Exact Approximation of Rao-Blackwellized Particle Filters

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

[⋆]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

February 21st, 2013
Séminaire BIG'MC, Paris

Context

Filering in State-Space Models:

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

Exact Approximation of Monte Carlo Algorithms:

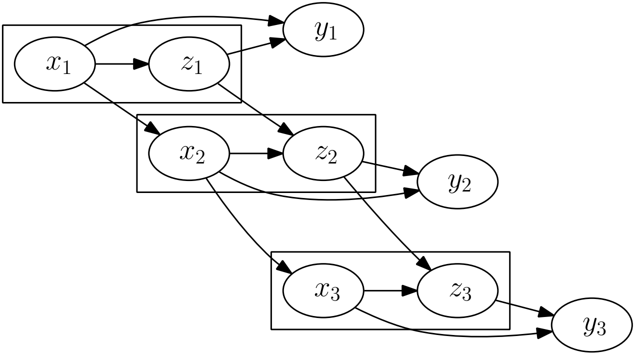
- Particle MCMC [ADH10]
- SMC^2 [CJP11]
- Local SMC [JD13]

Approximating the RBPF

- Approximated Rao-Blackwellized Particle Filters [CSOL11]

This: EA-RBPFs [JWD12]

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:

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

Formal Solutions

- Filtering and Prediction Recursions:

$$p(x_n, z_n|y_{1:n}) = \frac{p(x_n, z_n|y_{1:n-1})g(y_n|x_n, z_n)}{\int p(x_n', z_n'|y_{1:n-1})g(y_n|x_n', z_n')d(x_n', z_n')}$$

$$p(x_{n+1}, z_{n+1}|y_{1:n}) = \int p(x_n, z_n|y_{1:n})f(x_{n+1}, z_{n+1}|x_n, z_n)d(x_n, z_n)$$

- Smoothing:

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

Importance Sampling

- Given q, such that
  - $p(x) > 0 \Rightarrow q(x) > 0$
  - and $p(x)/q(x) < \infty$,

  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 $X_1, \ldots, X_N \stackrel{\text{iid}}{\sim} q$.
  - Estimate $\widehat{I} = \frac{1}{N} \sum_{i=1}^{N} w(X_i)\varphi(X_i)$.
- 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

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

- Estimate the numerator and denominator with the same sample:

$$\widehat{I} = \frac{\sum_{i=1}^{N} v(X_i)\varphi(X_i)}{\sum_{i=1}^{N} v(X_i)}.$$

Importance Sampling in The HMM Setting

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

$$
\begin{aligned}
w_n((x,z)_{1:n}) &\propto \frac{p((x,z)_{1:n}|y_{1:n})}{q_n((x,z)_n|(x,z)_{1:n-1})q_{n-1}((x,z)_{1:n-1})} \\
&= \frac{p((x,z)_{1:n}|y_{1:n})w_{n-1}((x,z)_{1:n-1})}{q_n((x,z)_n|(x,z)_{1:n-1})p((x,z)_{1:n-1}|y_{1:n-1})} \\
&\propto \frac{f((x,z)_n|(x,z)_{n-1})g(y_n|(x,z)_n)}{q_n((x,z)_n|(x,z)_{n-1})} w_{n-1}((x,z)_{1:n-1})
\end{aligned}
$$

Sequential Importance Sampling – Prediction & Update

- A first "particle filter":
  - Simple default: $q_n((x,z)_n|(x,z)_{n-1}) = f((x,z)_n|(x,z)_{n-1})$.
  - Importance weighting becomes:

$$w_n((x,z)_{1:n}) = w_{n-1}((x,z)_{1:n-1}) \times g(y_n|(x,z)_n)$$

- Algorithmically, at iteration n:
  - Given $\{W_{n-1}^i, (X,Z)_{1:n-1}^i\}$ for $i = 1, \ldots, N$:
    - Sample $(X,Z)_n^i \sim f(\cdot|(X,Z)_{n-1}^i)$ *(prediction)*
    - Weight $W_n^i \propto W_{n-1}^i g(y_n|(X,Z)_n^i)$ *(update)*

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

A Simple 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 $W_n^i \propto \frac{f((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)g(y_n|(X,Z)_n^i)}{q_n((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)}$

Actually:

- Resample efficiently.
- Only resample when necessary.
- ...

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 $\widehat{p}(z_{1:n}|y_{1:n}, X_{1:n}^i)$.

Is approximate; how does error accumulate?

Exactly Approximated Rao-Blackwellized SIR Filter

At time $n = 1$

- Sample, $X_1^i \sim q^x\left(\cdot\right| y_1)$.
- Sample, $Z_1^{i,j} \sim q^z\left(\cdot\middle| X_1^i, y_1\right)$.
- Compute and normalise the local weights

$$w_1^z\left(X_1^i, Z_1^{i,j}\right) := \frac{p(X_1^i, y_1, Z_1^{i,j})}{q^z\left(Z_1^{i,j}\middle| X_1^i, y_1\right)}, W_1^{z,i,j} := \frac{w_1^z\left(X_1^i, Z_1^{i,j}\right)}{\sum_{k=1}^{M} w_1^z\left(X_1^i, Z_1^{i,k}\right)},$$

$$\text{define } \widehat{p}(X_1^i, y_1) := \frac{1}{M}\sum_{j=1}^{M} w_1^z\left(X_1^i, Z_1^{i,j}\right).$$

- Compute and normalise the top-level weights

$$w_1^x\left(X_1^i\right) := \frac{\widehat{p}(X_1^i, y_1)}{q^x\left(X_1^i | y_1\right)},\ W_1^{x,i} := \frac{w_1^x\left(X_1^i\right)}{\sum_{k=1}^{N} w_1^x\left(X_1^k\right)}.$$

At times $n \geq 2$

- Resample

$$\left\{ W_{n-1}^{x,i}, \left(X_{1:n-1}^{i}, \left\{ W_{n-1}^{z,i,j}, Z_{1:n-1}^{i,j} \right\}_j \right) \right\}_i$$

to obtain

$$\left\{ \frac{1}{N}, \left(\widetilde{X}_{1:n-1}^{i}, \left\{ \overline{W}_{n-1}^{z,i,j}, \overline{Z}_{1:n-1}^{i,j} \right\}_j \right) \right\}_i .$$

- Resample $\{\overline{W}_{n-1}^{z,i,j}, \overline{Z}_{1:n-1}^{i,j}\}_j$ to obtain $\{\frac{1}{M}, \widetilde{Z}_{1:n-1}^{i,j}\}_j$.
- Sample $X_n^i \sim q^x(\cdot|\widetilde{X}_{1:n-1}^i, y_{1:n})$; set $X_{1:n}^i := (\widetilde{X}_{1:n-1}^i, X_n^i)$.
- Sample $Z_n^{i,j} \sim q^z\left(\cdot|\, X_{1:n}^i, y_{1:n}, \widetilde{Z}_{1:n-1}^{i,j}\right)$; set $Z_{1:n}^{i,j} := (\widetilde{Z}_{1:n-1}^{i,j}, Z_n^{i,j})$.

- Compute and normalise the local weights

$$w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right) := \frac{p\left(X_n^i, y_n, Z_n^{i,j}\,\middle|\,\widetilde{X}_{n-1}^i, \widetilde{Z}_{n-1}^{i,j}\right)}{q^z\left(Z_n^{i,j}\,\middle|\,X_{1:n}^i, y_{1:n}, \widetilde{Z}_{1:n-1}^{i,j}\right)},$$

$$\widehat{p}(X_n^i, y_n |\, \widetilde{X}_{1:n-1}^i, y_{1:n-1}) : \;=\; \frac{1}{M}\sum_{j=1}^{M} w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right),$$

$$W_n^{z,i,j} : \;=\; \frac{w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right)}{\sum_{k=1}^{M} w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,k}\right)}.$$

- Compute and normalise the top-level weights

$$w_n^x\left(X_{1:n}^i\right) := \frac{\widehat{p}(X_n^i, y_n |\, \widetilde{X}_{1:n-1}^i, y_{1:n-1})}{q^x(X_n^i | \widetilde{X}_{1:n-1}^i, y_{1:n})},$$

$$W_n^{x,i} := \frac{w_n^x\left(X_{1:n}^i\right)}{\sum_{k=1}^{N} w_n^x\left(X_{1:n}^k\right)}.$$

How can this be justified?

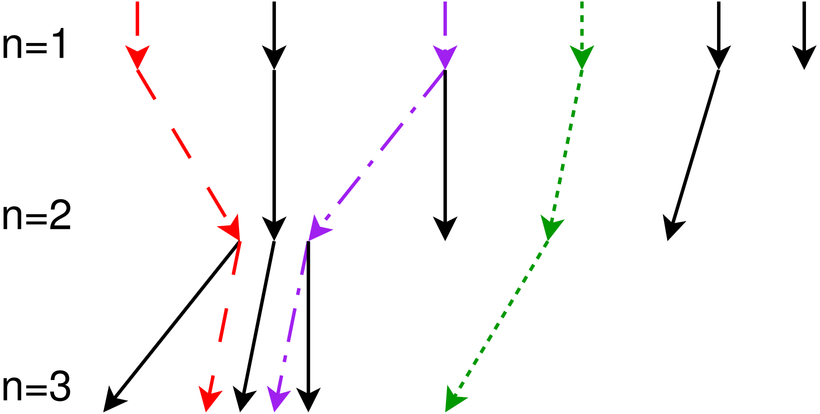
- As an extended space SIR algorithm.
- Via unbiased estimation arguments.

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 $p(x_n|x_{n-1}, z_{n-1})$.



$a_3^1 = 1$ $a_3^4 = 3$ $a_2^1 = 1$ $a_2^4 = 3$

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

Importance Sampling Intepretation – Proposal

Neglecting resampling at the top level, the time n proposal is:

$$q_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| y_{1:n}\right) = q_n\left(x_{1:n} \middle| y_{1:n}\right) q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| x_{1:n}, y_{1:n}\right)$$

where

$$q_n\left(x_{1:n} \middle| y_{1:n}\right) = q^x\left(x_1 \middle| y_1\right) \prod_{m=2}^{n} q^x\left(x_m \middle| x_{1:m-1}, y_{1:m}\right),$$

and assuming *multinomial resampling* locally
$q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| x_{1:n}, y_{1:n}\right)$ becomes

$$\prod_{j=1}^{M} \left\{ q^z\left(z_1^j \middle| x_1, y_1\right) \prod_{m=2}^{n} W_{m-1}^{z,a_{m-1}^{z,j}} q^z\left(z_m^j \middle| x_{1:m}, y_{1:m}, z_{1:m-1}^{a_{m-1}^{z,j}}\right) \right\}.$$

Importance Sampling Interpretation — Weights and Target

The importance weights are:

$$w_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) = \frac{\widehat{p}(x_1, y_1)}{q^x\left(x_1 | y_1\right)} \prod_{m=2}^{n} \frac{\widehat{p}\left(x_m, y_m | x_{1:m-1}, y_{1:m-1}\right)}{q^x(x_m | x_{1:m-1}, y_{1:m})},$$

implying the target distribution

$$\begin{aligned}\pi_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) \quad &\propto \quad w_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) q_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| y_{1:n}\right) \\ &\propto \quad \widehat{p}(x_{1:n}, y_{1:n}) q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| x_{1:n}, y_{1:n}\right)\end{aligned}$$

where

$$\widehat{p}(x_{1:n}, y_{1:n}) := \widehat{p}(x_1, y_1) \prod_{m=2}^{n} \widehat{p}\left(x_m, y_m | x_{1:m-1}, y_{1:m-1}\right)$$

Importance Sampling Interpretation — Marginal Target

It is now well-known that the marginal likelihood estimate provided by a particle filter is unbiased; i.e.

$$\sum_{a_{1:n-1}^{z,1:M}} \int \widehat{p}(x_{1:n}, y_{1:n})\; q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| y_{1:n}, x_{1:n}\right) dz_{1:n}^{1:M} = p\left(x_{1:n}, y_{1:n}\right).$$

It follows straightforwardly that the marginal target is:

$$\pi_n\left(x_{1:n}\right) = \sum_{a_{1:n-1}^{z,1:M}} \int \pi_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) dz_{1:n}^{1:M} = p\left(x_{1:n} \middle| y_{1:n}\right).$$

A Joint Target Distribution I

A little rearrangement yields:

$$\pi_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) = \frac{p(x_{1:n}|y_{1:n})}{M^{n-1}}$$
$$\times \frac{1}{M}\sum_{j=1}^{M} p\left(z_{1:n}^{j}\middle| x_{1:n}, y_{1:n}\right)\left\{\prod_{k=1,k\neq b_1^j}^{M} q^z\left(z_1^k\middle| x_1, y_1\right)\right\}$$
$$\times \prod_{m=2}^{n}\left\{\prod_{k=1,k\neq b_m^j}^{M} W_{m-1}^{a_{m-1}^{z,k}} q^z\left(z_m^k\middle| x_{1:m}, y_{1:m}, z_{1:m-1}^{a_{m-1}^{z,k}}\right)\right\}$$

[Easily verified by division by the joint proposal.]

A Joint Target Distribution II

Introducing an additional discrete random variable $L \in \{1, ..., M\}$ such that

$$\pi_n\left(x_{1:n}, a_{1:n-1}^{1:M}, l, z_{1:n}^{1:M}\right) = \frac{p\left(x_{1:n}, z_{1:n}^{l} \middle| y_{1:n}\right)}{M^n}\left\{\prod_{k=1, k\neq b_1^j}^{M} q^z\left(z_1^k \middle| x_1, y_1\right)\right\} \times \prod_{m=2}^{n}\left\{\prod_{k=1, k\neq b_m^j}^{M} W_{m-1}^{a_{m-1}^k} q^z\left(z_m^k \middle| x_{1:m}, y_{1:m}, z_{1:m-1}^{a_{m-1}^{z,k}}\right)\right\}.$$

This provides use with an estimate of the joint distribution of interest $p\left(x_{1:n}, z_{1:n} \middle| y_{1:n}\right)$.

Toy Example: Model

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

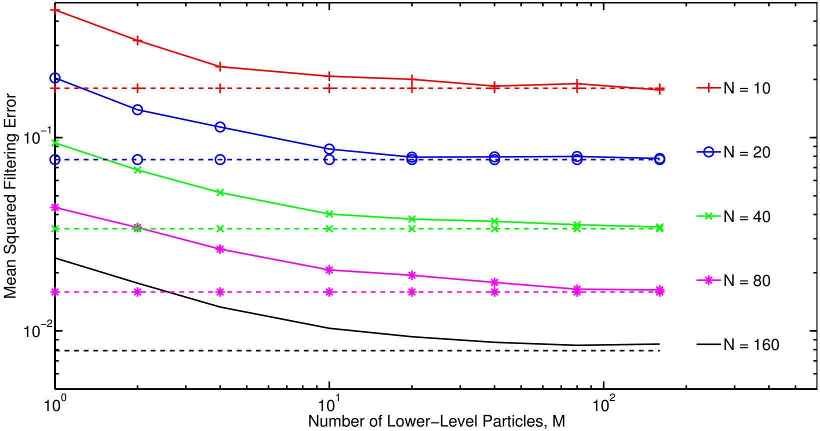
$$\mu(x_1, z_1) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ z_1 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

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

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

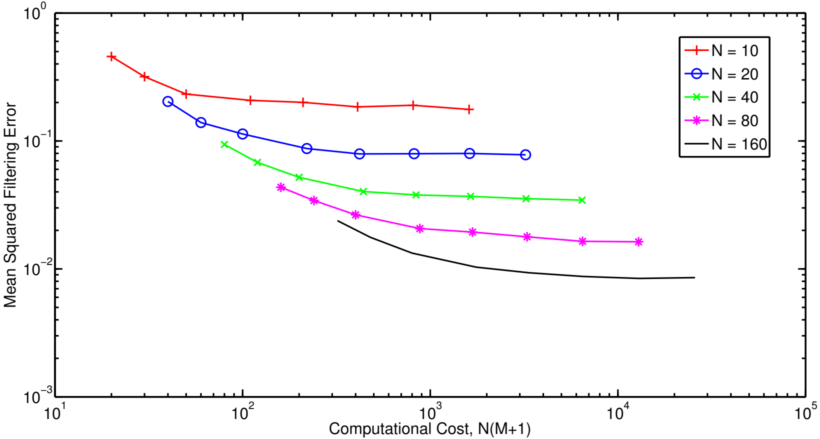
Consider IMSE (relative to optimal filter) of filtering estimate of first coordinate marginals.

Approximation of the RBPF



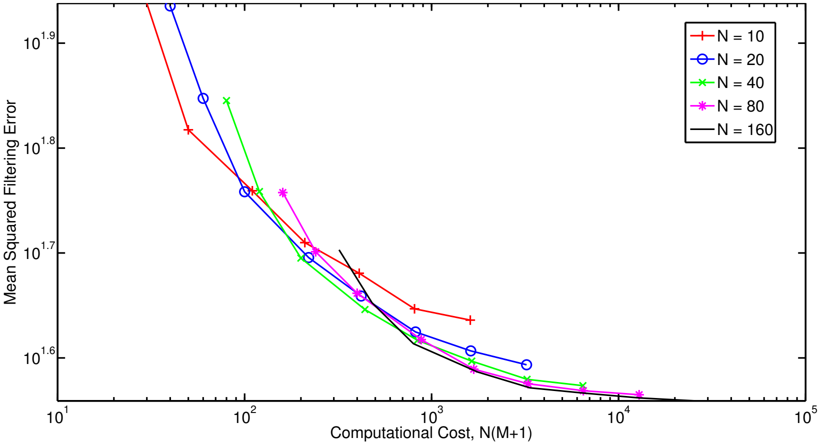
For $\sigma_x^2 = \sigma_z^2 = 1$.

Computational Performance



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

Computational Performance



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

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..

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.

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

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.

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.

References II

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*, Brussels, Belgium, July 2012. IFAC, IFAC.