SMC Methods to Avoid Self-resolving for Online Bayesian Parameter Estimation

Edson Hiroshi Aoki*, Yvo Boers†, Pranab K. Mandal* and Arunabha Bagchi*
*Department of Applied Mathematics, University of Twente, Enschede, The Netherlands
Email: {e.h.aoki, p.k.mandal, a.bagchi}@ewi.utwente.nl
†Thales Nederland B. V., Hengelo, The Netherlands
Email: yvo.boers@nl.thalesgroup.com

***Abstract*—The particle filter is a powerful filtering technique that is able to handle a broad scope of nonlinear problems. However, it has also limitations: a standard particle filter is unable to handle, for instance, systems that include static variables (parameters) to be estimated together with the dynamic states. This limitation is due to the well-known "self-resolving" phenomenon, which is caused by the gradual loss of information that occurs during the resampling steps. In the context of online Bayesian parameter estimation, some approaches to handle this problem have proposed, such as adding artificial dynamics to the parameter model. However, these approaches typically both introduce new parameters (e.g. the intensity of artificial process noise) and inherent biases to the estimation problem.**

In this paper, we will give a give a look at two Sequential Monte Carlo techniques that do not rely on biasing the system model: the Autonomous Multiple Model particle filter and the Rao-Blackwellized Marginal particle filter. These approaches are not new, but have not been applied yet to the problem of online Bayesian parameter estimation for non-structured models. We will derive suitable adaptations of these methods for this problem and evaluate them using simulations.

I. Introduction

In a Bayesian estimation problem, we refer to "self-resolving" as a situation where uncertainties and ambiguities, that exist in the exact posterior distribution, are for some reason not reflected in the output of the estimation algorithm. The phenomenon is known to affect Sequential Monte Carlo (SMC) methods, also known as particle filters, for certain classes of problems, and it is commonly referred as "degeneracy" in particle filter literature.

Vermaak, Doucet and Pérez [1] have observed that a particle filter applied to a multi-modal distribution does not consistently maintain this multi-modality. Other problems that have been identified to suffer from degeneracy include classification [2], smoothing [3], track labelling [4], and our problem of interest, parameter estimation [5].

For that reason, most proposed methods for parameter estimation based on particle filters are offline and/or non-Bayesian; an excellent survey about these methods has been made by Kantas *et al.* [5]. A few approaches to online Bayesian parameter estimation, described in the same survey, have been proposed. These include introducing artificial dynamics to the parameter [6] and filtering using a fixed-lag approximation [7]. However, since these methods operate by biasing the system model, one may question whether these methods can produce accurate representations of the true posterior distribution of the parameters. This may be extremely relevant if we are interested not only in obtaining point estimates for the parameters, but also other statistical information, such as the standard deviation.

In this paper, we will present two SMC approaches for online Bayesian parameter estimation, that do not rely on biasing the system model. One is the Autonomous Multiple Model particle filter (AMMPF), a static version of the Interacting Multiple Model particle filter (IMMPF), which has been proposed by Blom and Bloem [2] for the classification problem. Another is the Rao-Blackwellized marginal particle filter (RBMPF), which has been proposed by Lindsten, Schön and Svensson [8] for parameter estimation and smoothing for a certain class of structured system models. We will derive an extension of the algorithm for general non-structured models containing both dynamic states and parameters.

This work is organized as follows. Section II is a review of the "self-resolving" phenomenon in particle filters and its effect on parameter estimation. Sections III and IV present respectively the AMMPF and RBMPF SMC approaches. Section V presents simulations where we compare both methods with the artificial dynamics approach. Section VI draws conclusions.

II. The self-resolving phenomenon in the context of parameter estimation

Let us consider a standard, Sequential Importance Resampling (SIR) particle filter that estimates a stochastic process X_k using N_P particles. Each particle, say with index i, represents an hypothesis $x_0(i), \ldots, x_k(i)$ on the trajectory of the state (in this work, as usual, we use uppercase letters to represent random variables, and lowercase letters to represent realizations of these random variables). Periodic resampling, which consists of eliminating hypotheses and replicating others, causes all particles to eventually assume a common past trajectory, i.e. all hypotheses to have the form

$$x_0, \ldots, x_{j_{\text{sr}}}, x_{j_{\text{sr}+1}}(i), \ldots, x_k(i) \tag{1}$$

for some $0 \leq j_{\text{sr}} \leq k$. Effectively, the particle set, instead of approximating the true posterior $p(x_0, \ldots, x_k|Z^k)$ (where Z^k denotes all observations up to and including time k), becomes

biased towards

$$p(x_{j_{\text{sr}}+1}, \ldots, x_k | x_0, \ldots, x_{j_{\text{sr}}}, Z^k). \quad (2)$$

In filtering, we are interested in the expectation of some function $g(x_k)$ of the current state, conditioned on the available information Z^k; we denote this conditional expectation as $E_{Z^k}[g(X_k)] = \int g(x_k) p(x_k|Z^k) dx_k$. If we have the "forgetness condition"

$$E_{Z^k}[g(X_k)] \approx \int g(x_k) p(x_k | x_0, \ldots, x_{j_{\text{sr}}}, Z^k) dx_k \quad (3)$$

i.e. the condition that the filtering process gradually "forgets" old trajectories, then the bias does not have significant impact[1].

Unfortunately, it is obvious that this is not the case in the parameter estimation problem. Let us suppose that the state X_k has form $X_k = [S_k', \Theta']'$, where S_k denotes the dynamic part of the state, Θ corresponds to one or more parameters, and $'$ denotes the transpose operator. Let θ^* be the value of Θ assumed in the past trajectory $x_0, \ldots, x_{j_{\text{sr}}}$. It is straightforward to see that if we attempt to compute the mean value of Θ using the biased density (2), we will obtain $\hat{\theta} = \theta^*$. Besides, if we attempt to calculate the variance of Θ using the same biased density, we will obtain

$$\begin{aligned}
\sigma_\theta^2 &= \int (\theta - \theta^*)^2 \, p(x_k | x_0, \ldots, x_{j_{\text{sr}}}, Z^k) dx_k \\
&= \int (\theta - \theta^*)^2 \, p(\theta | x_0, \ldots, x_{j_{\text{sr}}}, Z^k) d\theta \\
&= \int (\theta - \theta^*)^2 \, \delta(\theta - \theta^*) d\theta \\
&= 0. \quad (4)
\end{aligned}$$

It is important to remark that self-resolving cannot be avoided by using standard particle filtering tuning techniques, i.e. increasing the number of particles or using a better importance sampling function. At best, these techniques can postpone self-resolving (i.e. decrease j_{sr}) which may make the point estimation error close to optimal but will unavoidably result in the variance of the estimated parameters to be eventually assumed to be zero.

III. The Autonomous Multiple Model particle filter

The Autonomous Multiple Model particle filter (AMMPF) that we are going to present consists of multiple particle filters running in parallel, each assuming a particular hypothesis on the value of the parameter vector. Since the true value of the parameter vector does not change with time, the AMMPF can be considered, as we mentioned earlier, a "static" version of the IMMPF presented in [2].

Assuming again that the state X_k has form $X_k = [S_k', \Theta']'$, where S_k and Θ denote respectively the dynamic and static parts of the state, the algorithm requires all components of the parameter vector Θ to be discrete. However, if the cardinality of Θ is sufficiently small, we may perform discretization of the parameter space as an approximation.

In order to explain the filter, let us first consider a standard, "plain vanilla" particle filter. Let $p(x_0, \ldots, x_k|Z^k)$ be the trajectory probability density, where Z^k is the collection of measurements up to and including time k. The density is approximated by a set of particles $\{u_k(i), w_k(i)\}_{i=1}^{N_P}$ (where $u_k(i) = (x_0(i), \ldots, x_k(i))$, $w_k(i)$ is the particle weight, and N_P is the number of particles). The expectation of a function $g(x_k)$ conditioned on Z^k, in the standard PF, is approximated according to

$$\begin{aligned}
E_{Z^k}[g(X_k)] &= \int g(x_k) p(x_0, \ldots, x_k | Z^k) dx_0 \ldots dx_k \\
&\approx \sum_{i=1}^{N_P} w_k(i) g(x_k(i)). \quad (5)
\end{aligned}$$

Let us now assume that Θ assumes value in some set Ω. In the AMMPF, we use $|\Omega|$ particle filters (where $|\cdot|$ denotes the cardinality operator), with each particle filter approximating a conditional density $p(s_0, \ldots, s_k|\theta, Z^k)$ for some $\theta \in |\Omega|$. Hence, the corresponding set of particles is $\{u_k^\theta(i), w_k^\theta(i)\}_{i=1}^{N_P}$, where $u_k^\theta(i) = \big(s_0^\theta(i), \ldots, s_k^\theta(i)\big)$. The marginal probability $p(\theta|Z^k)$ is thereafter (approximately) calculated. Finally, the conditional expectation $E_{Z^k}[g(X_k)]$ is calculated using:

$$\begin{aligned}
&E_{Z^k}[g(X_k)] \\
&= \sum_{\theta \in \Omega} \int g(s_k, \theta) p(s_0, \ldots, s_k, \theta | Z^k) ds_0 \ldots ds_k \\
&= \sum_{\theta \in \Omega} p(\theta|Z^k) \int g(s_k, \theta) p(s_0, \ldots, s_k | \theta, Z^k) ds_0 \ldots ds_k \\
&\approx \sum_{\theta \in \Omega} p(\theta|Z^k) \sum_{i=1}^{N_P} w_k^\theta(i) g(s_k^\theta(i), \theta). \quad (6)
\end{aligned}$$

A. Derivation of the AMMPF

Let us assume, as usual, that the system formed by the sequence of states and observations, $\{(X_k, Z_k)\}$, corresponds to a partially observed Markov-1 process (i.e. $p(z_k|x_k, Z^{k-1}) = p(z_k|x_k)$ and $p(x_k|x_{k-1}, Z^{k-1}) = p(x_k|x_{k-1})$). In this case, it is also easy to see that $p(s_k|s_{k-1}, \theta, Z^{k-1}) = p(s_k|s_{k-1}, \theta)$, and the particle filter algorithm to approximate $p(s_0, \ldots, s_k|\theta, Z^k)$ can be derived straightforwardly. Note that the importance sampling function for the particle filters will have form $q(s_k|s_{k-1}, \theta, z^k)$.

In order to calculate the marginal probability $p(\theta|Z^k)$,

[1]The "forgetness condition" that we present here is merely intended to give intuition to the problem. Rigorously speaking, to avoid degeneracy, the system must be "mixing" in a certain sense; a condition to classify a system as such appears in [9], but as mentioned in the same work, this condition may be too restrictive.

observe that

$$
\begin{aligned}
&p(\theta|Z^k) \\
&= \int p(\theta, s_k|Z^k)ds_k \\
&= \frac{1}{p(z_k|Z^{k-1})} \int p(z_k|\theta, s_k)p(\theta, s_k|Z^{k-1})ds_k \\
&= \frac{p(\theta|Z^{k-1})}{p(z_k|Z^{k-1})} \int p(z_k|\theta, s_k)p(s_k|\theta, Z^{k-1})ds_k \\
&= \frac{p(\theta|Z^{k-1})}{p(z_k|Z^{k-1})} \int \dots \int p(z_k|\theta, s_k) \\
&\quad \times p(s_k|s_0, \dots, s_{k-1}, \theta, Z^{k-1}) \\
&\quad \times p(s_0, \dots, s_{k-1}|\theta, Z^{k-1})ds_0 \dots ds_k \\
&= \frac{p(\theta|Z^{k-1})}{p(z_k|Z^{k-1})} \int \dots \int \frac{p(z_k|\theta, s_k)p(s_k|s_{k-1}, \theta)}{q(s_k|s_{k-1}, \theta, z^k)} \\
&\quad \times p(s_0, \dots, s_{k-1}|\theta, Z^{k-1})q(s_k|s_{k-1}, \theta, z^k)ds_0 \dots ds_k.
\end{aligned} \tag{7}
$$

Observe now that $p(s_0, \dots, s_{k-1}|\theta, Z^{k-1})$ is represented by the set of particles $\{u^\theta_{k-1}(i), w^\theta_{k-1}(i)\}^{N_P}_{i=1}$, with $u^\theta_{k-1}(i) = \big(s^\theta_0(i), \dots, s^\theta_{k-1}(i)\big)$, and that

$$
s^\theta_k(i) \sim q(s_k|s^\theta_{k-1}(i), \theta, z^k), i = 1, \dots, N_P \tag{8}
$$

such that we can approximate (7) as

$$
\begin{aligned}
p(\theta|Z^k) &= \frac{p(\theta|Z^{k-1})}{p(z_k|Z^{k-1})} \sum_{i=1}^{N_P} w^\theta_{k-1}(i) \\
&\quad \times \frac{p(z_k|\theta, s^\theta_k(i))p(s^\theta_k(i)|s^\theta_{k-1}(i), \theta)}{q(s^\theta_k(i)|s^\theta_{k-1}(i), \theta, z^k)} \\
&= \frac{p(\theta|Z^{k-1})}{p(z_k|Z^{k-1})} \sum_{i=1}^{N_P} w^\theta_{k-1}(i)\overline{w}^\theta_k(i)
\end{aligned} \tag{9}
$$

where $\overline{w}^\theta_k(i)$ corresponds to the *unnormalized* weight attributed to particle i at time k. Since $p(z_k|Z^{k-1})$ does not depend on θ, it does not need to be explicitly calculated. It can be taken into account by normalizing the marginal probabilities accross Ω.

B. Algorithm

Initialization: For each $\theta \in \Omega$

1) Set $p(\theta|Z^0) \equiv p(\theta)$
2) For each particle $i = 1, \dots, N_P$
   a) Sample $s^\theta_0(i) \sim p(s_0|\theta)$
   b) Make $w^\theta_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$**:**

1) For each $\theta \in \Omega$
   a) For each particle $i = 1, \dots, N_P$
      i) Sample

$$
s^\theta_k(i) \sim q(s_k|s^\theta_{k-1}(i), \theta, z_k) \tag{10}
$$

      where $q(s_k|s_{k-1}, \theta, z^k)$ is an importance sampling function
      ii) Calculate the unnormalized weight according to

$$
\overline{w}^\theta_k(i) = \frac{p(z_k|s^\theta_k(i), \theta)p(s^\theta_k(i)|s^\theta_{k-1}(i), \theta)}{q(s^\theta_k(i)|s^\theta_{k-1}(i), \theta, z^k)} \tag{11}
$$

   b) Compute the unnormalized marginal probability of θ according to

$$
\overline{p}(\theta|Z^k) = p(\theta|Z^{k-1}) \sum_{i=1}^{N_P} w^\theta_{k-1}(i)\overline{w}^\theta_k(i) \tag{12}
$$

   c) Normalize the particle weights according to

$$
w^\theta_k(i) = \frac{\overline{w}^\theta_k(i)}{\sum_{j=1}^{N_P} \overline{w}^\theta_k(j)} \tag{13}
$$

2) Normalize the marginal parameter probabilities according to

$$
p(\theta|Z^k) = \frac{\overline{p}(\theta|Z^k)}{\sum_{\theta^* \in \Omega} \overline{p}(\theta^*|Z^k)} \tag{14}
$$

C. Rationale and computational aspects

The rationale for using the AMMPF approach against the self-resolving phenomenon is simple. We know that calculating $p(x_0, \dots, x_k|Z^k)$ using a particle filter leads to self-resolving, due to the presence of static components in X_k. We deal with this problem by "splitting" the posterior into the conditional density $p(s_0, \dots, s_k|\theta, Z^k)$ and the marginal probability $p(\theta|Z^k)$, and apply the particle approximation only to $p(s_k|\theta, Z^k)$.

The marginal $p(\theta|Z^k)$ is calculated "exactly", on basis, of course, of the other approximations, i.e. the particle approximation of $p(s_0, \dots, s_k|\theta, Z^k)$ and the possible discretization of Θ. The success of this approach, naturally, depends on the mixing properties of the Bayesian recursion to obtain $p(s_0, \dots, s_k|\theta, Z^k)$. If the particle approximation of this conditional density avoids self-resolving, i.e. satisfies the "forgetness condition" (3), the AMMPF should be an effective strategy.

Since the AMMPF consists of $|\Omega|$ particle filters running in parallel, its computational complexity is $O(|\Omega|N_P)$, assuming that the individual particle filters use the systematic resampling scheme (and hence have complexity $O(N_P)$), and considering the dimensions of Z_k and X_k as constants.

IV. The Rao-Blackwellized marginal particle filter

The Rao-Blackwellized marginal particle filter (RBMPF) is a variant of the particle filter algorithm, designed to counter the self-resolving phenomenon. It has previously been applied [8] to the parameter estimation problem for a certain class of structured models. The algorithm consists of a combination of two well-known Sequential Monte Carlo (SMC) techniques: the Rao-Blackwellized particle filter (RBPF) [10] and the marginal particle filter[2] (MPF) [11].

Let us consider again that the state X_k has the form $X_k = [S'_k, \Theta']'$, with Θ corresponding to the parameter vector, and that $\{(X_k, Z_k)\}$ is a partially observed Markov-1 process. In the RBMPF, the marginal density $p(s_k|Z^k)$ is approximated using the set of particles $\{s_k(i), w_k(i)\}^{N_P}_{i=1}$. The conditional

[2]To not be confused with the *marginalized* particle filter, which is another name for the Rao-Blackwellized particle filter. The marginal particle filter has this name because instead of considering the full trajectory density $p(x_0, \dots, x_k|Z^k)$, it considers only the marginal density $p(x_k|Z^k)$.

density $p(\theta|s_k(i), Z^k)$ is computed analytically, leading to the following approximation for $E_{Z^k}[g(X_k)]$:

$$\begin{aligned} E_{Z^k}[g(X_k)] &= \int\int g(s_k, \theta_k) p(\theta, s_k|Z^k) d\theta ds_k \\ &= \int\int g(s_k, \theta) p(\theta|s_k, Z^k) p(s_k|Z^k) d\theta ds_k \\ &\approx \sum_{i=1}^{N_P} w_k(i) \int g(s_k(i), \theta) p(\theta|s_k(i), Z^k) d\theta. \end{aligned} \tag{15}$$

Naturally, this approximation is only useful if the integral in (15) can be solved. This condition typically does not hold. In fact, it does not hold even for models where a RBPF (i.e. "non-marginal"') could be applied, i.e. models where the estimation of Θ is a linear-Gaussian problem conditioned on a trajectory $s_0, \ldots, s_k$. For this case, an extra approximation, proposed in [8], is required for the RBMPF.

We would like, however, to apply the RBMPF to general models containing both dynamic states and parameters, without requiring them to have a partial linear-Gaussian structure. One situation where a closed form expression for (15) actually exists is when all parameters are discrete; if that is not case, we may perform discretization of the parameter space, in the same way that we have proposed in Section III for the AMMPF. In this case, we may replace the integral in (15) by a sum, and as we are going to show, it is always possible to approximately compute the conditional probability mass $p(\theta|s_k(i), Z^k)$. With Θ assuming values in Ω, the expectation (15) is calculated using

$$E_{Z^k}[g(X_k)] \approx \sum_{i=1}^{N_P} w_k(i) \sum_{\theta\in\Omega} g(s_k(i), \theta) p(\theta|s_k(i), Z^k) \tag{16}$$

and the set of particles produced by the algorithm at each time step k consists of

$$\left\{ s_k(i), w_k(i), \left\{ p(\theta|s_k(i), Z^k) \right\}_{\theta\in\Omega} \right\}_{i=1}^{N_P}. \tag{17}$$

We will show now how to iteratively obtain (17).

A. Derivation of the MPF for $p(s_k|Z^k)$

Since the set of particle states and weights $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$ has to represent $p(s_k|Z^k)$ instead of $p(s_0, \ldots, s_k|Z^k)$, we need to use a marginal particle filter. We cannot, however, use the MPF exactly as described by Klaas, Freitas and Doucet [11], because the sequence $\{(S_k, Z_k)\}$ is not necessarily partially observed Markov-1 (although $\{(X_k, Z_k)\}$ is assumed to be). Let us then derive a version of the MPF adapted to our case. Observe first that

$$\begin{aligned} p(s_k|Z^k) &= \sum_{\theta\in\Omega} p(s_k, \theta|Z^k) \\ &= \sum_{\theta\in\Omega} \frac{p(z_k|s_k, \theta) p(s_k, \theta|Z^{k-1})}{p(z_k|Z^{k-1})} \end{aligned} \tag{18}$$

where we can see that

$$\begin{aligned} & p(s_k, \theta|Z^{k-1}) \\ &= \sum_{\theta^*\in\Omega} \int p(s_k, \theta|s_{k-1}, \theta^*) p(s_{k-1}, \theta^*|Z^{k-1}) ds_{k-1} \\ &= \int p(s_k|s_{k-1}, \theta) p(s_{k-1}, \theta|Z^{k-1}) ds_{k-1} \end{aligned} \tag{19}$$

and therefore

$$\begin{aligned} & p(s_k|Z^k) \\ &= \sum_{\theta\in\Omega} \frac{p(z_k|s_k, \theta)}{p(z_k|Z^{k-1})} \int p(s_k|s_{k-1}, \theta) p(s_{k-1}, \theta|Z^{k-1}) ds_{k-1} \\ &= \frac{\sum_{\theta\in\Omega} \int p(z_k|s_k, \theta) p(s_k|s_{k-1}, \theta) p(s_{k-1}, \theta|Z^{k-1}) ds_{k-1}}{p(z_k|Z^{k-1})} \\ &= \frac{E_{Z^{k-1}}\left[p(z_k|s_k, \Theta) p(s_k|X_{k-1}) \right]}{p(z_k|Z^{k-1})}. \end{aligned} \tag{20}$$

Observe now that a conditional expectation of the form $E_{Z^k}[g(S_k)]$ is given by

$$\begin{aligned} & E_{Z^k}[g(S_k)] \\ &= \int g(s_k) p(s_k|Z^k) ds_k \\ &= \int g(s_k) \frac{E_{Z^{k-1}}\left[p(z_k|s_k, \Theta) p(s_k|X_{k-1}) \right]}{p(z_k|Z^{k-1})} ds_k \\ &= \int g(s_k) \frac{E_{Z^{k-1}}\left[p(z_k|s_k, \Theta) p(s_k|X_{k-1}) \right] q(s_k|z_k)}{p(z_k|Z^{k-1}) q(s_k|z_k)} ds_k \end{aligned} \tag{21}$$

where $q(s_k|z_k)$ is an appropriate importance sampling function. If we generate N_P independent, identically distributed particle states $s_k(i)$ by sampling from $q(s_k|z_k)$, then using the law of large numbers, we may approximate $E_{Z^k}[g(S_k)]$ using

$$E_{Z^k}[g(S_k)] \approx \sum_{i=1}^{N_P} g(s_k(i)) w_k(i) \tag{22}$$

where the particle weights are given by

$$w_k(i) = \frac{E_{Z^{k-1}}\left[p(z_k|s_k(i), \Theta) p(s_k(i)|X_{k-1}) \right]}{N_P p(z_k|Z^{k-1}) q(s_k(i)|z_k)}. \tag{23}$$

To calculate (23), we can approximate the numerator of (23) using (16), i.e. we use the set of particles (17) produced at the previous iteration $k-1$:

$$\begin{aligned} & E_{Z^{k-1}}\left[p(z_k|s_k(i), \Theta) p(s_k(i)|X_{k-1}) \right] \\ &\approx \sum_{j=1}^{N_P} w_{k-1}(j) \sum_{\theta\in\Omega} p(z_k|s_k(i), \theta) p(s_k(i)|s_{k-1}(j), \theta) \\ &\quad \times p(\theta|s_{k-1}(j), Z^{k-1}) \\ &= \sum_{\theta\in\Omega} p(z_k|s_k(i), \theta) \sum_{j=1}^{N_P} w_{k-1}(j) p(s_k(i)|s_{k-1}(j), \theta) \\ &\quad \times p(\theta|s_{k-1}(j), Z^{k-1}). \end{aligned} \tag{24}$$

Unlike the standard particle filter, the MPF does not contain a resampling step. There are also two special cases of interest:

1) Special case: if $p(z_k|s_k,\theta) = p(z_k|s_k)$: In this case, the following relationship (for the particle weights) holds:

$$w_k(i) \propto \frac{p(z_k|s_k(i))E_{Z^{k-1}}\left[p(s_k(i)|X_{k-1})\right]}{q(s_k(i)|z_k)} \tag{25}$$

where $E_{Z^{k-1}}\left[p(s_k(i)|X_{k-1})\right]$ can be approximated using the set of particles (17) produced at the previous iteration $k-1$. Note that if we choose $E_{Z^{k-1}}\left[p(s_k|X_{k-1})\right]$ as the importance sampling function $q(s_k|z_k)$, we will have

$$w_k(i) \propto p(z_k|s_k(i)) \tag{26}$$

and hence, $E_{Z^{k-1}}\left[p(s_k|X_{k-1})\right]$ corresponds to the "standard" importance sampling function of the MPF (i.e. analogous to the importance sampling function $p(x_k|x_{k-1})$ for the standard PF).

2) Special case: if $p(z_k|s_k,\theta) = p(z_k|s_k)$ and $p(s_k|x_{k-1}) = p(s_k|s_{k-1})$: This case corresponds to the original MPF as presented in [11]. One practical problem with this characteristic is the problem of track labelling for a multi-target scenario with no target births or deaths (see [12]). In this case, we have

$$w_k(i) \propto \frac{p(z_k|s_k(i))E_{Z^{k-1}}\left[p(s_k(i)|S_{k-1})\right]}{q(s_k(i)|z_k)} \tag{27}$$

where

$$E_{Z^{k-1}}\left[p(s_k(i)|S_{k-1})\right] \approx \sum_{j=1}^{N_P} w_{k-1}(j)p(s_k(i)|s_{k-1}(j))). \tag{28}$$

which does not require a summation over all $\theta \in \Omega$, and thus has lower computational cost. Note that (28) also corresponds to the "standard" importance sampling function for this special case.

B. Calculation of conditional probabilities $p(\theta|s_k, Z^k)$

Observe that the conditional probability $p(\theta|s_k(i), Z^k)$ is given by

$$p(\theta|s_k(i), Z^k) = \frac{p(z_k|\theta, s_k(i))p(\theta, s_k(i)|Z^{k-1})}{p(z_k|s_k(i), Z^{k-1})p(s_k(i)|Z^{k-1})} \tag{29}$$

and from (19), we have

$$p(\theta|s_k(i), Z^k) = \frac{p(z_k|\theta, s_k(i))\int p(s_k(i)|s_{k-1},\theta)p(s_{k-1},\theta|Z^{k-1})ds_{k-1}}{p(z_k|s_k(i), Z^{k-1})p(s_k(i)|Z^{k-1})}. \tag{30}$$

By using the set of particles (17) obtained at the previous iteration $k-1$, we can approximate (30) by

$$p(\theta|s_k(i), Z^k) \approx \frac{p(z_k|\theta, s_k(i))}{p(z_k|s_k(i), Z^{k-1})p(s_k(i)|Z^{k-1})} \times \sum_{j=1}^{N_P} w_{k-1}(j)p(s_k(i)|s_{k-1}(j),\theta)p(\theta|s_{k-1}(j), Z^{k-1}). \tag{31}$$

The denominator of (31) does not depend on θ, and hence, can be taken in account by normalization across all $\theta \in \Omega$, for each particle state $s_k(i)$.

C. Algorithm

Initialization: For each particle $i = 1, \ldots, N_P$

1) Sample $s_0(i) \sim p(s_0)$
2) Make $w_0(i) = \frac{1}{N_P}$
3) For each $\theta \in \Omega$, set $p(\theta|s_0(i))$.

At every time step $k = 1, 2, \ldots$:

1) For each particle $i = 1, \ldots, N_P$
   a) Sample
   $$s_k(i) \sim q(s_k|z_k) \tag{32}$$
   where $q(s_k|z_k)$ is the MPF importance sampling function
   b) Calculate the unnormalized weight according to
   $$\overline{w}_k(i) = q(s_k(i)|z_k)^{-1} \sum_{\theta\in\Omega} p(z_k|s_k(i),\theta) \sum_{j=1}^{N_P} w_{k-1}(j) \times p(s_k(i)|s_{k-1}(j),\theta)p(\theta|s_{k-1}(j), Z^{k-1}) \tag{33}$$
   c) For each $\theta \in \Omega$, compute the unnormalized conditional parameter probability according to
   $$\overline{p}(\theta|s_k(i), Z^k) = p(z_k|\theta, s_k(i)) \sum_{j=1}^{N_P} w_{k-1}(j) \times p(s_k(i)|s_{k-1}(j),\theta)p(\theta|s_{k-1}(j), Z^{k-1}) \tag{34}$$
   d) Normalize the conditional parameter probabilities according to
   $$p(\theta|s_k(i), Z^k) = \frac{\overline{p}(\theta|s_k(i), Z^k)}{\sum_{\theta^*\in\Omega}\overline{p}(\theta^*|s_k(i), Z^k)} \tag{35}$$
2) Normalize the particle weights according to
   $$w_k(i) = \frac{\overline{w}_k(i)}{\sum_{j=1}^{N_P}\overline{w}_k(j)} \tag{36}$$

D. Rationale and computational aspects

As the reader may have noted, the RBMPF is similar to a "switched" AMMPF:

- In the AMMPF, we use the particle approximation to obtain $p(s_0, \ldots, s_k|\theta, Z^k)$, and calculate $p(\theta|Z^k)$ analytically;
- In the RBMPF, we use the particle approximation to obtain $p(s_k|Z^k)$, and calculate $p(\theta|s_k, Z^k)$ analytically.

Therefore, the efficacy of the RBMPF approach depends on the mixing properties of $p(s_k|Z^k)$, instead of $p(s_k|\theta, Z^k)$. Note that using a RBPF instead of a RBMPF (i.e. using the particle approximation to obtain $p(s_0, \ldots, s_k|Z^k)$ and calculating $p(\theta|s_0, \ldots, s_k, Z^k)$ analytically) leads to degeneracy; see [8].

From a computational point of view, the RBMPF is more costly than the AMMPF (see Table I for a summary on the algorithms' complexities). However, the RBMPF has less memory requirements than the AMMPF, as it needs to store $|\Omega|N_P$ conditional probabilities, instead of $|\Omega|N_P$ particles.

TABLE I
COMPUTATIONAL COMPLEXITY OF SMC PARAMETER ESTIMATION METHODS (USING SYSTEMATIC RESAMPLING WHEN APPLICABLE)

SIR PF	$O(N_P)$
AMMPF	$O(\|\Omega\|N_P)$
RBMPF	$O(\|\Omega\|N_P^2)$

V. SIMULATION

A. Simulation description

For the simulation scenarios, we will consider the problem of estimating the turn rate of a target performing a constant turn. In this problem, we consider a moving target, that moves according to the constant turn model described in [13]. The state of the target is given by $X_k = \left[P_k^{\mathrm{x}}, P_k^{\mathrm{y}}, V_k^{\mathrm{x}}, V_k^{\mathrm{y}}, \Theta\right]'$, where x and y denote the Cartesian coordinates, $\left(P_k^{\mathrm{x}}, P_k^{\mathrm{y}}\right)$ and $\left(V_k^{\mathrm{x}}, V_k^{\mathrm{y}}\right)$ correspond respectively to the position and velocity components, and Θ is the turn rate, that we treat as an unknown parameter (Θ will always be expressed in °/second). Hence, the dynamic part of the state corresponds to $S_k = \left[P_k^{\mathrm{x}}, P_k^{\mathrm{y}}, V_k^{\mathrm{x}}, V_k^{\mathrm{y}}\right]'$.

We will evaluate two scenarios; one with the power spectral density of the process noise (see [13]) given by $\sigma_w^2 = 36$, and another with $\sigma_w^2 = 144$. In both scenarios, the true value of Θ is $\theta = 5$. The target trajectories are shown in Figs. 1(a) and 2(a). Observations will be based on the following observation model:

$$p(z_k|x_k) = \mathcal{N}\left(z_k; \begin{bmatrix} p^{\mathrm{x}} \\ p^{\mathrm{y}} \end{bmatrix}, \begin{bmatrix} 2.25 & 0 \\ 0 & 2.25 \end{bmatrix}\right). \tag{37}$$

Note that these scenarios correspond to the special case described in Section IV-A1.

We are going to compare the performance of four SMC algorithms: the standard SIR particle filter, the SIR particle filter with artificial dynamics (that we refer as SIR-AD PF), the AMMPF and the RBMPF. The SIR-AD PF attributes an artificial dynamic model to Θ, corresponding to

$$p(\Theta_{k+1}|\Theta_k) = \mathcal{N}(\Theta_{k+1}; \Theta_k; 0.01) \tag{38}$$

and for the AMMPF and the RBMPF, we consider Θ as a discrete parameter, assuming values in $\Omega = \{-9, -7, \ldots, 7, 9\}$. We assume the following prior densities for S_k and Θ:

$$p(s_0) = p(s_0|\theta) = \mathcal{N}\left(s_0; \begin{bmatrix} p_0^{\mathrm{x}} \\ p_0^{\mathrm{y}} \\ v_0^{\mathrm{x}} \\ v_0^{\mathrm{y}} \end{bmatrix}, \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}\right),$$
$$p(\theta) = p(\theta|s_0) = \mathcal{N}(\theta; 0.01, 36) \tag{39}$$

where for the AMMPF and the RBMPF, we consider discretized versions of $p(\theta)$ and $p(\theta|s_0)$ respectively, obtained by taking the true values of these densities for $\theta \in \Omega$ and thereafter doing a normalization. We use 1,000 particles for RBMPF and for each particle filter of the AMMPF, and 10,000 particles for the SIR and SIR-AD particle filters. This means that we use $|\Omega|$ times less particles for the AMMPF and the RBMPF, to compensate the fact that they have higher computational complexity. We make that choice to get a better insight over the results; it is not strictly fair since the RBMPF has higher complexity than the AMMPF.

All particles filters use the "standard" importance sampling function, i.e. $p(x_k|x_{k-1})$ for the SIR PF and the SIR-AD PF, $p(s_k|s_{k-1})$ for the AMMPF, and $E_{Z^{k-1}}\left[p(s_k|X_{k-1})\right]$ for the RBMPF. We perform a Monte Carlo simulation with 50 runs for both scenarios.

Note that the turn rate estimation problem has partially linear-Gaussian structure, such that a multiple model Kalman Filter with appropriate parameter discretization, or perhaps even an Extended Kalman Filter, would probably yield satisfactory results. The choice of problem was arbitrarily made: it should be clear to the reader that the algorithms that we are analyzing do not require any linear-Gaussian assumptions on the dynamic and observations models, neither rely on any linear-Gaussian approximations.

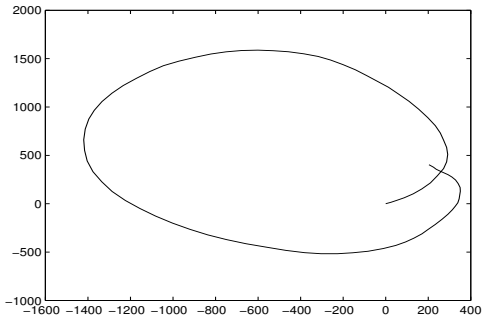
B. Simulation results

The results for the two scenarios are shown in Figs. 1 and 2. For each run, we compute the effective number of particles (NEff), given by $\left(\sum_{i=1}^{N_P} w_k(i)\right)^{-2}$ (for the AMMPF, we take the simple average of the NEff over the multiple particle filters).

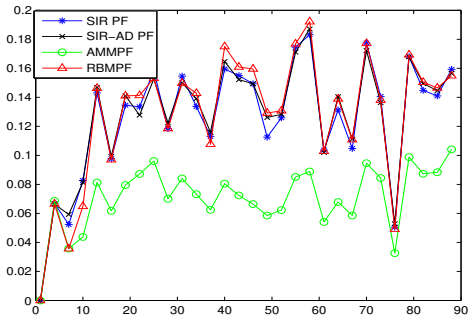
Figs. 1(b) and 2(b) shows the NEff averaged over the Monte Carlo runs and divided by the number of particles, for the four algorithms. The second scenario (with higher process noise) results in overall lower NEff, as expected from the normal behavior of SMC methods. We observe also that the AMMPF results in considerably lower NEff than the other algorithms. That also makes sense, since the AMMPF algorithm devotes a large number of particles to explore unlikely values of the parameter Θ.

Figs. 1(c) and 2(c) show the Root Mean Square (RMS) errors associated with the Minimum Mean Square Error (MMSE) parameter estimate $\hat{\theta}_k$ calculated at each time step. The algorithms with best estimation accuracy (in RMS sense) were the RBMPF and the AMMPF respectively for the first and second scenarios. For the first scenario, the RBMPF and the AMMPF had significantly better performance than the SIR and SIR-AD PFs. In the second scenario the difference is not much expressive, possibly because the parameter has much lower observability and the lower NEff impairs the RBMPF and AMMPF, since they are using smaller numbers of particles.

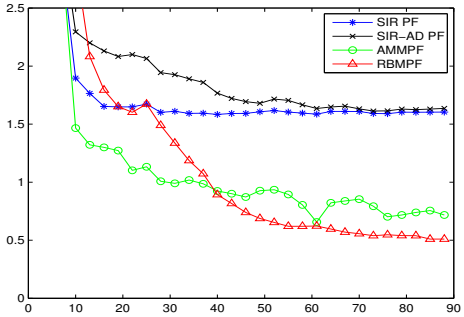
Besides the point estimate $\hat{\theta}_k$, we also calculate, for each run and time step, the standard deviation $\sigma_k = \sqrt{E_{Z^k}\left[\left(\hat{\theta}_k - \Theta\right)^2\right]}$, obtained from the output of the algorithms. Thereafter we find, for each sequence of Monte Carlo runs, the number of "outliers"'. We classify an estimate $\hat{\theta}_k$ as an "outlier" if its error w.r.t. the true value of θ is more than $3\sigma_k$. The rate of outliers is thus given by the number of outliers divided by the number of Monte Carlo runs, and shown in Figs. 1(d) and 2(d). Note that this definition of



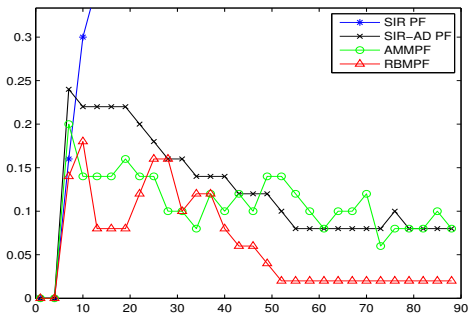
(a) Target trajectory



(b) Average NEff/N_P

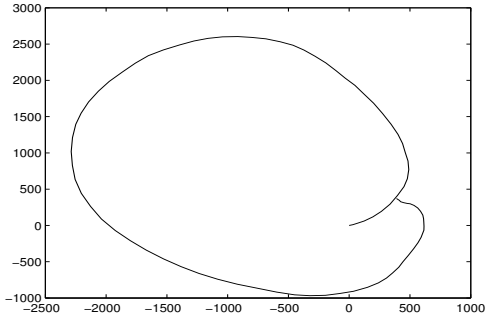


(c) RMS errors

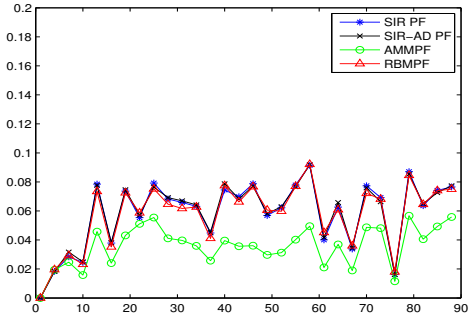


(d) Ratio of outliers

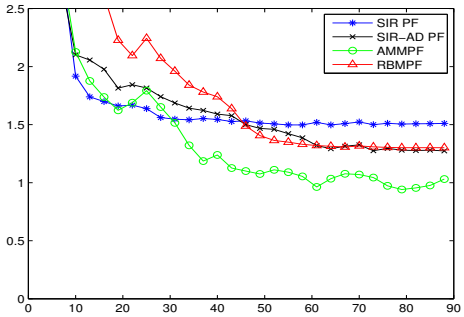
Fig. 1. Monte Carlo simulation results for $\sigma_w^2 = 36$



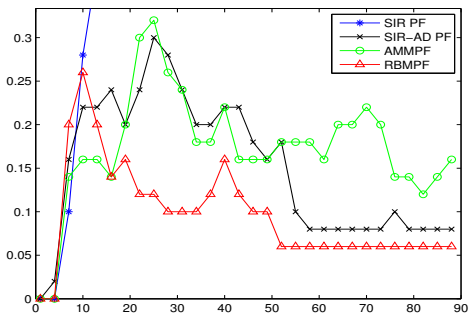
(a) Target trajectory



(b) Average NEff/N_P



(c) RMS errors



(d) Ratio of outliers

Fig. 2. Monte Carlo simulation results for $\sigma_w^2 = 144$

outliers is arbitrary, since the turn rate estimation problem is not linear-Gaussian and hence we do not know how "heavy-tailed" $p(\theta|Z^k)$ is. However, for practical purposes, the rate of outliers provides some intuition on how much the calculated standard deviation is consistent with actual estimation errors.

The results show that the SIR PF has unacceptable performance in that sense, which is obviously expected since we know that the algorithm leads to self-resolving, and thus zero standard deviation. The three other algorithms effectively avoid this situation of total degeneracy. For both scenarios, the RBMPF results in the lowest number of outliers.

The results of the AMMPF are more difficult to interpret. The algorithm seems to result in relatively higher number of outliers, even higher than the SIR-AD PF for the scenario with higher σ_w^2. That is somewhat counterintuitive, since unlike the SIR-AD PF, the AMMPF does not rely on biasing the system model and thus should lead to a better approximation of $p(\theta|Z^k)$.

A possible explanation is the lower NEff of the AMMPF, which perhaps impairs its ability of obtaining a good particle approximation, even more for the second scenario. If that is the case, a possible solution would be to use a larger number of particles or a better importance sampling function. We should also note that the apparent better σ_k-to-error consistency of the SIR-AD PF for the second scenario may be due to the inflation of σ_k (caused by the artificial process noise), rather than due to a better approximation of $p(\theta|Z^k)$.

VI. Conclusions

In this work, we presented two SMC algorithms, basically modified versions of existing methods, for online Bayesian parameter estimation: the AMMPF and the RBMPF. These methods avoid the self-resolving problem and do not rely on biasing the system model, being, in theory, suitable for obtaining both point estimates of the parameters and other statistical descriptions, such as the standard deviation. However, both methods rely on discretization of the parameter space, which makes them unsuitable for problems with high dimensional parameter vectors.

In the simulations, the RBMPF showed to have performance as good or better than a particle filter with artificial dynamics in terms of accuracy of point estimation, and better performance in terms of consistency between standard deviation and actual errors. On the other hand, the AMMPF was better than the artificial dynamics approach in terms of accuracy of point estimation, but as bad or worse in terms of standard deviation consistency. It is not clear of whether this is due to the use of an insufficient number of particles, or perhaps a theoretical deficiency of the algorithm.

To obtain more conclusive results, we consider, as future works, to test these algorithms with different estimation problems and with more variation on the number of particles. Naturally, proposing alternatives to these algorithms with reduced computational complexity, especially for the more intensive RBMPF, is also an interesting subject for future research.

Acknowledgments

The research leading to these results has received funding from the EU's Seventh Framework Programme under grant agreement n° 238710. The research has been carried out in the MC IMPULSE project: https://mcimpulse.isy.liu.se.

This research has been also supported by the Netherlands Organisation for Scientific Research (NWO) under the Casimir program, contract 018.003.004. Under this grant Yvo Boers holds a part-time position at the Department of Applied Mathematics at the University of Twente.

References

[1] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multi-modality through mixture tracking," in *Proc. 9th IEEE International Conference on Computer Vision*, vol. 2, Nice, France, 2003, pp. 1110–1116.

[2] H. A. P. Blom and E. A. Bloem, "Particle filtering for stochastic hybrid systems," in *Proc. 43rd IEEE Conf. Decision and Control*, Atlantis, Bahamas, Dec. 14–17, 2004.

[3] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space model," *Annals Institute Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010.

[4] Y. Boers, E. Sviestins, and J. N. Driessen, "Mixed labelling in multitarget particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 2, pp. 792–802, 2010.

[5] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski, "An overview of Sequential Monte Carlo methods for parameter estimation on general state space models," in *Proc. 15th IFAC Symp. System Identification (SYSID)*, Saint-Malo, France, 2009.

[6] T. Higuchi, "Self-organizing time series model," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. De Freitas, and N. J. Gordon, Eds. Springer, 2001.

[7] N. G. Polson, J. R. Stroud, and P. Müller, "Practical filtering with sequential parameter learning," *Journal Of The Royal Statistical Society Series B*, vol. 70, no. 2, pp. 413–428, 2008.

[8] F. Lindsten, T. B. Schön, and L. Svensson, "A non-degenerate Rao-Blackwellised particle filter for estimating static parameters in dynamical models," in *Proc. 16th IFAC Symposium on System Identification (SYSID)*, 2012.

[9] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 736–746, 2002.

[10] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *J. Royal Stat. Soc. B*, vol. 64, pp. 827–836, 2002.

[11] M. Klaas, N. de Freitas, and A. Doucet, "Toward practical N^2 Monte Carlo: the marginal particle filter," in *Proc. 21th Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. Arlington, Virginia: AUAI Press, 2005, pp. 308–315.

[12] E. H. Aoki, Y. Boers, L. Svensson, P. Mandal, and A. Bagchi, "The Rao-Blackwellized marginal M-SMC filter for Bayesian multi-target tracking and labelling," in *Proc. 15th International Conference of Information Fusion*, Singapore, Jul. 9–12, 2012.

[13] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.