The Applicability of MCMC inference to track Multiple targets within unthresholded Measurements

Meysam R. Danaee
Assistant Professor
Department of Electrical Engineering
Imam Hossein Comprehensive University (IHCU)
Tehran, Iran
Email: mrdanaee@ihu.ac.ir

Abstract—**This paper presents a Markov Chain Monte Carlo (MCMC) method for multitarget tracking within raw measurements. We derive the optimal proposal density so that the raw and unthresholded measurements could be used to generate approximating particles appropriately. However, because the optimal proposal density has exponential complexity, we apply the Gibbs sampler, the well-known MCMC method, to sample from the optimal proposal density and relieve sampling burden. Simulation results show that our strategy for using the Gibbs sampler could reach to a good compromise between accuracy and computation expense.**

Keywords-Optimal proposal; Gibbs particle filter; raw measurements.

I. Introduction

Particle filtering, a promising branch of numerical nonlinear inference named Monte Carlo filtering , has attracted attention of researchers in the multitarget tracking context recently [1-5]. It has been used as a unified procedure to render both the filtering and data association implicitly [6,7], where each particle is a representative of both target states and data association as well. Particle filter also makes it possible to cope with non-thresholded measurements. As for particle filtering point of view, one of the most important issues is the selection of suitable proposal density.

First, we derive the optimal proposal density for tracking multiple targets within raw measurements. However, the problem is that to draw sample from the optimal proposal density one needs to enumerate all possible hypotheses of the target cell occupations. Therefore, the computational burden grows at an exponential rate with the number of targets. That makes it to be included in the NP-hard problem category and therefore there would be no way to devise a method for reducing the complexity to the polynomial orders except by some suboptimal solutions.

This paper uses two methods, namely neighborhood cells gating and Gibbs sampling to relieve the difficulty of sampling from the proposal density. This makes a compromise, required to reach, between accuracy of sampling and the burn in time selection.

Remaining parts of this paper are organized as follows. Motion dynamics and sensor model are introduced in Section II. In Section III, optimal proposal density is derived for the model described in Section II and also two aforementioned strategies i.e. neighborhood cells gating and Gibbs sampler, are introduced to relieve the computation burden. Simulations proving efficiency of the Gibbs sampler are shown in section IV. Finally, section V contains a brief conclusion.

II. Motion Dynamics and Sensor Modeling

In a multitarget environment, consider presence of τ_t targets at time t. It is aimed to estimate posterior density of the joint state based on noisy observation. The joint state $\mathcal{S}^t$ is built by concatenation of all individual states together $\mathcal{S}^t = (\mathbf{s}_1^t, \mathbf{s}_2^t, \ldots, \mathbf{s}_{\tau_t}^t)$. The position components of the target construct the vector while its velocity components in x and y directions construct the vector $\mathbf{v}^t$ so that $\boldsymbol{\rho}^t$ and $\mathbf{v}^t$ create the space of the target state $\mathbf{s}^t$, as $\mathbf{s}^t = [x^t, v_x^t, y^t, v_y^t]'$. For constant velocity model the evolution process of the target motion dynamics has the Gaussian density [8]

$$\mathbf{s}^t \mid \mathbf{s}^{t-1} \sim \mathcal{N}(\cdot;\mathbf{F}\mathbf{s}^{t-1},\mathbf{Q}) \tag{1}$$

where $\mathcal{N}(\boldsymbol{\alpha};\mathbf{m},\boldsymbol{\Theta})$ is the Gaussian probability density function which is evaluated at $\boldsymbol{\alpha}$ with mean $\mathbf{m}$ and covariance matrix $\boldsymbol{\Theta}$, and

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \quad \mathbf{Q} = \rho \cdot \mathbf{I}_2 \otimes \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix} \tag{2}$$

where $\mathbf{I}_n$ indicates the $n \times n$ identity matrix, $\otimes$ denotes the Kronecker product, T stands for the length of the time step and ρ denotes the normalized variance of the velocity perturbation.

The observation area is divided into $\mathcal{L}_y \times \mathcal{L}_x$ cells where for each cell measurement is taken independently. Therefore, for the whole raw measurement $\mathcal{O}^t$, the total likelihood ratio could be written as the product of all cell likelihood function

$$\mathrm{p}(\mathcal{O}^t \mid \mathcal{S}^t, \tau_t) = \prod_{i=1}^{\mathcal{L}_y} \prod_{j=1}^{\mathcal{L}_x} \mathrm{p}(o_{i,j}^t \mid \mathcal{S}^t, \tau_t) \tag{3}$$

where $\mathcal{O}^t$ contains cell measurements $o_{i,j}^t, i = 1, \ldots, \mathcal{L}_y, j = 1, \ldots, \mathcal{L}_x$.

The multitarget reception intensity pertaining to the cell with the two-dimensional indices $\{i, j\}$ is added to the radio frequency thermal noise of the sensor and the result is then converted down to baseband. As a result, $o_{i,j}^{t}$ is the envelope of that addition and has a Rician distribution as

$$\mathrm{p}(o_{i,j}^{t} \mid \boldsymbol{\mathcal{S}}^{t}, \tau_{t}) = \frac{o_{i,j}^{t}}{\sigma_{v}^{2}} \exp\left(-\frac{\left[o_{i,j}^{t}\right]^{2} + [T^{i,j}(\boldsymbol{\mathcal{S}}^{t})]^{2}}{2\sigma_{v}^{2}} \right) \times I_{0}\left(\frac{o_{i,j}^{t}\, T^{i,j}(\boldsymbol{\mathcal{S}}^{t})}{\sigma_{v}^{2}} \right) \tag{4}$$

where σ_{v}^{2} is the measurement noise variance and I_0 is the modified Bessel function of the first kind and order zero.

Assuming the sensor response within each cell is uniform and vanishes outside [6], $T^{i,j}(\boldsymbol{\mathcal{S}}^{t}) = \mathrm{n} \cdot \mathrm{A}$ where A is a known return amplitude of a target assumed fixed and n is the number of targets inserted in the $\{i, j\}$ cell. However, a method to deal with unknown and varying number of targets with in unthresholded measurements can be found in [9]. Let ς be the index of cell having the row and column subscripts i and j respectively then $\varsigma = (j-1)\cdot\mathcal{L}_{y} + i, 1 \le j \le \mathcal{L}_{x}, 1 \le i \le \mathcal{L}_{y}$.

III. Optimal proposal and Gibbs particle filter

The objective of the multitarget tracking algorithm is to estimate the posterior density of the joint state which is computed recursively by updating the equation

$$\mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \mathcal{O}^{1:t}) = \frac{\mathrm{p}(\mathcal{O}^{t} \mid \boldsymbol{\mathcal{S}}^{t})}{\mathrm{p}(\mathcal{O}^{t} \mid \mathcal{O}^{1:t-1})} \int_{\boldsymbol{\mathcal{S}}^{t-1}} \mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{t-1}) \mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \mathcal{O}^{1:t-1})\, d\boldsymbol{\mathcal{S}}^{t-1} \tag{5}$$

Unless in linear-Gaussian process conditions, the recursion in (5) cannot be computed analytically and approximating methods should be utilized such as Monte Carlo filtering.

Particle filter [3], one of the most popular classes of the Monte Carlo methods, approximates the posterior density at each time by means of a set of particles where each particle possesses a sample $\boldsymbol{\mathcal{S}}^{i,t}$, drawn from the support area of the posterior, and a weight $w^{i,t}$ to approximate the posterior density as

$$\mathrm{p}\left(\boldsymbol{\mathcal{S}}^{t} \mid \mathcal{O}^{1:t}\right) = \sum_{i=1}^{\mathcal{N}_{p}} w^{i,t} \delta\left(\boldsymbol{\mathcal{S}}^{t} - \boldsymbol{\mathcal{S}}^{t,i}\right) \tag{6}$$

where δ represents the usual Dirac delta function.

Each sample in (6) is drawn from an easy to sample function named the proposal density $\mathrm{q}(\boldsymbol{\mathcal{S}}^{i,t} \mid \boldsymbol{\mathcal{S}}^{i,1:t-1}, \mathcal{O}^{t})$. [10] suggests using an auxiliary variable to help simulate from particles associated with large predictive likelihoods. That way, we hope that the particle weights would have less variance than the standard SIR method. This leads to higher survival rate and better usage of particle resources. The method is named Auxiliary Variable Particle Filter (AV PF) and described here.

Using approximation of (6), $\mathrm{p}(\boldsymbol{\mathcal{S}}^{t+1} \mid \mathcal{O}^{1:t+1})$ is represented by $\mathcal{M}$ mixture density components as

$$\mathrm{p}(\boldsymbol{\mathcal{S}}^{t+1} \mid \mathcal{O}^{1:t+1}) \propto \mathrm{p}(\mathcal{O}^{t+1} \mid \boldsymbol{\mathcal{S}}^{t+1}) \sum_{i=1}^{\mathcal{M}} w^{i,t} \mathrm{p}(\boldsymbol{\mathcal{S}}^{t+1} \mid \boldsymbol{\mathcal{S}}^{i,t}) \tag{7}$$

In the AV particle filter method, an auxiliary variable i is first simulated by some probability $\lambda(i \mid \mathcal{O}^{t})$, called the first stage weights, to select the most probable mixture component in (7).

Based upon the selected component density, a sample $\boldsymbol{\mathcal{S}}^{i,t}$ is drawn using the proposal density $\mathrm{q}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \mathcal{O}^{t})$ and then it is give the so-called second-stage weight as

$$w^{i,t} = \frac{\mathrm{p}(\mathcal{O}^{t} \mid \boldsymbol{\mathcal{S}}^{t}) \mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1})}{\lambda(i \mid \mathcal{O}^{t}) \mathrm{q}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \mathcal{O}^{t})} \tag{8}$$

The second stage weights in AV PF are used to correct the occurred discrepancies resulted by the first stage weights. Finally, after the resampling step, one would have evenly weighted particles.

As brought up by [11], the optimal proposal density, in the sense of minimizing the particle weights variance, has to take the form of

$$\mathrm{q}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \mathcal{O}^{t}) = \mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \mathcal{O}^{t}) \tag{9}$$

And as a results, $\lambda(i \mid \mathcal{O}^{t})$ would have the form of

$$\lambda(i \mid \mathcal{O}^{t}) = \frac{\mathrm{p}(\mathcal{O}^{t} \mid \boldsymbol{\mathcal{S}}^{t}) \mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1})}{\mathrm{q}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \mathcal{O}^{t})} \tag{10}$$

So by rewriting Bayesian equation as follows

$$\mathrm{p}(\mathcal{O}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}) = \frac{\mathrm{p}(\mathcal{O}^{t} \mid \boldsymbol{\mathcal{S}}^{t}) \mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1})}{\mathrm{p}(\boldsymbol{\mathcal{S}}^{t} \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \mathcal{O}^{t})} \tag{11}$$

the second stage weights in (8) would also be unity.

Regarding [10], that is a direct consequence of full adaption, so the resampling step could be done before the sampling step.

A. *Optimal proposal distribution for pixelized observation*

It is possible to derive the optimal proposal for the motion dynamic and sensor model defined in Section 2. The position vector of a sample is drawn over the observation area and other cases are of no interest to the algorithm. It means that if the observation area is large enough, (1) can be expanded as

$$\mathrm{p}(\mathbf{s}_{\kappa}^{t} \mid \mathbf{s}_{\kappa}^{t-1}) = \sum_{\varsigma_{\kappa}=1}^{\mathcal{L}_{x} \times \mathcal{L}_{y}} \mathrm{p}(\boldsymbol{\rho}_{\kappa}^{t} \in \mathcal{V}\varsigma_{\kappa}) \cdot \mathrm{p}(\mathbf{s}_{\kappa}^{t} \mid \mathbf{s}_{\kappa}^{t-1}, \boldsymbol{\rho}_{\kappa}^{t} \in \mathcal{V}\varsigma_{\kappa}) \tag{12}$$

where ς_{κ} is the index of the cell occupied by the target, $\mathcal{V}\varsigma_{\kappa}$ is the position spread of the ς_{κ}th cell and $\boldsymbol{\rho}_{\kappa}^{t} \in \mathcal{V}\varsigma_{\kappa}$ means that the position vector of κth target points towards the ς_{κ}th cell area at time t.

The mixture density component in (12) is a truncated Gaussian density defined as

$$\mathrm{p}(\mathbf{s}_{\kappa}^{t} \mid \mathbf{s}_{\kappa}^{t-1}, \boldsymbol{\rho}_{\kappa}^{t} \in \mathcal{V}\varsigma_{\kappa}) = \frac{\chi_{\mathcal{V}\varsigma_{\kappa}}(\kappa^{t}) \mathcal{N}(\mathbf{s}_{\kappa}^{t}; \mathcal{F}\mathbf{s}_{\kappa}^{t-1}, \mathcal{Q})}{\Upsilon_{\kappa,\varsigma_{\kappa}}} \tag{13}$$

where $\chi_{\mathcal{V}\varsigma_{\kappa}}(\kappa^{t})$ is the function equal to unity only when the κth target is located in the ς_{κ}th cell at time t, and zero elsewhere. Furthermore,

$$\begin{aligned} \Upsilon_{\kappa,\varsigma_{\kappa}} &= \mathrm{p}(\boldsymbol{\rho}_{\kappa}^{t} \in \mathcal{V}\varsigma_{\kappa}) \\ &= \int_{\mathcal{V}\varsigma_{\kappa}} \mathcal{N}(\mathbf{s}_{\kappa}^{t}; \mathcal{F}\mathbf{s}_{\kappa}^{t-1}, \mathcal{Q})\, d\boldsymbol{\rho}_{\kappa}^{t} \end{aligned} \tag{14}$$

In the case of the joint state $\boldsymbol{\mathcal{S}}^t$, according to independent target priors and (12), the following can be written

$$\begin{aligned}\mathrm{p}(\boldsymbol{\mathcal{S}}^t \mid \boldsymbol{\mathcal{S}}^{i,t-1}) = \prod_{\kappa=1}^{\tau} \sum_{\varsigma_\kappa=1}^{\mathcal{L}_x \times \mathcal{L}_y} \mathrm{p}(\boldsymbol{\rho}_\kappa^t \in \mathcal{V}_{\varsigma_\kappa}) \\ \times \mathrm{p}(\mathbf{s}_\kappa^t \mid \mathbf{s}_\kappa^{i,t-1}, \boldsymbol{\rho}_\kappa^t \in \mathcal{V}_{\varsigma_\kappa})\end{aligned} \tag{15}$$

Note that (15) can be considered as the product of τ truncated Gaussian distributions. As we discussed earlier, regarding to (15), for a given number of targets τ and $\mathcal{L}_x \times \mathcal{L}_y$ measurement cells, there are $(\mathcal{L}_x \times \mathcal{L}_y)^\tau$ different possible combinations of targets cell occupations. For each combination, the total likelihood function (3), possesses a unique value. In other words, (3) can be written as

$$\begin{aligned}\mathrm{p}(\boldsymbol{\mathcal{O}}^t \mid \boldsymbol{\mathcal{S}}^t) = \sum_{\varsigma_1=1}^{\mathcal{L}_x \times \mathcal{L}_y} \cdots \sum_{\varsigma_\tau=1}^{\mathcal{L}_x \times \mathcal{L}_y} \prod_{\kappa=1}^{\tau} \chi_{\mathcal{V}_{\varsigma_\kappa}}(\boldsymbol{\kappa}^t) \\ \times \mathrm{p}(\boldsymbol{\mathcal{O}}^t \mid \varsigma_1, \ldots, \varsigma_\tau)\end{aligned} \tag{16}$$

Then by rewriting (11) and integrating out over $\boldsymbol{\mathcal{S}}^t$, we find the first stage weights to be equal to

$$\mathrm{p}(\boldsymbol{\mathcal{O}}^t \mid \boldsymbol{\mathcal{S}}^{i,t-1}) = \sum_{\varsigma_1=1}^{\mathcal{L}_x \times \mathcal{L}_y} \cdots \sum_{\varsigma_\tau=1}^{\mathcal{L}_x \times \mathcal{L}_y} \mathrm{p}(\boldsymbol{\mathcal{O}}^t \mid \varsigma_1, \ldots, \varsigma_\tau) \cdot \prod_{\kappa=1}^{\tau} \Upsilon_{\kappa,\varsigma_\kappa} \tag{17}$$

Moreover, using (11), according to the (16),(15), the optimal proposal sampler (OPS) density expression can be expanded in the following way

$$\begin{aligned}\mathrm{p}(\boldsymbol{\mathcal{S}}^t \mid \boldsymbol{\mathcal{S}}^{i,t-1}, \boldsymbol{\mathcal{O}}^t) \propto \sum_{\varsigma_1=1}^{\mathcal{L}_x \times \mathcal{L}_y} \cdots \sum_{\varsigma_\tau=1}^{\mathcal{L}_x \times \mathcal{L}_y} \prod_{\kappa=1}^{\tau} \chi_{\mathcal{V}_{\varsigma_\kappa}}(\boldsymbol{\kappa}^t) \\ \times \mathcal{N}(\mathbf{s}_\kappa^t; \mathcal{F}\mathbf{s}_\kappa^{i,t-1}, \mathcal{Q}) \cdot \mathrm{p}(\boldsymbol{\mathcal{O}}^t \mid \varsigma_1, \ldots, \varsigma_\tau)\end{aligned} \tag{18}$$

B. Computation Burden of the Optimal Proposal Density

Suppose that the sample size is equal to $\mathcal{N}_p$ particles, and then the optimal proposal sampler has to compute $\mathcal{N}_p$ first stage weights. For every first stage weight in (17), the $(\mathcal{L}_x \times \mathcal{L}_y)^\tau$ different likelihood functions should be evaluated. It clearly gives signs of facing with a challenging NP-hard problem. We utilize two major approaches to relieve the computation burden: Neighborhood cells Gating and Gibbs-particles.

C. Neighborhood cells Gating

Until now, it was assumed that each target, regardless of its current position, can move towards each of the $\mathcal{L}_x \times \mathcal{L}_y$ measurement cells at the next time. If we just stick in probable cells (in terms of their occupation by each target at the next time), the computational expense will be decreased. It aims to select cells that their probability of covering the κth target is more than a threshold Γ. Those cell indices comprise the set $\mathcal{G}_{\kappa,t}$ as

$$\mathcal{G}_{\kappa,t} = \left\{ j \in \left\{1, \ldots, \mathcal{L}_x \times \mathcal{L}_y\right\} : \Upsilon_{\kappa,j} > \Gamma \right\}$$

For the simulation scenario in this paper, a value of 0.02 for Γ is suitable to reach an acceptable compromise between conflicting requirements.

D. Gibbs-particles

The gating procedure dramatically reduces the computation burden of algorithm, although the order of exponential growth rate dictated by the number of targets still remains.

However, it is an interesting idea to update the joint τ targets posterior density by updating separate target posteriors one by one without any assumption on the independency among the likelihood functions of targets. Our approach uses a particular single-component Metropolis-Hastings method named as Gibbs sampler as a family of Markov Chain Monte Carlo (MCMC) to update the target posteriors sequentially. Using a MCMC approach is a promising method to deal with MTT problem [11-14].

In this paper, contrary to what has been done in [14], Gibbs sampler does not approximate a posterior distribution, and neither does it serve as a means to rejuvenate degenerate samples after an assumed resampling step. The Gibbs sampler is just utilized here to draw samples from the optimal proposal as a means to relive computation expense. For each particle $\boldsymbol{\mathcal{S}}^{i,t-1}$, Gibbs sampler uses the kernel satisfying the reversibility condition to construct an irreducible and aperiodic chain in an iterative way that the chain converges to the invariant distribution which is in fact the optimal proposal density in (18).

At each time step t, MCMC goes through $\mathcal{N}_{it}$ iterations where in each iteration η the new move (the new sample $\mathcal{Y}$) proposed by the proposal density $\mathrm{q}(\mathcal{Y} \mid \boldsymbol{\mathcal{S}}_{\eta-1}^t, \boldsymbol{\mathcal{O}}^{1:t})$ which is accepted by the acceptance probability $\alpha(\boldsymbol{\mathcal{S}}_{\eta-1}^t; \mathcal{Y})$. The point is that for efficient computation, single-component Metropolis-Hastings method updates the components of joint state $\boldsymbol{\mathcal{S}}^t$ in sequential steps instead of renovating the whole state through each iteration.

Let us consider that the state at time t is divided into h components as $\boldsymbol{\mathcal{S}}^t = \{\boldsymbol{\mathcal{S}}_{.1}^t, \boldsymbol{\mathcal{S}}_{.2}^t, \ldots, \boldsymbol{\mathcal{S}}_{.h}^t\}$ then $\boldsymbol{\mathcal{S}}_{.-\kappa}^t = \{\boldsymbol{\mathcal{S}}_{.1}^t, \ldots, \boldsymbol{\mathcal{S}}_{.\kappa-1}^t, \boldsymbol{\mathcal{S}}_{.\kappa+1}^t, \ldots, \boldsymbol{\mathcal{S}}_{.h}^t\}$ includes all of the components of $\boldsymbol{\mathcal{S}}^t$ expect $\boldsymbol{\mathcal{S}}_{.\kappa}^t$. Then it could be proved that this sequential updating of components would not alter the invariant distribution if the acceptance probability was in the form of

$$\begin{aligned}\alpha\left(\boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{S}}_{.\kappa}^t, \mathcal{Y}_{.\kappa}\right) = \min\left(1, \frac{\pi\left(\mathcal{Y}_{.\kappa} \mid \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t}\right)}{\pi\left(\boldsymbol{\mathcal{S}}_{.\kappa}^t \mid \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t}\right)}\right. \\ \left. \cdot \frac{\mathrm{q}_\kappa\left(\boldsymbol{\mathcal{S}}_{.\kappa}^t \mid \mathcal{Y}_{.\kappa}, \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t}\right)}{\mathrm{q}_\kappa\left(\mathcal{Y}_{.\kappa} \mid \boldsymbol{\mathcal{S}}_{.\kappa}^t, \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t}\right)}\right)\end{aligned} \tag{19}$$

where $\mathcal{Y}_{.\kappa}$ is the κth proposed component and $\pi(\boldsymbol{\mathcal{S}}_{.\kappa}^t \mid \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t})$ is the full conditional distribution for $\boldsymbol{\mathcal{S}}_{.\kappa}^t$ under $\pi(\cdot)$ [15] and $\mathrm{q}_\kappa(\mathcal{Y}_{.\kappa} \mid \boldsymbol{\mathcal{S}}_{.\kappa}^t, \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t})$ is the κth component proposal density. Any iteration, then, is composed of h sequential steps.

To the Gibbs sampler, the acceptance probability is unity for all components since the proposal density to update the κth component of $\boldsymbol{\mathcal{S}}^t$ is identical with its full conditional distribution as

$$\mathrm{q}_\kappa\left(\mathcal{Y}_{.\kappa} \mid \boldsymbol{\mathcal{S}}_{.\kappa}^t, \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t}\right) = \pi\left(\mathcal{Y}_{.\kappa} \mid \boldsymbol{\mathcal{S}}_{.-\kappa}^t, \boldsymbol{\mathcal{O}}^{1:t}\right) \tag{20}$$

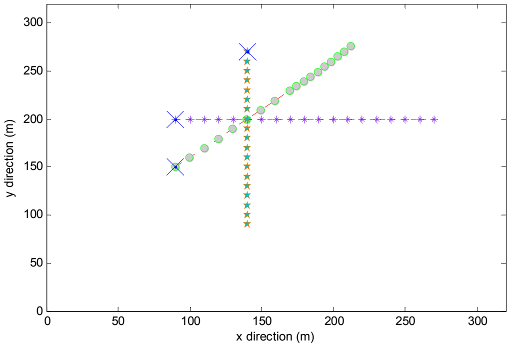

Figure 1. Simulated trajectories of three targets.

The ηth iteration of κth component at time t is to sample from the density

$$\pi\left(\mathcal{S}^t_{\eta.\kappa} \mid \mathcal{S}^t_{\eta.1},\ldots,\mathcal{S}^t_{\eta.\kappa-1},\mathcal{S}^t_{\eta-1.\kappa+1},\ldots,\mathcal{S}^t_{\eta-1.h},\mathcal{O}^{1:t}\right) \tag{21}$$

where the particular format of (21) is due to the fact that in sequential way of updating for single-component Metropolis-Hastings method, the components after the κth component, through the ηth iteration, are still at the previous states resulting from the $(\eta-1)$th iteration. From now on, we take $[\mathcal{S}^t_{\eta.1},\ldots,\mathcal{S}^t_{\eta.\kappa-1},\ \mathcal{S}^t_{\eta-1.\kappa+1},\ldots,\mathcal{S}^t_{\eta-1.h},]$ as $\mathcal{S}^t_{\eta.-\kappa}$. Each component of $\mathcal{S}^t$ is a target state which means that h is equal to τ or equivalently $\mathcal{S}^t_{\eta.\kappa} = \mathbf{s}^t_{\eta,\kappa}$. Where $\mathbf{s}^t_{\eta,\kappa}$ is the state of the κth target through the ηth iteration at time t.

The density in (21) can be written according to (18) for the ith particle sample as

$$\begin{aligned}\pi\left(\mathcal{S}^t_{\eta.\kappa} \mid \mathcal{S}^{i,t}_{\eta.-\kappa},\mathcal{O}^{1:t}\right) \propto \sum_{\varsigma_\kappa=1}^{\mathcal{L}_x\times\mathcal{L}_y} \chi_{\mathcal{V}_{\varsigma_\kappa}}(\kappa^t)\cdot\mathcal{N}(\mathbf{s}^t_\kappa;\mathcal{F}\mathbf{s}^{i,t-1}_\kappa,\mathcal{Q}) \\ \times\,\mathrm{p}(\mathcal{O}^t \mid \varsigma^{i,t}_{\eta,1},\ldots,\varsigma_\kappa,\ldots,\varsigma^{i,t}_{\eta-1,\tau})\end{aligned} \tag{22}$$

where $\varsigma^{i,t}_{\eta,\kappa}$ in (22) is the index of the cell occupied by the target $\mathbf{s}^{i,t}_{\eta,\kappa}$. After sampling from (22), the algorithm goes one step ahead and derives the full conditional distribution in the $(\kappa+1)$th updating step for $\mathcal{S}^t_{\eta.(\kappa+1)}$.

The final distribution format used by the Gibbs sampler in (22) sounds like a great idea: particle filter is composed of $\mathcal{N}_p$ samples to approximate the posterior density $\mathrm{p}(\mathcal{S}^{t+1} \mid \mathcal{O}^{1:t+1})$. To that end, each sample of the optimal proposal in (18) is sampled by the Gibbs sampler through $\mathcal{N}_{it}$ iterations where the first $\mathcal{N}_{bu}$ burns in iterations are dropped. Each iteration generates a joint sample where the $\mathcal{N}_{it}-\mathcal{N}_{bu}$ joint samples are thought to be from the optimal density in (18). As a matter of fact, the first stage weights can be computed over those $\mathcal{N}_{it}-\mathcal{N}_{bu}$ joint samples. An iteration comprises τ steps where each step has to deal with a density in the form of (22) consisting of only $\mathcal{L}_x\times\mathcal{L}_y$ combinations of target locations rather than $(\mathcal{L}_x\times\mathcal{L}_y)^\tau$ in (18). We named the method has just been described as Gibbs particle filter (GPF).

Apart from neighborhood cells gating, for each step, full conditional distribution in (22) has $\mathcal{L}_x\times\mathcal{L}_y$ mixture component densities. Besides that, there are τ steps through each iteration and each proposal density requires $\mathcal{N}_{it}$ iterations at each time and if the number of sample size is $\mathcal{N}_p$, then GPF has to evaluate $\mathcal{N}_p\cdot\mathcal{N}_{it}\cdot\tau\cdot\mathcal{L}_x\times\mathcal{L}_y$ truncated components. The value for directly sampling from the optimal proposal density in (18) is $\mathcal{N}_p\cdot(\mathcal{L}_x\times\mathcal{L}_y)^\tau$.

The point is that the GPF sounds appealing in the sense of reduction of computational burden when the number of $\mathcal{N}_p\cdot\mathcal{N}_{it}\cdot\tau\cdot\mathcal{L}_x\times\mathcal{L}_y$ is less than $\mathcal{N}_p\cdot(\mathcal{L}_x\times\mathcal{L}_y)^\tau$.

IV. PERFORMANCE ANALYSIS

A synthetic scenario is used to evaluate performance of the proposed algorithm as shown in Fig. 1. In this scenario, three different targets start out well separated, move closer and then separate again after some time. The beginning point of each target trajectory is designated by a cross. The time step for taking each measurement snapshot is T =1s.

The targets move in an area of size 320 m× 320 m for a total time of 19 seconds. The observation area is divided into $10.7m\ \times 10.7m$ cells (30 cells in each direction). σ_v^2 is always set to unity to make the comparison easy and ρ in (2) is set to half of the diameter of a measurement cell. For both methods (OPS and GPF) neighborhood cells gating approach is used with a threshold value of 0.02. Table I, shows the mean number of targets in track. Each target is in track if its trajectory is estimated not farther than a specific threshold, (here 1.5 times of the measurement cell diameter) from the true trajectory of the related target. Table II also represents the position RMS error of the targets in track.

TABLE I. MEAN NUMBER OF TARGETS IN TRACK BASED ON 25 REALIZATIONS.

Method	SNR=8 dB		SNR=10 dB		SNR=12 dB	
	$\mathcal{N}_p$ (50)	$\mathcal{N}_p$ (100)	$\mathcal{N}_p$ (50)	$\mathcal{N}_p$ (100)	$\mathcal{N}_p$ (50)	$\mathcal{N}_p$ (100)
OPS	2.6077	2.6931	2.9167	2.9242	2.9651	2.9655
GPF($\mathcal{N}_{it}=10$)	2.4472	2.4957	2.9005	2.8043	2.9743	2.9643
GPF($\mathcal{N}_{it}=100$)	2.5064	2.6317	2.8235	2.8931	2.8953	2.9596

TABLE II. RMS POSITION ERROR OF TARGETS IN TRACK BASED ON 25 REALIZATIONS.

Method	SNR=8 dB		SNR=10 dB		SNR=12 dB	
	$\mathcal{N}_p$ (50)	$\mathcal{N}_p$ (100)	$\mathcal{N}_p$ (50)	$\mathcal{N}_p$ (100)	$\mathcal{N}_p$ (50)	$\mathcal{N}_p$ (100)
OPS	6.1784	5.9791	4.9214	4.6852	4.5045	4.4867
GPF($\mathcal{N}_{it}=10$)	6.3463	6.0430	5.1652	5.4678	4.5688	4.6314
GPF($\mathcal{N}_{it}=100$)	6.5520	5.5203	5.1557	4.8395	4.8179	4.6848

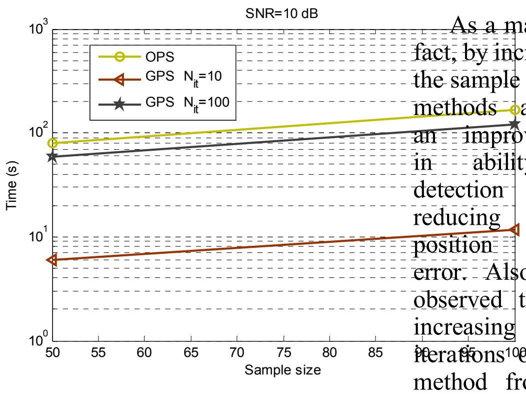

Figure 2. Average execution time with SNR 10 dB based on 25 iterations.

As a matter of fact, by increasing the sample size all methods achieve an improvement in ability of detection and reducing the position RMS error. Also it is observed that by increasing the iterations of GPF method from 10 to 100, we see its performance gets closer to the performance of OPS, though it consumes more computation time.

Fig. 2 plots the average execution time of OPS and GPF with iterations of 10 and 100 with SNR=10 dB and for two sample sizes of 50 and 100.

Although the GPF with 100 iterations produces closer performance to OPS compared to GPF with 10 iterations, in average, it runs 20 seconds (1.3 times with 50 particles) faster than OPS. However, GPF with 10 iterations is 74 seconds (13.3 times with 50 particles) quicker than OPS.

V. Conclusions

The optimal proposal sampler, OPS, is derived for multitarget tracking within raw measurements and two strategies are provided to relieve the burden of sampling, named neighborhood cell gating and GPF. Efficiency of the two methods is established through simulations showing that by increasing the iteration number of GPF, the performance gets closer to OPS but also a good compromise can be reached between accuracy of GPF and its computation time by selecting a moderate iteration number.

References

[1] Y. Boers and J. Driessen, "Multitarget particle filter track before detect application," Radar, Sonar and Navigation, IEE Proceedings, vol. 151, pp. 351 – 357, 2004.

[2] C. Hue, J. Cadre, P. Perez, "Tracking multiple objects with particle filtering," Aerospace and Electronic Systems, IEEE Transactions on vol. 38, pp. 791- 812, Jul 2002.

[3] A. Doucet, N. de Freitas, and N. Gordon, Sequential Monte Carlo Methods in Practice. New York: Springer, 2001.

[4] M. R. Morelande, C.M. Kreucher, and K. Kastella, "A Bayesian Approach to Multiple Target Detection and Tracking," Signal Processing, IEEE Transactions on, vol. 55, pp. 1589 –1604, May 2007.

[5] M. Rutten, N. Gordon, and S. Maskell, "Recursive track-before-detect with target amplitude fluctuations," Radar, Sonar and Navigation, IEE Proceedings vol. 152, pp. 345 – 352, oct. 2005.

[6] C. Kreucher, K. Kastella, and A. Hero, "Multitarget Tracking Using The Joint Multitarget Probability Density," IEEE Transactions on Aerospace And Electronic Systems, vol. 41, pp. 1396-1413, 2005.

[7] Fei Cai, Hongqi Fan, and Qiang Fu, "Track-before-detect filter for sensors with point spread function using thresholded measurements," in Information Fusion, 2014. Proceedings of the 17th International Conference on, 7-10 July 2014 , pp. 1 - 6.

[8] S. S. Blackman and R. F. Popoli, Design and Analysis of Modern Tracking Systems: Artech House Publishers, August 1999.

[9] M.R. Danaee, F. Behnia, "Enhanced strategy to sample newborn targets within nonthresholded measurements," in Electrical Engineering (ICEE), 2013 21st Iranian Conference on, 14-16 May 2013 , pp. 1- 5.

[10] M. K. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filters," J. Amer. Stat. Assoc., vol. 94, pp. 590-599, 1999.

[11] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," Statistics and Computing, vol. 10, pp. 197-208, 2000.

[12] F. Septier, K. Pang, A. Carmi, S. Godsill, "On MCMC-Based particle methods for Bayesian filtering: Application to multitarget tracking," in Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on, 2009 pp. 360-363.

[13] M. Bocquel, H. Driessen, and A. Bagchi, "Multitarget tracking with IP reversible jump MCMC-PF," in Information Fusion, 2013. Proceedings of the 16th International Conference on, 9-12 July 2013 , pp. 556 - 563.

[14] Z. Khan, Balch T., Dellaert F., "MCMC-based particle filtering for tracking a variable number of interacting targets," Pattern Analysis and Machine Intelligence, IEEE Transactions on vol. 27, pp. 1805-1819, Nov. 2005.

[15] W. R. Gilks, S. Richardson, D. J. Spiegelhalter, Markov Chain Monte Carlo in Practice: Chapman and Hall, 1996.