16th IFAC Symposium on System Identification
The International Federation of Automatic Control
Brussels, Belgium. July 11-13, 2012

# A non-degenerate Rao-Blackwellised particle filter for estimating static parameters in dynamical models

**Fredrik Lindsten** * **Thomas B. Schön** * **Lennart Svensson** **

* *Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden (e-mail: {lindsten,schon}@isy.liu.se)*
** *Division of Signals and Systems, Chalmers University, SE-412 96 Gothenburg, Sweden (e-mail: lennart.svensson@chalmers.se)*

**Abstract:** The particle filter (PF) has emerged as a powerful tool for solving nonlinear and/or non-Gaussian filtering problems. When some of the states enter the model linearly, this can be exploited by using particles only for the "nonlinear" states and employing conditional Kalman filters for the "linear" states; this leads to the Rao-Blackwellised particle filter (RBPF). However, it is well known that the PF fails when the state of the model contains some static parameter. This is true also for the RBPF, *even if the static states are marginalised* analytically by a Kalman filter. The reason is that the posterior density of the static states is computed conditioned on the nonlinear particle trajectories, which are bound to degenerate over time. To circumvent this problem, we propose a method for targeting the posterior parameter density, conditioned on just the current nonlinear state. This results in an RBPF-like method, capable of recursive identification of nonlinear dynamical models with affine parameter dependencies.

## 1. INTRODUCTION

We consider the filtering problem for a certain type of nonlinear dynamical state-space models, with static state components. The typical case for when such models arise is when the state is augmented with some unknown, static parameter. This is common in e.g. simultaneous localisation and mapping [Thrun and Leonard, 2008] and in recursive, Bayesian parameter estimation [Ljung and Söderström, 1983].

Let $\{x_t\}_{t\geq 1}$ be the state process in a state-space model (SSM). That is, $\{x_t\}_{t\geq 1}$ is a discrete-time Markov process evolving according to a transition density $p(x_{t+1} \mid x_t, \theta)$. The states are hidden, but observed through the measurements $y_t$, according to the observation density $p(y_t \mid x_t, \theta)$. Here, $\theta$ is an unknown static parameter with prior density $p(\theta)$. For the purpose of joint estimation of $x_t$ and $\theta$, we augment the state with a static component $\theta_t \equiv \theta$. Hence, the SSM is described by,

$$x_{t+1} \sim p(x_{t+1} \mid x_t, \theta_t), \tag{1a}$$
$$\theta_{t+1} = \theta_t, \tag{1b}$$
$$y_t \sim p(y_t \mid x_t, \theta_t). \tag{1c}$$

Let $\xi_t = \{x_t, \theta_t\}$. We are interested in finding the joint filtering density $p(\xi_t \mid y_{1:t})$, i.e. the posterior density of the state $x_t$ and the parameter $\theta_t$ given a sequence of measurements $y_{1:t} \triangleq \{y_1, \ldots, y_t\}$. Let $p(\xi_1)$ be the joint prior density of the state and the parameter. The filtering density is then given by the Bayesian filtering recursions,

$$p(\xi_t \mid y_{1:t}) \propto p(y_t \mid \xi_t) p(\xi_t \mid y_{1:t-1}),$$
$$p(\xi_{t+1} \mid y_{1:t}) = \int p(\xi_{t+1} \mid \xi_t) p(\xi_t \mid y_{1:t})\, d\xi_t,$$

for any $t \geq 1$, using the convention $p(\xi_1 \mid y_{1:0}) = p(\xi_1)$.

Despite the simplicity of these expressions, they are known to be intractable for basically any model, except linear Gaussian state-space (LGSS) models and models with finite state-space. For general, nonlinear and/or non-Gaussian models, some approximate method for computing the filtering density is required. One popular approach is to use sequential Monte Carlo (SMC) methods, commonly referred to as particle filters (PFs); see e.g. [Doucet and Johansen, 2011, Gustafsson, 2010, Cappé et al., 2007]. However, it is well known that the PF will fail when the state contains some static parameter [Andrieu et al., 2004, Cappé et al., 2007]. The reason is that the exploration of the parameter space is restricted to the first time instant. Once the particles are initiated, their positions are fixed. At consecutive time points, the particles will be reweighted and resampled, but not moved to new positions.

In this paper we shall study the filtering problem for a special case of (1). More precisely, we assume that model is Gaussian with an affine dependence on the parameters,

$$x_{t+1} = f(x_t) + A(x_t)\theta_t + v_t, \tag{2a}$$
$$\theta_{t+1} = \theta_t, \tag{2b}$$
$$y_t = h(x_t) + C(x_t)\theta_t + e_t, \tag{2c}$$

where the process noise and measurement noise are white and Gaussian. Hence, conditioned on the *trajectory* $x_{1:t}$, the $\theta$-process is given by an LGSS model. For each time $t \geq 0$, the model can be seen as an LGSS with state $\theta_t$, if we fix the state trajectory $x_{1:t}$ up to that time. Hence, the conditional filtering density of $\theta_t$ given $x_{1:t}$ is Gaussian

⋆ This work was supported by the project Calibrating Nonlinear Dynamical Models (Contract number: 621-2010-5876) funded by the Swedish Research Council and CADICS, a Linneaus Center also funded by the Swedish Research Council.

and available through the Kalman filter (KF). Models with this property are known as conditionally linear Gaussian state-space (CLGSS) models. It is worth to note that the method proposed in subsequent sections is applicable to any CLGSS model, and can be of interest also when the "linear state" is non-static, but slowly mixing (see [Lindsten, 2011] for details). However, we have chosen to present the method for the special case (2) for clarity.

The conditionally linear Gaussian substructure in a CLGSS model can be exploited when addressing the filtering problem using SMC methods. This leads to the Rao-Blackwellised particle filter (RBPF); see Section 2 and [Liu and Chen, 1998, Doucet et al., 2000, Schön et al., 2005]. Basically, the idea behind the RBPF is to marginalise out the "linear" states (here $\theta_t$) analytically (using a KF), and employ particles only for the "nonlinear" state.

One might think that the RBPF will circumvent the problems that arise in the PF, when the static parameters can be marginalised out analytically. However, as we will see in the coming section, this is not the case, due to the degeneracy of the RBPF. In this paper, we address this problem and propose an RBPF-like method, suitable for handling CLGSS models with static state components.

*Remark 1.* An alternative approach to enable the application of the PF or the RBPF to a model with static parameters is to add some artificial dynamic evolution to the static state, and hope that this has negligible effect on the estimates. The artificial dynamics are often of random walk type, with a small and possibly decaying (over time) variance. This technique is sometimes called roughening or jittering. It is employed by for instance Gordon et al. [1993], Kitagawa [1998], Liu and West [2001], using the PF. Similarly, Schön and Gustafsson [2003] use jittering noise in an RBPF setting.

## 2. DEGENERACY OF THE RBPF – THE MOTIVATION FOR A NEW APPROACH

The presence of a conditionally linear Gaussian substructure can be exploited when addressing the filtering problem using SMC methods, leading to the RBPF by Doucet et al. [2000], Schön et al. [2005]. The RBPF utilises the fact that the joint filtering density can be expressed according to

$$p(x_t, \theta_t \mid y_{1:t}) = \int p(\theta_t \mid x_{1:t}, y_{1:t}) p(x_{1:t} \mid y_{1:t})\, dx_{1:t-1}. \tag{3}$$

Now, since the model under study is CLGSS, the first factor of the integrand above is Gaussian and analytically tractable, using the KF. More precisely, it holds that

$$p(\theta_t \mid x_{1:t}, y_{1:t}) = \mathcal{N}(\theta_t; \bar{\theta}_{t|t}(x_{1:t}), P_{t|t}(x_{1:t})), \tag{4}$$

for some (tractable) sequence of mean and covariance functions, $\bar{\theta}_{t|t}$ and $P_{t|t}$, respectively. Note that these are functions of the state trajectory $x_{1:t}$. Clearly, they also depend on the measurement sequence, but we shall not make that dependence explicit.

The second factor of the integrand in (3), referred to as the smoothing density, is targeted with an SMC sampler. This is done by generating a sequence of weighted particle systems $\{x^i_{1:t}, \omega^i_t\}_{i=1}^N$ for $t = 1, 2, \ldots$, each defining an empirical point-mass distribution approximating the smoothing distribution at time $t$ according to

$$p(dx_{1:t} \mid y_{1:t}) \approx \sum_{i=1}^{N} \omega^i_t \delta_{x^i_{1:t}}(dx_{1:t}), \tag{5}$$

where the importance weights $\{\omega^i_t\}_{i=1}^N$ are normalised to sum to one.

There exist a vast amount of literature, concerning how to generate such particle systems; see e.g. [Doucet and Johansen, 2011, Gustafsson, 2010, Cappé et al., 2007] for an in-depth treatment. The basic procedure is as follows. Assume that we have generated a weighted particle system $\{x^i_{1:t-1}, \omega^i_{t-1}\}_{i=1}^N$ targeting the smoothing distribution at time $t-1$. We then proceed to time $t$ by proposing new particles from a (quite arbitrary) proposal kernel $x^i_t \sim r_t(x_t \mid x^i_{1:t-1}, y_{1:t})$ for $i = 1, \ldots, N$. These samples are appended to the existing particle trajectories, i.e., $x^i_{1:t} := \{x^i_{1:t-1}, x^i_t\}$. The particles are then assigned importance weights according to

$$\omega^i_t \propto \omega^i_{t-1} \frac{p(y_t \mid x^i_{1:t}, y_{1:t-1}) p(x^i_t \mid x^i_{1:t-1}, y_{1:t-1})}{r_t(x^i_t \mid x^i_{1:t-1}, y_{1:t})}, \tag{6}$$

where the weights are normalised to sum to one. For a CLGSS model, the densities involved in the expression (6) are of known form. In particular, for the model (2) the densities in the numerator are both Gaussian; see [Schön et al., 2005] for details. Finally, when the sampling procedure outlined above is iterated over time, it is crucial to complement it with a resampling stage to avoid weight depletion [Cappé et al., 2007]. This has the effect of discarding particles with low weights and duplicating particles with high weights.

As indicated by (5), the SMC sampler does in fact generate weighted particle trajectories targeting the smoothing density $p(x_{1:t} \mid y_{1:t})$. However, due to the consecutive resampling steps, the particle trajectories will suffer from degeneracy; see e.g. [Cappé et al., 2007]. This means that the SMC method underlying the RBPF in general only can provide good approximations of the marginal filtering density $p(x_t \mid y_{1:t})$, or a fixed-lag smoothing density with a short enough lag. Hence, we are not able to provide any good approximation of the smoothing density, which in turn means that we do not have all the components required to approximate the joint filtering density by using (3).

To get around this, one often relies on the mixing of the system. More precisely, the linear state at time $t$ is supposed to be more or less independent of $x_{t-\ell}$, if the lag $\ell$ is large enough. If this is the case, we can obtain an accurate representation of the linear states despite the degeneracy of the nonlinear particle trajectories. Clearly, the success of this approach heavily depends on how good the mixing assumption is. In our case, where the linear state is static, the dependence of $\theta_t \mid \{x_{1:t}, y_{1:t}\}$ on $\{x_s, s \leq t-\ell\}$ can be substantial. That is, if the approximation of the density $p(x_{1:t-\ell} \mid y_{1:t})$ is poor, using (3) to compute the joint filtering density can give very poor results. We illustrate the RBPF degeneracy problem in Example 1.

*Example 1.* (RBPF for a partially static system). The first order LGSS system,

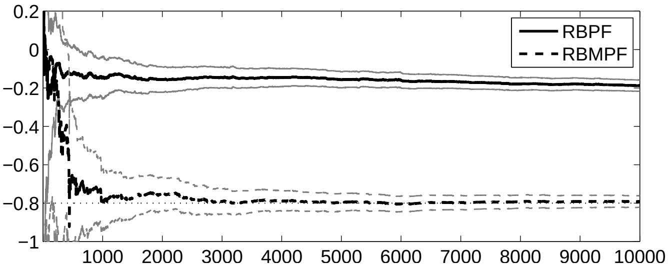


Fig. 1. RBPF and RBMPF (Algorithm 1) estimates of $\theta_t$ (thick black lines) and the estimated $3\sigma$-confidence interval (thin grey lines), as function of time $t$. The true value is $-0.8$.

$$x_{t+1} = ax_t + v_t, \qquad v_t \sim \mathcal{N}(0, 0.1),$$
$$y_t = x_t + e_t, \qquad e_t \sim \mathcal{N}(0, 1),$$

where $a = -0.8$, is simulated for $T = 10000$ time steps. We wish to estimate the parameter $a$ recursively and we therefore augment the state with $\theta_t \equiv a$. The prior distribution of $a$ is taken as Gaussian with mean $\bar{\theta}_{1|0} = 1$ and covariance $\bar{P}_{1|0} = 3$. A bootstrap RBPF using $N = 100$ particles is applied to the data and the results is shown in Figure 1.

The initial distribution of $\theta_1$ should have negligible effect at time $T = 10000$. Hence, we expect the estimate to converge to the "true" value $-0.8$. As can be seen in the figure, this is not the case. Also, the estimated confidence interval is way too small, i.e. the covariance of the linear state is underestimated. The intuitive explanation is that the RBPF particle trajectories, in some sense, degenerate faster than the estimate of $\theta_t$ converges. As a comparison, we also show the estimate provided by the proposed method, given in the consecutive section [1]. □

## 3. A NON-DEGENERATE RBPF FOR MODELS WITH STATIC PARAMETERS

In this paper, we propose an alternative to (3), which is to factorise the joint filtering density as,

$$p(x_t, \theta_t \mid y_{1:t}) = p(\theta_t \mid x_t, y_{1:t})p(x_t \mid y_{1:t}). \tag{7}$$

The marginal filtering density $p(x_t \mid y_{1:t})$ can be approximated using SMC without suffering from degeneracy. Thus, an approximation of the joint filtering density based on the factorisation (7), does not rely on the mixing properties of the system. However, as opposed to $p(\theta_t \mid x_{1:t}, y_{1:t})$ given in (4), the density

$$p(\theta_t \mid x_t, y_{1:t}), \tag{8}$$

is in general non-Gaussian and intractable. The problem we face is thus to find an appropriate way to approximate (8), while still enjoying the benefits of a Rao-Blackwellised setting.

Since this approach resembles the RBPF, but is based on the marginal density $p(x_t, \theta_t \mid y_{1:t})$ rather than the density $p(x_{1:t}, \theta_t \mid y_{1:t})$ it will be referred to as the Rao-Blackwellised marginal particle filter (RBMPF). We start the presentation of the RBMPF with a discussion on how to sample from the marginal filtering density. After this, we turn to the more central problem of approximating the conditional filtering density (8).

### 3.1 Sampling from the marginals

As indicated by (7), we wish to target the marginal filtering density $p(x_t \mid y_{1:t})$ with an SMC sampler. In fact, one way to do this is to perform the sampling exactly as in the RBPF, and then simply discard the particle trajectories up to time $t-1$. However, here we outline a different approach, inspired by the marginal particle filter (MPF) [Klaas et al., 2005].

Assume that we have completed the sampling at time $t-1$. We have thus generated a weighted particle system $\{x_{t-1}^j, \omega_{t-1}^j\}_{j=1}^N$ targeting $p(x_{t-1} \mid y_{1:t-1})$. Similarly to [Klaas et al., 2005] [2], we then construct a proposal as a mixture density

$$r'_t(x_t \mid y_{1:t}) = \sum_{j=1}^N \omega_{t-1}^j r_t(x_t \mid x_{t-1}^j, y_{1:t}), \tag{9}$$

from which we draw a set of new particles $\{x_t^i\}_{i=1}^N$. To compute the importance weights, i.e. the quotient between the target and the proposal densities, we note that the target density can be expanded according to,

$$p(x_t \mid y_{1:t}) = \int \frac{p(y_t \mid x_{t-1:t}, y_{1:t-1})p(x_t \mid x_{t-1}, y_{1:t-1})}{p(y_t \mid y_{1:t-1})} p(x_{t-1} \mid y_{1:t-1})\, dx_{t-1}.$$

By approximating $p(x_{t-1} \mid y_{1:t-1})$, using the weighted particles given at time $t-1$, we get

$$p(x_t \mid y_{1:t}) \approx \sum_{j=1}^N \omega_{t-1}^j \frac{p(y_t \mid x_{t-1}^j, x_t, y_{1:t-1})p(x_t \mid x_{t-1}^j, y_{1:t-1})}{p(y_t \mid y_{1:t-1})}.$$

Using the above approximation, we can compute the importance weights according to,

$$\omega_t^i \propto \frac{\sum_{j=1}^N \omega_{t-1}^j p(y_t \mid x_{t-1}^j, x_t^i, y_{1:t-1})p(x_t^i \mid x_{t-1}^j, y_{1:t-1})}{\sum_{j=1}^N \omega_{t-1}^j r_t(x_t^i \mid x_{t-1}^j, y_{1:t})}, \tag{10}$$

where the weights are normalised to sum to one.

### 3.2 Gaussian mixture approximation

We now turn to the more central problem in the RBMPF, namely to find an approximation of the density (8). The general idea that we will employ is to approximate it as Gaussian. Hence, let us assume that, for some $t \geq 2$,

$$p(\theta_{t-1} \mid x_{t-1}, y_{1:t-1}) \approx \widehat{p}(\theta_{t-1} \mid x_{t-1}, y_{1:t-1}) \triangleq \mathcal{N}\left(\theta_{t-1}; \bar{\theta}_{t-1|t-1}(x_{t-1}), P_{t-1|t-1}(x_{t-1})\right), \tag{11}$$

for some mean and covariance functions, $\bar{\theta}_{t-1|t-1}$ and $P_{t-1|t-1}$, respectively. At time $t = 2$, no approximation is needed, since (11) then coincides with (4).

Just as in the standard RBPF, if we augment the conditioning on the nonlinear state to $x_{t-1:t}$, and make a time update and a measurement update of (11), we obtain

[1] We emphasise that this example is provided as an illustration of the concept, and not an evaluation of the proposed method. We have only considered one realisation of data, which of course is not enough to draw any general conclusions. A more rigorous numerical evaluation is given in Section 4.

[2] The difference is that Klaas et al. [2005] targets $p(\xi_t \mid y_{1:t})$ rather than $p(x_t \mid y_{1:t})$.

$$\widehat{p}(\theta_t \mid x_{t-1}, x_t, y_{1:t}) = \mathcal{N}\left(\theta_t; \tilde{\theta}_{t|t}(x_{t-1:t}), \widetilde{P}_{t|t}(x_{t-1:t})\right), \tag{12}$$

for some mean and covariance functions, $\tilde{\theta}_{t|t}$ and $\widetilde{P}_{t|t}$, respectively. The problem is that once we "remove" the conditioning on $x_{t-1}$, the Gaussianity is lost. Hence, to obtain a recursion, i.e. to end up with (11) with time index $t-1$ replaced by $t$, we need to find a Gaussian approximation of $p(\theta_t \mid x_t, y_{1:t})$ based on (12).

To achieve this, we start by noting that the sought density (8) can be written,

$$\begin{aligned} p(\theta_t \mid x_t, y_{1:t}) &= \int p(\theta_t \mid x_{t-1:t}, y_{1:t}) p(x_{t-1} \mid x_t, y_{1:t})\, dx_{t-1} \\ &= \int p(\theta_t \mid x_{t-1:t}, y_{1:t}) \frac{p(y_t \mid x_{t-1:t}, y_{1:t-1}) p(x_t \mid x_{t-1}, y_{1:t-1})}{p(x_t, y_t \mid y_{1:t-1})} \\ &\qquad \times p(x_{t-1} \mid y_{1:t-1})\, dx_{t-1}. \end{aligned} \tag{13}$$

At time $t-1$, we have acquired a weighted particle system $\{x_{t-1}^j, \omega_{t-1}^j\}_{j=1}^N$ targeting the marginal filtering density $p(x_{t-1} \mid y_{1:t-1})$. By plugging this into (13), conditioned on $x_t^i$, we obtain,

$$p(\theta_t \mid x_t^i, y_{1:t}) \approx \sum_{j=1}^{N} \gamma_t^{j,i} p(\theta_t \mid x_{t-1}^j, x_t^i, y_{1:t}), \tag{14a}$$

with,

$$\gamma_t^{j,i} \triangleq \frac{\omega_{t-1}^j p(y_t \mid x_{t-1}^j, x_t^i, y_{1:t-1}) p(x_t^i \mid x_{t-1}^j, y_{1:t-1})}{\sum_{k=1}^N \omega_{t-1}^k p(y_t \mid x_{t-1}^k, x_t^i, y_{1:t-1}) p(x_t^i \mid x_{t-1}^k, y_{1:t-1})}. \tag{14b}$$

Furthermore, by the Gaussianity assumption (12), we see that (14) is a Gaussian mixture model (GMM). Recall that we seek to approximate the left hand side of (14a) with a single Gaussian. To keep the full GMM representation is generally not an option, since this would result in a mixture with a number of components increasing exponentially over time. Hence, we propose to approximate the GMM with a single Gaussian, by using moment matching. From (12), the mean and covariance of the GMM (14a) are given by,

$$\bar{\theta}_{t|t}(x_t^i) = \sum_{j=1}^{N} \gamma_t^{j,i} \tilde{\theta}_{t|t}^{j,i}, \tag{15a}$$

$$P_{t|t}(x_t^i) = \sum_{j=1}^{N} \gamma_t^{j,i} \left( \widetilde{P}_{t|t}^{j,i} + (\tilde{\theta}_{t|t}^{j,i} - \bar{\theta}_{t|t}^i)(\tilde{\theta}_{t|t}^{j,i} - \bar{\theta}_{t|t}^i)^{\mathsf{T}} \right), \tag{15b}$$

respectively. Here we have used the shorthand notation $\tilde{\theta}_{t|t}^{j,i}$ instead of $\tilde{\theta}_{t|t}(x_{t-1}^j, x_t^i)$, etc. In conclusion, the above results provide a Gaussian approximation of (8) according to, $\widehat{p}(\theta_t \mid x_t^i, y_{1:t}) \triangleq \mathcal{N}\left(\theta_t; \bar{\theta}_{t|t}(x_t^i), P_{t|t}(x_t^i)\right)$.

### *3.3 Resulting Algorithm*

The procedure outlined in the previous two sections provides an RBPF-like method targeting the filtering density using the factorisation (7). To be able to carry out the steps of this method, we require $p(y_t \mid x_{t-1}, x_t, y_{1:t-1})$ and $p(x_t \mid x_{t-1}, y_{1:t-1})$ to be available for evaluation, since these are used in (10) and in (14b), to compute the particle weights and the mixing weights, respectively (note the similarity between the two expressions). Strictly speaking, these densities are *not* analytically tractable in the general case, since we condition on just $x_{t-1:t}$ and not the full nonlinear state trajectory $x_{1:t}$ (cf. the densities appearing in the RBPF weight expression (6), which are tractable for any CLGSS model).

However, this will in fact not be an issue in the RBMPF setting. The reason is that the conditional filtering density for the linear state is approximated by a Gaussian at time $t-1$, according to (11). Given this approximation, conditioning on just $x_{t-1}$ in the RBMPF, will have the "same effect" as conditioning on $x_{1:t-1}$ in the RBPF. Hence, the densities appearing in (10) and (14b) will indeed be available for evaluation, under this Gaussianity approximation (see [Lindsten, 2011] for details). It can be said that the whole idea with the RBMPF, is to replace the conditioning on the nonlinear state trajectory, with a conditioning on the nonlinear state at a single time point. We summarise the RBMPF method in Algorithm 1.

**Algorithm 1** RBMPF (one time step)

Sample particles, $\{x_t^i\}_{i=1}^N$ from (9).
**for** $i = 1$ **to** $N$ **do**
  **for** $j = 1$ **to** $N$ **do**
    Compute the mean $\tilde{\theta}_{t|t-1}^{j,i}$ and the covariance $\widetilde{P}_{t|t}^{j,i}$ of the density (12), conditioned on $\{x_{t-1}^j, x_t^i\}$, using RBPF time and measurement updates.
    Compute the mixture weights $\gamma_t^{j,i}$ according to (14b). The involved densities are available from the RBPF time and measurement updates.
  **end for**
  Compute the mean $\bar{\theta}_{t|t}^i$ and covariance $P_{t|t}^i$ of the GMM according to (15).
  Compute the importance weights $\{\omega_t^i\}_{i=1}^N$ according to (10).
**end for**

## 4. NUMERICAL ILLUSTRATION

In this section we evaluate the RBMPF method for recursive identification on simulated data. We will consider the first order nonlinear system,

$$x_{t+1} = a x_t + b \frac{x_t}{1 + x_t^2} + c \cos(1.2t) + v_t, \tag{16a}$$

$$y_t = d x_t^2 + e_t, \tag{16b}$$

with $v_t \sim \mathcal{N}(0, 0.01)$ and $e_t \sim \mathcal{N}(0, 0.1)$. The initial state of the system is $x_1 \equiv 0$. The true parameters are given by $\theta^\star = (a\ b\ c\ d)^{\mathsf{T}} = (0.5\ 25\ 8\ 0.05)^{\mathsf{T}}$. This system has been studied e.g. by Gordon et al. [1993] and has become something of a benchmark example for nonlinear filtering. By augmenting the model with the static parameter state $\theta_t \equiv (a\ b\ c\ d)^{\mathsf{T}}$, we obtain a fifth order mixed linear/nonlinear system, where four of the states are conditionally linear. We can thus employ the RBMPF given in Algorithm 1 for recursive parameter estimation.

The RBMPF is compared with the RBPF, where the latter uses jittering noise as discussed in Remark 1. As suggested by Schön and Gustafsson [2003], we apply Gaussian jittering noise with decaying variance on both states and parameters. These artificial noise sources are Gaussian with time-decaying variances, $\sigma_x^2/t$ and $(\sigma_\theta^2/t)I_{4\times 4}$, respectively. Of course, the jittering noises are internal to the RBPF and are not used when simulating data from the system.

The evaluation was made by a Monte Carlo study over 100 realisations of data $y_{1:T}$ from the system (16), each consisting of $T = 200$ measurements. The parameters were modeled as Gaussian random variables, $\theta_1 \sim \mathcal{N}(\bar{\theta}_{1|0}, \text{diag}(0.5, 25, 8, 0.05))$. Here, $\bar{\theta}_{1|0}$ corresponds to the prior mean of the parameter. This vector was chosen randomly for each Monte Carlo simulation, uniformly over the intervals $\pm 50$ % from the true parameter values. The RBMPF and four versions of the RBPF were run in parallel, all using $N = 500$ particles. The first RBPF did not use any jittering noise, whereas the remaining three versions used jittering noise with $(\sigma_x^2 = \sigma_\theta^2 = \sigma^2)$, $\sigma^2 = 0.01$, $\sigma^2 = 0.1$ and $\sigma^2 = 1$, respectively. Furthermore, to increase the numerical robustness, a weight threshold was implemented in the filters. That is, if the sum of the unnormalised importance weights was below a certain threshold, here $10^{-12}$, the particles were discarded, the filter "rewinded" a few time step and new particles were generated.

Table 1 summarises the results from the different filters, in terms of the Monte Carlo means and standard deviations for the parameter estimates extracted at the final time point $t = T = 200$. Also, the rightmost column of the table shows the percentage of data realisations, in which the weight threshold (as mentioned above) was hit at least once. We conclude that jittering noise with $\sigma^2 = 0.1$ provides the best tuning for the RBPF, among the values considered here. The results from this filter and from the RBMPF, over the 100 realisations of data, are given in Figure 2. It is clear that the jittering noise in the RBPF introduces extra variance to the estimates and also that it slows down the convergence, when compared to the RBMPF. Furthermore, from Table 1 we see that the accuracy of the RBPF is highly dependent on the variance of the jittering. Tuning of this parameter can be problematic in a real world scenario. The absence of a jittering noise which needs to be tuned properly, is one of the main advantages with the RBMPF over the RBPF.

## 5. DISCUSSION AND FUTURE WORK

One of the main drawbacks with the RBMPF method is that it has quadratic complexity in the number of particles, as opposed to the RBPF, which has linear complexity. In fact, just as the RBPF can be seen as using $N$ parallel Kalman filters, the RBMPF uses $N^2$ Kalman filters. In this way, by viewing each particle as a separate model, the RBMPF very much resembles the 2$^{\text{nd}}$ order, generalised pseudo-Bayesian (GPB2) multiple model filter. Guided by this insight, we could also derive an RBMPF similar to the GPB1 filter (see [Bar-Shalom et al., 2001] for the two GPB filters). This would reduce the complexity to grow linearly with $N$, but at the cost of coarser approximations, likely to degrade the performance of the filter. A third approach in this direction, is to start from the interacting multiple model (IMM) filter by Blom and Bar-Shalom [1988], which is a popular multiple model filter, since it has lower complexity than GPB2 (still quadratic, but smaller constants), but is known to have similar performance [Blom and Bar-Shalom, 1988]. However, it is not clear that the ideas underlying the IMM filter, can be straightforwardly generalised to the RBMPF. This issue requires further attention.

Another way to reduce the complexity of the algorithm is by numerical approximations of the mixture models. Due to the exponential decay of the Gaussian components, truncation might aid in making fast, sufficiently accurate, evaluations of the GMM moments. A related approach, which could be adapted to the RBMPF, is used by Gray and Moore [2000, 2003] for fast, nonparametric density estimation. Also, fast summation methods, similar to the ideas underlying the fast Gauss transform by Greengard and Strain [1991], Greengard and Sun [1998], might be of use. However, as discussed by Boyd [2010], truncation methods should in general have more to offer than fast summation methods, for Gaussian components which are quickly decaying.

Finally, another option is of course to seek alternative approximations of the conditional filtering density (8), not based on a GMM as in (14). By doing so, one can possibly find good approximations, which can be evaluated more efficiently than the ones presented here.

## 6. CONCLUSIONS

The application of particle filters (PFs) for estimating static parameters is a well known and challenging problem. For models where the parameters enter linearly, they can be marginalised out analytically, by using conditional Kalman filters, leading to the Rao-Blackwellised particle filter (RBPF). However, this will not remedy the problem, as the degeneracy of the particle trajectories in the RBPF will result in erroneous parameter estimates. This issue can be traced back to the expression of the filtering density (3), which is the basis for the RBPF. When this form is used to approximate the filtering density, good accuracy is obtained only when the model under study is mixing sufficiently fast; this is not the case when the state is augmented with static parameters. Here, we have proposed a different factorisation of the filtering density, according to (7). By using a particle representation of the marginal filtering distribution for the nonlinear state, a Gaussian mixture arises as the natural approximation of the conditional filtering density for the parameters. To obtain a recursive method, we propose to approximate this mixture density with a single Gaussian, by using moment matching. This results in an RBPF-like method, suitable for recursive identification of nonlinear dynamical systems with affine parameter dependence. The main drawback with the proposed method is its computational complexity, which grows quadratically with the number of particles. However, we believe that this can be reduced significantly by using truncation techniques, motivated by the exponential decay of the Gaussian components.

## REFERENCES

C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić. Particle methods for change detection, system identification, and contol. *Proceedings of the IEEE*, 92(3):423–438, 2004.

Table 1. Monte Carlo means and standard deviations

| Method | $a$ ($\times 10^{-1}$) | $b$ | $c$ | $d$ ($\times 10^{-2}$) | Rew. (%) |
|---|---|---|---|---|---|
| True value ($\theta^\star$) | 5 | 25 | 8 | 5 | – |
| RBPF ($\sigma^2 = 0$) | $4.99 \pm 0.122$ | $24.9 \pm 3.84$ | $7.92 \pm 0.613$ | $5.26 \pm 1.056$ | 52 |
| RBPF ($\sigma^2 = 0.01$) | $4.95 \pm 0.090$ | $24.6 \pm 1.60$ | $7.95 \pm 0.258$ | $5.21 \pm 0.442$ | 4 |
| RBPF ($\sigma^2 = 0.1$) | $4.93 \pm 0.480$ | $22.7 \pm 0.91$ | $7.60 \pm 0.198$ | $5.84 \pm 0.669$ | 1 |
| RBPF ($\sigma^2 = 1$) | $4.73 \pm 0.287$ | $18.8 \pm 0.95$ | $6.45 \pm 0.300$ | $8.60 \pm 0.656$ | 0 |
| RBMPF | $5.00 \pm 0.030$ | $25.2 \pm 1.00$ | $8.05 \pm 0.146$ | $4.94 \pm 0.240$ | 11 |

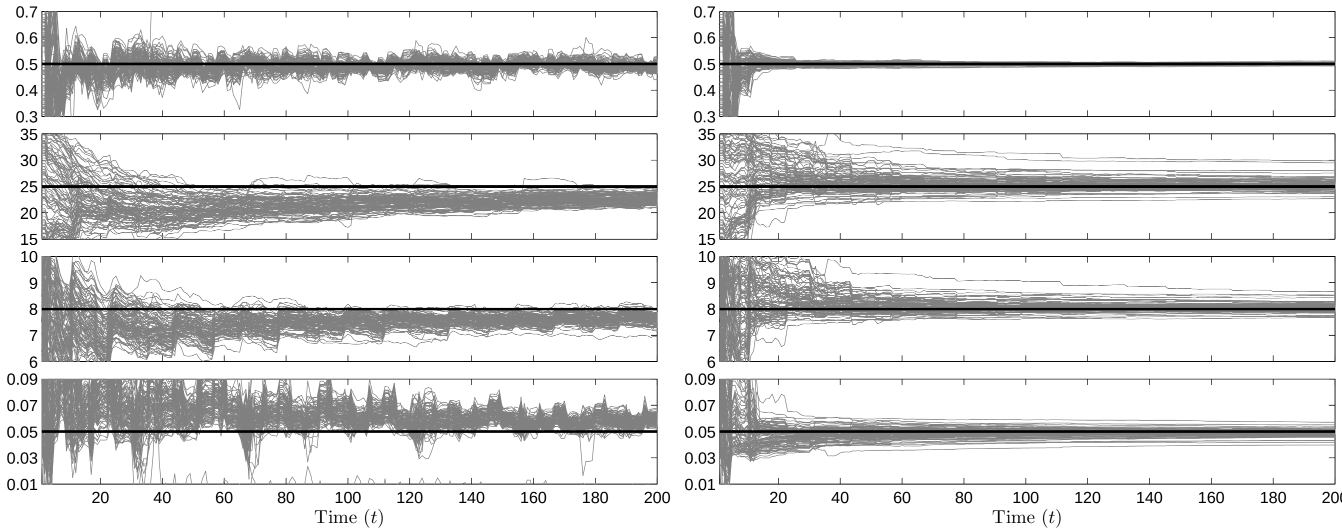


Fig. 2. Estimates of the parameters $a$, $b$, $c$ and $d$ (from top to bottom). The grey lines illustrate the estimates for the 100 realisations of data. The true parameter values are shown as thick black lines. *Left:* RBPF using jittering noise with $\sigma_x^2 = \sigma_\theta^2 = 0.1$. *Right:* RBMPF.


Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, 2001.

H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.

J. P. Boyd. The uselessness of the fast Gauss transform for summing Gaussian radial basis function series. *Journal of Computational Physics*, 229:1311–1326, 2010.

O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook of Nonlinear Filtering.* Oxford University Press, 2011.

A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107 –113, 1993.

A. G. Gray and A. W. Moore. 'n-body' problems in statistical learning. In *Proceedings of the 2000 Conference on Neural Information Processing Systems (NIPS)*, 2000.

A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *Proceedings of the SIAM International Conference on Data Mining*, 2003.

L. Greengard and S. Strain. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.

L. Greengard and X. Sun. A new version of the fast Gauss transform. *Documenta Mathematica*, Extra Volume ICM III:575–584, 1998.

F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, 1998.

M. Klaas, N. de Freitas, and A. Doucet. Toward practical $n^2$ Monte Carlo: the marginal particle filter. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

F. Lindsten. *Rao-Blackwellised particle methods for inference and identification.* Licentiate thesis no. 1480, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, June 2011.

J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice.* Springer Verlag, 2001.

J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443): 1032–1044, 1998.

L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* MIT Press, 1983.

T. Schön and F. Gustafsson. Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID)*, 2003.

T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, 2005.

S. Thrun and J. Leonard. Simultaneous localization and mapping. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 37, pages 871–889. Springer, 2008.