Advanced IP-MCMC-PF Design Ingredients

Fernando J. Iglesias García, Mélanie Bocquel, Hans Driessen
Thales Nederland B.V. - Sensors Development System Engineering,
Hengelo, Nederland
Email: {Fernandojose.Iglesiasgarcia, Melanie.Bocquel, Hans.Driessen} @nl.thalesgroup.com

***Abstract*—This paper proposes techniques to improve the properties of Sequential Markov Chain Monte Carlo (SMCMC) methods in the context of multi-target tracking. In particular, we extend the Interacting Population-based MCMC Particle Filter (IP-MCMC-PF) with three different methods: delayed rejection, genetic algorithms, and simulated annealing. Each of these methods furnishes the IP-MCMC-PF algorithm with different theoretical guarantees which are empirically analysed in this paper.**

Firstly, the use of delayed rejection in the Metropolis-Hastings (MH) samplers is proposed in order to reduce the asymptotic variance of the estimate. Secondly, the crossover operator, inspired by genetic algorithms, is presented as a mechanism to increase the interaction of the MH samplers. Thus, attaining fast convergence of the time-consuming MCMC step. Thirdly, simulated annealing is introduced with the goal of increasing the robustness of the algorithm against divergence due to e.g. poor initialisations.

Finally, the results from our experiments show that the proposed methods strengthen the multi-target tracker in the aforementioned aspects.

***Index Terms*—multi-target tracking, Bayes filter, particle filtering, Markov chain Monte Carlo, sequential Monte Carlo, Metropolis-Hastings, IP-MCMC-PF, delayed rejection, genetic algorithms, simulated annealing.**

I. Introduction

Multi-target tracking is a well-known problem which involves the joint detection of an unknown and time-varying number of targets as well as the estimation of their individual states from sensor data. This problem poses significant technical challenges and has found application in various disciplines, including econometrics, image and signal processing, and biomedical engineering [1], [2].

The Bayesian framework provides a rigorous tool to solve dynamic state estimation problems (e.g. target tracking) under a probabilistic approach. In our application of interest, multi-target tracking based on radar measurements, the model non-linearities together with the multi-modality of the state space make the application of Bayesian sequential state estimation particularly challenging.

Classical approaches to multi-target tracking rely on a pre-processing chain prior to the tracker which comprises target detection and measurement extraction (producing the so-called plot measurements). Both detection and extraction depend on threshold-based decisions, which are made before information from the past is integrated into the processing. However, in order to perform optimally, these hard decisions should be made using all available information, that is, past as well as present measurements. In contrast to the classical approach, in Track Before Detect (TBD) the hard decisions are postponed until the end of the processing chain [3]. In addition, the measurement-to-track or data association problem, which constitutes part of the tracker in the classical approach, need not be explicitly solved under the TBD approach, since the raw measurements are given to the tracker.

The Particle Filter (PF) [4] is an implementation of the prediction and update equations of the Bayes filter suitable for nonlinear and non-Gaussian problems. Particle filtering is a type of Sequential Monte Carlo (SMC) method that represents the distributions of interest by a set of weighted samples called particles. One of the most common forms of the particle filter is based on Sampling Importance Resampling (SIR). The SIR-PF suffers from a lack of efficiency especially relevant in the case of multi-target tracking. In particular, the number of particles necessary for the successful application of the filter grows exponentially with the dimension of the target distribution. To overcome this limitation, [5] proposed a strategy to exploit the independency among targets, thus circumventing the high dimensionality of the joint multi-target probability density. Another alternative introduced in [6] is the use of Markov Chain Monte Carlo (MCMC) within the PF in multi-target tracking. This partially explains the growing interest towards Sequential MCMC (SMCMC) methods that are suited to simulate complex, non-standard, multi-variate distributions [7]. In the context of multi-target tracking, a recent method leveraging SMCMC techniques is the Interacting Population-based MCMC-PF (IP-MCMC-PF) [8]. This algorithm is able to track a fixed and known number of targets.

The literature concerning the design of MCMC samplers is vast. It includes modifications and "tricks" one can apply to MCMC algorithms in order to improve their theoretical guarantees and performance in practice. The goal of the work presented in this paper is to improve the performance and the characteristics of the IP-MCMC-PF algorithm for multi-target tracking utilising strategies from statistics and optimisation theory.

This paper is organised as follows. Section II describes formally the multi-target tracking problem under a Bayesian approach. Next, section III is occupied with the core theoretical concepts relevant to this work; starting with a brief introduction to SMCMC methods and continuing with the fundamentals of IP-MCMC-PF. Section IV presents the methods we propose to use in conjunction with IP-MCMC-PF. In

particular, we show for each of the three proposed methods (delayed rejection, genetic algorithms, and simulated annealing) new properties beneficial for IP-MCMC-PF. First, delayed rejection reduces the asymptotic variance of the estimate or, in other words, keeps the correlation among the samples of the Markov chain low. Second, the crossover move from genetic algorithms increases the exchange of information among the MH samplers, achieving rapid convergence to the steady state of the chain. Third, annealing of the likelihood function enhances tracking robustness against divergence of the particle filter. Afterwards, Section V contains the experiments and results supporting the advantages that the suggested methods bring into the whole framework. Finally, the paper finishes with some conclusions and branches of future work in Section VI.

II. Multi-target Bayes Filter

This section describes the multi-target Bayesian filtering problem in the context of TBD tracking. Suppose that at time k there are n_k targets with states $\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,n_k}$, each taking values in a state space $\mathcal{X}$. In TBD, the multi-target observation at time k is an array $\mathbf{z}_k = [\mathbf{z}_k^1, \ldots, \mathbf{z}_k^m] \in \mathbf{Z}$, where each $\mathbf{z}_k^c \in \mathbb{C}$ is the complex (I/Q) signal in the radar cell indexed by c and m denotes the number of cells. In the Random Finite Set (RFS) framework [9], the finite set used to represent the joint states of the targets is referred to as the multi-target state, while the vector of intensities measured per cell at time k is the multi-target observation[1]. Formally,

$$\mathbf{X}_k = \{\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,n_k}\} \in \mathcal{F}(\mathcal{X}), \quad (1)$$

$$\mathbf{z}_k = [\mathbf{z}_k^1, \ldots, \mathbf{z}_k^m] \in \mathbf{Z}. \quad (2)$$

Let $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ denote the set of measurements collected up to and including time k. Then, the multi-target Bayes recursion propagates the multi-target posterior density $f_k(\mathbf{X}_k|\mathbf{z}_{1:k})$ in time according to the following update and prediction equations:

$$f_{k|k-1}(\mathbf{X}_k|\mathbf{z}_{1:k-1}) = \int \Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})\ f_{k-1}(\mathbf{X}_{k-1}|\mathbf{z}_{1:k-1})\ \delta\mathbf{X}_{k-1}, \quad (3)$$

$$f_k(\mathbf{X}_k|\mathbf{z}_{1:k}) = \frac{\vartheta_k(\mathbf{z}_k|\mathbf{X}_k)\ f_{k|k-1}(\mathbf{X}_k|\mathbf{z}_{1:k-1})}{\int \vartheta_k(\mathbf{z}_k|\mathbf{X})\ f_{k|k-1}(\mathbf{X}|\mathbf{z}_{1:k-1})\ \delta\mathbf{X}}, \quad (4)$$

where the integrals above are set integrals, $\Pi_{k|k-1}(\cdot|\cdot)$ is the multi-target transition kernel from time $k-1$ to k, and $\vartheta_k(\cdot|\cdot)$ denotes the multi-target likelihood function at time k.

Given the current multi-target state $\mathbf{X}$, each $\mathbf{x} \in \mathbf{X}$ either continues to exist at the next time step with probability $p_S(\mathbf{x})$ and moves to a new state $\mathbf{x}_+$ with probability $\pi(\mathbf{x}_+|\mathbf{x})$, or dies with probability $1 - p_S(\mathbf{x})$. In addition, the set $\mathbf{X}_B \subset \mathbf{X}_+$ of newborn targets, assumed independent of the surviving targets, can be modelled with a multi-Bernoulli RFS with parameters $\{p_B(\ell), b_+^{(\ell)}(\cdot)\}_{\ell=1}^{n_{MBer}}$.

The multi-target state $\mathbf{X}_+$ at the next time step results from the superposition of surviving and newborn targets, i.e. $\mathbf{X}_+ = \mathbf{X}_S \cup \mathbf{X}_B$. Under the assumption that, conditional on $\mathbf{X}$, the transition of the target kinematic states are mutually independent and that births are independent of surviving targets, it was shown in [9] that the multi-target transition kernel is given by the convolution

$$\Pi(\mathbf{X}_+|\mathbf{X}) = \sum_{\mathbf{X}_S \subseteq \mathbf{X}_+} \Pi_S(\mathbf{X}_S|\mathbf{X})\ \Pi_B(\mathbf{X}_+ \setminus \mathbf{X}_S), \quad (5)$$

where,

$$\Pi_S(\mathbf{X}_S|\mathbf{X}) = \begin{cases} p_S(\mathbf{x})\ \pi(\mathbf{x}_+|\mathbf{x}), & \text{if } \mathbf{x}_+ \in \mathbf{X}_S; \\ 1 - p_S(\mathbf{x}), & \text{otherwise;} \end{cases}$$

and

$$\Pi_B(\mathbf{X}_B) = \begin{cases} p_B(\ell)\ b_+^{(\ell)}(\mathbf{x}_+), & \text{if } \mathbf{x}_+ \in \mathbf{X}_B; \\ 1 - p_B(\ell), & \text{otherwise.} \end{cases}$$

The TBD approach defines a model for the raw measurement in terms of a multi-target state hypothesis. In this context, the target return signals measured by the radar are assumed to fluctuate according to the Swerling return amplitude fluctuation models [10], [11]. The Swerling fluctuation models are incorporated into the likelihood function of the filter, to account for the target return fluctuations. At a certain time step k, a radar illuminates a given target $\mathbf{x}_+ \in \mathbf{X}_+$ and the target reflection appears in the field of view of the radar. The complex (I/Q) signal in each cell of the observation image, denoted by z^c, is assumed to be

$$z^c = \sum_{\mathbf{x}_+ \in \mathbf{X}_+} h^c(\mathbf{x}_+) + w^c, \quad (6)$$

where w^c is the measurement noise in cell c with known statistics. In this case, the contribution $h^c(\mathbf{x}_+)$ of a target with state $\mathbf{x}_+$ to the signal in cell c depends on the so-called sensor point spread function, the target location, and the complex target echo [3]. In addition, the noise is assumed to be independent from cell to cell, additive, white, and Gaussian identically distributed.

Using the Finite Set Statistics (FISST) notion of integration and density, an explicit expression for $\Pi(\mathbf{X}_+|\mathbf{X})$ can be derived from (5). Then, the predicted density of the multi-target state can be computed using the total probability theorem. Finally, the posterior probability density of the multi-target state can be obtained from the Bayes rule for probability densities.

Note that the discussion in this section is generic, suitable for problems where the number of targets is time-varying. Nonetheless, the rest of the paper focuses on the IP-MCMC-PF algorithm, which deals with a fixed and known number of targets.

III. Fundamentals

In this section, the relevant work on SMCMC methods and the IP-MCMC-PF algorithm are reviewed.

[1]Random finite sets become a useful tool in multi-target tracking as the state is composed of an unknown, yet finite, time-varying number of targets. Using random finite sets, it is possible to formulate the multi-target Bayes filter for sequential estimation in a concise manner.

A. Sequential Markov Chain Monte Carlo

SMCMC was first investigated in [6], [7]. These methods are distinct from the traditionally employed resample-move scheme [12] since an SMC algorithm is used to design efficient high-dimensional proposal distributions for an MCMC sampler. In other words, the inefficient importance sampling step of the standard SIR-PF implementation is replaced by a more efficient MCMC sampling step. These methods allow effective MCMC algorithms to be designed in complex scenarios where standard strategies fail [6].

Within the Bayesian estimation framework, the goal is to compute the filtering density $f_k(\mathbf{X}_+|\mathbf{z}_{1:k})$ recursively by

$$f_k(\mathbf{X}_+|\mathbf{z}_{1:k}) \propto \int \vartheta_k(\mathbf{z}_k|\mathbf{X}_+)\ \Pi_{k|k-1}(\mathbf{X}_+|\mathbf{X}) \\ f_{k-1}(\mathbf{X}|\mathbf{z}_{1:k-1})\ \delta\mathbf{X}, \tag{7}$$

where $\Pi_{k|k-1}(\cdot|\cdot)$ and $\vartheta_k(\cdot|\cdot)$ are, as introduced in the previous section, the multi-target transition density and the multi-target likelihood, respectively. Let us represent the density $f_{k-1}(\mathbf{X}|\mathbf{z}_{1:k-1})$ by a set of equally weighted particles,

$$\hat{f}_{k-1}(\mathbf{X}|\mathbf{z}_{1:k-1}) \approx \frac{1}{N_p}\sum_{i=1}^{N_p}\delta_{\mathbf{X}^{(i)}}(\mathbf{X}), \tag{8}$$

where N_p denotes the number of particles and (i) the particle index. Then, by plugging this particle approximation into (7),

$$\hat{f}_k(\mathbf{X}_+|\mathbf{z}_{1:k}) \approx \frac{1}{N_p}\vartheta_k(\mathbf{z}_k|\mathbf{X}_+)\sum_{i=1}^{N_p}\Pi(\mathbf{X}_+|\mathbf{X}^{(i)}). \tag{9}$$

In [6] a Metropolis-Hastings (MH) algorithm is designed in a sequential setting in order to approximate the filtering distribution in (7). This is achieved by using the approximate posterior in (9) as the target distribution. Let us denote the proposal distribution by $q(\cdot|\mathbf{X}_+^{n_{MH}-1})$ and a sample from this distribution by $\mathbf{X}_+^*$. Thus, the acceptance ratio $\alpha(\mathbf{X}_+^*|\mathbf{X}_+^{n_{MH}-1})$ is

$$\alpha = \min\left(1, \frac{f_k(\mathbf{X}_+^*|\mathbf{z}_{1:k})}{f_k(\mathbf{X}_+^{n_{MH}-1}|\mathbf{z}_{1:k})}\ \frac{q(\mathbf{X}_+^{n_{MH}-1}|\mathbf{X}_+^*)}{q(\mathbf{X}_+^*|\mathbf{X}_+^{n_{MH}-1})}\right). \tag{10}$$

The desired approximation $\hat{f}_k(\mathbf{X}_+|\mathbf{z}_{1:k})$ is obtained by storing every N_{thin}^{th} accepted sample after the initial burn-in iterations.

B. Interacting population based MCMC-PF

In [8], an alternative algorithm well suited to deal with multi-target tracking problems for a given cardinality was derived from the Marginalized MCMC-based Particle method [6]. The proposed algorithm, detailed in [8], denoted by IP-MCMC-PF, is based on the parallel usage of multiple population-based Modified Metropolis-Hastings (MMH) [13] samplers and incorporates an interaction procedure for producing improved proposals.

The standard MH algorithm [14], [15] does not generally work well in high-dimensional spaces because the update of the Markov chain leads to the repeated sample $\mathbf{X}^{n_{MH}} = \mathbf{X}^{n_{MH}-1}$ with high probability [13]. In order to overcome this deficiency of the MH algorithm, a variation of the MMH, appropriate for sampling from high-dimensional distributions, was proposed in [8]. The MMH algorithm differs from the standard MH algorithm in the way that the candidate state $\mathbf{X}_+^*$ is generated. Instead of sampling from a proposal density $q(\cdot|\mathbf{X}_+^{n_{MH}-1})$ in the joint multi-target space to obtain the candidate state $\mathbf{X}_+^*$, a sequence of single-target proposals $q_j(\cdot|\mathbf{x}_{j+}^{n_{MH}-1})$ is used, where j indexes a partition of the joint multi-target state. Namely, each partition $\mathbf{x}_{j+}^*$ of the candidate state is drawn separately using a proposal distribution $q_j(\cdot|\mathbf{x}_{j+}^{n_{MH}-1})$ in the single-target state space.

Finally, a validation test is performed in order to check the convergence of the MCMC step on the basis of Gelman and Rubin's diagnostic [16]. For each state parameter, this approach consists of first computing the variance of the samples from each chain (after discarding burn-in), then averaging these within-chain variances, and finally comparing the averages to the variances of all the chains mixed together. This is done via the potential scale reduction factor, denoted by $\hat{\mathbf{R}}$. The parallel chains are considered well-mixed when $\hat{\mathbf{R}} \leq 1.1$ for all variables of the state space. Once the set of chains has reached convergence, the concatenated outputs from all the MH chains give the new set of particles $\{\mathbf{X}_+^{(i)}\}_{i=1}^{N_p}$, which approximates the target distribution $f_k(\mathbf{X}_+|\mathbf{z}_{1:k})$.

IV. IP-MCMC-PF Ingredients

This section is occupied with the description of the methods we propose to boost the multi-target tracking performance of IP-MCMC-PF. Namely, these methods are delayed rejection, crossover, and simulated annealing.

A. Metropolis-Hastings algorithm with delayed rejection

In the Metropolis-Hastings algorithm, the rejection of proposed moves $\mathbf{X}_+^*$ is an intrinsic part of ensuring that the chain converges to the intended target distribution $f_k(\mathbf{X}_k|\mathbf{z}_{1:k})$. However, persistent rejection, perhaps in particular parts of the state space, may indicate that the proposal distribution is poorly calibrated. Furthermore, remaining in the same state $\mathbf{X}_+$ for some time affects the quality of the corresponding Markov chain by increasing the correlation among its states. Therefore, the estimate obtained by averaging along the chain becomes less efficient. As an alternative to careful offline tuning of the state-dependent proposals, the MH algorithm with delayed rejection (MHDR) was proposed in [17], [18]. The key idea behind the MHDR algorithm is to reduce the number of rejected candidates, and thus the correlation between states of the Markov chain. This goal can be achieved in the following way: whenever a candidate is rejected, instead of taking the current state of a Markov chain as its new state (as in standard MH), a second proposal is made, let us denote it by $\mathbf{X}_+^{**}$, with density $q_2(\cdot|\mathbf{X}_+^*, \mathbf{X}_+^{n_{MH}-1})$. Of course, the acceptance probability of the new candidate must be adjusted in order to keep the distribution invariant. As shown in [17], the acceptance probability $\alpha_2(\mathbf{X}_+^{**}|\mathbf{X}_+^{n_{MH}-1})$ corresponding

to the second proposal can be chosen as

$$\begin{aligned}\alpha_2 &= \min\left(1, \frac{f_k(\mathbf{X}_+^{**}|\mathbf{z}_{1:k})}{f_k(\mathbf{X}_+^{n_{MH}-1}|\mathbf{z}_{1:k})}\right. \\ &\times \frac{q(\mathbf{X}_+^{*}|\mathbf{X}_+^{**})}{q(\mathbf{X}_+^{*}|\mathbf{X}_+^{n_{MH}-1})} \frac{q_2(\mathbf{X}_+^{n_{MH}-1}|\mathbf{X}_+^{**},\mathbf{X}_+^{*})}{q_2(\mathbf{X}_+^{**}|\mathbf{X}_+^{*},\mathbf{X}_+^{n_{MH}-1})} \\ &\times \left.\frac{1-\alpha(\mathbf{X}_+^{*}|\mathbf{X}_+^{**})}{1-\alpha(\mathbf{X}_+^{*}|\mathbf{X}_+^{n_{MH}-1})}\right). \end{aligned} \tag{11}$$

This new proposal preserves detailed balance at the second stage. If the second stage proposal is accepted, then the chain moves there. Otherwise, it is possible to either retain the current state $\mathbf{X}_+^{n_{MH}-1}$, or continue performing delayed rejection moving on to a third stage, and so on.

An interesting feature of the delayed rejection strategy is that the proposal distribution at the second stage is allowed to depend on the rejected value at the first stage as well as on the current state of the chain. Thus, the second stage proposal can learn from previously rejected candidates (without losing the Markovian property).

Before putting an end to the discussion about delayed rejection, note that this method involves more likelihood computations, which are expensive, compared to standard MH. Still, this cost may compensate in some situations where proposals are continuously rejected (consider for instance the case of ambiguous radar data [19]).

B. Genetic algorithms

Genetic algorithms (GAs) [20] belong to another class of optimizers widely used to perform heuristic search. GAs are based upon the process of natural selection from Darwinism and its Theory Of Evolution. A GA simulates the evolution of a population (that is, a set composed of possible solutions) whose members are in competition for their own survival. Mimicking the principle of survival of the fittest, the best individuals from the population breed in order to produce better offspring. Therefore, the overall fitness of the population is promoted. In our SMCMC framework for multi-target tracking, genetic operators shall be applied to improve the particles of the filter.

There exist several distinct classes of genetic operators [21]. Among them, in this paper the focus is on *crossover* because it is a flexible genetic operator in terms of interaction between chains. The crossover operator chooses two samples of the population (these will be the parents) to generate new offspring. In our implementation of the multi-target Bayes filter, the two parents $\mathbf{X}_+^{(i_1)}$ and $\mathbf{X}_+^{(i_2)}$ are synthesised as follows: for a certain j, the partitions $\mathbf{x}_{j+}^{(i_1)}$ and $\mathbf{x}_{j+}^{(i_2)}$ are combined with a certain probability of crossover P_{co},

$$\begin{aligned}\mathbf{x}_{j+}^{(i_1)*} &= \alpha\ \mathbf{x}_{j+}^{(i_1)} + (1-\alpha)\ \mathbf{x}_{j+}^{(i_2)}, \\ \mathbf{x}_{j+}^{(i_2)*} &= \beta\ \mathbf{x}_{j+}^{(i_2)} + (1-\beta)\ \mathbf{x}_{j+}^{(i_1)}. \end{aligned} \tag{12}$$

The weights α and β of the convex combinations are given by

$$\alpha = \frac{\varphi\left(\mathbf{x}_{j+}^{(i_1)}\right)}{\varphi\left(\mathbf{x}_{j+}^{(i_1)}\right)+\varphi\left(\mathbf{x}_{j+}^{(i_2)}\right)}, \beta = \frac{\varphi\left(\mathbf{x}_{j+}^{(i_2)}\right)}{\varphi\left(\mathbf{x}_{j+}^{(i_1)}\right)+\varphi\left(\mathbf{x}_{j+}^{(i_2)}\right)}, \tag{13}$$

where $\varphi(\cdot)$ is the fitness function.

Even though the crossover operator discussed in this section uses two samples to produce offspring, this idea can be extended in a natural manner by using more than two samples to produce a convex combination more general than (12).

Finally, it deserves mention that the inclusion of crossover in MH does not entail additional computational cost in comparison with standard MH. However, the parallel design of the algorithm must account for synchronisation of the samplers to propagate proposals before their combination in crossover.

C. Simulated annealing

Simulated annealing (SA) [22] is an optimisation algorithm particularly useful in problems dealing with multi-modal search spaces. Essentially, SA is based on the behaviour followed by a crystalline solid under the thermodynamical process of annealing. Annealing is controlled by the temperature of the process. In principle, at the beginning of annealing, when the temperature is high, the structure of the solid is highly deformable, whereas it becomes gradually fixed while it is cooled down. Analogously, simulated annealing favours transitions to proposals lying on a larger space (intuitively, "further" from the current seed) when the temperature is high than when it is low. Consequently, the algorithm is prevented from being trapped in local extrema.

It is also possible to interpret simulated annealing in terms of the spread of the function being optimised. For high temperature values the function is flattened, hence the acceptance ratio in (10) tends to one for a large range of proposal-seed pairings. Let us now revisit this concept formally for a choice of the objective function relevant to our application, as it shall be discussed before long. Let the objective function g be proportional to the negative exponential of a certain function h of the state,

$$g(\mathbf{X}) \propto \exp\left(-h(\mathbf{X})\right). \tag{14}$$

SA augments the objective function with a variable T, which stands for the temperature, that can take values within a certain set, i.e. $T \in \{T_i\}_{i=1}^{L}$, where $T_1 = 1$ and $T_1 < ... < T_L$. The augmented objective function is

$$\tilde{g}(\mathbf{X}, T) = \exp\left(-\frac{h(\mathbf{X})}{T}\right). \tag{15}$$

Naturally, the decay of the exponential is slower for larger values of the temperature.

In the multi-target tracking context, simulated annealing of the point spread function within the likelihood function (see [3] for a detailed derivation of these functions) can make filtering via SMCMC more robust. Particles located in low probability regions of the state space owing to, for instance, poor cloud initialisations or a large difference between the

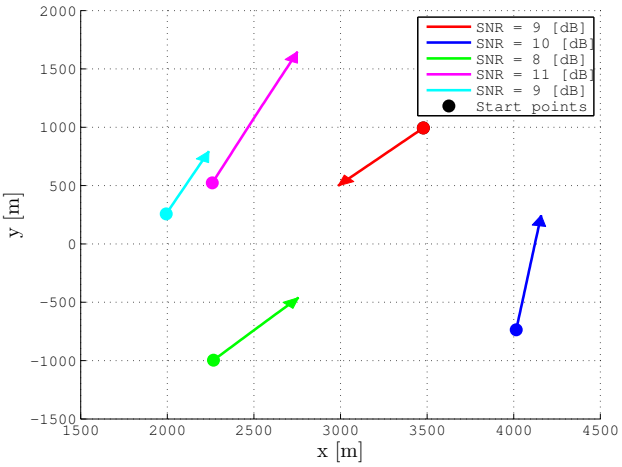


Figure 1: Experimental multi-target tracking scenario for the simulations. For each of the five targets, the dot represents the start position in the target trajectory, whereas the arrow denotes the final position and direction of the trajectory.

prior and the posterior can be brought to a better state by using simulated annealing. This effect is more significant in likelihood functions with a high peak and a quick roll-off. Note that annealing is only applied during a preliminary step of the MCMC procedure. Afterwards, the temperature is set to one so that the samples of the posterior are drawn using the true likelihood.

V. Simulation and Results

In this section, we investigate through simulations the improvements achieved with the three techniques introduced in Section IV (namely delayed rejection, crossover move, and simulated annealing) on traditional MH. In particular, the goal of the simulations is to provide empirical support of the advantages that the three proposed techniques bring to IP-MCMC-PF as a multi-target tracker.

To this end, a common scenario in which a fixed and known number of targets moves along straight lines is considered. The true tracks are shown in Figure 1. Five targets are present throughout the entire scenario lasting 50 seconds. In the experiments, we adopt a Nearly Constant Velocity (NCV) model to describe the dynamics of each target [23]. The state $\mathbf{x}_{k,j}$ of a single target j at time step k comprises the position and velocity of the target in the Cartesian plane, $s_{k,j} = (x_{k,j}, \dot{x}_{k,j}, y_{k,j}, \dot{y}_{k,j})$. The likelihood expression implemented is based on the Swerling II fluctuation model [24]. The power of a target echo in one range-bearing-Doppler cell is assumed to follow the exponential distribution. Additionally, the target echo is assumed to be independent from cell to cell. Thus, the likelihood of the measurement $\mathbf{z}_k$, conditioned on the multi-target state $\mathbf{X}_k$, is exponentially distributed. At the beginning of the simulations, the particle clouds are initialised drawing samples from a uniform distribution defined in a square window of side length equal to 100 [m] centered around the targets' true locations.

The remaining of this section discusses the experiments performed for each of the proposed techniques to enhance the design of the IP-MCMC-PF algorithm.

A. Delayed rejection

As explained in Section IV-A, introducing a second proposal in the MH algorithm increases the probability that the Markov chain moves to a new state. Therefore, the correlation among the samples of the chain per state dimension is reduced. For this experiment, the second proposal density (q_2 in (11)) consists of a random walk centered around the current state of the chain $\mathbf{X}_+^{n_{MH}-1}$,

$$\mathbf{X}_+^{**} \sim q_2(\cdot|\mathbf{X}_+^{n_{MH}-1}) = \mathcal{N}\left(\mathbf{X}_+^{n_{MH}-1}, \Sigma^{**}\right), \qquad (16)$$

where Σ^{**}, the covariance matrix of the walk, is a scaled-down version of the process noise covariance matrix in the NCV model. Hence, the random walk performs a local exploration around the seed.

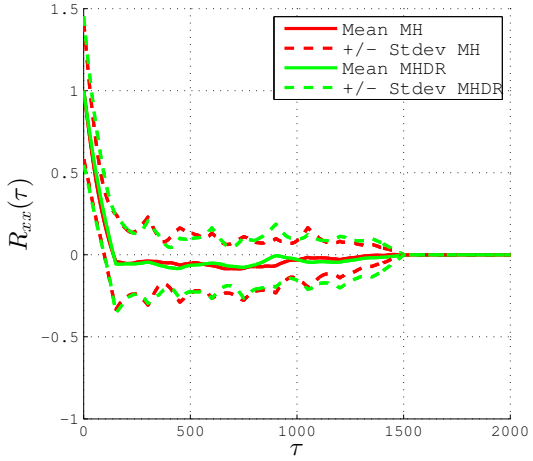
Figure 2 shows the autocorrelation function of the particle cloud at two time steps; after the first and after the last measurement have been incorporated. The x state variable of one of the targets is used to compute the autocorrelation. The figure reports the mean and the standard deviation of the autocorrelation over 50 Monte Carlo (MC) simulations. In Figure 2a, after the first SMCMC time step, it can be seen that the particles are roughly equally correlated, independently of whether Metropolis-Hastings is used with delayed rejection (MHDR), or without delayed rejection (MH). After the last SMCMC time step of the simulation, the standard deviation of the autocorrelation in Figure 2b is approximately five times larger in MH than in MHDR.

After the first SMCMC time step, the particle clouds are influenced by the uniform initialisation of the particles centered around the targets. In this case, the correlation among the samples is governed by this initialisation and it is not possible to appreciate a difference between using MHDR and MH from the autocorrelation function. On the other hand, after some SMCMC steps, the particle clouds shrink, converging to the true tracks. From this moment, the samples drawn using delayed rejection are less correlated than without delayed rejection, see Figure 2b.

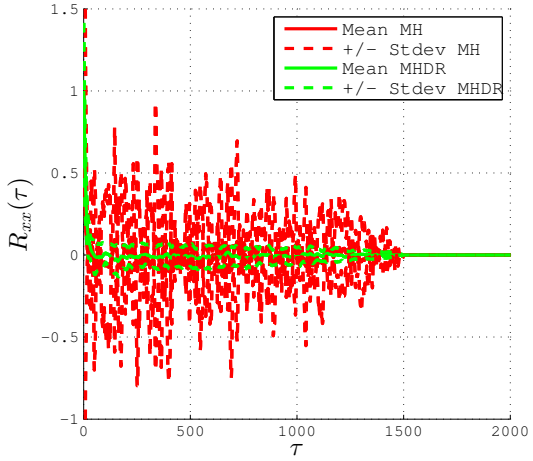
B. Genetic algorithms

In this section, we present two experiments that show the impact of using genetic algorithms (in particular, the crossover operator) on the convergence of MH. To this end, tools for convergence diagnostics of MCMC are required. In the work presented here, Gelman and Rubin's method for monitoring convergence is used, see Section III-B.

Before introducing the experiments in this section, our choice of the proposal distribution in MHGA is described. Let i_1 denote one of the MMH samplers in the IP-MCMC-PF algorithm. With certain probability of crossover P_{co}, another sampler denoted by i_2 ($i_1 \neq i_2$) and a partition j of the multi-target state are randomly chosen using integer uniform distributions. Then, a new proposal $\mathbf{x}_{j+}^{(i_1)*}$ is obtained as



(a) Autocorrelation after the first SMCMC time step. At this instant of the simulation, the autocorrelation using either of the methods is qualitatively the same.



(b) Autocorrelation after fifty SMCMC time steps. In this case, the autocorrelation of the samples drawn with MHDR is lower than with MH, which can be observed by studying the standard deviation.

Figure 2: Autocorrelation of the samples obtained after burn-in using standard Metropolis-Hastings (MH) and MH with delayed rejection (MHDR). The average and standard deviations are obtained using 50 Monte Carlo simulations.

indicated in (12). For the computation of the weight α in (13), the fitness function chosen is the likelihood function. The probability of crossover used in the experiments is $P_{co} = 0.2$.

In the first experiment, we run MH with the crossover operator (MHGA) and without crossover (MH) for the multi-target scenario presented at the beginning of Section V, and monitor the length of the burn-in period. Figure 3 shows the result of this experiment, obtained averaging the results of 100 MC simulations. As it was expected due to the interaction among the samplers, the figure shows that the chain reaches convergence in a considerably shorter burn-in for MHGA in the early time steps of SMCMC. Some steps after initialisation, once the tracking has converged, the length of the burn-in

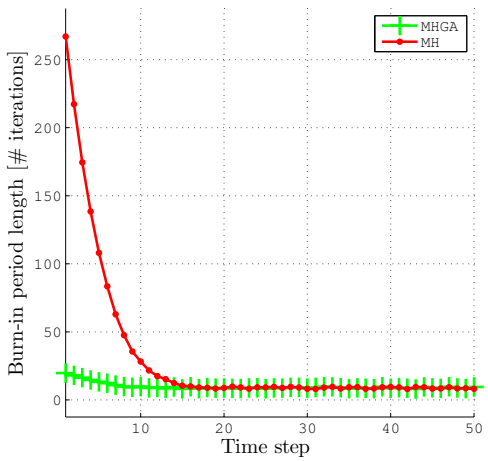


Figure 3: Length of the burn-in period (measured in number of iterations) using standard Metropolis-Hastings (MH) and MH with the crossover operator (MHGA). The results are obtained by averaging 100 Monte Carlo simulations. As it can be seen from the figure, MHGA reduces considerably the length of the burn-in period in the early steps of SMCMC.

Table I: MH and MHGA position RMSE (measured in metres) with burn-in period fixed to five.

	Target 1	Target 2	Target 3	Target 4	Target 5
MH	245	272	231	209	144
MHGA	49	60	45	38	36

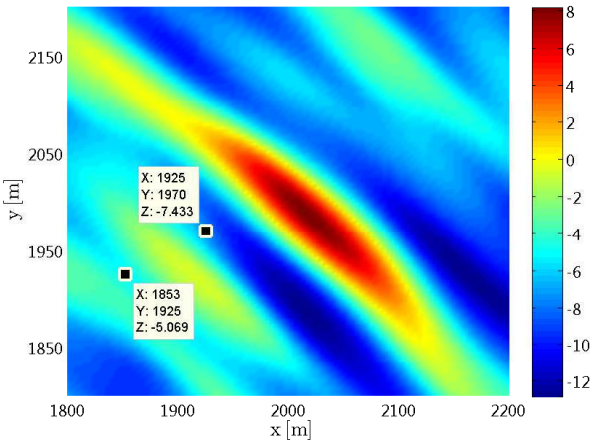
remains the same in both approaches.

Secondly, we fix the length of the burn-in to five and analyse the tracks obtained using both algorithms. The goal of this experiment is to challenge MHGA and standard MH utilising a short burn-in period, and study whether correct tracking can be achieved under this condition. Table I shows the Root-Mean-Square Error (RMSE) of the targets' positions. For each target, the RMSE shown in the table is obtained by taking the average of the RMSEs throughout the simulation. The average of 50 MC runs is used to compute the result. For MHGA, the RMSE is about five times lower than for MH. In fact, we have observed that in many of the MC runs MH is not able to perform tracking correctly in this experiment, whereas MHGA is able to track the targets despite the short burn-in period.

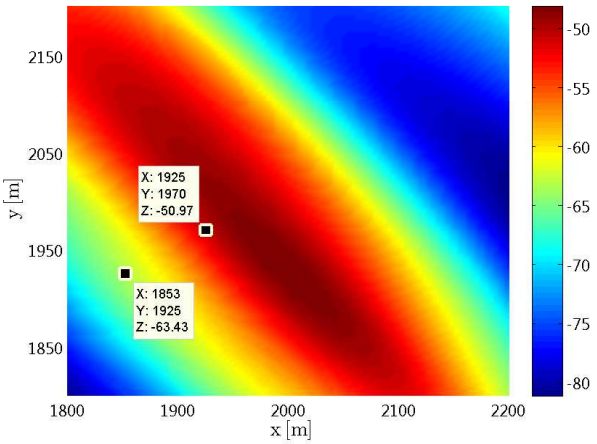
C. Simulated annealing

This section demonstrates the power of using simulated annealing in terms of robustness against faulty target detections and likelihood functions with a high peak and a quick roll-off. First of all, the concept of annealing the likelihood function is illustrated through a simple, static scenario. Afterwards, the gain in robustness against divergence of the particle filter is shown in a single-target tracking simulation.

Figure 4 shows two "likelihood maps" in a single-target scenario where the target is located at $(x_t, y_t) = (2000, 2000)$ with SNR equal to 5 [dB]. For this figure only one target is used to ease the illustration of simulation annealing. With likelihood map we refer to the result of evaluating the likelihood function in a region of the plane close to the position of the



(a) Annealing with temperature equal to one (equivalent to the true likelihood). Due to the noisy measurement, the likelihood evaluated at the point closer to the target is much smaller than the one evaluated at the further point.



(b) Annealing with temperature equal to ten. In this case, the main mode of the likelihood is more spread. Thus, the point closer to the true target's location has got a larger log-likelihood ratio.

Figure 4: Single-target likelihood function with simulated annealing. The target is at $(x_t, y_t) = (2000, 2000)$ with SNR equal to 5 [dB]. Two data points are shown, one corresponding to a position closer to the target. The Z values shown in the labels are log-likelihood ratios.

target. In Figure 4a, which corresponds to the true likelihood, there are some fluctuations which stem from peaks of noise. However, these fluctuations are no longer present in Figure 4b when annealing with $T = 10$ is applied; in this case it can be seen that the main mode of the likelihood is widespread.

Consider a particle far from the true target's location, which could be for instance caused by an erroneous cloud initialisation due to an error in the detection. This is represented in Figure 4 by the point at $(x_s, y_s) = (1853, 1925)$. Next, suppose that this particle is chosen to seed an MH sampler in the IP-MCMC-PF algorithm. During update, imagine that a certain proposal is $(x_p, y_p) = (1925, 1970)$. Intuitively, the proposal in the example should be accepted since it is better positioned than the seed with respect to the true target's location. Using the true likelihood function (i.e. without annealing, or, equivalently, with temperature $T = 1$) in Figure 4a, it is rather unlikely that the proposal would be accepted as the acceptance ratio is small, $\alpha < 0.1$. On the contrary, the same move would be accepted with probability equal to one using annealing with $T = 10$ shown in Figure 4b.

This simple example serves to illustrate the principle that simulated annealing brings more robustness into the algorithm. Notwithstanding that, annealing with a too high temperature may completely distort the measurement, losing any information regarding the target's location. In addition, note also that annealing should be used only during burn-in so that the samples drawn after burn-in used to update the particle cloud represent the true posterior density.

The remainder of this section is occupied with an experiment comparing the robustness of the tracking achieved with Metropolis-Hastings with simulated annealing (MHSA) and MH. In this experiment, the target follows a straight line trajectory, like the ones shown in Figure 1, and the SNR is equal to 5 [dB]. The particle cloud in this case is initialised 150 [m] far from the target's true location along each axis (i.e. 150 [m] in x and 150 [m] in y). This scenario simulates an error in the detector which is commonly used during initialisation in target tracking. The goal of this experiment is to analyse whether it is possible for MHSA and MH to find the right track despite the inaccurate particle cloud initialisation.

In the case of MHSA, the initial annealing temperature is $T_L = 10$ and the temperature step is $\Delta_T = 1$. For each value of the temperature, sampling with MH is performed until the steady state of the Markov chain is reached in the sense of the Gelman and Rubin's diagnostic, reviewed in Section III-B. Regarding our choice of the proposal in MHSA, it consists of a random walk similar to (16). In the case of MHSA, the random walk is centered around a randomly chosen particle from the predicted particle cloud.

After 100 MC simulations, we obtain that tracking with MH is able to find the target's true track only in 7 occasions, whereas MHSA succeeds 69 times. This is an improvement of almost 10 times in terms of robustness. In this exercise, we consider successful tracking if the estimate is within 1.5 times the size of a radar cell from the target's true location at the end of the simulation. Figure 5 shows the particle clouds throughout time obtained with MHSA and MH in a single MC simulation where only MHSA is able to find the true track.

VI. Conclusions

The work presented in this paper proposes the use of three techniques in conjunction with the IP-MCMC-PF algorithm to enhance its performance in the multi-target tracking problem. In fact, each of the mentioned techniques contributes to improve IP-MCMC-PF in a different aspect. Firstly, delayed rejection in Metropolis-Hastings (MH) reduces the correlation among the samples of the Markov chain by means of a local search, thus improving the overall quality of the estimated posterior. Secondly, the crossover operator from genetic algorithms shortens the burn-in period of the MCMC step –

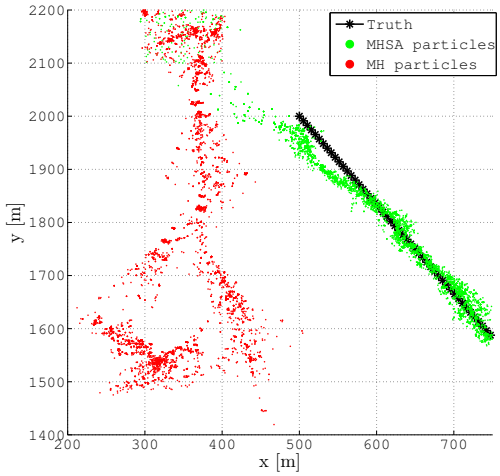


Figure 5: Particle clouds obtained with Metropolis-Hastings with simulated annealing (MHSA) and standard MH. The particle clouds are initialised somewhat far from the target's true location. MHSA is shown to be more robust than MH since MHSA is able to converge to the true track.

attaining convergence faster – by allowing the combination of samples from different MH samplers. In addition, MHGA is able to track successfully even if the burn-in period is constrained. Finally, simulated annealing improves robustness against divergence of the PF. All these claims are supported by empirical results in Section V.

Although this paper focuses on the multi-target tracking problem for a fixed number of targets, it is worth mentioning that this is not a limitation of the techniques proposed here. In fact, the IP-MCMC-PF algorithm can be extended in order to track a time-varying number of targets [25]. Naturally, delayed rejection, the crossover operator, and simulated annealing are still applicable even if the number of targets is neither known, nor fixed.

Apart from the techniques proposed in this work, there are many other well-known strategies in the statistics literature that are possible to use in conjunction with MH, as well as other MCMC algorithms. In the forthcoming work, we shall therefore focus on the study of these (and possibly other) research paths to develop more robust and efficient algorithms for multi-target tracking to manage applications with vast amounts of data.

Acknowledgement

The research leading to these results has received funding from the EU's Seventh Framework Programme under grant agreement Nº607400. The research has been carried out within the TRAX project.

Furthermore, the authors would like to thank Fotios Katsilieris at TU Delft and Dr. Martin Podt at Thales Nederland for their careful review and valuable feedback.

References

[1] A. Harvey, *Applications of the Kalman Filter in Econometrics*. Cambridge University Press, 1987.

[2] E. Meijering, I. Smal, and D. G., "Tracking in Molecular Bioimaging," *IEEE Signal Processing Magazine*, vol. 23(3), pp. 46–53, June 2006.

[3] Y. Boers and J. N. Driessen, "Multitarget Particle Filter Track-Before-Detect Application," *IEE Proc. on Radar, Sonar and Navigation*, vol. 151, pp. 351–357, 2004.

[4] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Berlin, 2001.

[5] C. Kreucher, K. Kastella, and A. O. Hero, "Multitarget tracking using the joint multitarget probability density," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, no. 4, pp. 1396–1414, 2005.

[6] Z. Khan, T. Balch, and F. Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27(11), pp. 1805–1819, 2005.

[7] F. Septier, S. Pang, A. Carmi, and S. Godsill, "On MCMC-Based Particle Methods for Bayesian filtering: Application to Multitarget Tracking," *3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 360–363, 2009.

[8] M. Bocquel, H. Driessen, and A. Bagchi, "Multitarget Tracking with Interacting Population-based MCMC-PF," *Proc. of the 15th Int. Conf. on Information Fusion*, pp. 74–81, Singapore, July 2012.

[9] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[10] P. Swerling, "Probability of Detection for Fluctuating Targets," *Information Theory, IRE Transactions on*, vol. 6(2), pp. 269–308, 1960.

[11] M. McDonald and B. Balaji, "Track-before-Detect using Swerling 0,1, and 3 Target Models for Small Manoeuvring Maritime Targets," *EURASIP Journal on Advances in Signal Processing*, vol. 2008:Article ID 326259, 2008.

[12] W. R. Gilks and C. Berzuini, "Following a moving target-Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B*, vol. 63, pp. 127–146, 2001.

[13] S. K. Au and J. L. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, vol. 16(4), pp. 263–277, 2001.

[14] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *J. Chemical Physics*, vol. 21, pp. 1087–1091, 1953.

[15] W. K. Hastings, "Monte Carlo Sampling Methods using Markov Chains and their Applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[16] A. Gelman and D. B. Rubin, "Inference from Iterative Simulation Using Multiple Sequences (with discussion)," *Statistical Science*, vol. 7, pp. 457–511, 1992.

[17] L. Tierney and A. Mira, "Some adaptive Monte Carlo methods for Bayesian inference," *Statistics in Medecine*, vol. 18, pp. 2507–2515, 1999.

[18] P. J. Green and A. Mira, "Delayed Rejection in Reversible Jump Metropolis-Hastings," *Biometrika*, vol. 88(4), pp. 1035–1053, December 2001.

[19] M. Bocquel, H. Driessen, and A. Bagchi, "Multitarget Particle Filter Addressing Ambiguous Radar Data in TBD," *Proc. of the IEEE Radar Conference*, pp. 575–580, Atlanta 2012.

[20] J. Holland, *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

[21] S. Park, J. P. Hwang, E. Kim, and H. J. Kang, "A new Evolutionary Particle Filter for the Prevention of Sample Impoverishment," *IEEE Transactions on Evolutionary Computation*, vol. 13(4), pp. 801–809, August 2009.

[22] S. Kirkpatrick, M. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[23] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. New York: Artech House, 1999.

[24] M. Skolnik, *Introduction to Radar Systems*. McGraw-Hill, Inc, Singapore, 2nd ed., 1981.

[25] M. Bocquel, "Random Finite Sets in Multi-target Tracking: Efficient Sequential MCMC implementation," Ph.D. dissertation, Faculty of Electrical Engineering, Mathematics and Computer Science. Department of Applied Mathematics. University of Twente, 2013.