Mitigating Degeneracy

A Local Particle-Filter Approach

Adam M. Johansen and Arnaud Doucet

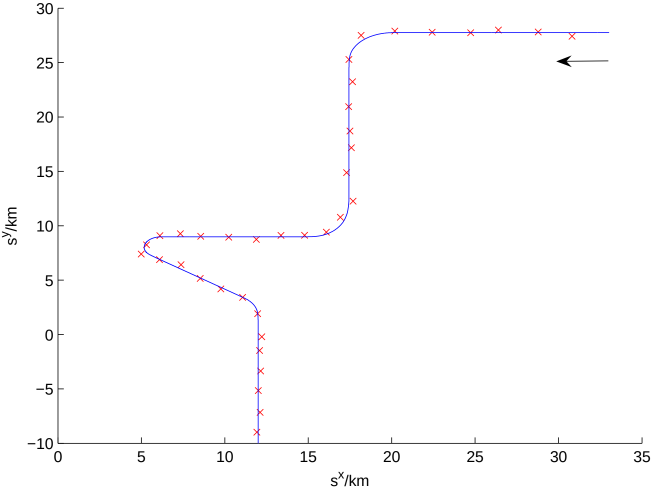
University of Warwick
a.m.johansen@warwick.ac.uk
http://go.warwick.ac.uk/amjohansen/

November 18th, 2011
AMSTAT Seminar

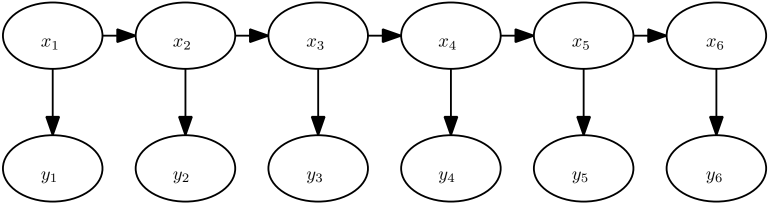
Outline

- Background
  - Hidden Markov Models / State Space Models
  - Particle Filters / Sequential Monte Carlo
  - Block Sampling (method and motivation)
- Local SMC
  - Motivation
  - Formulation
- Examples
  - A Toy Linear Gaussian Model
  - Stochastic Volatility

The Structure of the Problem



Hidden Markov Models / State Space Models



- Unobserved Markov chain $\{X_n\}$ transition f.
- Observed process $\{Y_n\}$ conditional density g.
- Joint density:

$$p(x_{1:n}, y_{1:n}) = f_1(x_1)g(y_1|x_1)\prod_{i=2}^{n} f(x_i|x_{i-1})g(y_i|x_i).$$

Motivating Examples

- Tracking, e.g. ACV Model:
  - States: $x_n = [s_n^x \; u_n^x \; s_n^y \; u_n^y]^T$
  - Dynamics: $x_n = Ax_{n-1} + \epsilon_n$

$$\begin{bmatrix} s_n^x \\ u_n^x \\ s_n^y \\ u_n^y \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{n-1}^x \\ u_{n-1}^x \\ s_{n-1}^y \\ u_{n-1}^y \end{bmatrix} + \epsilon_n$$

  - Observation: $y_n = Bx_n + \nu_n$

$$\begin{bmatrix} r_n^x \\ r_n^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_n^x \\ u_n^x \\ s_n^y \\ u_n^y \end{bmatrix} + \nu_n$$

- Stochastic Volatility, e.g.:

$$\begin{aligned} f(x_i|x_{i-1}) =& \mathcal{N}\left(\phi x_{i-1}, \sigma^2\right) \\ g(y_i|x_i) =& \mathcal{N}\left(0, \beta^2 \exp(x_i)\right) \end{aligned}$$

Formal Solutions

- Filtering: Prediction and Update Recursions:

$$p(x_n|y_{1:n-1}) = \int p(x_{n-1}|y_{1:n-1})f(x_n|x_{n-1})dx_{n-1}$$

$$p(x_n|y_{1:n}) = \frac{p(x_n|y_{1:n-1})g(y_n|x_n)}{\int p(x'_n|y_{1:n-1})g(y_n|x'_n)dx'_n}$$

- Smoothing:

$$p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n-1}|y_{1:n-1})f(x_n|x_{n-1})g(y_n|x_n)}{\int g(y_n|x'_n)f(x'_n|x_{n-1})p(x'_{n-1}|y_{1:n-1})dx'_{n-1:n}}$$

The Monte Carlo Method

- Given a probability density, p,

$$I = \int_E \varphi(x)p(x)dx$$

- Simple Monte Carlo solution:
  - Sample $X_1, \ldots, X_N \stackrel{\text{iid}}{\sim} p$.
  - Estimate $\widehat{I} = \frac{1}{N}\sum_{i=1}^{N} \varphi(X_i)$.
- Can also be viewed as approximating $\pi(dx) = p(x)dx$ with

$$\widehat{\pi}^N(dx) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_i}(dx).$$

Importance Sampling

- Given q, such that

$$p \ll q \qquad\qquad (\text{and } p(x)/q(x) < M < \infty),$$

  define $w(x) = p(x)/q(x)$ and:

$$I = \int \varphi(x)p(x)dx = \int \varphi(x)w(x)q(x)dx.$$

- This suggests the estimator:
  - Sample $X_1, \ldots, X_N \overset{\text{iid}}{\sim} q$.
  - Estimate $\widehat{I} = \frac{1}{N}\sum\limits_{i=1}^{N} w(X_i)\varphi(X_i)$.
- Can also be viewed as approximating $\pi(dx) = p(x)dx$ with

$$\widehat{\pi}^N(dx) = \frac{1}{N}\sum_{i=1}^{N} w(X_i)\delta_{X_i}(dx).$$

Self-Normalised Importance Sampling

- Often, p is known only up to a normalising constant.
- As $\mathbb{E}_q(Cw\varphi) = C\mathbb{E}_p(\varphi)\ldots$
- If $v(x) = Cw(x)$, then

$$\mathbb{E}_p(\varphi) = \frac{C\mathbb{E}_p(\varphi)}{C\mathbb{E}_p(\mathbf{1})} = \frac{\mathbb{E}_q(Cw\varphi)}{\mathbb{E}_q(Cw\mathbf{1})} = \frac{\mathbb{E}_q(v\varphi)}{\mathbb{E}_q(v\mathbf{1})}$$

- Estimate the numerator and denominator with the same sample:

$$\widehat{I} = \frac{\frac{1}{N}\sum\limits_{i=1}^{N} v(X_i)\varphi(X_i)}{\frac{1}{N}\sum\limits_{i=1}^{N} v(X_i)} = \sum_{i=1}^{N} \frac{v(X_i)}{\sum\limits_{j=1}^{n} v(X_j)}\varphi(X_i).$$

Importance Sampling in The HMM Setting

- Given $p(x_{1:n}|y_{1:n})$ for $n = 1, 2, \ldots$.
- Choose $q_n(x_{1:n}) = q_n(x_n|x_{1:n-1})q_{n-1}(x_{1:n-1})$.
- Weight:

$$
\begin{aligned}
w_n(x_{1:n}) &\propto \frac{p(x_{1:n}|y_{1:n})}{q_n(x_n|x_{1:n-1})q_{n-1}(x_{1:n-1})} \\
&= \frac{p(x_{1:n}|y_{1:n})}{q_n(x_n|x_{1:n-1})p(x_{1:n-1}|y_{1:n-1})} w_{n-1}(x_{1:n-1}) \\
&\propto \frac{f(x_n|x_{n-1})g(y_n|x_n)}{q_n(x_n|x_{n-1})} w_{n-1}(x_{1:n-1})
\end{aligned}
$$

Sequential Importance Sampling – Prediction & Update

- A very simple "particle filter":
  - Choose: $q_1(x_1) = f_1(x_1)$, $q_n(x_n|x_{n-1}) = f(x_n|x_{n-1})$.
  - Hence:

$$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1})g(y_n|x_n)$$

Initialisation: $n \leftarrow 1$

- For $i = 1, \ldots, n$:
  - Sample $X_1^i \sim f_1(\cdot)$
  - Weight $W_1^i \propto g(y_1|X_1^i)$

Recursion: $n \leftarrow n + 1$

- For $i = 1, \ldots, n$:
  - Sample $X_n^i \sim f(\cdot|X_{n-1}^i)$ *(prediction)*
  - Weight $W_n^i \propto W_{n-1}^i g(y_n|X_n^i)$ *(update)*

Actually:

- Better proposals exist…
- but even they aren't good enough.

Resampling

- Stabilisation of importance weights.
- Given $\{W_n^i, X_n^i\}$:
  - Draw $\{\widetilde{X}_n^i\}$ such that:

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\varphi(\widetilde{X}_n^i)\middle|\sigma(\{W_n^j, X_n^j\}_{j=1}^n)\right] = \frac{\sum_{i=1}^n W_n^i \varphi(X_n^i)}{\sum_{i=1}^n W_n^i}$$

  - Replace $\{W_n^i, X_n^i\}_{i=1}^N$ with $\{\frac{1}{N}, \widetilde{X}_n^i\}_{i=1}^N$.
- Simplest approach (multinomial) resampling:

$$\widetilde{X}_n^i \overset{\text{iid}}{\sim} \frac{\sum_{j=1}^n W_n^j \delta_{X_n^j}}{\sum_{j=1^n} W_n^j}$$

- Lower variance options preferable.

Sequential Importance Resampling

Initialisation: $n \leftarrow 1$

- For $i = 1, \ldots, n$:
  - Sample $X_1^i \sim q_1(\cdot)$
  - Weight $W_1^i \propto f_1(X_1^i) g(y_1|X_1^i)/q_1(X_1^i)$

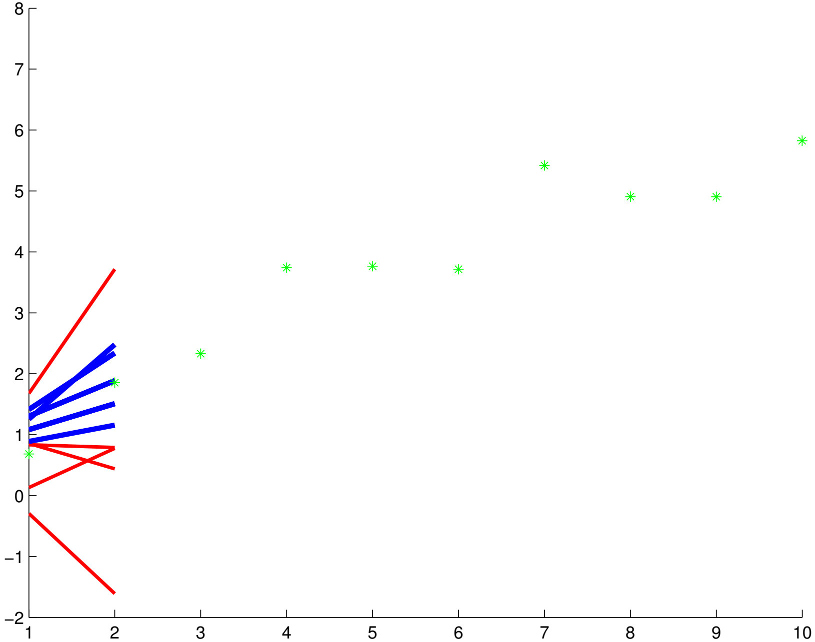
Recursion: $n \leftarrow n + 1$

- **Resample**, obtaining $\{1/N, \widetilde{X}_{1:n-1}^i\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^i)$
  - Weight $W_n^i \propto \frac{f(X_n^i|\widetilde{X}_{n-1}^i) g(y_n|X_n^i)}{q_n(X_n^i|\widetilde{X}_{n-1}^i)}$

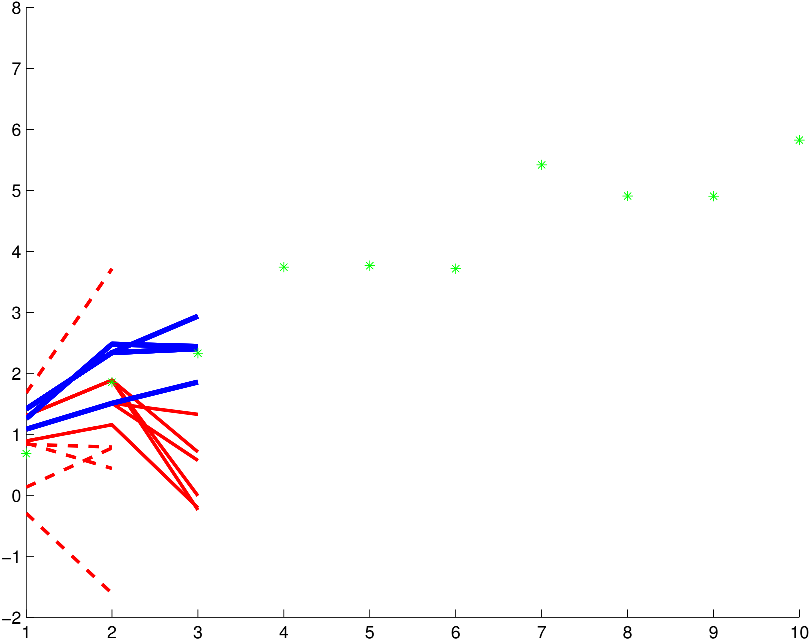
Actually:

- Resample efficiently.
- Only resample when necessary.

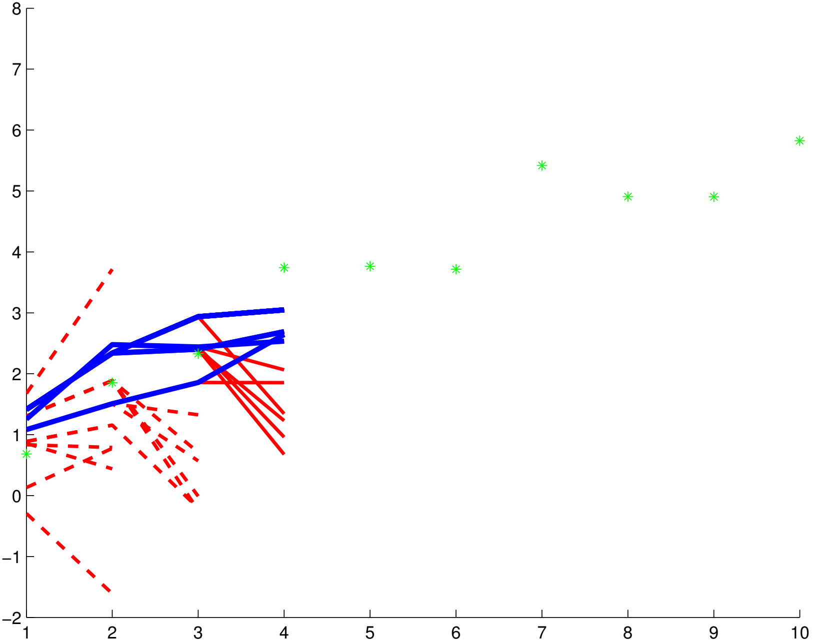
Iteration 2



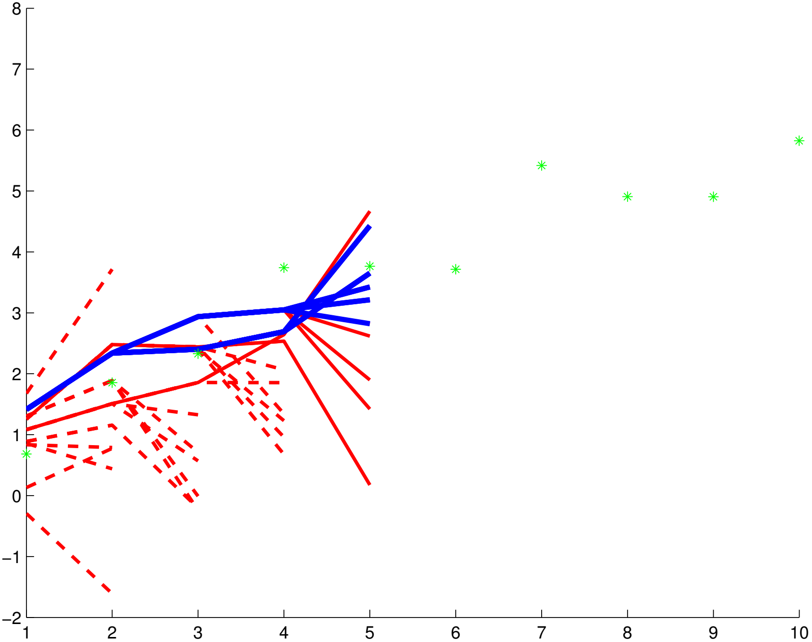
Iteration 3



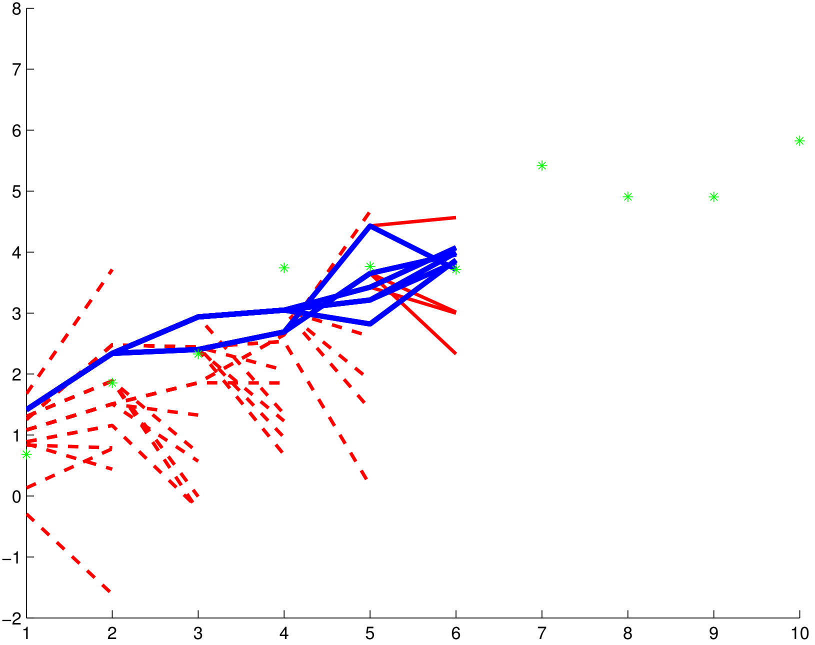
Iteration 4



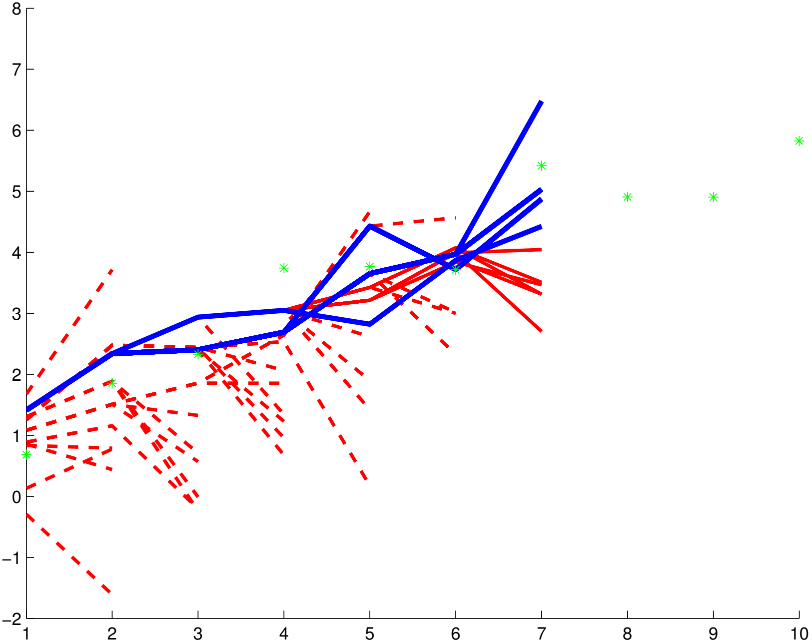
Iteration 5



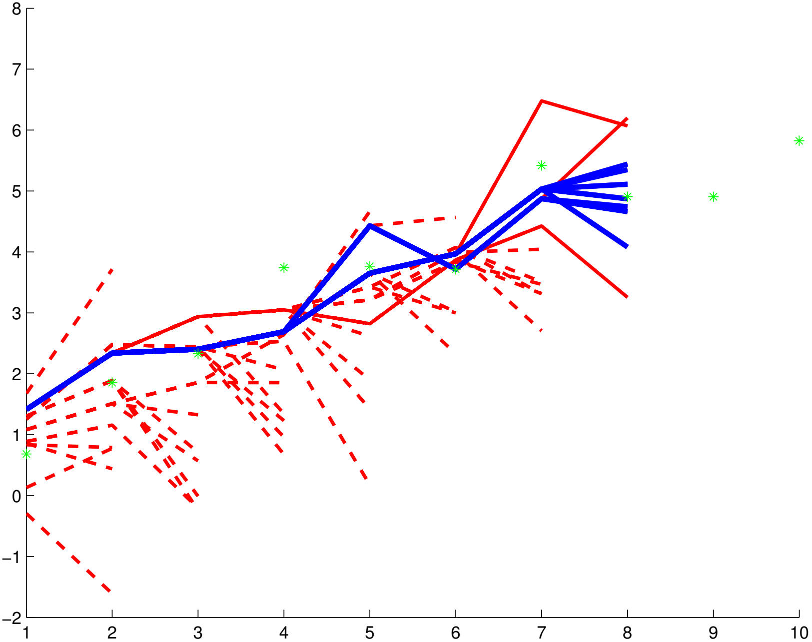
Iteration 6



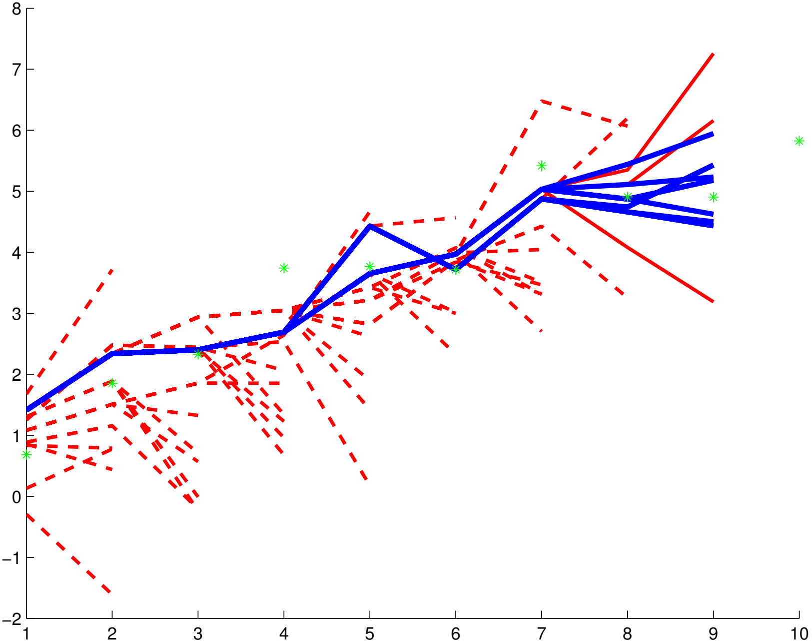
Iteration 7



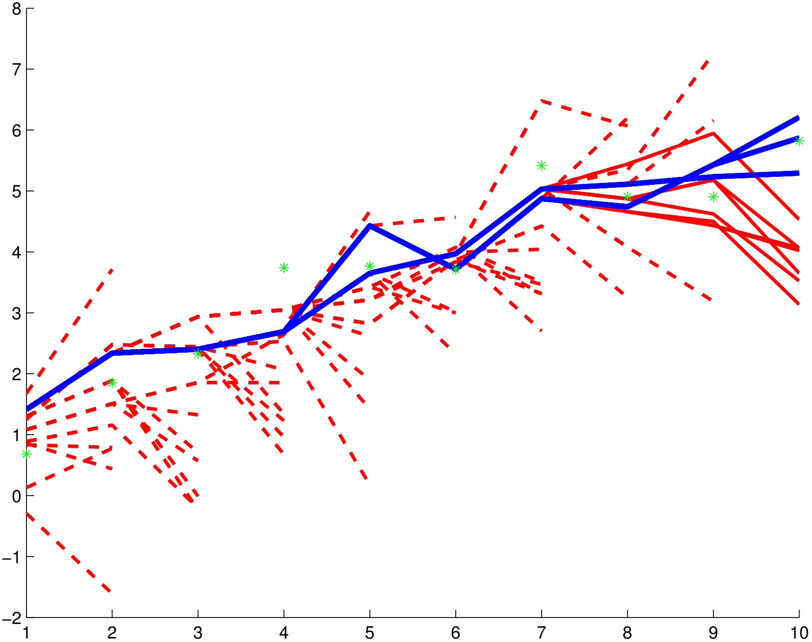
Iteration 8



Iteration 9



Iteration 10



Block Sampling: An Idealised Approach

At time n, given $x_{1:n-1}$; discard $x_{n-L+1:n-1}$:

- Sample from $q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})$.
- Weight with

$$W(x_{1:n}) = \frac{p(x_{1:n}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})q(x_{n-L+1:n}|x_{n-L}, y_{1:n-L+1:n})}$$

- Optimally,

$$q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n}) = p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})$$
$$W(x_{1:n}) \propto \frac{p(x_{1:n-L}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})} = p(y_n|x_{1:n-L}, y_{n-L+1:n-1})$$

- Typically intractable; auxiliary variable approach in [4].

Why *Try* To Block-Sample?

Explicit motivation from the linear Gaussian case:

$$
\begin{aligned}
&\mathsf{Var}_{p(x_{n-L}|y_{1:n-1})p(x_{n-L+1:n}|x_{n-L},y_{1:n})}\left[w(X_{n-L:n})\right] \\
=&\int_{-\infty}^{\infty} \frac{\mathcal{N}^2(x_{n-L};\mu_{n-L|n},\Sigma_{n-L|n})}{\mathcal{N}(x_{n-L};\mu_{n-L|n-1},\Sigma_{n-L|n-1})} dx_{n-L} - 1 \\
=&\frac{\Sigma_{n-L|n-1}}{\sqrt{\Sigma_{n-L|n}(2\Sigma_{n-L|n-1}-\Sigma_{n-L|n})}} \exp\left(\frac{(\mu_{n-L|n}-\mu_{n-L|n-1})^2}{2\Sigma_{n-L|n-1}-\Sigma_{n-L|n}}\right) - 1.
\end{aligned}
$$

Particle MCMC

- MCMC algorithms which employ SMC proposals [1]
- SMC algorithm as a collection of RVs
  - Values
  - Weights
  - Ancestral Lines
- Construct MCMC algorithms:
  - With many auxiliary variables
  - *Exactly* invariant for distribution on extended space
  - Standard MCMC arguments justify strategy

- Does this suggest anything about SMC?
- Can something similar help with smoothing?

More than one SMC Algorithm?

- Standard approach:
  - Run an SIR algorithm with N particles.
  - Use
$$\pi_n^N(dx_{1:n}) = \sum_{i=1}^{N} W_n^i \delta_{X_{1:n}^i}(dx_{1:n}).$$
- A crude alternative:
  - Run $L = \lfloor N/M \rfloor$ algorithms with M particles.
  - Use
$$\pi_n^{M,l}(dx_{1:n}) = \sum_{i=1}^{M} W_n^{l,i} \delta_{X_{1:n}^{l,i}}(dx_{1:n}).$$
  - Guarantees L i.i.d. samples.
  - For small M their distribution may be poor.

Toy Model: Linear Gaussian HMM

- Linear, Gaussian state transition:

$$f(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1}, 1)$$

- and likelihood

$$g(y_t|x_t) = \mathcal{N}(y_t; x_t, 1)$$

- Analytically: Kalman filter/smoother/etc.
- Simple bootstrap PF:
  - Proposal:

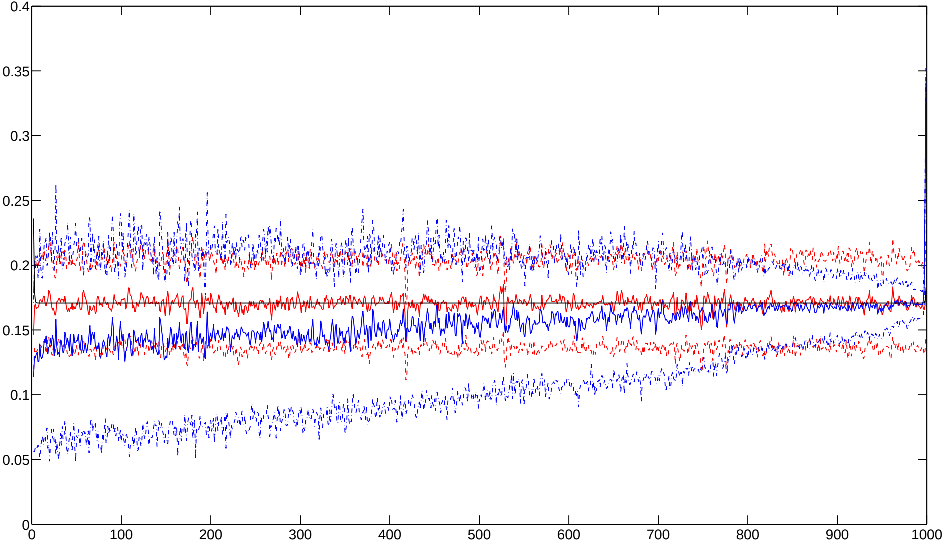
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

$$W(x_{t-1}, x_t) \propto g(y_t|x_t)$$

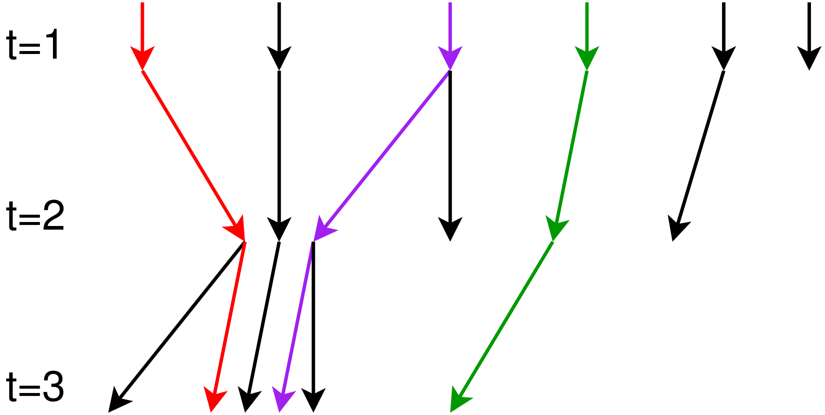
  - Resample residually every iteration.

Covariance Estimation: 1d Linear Gaussian Model



But we want our samples to interact...

Ancestral Trees



$a_3^2 = 1$ $\quad a_3^4 = 3$ $\quad a_2^1 = 1$ $\quad a_2^5 = 4$

$b_{3,1:3}^2 = (1, 1, 2)$ $\quad b_{3,1:3}^4 = (3, 3, 4)$ $\quad b_{3,1:3}^6 = (4, 5, 6)$

SMC Distributions

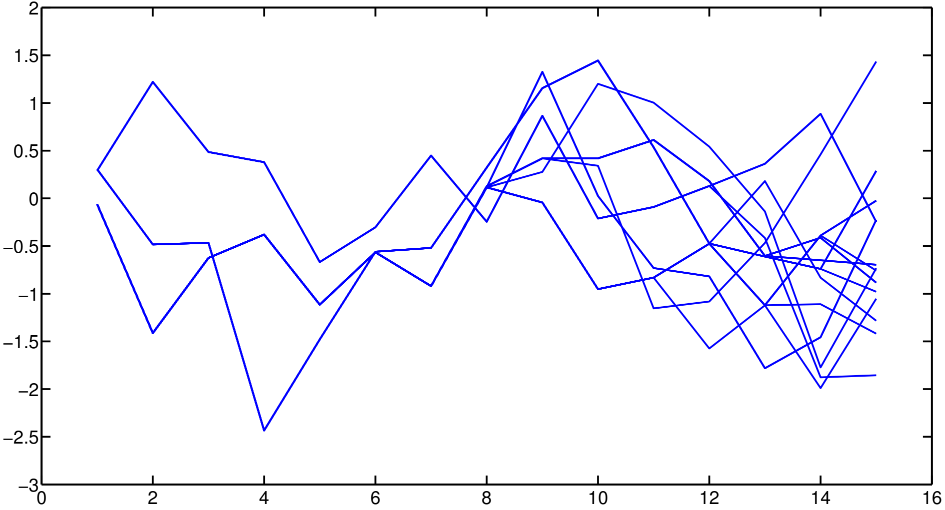
We'll need:

$$\psi^M_{n,L}\left(\overline{\mathbf{a}}_{n-L+2:n}, \overline{\mathbf{x}}_{n-L+1:n}, \overline{k}; x_{n-L}\right)$$
$$= \left[\prod_{i=1}^{M} q(\overline{x}^i_{n-L+1} \,\big|\, \overline{x}_{n-L})\right] \prod_{p=n-L+2}^{n} \left[r(\overline{\mathbf{a}}_p | \overline{\mathbf{w}}_{p-1}) \prod_{i=1}^{M} q\left(\overline{x}^i_p | \overline{x}^{\overline{a}^i_p}_{p-1}\right)\right] r(\overline{k}|\overline{\mathbf{w}}_{\mathbf{n}})$$

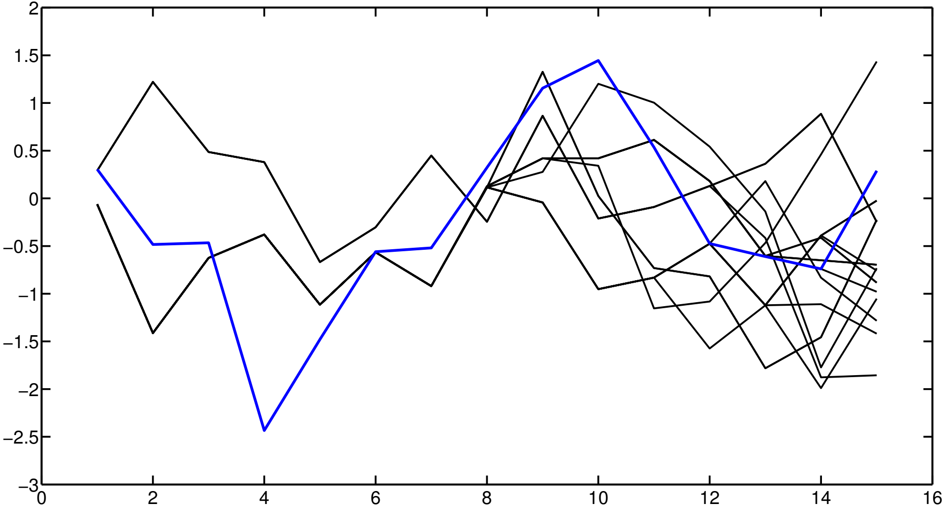
and

$$\widetilde{\psi}^M_{n,L}\left(\widetilde{\mathbf{a}}^{\ominus k}_{n-L+2:n}, \widetilde{\mathbf{x}}^{\ominus k}_{n-L+1:n}; x_{n-L} \,\Big|\Big|\, \widetilde{b}^k_{n-L+1:n-1}, k, \widetilde{x}^k_{n-L+1:n}\right)$$
$$= \frac{\psi^M_{n,L}(\widetilde{\mathbf{a}}_{n-L+2:n}, \widetilde{\mathbf{x}}_{n-L+1:n}, k; x_{n-L})}{q\left(\widetilde{x}^{\widetilde{b}^k_{n,n-L+1}}_{n-L+1} | x_{n-L}\right) \left[\prod_{p=n-L+2}^{n} r\left(\widetilde{b}^k_{n,p} | \widetilde{\mathbf{w}}_{\mathbf{p-1}}\right) q\left(\widetilde{x}^{\widetilde{b}^k_{n,p}}_p | \widetilde{x}^{\widetilde{b}^n_{n,p-1}}_{p-1}\right)\right] r(k|\widetilde{\mathbf{w}}_n)}$$

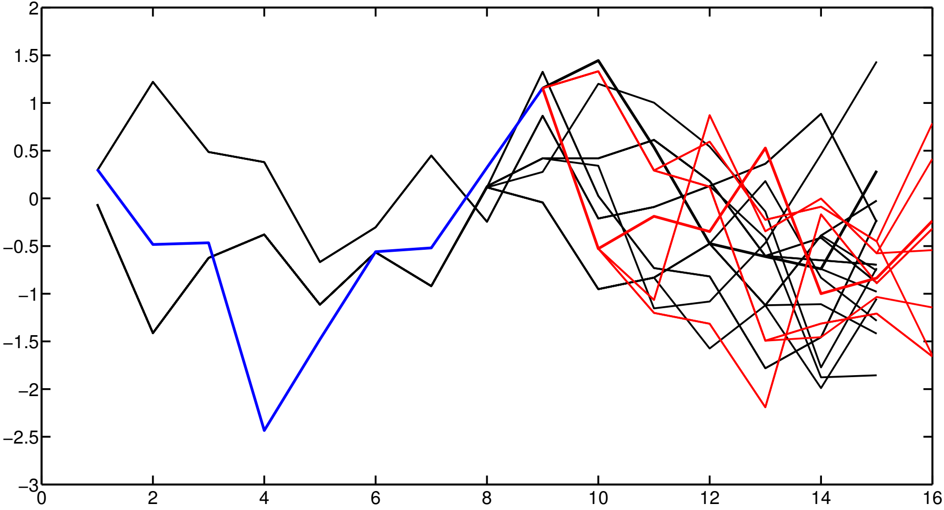
Local Particle Filtering: Current Trajectories



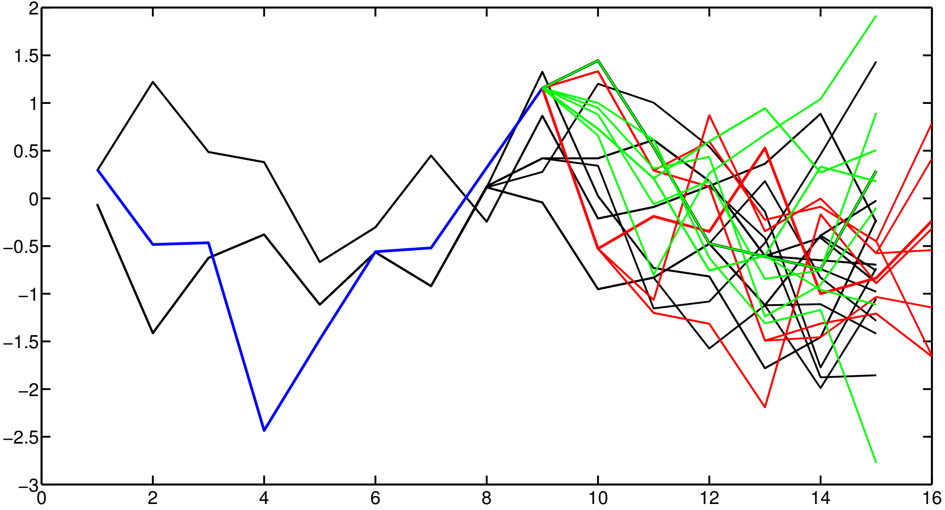
Local Particle Filtering: First Particle



Local Particle Filtering: SMC Proposal



Local Particle Filtering: CSMC Auxiliary Proposal



Local SMC

- Propose from:

$$\mathcal{U}_{1:M}^{\otimes n-1}(b_{1:n-2}, k) p(x_{1:n-1} | y_{1:n-1}) \psi_{n,L}^{M}(\overline{\mathbf{a}}_{n-L+2:n}, \overline{\mathbf{x}}_{n-L+1:n}, \overline{k}; x_{n-L})$$
$$\widetilde{\psi}_{n-1,L-1}^{M}\left(\widetilde{\mathbf{a}}_{n-L+2:n-1}^{\ominus k}, \widetilde{\mathbf{x}}_{n-L+1:n-1}^{\ominus k}; x_{n-L} \,||\, b_{n-L+2:n-1}, x_{n-L+1:n-1}\right)$$

- Target:

$$\mathcal{U}_{1:M}^{\otimes n}(b_{1:n-L}, \overline{b}_{n,n-L+1:n-1}^{\bar{k}}, \bar{k}) p(x_{1:n-L}, \overline{x}_{n-L+1:n}^{\overline{b}_{n,n-L+1:n}^{\bar{k}}} | y_{1:n})$$
$$\widetilde{\psi}_{n,L}^{M}\left(\overline{\mathbf{a}}_{n-L+2:n}^{\ominus \overline{k}}, \overline{\mathbf{x}}_{n-L+1:n}^{\ominus \overline{k}}; x_{n-L} \,\middle|\middle|\, \overline{b}_{n,n-L+1:n}^{\bar{k}}, \overline{x}_{n-L+1:n}^{\overline{b}_{n,n-L+1:n}^{\bar{k}}}\right)$$
$$\psi_{n-1,L-1}^{M}\left(\widetilde{\mathbf{a}}_{n-L+2:n-1}, \widetilde{\mathbf{x}}_{n-L+1:n-1}, k; x_{n-L}\right).$$

- Weight: $\bar{Z}_{n-L+1:n} / \widetilde{Z}_{n-L+1:n-1}$.

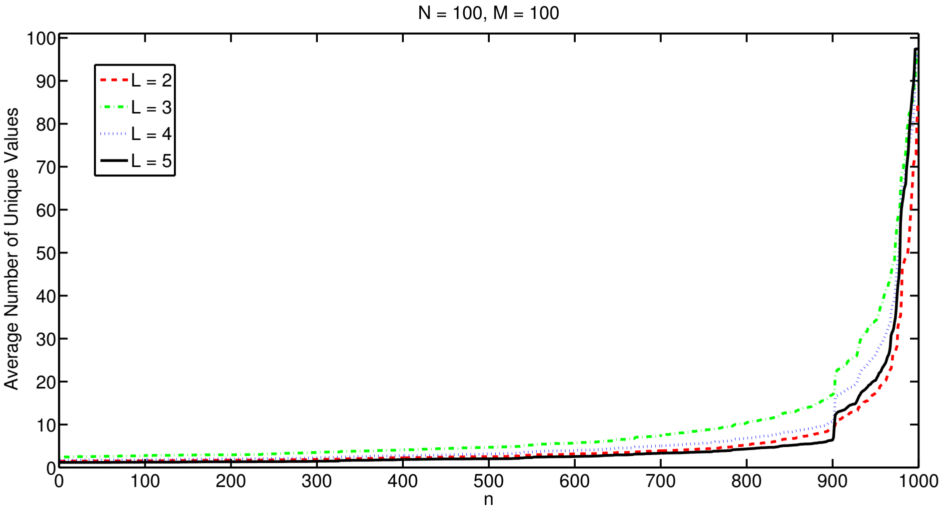
Key Identity

$$
\begin{aligned}
&\frac{\psi^M_{n,L}(\mathbf{a}_{n-L+2:n}, \mathbf{x}_{n-L+1:n}, k; x_{n-L})}{p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})\widetilde{\psi}^M_{n,L}(\mathbf{a}^{\ominus k}_{n-L+2:n}, \mathbf{x}^{\ominus k}_{n-L+1:n}, k; x_{n-L}\,||\ldots\ )} \\
=&\frac{q\left(x^{b^k_{n,n-L+1}}_{n-L+1}|x_{n-L}\right)\left[\prod_{p=n-L+2}^{n} r\left(b^k_{n,p}|\mathbf{w_{p-1}}\right) q\left(x^{b^k_{n,p}}_{p}|x^{b^n_{n,p-1}}_{p-1}\right)\right] r(k|\mathbf{w}_n)}{p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})} \\
=&\widehat{Z}_{n-L+1:n}/p(y_{n-L+1:n}|x_{n-L})
\end{aligned}
$$

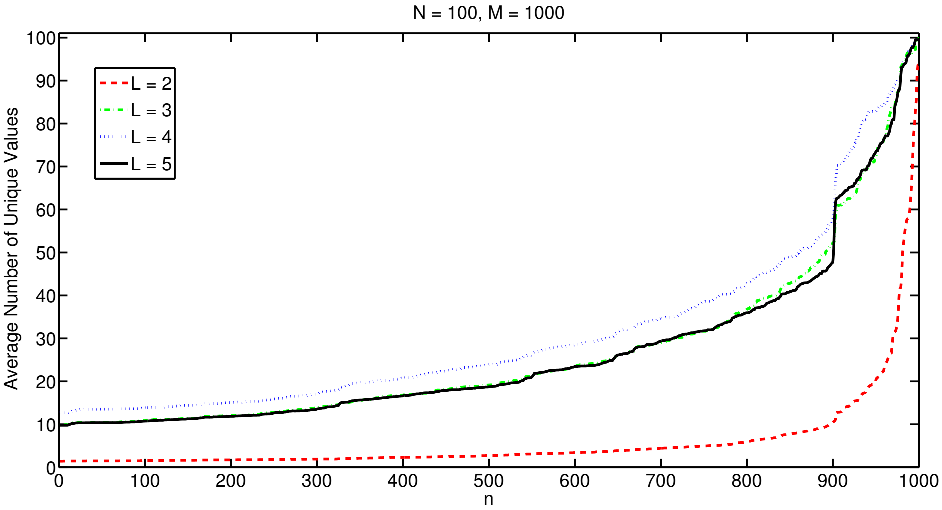
Bootstrap Local SMC

- Top Level:
  - Local SMC proposal.
  - Stratified resampling when ESS$< N/2$.
- Local SMC Proposal:
  - Proposal:
    $$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$
  - Weighting:
    $$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$
  - Resample multinomially every iteration.

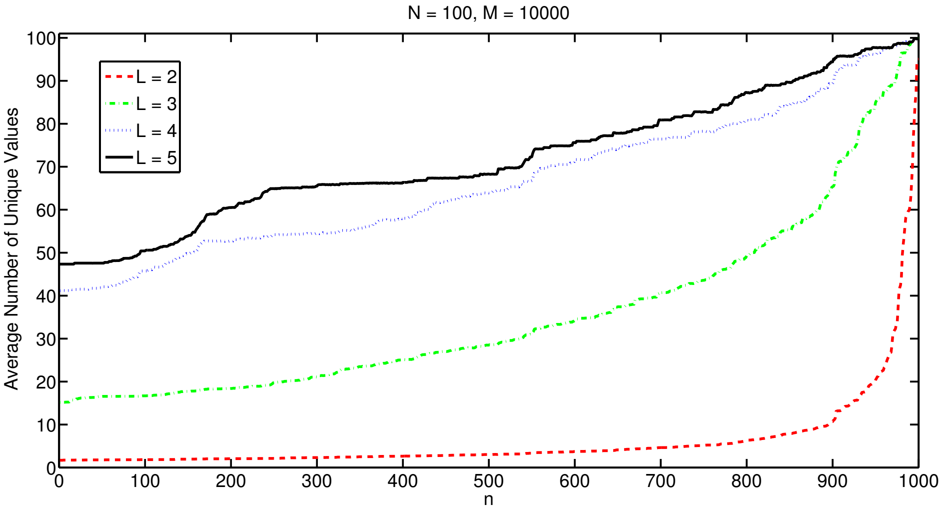
Bootstrap Local SMC: M=100



Bootstrap Local SMC: M=1000



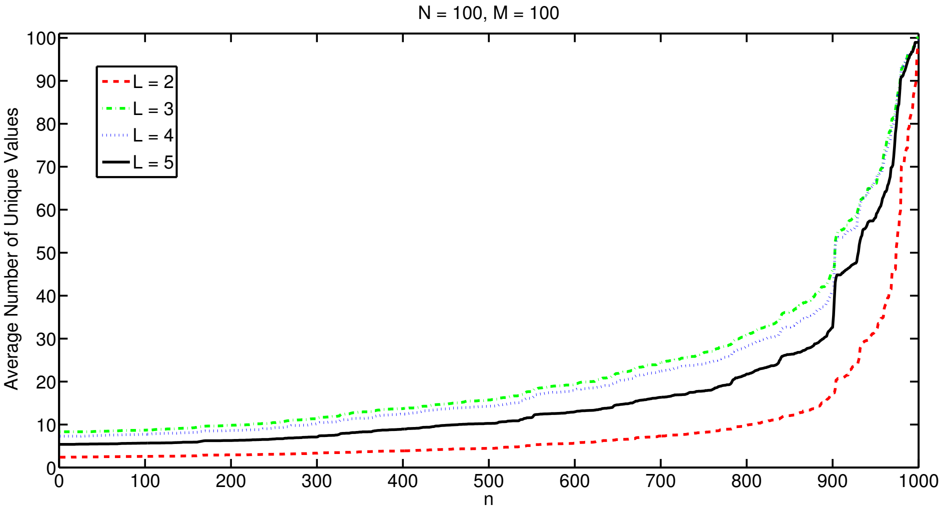
Bootstrap Local SMC: M=10000



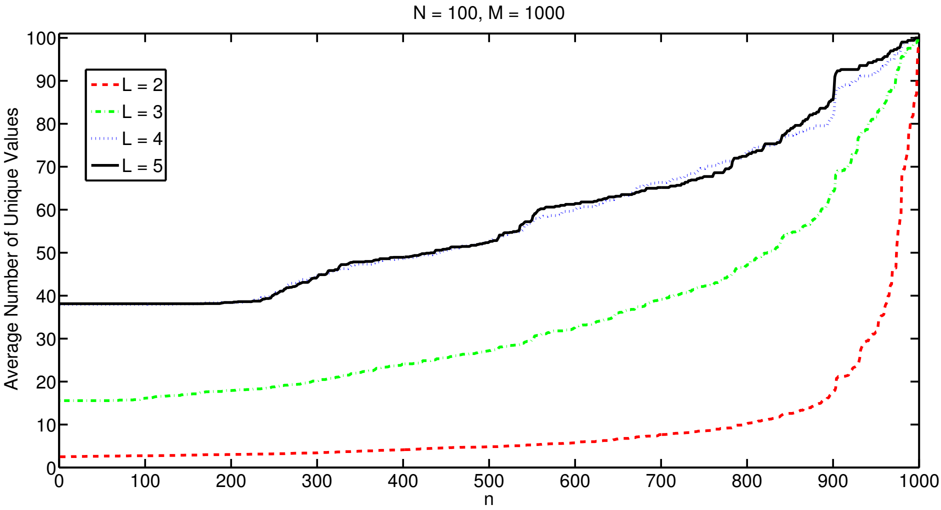
Tuned Local SMC

- Top Level:
  - Local SMC proposal.
  - Stratified resampling when ESS$< N/2$.
- Local SMC Proposal:
  - Proposal:
  $$q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, y_t)$$
  - Weighting:
  $$W(x_{t-1}, x_t) \propto p(y_t|x_{t-1})$$
  - Resample residually every iteration.

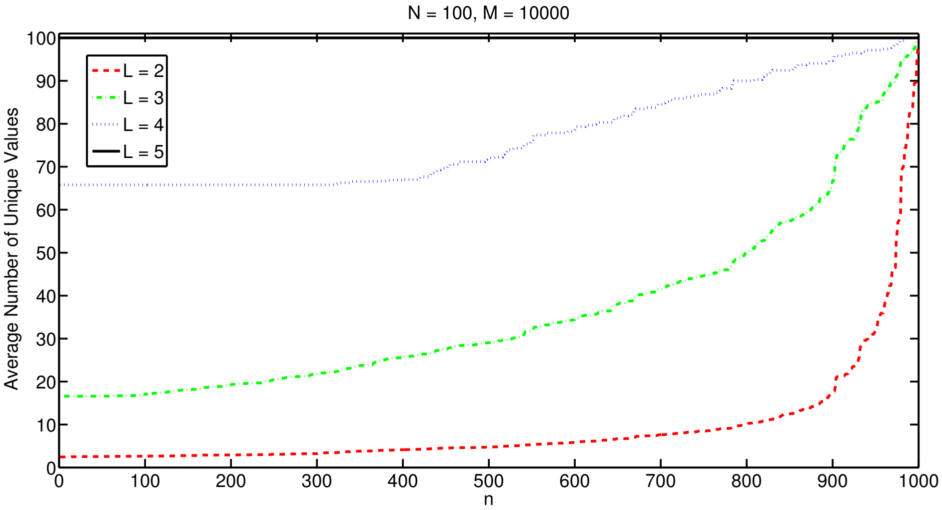
Tuned Local SMC: M=100



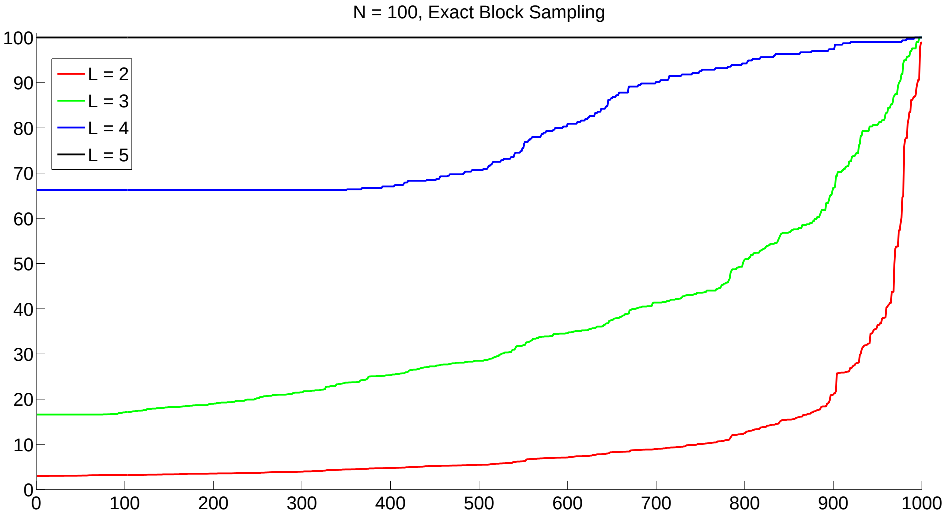
Tuned Local SMC: M=1000



Tuned Local SMC: M=10000



Optimal Block Sampling



Stochastic Volatility Bootstrap Local SMC

- Model:

$$\begin{aligned} f(x_i|x_{i-1}) =& \mathcal{N}\left(\phi x_{i-1}, \sigma^2\right) \\ g(y_i|x_i) =& \mathcal{N}\left(0, \beta^2 \exp(x_i)\right) \end{aligned}$$

- Top Level:
  - Local SMC proposal.
  - Stratified resampling when ESS$< N/2$.
- Local SMC Proposal:
  - Proposal:

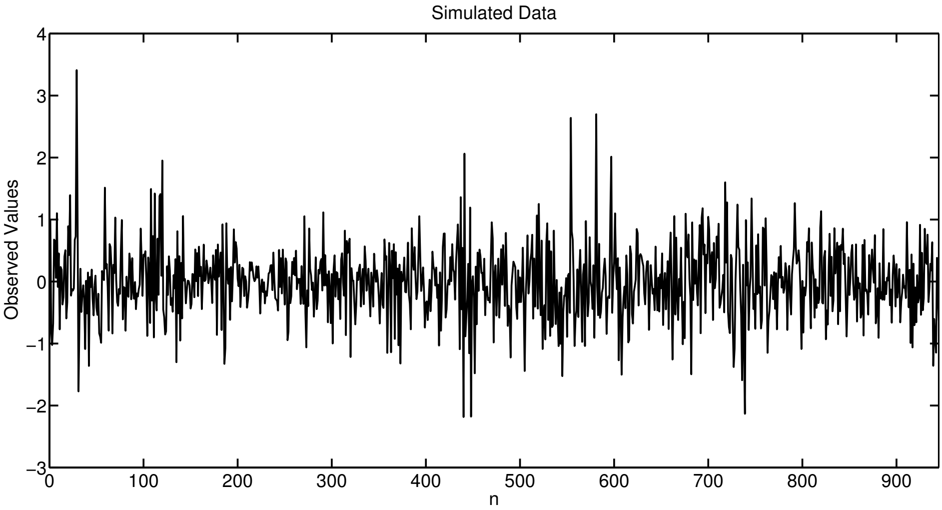
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

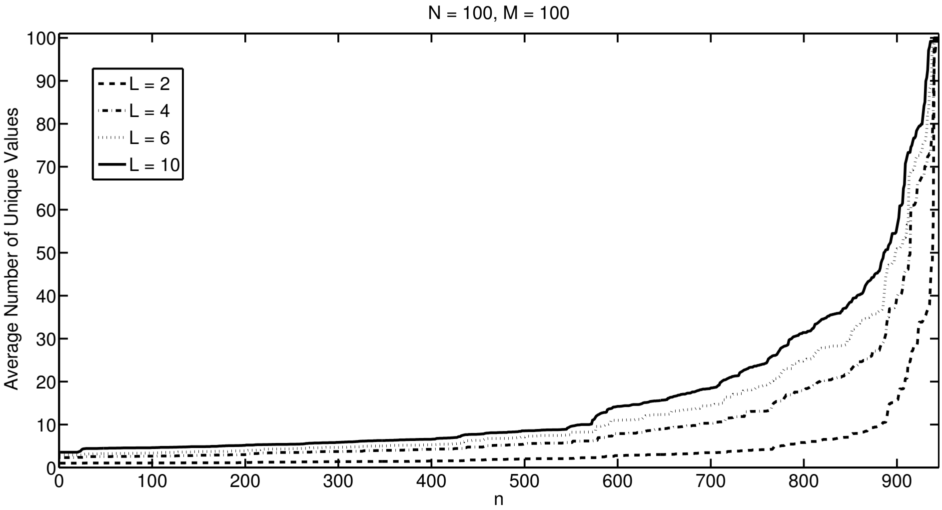
$$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$

  - Resample residually every iteration.

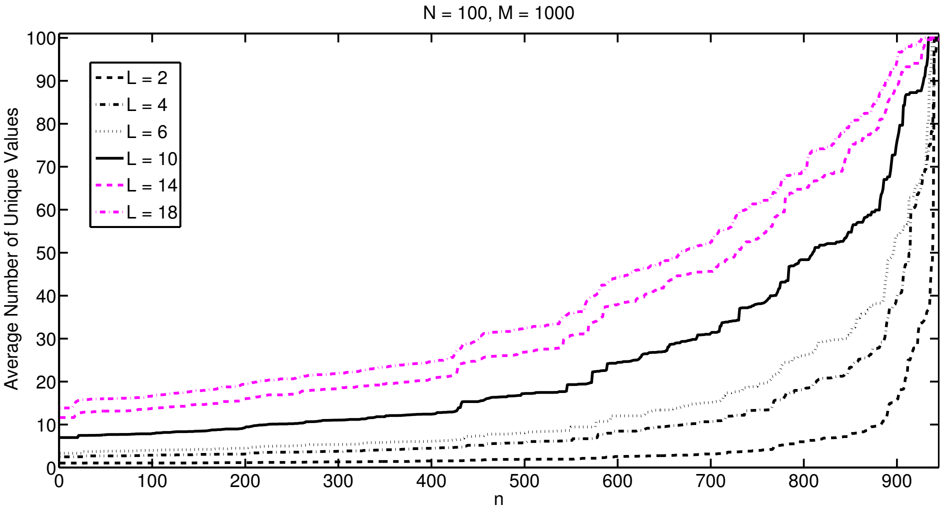
SV Simulated Data



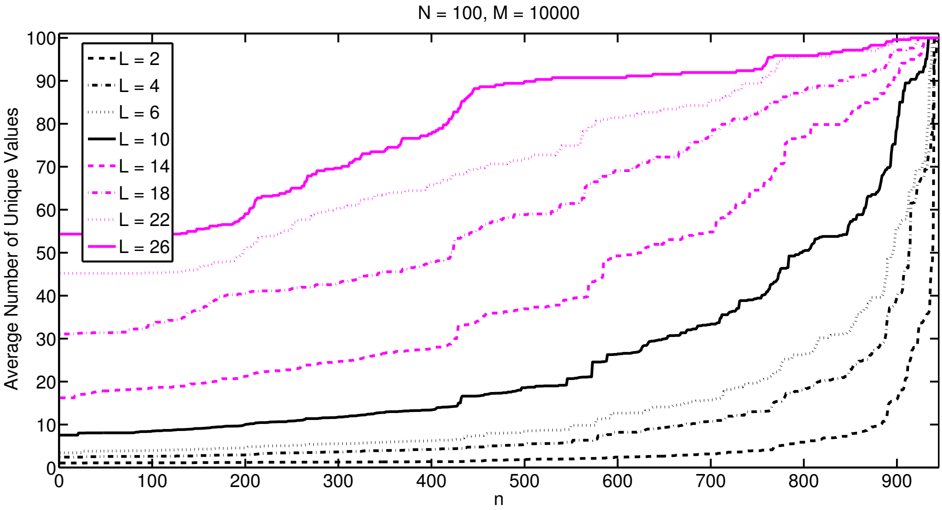
SV Bootstrap Local SMC: M=100



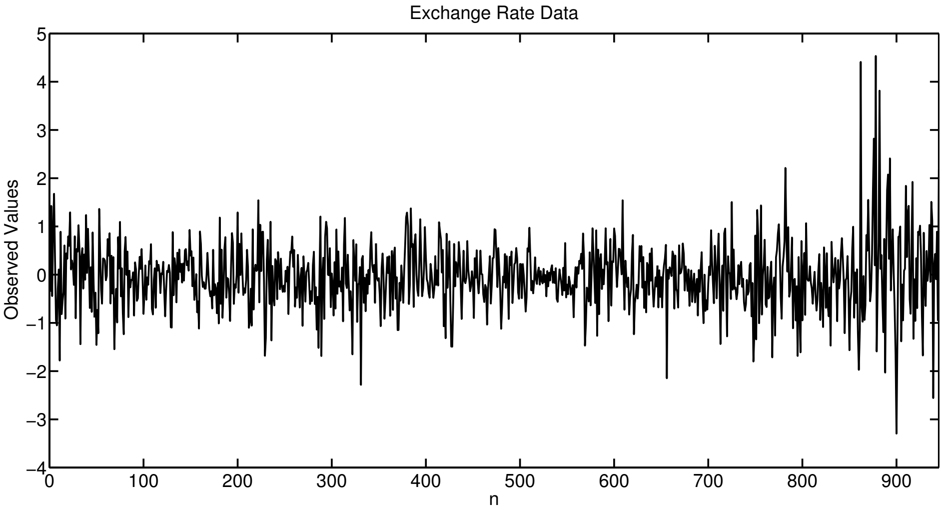
SV Bootstrap Local SMC: M=1000



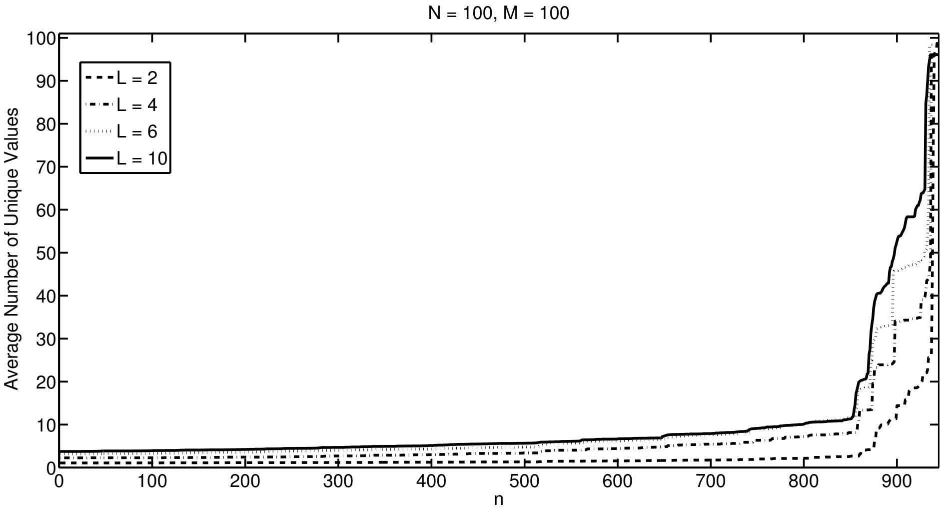
SV Bootstrap Local SMC: M=10000



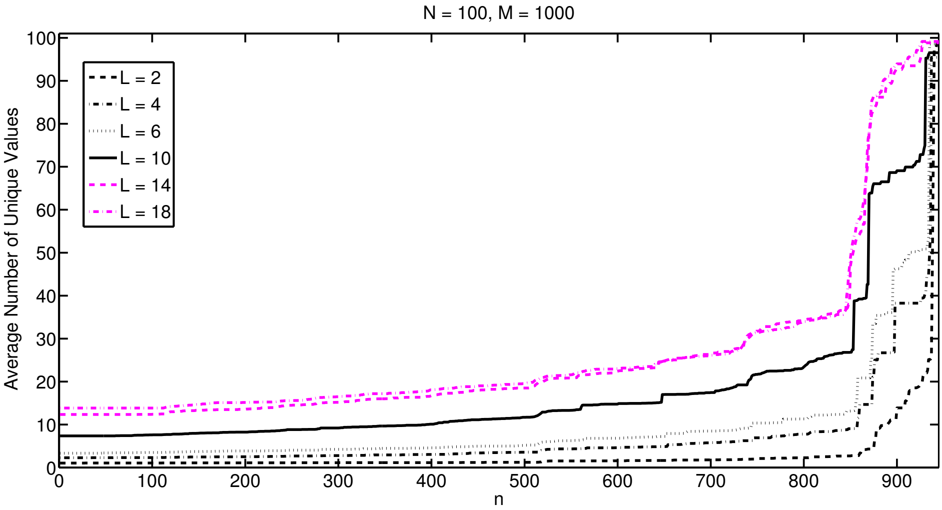
SV Exchange Rata Data



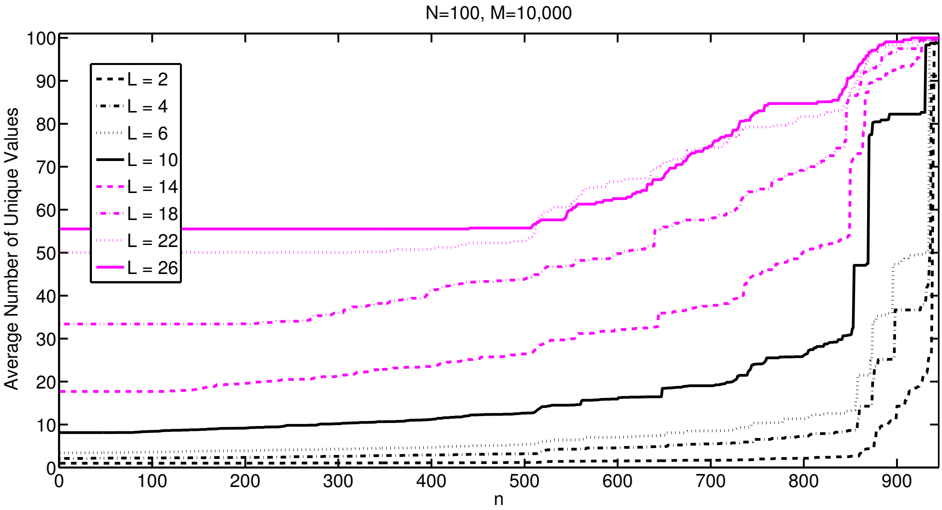
SV Bootstrap Local SMC: M=100



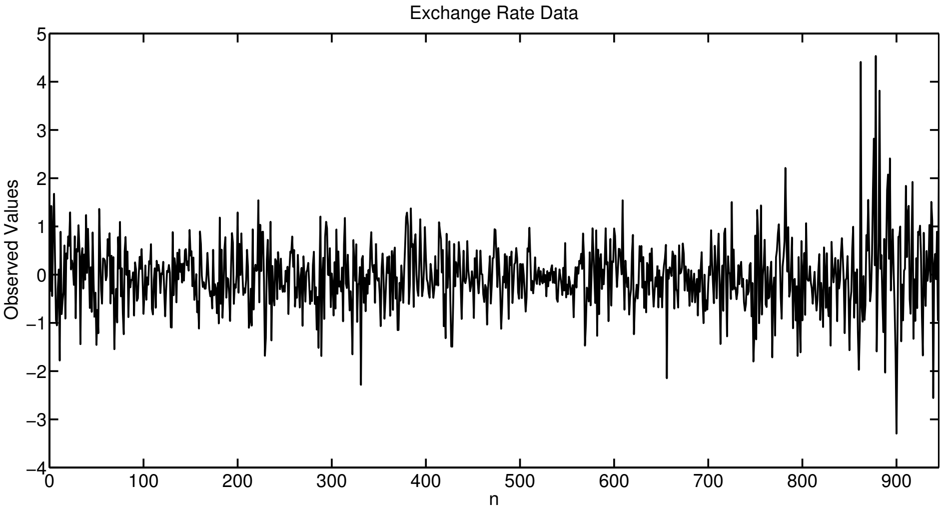
SV Bootstrap Local SMC: M=1000



SV Bootstrap Local SMC: M=10000



SV Exchange Rata Data



In Conclusion

- SMC can be used hierarchically.
- Software implementation is not difficult [6].
- Optimal block sampling can be approximated well:
  - Little specific tuning is *required*.
  - Minimizes need for resampling.
  - Robustness to outliers.
- The computational cost of this strategy is rather high.
- Parallel implementations are natural.

- Actually, similar techniques apply elsewhere [3, 2].
- And we can *approximately Rao-Blackwellise* [5].

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo. *Journal of the Royal Statistical Society B*, 72(3): 269–342, 2010.

[2] N. Chopin, P. Jacob, and O. Papaspiliopoulos. SMC^2. *ArXiv Mathematics e-prints*, 1101.1528, January 2011.

[3] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B*, 63(3): 411–436, 2006.

[4] A. Doucet, M. Briers, and S. Sénécal. Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711, 2006.

[5] A. M. Johansen, N. P. Whiteley and A. Doucet. Exact Approximation of Rao-Blackwellized Particle Filters *Submitted.*

[6] A. M. Johansen. SMCTC: Sequential Monte Carlo in C++. *Journal of Statistical Software*, 30(6):1–41, April 2009.