POZNAN UNIVERSITY OF TECHNOLOGY ACADEMIC JOURNALS
No 76 Electrical Engineering 2013

Piotr KOZIERSKI*
Marcin LIS*

# AUXILIARY AND RAO-BLACKWELLISED PARTICLE FILTERS COMPARISON

Particle filters are very popular – number of algorithms based on Sequential Monte Carlo methods is growing. Paper describes and compares the performance of two of them – Auxiliary and Rao-Blackwellised Particle Filters. Comparison includes also Bootstrap Filter and some variety of SIR algorithm.

## 1. INTRODUCTION

Particle filters (PF) are recently being used in many different problems, for example filtering (hidden variables tracking) [7], system identification [15], object tracking [4] or robot localization problem [17, 18]. Article focuses on the filtration problem.

PF popularity still increase, which is associated with technology development (algorithms requires high performance computing [16]) and with particle filter principle of operation – possibility of the parallel computing implementation which can reduce the computation time even by a few orders of magnitude [13].

In the 2nd chapter describes the principle of operation of particle filter and its specific case – Bootstrap Filter. In chapter 3 Auxiliary PF is described, and Rao-Blackwellised PF is described in chapter 4. Simulation results are presented in chapter 5.

## 2. PARTICLE FILTER

PF is based on Bayesian Filtering

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \tag{1}$$

whose aim is estimation of probability density function (PDF) $p(x_k|Y_k)$ (posterior), therefore density of state variable $x_k$ at time step $k$ based on measurements $Y_k$. It is assumed that

$$Y_k = \{y_1, y_2, ..., y_k\} \tag{2}$$

* Poznań University of Technology.

PF is one of the possible Bayesian filter implementations, in which estimated posterior PDF is not continuous, but composed of a set of $N$ samples, called particles [3], where each has a certain value and weight $\{x_k^i, w_k^i\}$ (for $i = 1,...,N$) – see Fig. 1. The higher the weight $w_k^i$, the greater chance of value $x_k^i$ appearance.

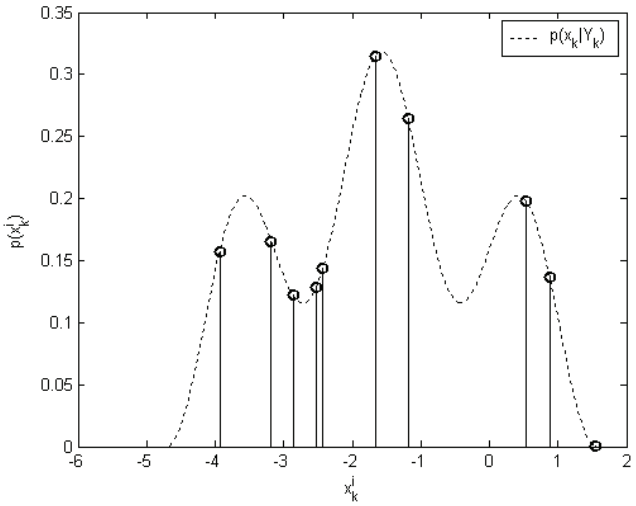


Fig. 1. Posterior PDF composed of particles

## 2.1. SIR algorithm

Bayesian filter (1) can be presented in the recursive form

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|y_{k-1})}{p(y_k|Y_{k-1})} \propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|y_{k-1}) \quad (3)$$

where sign "$\propto$" means "proportional to". $p(x_k|Y_k)$ is the normalizing constant given by [1]

$$p(y_k|Y_{k-1}) = \int p(y_k|x_k)p(x_k|Y_{k-1})dx_k \quad (4)$$

From formula (3) one can see, that posterior can be calculating using the distribution of particles in the previous step.

The SIR (Sequential Importance Resampling) principle of operation is shown in Algorithm 1.

**Algorithm 1**

1. Draw $N$ particles from initial PDF $x_0^i \sim p(x_0)$; set the initial weights $q_0^i = \frac{1}{N}$ and step $k = 1$.
2. Draw N particles from importance density $x_k^i \sim g(x_k|x_{k-1}^i, y_k)$.
3. Compute weights values

$$\tilde{q}_k^i \propto q_{k-1}^i \cdot \frac{p\left(y_k \middle| x_k^i\right) p\left(x_k^i \middle| x_{k-1}^i\right)}{g\left(x_k^i \middle| x_{k-1}^i, y_k\right)} \tag{5}$$

where $p\left(y_k \middle| x_k^i\right)$ is the likelihood available from the measurement model, whereas $p\left(x_k^i \middle| x_{k-1}^i\right)$ is given by transition model.

4. Weights normalization

$$q_k^i = \frac{\tilde{q}_k^i}{\sum_{j=1}^{N} \tilde{q}_k^j} \tag{6}$$

5. Check the condition for resampling; if it is not met, go to step 7.
6. Resampling (see Section 2.3).
7. Increase time step $k = k + 1$; go to step 2.

Importance density, which is used in Algorithm 1, is any PDF proposed by programmer. It can depend on the previous value of state variable and on the current measure value, but it can also be completely independent (but this will negatively affect the performance of the method).

### 2.2. Bootstrap Filter

Bootstrap Filter, proposed in 1993 by Gordon, Salmond and Smith [9] is particular case of SIR. They proposed that the draw should be from transition model

$$g\left(x_k \middle| x_{k-1}^i, y_k\right) = p\left(x_k \middle| x_{k-1}^i\right) \tag{7}$$

Hence the equation (5) simplifies to

$$\tilde{q}_k^i \propto q_{k-1}^i \cdot p\left(y_k \middle| x_k^i\right) \tag{8}$$

The second change to the standard SIR is abandonment of 5th algorithm step and the assumption that resampling will be done in every time. After resampling all weights have the same value $q_{k-1}^i = \frac{1}{N}$, therefore one can simplify expression (8) to

$$\tilde{q}_k^i \propto p\left(y_k \middle| x_k^i\right) \tag{9}$$

Bootstrap Filter principle of operation is presented below.

**Algorithm 2**

1. Draw $N$ particles from initial PDF $x_0^i \sim p(x_0)$; set initial weights $q_0^i = \frac{1}{N}$ and step $k = 1$.
2. Draw N particles from importance density $x_k^i \sim p\left(x_k \middle| x_{k-1}^i\right)$.
3. Compute weights according to formula (9).
4. Weights normalization according to (6).
5. Resampling.
6. Increase the time step $k = k + 1$; go to step 2.

### 2.3. Resampling and its execution condition

Resampling consist in drawing N new samples from PDF consisting of particles (their values and weights).

It is important that draw is only from values of individual particles (draw from discrete PDF), and the probability for i-th value selection is equal to $q_k^i$.

After resampling all weights are set to the same value $q_k^i = \frac{1}{N}$.

There are few different resampling methods, but the fastest is Systematic Resampling. It consist that after creation of the discrete distribution, the whole range from which it will be draw is divided into N equal ranges. From each range is drawn strictly one value.

The resampling steps are written in algorithm below.

**Algorytm 3**

1. Create discrete cumulative distribution function (CDF) based on particle weights.
2. For $j = 1,...,N$ execute steps 3-5.
3. Draw value from the range
$$\left\langle \frac{j-1}{N}; \frac{j}{N} \right) \tag{10}$$
4. Find i-th value (particle) in CDF, which corresponds to drawn value.
5. Remember value of i-th particle
$$\tilde{x}_k^j = x_k^i \tag{11}$$
6. Assign new particle values to the present set
$$x_k = \tilde{x}_k \tag{12}$$
7. Assign new particle weights $q_k^i = \frac{1}{N}$.

Condition for resampling is associated with the so-called Effective Sample Size (ESS), which can be understood as minimum required number of particles to approximate posterior PDF (particles with weights close to zero does not provide any useful information about PDF). ESS one can approximate by equation [2]

$$N_{ESS} = \frac{1}{\sum_{j=1}^{N} \left(q_k^j\right)^2} \tag{13}$$

If $N_{ESS}$ is smaller than certain adopted threshold, resampling should be done. Typically the threshold is taken equal to the half of the particles number $N_T = \frac{N}{2}$.

A more detailed description of PF can be found in [8, 11], whereas another resampling methods can be found in [5, 12].

## 3. AUXILIARY PARTICLE FILTER

Auxiliary Particle Filter (APF), also called Auxiliary SIR (ASIR) is a variant of the SIR algorithm. APF was proposed by Pitt and Shephard in 1999 [14], and it should reduce the two main PF disadvantages – high sensitivity to outliers values and poor posterior approximation [3].

The algorithm assumes adding an auxiliary variable – index, which is draw as particles in resamping, but it has an effect on the particle choice at time update.

APF principle of operation is shown in Algorithm 4.

**Algorithm 4**

1. Draw $N$ particles from initial PDF $x_0^i \sim p(x_0)$; set initial weights $q_0^i = \frac{1}{N}$ and initial time step $k = 1$.
2. Compute value $\mu_k^i = E\left[x_k | x_{k-1}^i\right]$ and weight
$$\tilde{q}_{k*}^i \propto p\left(y_k | \mu_k^i\right) q_{k-1}^i \tag{14}$$
where $\mu_k^i$ is a certain value dependent on $x_{k-1}^i$, characterizing present state value $x_k^i$ (in algorithm assume that it is expected value, but it can be also value drawn from transition model $\mu_k^i \sim p\left(x_k | x_{k-1}^i\right)$).
3. Normalize weights assuming $\sum_{j=1}^{N}\left(q_{k*}^j\right) = 1$.
4. Draw auxiliary indexes $a^i$ for $i = 1,...,N$. Index $a^i = b$ means that in the i-th drawing particle with b-th number was chosen. PDF to drawing is given by pairs $\left\{x_{k-1}^i, q_{k*}^i\right\}$.
5. For $i = 1,...,N$ draw samples, using the previously obtained auxiliary indexes $x_k^i \sim p\left(x_k | x_{k-1}^{a^i}\right)$.
6. Compute particle weights according to expression
$$\tilde{q}_k^i \propto \frac{p\left(y_k | x_k^i\right)}{p\left(y_k | \mu_k^{a^i}\right)} \tag{15}$$
7. Normalize particle weights according to (6).
8. Resampling.
9. Increase the time step $k = k + 1$, go to step 2.

In [1] it was found that in Algorithm 4 resampling in step 8 is unnecessary, since during previous steps have already occurred particles drawing, using information about the current measurement (this step was implemented to compare the original methods).

## 4. RAO-BLACKWELLISED PARTICLE FILTER

The approach proposed in Rao-Blackwellised Particle Filter (RBPF) is feasible only in the specific case, where at least one state variable is linearly dependent in all state space model equations.

It is assumed that state variables $x_k$ can be divided into two groups – $x_k^{PF}$ (nonlinearly dependent) and $x_k^{KF}$ (linearly dependent) [6]. Then system model can be written by equations

$$x_k^{PF} = f^{PF}\left(x_{k-1}^{PF}\right) + F^{PF}\left(x_{k-1}^{PF}\right)x_{k-1}^{KF} + G^{PF}\left(x_{k-1}^{PF}\right)v_{k-1}^{PF} \tag{16a}$$

$$x_k^{KF} = f^{KF}\left(x_{k-1}^{PF}\right) + F^{KF}\left(x_{k-1}^{PF}\right)x_{k-1}^{KF} + G^{KF}\left(x_{k-1}^{PF}\right)v_{k-1}^{KF} \tag{16b}$$

$$y_k = h\left(x_k^{PF}\right) + H\left(x_k^{PF}\right)x_k^{KF} + n_k \tag{16c}$$

The advantages of this algorithm is performance increase through the use of the Kalman Filter to certain of state variables. One can also reduce the number of particles, since value vector of particle has reduced dimension [10].

Further analysis will be made for a particular model

$$x_{1,k} = 0.8\frac{x_{2,k-1}}{\left(x_{1,k-1}\right)^2 + 1} + v_{1,k-1} \tag{17a}$$

$$x_{2,k} = 0.4x_{1,k-1} + 0.7x_{2,k-1} + v_{2,k-1} \tag{17b}$$

$$y_k = x_{1,k} - x_{2,k} + n_k \tag{17c}$$

where $v_{1,k-1}$, $v_{2,k-1}$ and $n_k$ are Gaussian noises with variances equal respectively $0.1$, $0.3$, $0.2$ (mean value equal to zero in each case).

Comparing equations (16) and (17) one can write following expressions

$$x_{1,k} = x_k^{PF}, \quad x_{2,k} = x_k^{KF}, \quad f^{PF}\left(x_{k-1}^{PF}\right) = 0, \quad f^{KF}\left(x_{k-1}^{PF}\right) = 0.4x_k^{PF} \tag{18a}$$

$$F^{PF}\left(x_{k-1}^{PF}\right) = \frac{0.8}{\left(x_{k-1}^{PF}\right)^2 + 1}, \quad F^{KF}\left(x_{k-1}^{PF}\right) = 0.7, \quad G^{PF}\left(x_{k-1}^{PF}\right) = 1, \quad G^{KF}\left(x_{k-1}^{PF}\right) = 1 \tag{18b}$$

$$v_{k-1}^{PF} = v_{1,k-1}, \quad v_{k-1}^{KF} = v_{2,k-1}, \quad h\left(x_k^{PF}\right) = x_k^{PF}, \quad H\left(x_k^{PF}\right) = -1 \tag{18c}$$

Moreover, based on variances assumed

$$Q^{PF} = 0.1, \quad Q^{KF} = 0.3, \quad R = 0.2 \tag{19}$$

One can see that due to low dimension of the system (17), all matrices become scalars.

Algorithm 5 describes RBPF principle of operation for system (17)

**Algorithm 5**

1. Initialization. For $i = 1,\ldots,N$ draw particles from initial PDF $x_0^{PF(i)} \sim p\left(x_0^{PF}\right)$ and set values $\left\{x_{0|0}^{KF(i)}, P_{0|0}^{(i)}\right\} = \left\{x_0^{KF}, P_0\right\}$. Set time step $k = 1$.
2. PF time update. For $i = 1,\ldots,N$ draw particles from transition model $x_{k*}^{PF(i)} \sim p\left(x_k^{PF} \middle| x_{k-1}^{PF(i)}, x_{k-1|k-1}^{KF(i)}\right)$.

3. KF time update. For $i = 1,...,N$ calculate

$$x_{k|k-1}^{KF(i)} = 0.7x_{k-1|k-1}^{KF(i)} + 0.4x_{k-1}^{PF(i)} + L_{k-1}^{(i)}\left(x_{k*}^{PF(i)} - \frac{0.8x_{k-1|k-1}^{KF(i)}}{\left(x_{k-1}^{PF(i)}\right)^2 + 1}\right) \tag{20}$$

$$P_{k|k-1}^{(i)} = 0.7^2 P_{k-1|k-1}^{(i)} + 0.3 - L_{k-1}^{(i)} N_{k-1}^{(i)} \left(L_{k-1}^{(i)}\right)^T \tag{21}$$

where

$$N_{k-1}^{(i)} = \frac{0.8^2 P_{k-1|k-1}^{(i)}}{\left(\left(x_{k-1}^{PF(i)}\right)^2 + 1\right)^2} + 0.1 \tag{22}$$

$$L_{k-1}^{(i)} = 0.7^2 P_{k-1|k-1}^{(i)} \left(N_{k-1}^{(i)}\right)^{-1} \tag{23}$$

4. PF measurement update. For $i = 1,...,N$ calculate particles weights

$$\tilde{q}_k^{(i)} \propto p\left(y_k \middle| x_{k*}^{PF(i)}, x_{k|k-1}^{KF(i)}\right) \tag{24}$$

For $i = 1,...,N$ normalize particle weights according to (6).

5. Resampling. Draw N samples $x_k^{PF(i)}$, assuming that probability to draw value $x_{k*}^{PF(j)}$ is given by

$$\Pr\left(x_k^{PF(i)} = x_{k*}^{PF(j)}\right) = q_k^{(j)} \tag{25}$$

6. KF measurement update. For $i = 1,...,N$ calculate

$$x_{k|k}^{KF(i)} = x_{k|k-1}^{KF(i)} + K_k^{(i)}\left(y_k - x_k^{PF(i)} + x_{k|k-1}^{KF(i)}\right) \tag{26}$$

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} M_k^{(i)} \left(K_k^{(i)}\right)^T \tag{27}$$

where

$$M_k^{(i)} = P_{k|k-1}^{(i)} + 0.2 \tag{28}$$

$$K_k^{(i)} = -P_{k|k-1}^{(i)} \left(M_k^{(i)}\right)^{-1} \tag{29}$$

7. Increase time step $k = k + 1$; go to step 2.

It should be noted that in this case the i-th particle is composed of $x_k^{PF(i)}$, $q_k^{(i)}$, $x_{k|k}^{KF(i)}$ and $P_{k|k}^{(i)}$.

## 5. SIMULATION RESULTS

System (17) was used in all simulations. In each case simulation length was $M = 1000$ steps. The sequence of state variables and measurements was identical for all cases. 4 algorithms Were compared – BF, RBPF, APF and SIR. In the SIR algorithm assumed that importance density was unconditional Gaussian function $\mathcal{N}(0,1.5)$ and that the resampling was performed at each algorithm iteration. Hence expression (5) was simplified to the form

$$\tilde{q}_k^i \propto \frac{p\left(y_k \middle| x_k^i\right) p\left(x_k^i \middle| x_{k-1}^i\right)}{g\left(x_k^i\right)} \tag{30}$$

It should be noted that state variable $x_k^i$ for system (17) is a two-element vector $x_k^i = \left[x_{1,k}^i \quad x_{2,k}^i\right]^T$, and hence probabilities in (30) are given by product of the two probability values

$$p\left(x_k^i \middle| x_{k-1}^i\right) = p\left(x_{1,k}^i \middle| x_{k-1}^i\right) p\left(x_{2,k}^i \middle| x_{k-1}^i\right) \tag{31}$$

$$g\left(x_k^i\right) = g\left(x_{1,k}^i\right) g\left(x_{2,k}^i\right) \tag{32}$$

Simulation results are shown in Table 1. The results includes Mean Square Error (MSE) for both state variables and computation time.

Table 1. Comparing performance tracking of BF, RBPF, APF and SIR – simulation results

| | N = 100 | N = 200 | N = 500 |
|---|---|---|---|
| BF | $MSE_1 = 0.1980$<br>$MSE_2 = 0.2889$<br>t = 0.40s | $MSE_1 = 0.1938$<br>$MSE_2 = 0.2817$<br>t = 0.74s | $MSE_1 = 0.1897$<br>$MSE_2 = 0.2758$<br>t = 1.70s |
| APF | $MSE_1 = 0.3515$<br>$MSE_2 = 0.4921$<br>t = 0.70s | $MSE_1 = 0.3595$<br>$MSE_2 = 0.4904$<br>t = 1.32s | $MSE_1 = 0.3471$<br>$MSE_2 = 0.4722$<br>t = 3.13s |
| RBPF | $MSE_1 = 0.1962$<br>$MSE_2 = 0.2815$<br>t = 3.81s | $MSE_1 = 0.1957$<br>$MSE_2 = 0.2818$<br>t = 7.56s | $MSE_1 = 0.1954$<br>$MSE_2 = 0.2821$<br>t = 19.53s |
| SIR | $MSE_1 = 0.5600$<br>$MSE_2 = 0.7145$<br>t = 0.52s | $MSE_1 = 0.5468$<br>$MSE_2 = 0.5410$<br>t = 0.99s | $MSE_1 = 0.5552$<br>$MSE_2 = 0.4403$<br>t = 2.30s |

## 6. CONCLUSIONS

Comparing the results from Table 1 one can see that the best tracking of hidden variables and the shortest calculations time occurs for simple BF algorithm. Tracking performance for RBPF is close to BF, but computation time is about 10-fold greater. Therefore one can conclude that RBPF is useful only when number of linearly dependent state variables is greater.

Auxiliary Particle Filter for proposed system has a very poor performance, whereas SIR with independent importance density has the worst tracking performance – it is shown that algorithm with such function can work, but it is far from expectations.

It is shown that choice of PF algorithm should be taken into account a system model, because instead of improve the results, it can be worse.

Future research will be focused on the APF and its poor tracking performance, to get better results. Likewise some attempts will be made to system classification in respect of PF algorithms – the best of PFs for different system models.

## REFERENCES


[1] Arulampalam S., Maskell S., Gordon N., Clapp T., A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking, IEEE Proceedings on Signal Processing, Vol.50, No.2, 2002, pp.174-188.

[2] Brzozowska-Rup K., Dawidowicz A.L., Metoda filtru cząsteczkowego, Matematyka Stosowana: matematyka dla społeczeństwa 2009, T. 10/51, pp.69-107.

[3] Candy J.V., Bayesian signal processing, WILEY, New Jersey 2009, pp.19-44, 237-298.

[4] Chang C., Ansari R., Khokhar A., Multiple Object Tracking with Kernel Particle Filter, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2005, Vol. 1, pp.566-573.

[5] Douc R., Cappe O., Moulines E., Comparison of Resampling Schemes for Particle Filtering, Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, September 2005, pp.64-69.

[6] Doucet A., Freitas N., Murphy K., Russell S., Rao-Blackwellised particle filtering for dynamic Bayesian networks, Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, pp.176-183.

[7] Doucet A., Godsill S., Andrieu C., On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing, 10, 2000, pp.197-208.

[8] Doucet A., Johansen A.M., A Tutorial on Particle Filtering and Smoothing: Fifteen years later, handbook of Nonlinear Filtering 2009/12, pp.656-704.

[9] Gordon N.J., Salmond N.J., Smith A.F.M., Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings-F, Vol.140, No.2, 1993, pp.107-113.

[10] Handeby G., Karlsson R., Gustafsson F., The Rao-Blackwellized Particle Filter: A Filter Bank Implementation, EURASIP Journal on Advances in Signal Processing, Volume 2010, Article ID 724087, pp.10.

[11] Kozierski P., Lis M., Filtr cząsteczkowy w problemie śledzenia – wprowadzenie, Studia z Automatyki i Informatyki, Tom 37, 2012, pp.79-94.

[12] Liu J.S., Chen R., Sequential Monte Carlo Methods for Dynamic Systems, Journal of the American Statistical Association, September 1998, Vol. 93, No. 443, pp.1032-1044.

[13] Mountney J., Obeid I., Silage D., Modular Particle Filtering FPGA Hardware Architecture for Brain Machine Interfaces, Conf Proc IEEE Eng Med Biol Soc. 2011, pp.4617-4620.

[14] Pitt M., Shephard N., Filtering via simulation: auxiliary particle filters, Journal of the American Statistical association, Vol.94, No.446, pp.590-599.

[15] Schön T.B., Wills A., Ninness B., System identification of nonlinear state-space models, Automatica 47 (2011), pp.39-49.

[16] Sutharsan S., Kirubarajan T., Lang T., McDonald M., An Optimization-Based Parallel Particle Filter for Multitarget Tracking, IEEE Transactions on Aerospace and Electronic Systems, Vol.48, No.2, 4/2012, pp.1601-1618.
[17] Thrun S., Particle Filters in Robotics, Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI), 2002.
[18] Woo J., Kim Y-J., Lee J., Lim M-T., Localization of Mobile Robot using Particle Filter, SICE-ICASE International Joint Conference 2006, pp. 3031-3034.