

Multiple Model Sequential MCMC for Jump Markov Systems

Mayazzurra Ruggiano, Melanie Bocquel, and Hans Driessen

Thales Nederland B.V. - Sensors TBU Radar Engineering,
Hengelo, Nederland

Email: {Mayazzurra.Ruggiano, Melanie.Bocquel, Hans.Driessen} @nl.thalesgroup.com

Abstract—The problem of discerning between multiple models arises in several domains and applications. In general the use of models allows an improved and more robust processing of the data, when there is a good match between the model and the data. Consequently correct model selection is essential. New algorithms are thus investigated here which can cope with multiple models. In particular this is shown for the example of tracking of targets presenting sudden maneuver. Particle filter (PF) techniques based on the Interacting Population Markov Chain Monte Carlo (IP-MCMC) scheme present more degrees of freedom in algorithm design with respect to classical Sampling importance resampling (SIR) PF. In this paper two families of IP-MCMC PF algorithms are proposed, respectively Interacting Multiple Models IP-MCMC (IMM-IP-MCMC) and Multiple Model Selection IP Reversible Jump MCMC (MMS-IP-RJMCMC), to tackle the problem of motion model selection for highly maneuverable targets and compared to the Interacting Multiple Models SIR (IMM-SIR) PF implementation. Simulations demonstrate that the proposed algorithms yield a more robust performance than the SIR implementation.

Keywords— *multiple model; Markov Chain Monte Carlo; particle filtering; Interacting Multiple Models; Reversible Jump*

I. INTRODUCTION

Jump Markov Systems (JMS) are dynamic systems described with the use of multiple models (MM), where the system model evolves over time according to a Markov chain. This representation has been widely used in signal processing, radar/sonar target tracking, and system control applications, among others. JMS are commonly used in target tracking to capture the dynamics of maneuvering targets, see [1], [2]. This paper deals with the problem of state estimation in such systems with an application in target tracking.

One of the currently most popular multiple model filtering techniques in target tracking is the Interacting Multiple Model (IMM) filter, refer to [1], [2]. The algorithm comprises a set of Kalman filters running in parallel, one for each of the models of target behavior, in conjunction with an interaction step referred to as mixing [1]. Although it still is one of the most cost-effective state estimation schemes in this field, it also is known to be sub-optimal even if all models are linear and Gaussian. Moreover in some applications, the models of the

target dynamics and/or the observations can exhibit severe nonlinearities. Since the IMM relies on (extended) Kalman filters and Gaussian approximation in the mixing, its performance can be deteriorated. In this paper we will explore Particle Filtering (PF) schemes to overcome such problems.

Solutions using Particle Filtering (PF) have been proposed in the past ten years, see [3], [4]. PF approach is a Sequential Monte Carlo (SMC) simulation-based method that approximately solves the full Bayes prediction and update equations recursively using a set of stochastic samples, known as particles. For an overview of the field we refer to [5]. The standard PF approach [6] resorts to a sequential version of the importance resampling (SIR) procedure. These techniques have been extended for multiple model filtering in two ways. The first approach [3] augments the state with the mode variable and applies a SIR PF to this augmented state. A drawback of this approach is that there is no direct control over the number of particles in a mode [7]. The latter approach, [4], [8], is an SMC version of the IMM filter, where one has full control over the number of particles in each mode. In the first version of this filter [8] the mixing step was implemented via regularized sampling. In [4] the mixing step was implemented more efficiently via a direct sampling approach. An extension of this technique for tracking in clutter was developed in [9]. However, all these filters are based on the basic SIR scheme, which makes them ineffective due to the problem of sample impoverishment, i.e. the lack of diversity among samples.

An alternative approach to the SIR filters is based on Markov Chain Monte Carlo (MCMC) sampling. In recent years there has been an increasing interest towards sequential MCMC methods. The Interacting Population based MCMC PF (IP-MCMC PF) detailed in [10] is an efficient implementation exploiting parallel computations. These MCMC techniques will be the focus of the multiple model algorithms developed in this paper.

The paper is organized as follows. The multiple model filtering problem is described in section II. Section III recalls the two families of algorithms, i.e. Bootstrap and SIR-based IMM. Section IV briefly reviews sequential Markov Chain Monte Carlo methods. We explain the new multiple model MCMC algorithms in section V. Section VI collects simulations and results. Final considerations and direction for future work are given in section VII.

This research is conducted as part of the Sensor Technology Applied in Reconfigurable systems for sustainable Security (STARS) project. For further information, please see the website www.starsproject.nl.

II. MULTIPLE MODEL FILTERING

A. System setup models

The filtering problem consists of a Jump Markov System (JMS) [6]. The hybrid state of the system is represented by the state vector $\mathbf{s}_{H,k}$, comprising continuous-valued parameters, the state dynamics $\mathbf{s}_{k,m_k} \in \mathcal{S} \subset \mathbb{R}^{d_{m_k}}$, and a discrete part, the state mode $m_k \in \mathcal{M} \subset \mathbb{N}$: $\mathbf{s}_{H,k} = [\mathbf{s}_{k,m_k}^T, m_k]^T$. The underlying model m_k can jump between different values in $\{1, 2, \dots, M\}$. The time behavior of m_k is generally modeled as first order homogeneous Markov chain with a fixed transition probability matrix. Consider the general JMS. The mode transition of the system is modeled by a Markov chain:

$$P\{m_{k+1} = i | m_k = j\} = [\Pi(t_k)]_{ij}. \quad (1)$$

The dynamical model of the state of the system is described by:

$$\mathbf{s}_{H,k+1} = f(t_k, \mathbf{s}_{H,k}, \mathbf{w}_{k,m_k}). \quad (2)$$

The measurement follows a non-linear equation, i.e.:

$$\mathbf{z}_k = h(t_k, \mathbf{s}_{H,k}, \mathbf{n}_k), \quad (3)$$

with $\Pi(t_k)$ the transition matrix, f the evolution function and $\mathbf{w}_{k,m_k}$ the process noise, t_k the k -th time instant, h the measurement function, the measurement $\mathbf{z}_k \in \mathbb{R}^p$, and the measurement noise $\mathbf{n}_k$. Note that the dimension d_{m_k} of the state vector $\mathbf{s}_{k,m_k}$ depends on the mode, however for ease of notation, in the following the subscript m_k in $\mathbf{s}_{k,m_k}$ will be dropped.

B. Problem statement

Consider the system represented by (1), (2) and (3), and assume available the initial probability density function (pdf) $p(\mathbf{s}_{H,0})$. Then the optimal hybrid filtering problem is the problem of constructing the posterior pdf $p(\mathbf{s}_{H,k} | \mathbf{z}_{1:k})$, for $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$ the collection of measurements over time.

Within the context of the particle filtering approach the goal is to approximate this posterior pdf with a finite and discrete set of points or particles. The goal of the implementation of the particle filters, in [1] and [4], has been the recursive construction of this pdf as new measurements enter the system.

III. FAMILIES OF ALGORITHMS

Since the system is a JMS and the mode of the system is unknown, just like the dynamics state vector, two alternatives are viable for the PF implementation, depending on whether the mode estimation is split or not from the state dynamics estimation done within the filtering in the PF. Here the two families are recalled for the SIR implementation of the PF, while in section V the new algorithms with the MCMC implementation of the PF. In the first case the solution comprises the extension of the dynamics state vector with the discrete mode and the straightforward application of the SIR PF [3]. In the second case the solution is of the interacting

multiple model type, characterized by two separate steps, the mixing stage and the mode conditioned filtering, both implemented via particle approximations, i.e. IMM-SIR PF in [4] and [8].

A. Bootstrap PF

The idea for the multiple model bootstrap filter directly follows from the Bayesian recursion of the joint density for the dynamics state vector and the discrete mode. This recursion is given by:

$$p(\mathbf{s}_{H,k} | \mathbf{z}_{1:k}) = \frac{1}{c} p(\mathbf{z}_k | \mathbf{s}_{H,k}) p(\mathbf{s}_{H,k} | \mathbf{z}_{1:k-1}), \quad (4)$$

where c is a normalization constant. The SIR or Bootstrap PF is then applied to the joint state $\mathbf{s}_{H,k}$. The prediction part simulates then the dynamics as well as the Markov process of the mode evolution.

B. IMM-SIR PF

The algorithm proposed in [1] decouples the probability of being in a mode and the probability of having a state given a mode by using the identity:

$$p(\mathbf{s}_{H,k} | \mathbf{z}_{1:k}) = p(m_k | \mathbf{z}_{1:k}) p(\mathbf{s}_k | m_k, \mathbf{z}_{1:k}). \quad (5)$$

The filtering problem is split into two stages, a mixing stage and a mode conditioned filtering stage. In the mixing stage compute:

$$p(\mathbf{s}_{k-1}, m_k | \mathbf{z}_{1:k-1}) = p(m_k | \mathbf{z}_{1:k-1}) p(\mathbf{s}_{k-1} | m_k, \mathbf{z}_{1:k-1}), \quad (6)$$

where the measurement conditioned mode update can be expressed as

$$p(m_k | \mathbf{z}_{1:k-1}) = \sum_{m_{k-1}} p(m_k | m_{k-1}) p(m_{k-1} | \mathbf{z}_{1:k-1}),$$

and

$$p(\mathbf{s}_{k-1} | m_k, \mathbf{z}_{1:k-1}) = \sum_{m_{k-1}} p(m_{k-1} | m_k, \mathbf{z}_{1:k-1}) p(\mathbf{s}_{k-1} | m_{k-1}, \mathbf{z}_{1:k-1}),$$

$$\text{with } p(m_{k-1} | m_k, \mathbf{z}_{1:k-1}) = \frac{p(m_k | m_{k-1}) p(m_{k-1} | \mathbf{z}_{1:k-1})}{p(m_k | \mathbf{z}_{1:k-1})}.$$

In the mode conditioned filtering stage compute:

$$p(\mathbf{s}_k, m_k | \mathbf{z}_{1:k}) = p(m_k | \mathbf{z}_{1:k}) p(\mathbf{s}_k | m_k, \mathbf{z}_{1:k}), \quad (7)$$

$$\text{where } p(\mathbf{s}_k | m_k, \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{s}_k, m_k) p(\mathbf{s}_k | m_k, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, m_k)}.$$

In [4] and [8], these steps were implemented by means of particle approximations in a SIR filter.

IV. MCMC METHODS

The objective is to obtain an approximation of the posterior pdf $p(\mathbf{s}_{H,k} | \mathbf{z}_{1:k})$ through the implementation of a PF capable of benefitting as much as possible from the prior and modeling knowledge. The ease of incorporating prior knowledge through an MCMC process put the focus of this work on exploring the opportunities and effects of this technique.

In particular the sampling MCMC procedures [11] considered here are the ones based on the Metropolis-Hastings sampler and its generalization Reversible Jump sampler [12], respectively for the two families of algorithms described in the next section in more detail. Both these sampling techniques are based on the concept of sample proposal and probabilistic acceptance. These samplers are proven to provide samples that are asymptotically distributed according to the targeted pdf, i.e. for large number of samples, by means of a guideline in the sequences of operations to perform. However in the actual design of the algorithm the main degrees of freedom are devising the proposal density and consequently the acceptance probability. Details on the design are given in the next sections. It should be noted that they present a burn-in period B that has to be discarded before the sampler has reached the stationary state and independency from the initialization point. The elapse of the burn-in can be verified by indirect measures such as convergence diagnostics like the Gelman-Rubin test [13]. For N_{MCMC} independent chains of length in samples $N+B$, compute the between-chain variance B_{MCMC} and the within-chain variance W and obtain the posterior variance as

$$\widehat{\sigma^2} = \frac{N-1}{N} W + B_{MCMC}. \text{ As a rule of thumb, the convergence}$$

test is passed when the value of the ratio $r_N = \sqrt{\widehat{\sigma^2}/W}$ is less than 1.1. It should be noted however that a priori dimensioning the length of the burn-in can be non-trivial and is not addressed here. Nevertheless a stationary chain does not guarantee the fact that the samples obtained are independently and identically distributed (iid) variables of the target distribution.

As the Reversible Jump sampler (RJ-sampler) [12] can be interpreted as a generalization of the Metropolis-Hastings sampler (MH-sampler) [14], the RJ-sampler will be first recalled and then the particular case of MH-sampler. The target density can be expressed as:

$$\pi(\mathbf{s}_{H,k}) = p(\mathbf{s}_{H,k} | \mathbf{z}_{1:k}) \propto p(\mathbf{s}_k | m_k) p(\mathbf{z}_{1:k} | \mathbf{s}_{H,k}) p(m_k). \quad (8)$$

The RJ-sampler comprises the following steps, given the current state vector $\mathbf{s}_k^{(\zeta)}$ and the mode $m_k^{(\zeta)}$:

- Propose a mode $\tilde{m}_k^{(\zeta)} \sim q(\tilde{m}_k | m_k^{(\zeta)})$;
- From the current state vector $\mathbf{s}_k^{(\zeta)}$, and the proposed mode $\tilde{m}_k^{(\zeta)}$, propose now $\tilde{\mathbf{s}}_k^{(\zeta)} \sim q(\tilde{\mathbf{s}}_k | \mathbf{s}_k^{(\zeta)}, \tilde{m}_k^{(\zeta)})$;
- Compare $u_k^{(\zeta)} \sim \mathcal{U}(0,1)$ with the acceptance probability $\alpha_{RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k)$, and select:

$$\mathbf{s}_k^{(\zeta+1)} = \begin{cases} \tilde{\mathbf{s}}_k^{(\zeta)} & \text{if } u_k^{(\zeta)} \leq \alpha_{RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \\ \mathbf{s}_k^{(\zeta)} & \text{otherwise} \end{cases}, \quad (9)$$

with

$$\alpha_{RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) = \min \left\{ \frac{\pi(\tilde{\mathbf{s}}_{H,k}) q(m_k | \tilde{m}_k) q(\mathbf{s}_k | \tilde{\mathbf{s}}_k, m_k)}{\pi(\mathbf{s}_{H,k}) q(\tilde{m}_k | m_k) q(\tilde{\mathbf{s}}_k | \mathbf{s}_k, \tilde{m}_k)}, 1 \right\}. \quad (10)$$

The ratio term in the acceptance probability of the RJ-sampler given in (10) will be developed further in the following section, so rewrite (10) as:

$$\alpha_{RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) = \min \{ \alpha_{f,RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k), 1 \}. \quad (11)$$

The collection of $\mathbf{s}_k^{(\zeta)}$ for $\zeta = 0, \dots, T_{MH} - 1$ are the desired samples. In the RJ-sampler the proposal is not only regarding the dynamics state vector but also a possible mode change through the proposal of $\tilde{m}_k$. Since for Markov transitions the new mode $\tilde{m}_k$ just depends on the current state vector mode m_k , then $q(\tilde{m}_k, \tilde{\mathbf{s}}_k | \mathbf{s}_k, m_k) = q(\tilde{m}_k | m_k) q(\tilde{\mathbf{s}}_k | \mathbf{s}_k, \tilde{m}_k)$. The RJ-sampler can be viewed as a generalization of the MH-sampler. For the MH-sampler the mode is not proposed but given, so an MH-sampler is applied separately for each mode. For the MH-sampler the proposal density, instead of $q(\tilde{m}_k, \tilde{\mathbf{s}}_k | \mathbf{s}_k, m_k)$, is set to $q(\tilde{\mathbf{s}}_k | \mathbf{s}_k, m_k)$. So the ratio term for the MH-sampler is:

$$\alpha_{f,MH}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) = \frac{\pi(\tilde{\mathbf{s}}_k, m_k) q(\mathbf{s}_k | \tilde{\mathbf{s}}_k, m_k)}{\pi(\mathbf{s}_k, m_k) q(\tilde{\mathbf{s}}_k | \mathbf{s}_k, m_k)}. \quad (12)$$

V. MULTIPLE MODEL MCMC ALGORITHMS

In section III, two multiple model families of algorithms have been recalled based on the SIR implementation of the PF to estimate the hybrid state of the JMS. Analogously here two new multiple model algorithms are proposed. In the first case the implementation is of the interacting multiple model type, with two separated steps of the mixing and mode conditioned filtering. This leads to the IMM-IP-MCMC PF algorithm. In the second case the implementation is a single step of mode conditioned filtering, where the difference lies in allowing mode mixing, or better “sampling” with the RJ sampler MCMC implementation, internally to the filter. This leads to the MMS-IP-RJMCMC PF. These two implementations are described in the following subsections. Note that a hybrid implementation between the two is also possible but is not addressed here. Both families of algorithms present mode conditioned filtering. This includes a mode conditioned prediction step and an update of the likelihood $p(\mathbf{z}_k | \mathbf{s}_{H,k})$.

A. IMM-IP-MCMC

The interest here is to use the Interacting Multiple Model concept in construction with an MCMC implementation. The IMM implementation follows the one by direct sampling given in [4] and [8], which is here recalled. A two stage recursive algorithm is constructed with a mixing stage separated from the mode conditioned filtering. In the algorithm presented in this paper the mode conditioned filtering is based on MCMC instead of the SIR implementation in [4].

The scheme is based on the factorization of the expression of the posterior density in (5). The mode prior probability distribution $p(m_{k-1})$ is computed at every time step. A recursion is applied and composed of the two terms of (5), which are computed sequentially at every time step: a mixing step computes $p(\mathbf{s}_{k-1, m_k}, m_k | \mathbf{z}_{1:k-1})$; the filtering scheme, in this case MCMC-based, computes $p(\mathbf{s}_{k, m_k} | m_k, \mathbf{z}_{1:k})$. In the mixing stage is a weighted sum of distributions with the

particle cloud representing an approximation of the posterior pdf at the previous time step for each mode.

Consequently in this implementation the IMM is applied on the M PF clouds at $k-1$ and performs the mode mixing between the M clouds; this is then provided as an input to the MCMC scheme, which is applied to each m -th cloud separately, and through MH-sampling explores the space of the state dynamics vector for that mode. Finally the outputs of the MCMC schemes provide the M clouds at k .

In the case of the IMM-IP-MCMC the ratio term in the acceptance probability of the MH-sampler (12) can be expressed using (8) as:

$$\alpha_{f,MH}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \propto \frac{p(\tilde{\mathbf{s}}_k | m_k) p(\mathbf{z}_{1:k} | \tilde{\mathbf{s}}_k, m_k) p(m_k) q(\mathbf{s}_k | \tilde{\mathbf{s}}_k, m_k)}{p(\mathbf{s}_k | m_k) p(\mathbf{z}_{1:k} | \mathbf{s}_k, m_k) p(m_k) q(\tilde{\mathbf{s}}_k | \mathbf{s}_k, m_k)}, \quad (13)$$

and due to the particle representation, i.e. the prior distribution of the state given the mode is the particle instance, can be developed further:

$$\alpha_{f,MH}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \propto \frac{p(\mathbf{z}_{1:k} | \tilde{\mathbf{s}}_k, m_k)}{p(\mathbf{z}_{1:k} | \mathbf{s}_k, m_k)}. \quad (14)$$

So in this case the ratio in the acceptance probability is a ratio of likelihoods of the proposal and the seed respectively. This result is intuitive because these two states belong to the same cloud and have the same mode.

The parameters are: the total number of particles $N_p(m_k)$, $N(m_k)$ particles per chain, and $N_{MCMC}(m_k) = N_p(m_k) / N(m_k)$ number of parallel chains, a burn-in of $B(m_k)$, respectively per mode m_k . Let $N_{tot}(m_k) = N_p(m_k) + N_{MCMC}(m_k)(N(m_k) + B(m_k))$ be the total number of particles that are generated in the prediction step and in the MH sampler (independently from the fact that they are accepted or not). All the likelihoods of these particles can be used to estimate the mode probability. The mode probability is then computed per mode $\beta \in \mathcal{M}$ at each time step k as

$$p(m_k = \beta) = \frac{\sum_{i=1}^{N_{tot}(m_k)} l_{k,\beta}^{(i)}}{\sum_{m_k=1}^M \sum_{i=1}^{N_{tot}(m_k)} l_{k,m_k}^{(i)}}. \quad (15)$$

Note that in fact using the $N_{tot}(m_k)$ particles, instead of just $N_p(m_k)$, allows to benefit from the likelihoods anyways calculated within the MH sampling procedure for computing the acceptance probability.

Algorithm A: IMM-IP-MCMC

Input: $p(m_{k-1})$ mode prior probability distribution,

$\{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p(m_{k-1})}$ with $m_{k-1} = 1, \dots, M$ and $N_p(m_{k-1})$ particle clouds at time $k-1$ for a given mode, and measurement $\mathbf{z}_k$.

Output: $p(m_k)$ mode posterior probability, $\{\mathbf{s}_k^{(i)}\}_{i=1}^{N_p(m_k)}$ with $m_k = 1, \dots, M$ and $N_p(m_k)$ updated particle clouds at time k .

1. Mixing stage

Propagate $p(m_{k-1})$ using $\Pi(t_k)$, obtain $\{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p(m_k)}$.

for $m_k = 1, \dots, M$

Apply (2) to $\{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p(m_k)}$ and associate to the seeds their likelihood $l_k^{(i)} = p(\mathbf{z}_k | \mathbf{s}_k^{(i)})$ and obtain $\{\mathbf{s}_k^{(i)}, l_k^{(i)}\}_{i=1}^{N_p(m_k)}$.

2. Initialization or “seed generation” step

Randomly pick $\{\mathbf{s}_k^{(v)}, l_k^{(v)}\}_{v=1}^{N_{MCMC}(m_k)} \subset \{\mathbf{s}_k^{(i)}, l_k^{(i)}\}_{i=1}^{N_p(m_k)}$.

for $v=1: N_{MCMC}(m_k)$

3. MH sampling step

Initialize $\mathbf{s}_k^{(v, n_{samp})} = \mathbf{s}_k^{(v)}$, with $n_{samp}=1$.

for $n_{samp}=1: N(m_k) + B(m_k)$

Randomly pick $\mathbf{s}_{k-1}^{(i)}$ from $\{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p(m_k)}$. Draw $u \sim \mathcal{U}(0,1)$, update according to the acceptance probability and obtain

$$\mathbf{s}_k^{(v, n_{samp})} = \begin{cases} \mathbf{s}_k^{(i)} & \text{if } u \leq \alpha_{MH}(\mathbf{s}_k^{(v, n_{samp})}, \mathbf{s}_k^{(i)}) \\ \mathbf{s}_k^{(\xi, n_{samp})} & \text{otherwise} \end{cases}$$

end

Thus obtain $\{\mathbf{s}_k^{(v, n_{samp})}\}_{n_{samp}=1}^{N(m_k)}$.

end

Compute r_N , extend the burn-in $B(m_k)$ if $r_N > 1.1$. Obtain:

$$\{\mathbf{s}_k^{(i)}\}_{i=1}^{N_p(m_k)} = \{\mathbf{s}_k^{(v, n_{samp})}\}_{v=1, n_{samp}=1}^{N_{MCMC}(m_k), N(m_k)}.$$

end

4. Obtain the mode probability $p(m_k)$ as in (15).

B. MMS-IP-RJMCMC

In the Multiple Model Selection IP-RJMCMC (MMS-IP-RJMCMC) the mode is, together with the dynamics vector, composing the state vector that is estimated by means of the RJ-sampler within each MCMC chain of the IP-MCMC. This formulation of the problem is completely different from the one described in the previous subsection where the MH-samplers are applied independently per mode.

The fact that the target can have different modes implicitly implies that these modes can have different terms in the state vector and different total state vector dimensions, and this imposes a different scheme for the sampling of the extended state vector, comprising the mode and the dynamics. Reversible Jump sampler is an MCMC framework specifically designed to sample from target spaces that contain vectors of

different dimensions. The main idea behind the RJMCMC is that the process is allowed to jump between parameter spaces belonging to different models and of different dimensions. As mentioned in the previous section the RJMCMC can be considered as a generalization of the MH-sampler.

In the case of the MMS-IP-RJMCMC the ratio term (10) in the acceptance probability of the RJ-sampler can be expressed using (8) as:

$$\alpha_{f,RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \propto \frac{p(\tilde{\mathbf{s}}_k | \tilde{\mathbf{m}}_k) p(\mathbf{z}_{1:k} | \tilde{\mathbf{s}}_k, \tilde{\mathbf{m}}_k) p(\tilde{\mathbf{m}}_k) q(\mathbf{m}_k | \tilde{\mathbf{m}}_k) q(\mathbf{s}_k | \tilde{\mathbf{s}}_k, \mathbf{m}_k)}{p(\mathbf{s}_k | \mathbf{m}_k) p(\mathbf{z}_{1:k} | \mathbf{s}_k, \mathbf{m}_k) p(\mathbf{m}_k) q(\tilde{\mathbf{m}}_k | \mathbf{m}_k) q(\tilde{\mathbf{s}}_k | \mathbf{s}_k, \tilde{\mathbf{m}}_k)} \quad (16)$$

and due to the particle representation can be developed further:

$$\alpha_{f,RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \propto \frac{p(\mathbf{z}_{1:k} | \tilde{\mathbf{s}}_k, \tilde{\mathbf{m}}_k) p(\tilde{\mathbf{m}}_k) q(\mathbf{m}_k | \tilde{\mathbf{m}}_k)}{p(\mathbf{z}_{1:k} | \mathbf{s}_k, \mathbf{m}_k) p(\mathbf{m}_k) q(\tilde{\mathbf{m}}_k | \mathbf{m}_k)}, \quad (17)$$

and then simplified to:

$$\alpha_{f,RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \propto \frac{l(\tilde{\mathbf{s}}_k, \tilde{\mathbf{m}}_k) p(\tilde{\mathbf{m}}_k) q(\mathbf{m}_k | \tilde{\mathbf{m}}_k)}{l(\mathbf{s}_k, \mathbf{m}_k) p(\mathbf{m}_k) q(\tilde{\mathbf{m}}_k | \mathbf{m}_k)}. \quad (18)$$

From inspecting (18) we can observe that the ratio term of the acceptance probability is proportional to the likelihood, prior, and mode transition proposal respectively of the newly proposed state vector/mode on the numerator and the seed ones in the denominator. When observing (18), the symmetry of the ratio stands out, as in fact the essential requirement for the RJ sampling is indeed the reversibility of any state and this is achieved by maintaining symmetries that allow forward and back mode transitions. This essential property is explicitly present in (18) and has to be fulfilled also in proposal step(s), which is a degree of freedom in any RJ-sampler based algorithm design. So while the likelihood terms in (18) are straight forward once a likelihood formula is assumed, for example (29), a closer look is required for the product of the prior and the proposal because it is dependent on the specific choice in the algorithm design and it should respect the reversibility principle.

First of all it should be noted that the RJ-sampling scheme is, as the name suggests, a mechanism to generate samples in an iterative way provided a given initial state. In this context the RJ-sampling concept is embedded within the MCMC chains. The proposal mechanism is now described. Every chain's initial state is randomly picked from the particle cloud at time $k-1$ and the proposal is also randomly picked from the same cloud at $k-1$, a mode is proposed for each of these state vectors given their mode at $k-1$ and the mode transition probabilities, a mode-dependent prediction is applied to each state to propagate the state from time $k-1$ to k . Consequently (18) can be expressed as:

$$\alpha_{f,RJ}(\mathbf{s}_k, \tilde{\mathbf{s}}_k) \propto \frac{l(\tilde{\mathbf{s}}_k, \tilde{\mathbf{m}}_k) p(\tilde{\mathbf{m}}_k) P(\tilde{\mathbf{m}}_{k-1} | \tilde{\mathbf{m}}_k) P(\tilde{\mathbf{m}}_{k-1} | \mathbf{m}_{k-1}) P(\mathbf{m}_k | \mathbf{m}_{k-1})}{l(\mathbf{s}_k, \mathbf{m}_k) p(\mathbf{m}_k) P(\mathbf{m}_{k-1} | \mathbf{m}_k) P(\mathbf{m}_{k-1} | \tilde{\mathbf{m}}_{k-1}) P(\tilde{\mathbf{m}}_k | \tilde{\mathbf{m}}_{k-1})}. \quad (19)$$

It should be noted that even though the mode transition follows a Markov model, and as such memoryless by construction, the starting points (at time $k-1$) in the particle cloud (before the possible mode transition) of the proposal and the seed

respectively are in principle different because they are two distinct random picks from the cloud at the previous time step. Consequently to describe the proposal of the mode mechanism the mode evolution at time k is considered originated at time $k-1$. The mode probability is then computed per mode $\beta \in \mathcal{M}$ at each time step k as:

$$p(\mathbf{m}_k = \beta) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta_\beta(\mathbf{m}_k^{(i)}), \quad (20)$$

so that $\sum_{\mathbf{m}_k=1}^M p(\mathbf{m}_k) = 1$. The steps of the MMS-IP-RJMCMC are given in the procedure below, for a given total number of particles N_p , N particles per chain, and $N_{MCMC} = N_p/N$ number of parallel chains, a burn-in of B .

Algorithm B: MMS-IP-RJMCMC

Input: $\{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p}$, N_p particles at time $k-1$ and measurement $\mathbf{z}_k$ at time k .

Output: $\{\mathbf{s}_k^{(i)}\}_{i=1}^{N_p}$, N_p updated particles at time k .

1. Initialization or "seed generation" step

Randomly pick N_{MCMC} particles $\{\mathbf{s}_{k-1}^{(v)}\}_{v=1}^{N_{MCMC}} \subset \{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p}$.

Propagate the mode according to $\Pi(t_k)$. Apply (2) to

$\{\mathbf{s}_{k-1}^{(v)}\}_{v=1}^{N_{MCMC}}$, $l_k^{(v)} = p(\mathbf{z}_k | \mathbf{s}_{k-1}^{(v)})$ and obtain $\{\mathbf{s}_k^{(v)}, l_k^{(v)}\}_{v=1}^{N_{MCMC}}$.

for $v=1:N_{MCMC}$

2. RJ-based sampling step

Initialize $\mathbf{s}_k^{(v, n_{samp})} = \mathbf{s}_k^{(v)}$, with $n_{samp}=1$.

for $n_{samp}=1:N+B$

Randomly pick a particle $\mathbf{s}_{k-1}^{(i)} \subset \{\mathbf{s}_{k-1}^{(i)}\}_{i=1}^{N_p}$. Propagate the mode according to $\Pi(t_k)$. Apply (2) and obtain $\mathbf{s}_k^{(i)}, l_k^{(i)}$.

Draw $u \sim \mathcal{U}(0,1)$, update using the acceptance probability

$$\mathbf{s}_k^{(v, n_{samp})} = \begin{cases} \mathbf{s}_k^{(i)} & \text{if } u \leq \alpha_{RJ}(\mathbf{s}_k^{(v, n_{samp})}, \mathbf{s}_k^{(i)}) \\ \mathbf{s}_k^{(v, n_{samp})} & \text{otherwise} \end{cases}.$$

end

Thus obtain $\{\mathbf{s}_k^{(v, n_{samp})}\}_{n_{samp}=1}^N$

end

Compute r_N , extend B if $r_N > 1.1$. Thus obtain

$$\{\mathbf{s}_k^{(i)}\}_{i=1}^{N_p} = \{\mathbf{s}_k^{(v, n_{samp})}\}_{v=1, n_{samp}=1}^{N_{MCMC}, N}.$$

3. Compute the posterior mode probability $p(\mathbf{m}_k)$ as in (20).

VI. SIMULATIONS

In this section the algorithms developed in section V are applied and some simulation results provided. For convenience, the models used for the simulations in the system setup are provided below, [15].

A. System setup

In the examples in this paper, the modes assumed are $m_k \in \{1, 2\}$, where $m_k = 1$ is the constant velocity model, $m_k = 2$ is the constant turn rate model. Assuming then that the update interval is constant, i.e. $t_k - t_{k-1} = T$, the state evolution functions can be expressed for each mode. Let the state vector for modes 1 and 2 be $\mathbf{s}_{k,1}$ and $\mathbf{s}_{k,2}$ respectively. The evolution model is given by:

$$f(t_k, \mathbf{s}_{k,m_k}, \mathbf{w}_{k,m_k}) = \mathbf{F}(\mathbf{s}_{k,m_k}) \mathbf{s}_{k,m_k} + \mathbf{G}(\mathbf{s}_{k,m_k}) \mathbf{w}_{k,m_k}. \quad (21)$$

Recall for the constant model:

$$\mathbf{F}(\mathbf{s}_{k,1}) = \text{diag}(\mathbf{F}_s, \mathbf{F}_s, 1), \quad (22)$$

$\mathbf{F}_s = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, $\mathbf{s}_{k,1} = [x_k, v_{x,k}, y_k, v_{y,k}, z_k]^T$, and the process noise input function for the constant velocity model is:

$$\mathbf{G}(\mathbf{s}_{k,1}) = \text{diag}(\mathbf{G}_{a_{k,1}}, \mathbf{G}_{a_{k,1}}, \frac{a_{k,1} T^2}{\sqrt{3}}), \quad (23)$$

where for $a_{k,1} = a_{cv}$, the constant velocity model acceleration

$$\mathbf{G}_{a_{k,1}} = \begin{bmatrix} \frac{a_{k,1} T^2}{\sqrt{3}} & 0 \\ \frac{a_{k,1} T}{\sqrt{3}} & 0 \end{bmatrix}. \text{ The matrix for the constant turn rate}$$

model is for $\mathbf{s}_{k,2} = [x_k, v_{x,k}, y_k, v_{y,k}, z_k, v_{z,k}, \omega_k]^T$ is:

$$\mathbf{F}(\mathbf{s}_{k,2}) = \text{diag}(\mathbf{D}_s, \mathbf{F}_s, 1), \quad (24)$$

$$\text{with } \mathbf{D}_s = \begin{bmatrix} 1 & s & 0 & -c \\ 0 & co & 0 & -si \\ 0 & c & 1 & s \\ 0 & si & 0 & co \end{bmatrix}, c = \cos(\omega_k T)/\omega_k,$$

$$si = \sin \omega_k T, co = \cos \omega_k T, s = \sin(\omega_k T)/\omega_k.$$

The process noise function and its terms are respectively:

$$\mathbf{G}(\mathbf{s}_{k,2}) = \mathbf{K}(\mathbf{s}_{k,2}) \mathbf{L}(\mathbf{a}_{k,2}), \quad (25)$$

$$\mathbf{K}(\mathbf{s}_{k,2}) = \text{diag}(\mathbf{E}_s, \mathbf{I}, \frac{1}{|v_k|}), \quad (26)$$

$$\text{with } \mathbf{I} \text{ the identity matrix, and } \mathbf{E}_s = \begin{bmatrix} ca & 0 & -sa & 0 \\ 0 & -sa & 0 & -sa \\ sa & 0 & ca & 0 \\ 0 & ca & 0 & ca \end{bmatrix},$$

$$ca = \cos(\text{atan2}(v_{y,k}/v_{x,k})), sa = \sin(\text{atan2}(v_{y,k}/v_{x,k}));$$

$$\mathbf{L}(\mathbf{a}_{k,2}) = [\text{diag}(\mathbf{g}_{a_{k,2}(1)}, \mathbf{g}_{a_{k,2}(2)}, \mathbf{g}_{a_{k,2}(3)}); 0 \frac{a_{k,2}(2)}{\sqrt{3}} 0], \quad (27)$$

$$\text{with } \mathbf{a}_{k,2} = [a_{long}, a_{lat}, a_{vert}]^T, \mathbf{g}_{a_{k,2}} = [\frac{a_{k,2}(1) T^2}{\sqrt{3}} \frac{a_{k,2}(2)}{2} \frac{a_{k,2}(3)}{\sqrt{3}} T]^T.$$

The radar measurement in (3) can be expressed as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{s}_{H,k}) + \mathbf{n}_k, \quad (28)$$

where $\mathbf{h}(\mathbf{s}_{H,k}) = [r_k, b_k, e_k, v_{d,k}]^T$, $\mathbf{v}_k = [v_{x,k}, v_{y,k}, v_{z,k}]^T$,

$$\text{where } \begin{bmatrix} r_k \\ b_k \\ e_k \\ v_{d,k} \end{bmatrix} = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \text{atan2}(y_k/x_k) \\ \text{asin}(z_k/r_k) \\ (x_k v_{x,k} + y_k v_{y,k} + z_k v_{z,k})/r_k \end{bmatrix}.$$

The likelihood function has then an exponential form:

$$l_k = \frac{1}{\sqrt{2\pi|\mathbf{R}_z|}} e^{\left\{ -\frac{1}{2} (\mathbf{z}_k - \mathbf{h}(\mathbf{s}_{H,k}))^T \mathbf{R}_z^{-1} (\mathbf{z}_k - \mathbf{h}(\mathbf{s}_{H,k})) \right\}}, \quad (29)$$

with $\mathbf{R}_z$ the covariance matrix with σ_r^2 , σ_b^2 , σ_e^2 , $\sigma_{v_d}^2$ variances for range, bearing, elevation, Doppler velocity.

B. Settings and scenario

The radar measurement settings are selected to be the following: the radar scan time $T=2s$, the radar plots have standard deviation of the measurement errors in range, bearing, elevation, and radial velocity respectively equal to $\sigma_r=6m$, $\sigma_b=6mrad$, $\sigma_e=8mrad$, $\sigma_{v_d}=0.5m/s$. These values are explicitly taken because they are particularly demanding in the multiple model context. The same applies to the trajectory selected. The radar is positioned in the Cartesian position (0,0,0). The mode transition matrix used was set to:

$$\mathbf{\Pi}(t_k) = \begin{bmatrix} 0.99 & 0.1 \\ 0.01 & 0.9 \end{bmatrix}, \text{ and}$$

the prior mode probabilities are set equal to 0.5 for both modes.

The model settings for the process noise are $a_{cv}=5m/s^2$ and $a_{long} = a_{lat} = a_{vert} = 10m/s^2$. The parameters for the two MCMC algorithms proposed in this paper are the following: total number of particles $N_p = 300, N = 15$ particles per chain, and $N_{MCMC} = N_p/N = 20$, number of parallel chains. The burn-in is fixed to $B=200$ for the MMS-IP-RJMCMC and $B=120$ per each mode for the IMM-IP-MCMC. These algorithms are also compared with the implementation of the IMM with SIR filtering [4], having total number of particles per mode $N_p = [1600, 4400]$. The scenario is composed by a single target in noise performing a straight line trajectory towards the radar, followed by a 6g turn to the right, and finishing with a straight line trajectory.

In order to get a measure for convergence of the algorithm over a number of runs, several options are available. Here we consider the criterion of ellipsoidal gating (used in association for target tracking [15]) as a measure of the distance between the track estimate and the measurement. Let the error be:

$$\mathbf{\varepsilon}_{m_k,k} = \mathbf{z}_k - \mathbf{h}(\mathbf{s}_{H,k|k-1}), \quad (30)$$

where $\mathbf{s}_{H,k|k-1}$ is the output at time k of the filter under test. The gating score is then expressed as:

$$d_{m_k,k}^2 = \boldsymbol{\varepsilon}_{m_k,k}^T \mathbf{S}_{m_k,k}^{-1} \boldsymbol{\varepsilon}_{m_k,k}, \quad (31)$$

where $\mathbf{S}_{m_k,k} = \mathbf{C}_{m_k,k} + \mathbf{R}_{\sigma^2}$ with $\mathbf{C}_{m_k,k}$ being the sample covariance matrix and $\mathbf{R}_{\sigma^2}$ is the diagonal matrix with the measurement variance. Collect the gating score per mode, and let $\mathbf{d}_k^2 = [d_{1,k}^2, d_{2,k}^2, \dots]^T$. A divergence is said to occur if $\min(\mathbf{d}_k^2) > T_G$, with T_G being the threshold.

C. Results

Given the parameters set in the previous paragraphs, here some figures showing the output of the filters are provided and discussed, also with the purpose of highlighting the differences between the two families.

In Fig.1 and 2, the outputs in Cartesian coordinates x and y only (for visualization purposes) are shown over time, as the target proceeds approaching the radar and turning away from it, for the IMM-IP-MCMC and MMS-IP-RJMCMC outputs respectively. In both figures the true position is shown with a plus, the noisy measurements with an x , and the particles in

color. The two models are indicated in different colors: the constant velocity model particles are shown with green dots and the coordinated turn model particles are shown blue circles for the output plots. The mean of particle cloud in x and y is shown with a magenta star. It is important to mention that the mean of the cloud is used in the figures for visualization purposes, however at moments of uncertainty the particle cloud can present bimodal or multimodal behavior so a MAP estimate is better suited than a mean estimate. Overall a MAP estimate should be used if an estimate should be used in further processing, however as here it is mainly for visualization only the mean is adequate. In particular, in Fig. 1, at the time step after the start of the curve, the bimodality of the particle cloud associated to the constant velocity (in green) is clearly visible, this is due to the uncertainty in the non-dominant mode.

Note that in the IMM-IP-MCMC the number of particles in each mode is fixed beforehand by the user, and it is visible in Fig. 1 that both particle clouds coexist at all time and at mode transitions the non-dominant mode presents a large spread because of uncertainty while the mode change is not yet confirmed. Analogously at startup there is also a spread in both particle clouds because of the initial uncertainty.

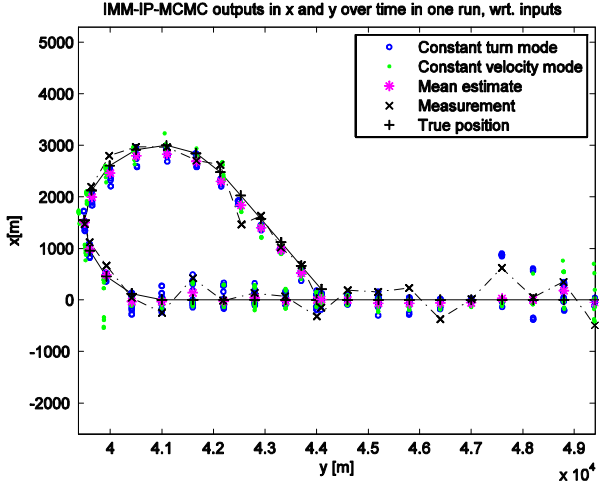


Fig. 1 IMM-IP-MCMC output in x and y , measurement and true values

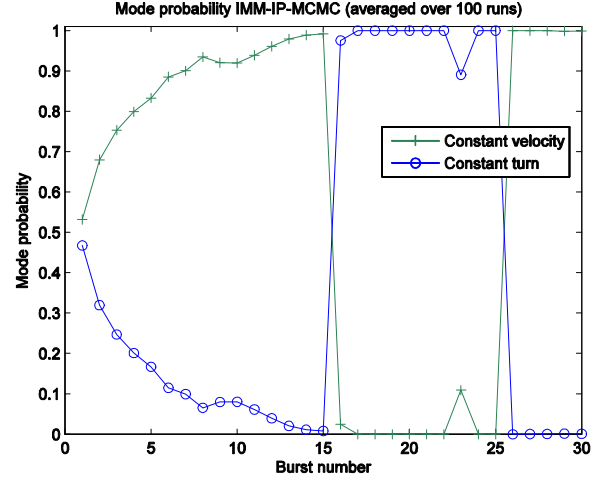


Fig. 3 IMM-IP-MCMC mode probability average over 100 runs

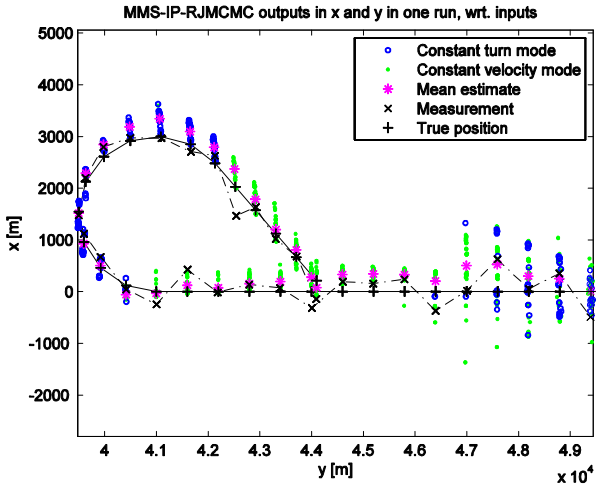


Fig.2 MMS-IP-RJMCMC output in x and y , measurement and true values

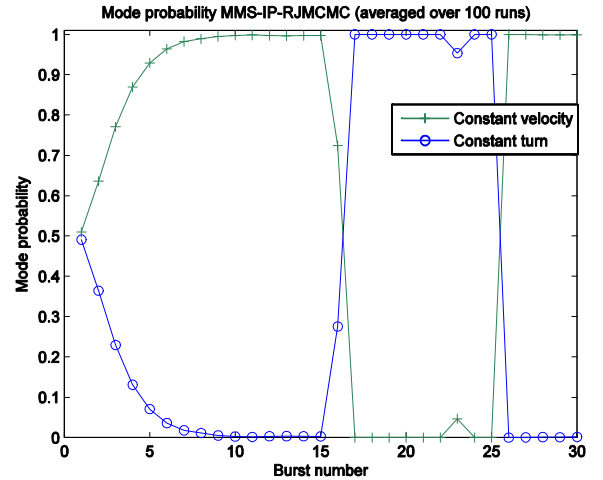


Fig.4 MMS-IP-RJMCMC mode probability, average over 100 runs

Depending on the actual realization of the measurement noise samples the uncertainty period can be shorter or longer. Since more particles are used to obtain the mode probability estimate it shows to be more sensitive and so uncertain at the start of the tracking. In Fig. 2 the output of the MMS-IP-RJMCMC is shown, and, after the run-in period at startup characterized by uncertainty, provides a more visible impression of the dominant mode at the different time steps. This is due to the fact that there is no fixed number of particles assigned to each model. It is also visible how at mode transitions, in particular at the transition between the incoming straight trajectory and the right turn, the particle cloud shows multi-modality because of the uncertainty in the mode change or in the direction of the turn.

The mode probability in the IMM-IP-MCMC is a parameter belonging to the mixing stage, common to the IMM PF with the SIR implementation [4], and was indicated as $p(m_k)$ in the previous sections. For the MMS-IP-RJMCMC, the mode probability is obtained by calculating over time the fraction of particles in each mode given the total number of particles N_p . The mode probability over time, averaged over 100 runs, is shown in Fig. 3 and 4 respectively for the IMM-IP-MCMC and MMS-IP-RJMCMC. For the probability of the model, the constant velocity model is shown in green plus marker and the coordinated turn model is shown with blue circle marker. It is possible to notice some initial uncertainty in the dominant mode at the initial time steps, followed by a bigger certainty on the dominant mode. The transition between modes is also clearly visible.

We compare both algorithms over 100 Monte Carlo runs in terms of number of divergences. The measurements present different noise realizations at each run. Among the various measures possible, the ellipsoidal gating has been selected as a measure of divergence. The threshold for the ellipsoidal gating is set to $T_G = 16.25$. The IMM-IP-MCMC presents a divergence of 32% and the IMM SIR-based PF [4] of 60%, and the MMS-IP-RJMCMC of 26%. These percentages correspond to a stringent scenario; if the standard deviation in Doppler velocity is increased these percentages reduce largely to provide an even better performance. In theory the accuracies of the filters should be comparable because of targeting the same posterior pdf, however the main topic addressed here is robustness, not accuracies; computing the latter would require an analysis over several different trajectories and this is left for future research. It is important to consider the divergences above in relation with the number of likelihood calculations. These are equal to 6000 for the IMM SIR-based PF, 3000 for the IMM-IP-MCMC and 4600 for the MMS-IP-RJMCMC.

VII. CONCLUSIONS

In this paper two MCMC algorithms are presented to tackle the problem of multiple motion models for target tracking. These new algorithms are added to the two families of SIR based PF multiple model filters already existing in literature, [3], [4]. The main difference between these two MCMC algorithms is the fact that the mode mixing can be applied internally to the filtering scheme or a separate phase in a two stage recursions.

Simulation results are provided for both families of algorithms and the differences highlighted. Both algorithms can cope with the demanding scenario. They correctly identify the mode that the target is moving with, after an initial uncertainty during setting in at startup. At the abrupt motion transitions the mode change is correctly selected even after the target has been in the same mode for a longer interval of time.

The main differences in the implementations are also visible by the simulation results. In the IMM-IP-MCMC both particle clouds are present at all times, so the standard deviation of the particle cloud in the non-dominant mode can be large. For the MMS-IP-RJMCMC the particle cloud over time shows clearly a dominant mode. Multi-modality becomes evident before a decision is taken on a left or right constant turn.

The two algorithms present similar performance when evaluating over a number of runs the correctness of the tracking output, outperforming the IMM SIR-based PF (already proved in [4] to outperform the Bootstrap version [3]) in this demanding scenario.

REFERENCES

- [1] H. A. P. Blom, "An efficient filter for abruptly changing systems", Proc. of the 23rd IEEE Conf. on Decision and Control, Dec. 1984.
- [2] E. Mazar, A. Averbuch, Y. Bar-Shalom, J. Dayan, "Interacting Multiple Model methods in target tracking: a survey", IEEE Trans. Aerospace and Electronic Systems, 34(1), 1998.
- [3] McGinnity and G. W. Irwin, "Multiple model Bootstrap filter for maneuvering target tracking", IEEE Trans. on Aerospace and Electronic Systems, 36(3):1106-1112, 2000.
- [4] H. Driessen and Y. Boers, "An efficient particle filter for Jump Markov Nonlinear systems", IEE Proc. for Radar Sonar Navig., 152(5):323-326, 2005.
- [5] A. Doucet, N. de Freitas, and N. Gordon. Sequential Monte Carlo Methods in Practice. Springer-Verlag, Berlin, 2001.
- [6] S. Arulampalam, N. Gordon, and R. Ristic. Beyond the Kalman Filter - Particle Filters for Tracking Applications. Boston - London: Artech House, 2004..
- [7] N. Gordon, S. Maskell, T. Kirubarajan, and M. Mallick, "Littoral Tracking using Particle Filter", Proc. of the Conf. on Information Fusion, 2:935-942, July 2002.
- [8] Y. Boers and J.N. Driessen, "Interacting multiple model particle filter", IEE Proc. for Radar Sonar Navig, 150(5):344-349, 2003.
- [9] Jiantao Wang, Bo Fan, Yanpeng Li, and Zhaowen Zhuang, "A novel interacting multiple model particle filtering for maneuvering target tracking in clutter", Progress In Electromagnetics Research C, Vol. 35, 177-191, 2013.
- [10] M. Bocquel, H. Driessen, and A. Bagchi, "Multitarget tracking with Interacting Population-based MCMC-PF", Proc. of the 15th Int. Conf. on Information Fusion, 2012.
- [11] D. P. Kroese, T. Taimre, Z. I. Botev, Handbook of Monte Carlo Methods. John Wiley & Sons, 2011.
- [12] P. J. Green. "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination". Biometrika, 82(4):711-732, December 1995.
- [13] A. Gelman and D. Rubin, "Inference from iterative simulation using multiple sequences (with discussion)". Statistical Science, 7(2):457-511, 1992.
- [14] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," Biometrika, 57:97-109, 1970.
- [15] S. Blackman and R. Popoli. Design and Analysis of Modern Tracking Systems. New York: Artech House, 1999.