

A New Formulation of the Rao-Blackwellized Particle Filter

Gustaf Hendeby, Rickard Karlsson, Fredrik Gustafsson

Division of Automatic Control

E-mail: hendeby@isy.liu.se, rickard@isy.liu.se, fredrik@isy.liu.se

13th August 2007

Report no.: [LiTH-ISY-R-2813](#)

Accepted for publication in Proceedings of the IEEE Statistical Signal Processing Workshop, Madison, WI, 2007

Address:

Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET



Abstract

For performance gain and efficiency it is important to utilize model structure in particle filtering. Applying Bayes' rule, present linear Gaussian substructure can be efficiently handled by a bank of Kalman filters. This is the standard formulation of the *Rao-Blackwellized particle filter* (RBPF), by some authors denoted the *marginalized particle filter* (MPF), and 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 reuse of 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 is exemplified in a target tracking simulation study.

Keywords: Particle filter, Rao-Blackwellization, Kalman filter, Object oriented design, Marginalized particle filter

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. For linear and Gaussian systems the optimal solution is given by the *Kalman filter* (KF), [3, 4]. Furthermore, the *Rao-Blackwellized particle filter* (RBPF), sometimes denoted the *marginalized particle filter* (MPF) or mixture Kalman filters, [5, 6, 7, 8, 9, 10, 11] improves the performance when a linear-Gaussian substructure is present. A recent survey of different particle filter related methods is given in [12].

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 KFs 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

$$p(\mathbb{X}^p, x_t^k | \mathbb{Y}_t) = \underbrace{p(x_t^k | \mathbb{X}^p, \mathbb{Y}_t)}_{\text{KF}} \underbrace{p(\mathbb{X}_t^p | \mathbb{Y}_t)}_{\text{PF}}. \quad (1)$$

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, [10, 13, 11], is given by the model

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

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

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

where the noises are assumed independent and Gaussian,

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

2.1. RBPF — Standard Formulation

The standard approach to implementing the RBPF for the model structure in (2) is given in for instance [10, 13, 11]. The algorithm is summarized in Algorithm 1. Here, the KF and PF parts interact heavily and steps in the respective algorithms are mixed, it is therefore 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 clearly stated. This hinders reuse of standard components such as existing KF and PF filter code when implementing the filter.

Note that for the special case, when there is no linear term in (2c) and that the linear dynamics is independent of x^p , *i.e.*,

Algorithm 1 Rao-Blackwellized PF (Standard formulation)

- 1: Initialization: For $i = 1, \dots, N$, $x_{0|-1}^p \sim p_{x_0^p}(x_0^p)$ and set $\{x_{0|-1}^k, P_{0|-1}\} = \{x_0^k, P_0\}$. Let $t = 0$.
- 2: PF measurement update: For $i = 1, \dots, N$, evaluate and normalize the importance weights using to the likelihood

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

Resample the particles if needed.

- 3: PF time update and Kalman filter update

(a) KF measurement update

$$\hat{x}_{t|t}^k = \hat{x}_{t|t-1}^k + K_t(y_t - h(x_t^p) - H(x_t^p)\hat{x}_{t|t-1}^k)$$

$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T$$

$$M_t = H(x_t^p)P_{t|t-1}H^T(x_t^p) + R_t$$

$$K_t = P_{t|t-1}H^T(x_t^p)M_t^{-1}.$$

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

$$x_{t+1}^p \sim \mathcal{N}\left(f^p(x_t^p) + F^p(x_t^p)\hat{x}_{t|t}^k, \right.$$

$$\left. F^p(x_t^p)P_{t|t}(F^p(x_t^p))^T + G^p(x_t^p)Q_t^p(G^p(x_t^p))^T\right)$$

(c) KF time update,

$$\hat{x}_{t+1|t}^k = F^k\hat{x}_{t|t}^k + f^k(x_t^p)$$

$$+ L_t(x_{t+1|t}^p - f^p(x_t^p) - F^p(x_t^p)\hat{x}_{t|t}^k)$$

$$P_{t+1|t} = F^k(x_t^p)P_{t|t}(F^k(x_t^p))^T + G^k(x_t^p)Q_t^k(G^k(x_t^p))^T$$

$$- L_t M_t L_t^T$$

$$M_t = F^k(x_t^p)P_{t|t}(F^k(x_t^p))^T + G^p(x_t^p)Q_t^p(G^p(x_t^p))^T$$

$$L_t = F^k(x_t^p)P_{t|t}(F^p(x_t^p))^T M_t^{-1}.$$

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

$H = 0$ and $F^p(x_t^p)$, $F^k(x_t^k)$, $G^p(x_t^p)$, and $G^k(x_t^k)$ are constant matrices, the covariance recursion is the same for all filters. Hence, this can be utilized in order to save computations.

2.2. RBPF — New Formulation

In [14] 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

¹F++: <http://www.control.isy.liu.se/resources/f++>

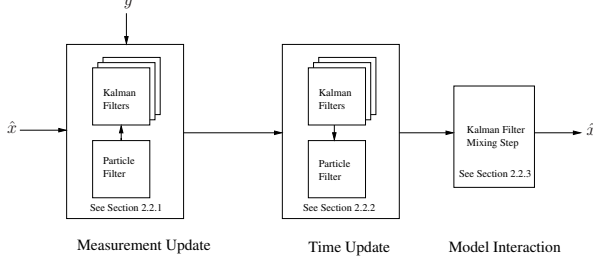


Fig. 1. Illustration of the new RBPF formulation. The internal dependency between the filters in the measurement and time update are indicated with arrows.

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 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, which has the nonlinear components as input, and the nonlinear effects are captured in a nonlinear model with only the nonlinear parts. The noise in the nonlinear model depends on the estimates from the linear model. This is illustrated in Fig. 1, and described in the subsequent sections. Of course, the above description is a bit overly simplified since it is sometimes motivated to put linear states in the nonlinear model. For a discussion about that, see [13]. 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

The model for the Kalman filters is obtained from model (2), using the conditional linear Gaussian property in (1) (i.e., x_t^p is considered known). The model has the state vector $\bar{x}^k = \begin{pmatrix} \xi \\ x^k \end{pmatrix}$, where ξ basically is x^p . It turns out that keeping ξ provides an easy and efficient way to compute many quantities needed in the algorithm. The linear and Gaussian model (when x_t^p is given) becomes:

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

$$= \bar{F}_t^k \bar{x}_t^k + \bar{G}_t^k \bar{w}_t^k + \bar{u}_t^k \quad (4b)$$

$$y_t = \begin{pmatrix} 0 & H(x_t^p) \end{pmatrix} \begin{pmatrix} \xi_t \\ x_t^k \end{pmatrix} + u_t^h + e_t = \bar{H}_t^k \bar{x}_t^k + u_t^h + e_t, \quad (4c)$$

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 $u_t^h = h(x_t^p)$, and $\bar{Q}_t = \begin{pmatrix} Q_t^p & 0 \\ 0 & Q_t^k \end{pmatrix}$.

2.2.2. Particle Filter Model

The nonlinear effects that are not taken care of in the linear model in Section 2.2.1 are put into a model to be handled by a PF. The resulting model for the model structure used in this paper is

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

$$y_t = \bar{e}_t, \quad (5b)$$

with the noise

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

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

where

$$S_t = H(x_t^p)P_{t|t-1}^k(H(x_t^p))^T + R_t, \quad (5e)$$

and where the covariance P^k and P^ξ are given in detail in (8) (Section 2.2.4). Here, $\hat{\xi}$ and \hat{x}^k are used to denote the estimate of the two parts of \hat{x} . At a first glance this seems to be a very simple model, however, there is a nontrivial connection to the linear model hidden in the noise description.

2.2.3. Model Interaction

The suggested formulation of the RBPF is now the one given in Algorithm 2. In it KF and PF are run in parallel followed by a mixing/combining step, which is performed after that both the KF and the PF have been time updated to adjust the KF based on the PF results:

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

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

For details regarding the covariance calculation refer to Section 2.2.4.

Also, see Section 2.2.5 for an alternative interpretation of this as a KF measurement update in response to a virtual measurement.

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.

Algorithm 2 Rao-Blackwellized PF (New formulation)

1: Initialization (same as standard formulation): For $i = 1, \dots, N$, $x_{0|-1}^{p(i)} \sim p_{x_0^p}(x_0^p)$ and set $\{\hat{x}_{0|-1}^{k(i)}, \bar{P}_{0|-1}^{(i)}\} = \left\{ \begin{pmatrix} x_0^p \\ x_0^k \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & P_0 \end{pmatrix} \right\}$. Let $t = 0$. ($^{(i)}$ is used to denote particle number, and $\hat{\xi}$ and \hat{x}^p indicate estimates of the parts of \hat{x}^k .)

2: Measurement update

(a) PF measurement update using (5): For $i = 1, \dots, N$, evaluate and normalize the importance weights using the likelihood

$$\begin{aligned} p(y_t | \mathbb{X}_t^p, \mathbb{Y}_{t-1}) &= \mathcal{N}(\hat{y}_t^{(i)}, S_t) \\ \hat{y}_t^{(i)} &= \bar{H}_t^{k(i)} \hat{x}_{t|t-1}^{k(i)} + u_t^{h(i)} \\ S_t^{(i)} &= \bar{H}_t^{k(i)} \bar{P}_{t|t-1}^{k(i)} (\bar{H}_{t|t-1}^{k(i)})^T + R_t, \end{aligned}$$

with $u_t^h = h(x_t^{p(i)})$.

(b) KF measurement update using (4) (depends on the PF):

$$\begin{aligned} \hat{x}_{t|t}^{k(i)} &= \bar{x}_{t|t-1}^{k(i)} - \bar{P}_t^{k(i)} (\bar{H}_{t|t-1}^{k(i)})^T (S_t^{(i)})^{-1} \epsilon_t^{(i)} \\ \epsilon_t^{(i)} &= y_t - \hat{y}_t^{(i)}, \\ \bar{P}_{t|t}^{k(i)} &= \bar{P}_{t|t-1}^{k(i)} \\ &\quad - \bar{P}_{t|t-1}^{k(i)} (\bar{H}_{t|t-1}^{k(i)})^T (S_t^{(i)})^{-1} \bar{H}_{t|t-1}^{k(i)} \bar{P}_{t|t-1}^{k(i)}. \end{aligned}$$

(c) Resample if necessary.

3: Time update

(a) KF time update using (4):

$$\begin{aligned} \hat{x}_{t+1|t}^{k(i)} &= \bar{F}_t^{k(i)} \hat{x}_{t|t}^{k(i)} + \bar{u}_t^{k(i)} \\ \bar{P}_{t+1|t}^{k(i)} &= \bar{F}_t^{k(i)} \bar{P}_{t|t}^{k(i)} (\bar{F}_t^{k(i)})^T + \bar{G}_t^{k(i)} \bar{Q}_t^k (\bar{G}_t^{k(i)})^T \end{aligned}$$

with $\bar{u}_t^{k(i)} = (f^p(x_t^{p(i)})^T f^k(x_t^{p(i)})^T)^T$.

(b) PF time update using (5) (depends on the KF):

$$x_{t+1}^{p(i)} \sim \mathcal{N}(\hat{\xi}_{t+1|t}^{(i)}, P_{t+1|t}^{\xi(i)})$$

(c) Mixing step, update the KF using (6).

$$\begin{aligned} \hat{x}_{t+1|t}^{k(i)} &:= \hat{x}_{t+1|t}^{k(i)} + P_{t+1|t}^{k\xi(i)} (P_{t+1|t}^{\xi(i)})^{-1} (x_{t+1}^{p(i)} - \hat{\xi}_{t+1|t}^{(i)}) \\ P_{t+1|t}^{k(i)} &:= P_{t+1|t}^{k(i)} - P_{t+1|t}^{k\xi(i)} (P_{t+1|t}^{\xi(i)})^{-1} P_{t+1|t}^{\xi k(i)}. \end{aligned}$$

4: Increase time and repeat from step 2.

Note again that for some special cases the covariance recursion is the same for all filters. Hence, this can be utilized in order to save computations. The computational complexity of different partitions and the complexity between PF and RBPF are analyzed extensively in [13].

Derivation of Algorithm 2

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

Applying the KF to the linear system (4) yields

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

and the covariance matrix $\bar{P}^{k\xi} := \begin{pmatrix} P^{\xi} & P^{\xi k} \\ P^{k\xi} & P^k \end{pmatrix}$ given by

$$P_{t+1|t}^{\xi} = F^p(x_t^p) P_{t|t}^{\xi} (F^p(x_t^p))^T + G^p(x_t^p) Q_t^p (G^p(x_t^p))^T \quad (7b)$$

$$P_{t+1|t}^k = F^k(x_t^p) P_{t|t}^k (F^k(x_t^p))^T + G^k(x_t^p) Q_t^k (G^k(x_t^p))^T \quad (7c)$$

$$P_{t+1|t}^{k\xi} = F^p(x_t^p) P_{t|t}^{k\xi} (F^k(x_t^p))^T \quad (7d)$$

$$P_{t+1|t}^{\xi k} = (P_{t+1|t}^{k\xi})^T. \quad (7e)$$

Applying the KF measurement update then yields

$$\begin{pmatrix} \hat{\xi}_{t|t} \\ \hat{x}_{t|t}^k \end{pmatrix} = \begin{pmatrix} \hat{\xi}_{t|t-1} \\ \hat{x}_{t|t-1}^k \end{pmatrix} + \begin{pmatrix} P_{t|t-1}^{\xi k} \\ P_{t|t-1}^k \end{pmatrix} (H(x_{t|t-1}^p))^{-1} S_t^{-1} \epsilon_t \quad (8a)$$

$$\epsilon_t = y_t - H(x_{t|t-1}^p) \hat{x}_{t|t-1}^k - u_t^h$$

$$S_t = H(x_{t|t-1}^p) P_{t|t-1}^k (H(x_{t|t-1}^p))^T + R_t.$$

with

$$P_{t|t}^{\xi} = P_{t|t-1}^{\xi} - P_{t|t-1}^{\xi k} (H(x_{t|t-1}^p))^{-1} S_t^{-1} H(x_{t|t-1}^p) P_{t|t-1}^{k\xi} \quad (8b)$$

$$P_{t|t}^k = P_{t|t-1}^k - P_{t|t-1}^k (H(x_{t|t-1}^p))^{-1} S_t^{-1} H(x_{t|t-1}^p) P_{t|t-1}^{k\xi} \quad (8c)$$

$$P_{t|t}^{k\xi} = P_{t|t-1}^{k\xi} - P_{t|t-1}^{k\xi} (H(x_{t|t-1}^p))^{-1} S_t^{-1} H(x_{t|t-1}^p) P_{t|t-1}^k \quad (8d)$$

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

The covariance follows from expanding the covariance measurement update in the KF for the extended linear system:

$$\begin{aligned} \begin{pmatrix} P_{t|t}^{\xi} & P_{t|t}^{\xi k} \\ P_{t|t}^{k\xi} & P_{t|t}^k \end{pmatrix} &= \begin{pmatrix} P_{t|t-1}^{\xi} & P_{t|t-1}^{\xi k} \\ P_{t|t-1}^{k\xi} & P_{t|t-1}^k \end{pmatrix} \\ &\quad - \begin{pmatrix} P_{t|t-1}^{\xi} & P_{t|t-1}^{\xi k} \\ P_{t|t-1}^{k\xi} & P_{t|t-1}^k \end{pmatrix} \begin{pmatrix} 0 \\ (H(\hat{x}_{t|t-1}^p))^T \end{pmatrix} S_t^{-1} \\ &\quad \begin{pmatrix} P_{t|t-1}^{\xi} & P_{t|t-1}^{\xi k} \\ P_{t|t-1}^{k\xi} & P_{t|t-1}^k \end{pmatrix} \begin{pmatrix} P_{t|t-1}^{\xi} & P_{t|t-1}^{\xi k} \\ P_{t|t-1}^{k\xi} & P_{t|t-1}^k \end{pmatrix}. \end{aligned}$$

The result follows directly from conducting the matrix multiplications block-wise.

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 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.

2.2.5. Alternative Interpretation

It should also be noted that it is possible to show an alternative interpretation of Algorithm 2. Let ξ be x^p , and then combine steps 3(b) and 3(c) in Algorithm 2 to one, where step 3(b) generates virtual measurements, $y^{p(i)}$, and look at 3(c) as a measurement update for the new measurement

$$y^p = (I \ 0) \bar{x}.$$

Furthermore, with this formulation, step 2(a) can be interpreted as a weighting step of a filter bank. Now the RBPF is in fact a conditional Kalman filter bank, with random virtual measurements, rather than a modified particle filter.

3. EXAMPLE

To exemplify the new RBPF algorithm formulation, the target tracking example from [13] is revisited.

3.1. Model

The problem of estimating the position and velocity of an aircraft is studied in a target tracking example using a simple 2D constant acceleration model, where the range and bearing to the aircraft are measured:

$$x_{t+1} = \begin{pmatrix} 1 & 0 & T & 0 & T^2/2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2/2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_t + w_t \quad (9a)$$

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

where 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, with sample period $T = 1$, and where r and φ are range and bearing measurements.

The dashed line indicate the division into the parts present in (2).

Using the notation in model (2) and the formulation in Section 2.2.1, the system can be written as:

$$F^p = \begin{pmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \end{pmatrix}, G^p = I_{2 \times 2}, G^k = I_{4 \times 4},$$

$$H = O_{2 \times 4}, u_t^h = h(x_t^p) = \begin{pmatrix} \sqrt{p_x^2 + p_y^2} \\ \arctan(p_y/p_x) \end{pmatrix}.$$

The input signal are $u_t^p = f^p(\bar{x}) = x_t^p$ and $u_t^k = f^k(\bar{x}) = 0$. The noise e_t is Gaussian with zero mean and covariance $R = \text{cov } e = \text{diag}(100, 10^{-6})$. The process noises are assumed Gaussian with zero mean and covariances

$$Q^p = \text{cov } w^p = \text{diag}(1, 1)$$

$$Q^k = \text{cov } w^k = \text{diag}(1, 1, 0.01, 0.01).$$

The nonlinear effects that are not taken care of in the linear model are put into a model to be handled by a PF according to (5).

The resulting linear model is therefore,

$$\bar{x}_{t+1}^k = \begin{pmatrix} 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \bar{x}_t^k + \begin{pmatrix} u_t^p \\ 0 \end{pmatrix} + w_t, \quad (10a)$$

$$y = u_t^h + e_t, \quad (10b)$$

and the nonlinear model follows immediately.

To verify the algorithm numerically the F++ software was used for Monte Carlo simulations using the above model structure. In Fig. 2 the position RMSE from 100 Monte Carlo simulations for the PF and the RBPF is depicted using $N = 2000$ particles. The computational complexity for RBPF vs PF for the described system is analyzed in detail in [13], and not part of this paper.

4. CONCLUSIONS

This paper presents the *Rao-Blackwellized particle filter* (RBPF) in a new 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 exemplified in a Monte Carlo study.

5. REFERENCES

- [1] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state esti-

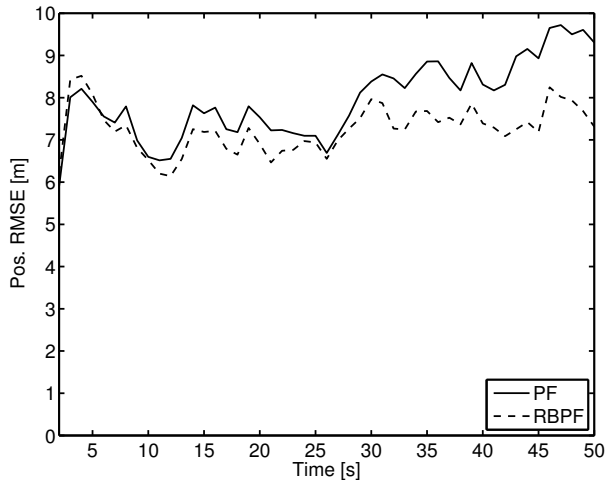


Fig. 2. Position RMSE for the target tracking example using 100 Monte Carlo simulation with $N = 2000$ particles. The PF estimates are compared to those from the RBPF.

mation,” *IEE Proceedings-F Radar and Signal Processing*, 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] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the American Society of Mechanical Engineering — Journal Basic Engineering, Series D*, vol. 82, pp. 35–45, Mar. 1960.
- [4] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice-Hall, Inc, 2000.
- [5] 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.
- [6] G. Casella and C. P. Robert, “Rao-Blackwellisation of sampling schemes,” *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996.
- [7] A. Doucet, N. J. Gordon, and V. Krishnamurthy, “Particle filters for state estimation of jump markov linear systems,” *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar. 2001.
- [8] R. Chen and J. S. Liu, “Mixture Kalman filter,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 62, no. 3, pp. 493–508, 2000.
- [9] C. Andrieu and A. Doucet, “Particle filter for partially observed Gaussian state space models,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 64, no. 4, pp. 827–836, 2002.

- [10] T. Schön, F. Gustafsson, and P.-J. Nordlund, “Marginalized particle filters for mixed linear / nonlinear state-space models,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.
- [11] T. B. Schön, R. Karlsson, and F. Gustafsson, “The marginalized particle filter in practice,” in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2006.
- [12] O. Cappé, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo,” *Proceedings of the IEEE*, 2007, (To Appear.).
- [13] R. Karlsson, T. B. Schön, and F. Gustafsson, “Complexity analysis of the marginalized particle filter,” *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4408–4411, Nov. 2005.
- [14] G. Hendeby and R. Karlsson, “Target tracking performance evaluation — a general software environment for filtering,” in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2007.

**Avdelning, Institution**

Division, Department

Division of Automatic Control
Department of Electrical Engineering**Datum**

Date

2007-08-13

Språk

Language

 Svenska/Swedish Engelska/English _____**Rapporttyp**

Report category

 Licentiatavhandling Examensarbete C-uppsats D-uppsats Övrig rapport _____**ISBN**

—

ISRN

—

Serietitel och serienummer

Title of series, numbering

ISSN

1400-3902

LiTH-ISY-R-2813

URL för elektronisk version<http://www.control.isy.liu.se>**Titel**

Title

A New Formulation of the Rao-Blackwellized Particle Filter

Författare

Author

Gustaf Hendeby, Rickard Karlsson, Fredrik Gustafsson

Sammanfattning

Abstract

For performance gain and efficiency it is important to utilize model structure in particle filtering. Applying Bayes' rule, present linear Gaussian substructure can be efficiently handled by a bank of Kalman filters. This is the standard formulation of the *Rao-Blackwellized particle filter* (RBPF), by some authors denoted the *marginalized particle filter* (MPF), and 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 reuse of 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 is exemplified in a target tracking simulation study.

Nyckelord

Keywords

Particle filter, Rao-Blackwellization, Kalman filter, Object oriented design, Marginalized particle filter

