Monte Carlo Approximation of Monte Carlo Filters

Adam M. Johansen[⋆], Arnaud Doucet[‡] and Nick Whiteley[†]

[⋆]University of Warwick Department of Statistics,
[†]University of Bristol School of Mathematics,
[‡]University of Oxford Department of Statistics

a.m.johansen@warwick.ac.uk
http://go.warwick.ac.uk/amjohansen/talks

January 8th, 2013

Context & Outline

Filtering in State-Space Models:

- SIR Particle Filters [GSS93]
- Rao-Blackwellized Particle Filters [AD02, CL00]
- Block-Sampling Particle Filters [DBS06]

Exact Approximation of Monte Carlo Algorithms:

- Particle MCMC [ADH10]

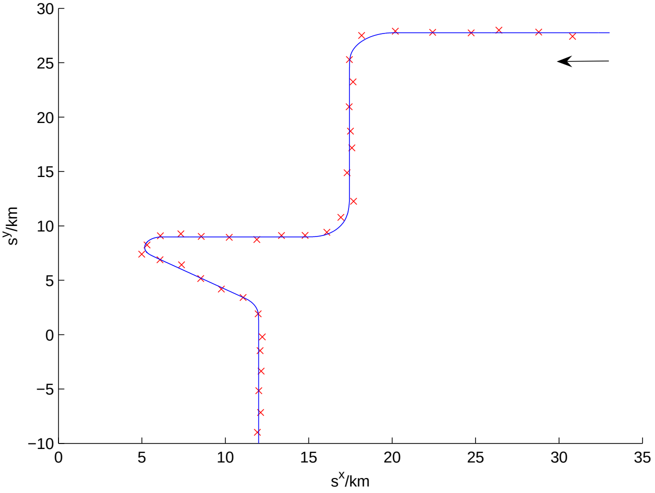
Approximating the RBPF

- Approximated Rao-Blackwellized Particle Filters [CSOL11]
- Exactly-approximated RBPFs [JWD12]

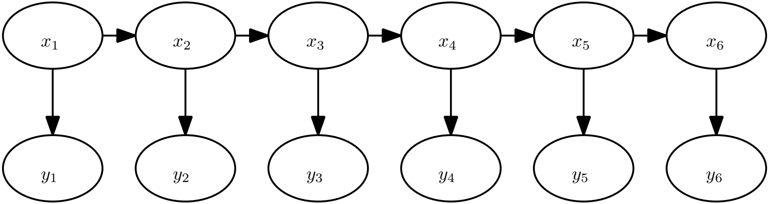
Approximating the BSPF

- Local SMC [JD13]

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.
- 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: $x_n = Ax_{n-1} + \epsilon_n$
  - Observation: $y_n = Bx_n + \nu_n$
- Stochastic Volatility, e.g.:
  - States: x_n is latent volatility.
  - Dynamics: $f(x_i|x_{i-1}) = \mathcal{N}\left(\phi x_{i-1}, \sigma^2\right)$
  - Observations: $g(y_i|x_i) = \mathcal{N}\left(0, \beta^2 \exp(x_i)\right)$

Solutions

- Formally:

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

- 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:

$$w_n(x_{1:n}) \propto \frac{p(x_{1:n}|y_{1:n})}{q_n(x_{1:n})} \propto w_{n-1}(x_{1:n-1}) \frac{f(x_n|x_{n-1}) g(y_n|x_n)}{q_n(x_n|x_{n-1})}$$

Sequential Importance Sampling – Prediction & Update

- A first "particle filter":
  - Simple default: $q_n(x_n|x_{n-1}) = f(x_n|x_{n-1})$.
  - Importance weighting becomes:

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

- Algorithmically, at iteration n:
  - Given $\{W_{n-1}^i, X_{1:n-1}^i\}$ 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)*

Resampling

- Simplest approach:

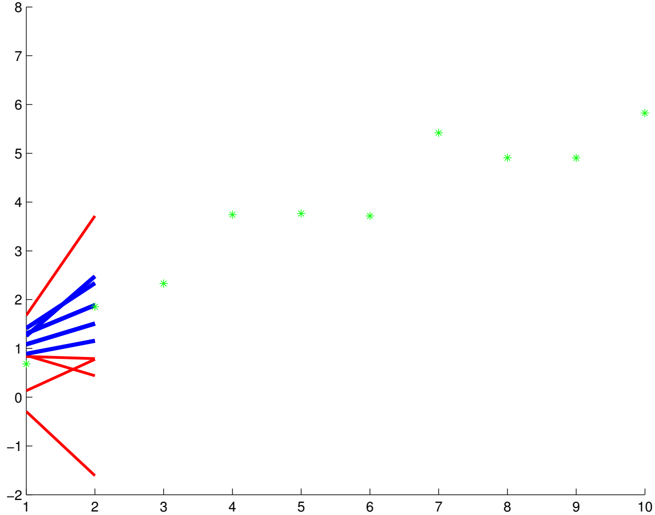
$$\widetilde{X}_n^i \overset{\text{iid}}{\sim} \sum_{j=1}^{n} W_n^j \delta_{X_n^j}$$

- Replace $\{W_n^i, X_n^i\}_{i=1}^N$ with $\{\frac{1}{N}, \widetilde{X}_n^i\}_{i=1}^N$.
- Lower variance options preferable.

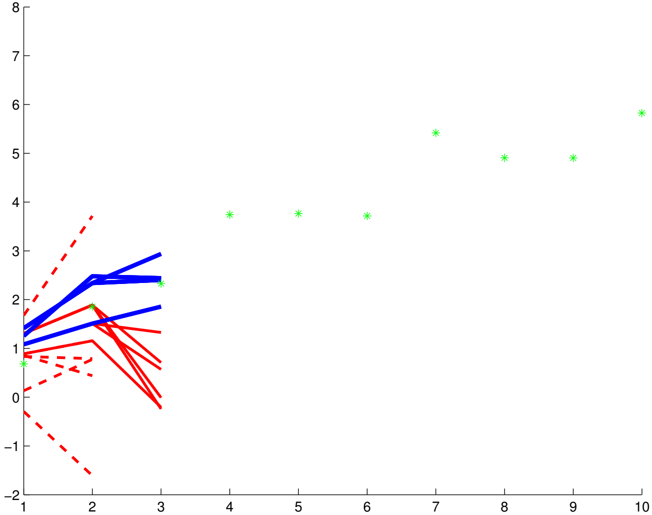
Algorithmically, at iteration n:

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

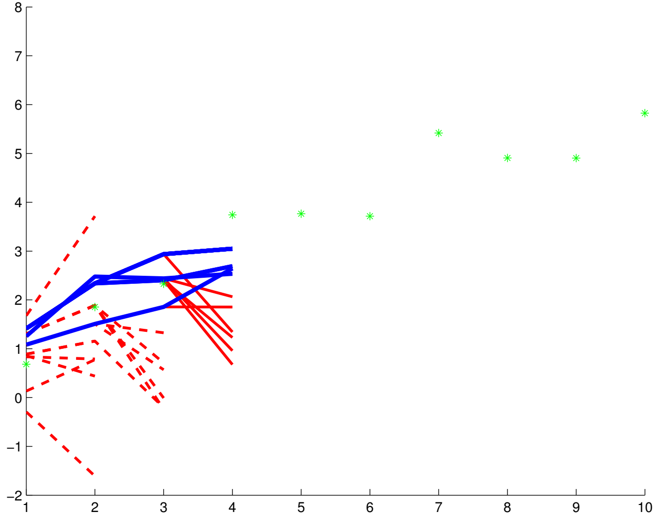
Iteration 2



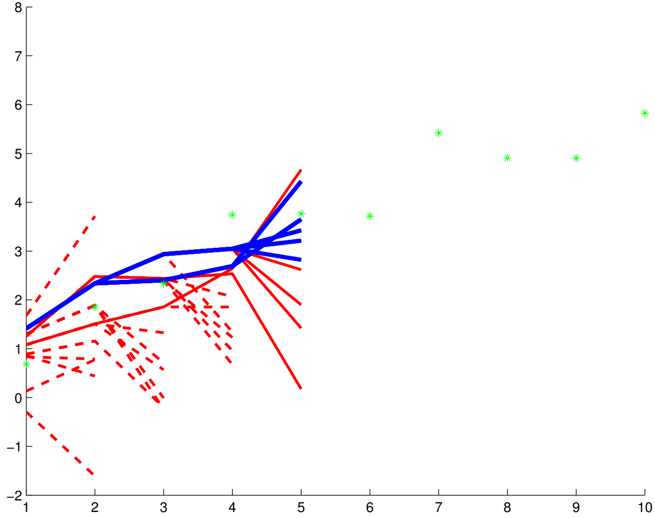
Iteration 3



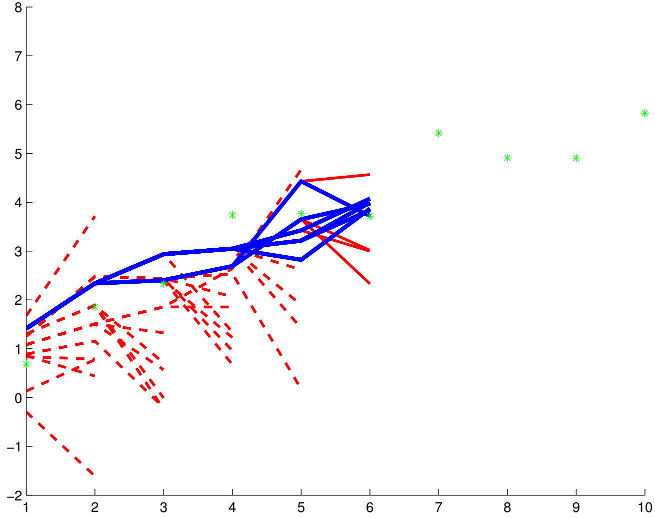
Iteration 4



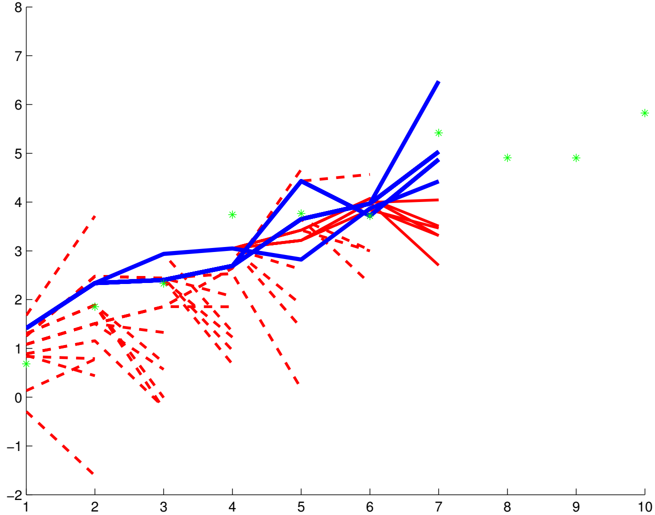
Iteration 5



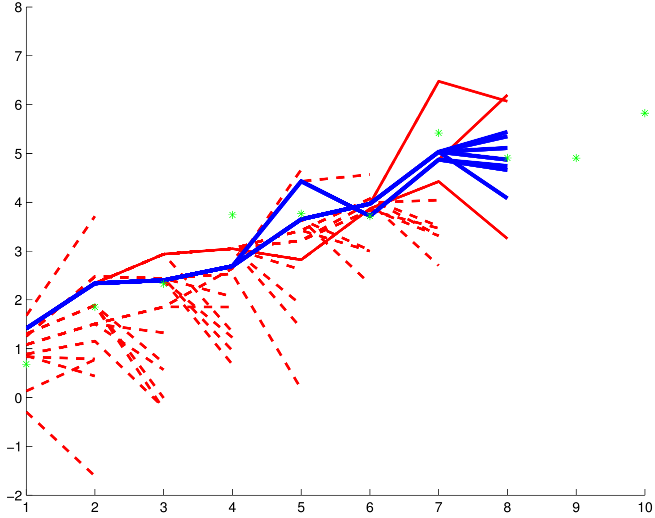
Iteration 6



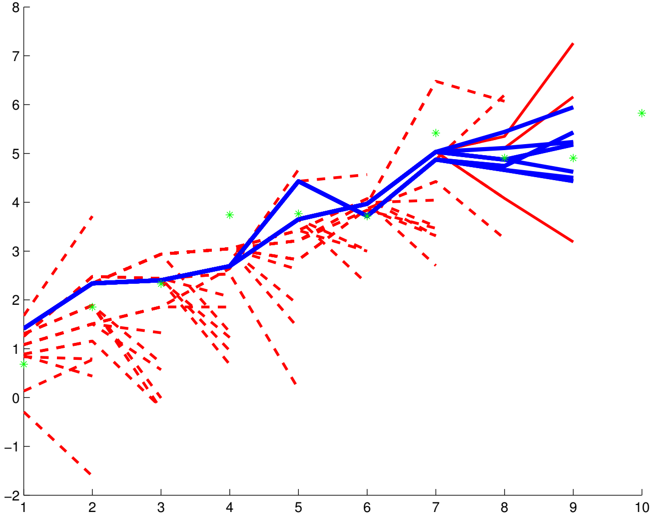
Iteration 7



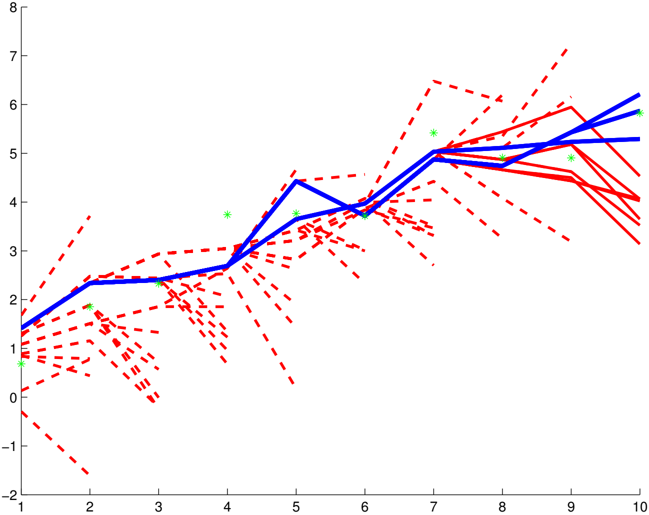
Iteration 8



Iteration 9

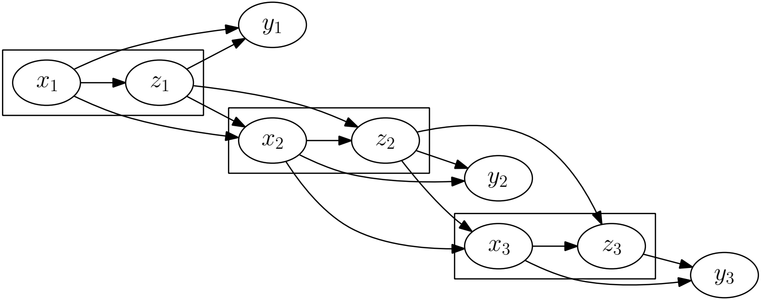


Iteration 10



Rao-Blackwellised Particle Filters

A (Rather Broad) Class of Hidden Markov Models



- Unobserved Markov chain $\{(X_n, Z_n)\}$ transition f.
- Observed process $\{Y_n\}$ conditional density g.
- Density:

$$p(x_{1:n}, z_{1:n}, y_{1:n}) = f_1(x_1, z_1)g(y_1|x_1, z_1) \prod_{i=2}^{n} f(x_i, z_i|x_{i-1}, z_{i-1})g(y_i|x_i, z_i).$$

Simple Solutions

- Formally:

$$p((x,z)_{1:n}|y_{1:n}) \propto p((x,z)_{1:n-1}|y_{1:n-1})f((x,z)_n|(x,z)_{n-1})g(y_n|(x,z)_n)$$

- An SIR Filter: Algorithmically, at iteration n:
  - Given $\{W_{n-1}^i, (X,Z)_{1:n-1}^i\}$ for $i = 1, \ldots, N$:
  - **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}, \widetilde{Z})_{1:n-1}^i\}$.
    - Sample $(X,Z)_n^i \sim q_n(\cdot|(\widetilde{X}, \widetilde{Z})_{n-1}^i)$
    - Weight $W_n^i \propto \frac{f((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)g(y_n|(X,Z)_n^i)}{q_n((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)}$

A Rao-Blackwellized SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^{X,i}, (X_{1:n-1}^{i}, p(z_{1:n-1}|X_{1:n-1}^{i}, y_{1:n-1})\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}_{1:n-1}^{i}, p(z_{1:n-1}|\widetilde{X}_{1:n-1}^{i}, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^{i})$
  - Set $X_{1:n}^{i} \leftarrow (\widetilde{X}_{1:n-1}^{i}, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{p(X_n^i, y_n|\widetilde{X}_{n-1}^{i})}{q_n(X_n^i|\widetilde{X}_{n-1}^{i})}$
  - Compute $p(z_{1:n}|y_{1:n}, X_{1:n}^{i})$.

Requires analytically tractable substructure.

An Approximate Rao-Blackwellized SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^{X,i}, (X_{1:n-1}^i, \widehat{p}(z_{1:n-1}|X_{1:n-1}^i, y_{1:n-1})\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}_{1:n-1}^i, \widehat{p}(z_{1:n-1}|\widetilde{X}_{1:n-1}^i, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^i)$
  - Set $X_{1:n}^i \leftarrow (\widetilde{X}_{1:n-1}^i, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{\widehat{p}(X_n^i, y_n|\widetilde{X}_{n-1}^i)}{q_n(X_n^i|\widetilde{X}_{n-1}^i)}$
  - Compute $\widehat{p}(z_{1:n}|y_{1:n}, X_{1:n}^i)$.
- Our proposal: use N "local" M-particle filters to provide $\widehat{p}(z_{1:n}|y_{1:n}, X_{1:n}^i)$ *and* $\widehat{p}(X_n^i, y_n|\widetilde{X}_{n-1}^i)$.

Is approximate; how does error accumulate?

How can this be justified?

- As an extended space SIR algorithm.
- Via unbiased estimation arguments.

Note also the $M = 1$ and $M \to \infty$ cases.

How does this differ from CSOL11?

Principally in the *local* weights, benefits including:

- Valid (N-consistent) for all $M \geq 1$ rather than (M, N)-consistent.
- Computational cost $\mathcal{O}(MN)$ rather than $\mathcal{O}(M^2N)$.
- Only requires knowledge of joint behaviour of x or z; doesn't require say $p(x_n|x_{n-1}, z_{n-1})$.

Toy Example: Model

We use a simulated sequence of 100 observations from the model defined by the densities:

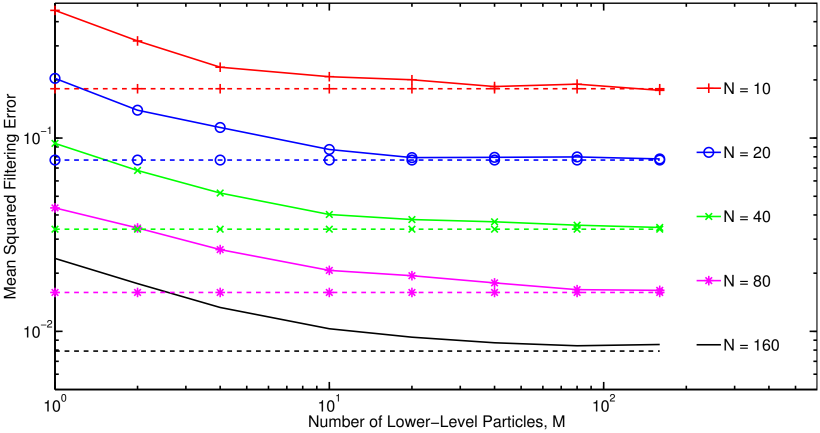
$$\mu(x_1, z_1) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ z_1 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$f(x_n, z_n | x_{n-1}, z_{n-1}) = \mathcal{N}\left(\begin{pmatrix} x_n \\ z_n \end{pmatrix}; \begin{pmatrix} x_{n-1} \\ z_{n-1} \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$g(y_n | x_n, z_n) = \mathcal{N}\left(y_n; \begin{pmatrix} x_n \\ z_n \end{pmatrix}, \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_z^2 \end{bmatrix}\right)$$

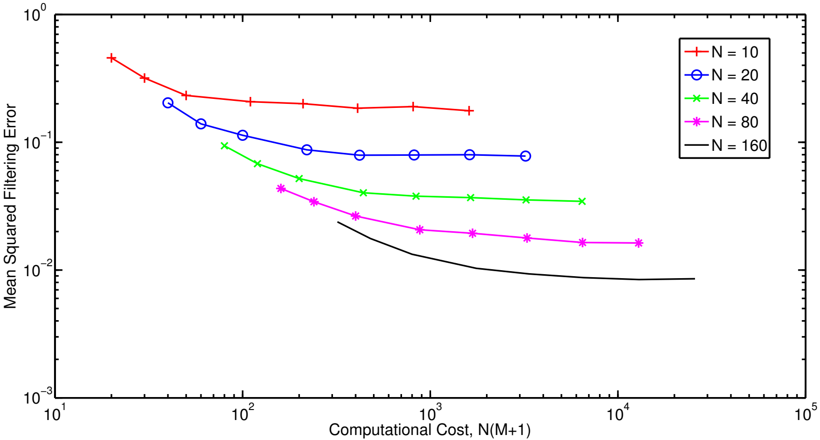
Consider IMSE (relative to optimal filter) of filtering estimate of first coordinate marginals.

Approximation of the RBPF



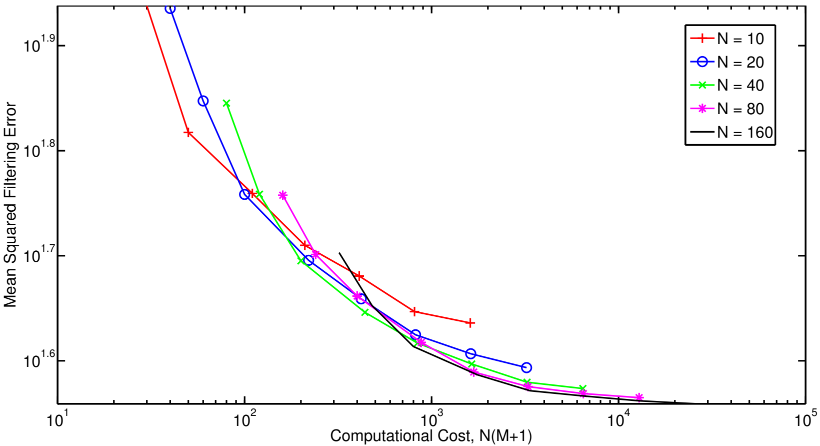
For $\sigma_x^2 = \sigma_z^2 = 1$.

Computational Performance



For $\sigma_x^2 = \sigma_z^2 = 1$.

Computational Performance

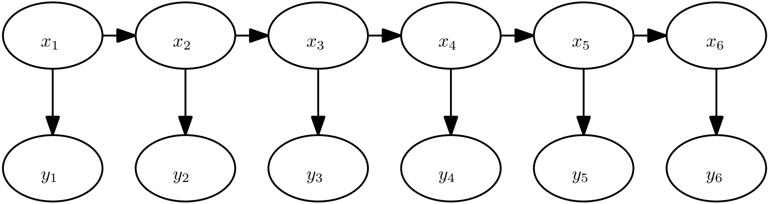


For $\sigma_x^2 = 10^2$ and $\sigma_z^2 = 0.1^2$.

Local Particle Filters

What About Other HMMs / Algorithms?

Returning to:



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

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

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,

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

- Typically intractable; auxiliary variable approach in [DBS06].

Motivating Example

- Model:

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

- Simple Bootstrap SIR Algorithm:
  - Proposal:

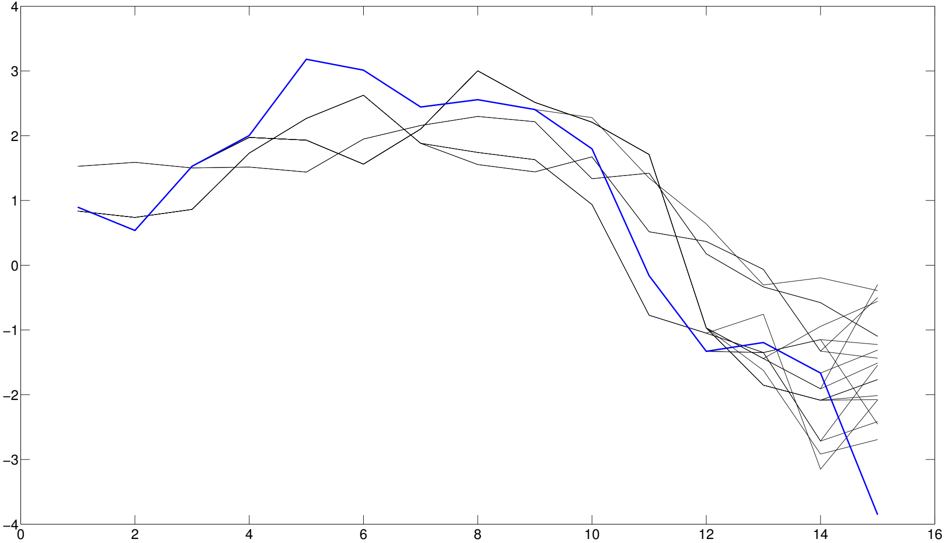
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

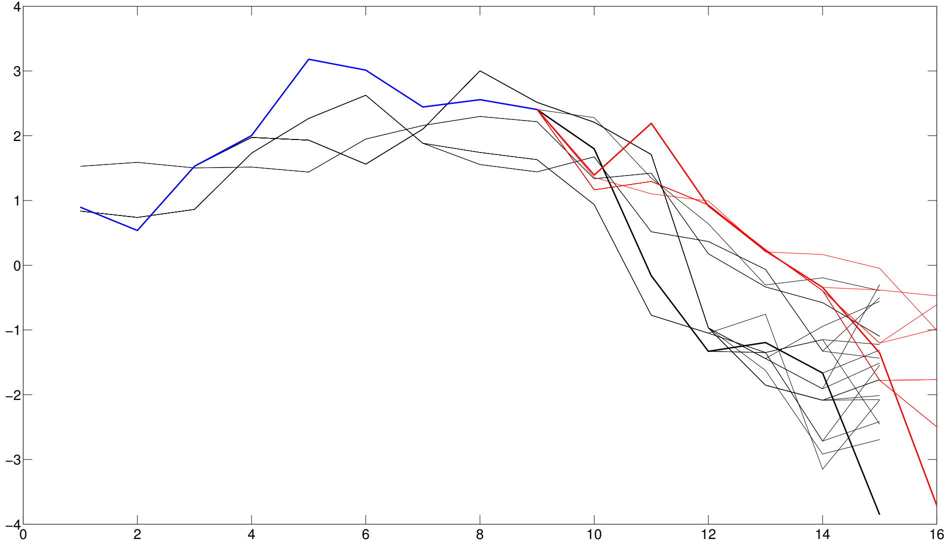
$$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$

  - Resample residually every iteration.
- Is hierarchical SMC possible here?

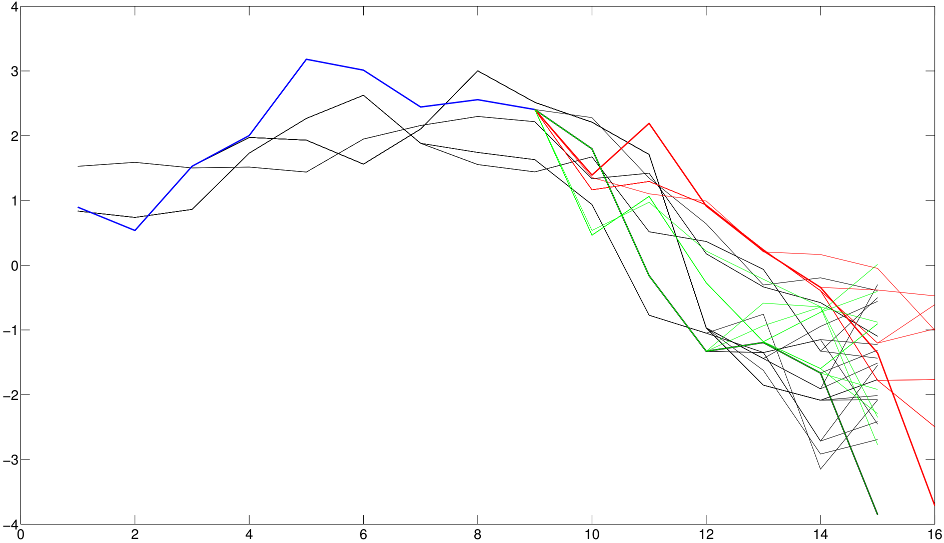
Local Particle Filtering: First Particle



Local Particle Filtering: SMC Proposal



Local Particle Filtering: CSMC Auxiliary Proposal



The Justification of Local SMC

- Propose from:

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

- Target:

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

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

Stochastic Volatility Bootstrap Local SMC

- Model:

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

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

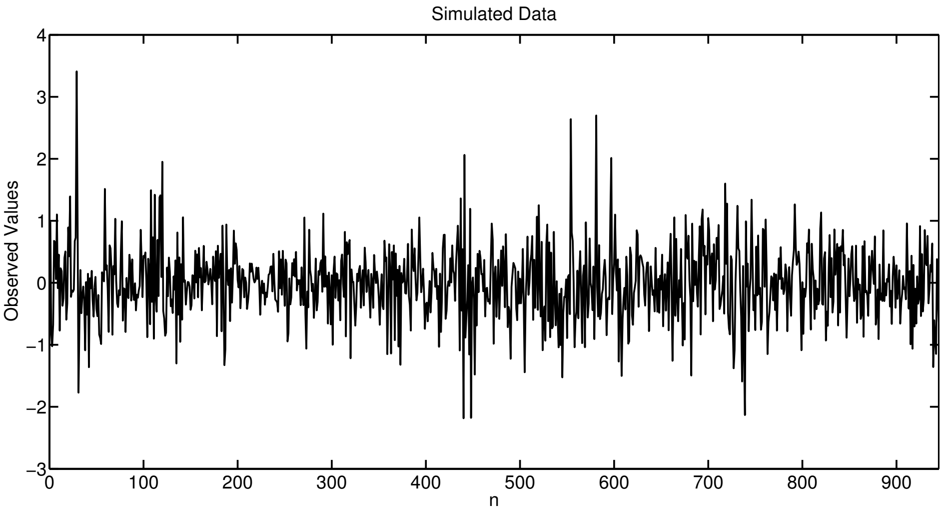
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

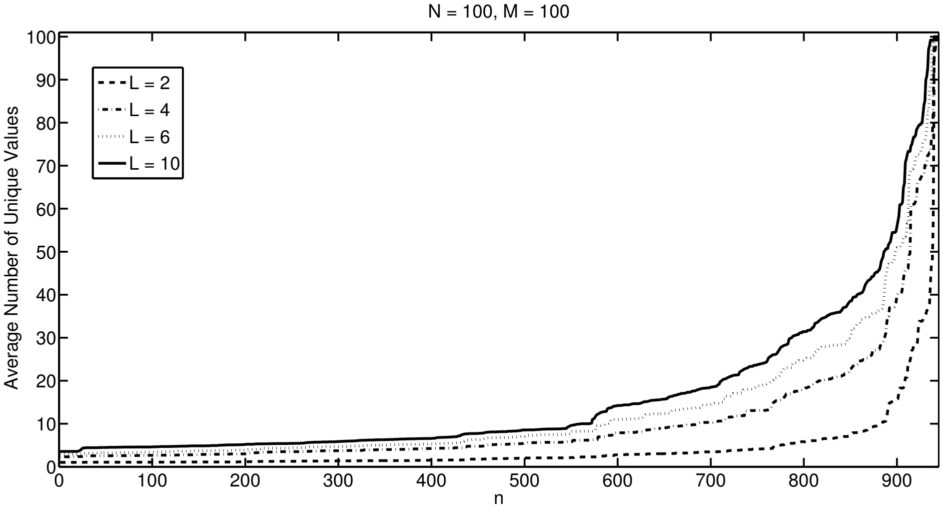
$$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1}) g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$

  - Resample residually every iteration.

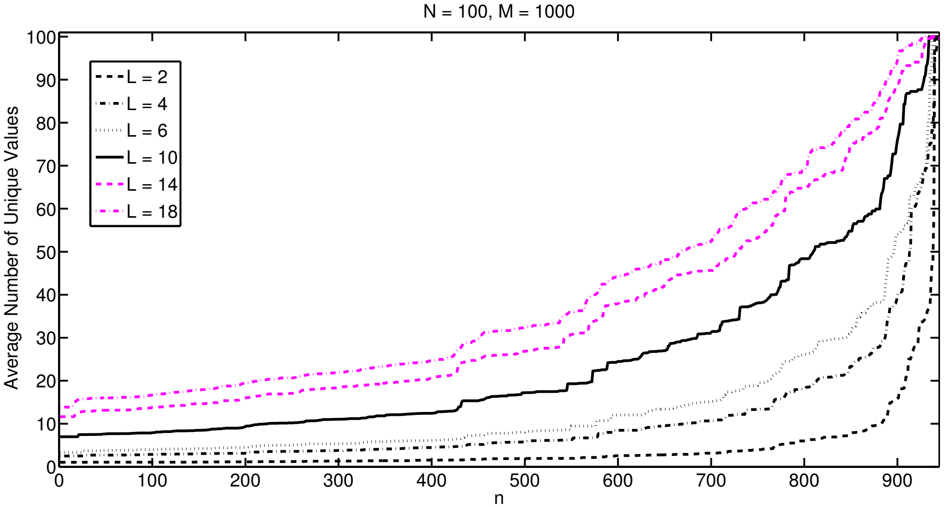
SV Simulated Data



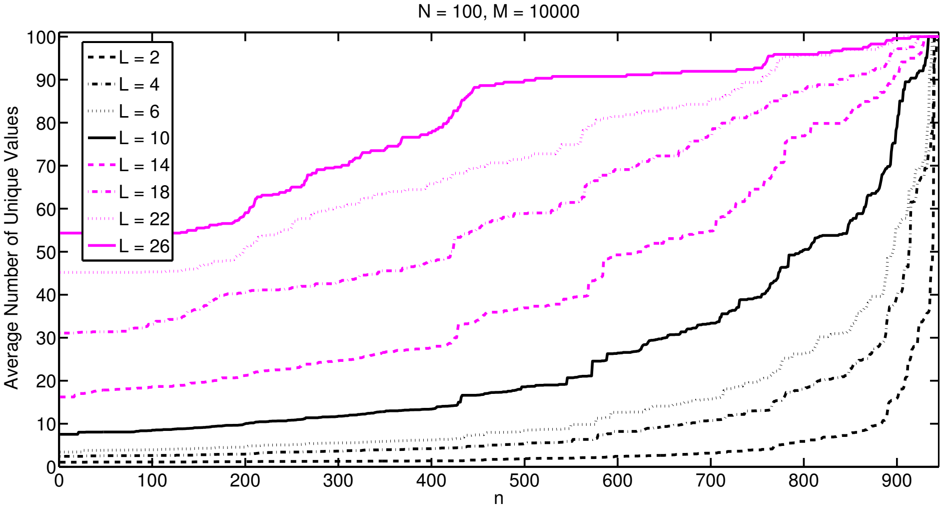
SV Bootstrap Local SMC: M=100



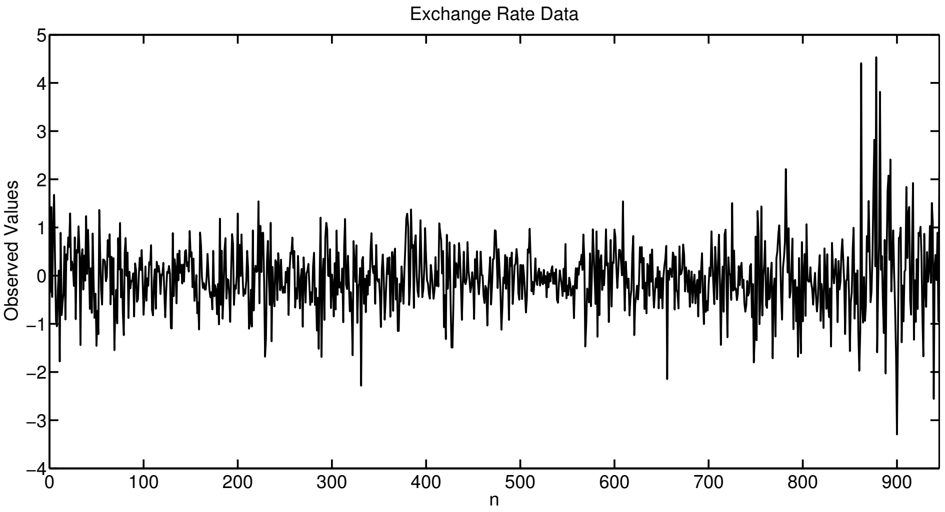
SV Bootstrap Local SMC: M=1000



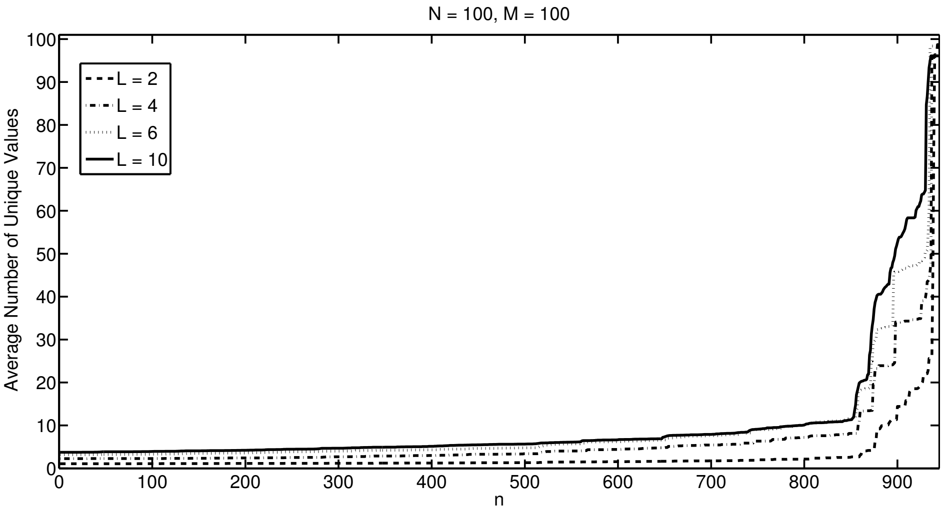
SV Bootstrap Local SMC: M=10000



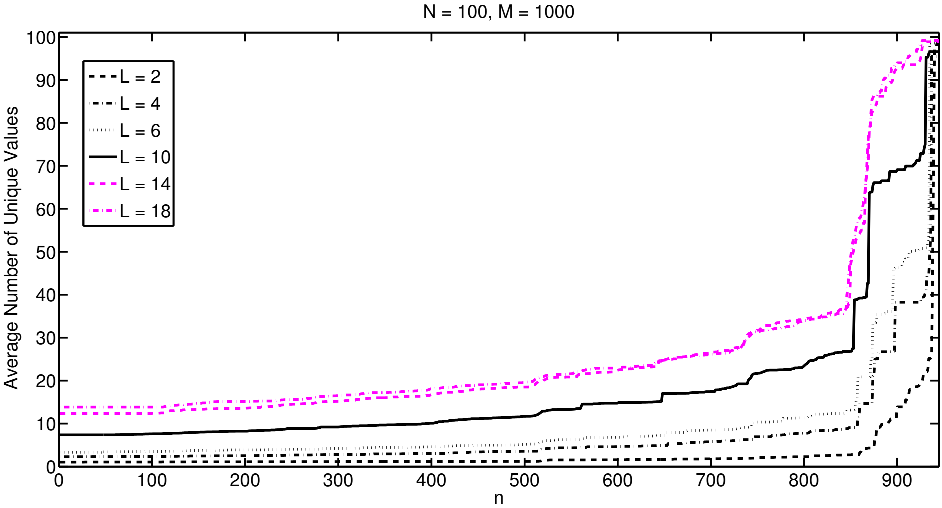
SV Exchange Rata Data



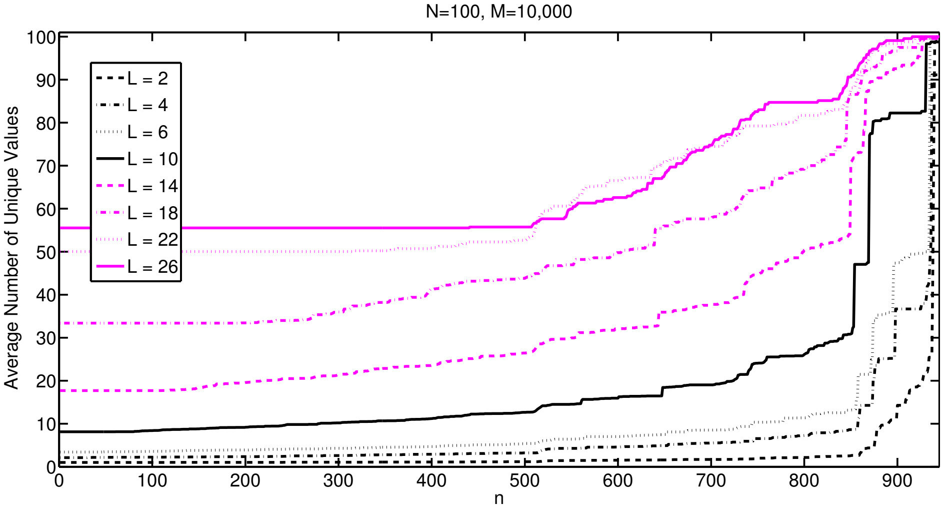
SV Bootstrap Local SMC: M=100



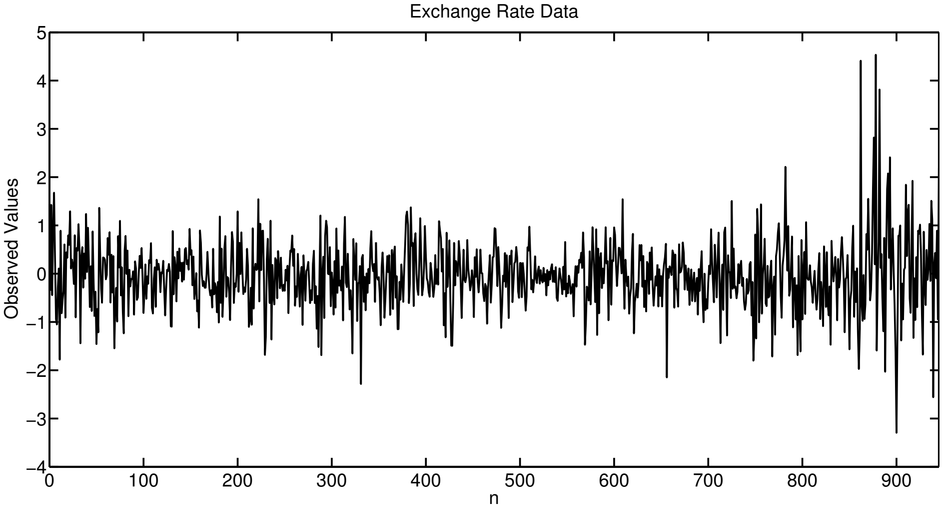
SV Bootstrap Local SMC: M=1000



SV Bootstrap Local SMC: M=10000



SV Exchange Rata Data



In Conclusion

- SMC can be used hierarchically.
- Software implementation is not difficult [Joh09].
- The Rao-Blackwellized particle filter can be approximated *exactly*
  - Can reduce estimator variance at fixed cost.
  - Attractive for distributed/parallel implementation.
  - Allows combination of different sorts of particle filter.
  - Can be combined with other techniques for parameter estimation etc..
- The optimal block-sampling particle filter can be approximated *exactly*
  - Requiring only simulation from the transition and evaluation of the likelihood
  - Easy to parallelise
  - Low storage cost

References I

C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society B*, 64(4):827–836, 2002.

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

R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society B*, 62(3):493–508, 2000.

T. Chen, T. Schön, H. Ohlsson, and L. Ljung. Decentralized particle filter with arbitrary state decomposition. *IEEE Transactions on Signal Processing*, 59(2):465–478, February 2011.

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.

N. J. Gordon, S. 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.

A. M. Johansen and A. Doucet. Hierarchical particle sampling for intractable state-space models. CRiSM working paper, University of Warwick, 2013. In preparation.

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

A. M. Johansen, N. Whiteley, and A. Doucet. Exact approximation of Rao-Blackwellised particle filters. In *Proceedings of 16th IFAC Symposium on Systems Identification*. IFAC, 2012.