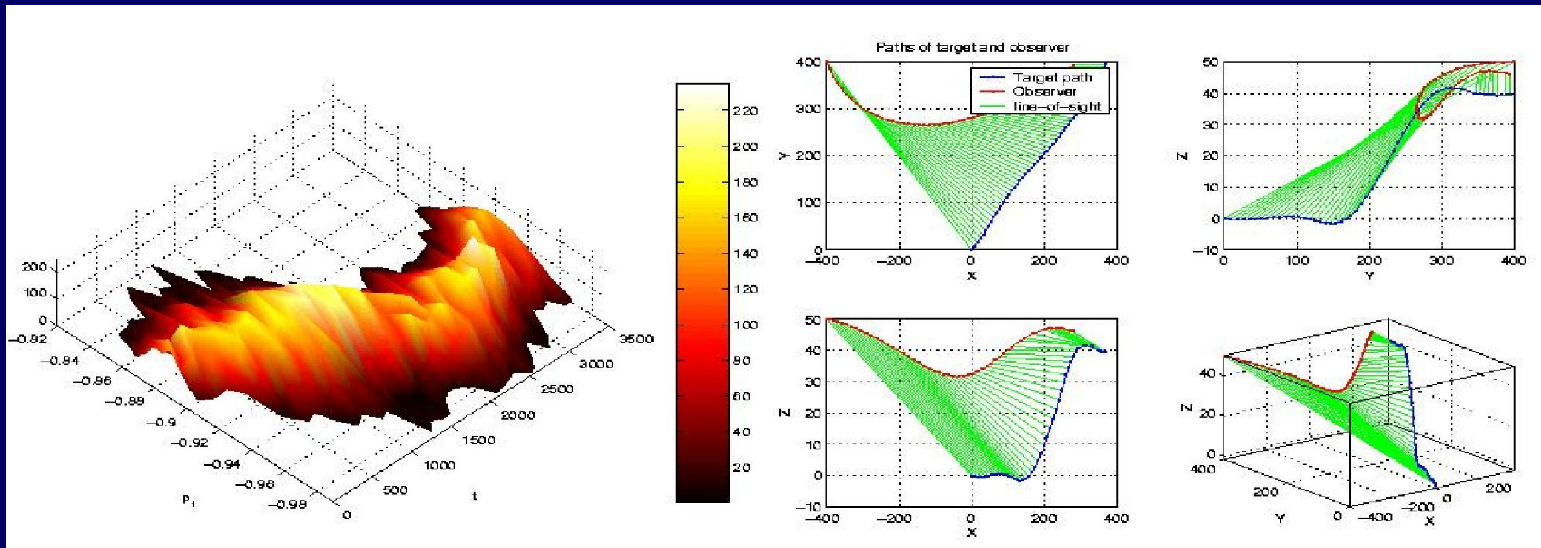
Bayesian Computer intensive methods for statistical signal processing

Simon Godsill

September, 2003



University of Cambridge

Acknowledgements

This talk includes collaborative work with:

William Fong, Jaco Vermaak, Arnaud Doucet, Mike West (Duke University), Tai Lam, ...

Thanks also to all those at the Signal Processing Laboratory in Cambridge for inspiration and discussion.

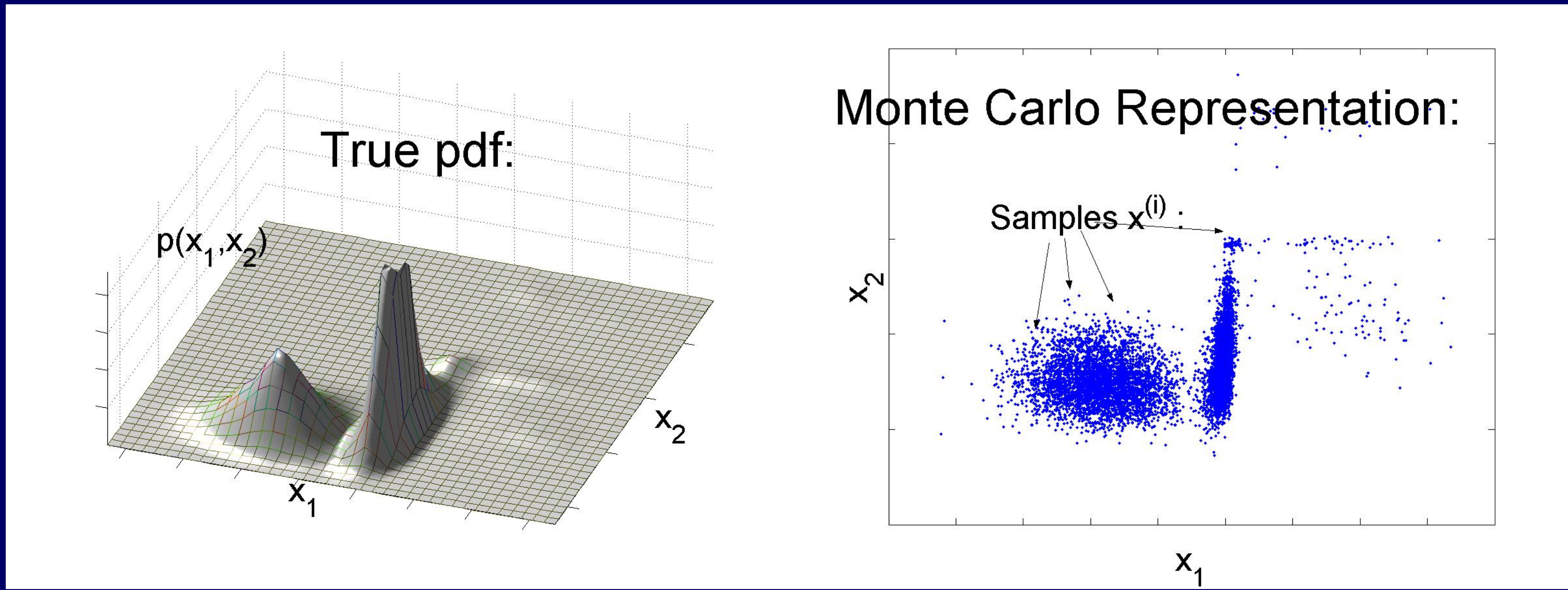
Overview

- As tasks in signal processing inference become more complex and subtle, it becomes appropriate to adopt compute-intensive methodologies
- Consider here Monte Carlo methods for inference in (principally) Bayesian probabilistic settings
- In particular I will describe Sequential Monte Carlo methods, or particle filters for non-linear, non-Gaussian settings.
- These find application in numerous sequential settings: tracking, computer vision, speech and audio, robotics, financial time series, ...

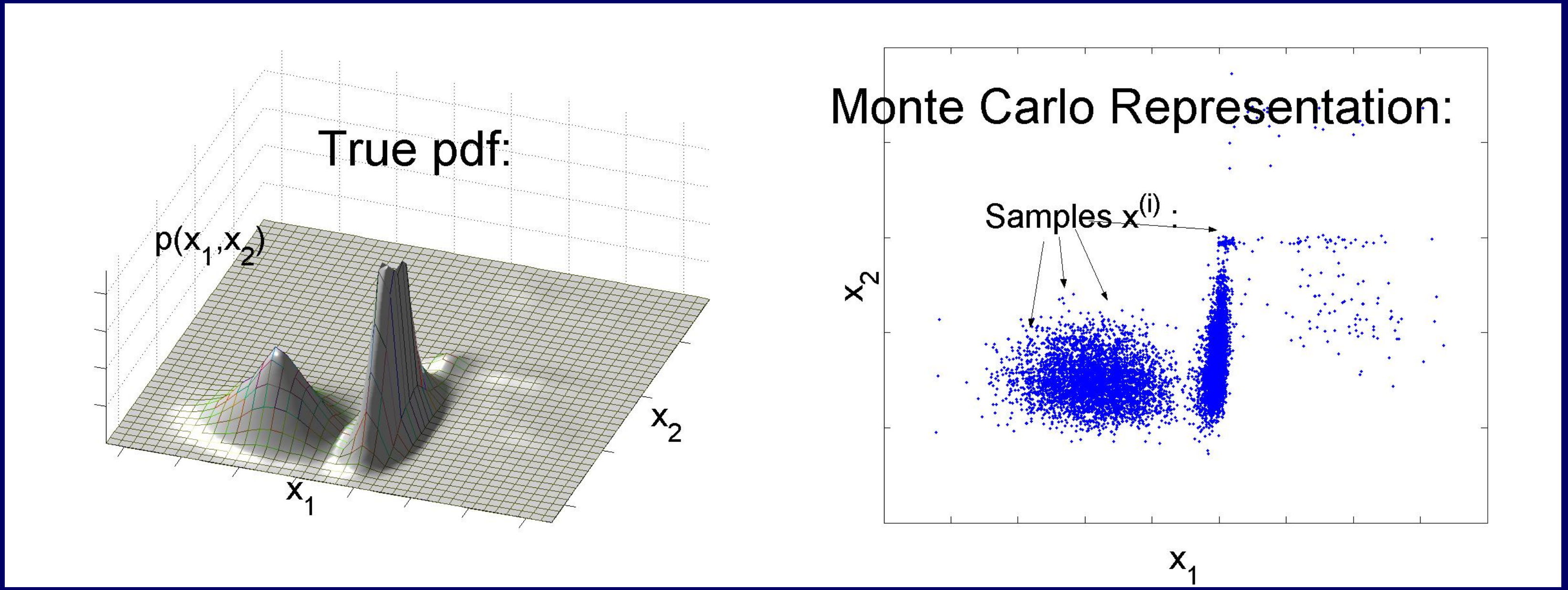
Contents

- Monte Carlo inference
- State space models, filtering and smoothing
- Particle filtering
- Research topics and applications:
  - Particle smoothing
  - Multirate (trans-dimensional) particle filters
- Discussion

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:

$x^{(i)}$, $i = 1, ..., N$, 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:

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

We can think of the Monte Carlo representation informally as:

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

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:

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

We can think of the Monte Carlo representation informally as:

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

Then the Monte Carlo expectation falls out easily as:

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

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:

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

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

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:

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

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

$$p(x) \approx \sum_{i=1}^{N}w^{(i)}\delta(x - x^{(i)}), \;\; \sum_{i=1}^{N}w^{(i)} = 1$$

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.

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, ...

State space models, filtering and smoothing

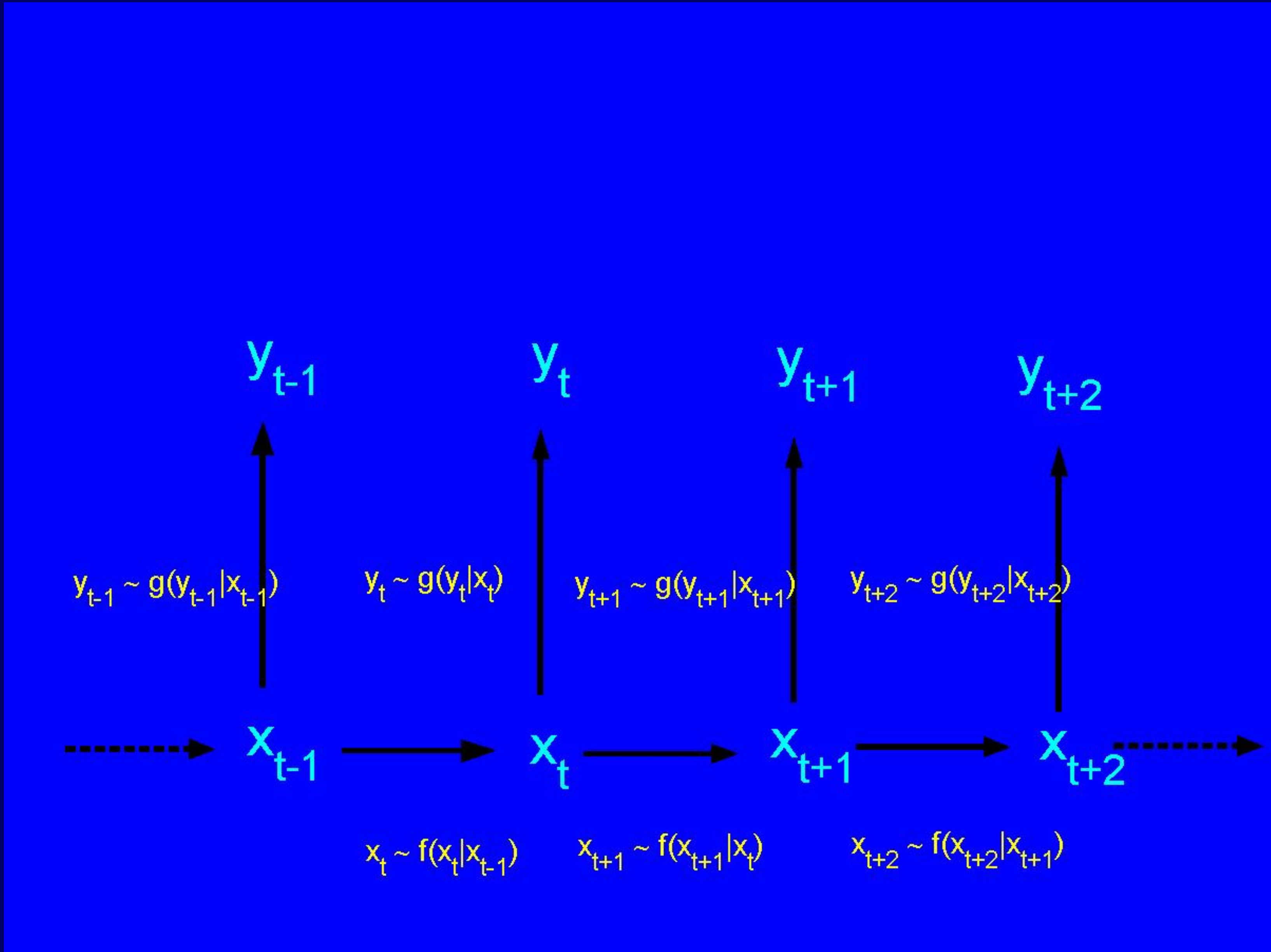
We will focus here on a broad and general class of models. Examples include:

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

Summarise the statistics as a probabilistic 'state space' or 'dynamical' model with unknown states x_t and observations y_t:

$$x_{t+1} \sim f(x_{t+1}|x_t) \qquad \text{State evolution density}$$

$$y_{t+1} \sim g(y_{t+1}|x_{t+1}) \qquad \text{Observation density}$$


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

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)

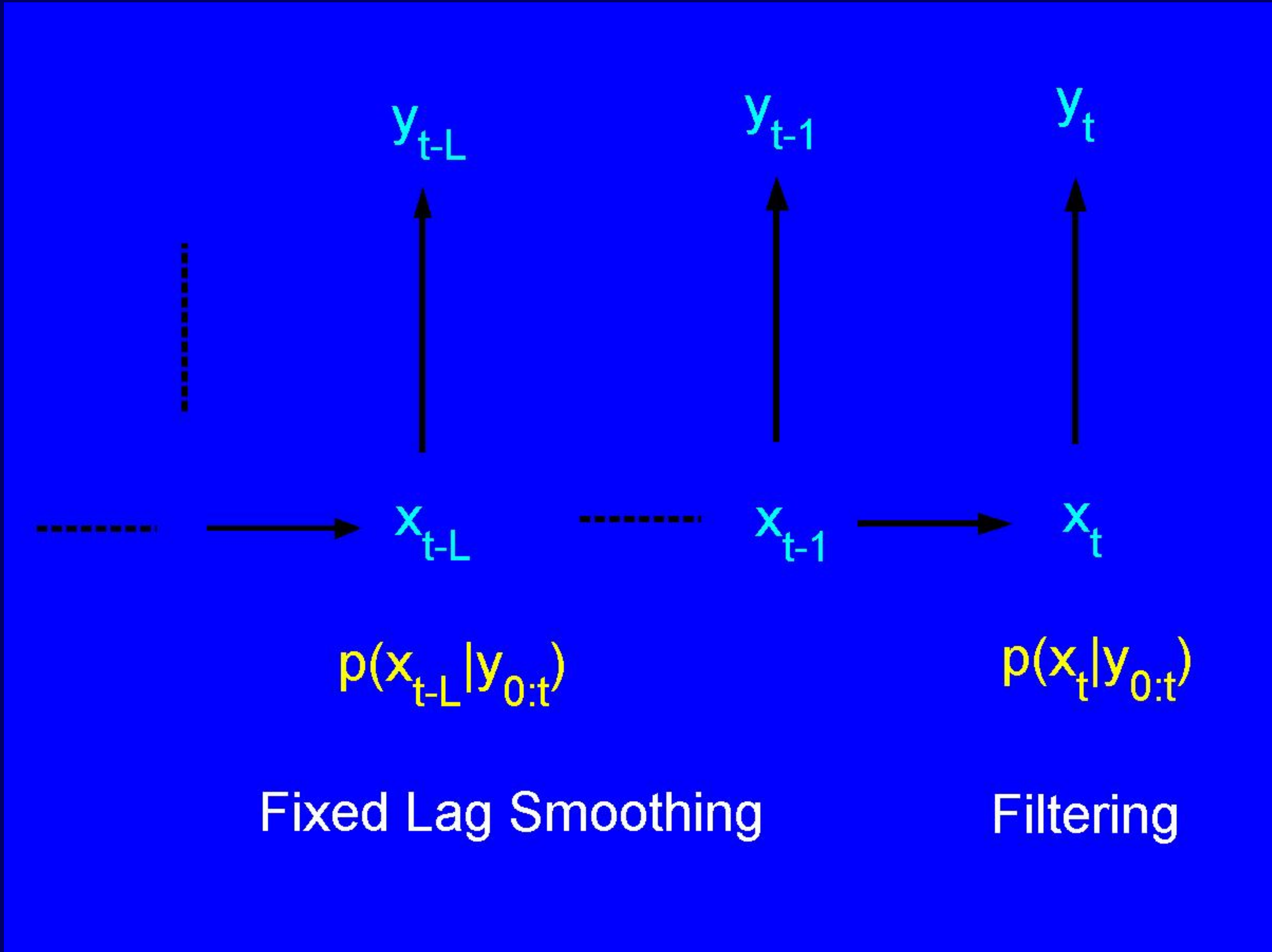
- Smoothing ('fixed lag'):

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

- 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:

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

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:

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

Correction step (Bayes' Theorem):

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

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, ...)

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:

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

- 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.

A Basic 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 $w_{t+1}^{(i)} = 1/N$.

Example: standard nonlinear model

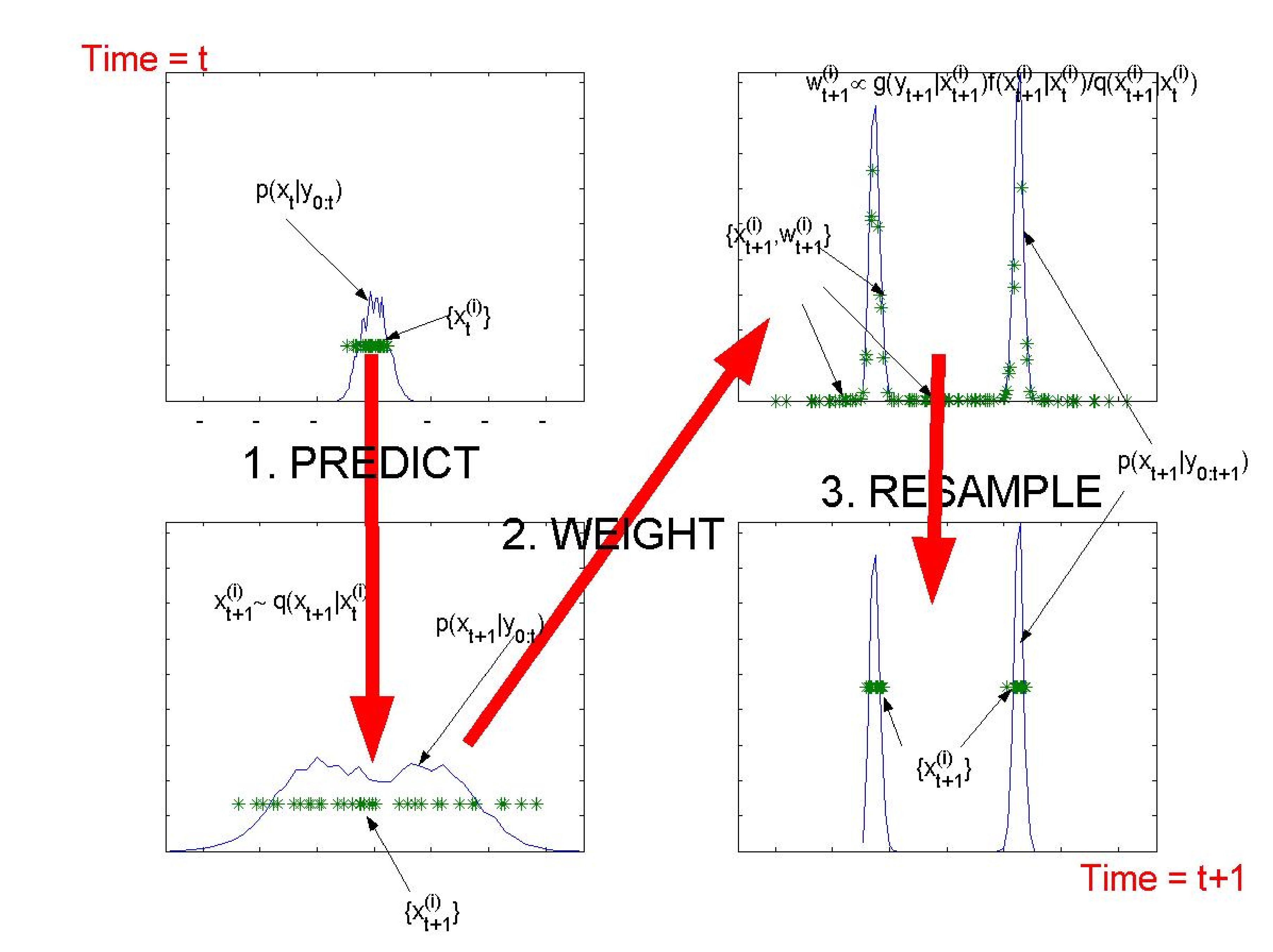
$$
\begin{aligned}
x_t &= A(x_{t-1}) + v_t \\
&= \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2t) + v_t \\
y_t &= B(x_t) + w_t \\
&= \frac{(x_t)^2}{20} + w_t
\end{aligned}
$$

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:

$$
f(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}|A(x_t), \sigma_v^2)
$$

$$
g(y_t|x_t) = \mathcal{N}(y_t|B(x_t), \sigma_w^2)
$$



Time = t
$p(x_t|y_{0:t})$
$\{x_t^{(i)}\}$
1. PREDICT
$x_{t+1}^{(i)} \sim q(x_{t+1}|x_t^{(i)})$
$p(x_{t+1}|y_{0:t})$
$\{x_{t+1}^{(i)}\}$
2. WEIGHT
$w_{t+1}^{(i)} \propto g(y_{t+1}|x_{t+1}^{(i)})f(x_{t+1}^{(i)}|x_t^{(i)})/q(x_{t+1}^{(i)}|x_t^{(i)})$
$\{x_{t+1}^{(i)}, w_{t+1}^{(i)}\}$
3. RESAMPLE
$p(x_{t+1}|y_{0:t+1})$
$\{x_{t+1}^{(i)}\}$
Time = t+1

Smoothing with particle filters

[Work with Arnaud Doucet, Mike West and William Fong, see Godsill, Doucet and West JASA (to appear), Fong, Godsill, Doucet and West (IEEE 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 (to appear), Fong, Godsill, Doucet and West (IEEE 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.
- 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

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

- 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 assuptions 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

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

- 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 assuptions 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

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

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

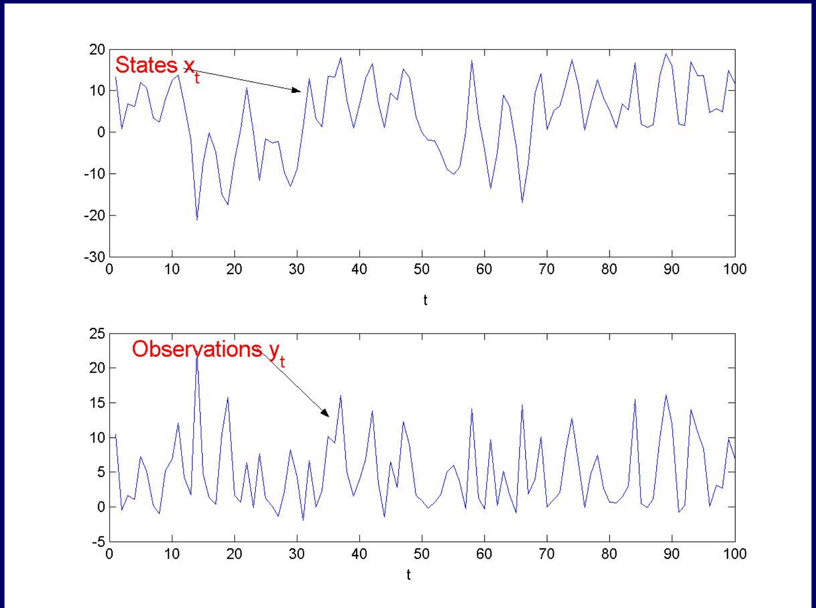
is then an (approximate) random draw from

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

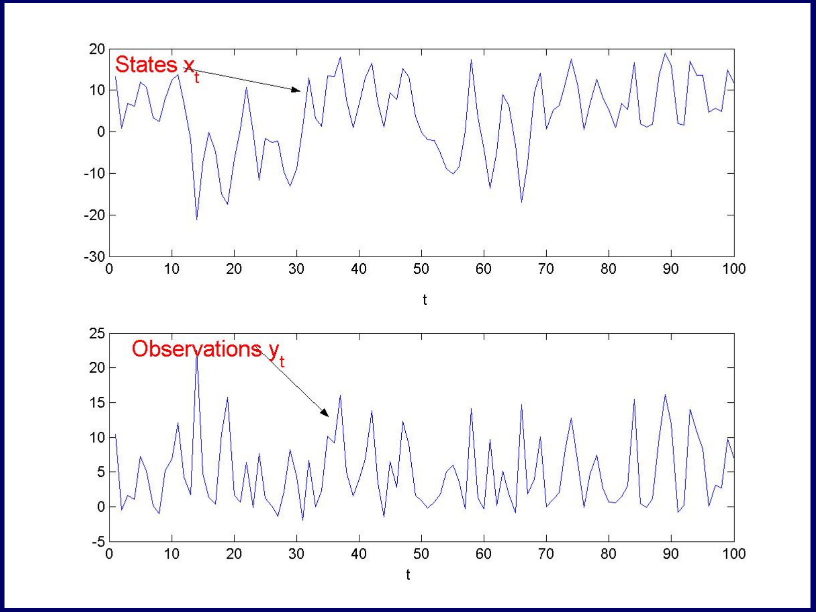
Repeated application allows Monte Carlo estimation of the smoothed state sequence.

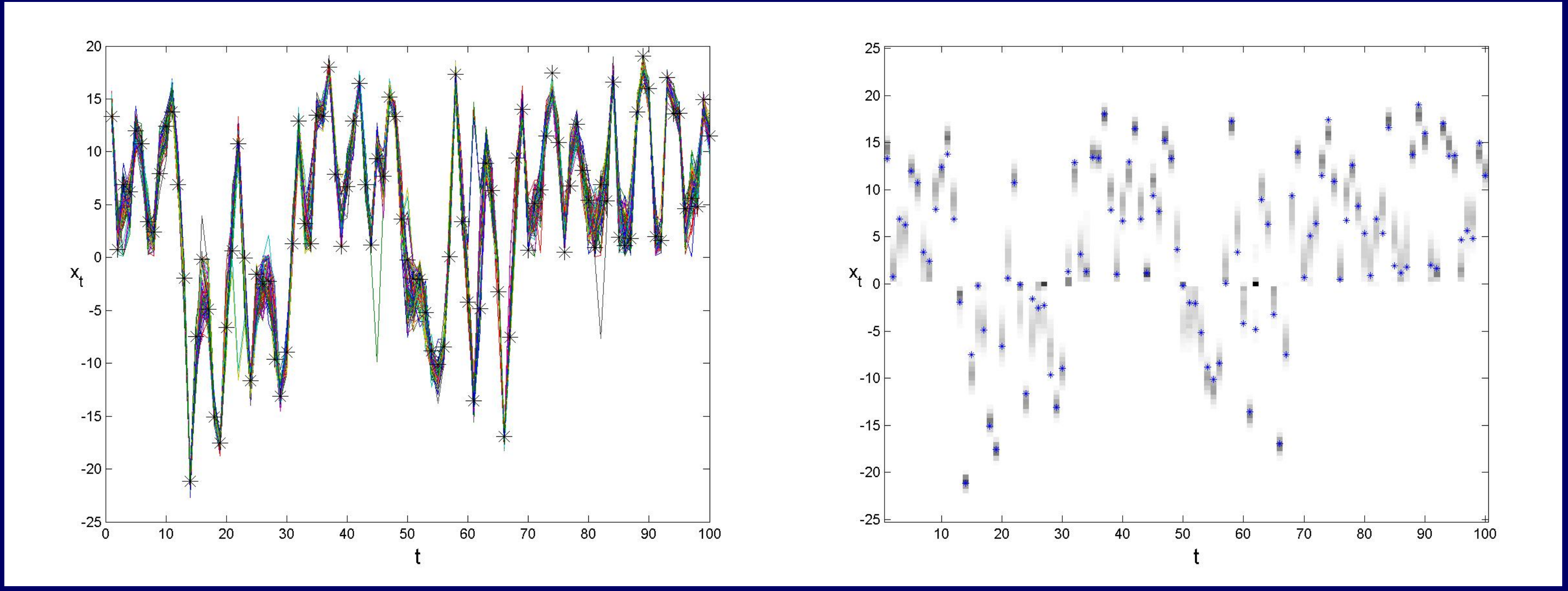
Variants on the algorithm also allow MAP smoothing, see Godsill, Doucet and West 2001 (Ann. Inst. St. Math.)

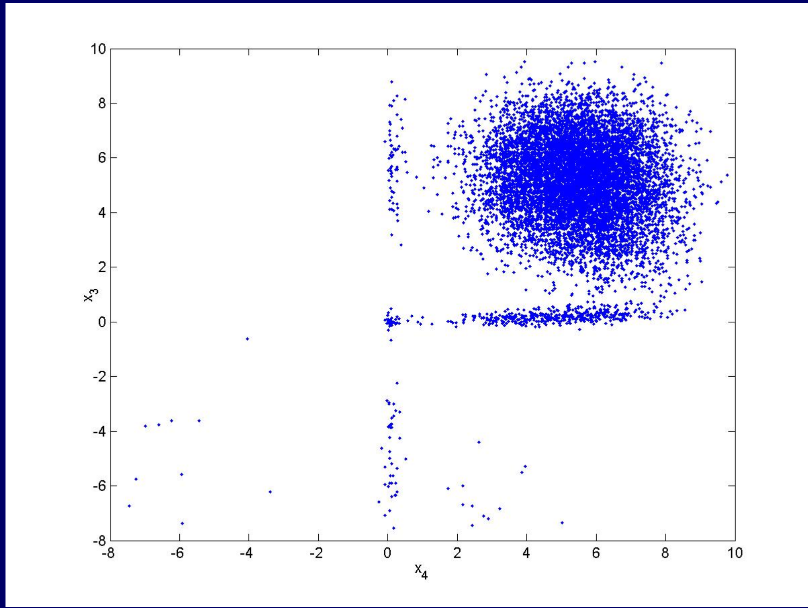
Example - the nonlinear model

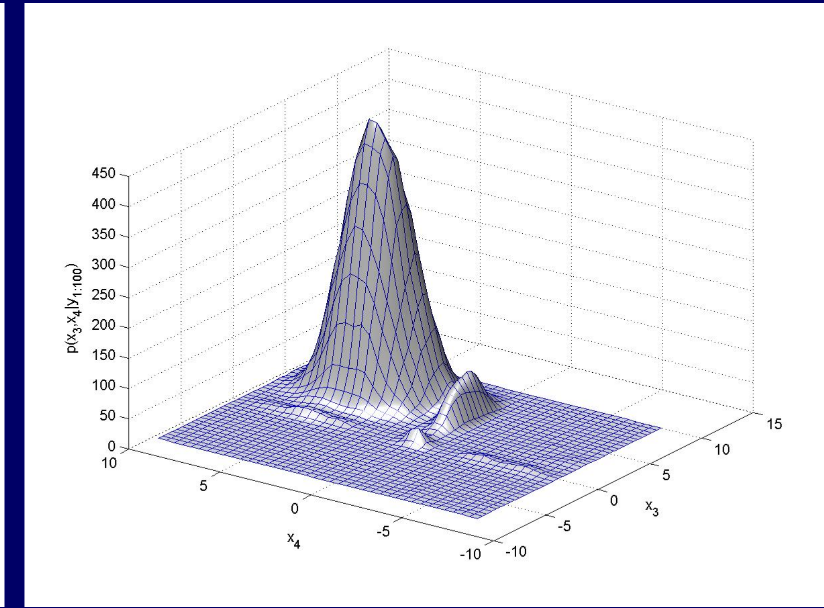


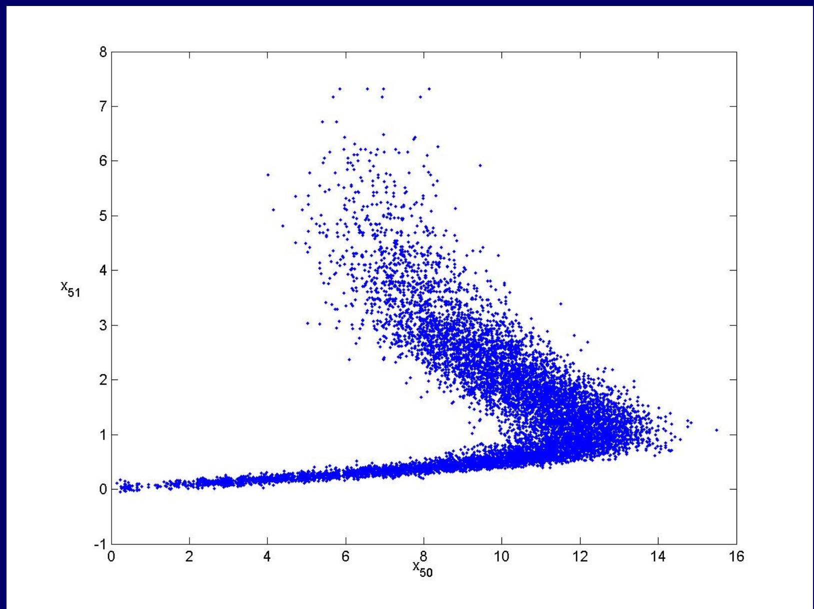
Example - the nonlinear model

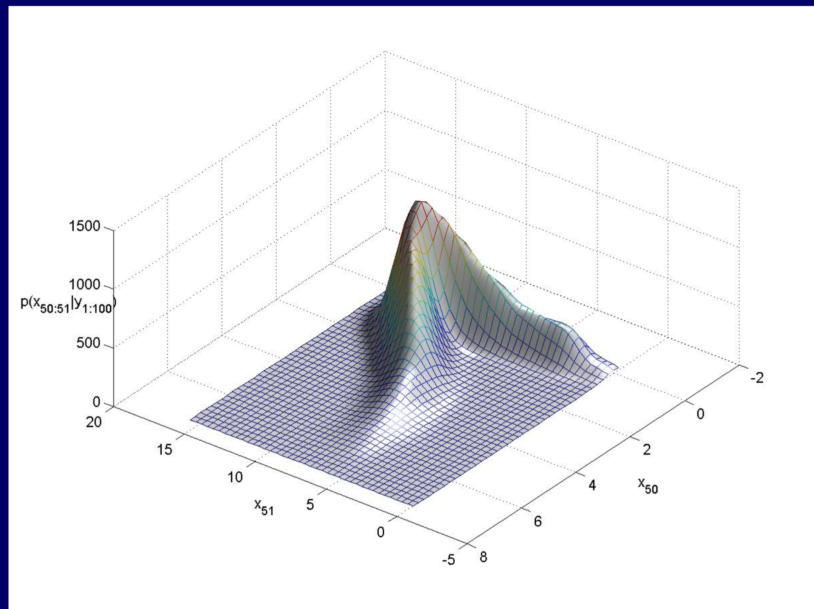





x3
x4


p(x3,x4|y1:100)
x4
x3


x51
x50


p(x50:51|y1:100)
x51
x50

Example - a nonlinear TVAR model for non-stationary speech

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

Example - a nonlinear TVAR model for non-stationary speech

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

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

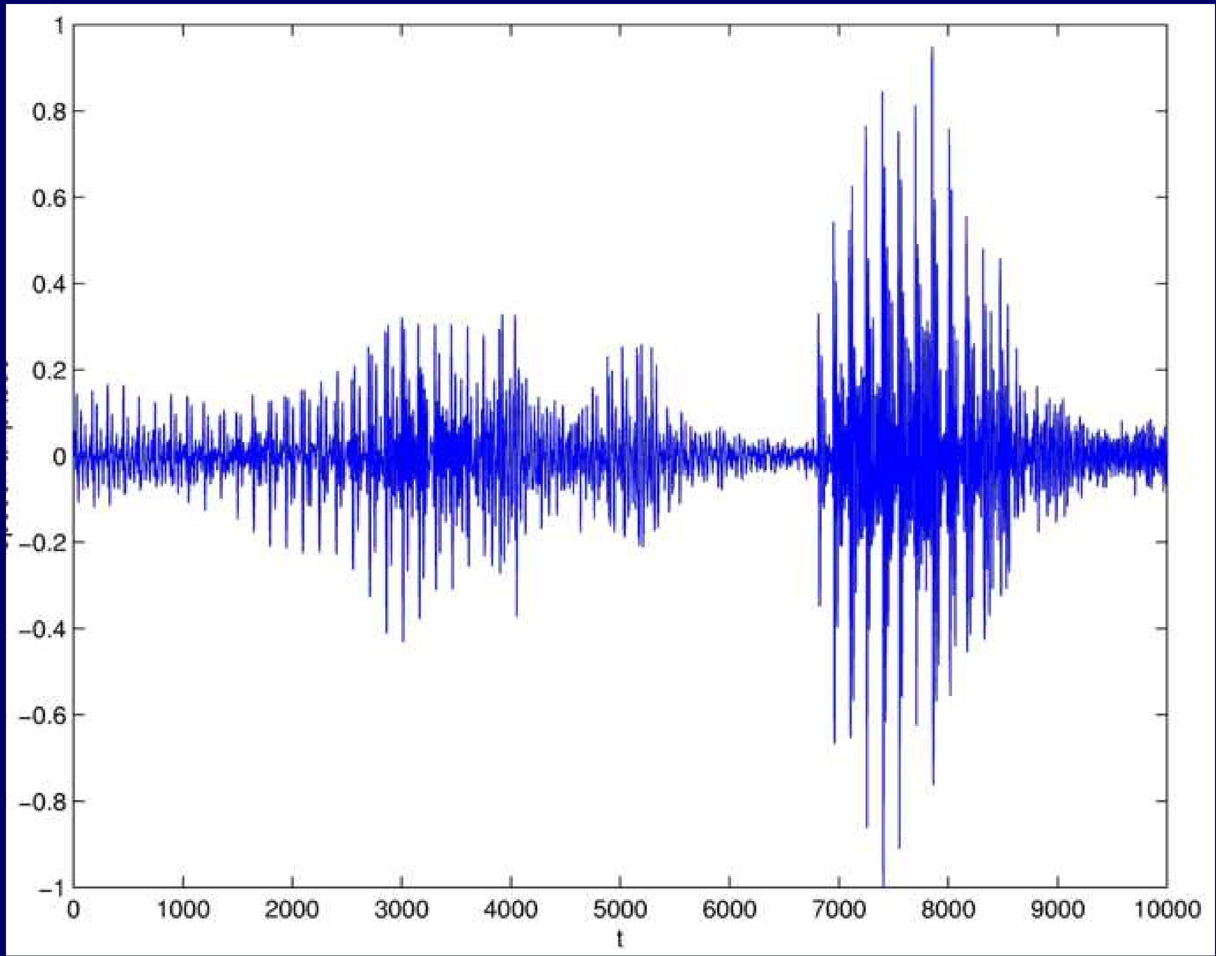


Figure 1: Speech data. 0.62s of a US male speaker saying the words '...rewarded by...'. Sample rate 16kHz, resolution 16-bit, from the TIMIT speech database

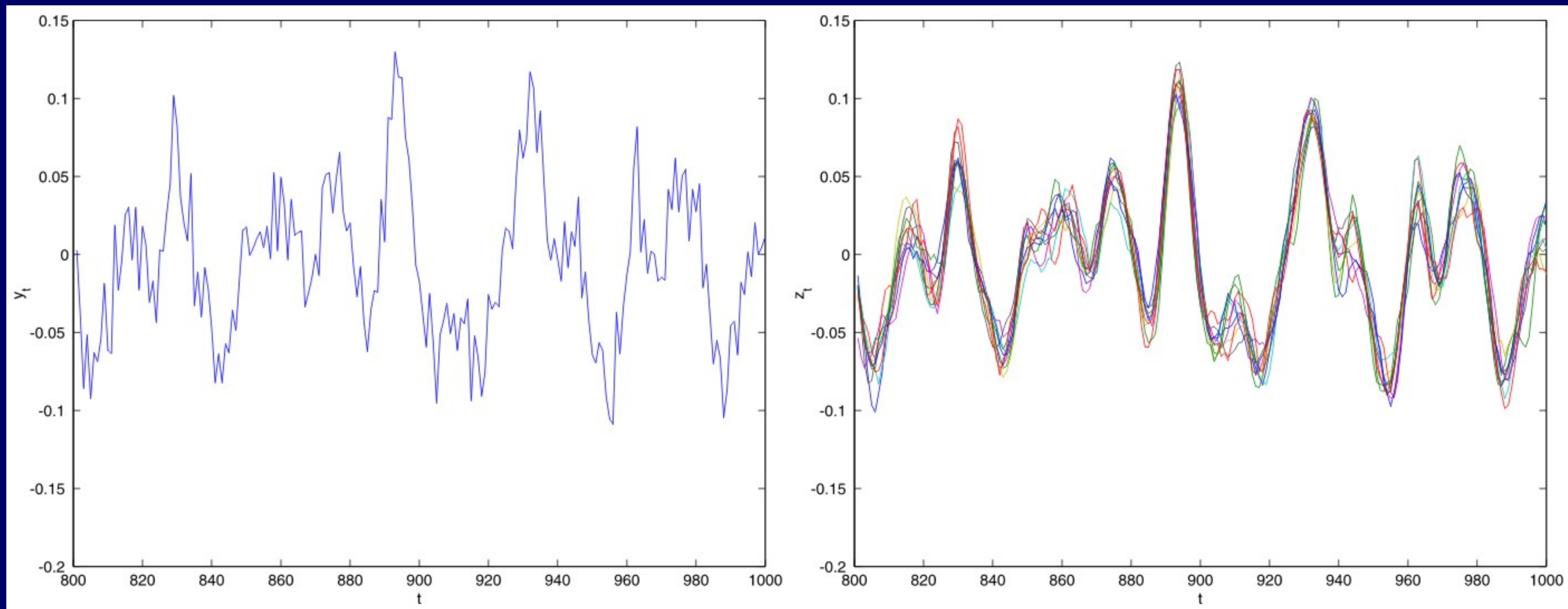


Figure 2: Noisy speech, t=801,...,1000, and smoothed realisations

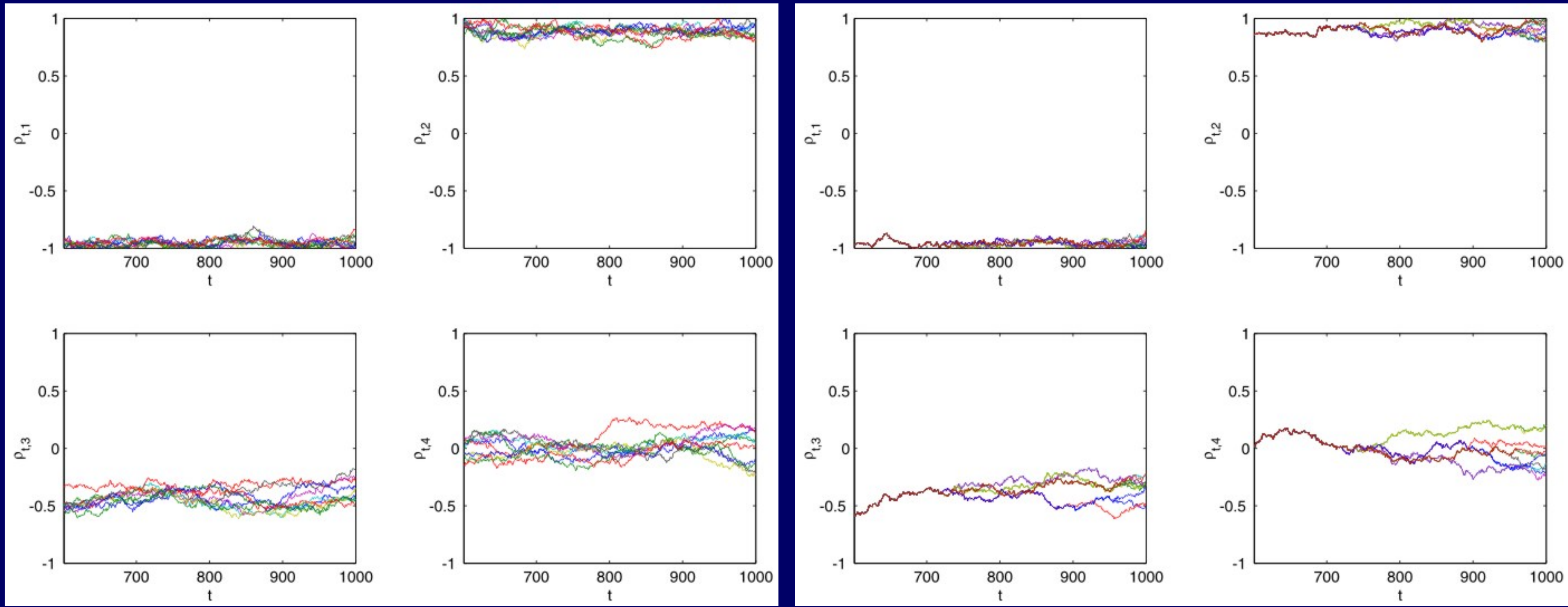


Figure 3: 10 realizations from the smoothing density for the TV-PARCOR coefficients (LHS) compared with standard trajectory-based method (RHS).

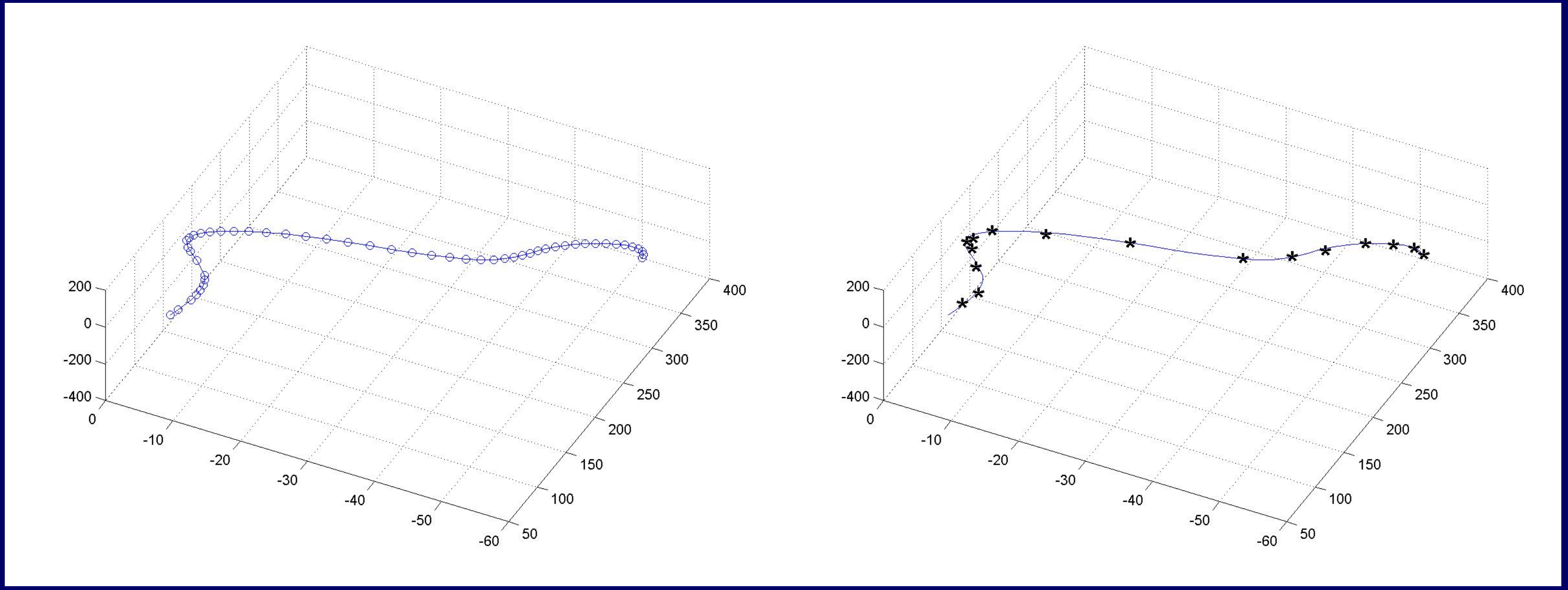
Several improvements to the basic smoothing method have been developed (Fong, Godsill, Doucet and West (2002)), motivated by the TVAR application:

- Block-based smoother - smoothing performed in small batches of N data points. Saves on memory requirements and suits applications where data arrive sequentially in batches.

- Rao-Blackwellised smoothing. As with Monte Carlo filtering, improvements are achieved if some states are marginalised. The Monte Carlo smoother formulae are modified appropriately.

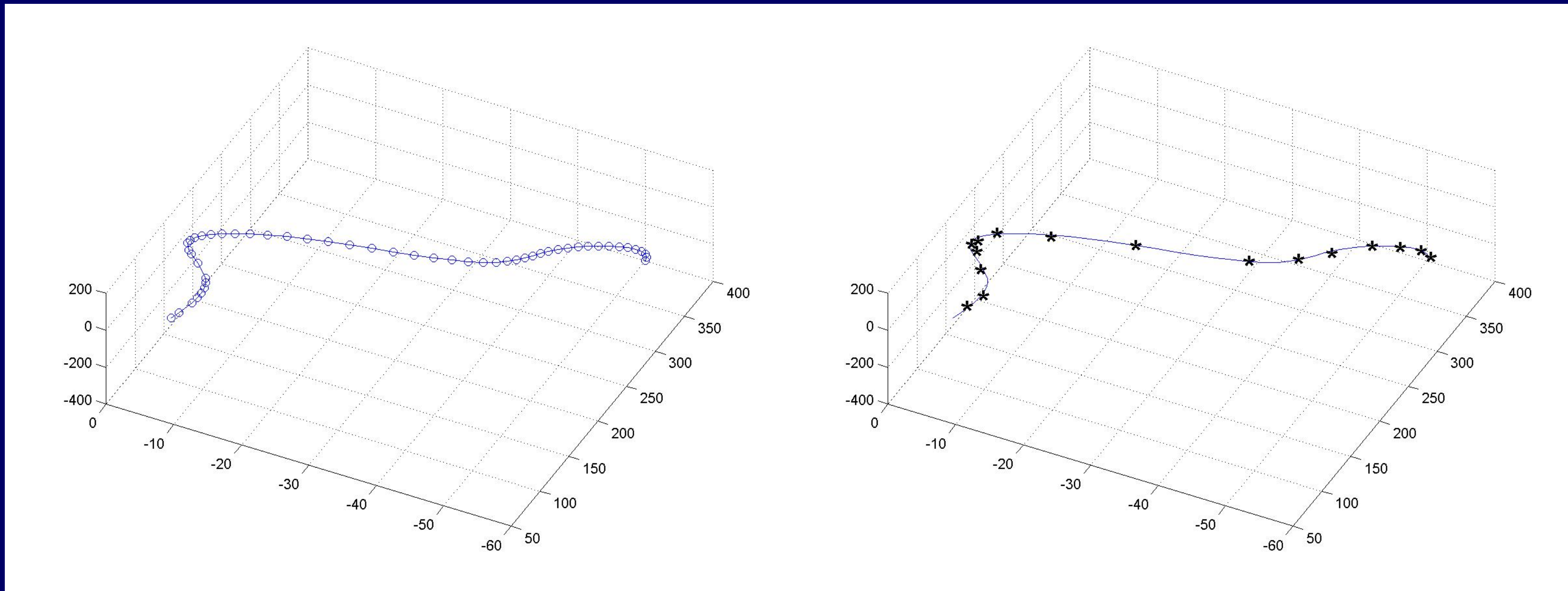
Multirate and trans-dimensional particle filters

Work with William Fong, Jaco Vermaak and Arnaud Doucet



Multirate and trans-dimensional particle filters

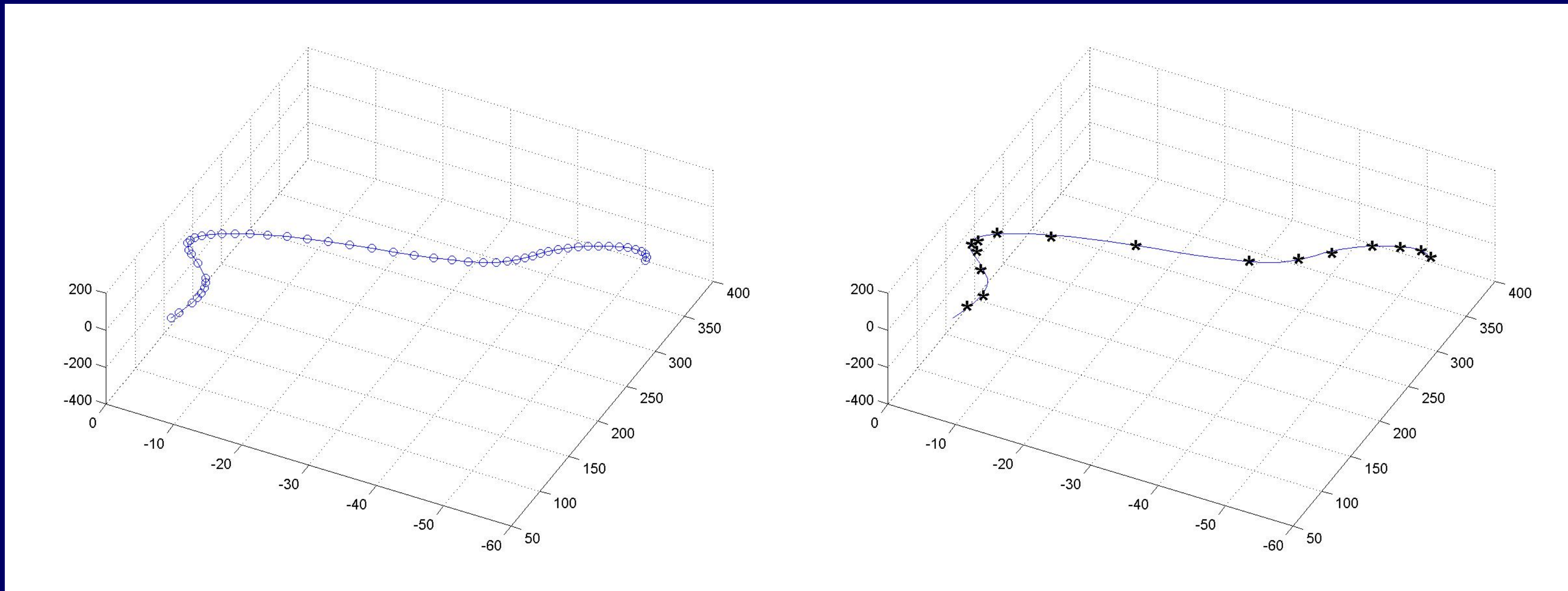
Work with William Fong, Jaco Vermaak and Arnaud Doucet



- In this work particle filters are extended to cases where the state process arrives at a different rate to the observation process

Multirate and trans-dimensional particle filters

Work with William Fong, Jaco Vermaak and Arnaud Doucet



- In this work particle filters are extended to cases where the state process arrives at a different rate to the observation process
- This allows for dynamical model selection within the SMC framework
- Motivated by examples in radar tracking, Bayesian curve fitting, audio parameter modelling, musical beat tracking and statistical learning theory

We now construct a modifed dynamical model having random time indices:

$$\tau_k \sim f_1(\tau_k|\tau_{k-1})$$

and corresponding parameter values:

$$\theta_k \sim f_2(\theta_k|\theta_{k-1})$$

We now construct a modifed dynamical model having random time indices:

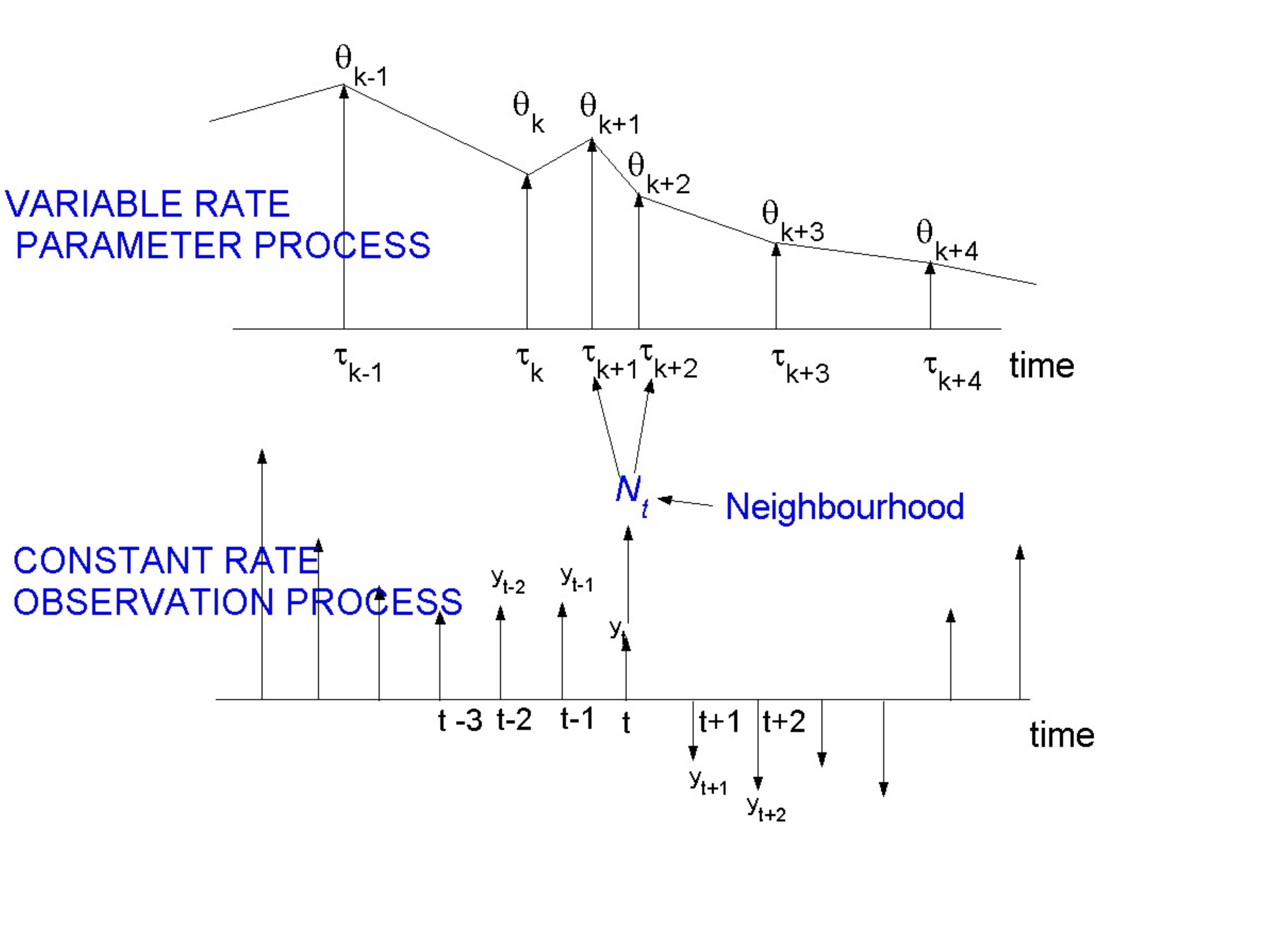
$$\tau_k \sim f_1(\tau_k|\tau_{k-1})$$

and corresponding parameter values:

$$\theta_k \sim f_2(\theta_k|\theta_{k-1})$$

Each observation y_t now depends on a local neighbourhood $\mathcal{N}_t$ of θ_k values:

$$y_t \sim g(y_t|\{\theta_k;\ k \in \mathcal{N}_t)$$



θ k-1
θ k
θ k+1
θ k+2
θ k+3
θ k+4
VARIABLE RATE
PARAMETER PROCESS
τ k-1
τ k
τ k+1
τ k+2
τ k+3
τ k+4
time
N t
Neighbourhood
CONSTANT RATE
OBSERVATION PROCESS
y t-2
y t-1
y t
t -3
t-2
t-1
t
t+1
t+2
time
y t+1
y t+2

- An effective particle filter and smoother can be derived and generalised further for this more sophisticated setting - the trans-dimensional particle filter - see Vermaak, Godsill and Doucet - poster this morning

- Results so far encouraging for applications in TVAR speech modelling, Bayesian curve-fitting and statistical learning theory

Example: musical beat tracking

[work with Tai Lam]

- Musical beat is to be estimated from detected 'onset times' from a musical audio track - formulate as a binary observation process (no amplitude information used here):

$$\begin{cases} y_t = 1 & \text{Candidate onset detected at frame } t \\ y_t = 0 & \text{No detection at frame } t \end{cases}$$

Example: musical beat tracking

[work with Tai Lam]

- Musical beat is to be estimated from detected 'onset times' from a musical audio track - formulate as a binary observation process (no amplitude information used here):

$$\begin{cases} y_t = 1 & \text{Candidate onset detected at frame } t \\ y_t = 0 & \text{No detection at frame } t \end{cases}$$

- Model times of successive beats in the audio using a variable rate process:

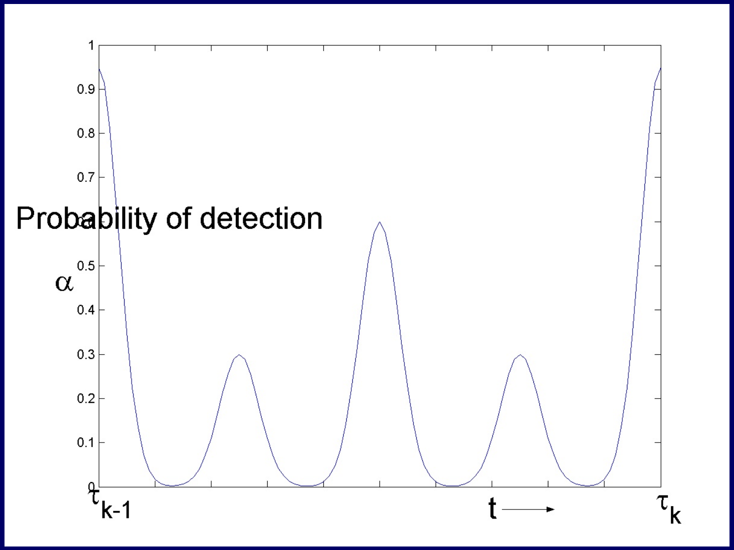
$$\tau_k = h(\tau_{k-1}, \tau_{k-2}) + v_k$$

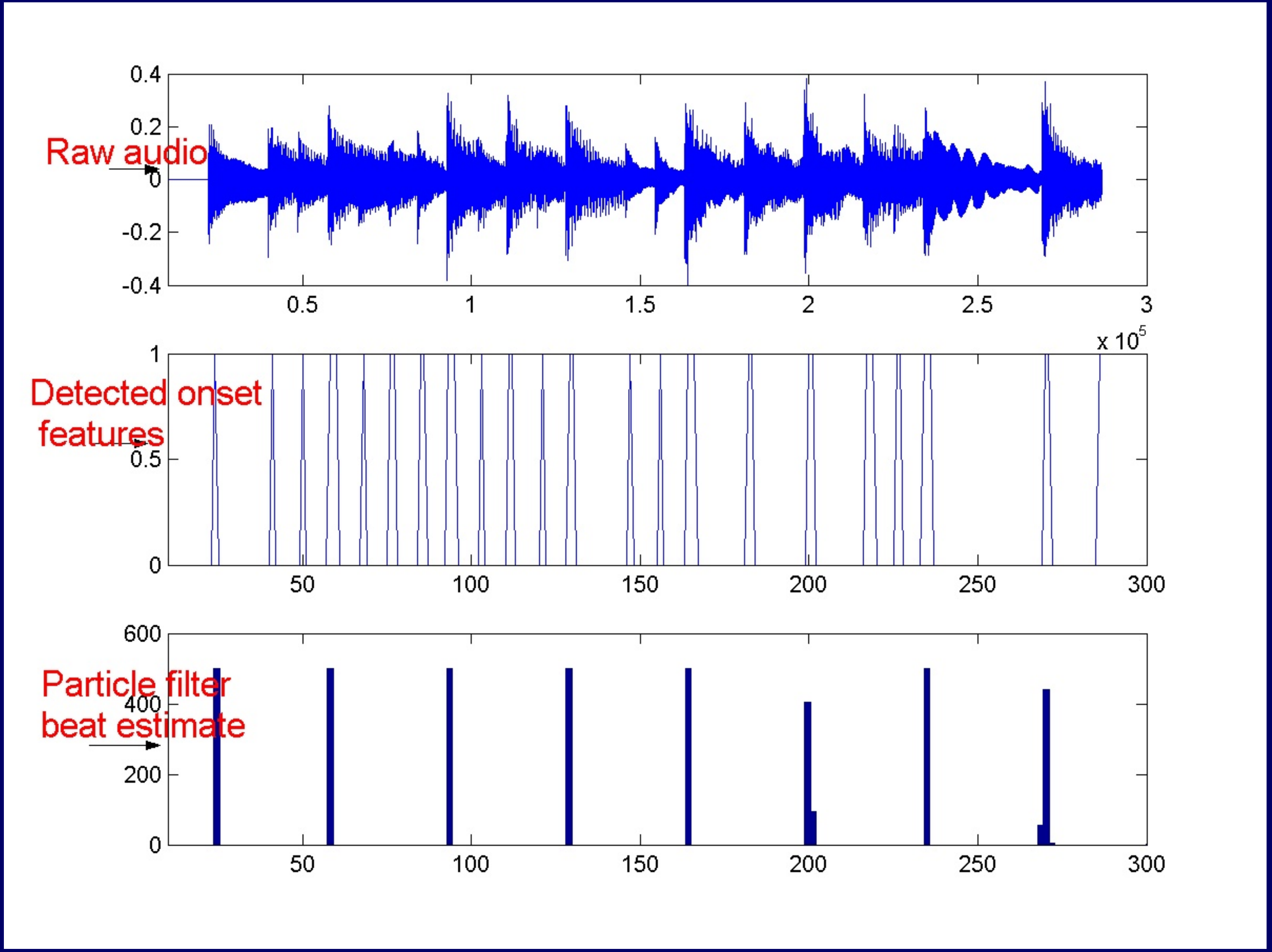
where $h()$ gives the next predicted beat time in terms of the previous two, and v_k is a random disturbance.

- Connect $\{y_t\}$ and τ_k via a Bernoulli likelihood function:

$$y_t \sim \text{Bernoulli}(\alpha(\{\tau_k;\, k \in \mathcal{N}_t\}))$$

Here $\mathcal{N}_t$ contains the two closest beat times to the current frame t.




Raw audio
Detected onset features
Particle filter beat estimate
0.4
0.2
0
-0.2
-0.4
0.5
1
1.5
2
2.5
3
x 10^5
1
0.5
0
50
100
150
200
250
300
600
400
200
0
50
100
150
200
250
300

This is a very simplified model that works nicely on straightforward data. For more elaborate and robust particle filter models, see the work of Robin Morris or Steve Hainsworth.

Example: TVAR speech modelling

[work with William Fong]

- In many modelling scenarios some or all parameters are expected to be slowly and smoothly varying with time - e.g. in the TVAR speech audio model, the AR coefficients $a_{t,i}$ vary much more slowly and smoothly than the signal z_t and observation y_t.

Example: TVAR speech modelling

[work with William Fong]

- In many modelling scenarios some or all parameters are expected to be slowly and smoothly varying with time - e.g. in the TVAR speech audio model, the AR coefficients $a_{t,i}$ vary much more slowly and smoothly than the signal z_t and observation y_t.

- In a 'standard' modelling setup these STV parameters might be modelled by a random walk with very low variance (or some higher order [smooth] difference equation). This can lead to computational and numerical problems.

Example: TVAR speech modelling

[work with William Fong]

- In many modelling scenarios some or all parameters are expected to be slowly and smoothly varying with time - e.g. in the TVAR speech audio model, the AR coefficients $a_{t,i}$ vary much more slowly and smoothly than the signal z_t and observation y_t.
- In a 'standard' modelling setup these STV parameters might be modelled by a random walk with very low variance (or some higher order [smooth] difference equation). This can lead to computational and numerical problems.
- This can be 'fixed' by inflating the variance of the random walk model, but then sampled parameter traces vary too rapidly and model short term signal fluctuations rather than overall parameter trends.

- We propose the alternative approach using the multirate state space model and particle filter, in which some parameters vary on a different time-scale to others in the model (see Fong and Godsill ICASSP (2002)).

Multirate TVAR models

The model now contains two dynamic parameters: θ, the AR coefficients and z, the signal. θ is parametrised on a time grid K times coarser than z:

$$\theta_\tau \sim f(\theta_\tau|\theta_{\tau-1}) \qquad\qquad \phi_t = h_t(\{\theta_\tau;\, \tau \in \mathcal{N}_t\})$$

$$z_t \sim f(z_t|z_{t-1}, \phi_t)$$

$$y_t \sim g(y_t|z_t, \phi_t)$$

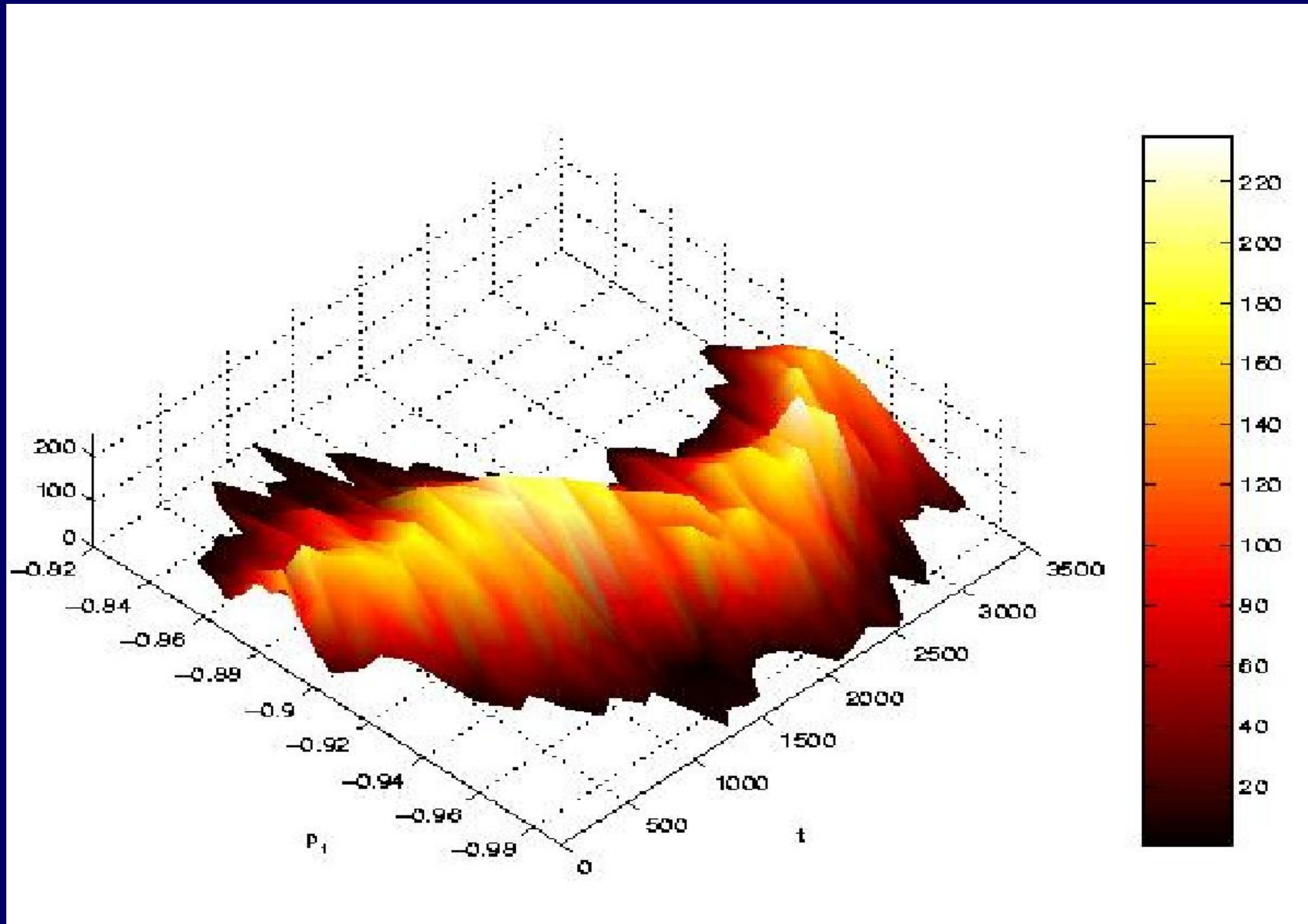
Here $h_t()$ is some suitably smooth interpolation function which interpolates intermediate ϕ_t values from a local neighbourhood of coefficients θ_τ. We have used linear interpolators and spline interpolators, but many other possibilities.

Experimental Results

- Speech Data:
  S1: Good service should be rewarded by big tips
  S2: Draw every outer line first, then fill in the interior

Clip	Input SNR	Proposed	Extended Kalman filter/smoother
S1	$0dB$	$3.86dB$	$1.92dB$
S1	$10dB$	$2.54dB$	$0.99dB$
S1	$20dB$	$1.08dB$	$0.87dB$
S2	$0dB$	$4.31dB$	$2.21dB$
S2	$10dB$	$2.80dB$	$1.57dB$
S2	$20dB$	$1.35dB$	$1.09dB$

Plot of time-varying posterior distribution for $\rho_{t,1}$



Oustanding Challenges

- The particle filter/smoother plus its adaptations, make a powerful, computationally intensive, suite of methods for inference in large datasets.
- Fixed parameter problems $p(x_t|\theta, y_{0:t})$ are an on-going challenge
- Large scale problems with many objects, parameters...

References

[1] N J Gordon, D J Salmond, and A F M Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings-F, 140(2):107–113, April 1993.

[2] S. J. Godsill, A Doucet, and M West. Monte Carlo smoothing for non-linear time series. J. Am. Stat. Soc., 2003. In Press.

[3] W. Fong, S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing with application to speech enhancement. IEEE Trans. on Signal Processing, 50(2):438–449, February 2002. Special issue on Monte Carlo Methods.

[4] S. J. Godsill, A Doucet, and M West. Maximum a posteriori sequence estimation using Monte Carlo particle filters. Ann. Inst. Stat. Math., 53(1):82–96, March 2001.