A Simple Tutorial on RBPF for DBNs

Arnaud Doucet, Nando de Freitas, Kevin Murphy & Stuart Russell

Abstract

This tutorial describes how to apply particle filtering (PF) and Rao Blackwellised particle filtering (RBPF) to a simple dynamic Bayesian network (DBN). Use it in conjunction with our UAI2000 paper.

1 Model

In this paper, we estimate the distribution of the states ($\mathbf{A}_{1:t} \in \{0,1\}^t, \mathbf{B}_{1:t} \in \{0,1\}^t, \mathbf{C}_{1:t} \in \{0,1\}^t$) of the DBN shown in Figure 1. Each node is assumed to have private evidence (observed data), denoted by ($\mathbf{y}^A_{1:t} \in \{0,1\}^t, \mathbf{y}^B_{1:t} \in \{0,1\}^t, \mathbf{y}^C_{1:t} \in \{0,1\}^t$). Such a network is an example of a large family of

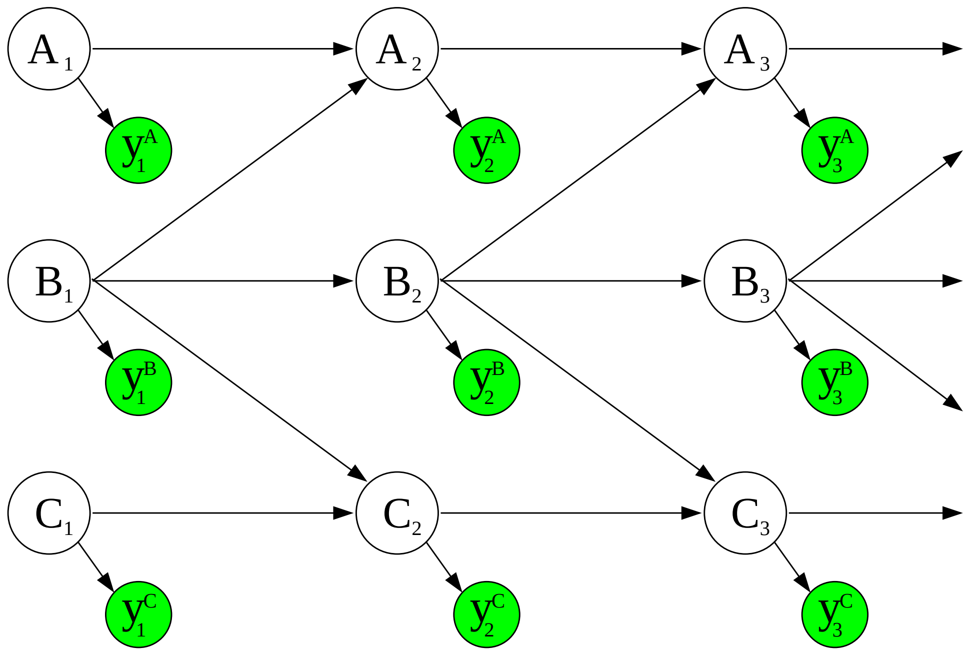

Figure 1: Tree structured DBN, with one root note (B) and two leaves (A, C), for the first 3 time steps.

DBNs, which can be interpreted as trees being propagated over time. The various nodes could correspond to aggregated nodes. That is, node, say, A could correspond to an entire sub-network. Moreover, one could easily have mixtures of continuous and q-ary discrete states and observations. Therefore, *many complex DBNs may be reduced to this architecture.*

Our example network admits the following factorisation

$$\begin{aligned} p(\mathbf{A}_{1:t}, \mathbf{B}_{1:t}, \mathbf{C}_{1:t}|\mathbf{y}_{1:t}) &= p(\mathbf{A}_{1:t}, \mathbf{C}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t})p(\mathbf{B}_{1:t}|\mathbf{y}_{1:t}) \\ &= p(\mathbf{A}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t})p(\mathbf{C}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t})p(\mathbf{B}_{1:t}|\mathbf{y}_{1:t}) \\ &= p(\mathbf{A}_{1:t}|\mathbf{y}^A_{1:t}, \mathbf{B}_{1:t-1})p(\mathbf{C}_{1:t}|\mathbf{y}^C_{1:t}, \mathbf{B}_{1:t-1})p(\mathbf{B}_{1:t}|\mathbf{y}_{1:t}) \qquad (1) \end{aligned}$$

We assume that we can compute exactly the distributions of the leaves, namely $p(\mathbf{A}_{1:t}|\mathbf{y}^A_{1:t}, \mathbf{B}_{1:t-1})$ and $p(\mathbf{C}_{1:t}|\mathbf{y}^C_{1:t}, \mathbf{B}_{1:t-1})$, given the distribution of the root $p(\mathbf{B}_{1:t}|\mathbf{y}_{1:t})$. As a result, we only need to sample the root node.

In the following sections, we will show that the joint filtering distribution $p(A_t, B_t, C_t|\mathbf{y}_{1:t})$ can be computed analytically using the hidden Markov model (HMM) filter, or approximated with particle filters (PFs) and Rao-Blackwellised particle filters (RBPFs). The computational complexity of a brute force analytical method for calculating the filtering density of a network with 3 q-ary states is $O(q^6)$ operations per time step. By converting sums of products to products of sums, the computational complexity of the HMM filter can be reduced by a constant factor as shown in the next section. PFs allow us to use a set of N samples $\{(\mathbf{A}^{(i)}_{1:t}, \mathbf{B}^{(i)}_{1:t}, \mathbf{C}^{(i)}_{1:t}); i = 1, \ldots, N\}$ to approximate the filtering distribution in the q^3 dimensional space. The computational complexity of this method is $O(N)$; that is, it is independent of the dimension of the state space. In high dimensions, this proves to be of great benefit as one cannot apply the exact filtering algorithms. However, particle filters can have high variance. To surmount this problem, while at the same time reducing the computational complexity of exact filters, one can adopt a strategy that combines PFs and exact filters; this is the well known RBPF method. The computational complexity of RBPFs in our example is $O(q^2N)$ multiplications per time step. That is, *RBPFs allow us to decompose large network into smaller sub-networks, thereby allowing one to apply exact filtering algorithms to each of the sub-networks separately.*

2 HMM filter

The HMM filter involves the following two analytical recursive steps at time t (note how the sums of products are replaced by products of sums to reduce the computational complexity)

Prediction: By standard marginalisation

$$\begin{aligned}
p(A_t, B_t, C_t|\mathbf{y}_{1:t-1}) &= \sum_{A_{t-1}} \sum_{B_{t-1}} \sum_{C_{t-1}} p(A_t, B_t, C_t|A_{t-1}, B_{t-1}, C_{t-1}) \\
&\quad \times p(A_{t-1}, B_{t-1}, C_{t-1}|\mathbf{y}_{1:t-1}) \\
&= \sum_{A_{t-1}} \sum_{B_{t-1}} \sum_{C_{t-1}} p(A_t|A_{t-1}, B_{t-1}) p(B_t|B_{t-1}) \\
&\quad \times p(C_t|B_{t-1}, C_{t-1}) p(A_{t-1}, B_{t-1}, C_{t-1}|\mathbf{y}_{1:t-1}) \\
&= \sum_{B_{t-1}} p(B_t|B_{t-1}) \sum_{A_{t-1}} p(A_t|A_{t-1}, B_{t-1}) \\
&\quad \times \sum_{C_{t-1}} p(C_t|B_{t-1}, C_{t-1}) p(A_{t-1}, B_{t-1}, C_{t-1}|\mathbf{y}_{1:t-1})
\end{aligned}$$

Update: Using Bayes' rule

$$\begin{aligned}
p(A_t, B_t, C_t|\mathbf{y}_{1:t}) &= \frac{p(y_t|A_t, B_t, C_t) p(A_t, B_t, C_t|\mathbf{y}_{1:t-1})}{\sum_{A_t} \sum_{B_t} \sum_{C_t} p(y_t|A_t, B_t, C_t) p(A_t, B_t, C_t|\mathbf{y}_{1:t-1})} \\
&= \frac{p(y_t^A|A_t) p(y_t^B|B_t) p(y_t^C|C_t) p(A_t, B_t, C_t|\mathbf{y}_{1:t-1})}{\sum_{A_t} \sum_{B_t} \sum_{C_t} p(y_t|A_t, B_t, C_t) p(A_t, B_t, C_t|\mathbf{y}_{1:t-1})}
\end{aligned}$$

3 Particle filter

We describe here a particle filter based on sequential importance sampling with resampling. The transition priors $p(B_t|B_{t-1})$, $p(A_t|A_{t-1}, B_{t-1})$ and $p(C_t|B_{t-1}, C_{t-1})$ are used to propose the particles (samples) at time t. In addition, the past trajectories are not modified. Hence, the proposal distribution can be written as

$$\begin{aligned}
q(\mathbf{A}_{1:t}, \mathbf{B}_{1:t}, \mathbf{C}_{1:t}|\mathbf{y}_{1:t}) &= p(A_t|A_{t-1}, B_{t-1}) p(C_t|B_{t-1}, C_{t-1}) p(B_t|B_{t-1}) \\
&\quad \times p(\mathbf{A}_{1:t-1}, \mathbf{B}_{1:t-1}, \mathbf{C}_{1:t-1}|\mathbf{y}_{1:t-1})
\end{aligned}$$

For this choice of proposal distribution, the importance weights w_t are given by

$$w_t = \frac{p(\mathbf{A}_{1:t}, \mathbf{B}_{1:t}, \mathbf{C}_{1:t} | \mathbf{y}_{1:t})}{q(\mathbf{A}_{1:t}, \mathbf{B}_{1:t}, \mathbf{C}_{1:t} | \mathbf{y}_{1:t})} \propto p(y_t | A_t, B_t, C_t)$$

The algorithm's pseudo-code follows.

Simple Particle Filter

1. *Initialisation*, $t = 1$.

- For $i = 1, ..., N$, sample $(\mathbf{A}_1^{(i)}, \mathbf{B}_1^{(i)}, \mathbf{C}_1^{(i)}) \sim p(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1)$ and set $t = 2$.

2. *Importance sampling step*

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

$$\begin{aligned} \widetilde{A}_t^{(i)} &\sim p(\widetilde{A}_t | A_{t-1}^{(i)}, B_{t-1}^{(i)}) \\ \widetilde{B}_t^{(i)} &\sim p(\widetilde{B}_t | B_{t-1}^{(i)}) \\ \widetilde{C}_t^{(i)} &\sim p(\widetilde{C}_t | C_{t-1}^{(i)}, B_{t-1}^{(i)}) \end{aligned}$$

  and set $(\widetilde{A}_{1:t}^{(i)}, \widetilde{B}_{1:t}^{(i)}, \widetilde{C}_{1:t}^{(i)}) = (\widetilde{A}_t^{(i)}, \widetilde{B}_t^{(i)}, \widetilde{C}_t^{(i)}, \mathbf{A}_{1:t-1}^{(i)}, \mathbf{B}_{1:t-1}^{(i)}, \mathbf{C}_{1:t-1}^{(i)})$.

- For $i = 1, ..., N$, evaluate the importance weights

$$w_t \propto p(y_t | \widetilde{A}_t^{(i)}, \widetilde{B}_t^{(i)}, \widetilde{C}_t^{(i)})$$

- Normalise the importance weights.

3. *Selection step*

- Resample with replacement N particles $(\mathbf{A}_{1:t}^{(i)}, \mathbf{B}_{1:t}^{(i)}, \mathbf{C}_{1:t}^{(i)}; i = 1, \ldots, N)$ from the set $(\widetilde{A}_{1:t}^{(i)}, \widetilde{B}_{1:t}^{(i)}, \widetilde{C}_{1:t}^{(i)}; i = 1, \ldots, N)$ according to the normalised importance weights.

- Set $t \leftarrow t + 1$ and go to step 2.

The output of the particle filter is a set of samples that can be easily binned to obtain estimates of the marginal and joint filtering distributions.

4 Rao-Blackwellised particle filter

Using the decomposition given by equation (1), we notice that we only need to sample $\{\mathbf{B}_{1:t}^{(i)}; i = 1, \ldots, N\}$ and, subsequently, compute $p(\mathbf{A}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$ and $p(\mathbf{C}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$ analytically. That is, a particle corresponds to the set of variables $(\mathbf{B}_{1:t}^{(i)}, p(\mathbf{A}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)}), p(\mathbf{C}_{1:t}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)}))$. To compute the sufficient statistics $p(A_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$ and $p(C_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$, we need implement the following recursive steps

Prediction:

$$\begin{aligned}
p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) &= \sum_{A_{t-1}} p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}, A_{t-1}) p(A_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) \\
&= \sum_{A_{t-1}} p(A_t|B_{t-1}, A_{t-1}) p(A_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1})
\end{aligned}$$

and similarly

$$\begin{aligned}
p(C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) &= \sum_{C_{t-1}} p(C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}, C_{t-1}) p(C_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) \\
&= \sum_{C_{t-1}} p(C_t|B_{t-1}, C_{t-1}) p(C_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1})
\end{aligned}$$

Update:

$$\begin{aligned}
p(A_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}) &\propto p(y_t|\mathbf{y}_{1:t-1}, A_t, \mathbf{B}_{1:t}) p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) \\
&\propto p(y_t^A|A_t) p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t})
\end{aligned}$$

and similarly,

$$p(C_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}) \propto p(y_t^C|C_t) p(C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t})$$

Once again, we use the transition prior as proposal distribution

$$q(\mathbf{B}_{1:t}|\mathbf{y}_{1:t}) = p(B_t|B_{t-1}) p(\mathbf{B}_{1:t-1}|\mathbf{y}_{1:t-1})$$

For this choice of proposal distribution, the importance weights w_t are given by

$$w_t = \frac{p(\mathbf{B}_{1:t}|\mathbf{y}_{1:t})}{q(\mathbf{B}_{1:t}|\mathbf{y}_{1:t})} \propto p(y_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t})$$

Note that in this case we cannot eliminate the dependence of y_t on the past trajectories because we do not know $\mathbf{A}_{1:t}$ and $\mathbf{C}_{1:t}$. The importance weights are thus given by the predictive density (also known as the innovations or evidence distribution) and not by the likelihood. This expression can be evaluated analytically using the minimum statistics as follows

$$\begin{aligned}
p(y_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) &= \sum_{A_t}\sum_{C_t} p(y_t, A_t, C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) \\
&= \sum_{A_t}\sum_{C_t} p(y_t|\mathbf{y}_{1:t-1}, A_t, C_t, \mathbf{B}_{1:t}) p(A_t, C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t}) \\
&= \sum_{A_t}\sum_{C_t} p(y_t^A|A_t) p(y_t^B|B_t) p(y_t^C|C_t) p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1}) \\
&\qquad \times p(C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1}) \\
&= p(y_t^B|B_t) \sum_{A_t} p(y_t^A|A_t) p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1}) \\
&\qquad \times \sum_{C_t} p(y_t^C|C_t) p(C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1})
\end{aligned}$$

The pseudo-code follows

Rao-Blackwellised Particle Filter

1. *Initialisation*, $t = 1$.

- For $i = 1, ..., N$, sample $\mathbf{B}_1^{(i)} \sim p(\mathbf{B}_1)$ and set $t = 2$.

2. *Importance sampling step*

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

$$\widetilde{B}_t^{(i)} \sim p(\widetilde{B}_t | B_{t-1}^{(i)})$$

  and set $\widetilde{B}_{1:t}^{(i)} = (\widetilde{B}_t^{(i)}, \mathbf{B}_{1:t-1}^{(i)})$.

- For $i = 1, ..., N$, evaluate the importance weights

$$w_t \propto p(y_t | \mathbf{y}_{1:t-1}, \widetilde{B}_{1:t}^{(i)})$$

- Normalise the importance weights.

3. *Selection step*

- Resample with replacement N particles $(\mathbf{B}_{1:t}^{(i)}, p(A_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1}^{(i)})$ $p(C_t|\mathbf{y}_{1:t-1}, \mathbf{B}_{1:t-1}^{(i)}); i = 1, \ldots, N)$ from the set $(\widetilde{B}_{1:t}^{(i)}$, $p(A_t|\mathbf{y}_{1:t-1}, \widetilde{B}_{1:t-1}^{(i)}), p(C_t|\mathbf{y}_{1:t-1}, \widetilde{B}_{1:t-1}^{(i)}); i = 1, \ldots, N)$ according to the normalised importance weights.

4. *Exact step*

- Compute the filtering sufficient statistics $p(A_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$ and $p(C_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$.

- Compute the predictive sufficient statistics $p(A_{t+1}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$ and $p(C_{t+1}|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)})$.

- Set $t \leftarrow t + 1$ and go to step 2.

We can obtain an estimate of $p(B_t|\mathbf{y}_{1:t})$ by simply computing the histogram of $\{B_t^{(i)}; i = 1, \ldots, N\}$. Estimates of the other marginal filtering distributions can be obtained as follows

$$p(A_t|\mathbf{y}_{1:t}) \approx \frac{1}{N}\sum_{i=1}^{N} p(A_t|\mathbf{y}_{1:t}, \mathbf{B}_{1:t}^{(i)}) \tag{2}$$

The joint filtering distribution can be computed as the product of the marginals.

5 Comparative results

We computed the the joint filtering distribution for the network shown in Figure 1 using the three previously described methods. We used two sets of transition and observation matrices to reflect different degrees of noise. The two exact joint filtering distributions and the PF and RBPF approximations for 50 particles are depicted in Figures 2 and 3

Figures 4 and 5 show clearly that for a particular computational cost, RBPF does better than PF. This gain can be much higher if the network is larger than the ABC network and can be broken into several subnetworks.

6 Conclusions

We have shown in this tutorial that RBPF can outperform PF. RBPF exploits the structure of some networks efficiently. *It applies to any DBNs where one of the marginals can be integrated out analytically.* Of course, one can also think of approximate integration to get some reasonable performance for specific applications, but we would no longer be obeying the Rao-Blackwell theorem. Finally, the software for the demo described in this tutorial is available at `http://www.cs.berkeley.edu/~jfgf/`.

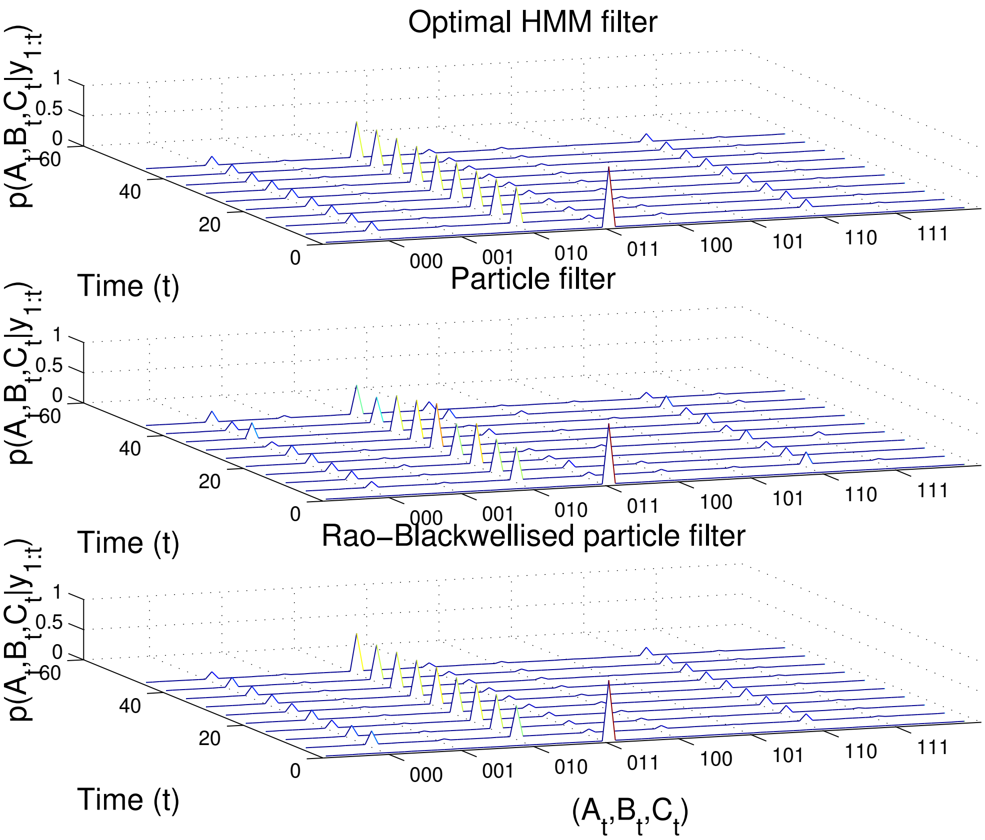

Figure 2: Joint filtering distribution computed with the HMM optimal filter (top) PF (middle) and RBPF (bottom) for high noise transition and observation parameters.

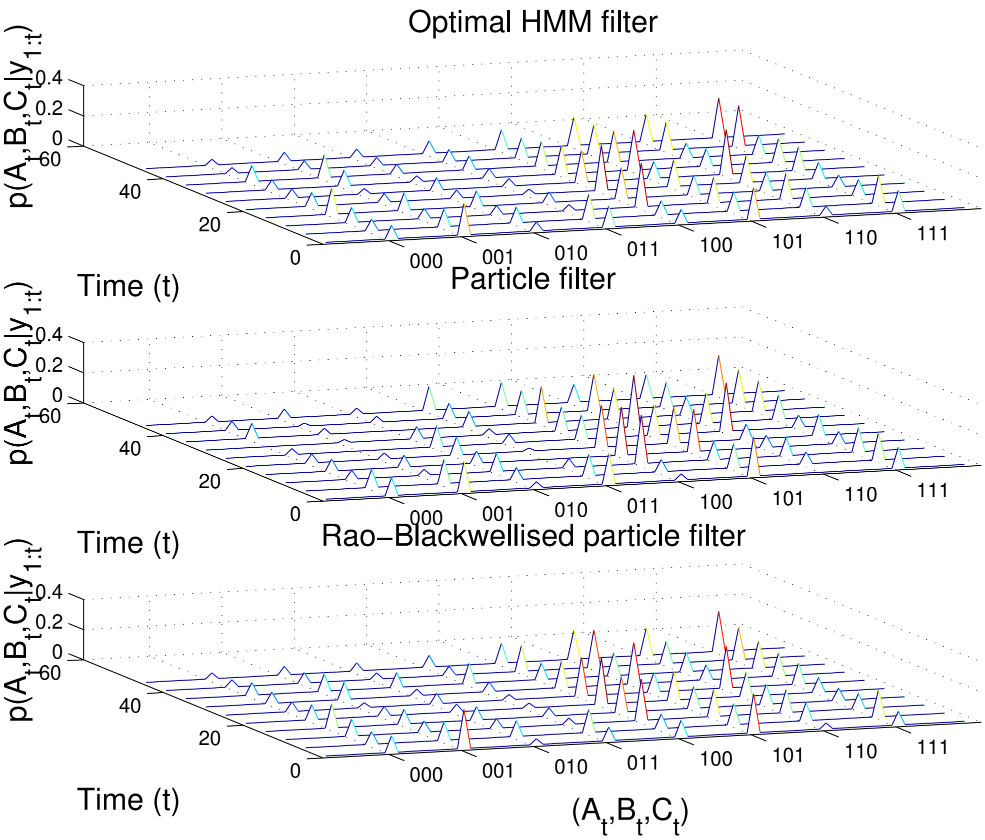

Figure 3: Joint filtering distribution computed with the HMM optimal filter (top) PF (middle) and RBPF (bottom) for high noise transition and observation parameters.

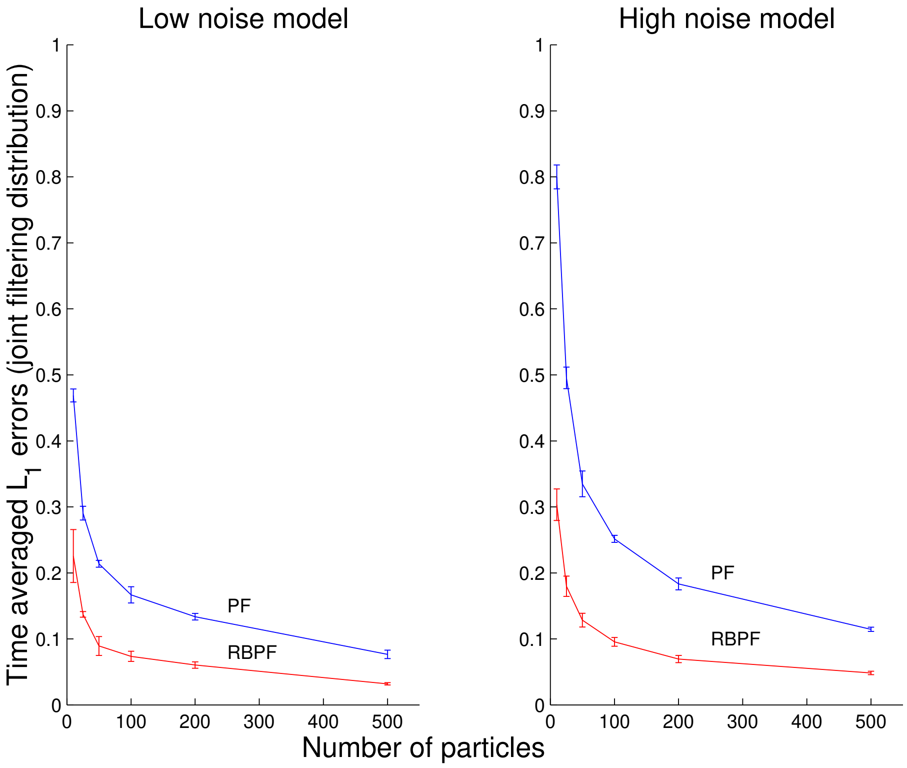

Figure 4: Comparision between the performance of RBPF and PF for both noise models as the number of particles varies.

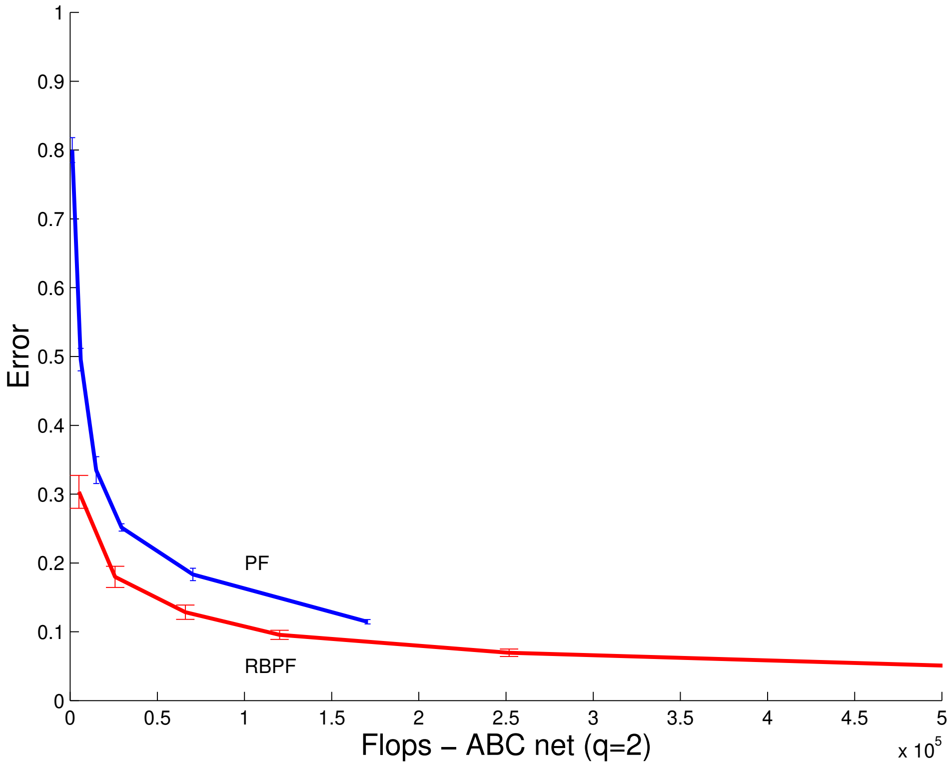

Figure 5: Comparision between the performance of RBPF and PF as a function of the number of floating point operations. "Value for money", RBPF does better.