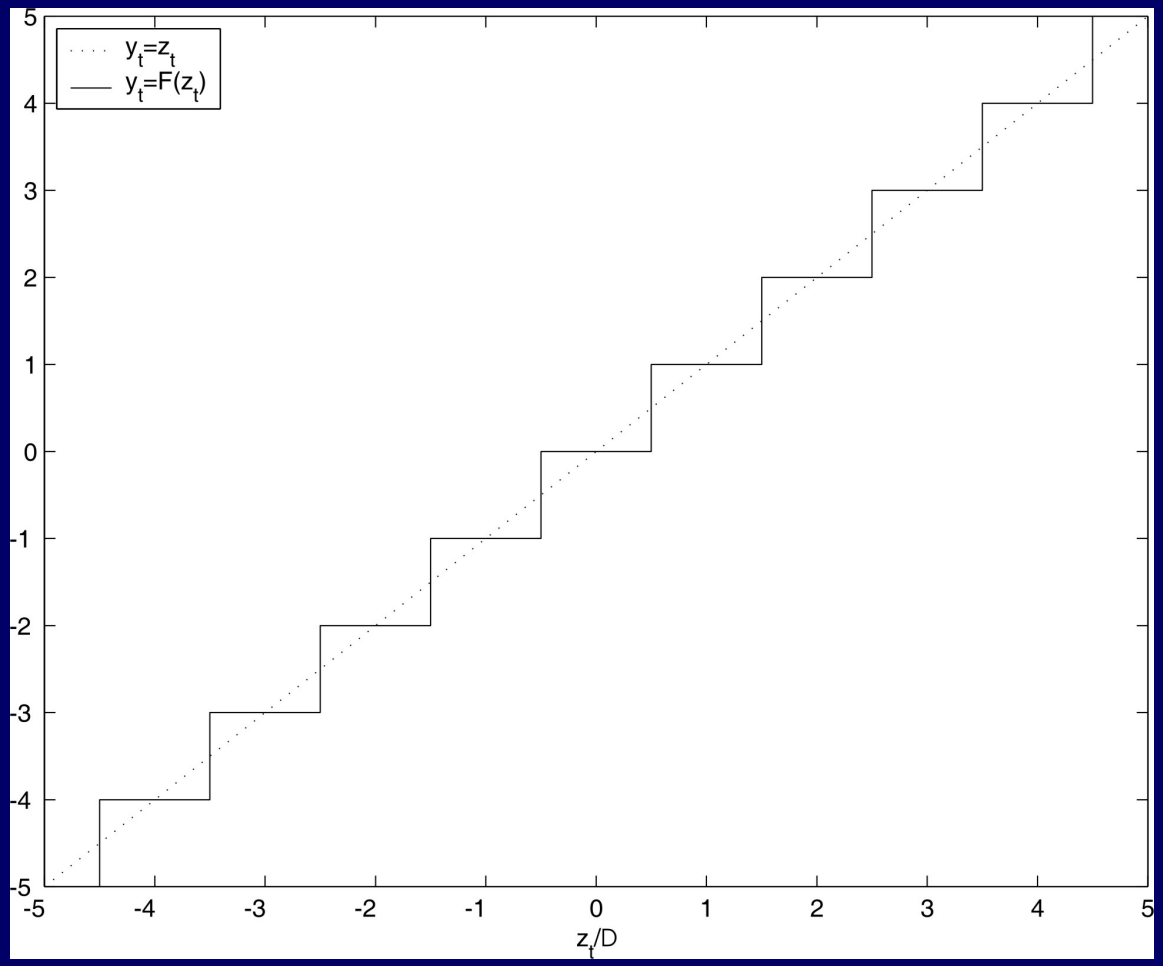


Figure 11: Many-to-one mapping for quantised signal

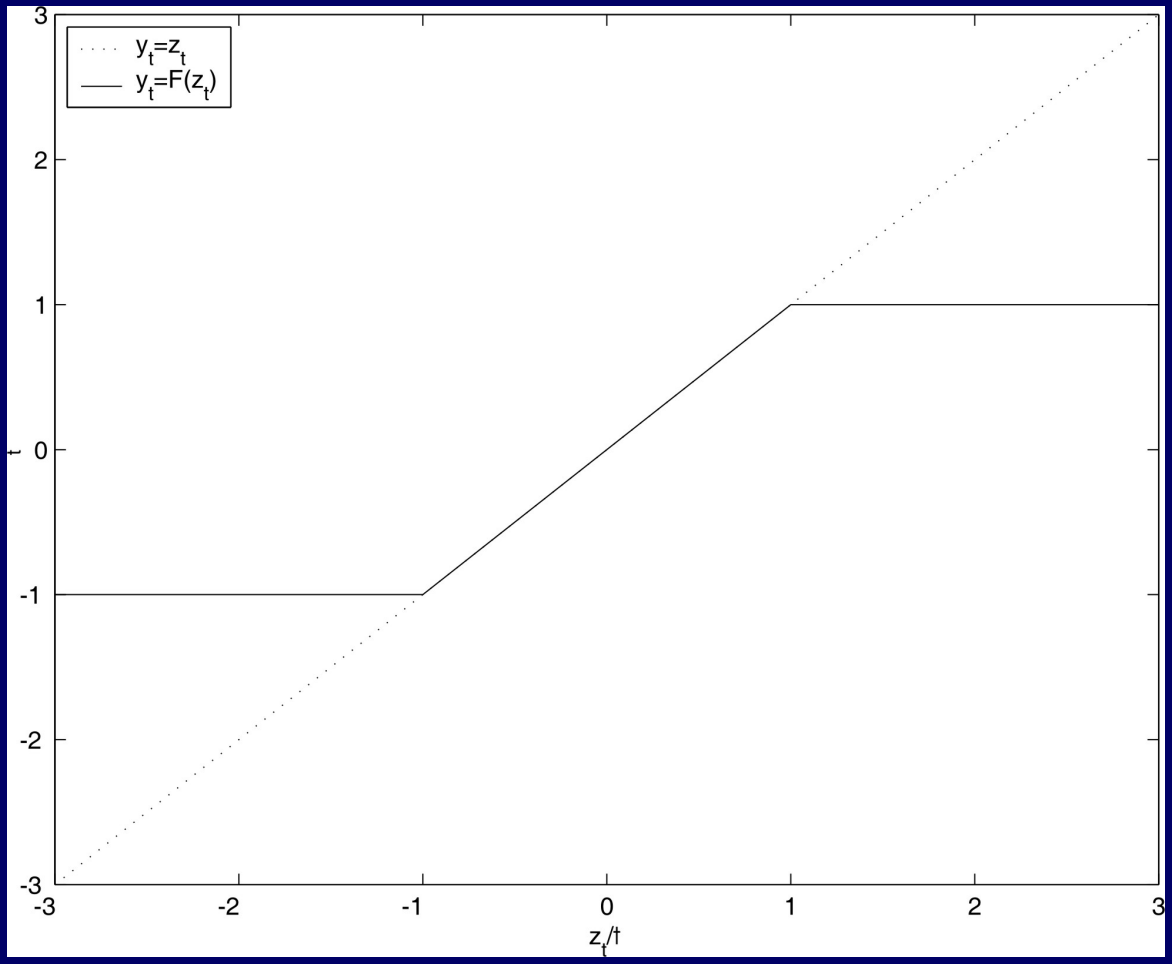


Figure 12: Many-to-one mapping for clipped signal

With these models, the TVAR parameters may be marginalised completely, i.e. we sample from:

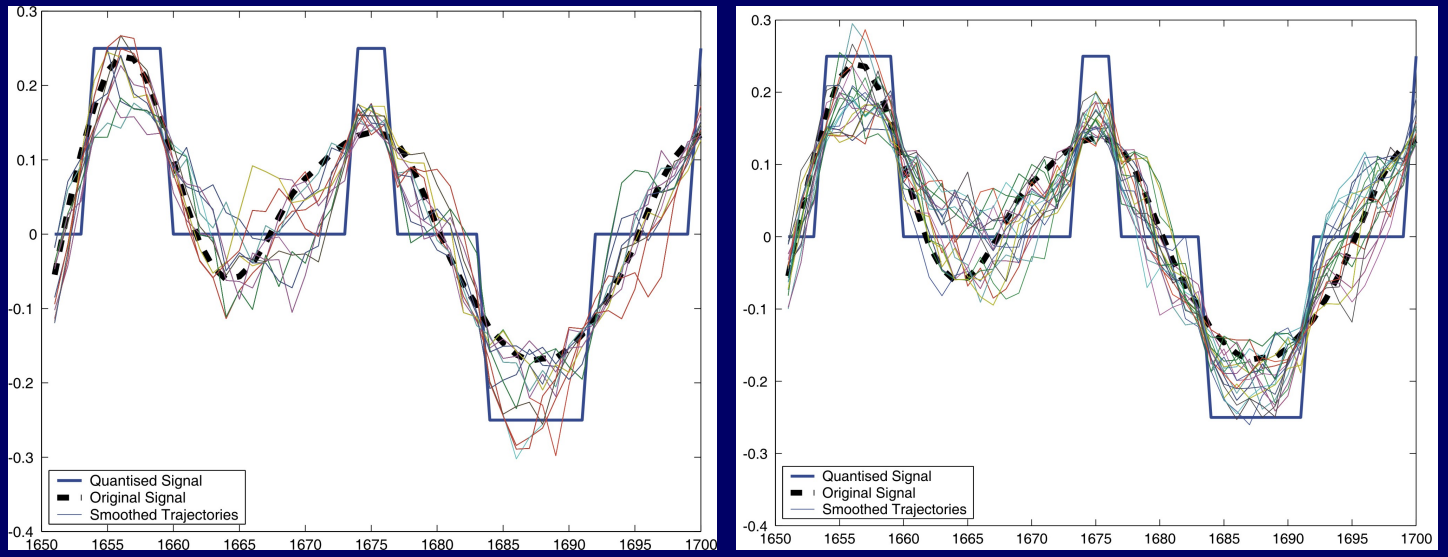
$$p(z_{1:T}|y_{1:T}) = \int p(z_{1:T}, a_{1:T}|y_{1:T})da_{1:T}$$

[In the filtering case, this was demonstrated previously by Andrieu and Doucet (2002, JRSSB)].

The required backward conditional density function is then given by

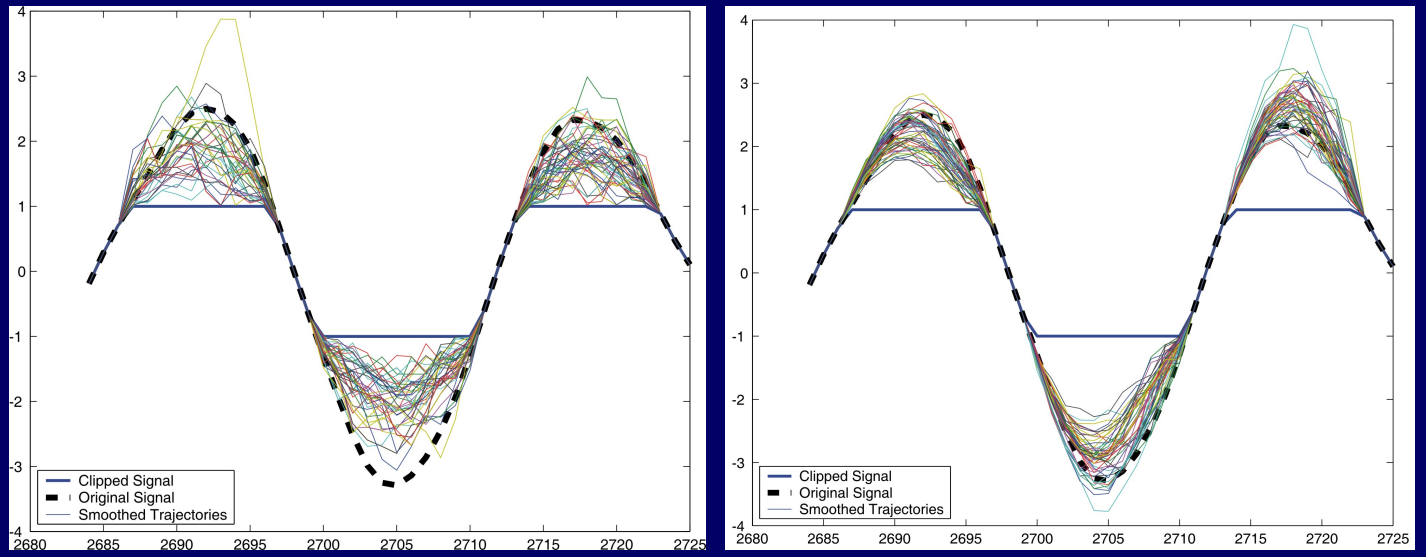
$$\begin{aligned} p(z_t|z_{t+1:T}, y_{1:T}) &\approx \int p(z_{t+1}|z_{1:t}) \sum_{i=1}^{N} w_t^{(i)} \delta(z_{1:t} - z_{1:t}^{(i)}) dz_{1:t-1} \\ &\approx \sum_{i=1}^{N} w_t^{(i)} p(z_{t+1}|z_{1:t}^{(i)}) \delta(z_t - z_{1:t}^{(i)}) \\ &\approx \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(z_t - z_t^{(i)}) \end{aligned}$$

with the modified weight $w_{t|t+1}^{(i)} \propto w_t^{(i)} p(z_{t+1}|z_{1:t}^{(i)})$. This weight is computed sequentially by sequential computation of the sufficient statistics (Kalman filter) and the prediction error decomposition.



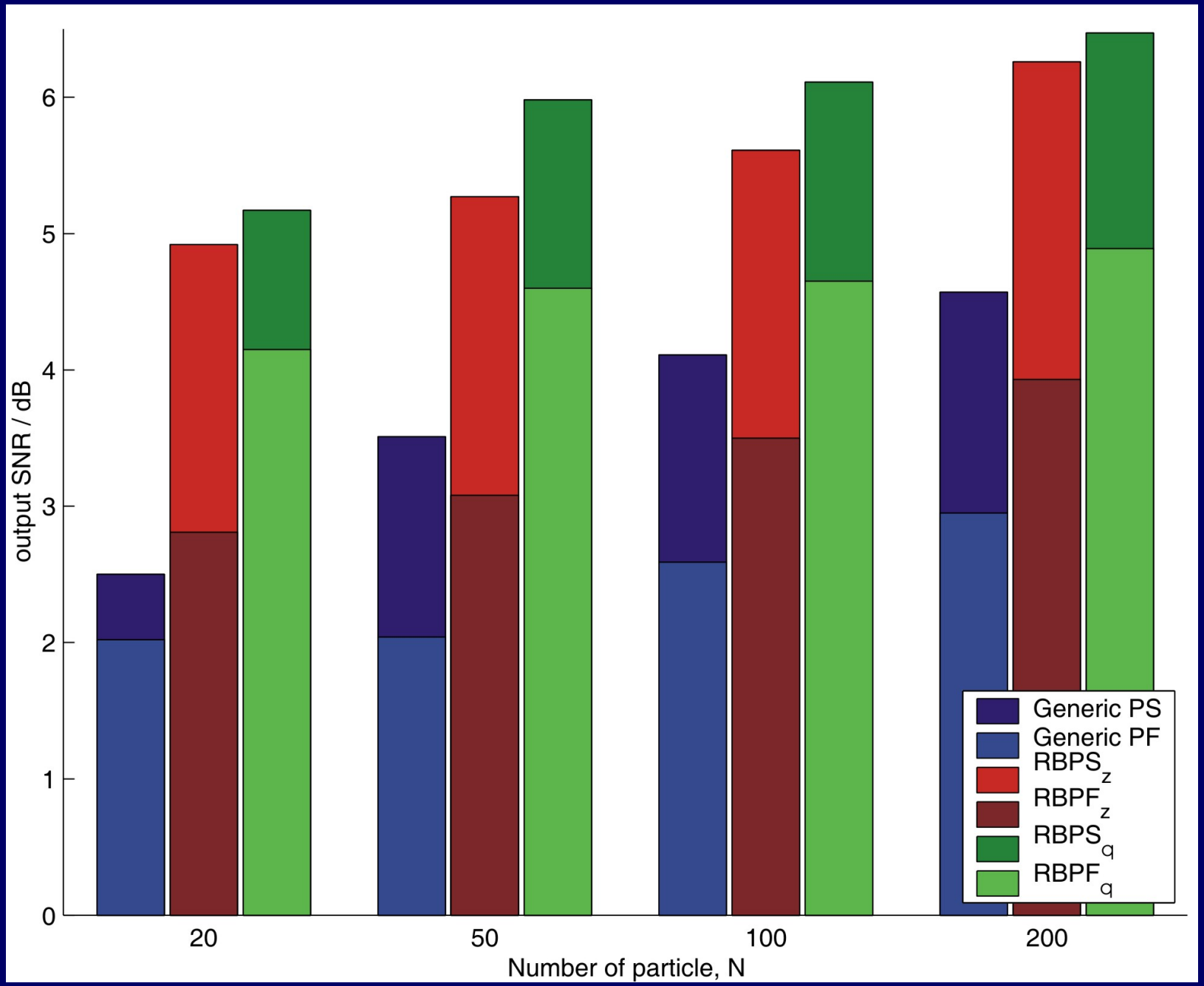
(a) Generic particle smoother (b) Rao-Blackwellised particle smoother

Figure 13: Typical frame showing the original, quantised and simulated trajectories generated using different non-linear smoothers



(a) Generic particle smoother (b) Rao-Blackwellised particle smoother

Figure 14: Typical frame showing the original, clipped and simulated trajectories generated using different non-linear smoothers


output SNR / dB
0
1
2
3
4
5
6
20
50
100
200
Number of particle, N
Generic PS
Generic PF
RBPS$_z$
RBPF$_z$
RBPS$_q$
RBPF$_q$

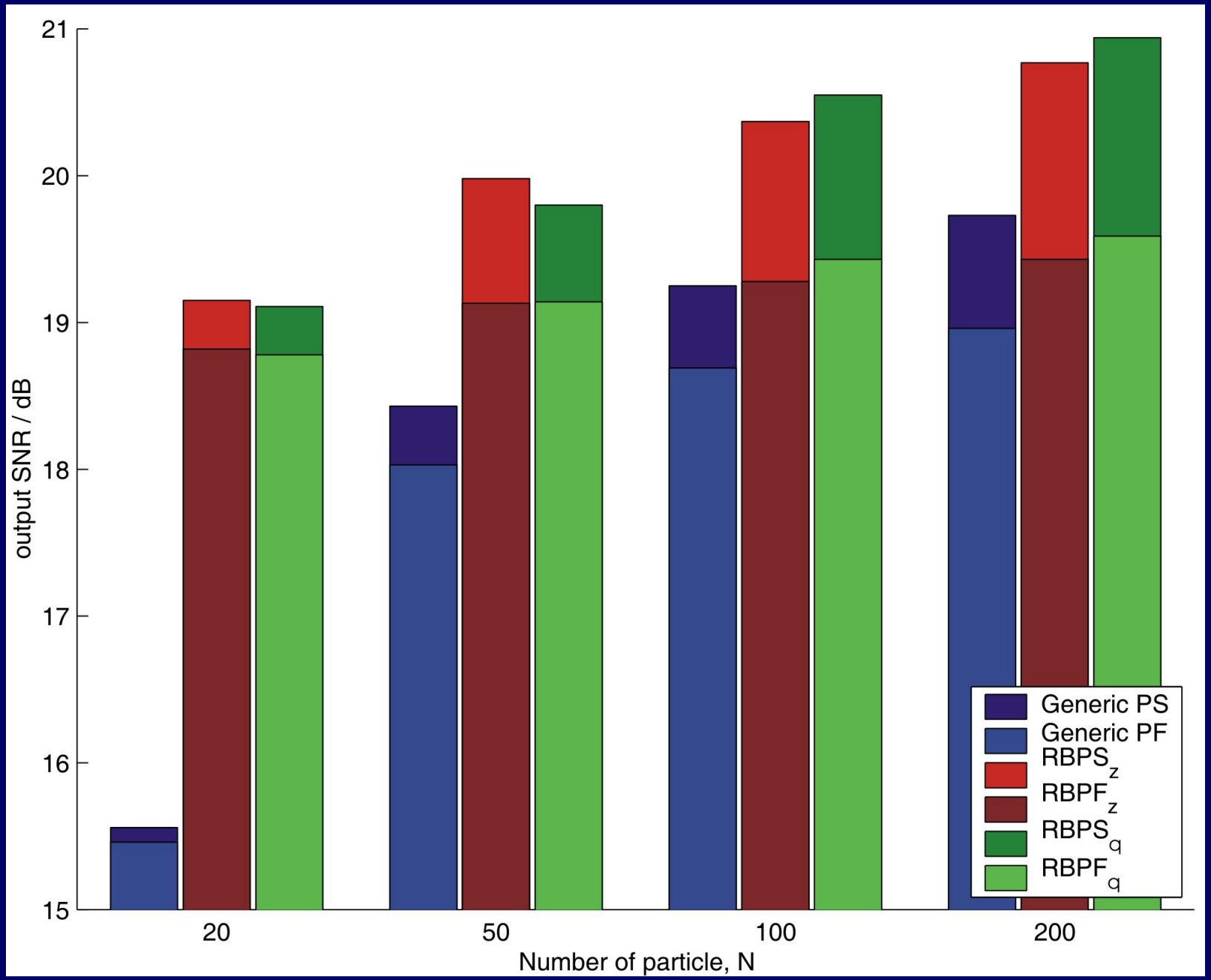


Figure 16: Average output SNR for different algorithms given a lightly

MCMC and particle filters

- A powerful combination is particle filters with MCMC
- There are two principal schemes:
- – We have, approximately, after resampling in the importance sampling filter:

$$x_{0:t}^{(i)}) \sim p(x_{0:t}|y_{0:t})$$

  – Apply an MCMC kernel with $p(x_{0:t}|y_{0:t})$ as target density, e.g. fixed-lag Gibbs-MH moves on $p(x_{t-L+1:t}|x_{0:t-L}, y_{0:t})$. Intuition is that as the MCMC converges the particle representation will 'improve'. No need to diagnose convergence of chains...

  – This is is the 'resample-move' scheme - Berzuini and Gilks (1998), see also MacEachern, Clyde and Liu (1997)

  – Generally very effective, but does not remove the importance sampling element.

- – Instead, recall that the SMC scheme approximates the filtering recursions as follows:

$$p(x_{0:t+1}|y_{0:t+1}) \approx \frac{1}{N} \frac{g(y_{t+1}|x_{t+1}) \sum_{i=1}^{N} f(x_{t+1}|x_t^{(i)}) \delta_{x_{0:t}^{(i)}}(x_{0:t})}{p(y_{t+1}|y_{1:t})}$$

or, equivalently (integrate over $x_{0:t}$):

$$p(x_{t+1}|y_{0:t+1}) \approx \frac{1}{N} \frac{g(y_{t+1}|x_{t+1}) \sum_{i=1}^{N} f(x_{t+1}|x_t^{(i)})}{p(y_{t+1}|y_{1:t})}$$

– We can run MCMC directly on either of these target distributions to replace the importance sampler altogether - this is done in Gilks et al (1997, JASA) and Khan, Balcher and Dellaert (2006, PAMI).

– Very effective schemes - can incorporater Gibbs moves, joint MH moves - all the power of MCMC.

Discussion

- The particle filter/smoother plus its adaptations, make a powerful, computationally intensive, suite of methods for inference in large datasets.
- We have covered most of the basics. Pointers to other areas, especially:
  - Parameter estimation - on-going challenge
  - Population Monte Carlo/ SMC samplers - hot topic

  can be found in our review paper:

  Cappé, Godsill and Moulines, Proc. IEEE 2007

References

[1] G. Kitagawa, "Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Statist.*, vol. 1, pp. 1–25, 1996.

[2] M. West, "Mixture models, Monte Carlo, Bayesian updating and dynamic models," *Computing Science Statistics*, vol. 24, pp. 325–333, 1993.

[3] N. Gordon, D. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F, Radar Signal Process.*, vol. 140, pp. 107–113, 1993.

[4] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[5] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice-Hall, 2000.

[6] B. Ristic, M. Arulampalam, and A. Gordon, *Beyond Kalman Filters: Particle Filters for Target Tracking*. Artech House, 2004.

[7] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer, 2005.

[8] A. Jazwinski, *Stochastic processes and filtering theory*. New York: Academic Press, 1970.

[9] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Automat. Control*, vol. 17, no. 4, pp. 439–448, 1972.

[10] R. Kulhavy, "Recursive nonlinear estimation: a geometric approach," *Automatica*, vol. 26, no. 3, pp. 545–555, 1990.

[11] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[12] R. Van der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Adv. Neural Inf. Process. Syst.*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2000, vol. 13.

[13] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Automat. Control*, vol. 45, pp. 910–927, 2000.

[14] J. Handschin and D. Mayne, "Monte Carlo techniques to estimate the conditionnal expectation in multi-stage non-linear filtering," in *Int. J. Control*, vol. 9, 1969, pp. 547–559.

[15] J. Handschin, "Monte Carlo techniques for prediction and filtering of non-linear stochastic processes," *Automatica*, vol. 6, pp. 555–563, 1970.

[16] D. B. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm (discussion of Tanner and Wong)," *J. Am. Statist. Assoc.*, vol. 82, pp. 543–546, 1987.

[17] A. Blake and M. Isard, *Active Contours*. Springer, 1998.

[18] J. Liu and R. Chen, "Blind deconvolution via sequential imputations," *J. Roy. Statist. Soc. Ser. B*, vol. 430, pp. 567–576, 1995.

[19] P. Del Moral, "Nonlinear filtering: interacting particle solution," *Markov Process. Related Fields*, vol. 2, pp. 555–579, 1996.

[20] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[21] J. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2001.

[22] J. Liu and R. Chen, "Sequential Monte-Carlo methods for dynamic systems," *J. Roy. Statist. Soc. Ser. B*, vol. 93, pp. 1032–1044, 1998.

[23] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte-Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, pp. 197–208, 2000.

[24] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 241–254, 2002.

[25] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, 2003.

[26] H. Tanizaki, "Nonlinear and non-Gaussian state-space modeling with Monte-Carlo techniques: a survey and comparative study," in *Handbook of Statistics 21. Stochastic processes: Modelling and Simulation*, D. N. Shanbhag and C. R. Rao, Eds. Elsevier, 2003, pp. 871–929.

[27] C. Andrieu, A. Doucet, and C. P. Robert, "Computational advances for and from Bayesian analysis," *Statist. Sci.*, vol. 19, no. 1, pp. 118–127, 2004.

[28] J. Geweke, "Bayesian inference in econometric models using Monte-Carlo integration," *Econometrica*, vol. 57, no. 6, pp. 1317–1339, 1989.

[29] D. B. Rubin, "Using the SIR algorithm to simulate posterior distribution," in *Bayesian Statistics 3*, J. M. Bernardo, M. DeGroot, D. Lindley, and A. Smith, Eds. Clarendon Press, 1988, pp. 395–402.

[30] J. Carpenter, P. Clifford, and P. Fearnhead, "An improved particle filter for non-linear problems," *IEE Proc., Radar Sonar Navigation*, vol. 146, pp. 2–7, 1999.

[31] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *J. Roy. Statist. Soc. Ser. B*, vol. 65, pp. 887–899, 2003.

[32] H. R. Künsch, "Recursive Monte-Carlo filters: algorithms and theoretical analysis," *Ann. Statist.*, vol. 33, no. 5, pp. 1983–2021, 2005.

[33] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering," in *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, September 2005, arXiv: cs.CE/0507025.

[34] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Automat. Control*, vol. 9, no. 4, pp. 333–339, 1964.

[35] P. Del Moral and J. Jacod, "Interacting particle filtering with discrete-time observations: Asymptotic behaviour in the Gaussian case," in *Stochastics in Finite and Infinite Dimensions: In Honor of Gopinath Kallianpur*, T. Hida, R. L. Karandikar, H. Kunita, B. S. Rajput, S. Watanabe, and J. Xiong, Eds. Boston, MA: Birkhäuser, 2001, pp. 101–122.

[36] A. Kong, J. S. Liu, and W. Wong, "Sequential imputation and Bayesian missing data problems," *J. Am. Statist. Assoc.*, vol. 89, no. 278-288, pp. 590–599, 1994.

[37] V. Zaritskii, V. Svetnik, and L. Shimelevich, "Monte-Carlo techniques in problems of optimal data processing," *Autom. Remote Control*, vol. 12, pp. 2015–2022, 1975.

[38] H. Akashi and H. Kumamoto, "Random sampling approach to state estimation in switching environment," *Automatica*, vol. 13, pp. 429–434, 1977.

[39] R. Chen and J. S. Liu, "Mixture Kalman filter," *J. Roy. Statist. Soc. Ser. B*, vol. 62, no. 3, pp. 493–508, 2000.

[40] N. Shephard and M. Pitt, "Likelihood analysis of non-Gaussian measurement time series," *Biometrika*, vol. 84, no. 3, pp. 653–667, 1997, erratum in volume 91, 249–250, 2004.

[41] P. Del Moral, "Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems," *Ann. Appl. Probab.*, vol. 8, pp. 69–95, 1998.

[42] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, 2002.

[43] N. Chopin, "Central limit theorem for sequential monte carlo methods and its application to bayesian inference," *Ann. Statist.*, vol. 32, no. 6, pp. 2385–2411, 2004.

[44] P. Del Moral, *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

[45] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Am. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

[46] J. L. Zhang and J. S. Liu, "A new sequential importance sampling method and its application to the two-dimensional hydrophobic-hydrophilic model," *J. Chem. Physics*, vol. 117, no. 7, 2002.

[47] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, 2003.

[48] S. Godsill and T. Clapp, "Improvement strategies for monte carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[49] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *J. Roy. Statist. Soc. Ser. B*, vol. 64, no. 4, pp. 827–836, 2002.

[50] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, 2005.

[51] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, 2005.

[52] R. E. Kalman and R. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng., Trans. ASME, Series D*, vol. 83, no. 3, pp. 95–108, 1961.

[53] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer, 2004.

[54] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[55] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.

[56] W. R. Gilks and C. Berzuini, "Following a moving target—Monte Carlo inference for dynamic Bayesian models," *J. Roy. Statist. Soc. Ser. B*, vol. 63, no. 1, pp. 127–146, 2001.

[57] C. Berzuini and W. R. Gilks, "Resample-move filtering with cross-model jumps," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[58] S. N. MacEachern, M. Clyde, and J. Liu, "Sequential importance sampling for nonparametric bayes models: The next generation," *Can. J. Statist.*, vol. 27, pp. 251–267, 1999.

[59] P. Fearnhead, "Markov chain Monte Carlo, sufficient statistics and particle filter," *J. Comput. Graph. Statist.*, vol. 11, no. 4, pp. 848–862, 2002.

[60] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1918, 2005.

[61] T. C. Clapp and S. J. Godsill, "Fixed-lag smoothing using sequential importance sampling," in *Bayesian Statistics VI*, J. Bernardo, J. Berger, A. Dawid, and A. Smith, Eds. Oxford University Press, 1999, pp. 743–752.

[62] L. Tierney, "Markov chains for exploring posterior disiributions (with discussion)," *Ann. Statist.*, vol. 22, no. 4, pp. 1701–1762, 1994.

[63] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Automat. Control*, vol. 45, no. 2, pp. 188–202, 2000.

[64] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, pp. 613–624, 2001.

[65] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics Series. Chapman & Hall, 1996.

[66] S. J. Godsill, A. Doucet, and M. West, "Monte carlo smoothing for non-linear time series," *J. Am. Statist. Assoc.*, vol. 50, pp. 438–449, 2004.

[67] H. R. Künsch, "State space and hidden markov models." in *Complex Stochastic Systems*, O. E. Barndorff-Nielsen, D. R. Cox, and C. Kluepelberg, Eds. Boca raton: CRC Publisher, 2001, pp. 109–173.

[68] W. Fong, S. Godsill, A. Doucet, and M. West, "Monte carlo smoothing with application to audio signal enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 438–449, 2002.

[69] G. Kitagawa, "Non-Gaussian state space modeling of nonstationary time series," *J. Am. Statist. Assoc.*, vol. 82, no. 400, pp. 1023–1063, 1987.

[70] M. Hürzeler and H. R. Künsch, "Monte Carlo approximations for general state-space models," *J. Comput. Graph. Statist.*, vol. 7, pp. 175–193, 1998.

[71] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *Int. J Compuer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[72] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," University of Cambridge, Department of Engineering, Tech. Rep. TR-CUED-F-INFENG 498, 2004.

[73] M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If i had a million particles," in *23rd Int. Conf. Machine Learning (ICML)*, Pittsburgh, Pennsylvania, June 25-29 2006.

[74] H. Tanizaki, "Nonlinear and non-Gaussian state space modeling using sampling techniques," *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 63–81, 2001.

[75] S. J. Godsill, A. Doucet, and M. West, "Maximum *a posteriori* sequence estimation using Monte Carlo particle filters," *Ann. Inst. Stat. Math.*, vol. 53, no. 1, pp. 82–96, Mar. 2001.

[76] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[77] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, pp. 539–552, 2002.

[78] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *J. Roy. Statist. Soc. Ser. B*, vol. 68, no. 3, p. 411, 2006.

[79] J. Olsson and T. Rydén, "Asymptotic properties of the bootstrap particle filter maximum likelihood estimator for state space models," Lund University, Tech. Rep. LUTFMS-5052-2005, 2005.

[80] R. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the em algorithm," *J. Time Ser. Anal.*, vol. 3, no. 4, pp. 253–264, 1982.

[81] F. Campillo and F. Le Gland, "MLE for patially observed diffusions: Direct maximization vs. the EM algorithm," *Stoch. Proc. App.*, vol. 33, pp. 245–274, 1989.

[82] M. Segal and E. Weinstein, "A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems," *IEEE Trans. Inform. Theory*, vol. 35, pp. 682–687, 1989.

[83] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification, and control," *IEEE Proc.*, vol. 92, no. 3, pp. 423–438, 2004.

[84] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38 (with discussion), 1977.

[85] O. Zeitouni and A. Dembo, "Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, Jul. 1988.

[86] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. New York: Springer, 1995.

[87] O. Cappé, "Recursive computation of smoothed functionals of hidden Markovian processes using a particle approximation," *Monte Carlo Methods Appl.*, vol. 7, no. 1–2, pp. 81–92, 2001.

[88] F. Cérou, F. Le Gland, and N. Newton, "Stochastic particle methods for linear tangent filtering equations," in *Optimal Control and PDE's - Innovations and Applications, in Honor of Alain Bensoussan's 60th Anniversary*, J.-L. Menaldi, E. Rofman, and A. Sulem, Eds. Amsterdam: IOS Press, 2001, pp. 231–240.

[89] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Ann. Inst. Statist. Math.*, vol. 55, no. 2, pp. 409–422, 2003.

[90] J. Fichou, F. Le Gland, and L. Mevel, "Particle based methods for parameter estimation and tracking: Numerical experiments," INRIA, Tech. Rep. PI-1604, 2004.

[91] G. Poyiadjis, A. Doucet, and S. S. Singh, "Particle methods for optimal filter derivative: application to parameter estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 18-23 March 2005, pp. v/925–v/928.

[92] C. Andrieu, A. Doucet, and V. B. Tadic, "Online simulation-based methods for parameter estimation in non linear non gaussian state-space models," in *Proc. IEEE Conf. Decis. Control*, 2005.

[93] O. Cappé and E. Moulines, "On the use of particle filtering for maximum likelihood parameter estimation," in *European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, September 2005.

[94] J. Olsson, O. Cappé, R. Douc, and E. Moulines, "Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models," Lund University, Tech. Rep., 2006, arXiv:math.ST/0609514.

[95] T. Rydén, "Consistent and asymptotically normal parameter estimates for hidden Markov models," *Ann. Statist.*, vol. 22, no. 4, pp. 1884–1895, 1994.

[96] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

[97] G. Kitagawa, "A self-organizing state-space model," *J. Am. Statist. Assoc.*, vol. 93, no. 443, pp. 1203–1215, 1998.

[98] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, N. D. F. A. Doucet and N. Gordon, Eds. Springer, 2001.

[99] P. Stavropoulos and D. M. Titterington, "Improved particle filters and smoothing," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[100] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularized particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[101] R. M. Neal, "Annealed importance sampling," *Stat. Comput.*, vol. 11, no. 2, pp. 125–139, 2001.

[102] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. New York: Chapman, 1995.

[103] G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Trans. Signal Process.*, pp. 281–289, 2002.

[104] J. Stroud, N. Polson, and P. Müller, "Practical filtering for stochastic volatility models," in *State Space and Unobserved Component Models*, A. Harvey, S. J. Koopman, and N. Shephard, Eds. Cambridge University Press, 2003.

[105] A. Papavasiliou, "A uniformly convergent adaptive particle filter," *J. Appl. Probab.*, vol. 42, no. 4, pp. 1053–1068, 2005.

[106] Y. Iba, "Population-based Monte Carlo algorithms," *Trans. Japanese Soc. Artificial Intell.*, vol. 16, no. 2, pp. 279–286, 2000.

[107] ——, "Extended ensemble monte carlo," *nt. J. Mod. Phys. C*, vol. 12, no. 5, pp. 623–656, 2001.

[108] G. Ridgeway and D. Madigan, "A sequential Monte Carlo method for Bayesian analysis of massive datasets," *Data Mining and Knowledge Discovery*, vol. 7, no. 3, pp. 301–319, 2003.

[109] S. Balakrishnan and D. Madigan, "A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets," *Bayesian Analysis*, vol. 1, no. 2, pp. 345–362, 2006.

[110] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert, "Population Monte Carlo," *J. Comput. Graph. Statist.*, vol. 13, no. 4, pp. 907–929, 2004.

[111] H. Haario, E. Saksman, and J. Tamminen, "Adaptive proposal distribution for random walk Metropolis algorithm," *Computational Statistics*, vol. 14, pp. 375–395, 1999.

[112] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, "Convergence of adaptive mixtures of importance sampling schemes," *Ann. Statist.*, vol. 35, no. 1, 2007, to appear.

[113] ——, "Minimum variance importance sampling via population monte carlo," CEREMADE, Tech. Rep., 2005.

[114] P. J. V. Laarhoven and E. H. L. Arts, *Simulated Annealing: Theory and Applications*. Reidel Publisher, 1987.

[115] P. Del Moral and J. Garnier, "Genealogical particle analysis of rare events," *Ann. Appl. Probab.*, vol. 15, no. 4, pp. 2496–2534, 2005.

[116] F. Cerou, P. Del Moral, F. Le Gland, and P. Lezaud, "Limit theorems for multilevel splitting algorithms in the simulation of rare events," in *Proceedings of the 37th Winter Simulation Conference*, Orlando, Florida, 2005.

[117] R. M. Neal, "Markov chain Monte Carlo methods based on 'slicing' the density function," University of Toronto, Tech. Rep., 1997.

[118] A. Jasra, D. A. Stephens, and C. C. Holmes, "On population-based simulation for static inference," Department of Mathematics, Imperial College, Tech. Rep., 2005.

[119] A. Johansen, A. Doucet, and M. Davy, "Maximum likelihood parameter estimation for latent variable models using sequential Monte Carlo," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006.

[120] A. M. Johansen, P. Del Moral, and A. Doucet, "Sequential Monte Carlo samplers for rare events," in *Proceedings of the 6th International Workshop on Rare Event Simulation*, Bamberg, Germany, Oct. 2006.

[121] J. Vermaak, S. J. Godsill, and A. Doucet, "Sequential Bayesian kernel regression," in *Adv. Neural Inf. Process. Syst.* MIT Press, 2003.

[122] B.-N. Vo, S. S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Aerospace and Electronic Systems*, 2007, to appear.

[123] A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead, "Exact and efficient likelihood-based estimation for discretely observed diffusions processes," *J. Roy. Statist. Soc. Ser. B*, vol. 68, no. 2, pp. 1–29, 2006.