A NEW FORMULATION OF THE RAO-BLACKWELLIZED PARTICLE FILTER

Gustaf Hendeby, Rickard Karlsson, Fredrik Gustafsson

Division of Automatic Control
Department of Electrical Engineering
Linköpings University, Sweden

ABSTRACT

The standard formulation of the Rao-Blackwellized particle filter (RBPF) is usually presented in a way that makes it hard to implement in an object oriented fashion. This paper discusses how the solution can be rewritten in order to increase the understanding as well as simplify the implementation and use standard filtering components, such as Kalman filter banks and particle filters. Calculations show that the new algorithm is equivalent to the classical formulation, and the new algorithm will be exemplified in a simulation study.

***Index Terms*—** Particle filter, Rao-Blackwellisation, Kalman filter, Object oriented design

1. INTRODUCTION

The *particle filter* (PF), [1, 2], is the fundamental solution to many recursive Bayesian filtering problems, incorporating both nonlinearity and non-Gaussianity in an optimal way. Furthermore, the *Rao-Blackwellized particle filter* (RBPF), sometimes denoted the *marginalized particle filter* (MPF), [3–9] improves the performance when a linear-Gaussian substructure is present.

The RBPF divides the state vector x_t into two parts, one part x_t^p, which is estimated using the PF, and another part x_t^k, where *Kalman filters* (KF) are used. Basically, denoting the measurements and states up to time t with $\mathbb{Y}_t = \{y_j\}_{j=0}^t$ and $\mathbb{X}_t = \{x_j\}_{j=1}^t$, respectively, the joint *probability density function* (PDF) is given using Bayes' rule as

$$p(\mathbb{X}^p, x_t^k|\mathbb{Y}_t) = p(x_t^k|\mathbb{X}^p, \mathbb{Y}_t)p(\mathbb{X}_t^p|\mathbb{Y}_t). \tag{1}$$

If the model is conditionally linear Gaussian, *i.e.*, if the term $p(x_t^k|\mathbb{X}_t^p, \mathbb{Y}_t)$ is linear Gaussian, it can be optimally estimated using the KF. To obtain the second factor it is necessary to resort to using the PF.

2. RAO-BLACKWELLIZED PARTICLE FILTER (RBPF)

A fairly general state space setting where the RBPF can be applied, [8–10], is given by the model

$$x_{t+1}^p = f^p(x_t^p) + F^p(x_t^p)x_t^k + G^p(x_t^p)w_t^p \tag{2a}$$

$$x_{t+1}^k = f^k(x_t^p) + F^k(x_t^p)x_t^k + G^k(x_t^p)w_t^k \tag{2b}$$

$$y_t = h(x_t^p) + H(x_t^p)x_t^k + e_t, \tag{2c}$$

where the noise should be independent and Gaussian,

$$w_t^p \sim \mathcal{N}(0, Q_t^p), \quad w_t^k \sim \mathcal{N}(0, Q_t^k), \quad \text{and} \quad e_t \sim \mathcal{N}(0, R_t).$$

The authors gratefully acknowledge fundings from SSF (Swedish Foundation for Strategic Research) Strategic Research Center MOVIII, and VR (the Swedish Research Council) project Sensor Informatics.

2.1. RBPF — Standard Formulation

The standard approach to implementing the RBPF for the model structure in (2) is given in for instance [8–10]. The algorithm is summarized in Algorithm 1. Here, the KF and PF parts interact heavily and steps in the respective algorithms are mixed, therefore it is difficult to clearly see the problem structure and how to use standard components in the filtering. Also, the time update and measurement update of the various filters are neither completely separated nor straightfor-

Algorithm 1 Rao-Blackwellized PF (Standard formulation)

1: Initialization: For $i = 1, \dots, N$, $x_{0|-1}^{p,(i)} \sim p_{x_0^p}(x_0^p)$ and set $\{x_{0|-1}^{k,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^k, \bar{P}_0\}$. Let $t = 0$.

2: PF measurement update: For $i = 1, \dots, N$, evaluate and normalize the importance weights using to the likelihood

$$p(y_t|\mathbb{X}_t^p, \mathbb{Y}_{t-1}) = \mathcal{N}(h(x_t^p) + H(x_t^p)\hat{x}_{t|t-1}^k, \; H(x_t^p)P_{t|t-1}(H(x_t^p))^T + R_t).$$

Resample the particles if needed.

3: PF time update and Kalman filter update

(a) KF measurement update

$$\begin{aligned} \hat{x}_{t|t}^{k,(i)} &= \hat{x}_{t|t-1}^{k,(i)} + K_t\big(y_t - h(x_t^p) - H(x_t^p)\hat{x}_{t|t-1}^k\big), \\ P_{t|t} &= P_{t|t-1} - K_t M_t K_t^T, \\ M_t &= H(x_t^p)P_{t|t-1}H_t^T(x_t^p) + R_t, \\ K_t &= P_{t|t-1}H^T(x_t^p)M_t^{-1}. \end{aligned}$$

(b) PF time update: For $i = 1, \dots, N$,

$$x_{t+1}^{p,(i)} \sim \mathcal{N}\big(f^p(x_t^{p,(i)}) + F^p(x_t^p)\hat{x}_{t|t}^{k,(i)}, \; F^p(x_t^p)P_{t|t}(F^p(x_t^p))^T + Q_t^p\big)$$

(c) KF time update,

$$\begin{aligned} \hat{x}_{t+1|t}^{k,(i)} &= F^k\hat{x}_{t|t}^{k,(i)} + f^k(x_t^p) \\ &\quad + L_t(x_{t+1|t}^{p,(i)} - f^p(x_t^{p,(i)}) - F^p(x_t^p)\hat{x}_{t|t}^{k,(i)}), \\ P_{t+1|t} &= F^k(x_t^p)P_{t|t}(F^k(x_t^p))^T + Q_t^k - L_t M_t L_t^T, \\ M_t &= F^k(x_t^p)P_{t|t}(F^k(x_t^p))^T + Q_t^p, \\ L_t &= F^k(x_t^p)P_{t|t}(F^p(x_t^p))^T M_t^{-1}, \end{aligned}$$

4: Set $t := t + 1$ and repeat from step 2.

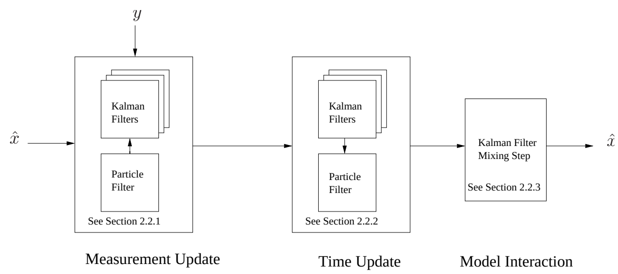

Fig. 1. Illustrating the RBPF (New formulation). The dependency order in the measurement and time update are also indicated.

wardly explained. This hinders reuse of standard components such as existing KF and PF filter implementations when implementing the filter.

2.2. RBPF — New Formulation

In [11] an object oriented general filtering software in C++ called F++[1] is presented. When implementing new algorithms, such as the RBPF, it is important to reuse various software components. For instance the RBPF could in principle use previously developed PF and KF components. However, using Algorithm 1 this is neither very easy nor straightforward to do.

In this section a new RBPF algorithm is presented that is formulated in terms of well-known KF and PF steps followed by a simple compensation step. The new structural insight given by this new algorithm formulation helps to give a better understanding of the RBPF algorithm and at the same time it allows for efficient software development in terms of standard signal processing components.

The key to the new algorithm formulation is to view the RBPF as interacting filters working with different system models. The linear part is captured in one model, that has the nonlinear components as input, and the nonlinear effects are captured in a nonlinear model including only the nonlinear parts, where the noise depends on the linear estimates. This is illustrated in Fig. 1, and in the subsequent sections.

The sequel of this section will explain this in more detail. First the result is stated and then the derivation is given in Section 2.2.4.

2.2.1. Kalman Filter Model

Consider the model (2), from the conditional linear Gaussian property in (1), or by studying Algorithm 1, the expected filtering densities are known. They can be obtained using this linear model, where ξ_t is a dummy variable with much in common with the nonlinear state x_t^p. This extension with ξ will prove useful later. The linear model becomes:

$$\begin{pmatrix}\xi_{t+1}\\ x_{t+1}^k\end{pmatrix} = \begin{pmatrix}0 & F^p(x_t^p)\\ 0 & F^k(x_t^p)\end{pmatrix}\begin{pmatrix}\xi_t\\ x_t^k\end{pmatrix} + \begin{pmatrix}G^p(x_t^p) & 0\\ 0 & G^k(x_t^p)\end{pmatrix}\begin{pmatrix}w_t^p\\ w_t^k\end{pmatrix} + \begin{pmatrix}u_t^p\\ u_t^k\end{pmatrix} \tag{4a}$$

$$y_t = H(x_t^p)x_t^k + u_t^h + e_t, \tag{4b}$$

where the nonlinear effects in (2) enter as external input to the system, $u_t^p = f^p(x_t^p)$, $u_t^k = f^k(x_t^p)$, and $u_t^h = h(x_t^p)$.

[1] F++: `http://www.control.isy.liu.se/resources/f++`

Algorithm 2 Rao-Blackwellized PF (New formulation)

1: Initialization (same as standard formulation)
2: Measurement update
 (a) PF measurement update using (5).
 (b) KF measurement update using (4) (depends on the PF)
3: Time update
 (a) KF time update using (4).
 (b) PF time update using (5) (depends on the KF)
 (c) Mixing step, update the KF using (6).
4: Increase time and repeat from step 2.

2.2.2. Particle Filter Model

The nonlinear effects that are not taken care of in the linear model are put into a model to be handled by a PF. The resulting model is

$$x_{t+1}^p = \bar{w}_t^p \tag{5a}$$

$$y_t = \bar{e}_t, \tag{5b}$$

with the noise

$$\bar{w}_t^p \sim \mathcal{N}(\hat{\xi}_{t+1|t}, P_{t+1|t}^{\xi}), \tag{5c}$$

$$\bar{e}_t \sim \mathcal{N}\big(h(x_t^p) + H(\hat{x}_{t+1|t}^p)x_{t+1|t}^k, S_t\big), \tag{5d}$$

where

$$S_t = H(\hat{x}_{t|t-1}^p)P_{t|t-1}^k\big(H(\hat{x}_{t|t-1}^p)\big)^T + R_t. \tag{5e}$$

Hence, through the noise the model depends non-trivially on the linear model.

2.2.3. Model Interaction

The suggested formulation of the RBPF is now the one given in Algorithm 2, where KF and PF are run in parallel, with the following mixing step performed after both KF and PF has been time updated to adjust the KF based on the PF results:

$$\hat{x}_{t+1|t}^k := \hat{x}_{t+1|t}^k + P_{t+1|t}^{k\xi}P_{t+1|t}^{-\xi}(x_{t+1}^p - \hat{\xi}_{t+1|t}) \tag{6a}$$

$$P_{t+1|t}^k := P_{t+1|t}^k - P_{t+1|t}^{\xi k}P_{t+1|t}^{-\xi}P_{t+1|t}^{k\xi}. \tag{6b}$$

2.2.4. The New RBPF Algorithm

In Algorithm 2 the new formulation is summarized. Note that the KF measurement and time update steps are the traditional update of the mean value and the covariance in a KF. For the PF they represent a standard PF with weight computation and resampling, and simulation of the system. This is all standard components. Schematically, Algorithm 2 can be depicted as in Fig. 1.

Derivation of Algorithm 2

Here, Algorithm 2 will be verified by showing that the resulting equations are identical to those in the classic algorithm.

Applying the KF on the linear system (4) yields

$$\begin{pmatrix}\hat{\xi}_{t+1|t}\\ \hat{x}_{t+1|t}^k\end{pmatrix} = \begin{pmatrix}F^p(x_t^p)\hat{x}_{t|t}^k + u_t^p\\ F^k(x_t^p)\hat{x}_{t|t}^k + u_t^k\end{pmatrix}, \tag{7a}$$

and the covariance matrix $P = \left(\begin{smallmatrix} P^{\xi} & P^{\xi k} \\ P^{k\xi} & P^{k}\end{smallmatrix}\right)$ given by

$$P^{\xi}_{t+1|t} = F^p(x^p_t)P^k_{t|t}(F^p(x^p_t))^T + G^p(x^p_t)Q^p_t(G^p(x^p_t))^T \quad (7b)$$

$$P^{k}_{t+1|t} = F^k(x^p_t)P^k_{t|t}(F^k(x^p_t))^T + G^k(x^p_t)Q^k_t(G^k(x^p_t))^T \quad (7c)$$

$$P^{\xi k} = (P^{k\xi})^T = F^p(x^p_t)P^k_{t|t}(F^k(x^p_t))^T. \quad (7d)$$

Applying the KF measurement update then yields

$$\begin{pmatrix}\hat{\xi}_{t|t} \\ \hat{x}^k_{t|t}\end{pmatrix} = \begin{pmatrix}\hat{\xi}_{t|t-1} \\ \hat{x}^k_{t|t-1}\end{pmatrix} - \begin{pmatrix}P^{\xi k}_{t|t-1} \\ P^{k}_{t|t-1}\end{pmatrix}(H(x^p_{t|t-1}))^T S_t^{-1}\epsilon_t, \quad (8a)$$

$$\epsilon_t = y_t - H(x^p_{t|t-1}) - u^h_t,$$

$$S_t = H(\hat{x}^p_{t|t-1})P^k_{t|t-1}(H(\hat{x}^p_{t|t-1}))^T + R_t.$$

with

$$P^{\xi}_{t|t} = P^{\xi}_{t|t-1} - P^{\xi k}_{t|t-1}(H(\hat{x}^p_{t|t-1}))^T S_t^{-1} H(\hat{x}^p_{t|t-1})P^{k\xi}_{t|t-1} \quad (8b)$$

$$P^{k}_{t|t} = P^{k}_{t|t-1} - P^{\xi k}_{t|t-1}(H(\hat{x}^p_{t|t-1}))^T S_t^{-1} H(\hat{x}^p_{t|t-1})P^{k}_{t|t-1} \quad (8c)$$

$$P^{\xi k}_{t|t} = P^{\xi k}_{t|t-1} - P^{\xi k}_{t|t-1}(H(\hat{x}^p_{t|t-1}))^T S_t^{-1} H(\hat{x}^p_{t|t-1})P^{k}_{t|t-1} \quad (8d)$$

$$P^{\xi k}_{t|t} = (P^{k\xi}_{t|t})^T. \quad (8e)$$

Combine the time update step with the model interaction in (6) and the equations in the linear part in Algorithm 1 are obtained.

Furthermore, looking at the equations for the ξ estimates, it is clear that with the noise in (5) the PF is identical to that in Algorithm 1. Thus, the new algorithm formulation is equivalent to the standard formulation.

Note here that appending the linear part with the dummy variable ξ does not impose any unnecessary computations being made since the ξ information is used in the PF and in the model interaction step. Given that the KF is a well studied algorithm, with many efficient implementations, the implementation of the algorithm is likely to be both more efficient and numerically well-behaved using the KF formulation, than if reimplemented from scratch.

It should also be noted that the model interaction step can be interpreted as updating the KF for the measurement,

$$x^p_t = \xi_t + w^p_t.$$

This fact can be used in order to get an efficient the implementation of the interaction step.

3. EXAMPLE

To exemplify the new RBPF algorithm formulation, the target tracking example from [10] will be revisited. The problem of estimating the position and velocity of an aircraft is studied using the model

$$x_{t+1} = \left(\begin{array}{cc|cccc} 1 & 0 & T & 0 & T^2/2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2/2 \\ \hline 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) x_t + w_t, \quad (9a)$$

$$y_t = \begin{pmatrix}\sqrt{p_x^2 + p_y^2} \\ \arctan(p_y/p_x)\end{pmatrix} + e_t, \quad (9b)$$

where $Q = \operatorname{cov} w = \operatorname{diag}(1\ 1\ 1\ 1\ 0.01\ 0.01), R = \operatorname{cov} e = \operatorname{diag}(100\ 10^{-6})$ and the state vector is $x_t = (p_x\ p_y\ v_x\ v_y\ a_x\ a_y)^T$, *i.e.* position, velocity and acceleration. The measurement equation gives the range and azimuth to the radar system. Here the state space can be split into $x^p_t = (p_x\ p_y)^T$ and $x^k_t = (v_x\ v_y\ a_x\ a_y)^T$, and then the RBPF be applied to lessen the computational burden compared to using a regular PF.

Simulations conducted on the example shows that the performance is comparable to the results obtained in [10].

In the final version this paper will include an description on how to set up and apply the new formulation to the RBPF using existing KF and PF implementations.

4. CONCLUSIONS

This paper presents the *Rao-Blackwellized particle filter* (RBPF) in a new more structured way, that is more suitable for implementation. This enables easy integration with previously well-studied algorithms, such as PF and KF, together with an increased insight into the inner workings of the algorithm. This is particularly interesting in an object oriented framework, such as the software package F++. The algorithm is verified in calculations, and will be exemplified in the full-length paper and possibly also with a the simulation where the method is applied to more challenging problems where the RBPF is crucial.

5. REFERENCES

[1] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc.-F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[2] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. New York: Springer-Verlag, 2001.

[3] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[4] G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996.

[5] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 613–624, Mar. 2001.

[6] R. Chen and J. S. Liu, "Mixture Kalman filter," *J. Royal Statist. Soc. B*, vol. 62, no. 3, pp. 493–508, 2000.

[7] C. Andrieu and A. Doucet, "Particle filter for partially observed Gaussian state space models," *J. Royal Statist. Soc. B*, vol. 64, no. 4, pp. 827–836, 2002.

[8] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear / nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[9] T. B. Schön, R. Karlsson, and F. Gustafsson, "The marginalized particle filter in practice," in *Proc. IEEE Aerosp. Conf*, Big Sky, MT, USA, Mar. 2006.

[10] R. Karlsson, T. B. Schön, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, Nov. 2005.

[11] G. Hendeby and R. Karlsson, "Target tracking performance evaluation — a general software environment for filtering," in *Proc. IEEE Aerosp. Conf*, Big Sky, MT, USA, Mar. 2007.