Acceptance probability of IP-MCMC-PF: revisited

Fernando J. Iglesias García* †, Mélanie Bocquel*, Pranab K. Mandal†, and Hans Driessen*.
*Sensors Development System Engineering, Thales Nederland B.V. Hengelo, Netherlands.
†Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente. Enschede, Netherlands.
f.j.iglesiasgarcia@utwente.nl, melanie.bocquel@nl.thalesgroup.com, p.k.mandal@utwente.nl., hans.driessen@nl.thalesgroup.com

***Abstract*—Tracking of multiple objects via particle filtering faces the difficulty of dealing effectively with high dimensional state spaces. One efficient solution consists of integrating Markov chain Monte Carlo (MCMC) sampling at the core of the particle filter. To accomplish such integration, a few different approaches have been proposed in the literature during the last decade. In this paper, we introduce the derivation of the acceptance probability of the interacting population MCMC particle filter (IP-MCMC-PF), one of the most recent approaches to MCMC-based particle filtering. Additionally, we show that the previous expression known in the literature was incomplete.**

I. Introduction

Inference in hidden Markov models described by nonlinear transition kernels and/or nonlinear observation functions is a hard problem. In lieu of analytical and efficient solutions, a common practice is to employ approximate techniques such as sequential Monte Carlo methods or particle filters [1], [2]. Their attractiveness, reason why their use is widespread, is twofold: first, they do not impose restrictions on the mathematical models; and secondly, they guarantee convergence provided enough computational resources.

The pioneering particle filter [3], [4] based on sequential importance sampling and resampling becomes inefficient as the dimension of the state space increases [5]. Increasing the dimension of the state space is crucial when aiming at tracking multiple objects *jointly*. To circumvent the "curse of dimensionality", the integration of Markov chain Monte Carlo (MCMC) methods[a] with particle filters has become popular during the most recent years [7]–[12].

The interacting population MCMC particle filter (IP-MCMC-PF) [9] is one of the MCMC-based particle filters[b] developed recently. The IP-MCMC-PF is a fixed cardinality filter and it is used as a building block in a multiple cardinality algorithm which tackles the more general, unknown cardinality multiple object tracking problem [13].

Although the results of the IP-MCMC-PF seem promising, we carefully derive in Section III the expression of its acceptance probability and show that the expression known so far is not complete. We also demonstrate through a counterexample in Section IV-A that the previous sampling scheme does not necessarily converge to the correct posterior distribution. Furthermore, Section IV-B analyses the behaviours of the new expression, the previous one, and a different MCMC-based particle filter [8] in a multi-object tracking scenario.

The filter in [8] and the IP-MCMC-PF have both similarities and differences. On the one hand, they are similar because they rely neither on importance sampling nor resampling, but only on MCMC sampling. On the other hand, whereas the proposed samples in the IP-MCMC-PF are drawn from a subset of the variables in the complete space, the filter in [8] combines two proposals: one that draws from the complete state space, and a second one similar to the IP-MCMC-PF. Therefore we shall refer to the filer in [8] as the two-step MCMC filter. In Section IV these two filters will be compared further.

II. Bayesian filtering

Before presenting the derivation of the acceptance probability, let us briefly recall the fundamentals of recursive state estimation. Consider a generic nonlinear and non-Gaussian dynamical system described by the time evolution and observation equations:

$$\mathbf{s}_\mathrm{k} = f\left(\mathbf{s}_{\mathrm{k}-1}, \mathbf{v}_\mathrm{k}\right), \tag{1}$$

$$\mathbf{z}_\mathrm{k} = g\left(\mathbf{s}_\mathrm{k}, \mathbf{w}_\mathrm{k}\right). \tag{2}$$

Here $\mathbf{s}_\mathrm{k}$ and $\mathbf{z}_\mathrm{k}$ denote respectively the state of the system and the observations, while $\mathbf{v}_\mathrm{k}$ and $\mathbf{w}_\mathrm{k}$ are respectively the process and measurement noise. The subindex k is used to denote the time step.

The process noise $\mathbf{v}$ together with the transition function f specify the transition distribution $\mathrm{p}\left(\mathbf{s}_\mathrm{k}|\mathbf{s}_{\mathrm{k}-1}\right)$. Likewise, the measurement noise $\mathbf{w}$ and the function g specify the likelihood function $\mathrm{p}\left(\mathbf{z}_\mathrm{k}|\mathbf{s}_\mathrm{k}\right)$. Following a Bayesian approach, the filtering problem consists of finding $\mathrm{p}\left(\mathbf{s}_\mathrm{k}|\mathbf{z}_{1:\mathrm{k}}\right)$, which encompasses the time prediction (3) and measurement update (4) steps as below:

$$\mathrm{p}\left(\mathbf{s}_\mathrm{k}|\mathbf{z}_{1:\mathrm{k}-1}\right) = \int \mathrm{p}\left(\mathbf{s}_\mathrm{k}|\mathbf{s}_{\mathrm{k}-1}\right)\ \mathrm{p}\left(\mathbf{s}_{\mathrm{k}-1}|\mathbf{z}_{1:\mathrm{k}-1}\right)\ \mathrm{d}\mathbf{s}_{\mathrm{k}-1} \tag{3}$$

$$\mathrm{p}\left(\mathbf{s}_\mathrm{k}|\mathbf{z}_{1:\mathrm{k}}\right) \propto \mathrm{p}\left(z_\mathrm{k}|\mathbf{s}_\mathrm{k}\right)\ \mathrm{p}\left(\mathbf{s}_\mathrm{k}|\mathbf{z}_{1:\mathrm{k}-1}\right) \tag{4}$$

III. Derivation of the acceptance probability

The goal of this section is twofold. First, to recall the proposal mechanism of the MCMC-PF introduced by [9]. Second, to introduce the derivation of its corresponding acceptance ratio.

The proposal mechanism in the IP-MCMC-PF is in Algorithm 1. In the algorithm, the particle approximation of the

[a]Traditionally, MCMC has been employed in batch data processing [6]. We wish to remark that in this paper we focus on a different use of MCMC in which it substitutes the importance sampling and resampling stages found in traditional particle filters.

[b]In this paper, we use the terminology *MCMC-based particle filter* introduced in [7] to refer to the family of particle filters leveraging MCMC in substitution of importance sampling and resampling. Other terminology, namely *sequential MCMC*, is also common.

posterior at the previous time step is written as

$$\widehat{\mathrm{p}}\left(\mathbf{s}_{\mathrm{k-1}}|\mathbf{z}_{1:\mathrm{k-1}}\right) = \sum_{i=1}^{\mathrm{N_p}} \frac{1}{\mathrm{N_p}}\, \delta\left(\mathbf{s}_{\mathrm{k-1}} - \mathbf{s}_{\mathrm{k-1}}^{i}\right). \tag{5}$$

Suppose $\mathbf{s}_{\mathrm{k}}^{m}$ denotes the current state of the Markov chain, $\mathrm{N_b}$ the number of blocks[c] or partitions which constitute the complete state vector, and δ the Dirac delta function. $\mathrm{Unif}\{a,\ldots,b\}$ represents the discrete uniform distribution between a and b. Furthermore, $\mathbf{s}_{\mathrm{k}}^{*}$ is the state where the chain is proposed to move and $\mathbf{s}_{\mathrm{k}}^{m+1}$ is the new sample of the posterior $\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k}}\right)$. Note that $\mathbf{s}_{\mathrm{k}}$ represents the complete state vector, containing all blocks together. The samples drawn via MCMC constitute an empirical distribution as in Equation (5) that approximates $\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k}}\right)$.

Input : $\widehat{\mathrm{p}}\left(\mathbf{s}_{\mathrm{k-1}}|\mathbf{z}_{1:\mathrm{k-1}}\right)$, $\mathbf{z}_{\mathrm{k}}$, $\mathbf{s}_{\mathrm{k}}^{m}$
Output: $\mathbf{s}_{\mathrm{k}}^{m+1}$

1 $j \sim \mathrm{Unif}\{1,\ldots,\mathrm{N_b}\}$
2 $\mathbf{s}_{\mathrm{k|k-1}}^{*} \sim \widehat{\mathrm{p}}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k-1}}\right) = \sum_{i=1}^{\mathrm{N_p}} \frac{1}{\mathrm{N_p}}\, \mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k-1}}^{i}\right)$
3 $\mathbf{s}_{\mathrm{k}}^{*} = \left[\mathbf{s}_{\mathrm{k},1}^{m\,\mathrm{T}} \;\ldots\; \mathbf{s}_{\mathrm{k},j-1}^{m\,\mathrm{T}}\; \mathbf{s}_{\mathrm{k|k-1},j}^{*}\; \mathbf{s}_{\mathrm{k},j+1}^{m\,\mathrm{T}} \;\ldots\; \mathbf{s}_{\mathrm{k},\mathrm{N_b}}^{m\,\mathrm{T}}\right]^{\mathrm{T}}$
4 $\mathcal{A}\left(\mathbf{s}_{\mathrm{k}}^{m}, \mathbf{s}_{\mathrm{k}}^{*}\right) = \min\left\{1, \frac{\mathrm{p}(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{z}_{1:\mathrm{k}})}{\mathrm{p}(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{z}_{1:\mathrm{k}})}\, \frac{\mathrm{q}(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{s}_{\mathrm{k}}^{*})}{\mathrm{q}(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{s}_{\mathrm{k}}^{m})}\right\}$
5 $u \sim \mathrm{Unif}\,(0,1)$
6 **if** $u < \mathcal{A}\left(\mathbf{s}_{\mathrm{k}}^{m}, \mathbf{s}_{\mathrm{k}}^{*}\right)$ **then**
7 $\quad \mathbf{s}_{\mathrm{k}}^{m+1} = \mathbf{s}_{\mathrm{k}}^{*}$
8 **else**
9 $\quad \mathbf{s}_{\mathrm{k}}^{m+1} = \mathbf{s}_{\mathrm{k}}^{m}$
10 **end**

Algorithm 1: Proposal mechanism and Metropolis-Hastings sampling in the IP-MCMC-PF [9].

In order to calculate the acceptance ratio $\mathcal{A}\left(\mathbf{s}_{\mathrm{k}}^{m}, \mathbf{s}_{\mathrm{k}}^{*}\right)$ in Algorithm 1, an expression for the proposal distribution q must be specified. It follows from steps 1 to 3 that

$$\mathrm{q}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{s}_{\mathrm{k}}^{m}\right) = \sum_{\mathrm{b}=1}^{\mathrm{N_b}} \frac{1}{\mathrm{N_b}}\, \mathbb{1}\left(\mathbf{s}_{\mathrm{k},\backslash\mathrm{b}}^{*} - \mathbf{s}_{\mathrm{k},\backslash\mathrm{b}}^{m}\right) \sum_{i=1}^{\mathrm{N_p}} \frac{1}{\mathrm{N_p}}\, \mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}^{*}|\mathbf{s}_{\mathrm{k-1,b}}^{i}\right). \tag{6}$$

In Equation (6), $\mathbb{1}$ denotes the indicator function and $\backslash\mathrm{b}$ denotes all the blocks except for block b. That is,

$$\mathbf{s}_{\mathrm{k},\backslash\mathrm{b}} = \left[\mathbf{s}_{\mathrm{k},1}^{\mathrm{T}} \;\ldots\; \mathbf{s}_{\mathrm{k,b-1}}^{\mathrm{T}}\; \mathbf{s}_{\mathrm{k,b+1}}^{\mathrm{T}} \;\ldots\; \mathbf{s}_{\mathrm{k},\mathrm{N_b}}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{7}$$

Equation (6) assumes that the time evolution of the b^{th} block is self-contained. In other words, that the prediction of the variables within block b is independent from the variables in all other blocks $\backslash$b. Formally, block-independent time evolution means that Equation (1) is equivalent to

$$\mathbf{s}_{\mathrm{k,b}} = \widetilde{f}\left(\mathbf{s}_{\mathrm{k-1,b}}, \mathbf{v}_{\mathrm{k,b}}\right), \forall\ \mathrm{b} = 1,\ldots,\mathrm{N_b}. \tag{8}$$

[c] In the context of multiple object tracking, a block may be associated with one of the objects under track. In such a case, the vector $\mathbf{s}_{\mathrm{k,b}}$ would contain, for instance, the variables corresponding to the dynamics (position and velocity) of a single object.

The single-block transition function $\widetilde{f}$ in Equation (8) models the transition of a single block as described by the full-state transition f in Equation (1). In Equation (8), it is assumed that $\widetilde{f}$ is the same for every block without loss of generality.

The first sum in Equation (6) accounts for the selection of the block to modify in the proposal. See steps 1 to 4 in Algorithm 1. Note that the proposed state $\mathbf{s}_{\mathrm{k}}^{*}$ and the current state of the Markov chain $\mathbf{s}_{\mathrm{k}}^{m}$ differ in only one block $j \mid j \in \{1,\ldots,\mathrm{N_b}\}$. As a consequence, the sum in Equation (6) across every block is equal to zero for all terms except for one term, when $\mathrm{b} = j$. Thus, when it comes to evaluate the proposal distribution, only the non-zero term is relevant:

$$\frac{1}{\mathrm{N_b}} \sum_{i=1}^{\mathrm{N_p}} \frac{1}{\mathrm{N_p}}\, \mathrm{p}\left(\mathbf{s}_{\mathrm{k},j}^{*}|\mathbf{s}_{\mathrm{k-1},j}^{i}\right) \tag{9}$$

Let us now write the expression for the acceptance ratio in the Metropolis-Hastings algorithm [14],

$$\mathcal{A}\left(\mathbf{s}_{\mathrm{k}}^{m}, \mathbf{s}_{\mathrm{k}}^{*}\right) = \min\left\{1, \frac{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{z}_{1:\mathrm{k}}\right)}{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{z}_{1:\mathrm{k}}\right)}\, \frac{\mathrm{q}\left(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{s}_{\mathrm{k}}^{*}\right)}{\mathrm{q}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{s}_{\mathrm{k}}^{m}\right)}\right\}. \tag{10}$$

Recall that our goal is to obtain an expression from (10) that can be evaluated as required by Algorithm 1. It follows from Equation (4) that

$$\frac{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{z}_{1:\mathrm{k}}\right)}{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{z}_{1:\mathrm{k}}\right)} = \frac{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{*}\right)}{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{m}\right)}\, \frac{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{z}_{1:\mathrm{k-1}}\right)}{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{z}_{1:\mathrm{k-1}}\right)}. \tag{11}$$

Furthermore, Chapman-Kolmogorov equation (3) together with (5) give an approximation of the prediction $\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k-1}}\right)$ in terms of an equally weighted sum of transition distributions:

$$\begin{aligned}\widehat{\mathrm{p}}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k-1}}\right) = \sum_{i=1}^{\mathrm{N_p}} \frac{1}{\mathrm{N_p}}\, \mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k-1}}^{i}\right) = \\ \sum_{i=1}^{\mathrm{N_p}} \frac{1}{\mathrm{N_p}} \prod_{\mathrm{b}=1}^{\mathrm{N_b}} \mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}|\mathbf{s}_{\mathrm{k-1,b}}^{i}\right),\end{aligned} \tag{12}$$

using the assumption of block-independent time evolution. At this moment, by means of inserting Equation (12) and Equation (11) into Equation (10), Equation (9) into Equation (10), and simplifying the constant terms, we obtain:

$$\begin{aligned}\frac{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{z}_{1:\mathrm{k}}\right)}{\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{z}_{1:\mathrm{k}}\right)} \frac{\mathrm{q}\left(\mathbf{s}_{\mathrm{k}}^{m}|\mathbf{s}_{\mathrm{k}}^{*}\right)}{\mathrm{q}\left(\mathbf{s}_{\mathrm{k}}^{*}|\mathbf{s}_{\mathrm{k}}^{m}\right)} = \frac{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{*}\right)}{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{m}\right)} \times \\ \times \frac{\sum_{i=1}^{\mathrm{N_p}} \prod_{\mathrm{b}=1}^{\mathrm{N_b}} \mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}^{*}|\mathbf{s}_{\mathrm{k-1,b}}^{i}\right)}{\sum_{i=1}^{\mathrm{N_p}} \prod_{\mathrm{b}=1}^{\mathrm{N_b}} \mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}^{m}|\mathbf{s}_{\mathrm{k-1,b}}^{i}\right)} \frac{\sum_{i=1}^{\mathrm{N_p}} \mathrm{p}\left(\mathbf{s}_{\mathrm{k},j}^{m}|\mathbf{s}_{\mathrm{k-1},j}^{i}\right)}{\sum_{i=1}^{\mathrm{N_p}} \mathrm{p}\left(\mathbf{s}_{\mathrm{k},j}^{*}|\mathbf{s}_{\mathrm{k-1},j}^{i}\right)}.\end{aligned} \tag{13}$$

This long expression can be written more succinctly as

$$\frac{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{*}\right)}{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{m}\right)} \frac{\sum_{i=1}^{\mathrm{N_p}} \alpha_{\mathrm{k},j}^{*}(i)\ \beta_{\mathrm{k},j}(i)}{\sum_{i=1}^{\mathrm{N_p}} \alpha_{\mathrm{k},j}^{m}(i)\ \beta_{\mathrm{k},j}(i)} \frac{\sum_{i=1}^{\mathrm{N_p}} \alpha_{\mathrm{k},j}^{m}(i)}{\sum_{i=1}^{\mathrm{N_p}} \alpha_{\mathrm{k},j}^{*}(i)}, \tag{14}$$

where

$$\alpha_{\mathrm{k},j}^{m}(i) = \mathrm{p}\left(\mathbf{s}_{\mathrm{k},j}^{m}|\mathbf{s}_{\mathrm{k-1},j}^{i}\right) \tag{15}$$

and

$$\beta_{\mathrm{k},j}\left(i\right)=\prod_{\substack{\mathrm{b}=1\\ \mathrm{b}\neq j}}^{\mathrm{N_b}}\mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}^{m}|\mathbf{s}_{\mathrm{k-1,b}}^{i}\right)=\prod_{\substack{\mathrm{b}=1\\ \mathrm{b}\neq j}}^{\mathrm{N_b}}\mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}^{*}|\mathbf{s}_{\mathrm{k-1,b}}^{i}\right). \quad (16)$$

Finally,

$$\mathcal{A}\left(\mathbf{s}_{\mathrm{k}}^{m},\mathbf{s}_{\mathrm{k}}^{*}\right)= \min\left\{1,\frac{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{*}\right)}{\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}^{m}\right)}\frac{\sum_{i=1}^{\mathrm{N_p}}\alpha_{\mathrm{k},j}^{*}\left(i\right)\ \beta_{\mathrm{k},j}\left(i\right)}{\sum_{i=1}^{\mathrm{N_p}}\alpha_{\mathrm{k},j}^{m}\left(i\right)\ \beta_{\mathrm{k},j}\left(i\right)}\frac{\sum_{i=1}^{\mathrm{N_p}}\alpha_{\mathrm{k},j}^{m}\left(i\right)}{\sum_{i=1}^{\mathrm{N_p}}\alpha_{\mathrm{k},j}^{*}\left(i\right)}\right\}. \quad (17)$$

Remark: at first sight, it might result surprising that the right hand side in Equation (17) seems to depend on j, whereas this variable does not appear in the left hand side. However, further consideration resolves this issue since in fact j is determined by $\mathbf{s}_{\mathrm{k}}^{m}$ and $\mathbf{s}_{\mathrm{k}}^{*}$. By construction in Algorithm 1, $\mathbf{s}_{\mathrm{k}}^{*}$ is different from $\mathbf{s}_{\mathrm{k}}^{m}$ in *only one* block, namely block j. Hence, j is known for a given pair $(\mathbf{s}_{\mathrm{k}}^{m},\mathbf{s}_{\mathrm{k}}^{*})$.

IV. Simulations

A. Two states Gaussian distributed with strong correlation

This section illustrates the difference between the particle approximations of the posteriors obtained via Algorithm 1 using two distinct expressions for the acceptance ratio: one, the ratio of likelihoods as suggested in [9]; and two, Equation (17) as derived in Section III. To this end, we present a rather simple scenario with linear and Gaussian dynamics, Gaussian likelihood, and Gaussian posterior at $\mathrm{k}-1$. Consequently, the updated posterior at time step k can be analytically computed via the classical prediction and update equations of the Kalman filter [15]. Hence, it is possible to compare the particle approximations of the posteriors with the exact closed-form solution.

The scenario considered has state dimension two (composed of two one-dimensional blocks). The time evolution of the system is

$$\mathbf{s}_{\mathrm{k}}=\mathrm{F}\ \mathbf{s}_{\mathrm{k-1}}+\mathbf{v}_{\mathrm{k-1}}, \quad (18)$$

where $\mathrm{F}=3\ \mathrm{I}_2$ and $\mathbf{v}_{\mathrm{k-1}}$ is zero-mean Gaussian noise with covariance $\mathrm{Q}=0.01\ \mathrm{I}_2$. I_n denotes the identity matrix of dimension n. Regarding the observation model, the state is directly observable and the noise is also zero-mean Gaussian with covariance I_2. The observation at time k is $\mathbf{z}_{\mathrm{k}}=[1.5\ 4.5]^{\mathrm{T}}$. Thus, the likelihood specified as a function of $\mathbf{s}_{\mathrm{k}}$, is $\phi\left(\mathbf{s}_{\mathrm{k}};[1.5\ 4.5]^{\mathrm{T}},\mathrm{I}_2\right)$. $\phi\left(\cdot;\boldsymbol{\mu},\Sigma\right)$ denotes the probability density function of a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. Finally, the posterior at the previous time step is given by

$$\mathrm{p}\left(\mathbf{s}_{\mathrm{k-1}}|\mathbf{z}_{1:\mathrm{k-1}}\right)=\phi\left(\mathbf{s}_{\mathrm{k-1}};\begin{bmatrix}0.5\\1.5\end{bmatrix},\frac{1}{4}\begin{bmatrix}1 & 0.98\\0.98 & 1\end{bmatrix}\right). \quad (19)$$

The complete scenario is illustrated in Figure 1. In addition, the exact prediction and updated posterior densities are also plotted.

Figure 2 depicts the results obtained via MCMC sampling using the proposal mechanism in Algorithm 1 for the two

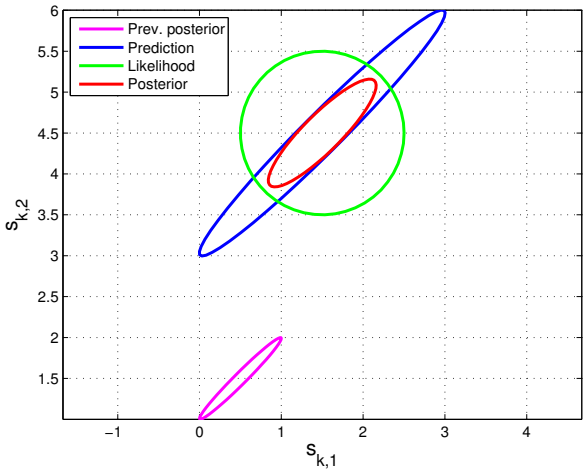

Figure 1. Scenario described in Section IV-A. Since all the functions are Gaussian, contour curves at one standard deviation are used for the representation. Note that the labels along the axes refer to time step k, although for the previous posterior and the prediction more appropriate labels would be $\mathrm{k}-1$ and $\mathrm{k}-1|\mathrm{k}$, respectively.

choices of the acceptance ratio. The true updated posterior is also included for comparison (leftmost plot). Clearly, the sampler where the acceptance probability is equal to the ratio of likelihoods (plot in the middle) does not represent correctly the covariance of the true updated posterior. The remaining of this section provides insight into the results.

Let us consider the two samplers at a certain iteration m. Assume the current sample in both samplers is $\mathbf{s}_{\mathrm{k}}^{m}=[2\ 5]^{\mathrm{T}}$. We say this sample represents a state of high probability in the prediction and in the true updated posterior because it is within one standard deviation in both densities (see Figure 1). Consider now the following possible outcome from the different steps in Algorithm 1: (1) $j=1$; (2) $\mathbf{s}_{\mathrm{k|k-1}}^{*}=[1\ 4]^{\mathrm{T}}$; (3) $\mathbf{s}_{\mathrm{k}}^{*}=[1\ 5]^{\mathrm{T}}$. At this moment, the MCMC samplers perform accept/reject tests. Recall that the current sample is $\mathbf{s}_{\mathrm{k}}^{m}=[2\ 5]^{\mathrm{T}}$ and the proposed sample is $\mathbf{s}_{\mathrm{k}}^{*}=[1\ 5]^{\mathrm{T}}$. The likelihood is the same for both the current and the proposed sample, i.e.

$$\phi\left(\begin{bmatrix}2\\5\end{bmatrix};\begin{bmatrix}1.5\\4.5\end{bmatrix},\mathrm{I}_2\right)=\phi\left(\begin{bmatrix}1\\5\end{bmatrix};\begin{bmatrix}1.5\\4.5\end{bmatrix},\mathrm{I}_2\right).$$

As a consequence, the acceptance probability is equal to one in the sampler using the ratio of likelihoods. Note, however, that the proposed sample is not a state of high probability of the true updated posterior!

Accept/reject tests like the one just illustrated produce precisely the posterior shown in the middle of Figure 2. In other words, consider the region of the $\mathbf{s}_{\mathrm{k},1}\times\mathbf{s}_{\mathrm{k},2}$ plane delimited by the boundaries of the Cartesian product $[0,3]\times[3,6]$. The proposal mechanism, due to its block-wise behaviour, may generate proposals from everywhere within $[0,3]\times[3,6]$, including the upper left and lower right quadrants of the region. Also, the likelihood weighs equally for the points in the four quadrants that are equally far from the region's centre.

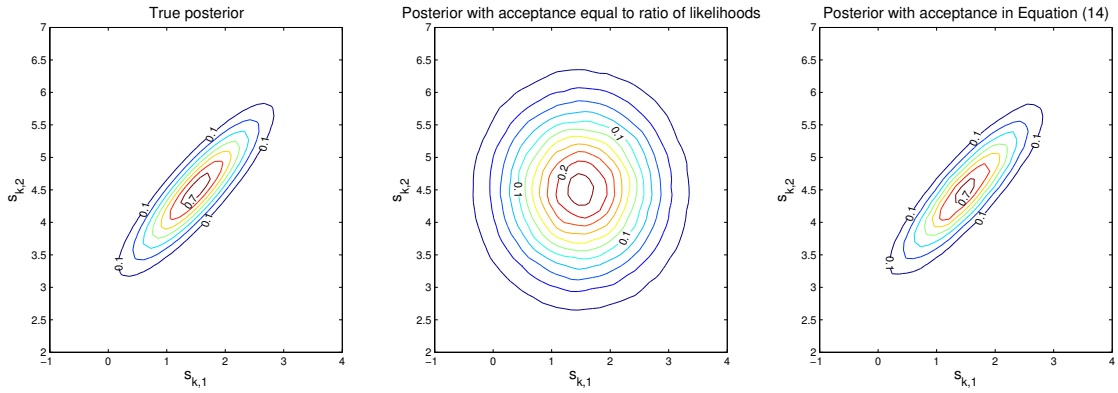

Figure 2. Simulation results in Section IV-A. The leftmost plot corresponds to the true posterior obtained via the Kalman filter. The center and right plots correspond to the distributions obtained via MCMC sampling with acceptance probabilities equal to ratio of likelihoods and the full expression in Equation (17), respectively. The three distributions are represented with contour lines and the numerical labels are associated with density values. One million samples are used by each MCMC sampler and the first one thousand samples are discarded, regarded as burn-in. As it can be seen from the figure, using the ratio of likelihoods as acceptance probability produces a very different posterior than the true one.

However, only the upper right and lower left quadrants are of high probability in the true updated posterior because only in those quadrants the prediction is of high probability too. Since the information of the prediction is disregarded in the acceptance probability when using the ratio of likelihoods, the posterior is wrongly estimated.

On the other hand, the sampler that uses acceptance probability equal to Equation (17) will with high probability reject the proposed sample. Let us explain further. Note that in Equation (13), the term $\sum_{i=1}^{\mathrm{N_p}} \prod_{\mathrm{b}=1}^{\mathrm{N_b}} \mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}}^{*} | \mathbf{s}_{\mathrm{k-1,b}}^{i}\right)$ evaluates the prediction density at the proposed sample $\mathbf{s}_{\mathrm{k}}^{*} = [1\ 5]^{\mathrm{T}}$, which is equal to zero[d] as illustrated in Figure 1. Thus, making the acceptance probability zero.

B. Tracking of two targets

Consider a tracking application based on range-bearing measurements, with no clutter and known data association. More general observation models that do not make these assumptions would also be possible, but the following scenario is simple enough and adequate for our arguments. After introducing the dynamic and observation models, we will compare in this section three MCMC-based particle filters: the original IP-MCMC-PF, the IP-MCMC-PF with the new acceptance probability derived in Section III, and the two-step MCMC particle filter introduced by [8].

Simulation of Trajectories: The trajectories of the targets are simulated using the following equations:

$$\mathbf{a}_{\mathrm{k}} \sim \mathcal{N}\left(\mathbf{0}, \sigma_a^2\ \mathrm{I}_4\right) \tag{20}$$

$$\mathbf{x}_{\mathrm{k}} = \mathbf{x}_{\mathrm{k-1}} + \dot{\mathbf{x}}_{\mathrm{k-1}}\ \Delta_t + \frac{1}{2}\ \mathbf{a}_{\mathrm{k}}\ \Delta_t^2 \tag{21}$$

$$\dot{\mathbf{x}}_{\mathrm{k}} = \dot{\mathbf{x}}_{\mathrm{k-1}} + \mathbf{a}_{\mathrm{k}}\ \Delta_t \tag{22}$$

[d]Consider zero just for the sake of simplicity in the discussion. Rather than exactly zero, the prediction takes a very small value at the since it is Gaussian. In practice, the value is so small compared to the other terms in the acceptance probability that it can be regarded as zero.

Δ_t denotes the time differential between consecutive time steps, $\mathbf{x}$ and $\dot{\mathbf{x}}$ are a vectors formed by the concatenation of the two-dimensional positions and velocities of each target, respectively, and $\mathbf{a}$ represents a random perturbation of the target velocities.

The state vector of the b[th] target comprises four components:

$$\mathbf{s}_{\mathrm{k,b}} = \begin{bmatrix} x_{\mathrm{k,b}} & y_{\mathrm{k,b}} & \dot{x}_{\mathrm{k,b}} & \dot{y}_{\mathrm{k,b}} \end{bmatrix}^{\mathrm{T}}, \tag{23}$$

corresponding to the position and velocity vectors. The complete state vector is the concatenation of the two individual target states: $\mathbf{s}_{\mathrm{k}} = \left[\mathbf{s}_{\mathrm{k},1}{}^{\mathrm{T}}\ \mathbf{s}_{\mathrm{k},2}{}^{\mathrm{T}}\right]^{\mathrm{T}}$.

Motion Model: The motion is specified by a near constant velocity model with Gaussian noise as in Equation (18). In this scenario, $\mathrm{F} = \mathrm{I}_2 \otimes \widetilde{\mathrm{F}}$, where $\otimes$ denotes the Kronecker product and

$$\widetilde{\mathrm{F}} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

The covariance matrix of the process noise is $\mathrm{Q} = \mathrm{I}_2 \otimes \widetilde{\mathrm{Q}}$, where

$$\widetilde{\mathrm{Q}} = \Delta_t^2 \sigma_a^2\ \mathrm{diag}\left(\begin{bmatrix} \frac{\Delta_t^2}{4} & \frac{\Delta_t^2}{4} & 1 & 1 \end{bmatrix}\right). \tag{25}$$

Note that the matrix Q does not follow from Equations (20) to (22); it is our choice of process noise covariance for the motion model. The complete transition distribution can be written as

$$\mathrm{p}\left(\mathbf{s}_{\mathrm{k}} | \mathbf{s}_{\mathrm{k-1}}\right) = \mathrm{p}\left(\mathbf{s}_{\mathrm{k},1} | \mathbf{s}_{\mathrm{k-1},1}\right)\ \mathrm{p}\left(\mathbf{s}_{\mathrm{k},2} | \mathbf{s}_{\mathrm{k-1},2}\right) \tag{26}$$

where the single-target transition is

$$\mathrm{p}\left(\mathbf{s}_{\mathrm{k,b}} | \mathbf{s}_{\mathrm{k-1,b}}\right) = \phi\left(\mathbf{s}_{\mathrm{k,b}}; \widetilde{\mathrm{F}}\ \mathbf{s}_{\mathrm{k-1,b}}, \widetilde{\mathrm{Q}}\right). \tag{27}$$

Observation model: Consider a simplified observation model with perfect measurement-to-track association where a radar sensor generates a single range-bearing measurement per target at each time step. Of course, this approach is only possible because the scenario is simulated and in real applications handling data association is a requirement. We take these simplifying assumptions because they ease the study of the properties of the MCMC samplers, main goal of this paper.

Similar to the state vector, the complete observation is the concatenation of the individual target observations:

$$\mathbf{z}_{\mathrm{k}} = \begin{bmatrix} \mathbf{z}_{\mathrm{k},1}^{\mathrm{T}} & \mathbf{z}_{\mathrm{k},2}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{28}$$

$$\mathbf{z}_{\mathrm{k,b}} = \begin{bmatrix} r_{\mathrm{k,b}} & \alpha_{\mathrm{k,b}} \end{bmatrix}^{\mathrm{T}}, \tag{29}$$

where $r_{\mathrm{k,b}}$ and $\alpha_{\mathrm{k,b}}$ denote the observed range and bearing angle. The noise in the observations of each target is additive and Gaussian distributed with variance σ_r^2 for the range and σ_α^2 for the bearing. The complete likelihood function is

$$\mathrm{p}\left(\mathbf{z}_{\mathrm{k}}|\mathbf{s}_{\mathrm{k}}\right) = \mathrm{p}\left(\mathbf{z}_{\mathrm{k},1}|\mathbf{s}_{\mathrm{k},1}\right)\ \mathrm{p}\left(\mathbf{z}_{\mathrm{k},2}|\mathbf{s}_{\mathrm{k},2}\right) \tag{30}$$

where the single-target likelihood is given by

$$\mathrm{p}\left(\mathbf{z}_{\mathrm{k,b}}|\mathbf{s}_{\mathrm{k,b}}\right) = \phi\left(r_{\mathrm{k,b}}; g^r\left(\mathbf{s}_{\mathrm{k,b}}\right), \sigma_r^2\right)\ \phi\left(b_{\mathrm{k,b}}; g^\alpha\left(\mathbf{s}_{\mathrm{k,b}}\right), \sigma_\alpha^2\right). \tag{31}$$

Range and bearing are defined as

$$g^r\left(\mathbf{s}_{\mathrm{k,b}}\right) = \sqrt{x_{\mathrm{k,b}}^2 + y_{\mathrm{k,b}}^2} \tag{32}$$

$$g^\alpha\left(\mathbf{s}_{\mathrm{k,b}}\right) = \arctan\left(\frac{y_{\mathrm{k,b}}}{x_{\mathrm{k,b}}}\right). \tag{33}$$

Parameters: Targets are initialised with $\mathbf{x}_0 = [2\ 2]^{\mathrm{T}}$ [km] and $\mathbf{v}_0 = [100\ 100]^{\mathrm{T}}$ [m s^{-1}]. The simulations start at time step $\mathrm{k} = 1$ and last for 50 time steps. Time between consecutive time steps is equal to $\Delta_t = 1$ [s]. The variances of the noises are $\sigma_a^2 = 50$ [m^2 s^{-4}], $\sigma_r^2 = 10^3$ [m^2], and $\sigma_\alpha^2 = 10^{-5}$ [rad^2].

For the initialisations of the filters, Gaussian densities centered around the true positions and velocities are specified. The position and velocity standard deviations are respectively equal to $500/3$ [m] and 1 [m s^{-1}]. Concerning the MCMC-based filters, the burn-in length is equal to 50.

Results: In spite of the nonlinearity in the measurement function, the extended Kalman filter (EKF) is able to perform well in this problem. In fact, we will consider the performance attained by the EKF as the baseline. Consequently, we are able to conclude when each of the MCMC-based particle filters analysed here attains a (close-to) optimal performance.

The performance measure chosen is the average position error (APE), defined as

$$\mathrm{APE}_{\mathrm{k}} := \frac{1}{2\mathrm{N}_{\mathrm{MC}}} \sum_{n=1}^{\mathrm{N}_{\mathrm{MC}}} \sum_{\mathrm{b}=1}^{2} \sqrt{\left(\hat{x}_{\mathrm{k,b}}^n - x_{\mathrm{k,b}}^n\right)^2 + \left(\hat{y}_{\mathrm{k,b}}^n - y_{\mathrm{k,b}}^n\right)^2}, \tag{34}$$

where N_{MC} denotes the number of Monte Carlo simulations (100 in our experiments), $x_{\mathrm{k,b}}^n$ and $y_{\mathrm{k,b}}^n$ the true target locations, and $\hat{x}_{\mathrm{k,b}}^n$ and $\hat{y}_{\mathrm{k,b}}^n$ the estimated locations.

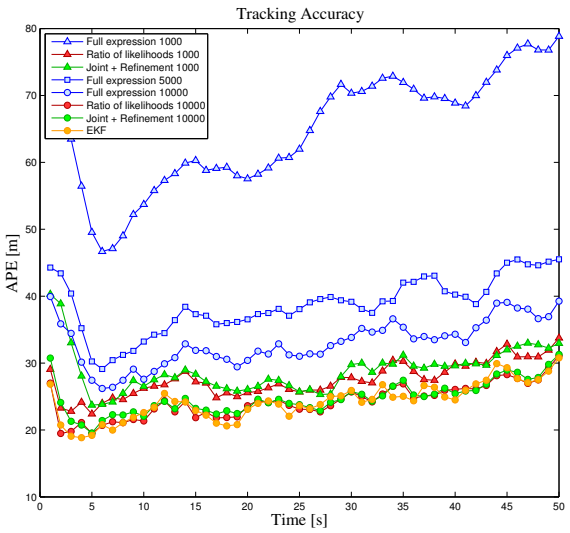

Figure 3. Simulation results in Section IV-B.

Figure 3 shows the results of the MCMC-PF with the two choices of acceptance probability, namely ratio of likelihoods and the full expression in Equation (17), and the two-step sequential MCMC introduced in [8]. The results are perhaps surprising at this moment since the ratio of likelihoods MCMC presents a good performance already with 1000 particles, whereas the full expression MCMC's performance is worse than the former even with 10000 particles. Nevertheless, the results do agree with the good performance of the ratio of likelihoods previously reported in the literature. Let us analyse these results further in the following paragraphs.

Recall the factorisation of the time evolution in Equation (26) and the likelihood function in Equation (30). Providing the initial distribution $\mathrm{p}\left(\mathbf{s}_0\right)$ also factorises,

$$\mathrm{p}\left(\mathbf{s}_0\right) = \mathrm{p}\left(\mathbf{s}_{0,1}\right)\ \mathrm{p}\left(\mathbf{s}_{0,2}\right), \tag{35}$$

the prediction $\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k}-1}\right)$ will also factorise as a result of the recursive application of the Bayes filter's prediction and update equations. That is,

$$\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}|\mathbf{z}_{1:\mathrm{k}-1}\right) = \mathrm{p}\left(\mathbf{s}_{\mathrm{k},1}|\mathbf{z}_{1:\mathrm{k}-1,1}\right)\ \mathrm{p}\left(\mathbf{s}_{\mathrm{k},2}|\mathbf{z}_{1:\mathrm{k}-1,2}\right). \tag{36}$$

The acceptance ratio as written in Equation (13) consists of three factors: the ratio of likelihoods, the ratio of predictions, and the inverse ratio of predictions *only for the block* that is modified in the proposal. We say inverse ratio to denote that in the last factor, differently from the former two, the proposed and current states are in the denominator and numerator, respectively. Providing Equation (36), all the factors in Equation (13) except for the ratio of likelihoods are cancelled.

The discussion so far concludes why the ratio of likelihoods sampler produces good results in the scenario analysed in this section even though it is known from Sections III and IV-A that the ratio of likelihoods does not correspond with the complete expression of the acceptance probability. Still, we notice in

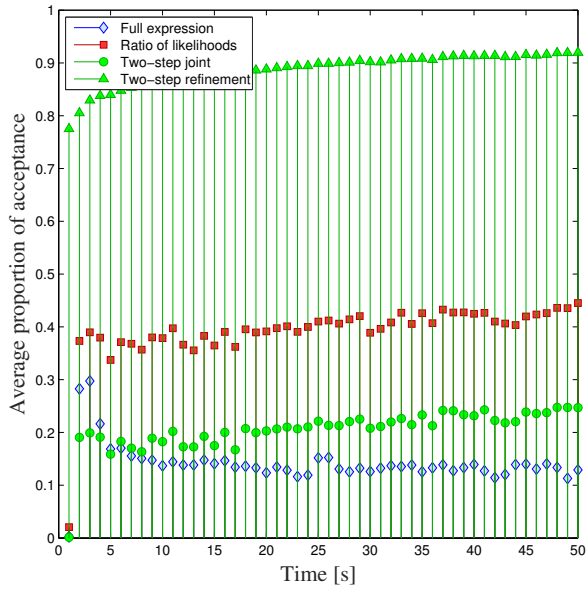

Figure 4. Proportion of acceptance of the MCMC-based particle filters in the scenario described in Section IV-B. The number of particles used in the filters is equal to 10000 and the results ae obtained averaging the proportions of acceptance across 100 Monte Carlo runs. In case of the two-step sampler [8], the joint draw has lower acceptance than the refinement draw, which is expected because the joint draw operates in a state space of higher dimension than the refinement. Compare with Figure 3 to note that the low acceptance in the full expression corresponds to low performance. Similar reasoning applies to the good performance of the ratio of likelihoods. In case of the two-step sampler, the low acceptance of the joint step together with the rather large acceptance of the refinement results in good performance.

Figure 3 that the full expression needs more particles to attain less error. That is, it has slow convergence. Figure 4 shows that the proportion of acceptance[e] of the full expression is very low. In other words, the use of the full expression results in an inefficient sampler in the sense that most of the proposed samples are rejected (at least when using the prediction as proposal and in this scenario).

Finally, the third MCMC-based particle filter presented in [8] is included in this experiment for comparison. The performance of this filter in this scenario is good and we remark that the derivation of its acceptance probability does not rely on factorisations of the prediction. In fact, the acceptance probability in this filter is equal to the ratio of likelihoods in the joint draw, and the same ratio of likelihoods weighted by the ratio of the transition distribution in the refinement draw. This stems from sampling the joint (k and $\mathrm{k}-1$) filtering distribution $\mathrm{p}\left(\mathbf{s}_{\mathrm{k}}, \mathbf{s}_{\mathrm{k}-1} | \mathbf{z}_{1:\mathrm{k}}\right)$.

V. Conclusion

The main focus of this paper has been to complete the acceptance probability expression of the IP-MCMC-PF [9]. This is an MCMC-based particle filter whose main appealing lies in leveraging parallel processing in modern computer architectures. In order to clarify this contribution, we have not emphasised the simulation of several Markov chains in parallel. The use of multiple Markov chains is what gives the filter in [9] the interacting qualifier (thus IP – interacting population). Nonetheless, the acceptance probability derived in Section III is the same independently of the number of chains. Furthermore, in Section IV-A we have shown the importance of the new terms in the acceptance probability.

Future work includes the study of the ratio of likelihoods as a procedure to compute the acceptance probability. This is correct in the IP-MCMC-PF when the initial distribution, the transition distribution, and the likelihood function factorise as in Section IV-B. Further exciting analysis comprises whether less restricting assumptions on the models, or, alternatively, modifications in the proposal mechanism outlined in Algorithm 1, result in the acceptance probability being the ratio of likelihoods, which produces an efficient sampler as observed in the experiments.

[e]We define the proportion of acceptance as the number of accepted proposals divided by the total number of MCMC iterations.

Acknowledgment

The research leading to these results has received funding from the EU's Seventh Framework Programme under grant agreement N°607400. The research has been carried out within the **TRAX** project.

References

[1] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer, 2001.
[2] Z. Chen, "Bayesian filtering: from Kalman filters to particle filters, and beyond," *Statistics*, 2003.
[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Transactions on Signal Processing*, 2002.
[4] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *Proceedings of Radar and Signal Processing*, 1993.
[5] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, 2008.
[6] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, 2003.
[7] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *Transactions on Pattern Analysis and Machine Intelligence*, 2005.
[8] S. K. Pang, J. Li, and S. J. Godsill, "Models and algorithms for detection and tracking of coordinated groups," in *Proceedings of the International Aerospace Conference*, 2008.
[9] M. Bocquel, H. Driessen, and A. Bagchi, "Multitarget tracking with interacting population-based MCMC-PF," in *Proceedings of the International Conference on Information Fusion*, 2012.
[10] A. Carmi, F. Septier, and S. J. Godsill, "The Gaussian mixture MCMC particle algorithm for dynamic cluster tracking," *Automatica*, 2012.
[11] F. J. Iglesias García, M. Bocquel, and H. Driessen, "Langevin Monte Carlo filtering for target tracking," in *Proceedings of the International Conference on Information Fusion*, 2015.
[12] F. Septier and G. W. Peters, "Langevin and Hamiltonian based Sequential MCMC for Efficient Bayesian Filtering in High-dimensional Spaces," *arXiv preprint arXiv:1504.05715*, 2015.
[13] M. Bocquel, "Random finite sets in multi-target tracking: efficient sequential MCMC implementation," Ph.D. dissertation, Universiteit Twente, 2013.
[14] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *The American Statistician*, 1995.
[15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.