

An MCMC-BASED PARTICLE FILTER TBD ALGORITHM FOR MULTI-TARGET DETECTION AND TRACKING

Fan Ling *

*School of Physics and Electronic Engineering, Leshan Normal University, Leshan 614000, China

Email: lingfft@gmail.com

Keywords: track-before-detect, MCMC, particle filter.

Abstract. This paper presents an MCMC-based particle filter TBD algorithm for multi-target detection and tracking in low SNR environment. In this algorithm, a set of unweighted samples is used to represent the posterior distribution of the joint multi-target state. Then, a MH algorithm is adopted to recursively obtain the unweighted samples approximation for the posterior. Simulation results show that the performance of the proposed algorithm is very well for a fix known number of targets, even targets' paths are close proximity.

Introduction

Track-before-detect (TBD) techniques are effectively to deal with the low signal-to-noise ratio (SNR) targets through a joint processing of multi-frame of unthresholded data and integrating the targets' information. Because TBD techniques show superior detection performance compared with the conventional methods, more and more researchers pay attention to them, especially in the field of radar.

The existing studies on TBD mainly focused on the detection of single target in the radar scenario [1-3]. However, in the radar scenario, the number of targets is unknown and time-varying. Thus, the problem of multi-target more accords with the actual radar scenario. In [4], the authors have proposed a multitarget particle filter track before detect algorithm. The Sampling Importance Resampling (SIR) particle filter (PF) is used in this algorithm, i.e., 1) the importance sampling function is chosen to be the state transfer probability function, and 2) the resampling step is to be applied at every time index[5]. Due to the dimensions of the target state spaces increase with the increase of number of targets, the performance of the detection and tracking of the SIR-PF is declined with the increase of number of targets. Thus, the first problem in SIR-PF for multi-target tracking is the dimension disaster. Another problem is sample impoverishment caused by the resampling. In [6], the authors proposed a MCMC-based PF algorithm for tracking multi-target in video surveillance.

Based on the work of [6], in this paper, we present a PF TBD algorithm based on Markov Chain Monte Carlo (MCMC) sampling, in which the sampling importance resampling step of the PF is replaced by an efficient MCMC sampling step. Using a set of unweighted samples, MCMC method efficiently samples from the posterior distribution over the joint target state, which can reduce the two problems of SIR-PF discussed above.

The rest of this paper is organized in the following manner. In Section 2, we introduce the model of the multi-target moving and measurement. An MCMC-based PF TBD algorithm is presented in Section 3. In Section 4, the simulation results are given and the performance is analyzed. A summary are presented in Section 5 finally.

Moving and Measurement Model

Multi-target Moving Model

Assuming that there are E targets in the surveillance zone, and this number of targets is known and fixed. Let $\mathbf{x}_k^j = (x_k^j \ \mathbf{x}_k^j \ y_k^j \ \mathbf{y}_k^j)$ denote j target states at k frame, where, $(x_k^j \ y_k^j)$ and $(\mathbf{x}_k^j \ \mathbf{y}_k^j)$

are the position and the velocity of target j , respectively. \mathbf{x}_k^j evolves according to a discrete-time linear Gaussian target motion model,

$$\mathbf{x}_k^j | \mathbf{x}_{k-1}^j \sim N \left(\cdot; F\mathbf{x}_{k-1}^j, \mathbf{q}; \right) \quad j=1, \dots, E \quad (1)$$

where, $N(\mathbf{x}; \mathbf{m}, \Sigma)$ is Gaussian probability distribution function with \mathbf{m} mean and Σ variance; F is transition matrix, \mathbf{q} is the covariance matrix of process noise,

$$F = \begin{pmatrix} \bar{F} & 0 \\ 0 & \bar{F} \end{pmatrix}, \quad \bar{F} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \quad Q = q^j \begin{pmatrix} \bar{Q} & 0 \\ 0 & \bar{Q} \end{pmatrix}, \quad \bar{Q} = \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix} \quad (2)$$

where, q^j is the power spectral density of process noise of j target, T is the sample interval.

Thus, the joint multi-target state at k frame can denote as the collection of individual target states $\mathbf{X}_k = (\mathbf{x}_k^1, \mathbf{x}_k^2, \dots, \mathbf{x}_k^E)$. In multi-target tracking, the joint state \mathbf{X}_k is needed to be estimated.

Measurement Model

The measurements of each frame are the reflected power on azimuth-range domain. The azimuth and range domain are divided into N_a and N_r cells. Let $\mathbf{z}_k = \{z_k^{m,n}, m=1, \mathbf{K}, N_a; n=1, \mathbf{K}, N_r\}$ denote the power measurements at k frame, where each pixel $z_k^{m,n}$ can be expressed as

$$z_k^{m,n} = \sum_{j=1}^{M_k^{m,n}(\mathbf{X}_k)} P_k^j h^{m,n}(\mathbf{x}_k^j) + w_k^{m,n} \quad M_k^{m,n}(\mathbf{X}_k) \geq 1 \quad (3)$$

$$z_k^{m,n} = w_k^{m,n} \quad M_k^{m,n}(\mathbf{X}_k) = 0 \quad (4)$$

where, $w_k^{m,n}$ is an exponential distribution noise, $M_k^{m,n}(\mathbf{X}_k)$ is the number of targets at (m,n) cell, P_k^j is the power of j target at k frame, $h(\cdot)$ is the sensor's point spread function and is given by

$$h^{m,n}(\mathbf{x}_k^j) = \exp \left(-\frac{(m\Delta_r - r_k^j)^2}{d_r^2} - \frac{(n\Delta_a - a_k^j)^2}{d_a^2} \right) \quad (5)$$

$$r_k^j = \sqrt{(x_k^j)^2 + (y_k^j)^2}, \quad a_k^j = \arctan(y_k^j/x_k^j) \quad (6)$$

which are the range and an azimuth respectively of j target. Assuming that the noise variance is S_w^2 , targets are Swerling I target model. Thus, the pdf of $z_k^{m,n}$ is

$$p(z_k^{m,n} | M_k^{m,n}(\mathbf{X}_k)) = \frac{1}{S_w^2 + \sum_{j=1}^M P_k^j h^{m,n}(\mathbf{x}_k^j)} \exp \left(-\frac{z_k^{m,n}}{S_w^2 + \sum_{j=1}^M P_k^j h^{m,n}(\mathbf{x}_k^j)} \right) \quad (7)$$

if $M_k^{m,n}(\mathbf{X}_k) \geq 1$, or

$$p(z_k^{m,n} | M_k^{m,n} = 0) = \frac{1}{S_w^2} \exp \left(-\frac{z_k^{m,n}}{S_w^2} \right) \quad (8)$$

if $M_k^{m,n}(\mathbf{X}_k) = 0$.

Let $C(\mathbf{X}_k)$ denote the collection of azimuth and range cells in which E targets are located. Assuming that all the pixels of \mathbf{z}_k are independent, the likelihood function of \mathbf{z}_k is given by

$$p(\mathbf{z}_k | \mathbf{X}_k) = \prod_{(m,n) \in C(\mathbf{X}_k)} p(z_k^{m,n} | M_k^{m,n}(\mathbf{X}_k)) \prod_{(m,n) \notin C(\mathbf{X}_k)} p(z_k^{m,n} | M_k^{m,n} = 0) \quad (9)$$

MCMC-based PF TBD Multitarget Algorithm

According to the Bayesian tracking, the multi-target posterior pdf $p(\mathbf{X}_k | \mathbf{Z}^k)$ may be obtained under the commonly assumption that target motion is Markovian, recursively, in two stages:

$$\text{Prediction: } p(\mathbf{X}_k | \mathbf{Z}^{k-1}) = \int p(\mathbf{X}_k | \mathbf{X}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{Z}^{k-1}) d\mathbf{X}_{k-1}$$

$$\text{Update: } p(\mathbf{X}_k | \mathbf{Z}^k) = \frac{p(\mathbf{z}_k | \mathbf{X}_k) p(\mathbf{X}_k | \mathbf{Z}^{k-1})}{p(\mathbf{z}_k | \mathbf{Z}^{k-1})}$$

Substituting the prediction equation into the update equation, the Bayes recursive estimation of the posterior distribution is recast as

$$p(\mathbf{X}_k | \mathbf{Z}^k) = c p(\mathbf{z}_k | \mathbf{X}_k) \int p(\mathbf{X}_k | \mathbf{X}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{Z}^{k-1}) d\mathbf{X}_{k-1} \quad (10)$$

where, c is a normalization constant.

In order to using MCMC sampling instead of SIR in particle filter, firstly, we represent the posterior $p(\mathbf{X}_{k-1} | \mathbf{Z}^{k-1})$ at frame $k-1$ as a set of N unweighted samples $p(\mathbf{X}_{k-1} | \mathbf{Z}^{k-1}) \approx \{\mathbf{X}_{k-1}^i\}_{i=1}^N$. Consequently, we obtain the following Monte Carlo approximation to the Equation (11)

$$p(\mathbf{X}_k | \mathbf{Z}^k) \approx c p(\mathbf{z}_k | \mathbf{X}_k) \sum_i p(\mathbf{X}_k | \mathbf{X}_{k-1}^i) \quad (11)$$

The posterior pdf given by equation (11) is constructed by the prior $\sum_i p(\mathbf{X}_k | \mathbf{X}_{k-1}^i)$ at frame $k-1$ and the likelihood $p(\mathbf{z}_k | \mathbf{X}_k)$ at frame k . We use MCMC to sample from (11) at each time step and obtain a unweighted particle approximation for the posterior $p(\mathbf{X}_k | \mathbf{Z}^k) \approx \{\mathbf{X}_k^i\}_{i=1}^N$.

MCMC Methods

MCMC methods dynamically construct a Markov Chain, of which the stationary distribution is equal to the target distribution [7]. The Metropolis-Hastings (MH) algorithm [8] and Gibbs Sampling [9] are common MCMC methods. Here, the target distribution is the approximate posterior (11) and we use the MH algorithm to generate a set of samples from it.

Combined with the multi-target tracking for the radar system, the MH algorithm is as follows.

Assuming that the current sample of the Markov Chain at k frame are the joint multi-target state \mathbf{X}_k^n , where n is the number of samples. Then obtain all of the Markov Chain according to the following steps to iterate.

1). Propose a new candidate sample \mathbf{X}_k^* , which is a new joint multi-target state, by sampling from the proposal density $q(\mathbf{X}_k^*; \mathbf{X}_k^n)$.

2). Calculate the acceptance ratio r for \mathbf{X}_k^*

$$r = \frac{p(\mathbf{X}_k^* | \mathbf{Z}^k) q(\mathbf{X}_k^n; \mathbf{X}_k^*)}{p(\mathbf{X}_k^n | \mathbf{Z}^k) q(\mathbf{X}_k^*; \mathbf{X}_k^n)} \quad (12)$$

3). Accept \mathbf{X}_k^* with probability $a = \min\{1, r\}$, and set $\mathbf{X}_k^{n+1} = \mathbf{X}_k^*$; otherwise, keep the current sample, i.e., $\mathbf{X}_k^{n+1} = \mathbf{X}_k^n$.

Repeat $B + MN$ times for above steps, where B is the length of the burn-in period and M is the length of the thinning interval [8], we can obtain a Markov Chain $\{\mathbf{X}_k^n\}_{n=1}^{B+MN}$. Storing every M th sample after the initial B burn-in iterations, a new sample set $\{\mathbf{X}_k^i\}_{i=1}^N$ can approximate the current posterior $p(\mathbf{X}_k | \mathbf{Z}^k)$, i.e. $p(\mathbf{X}_k | \mathbf{Z}^k) \approx \{\mathbf{X}_k^i\}_{i=1}^N$.

The proposal density is chosen as the normal distribution in the MH algorithm, usually [8]. Thus, for the multi-target TBD, we use the normal distribution of the joint multi-target state as the proposal density

$$q(\mathbf{X}_k^*; \mathbf{X}_k^n) = N(\mathbf{X}_k^*; \mathbf{X}_k^n, \mathbf{S}) \quad (13)$$

where \mathbf{S} is the covariance matrix.

Substituting (13) and (11) into (12), the acceptance ratio for the multi-target TBD is recast as

$$r = \frac{p(\mathbf{z}_k | \mathbf{X}_k^*) \sum_i p(\mathbf{X}_k^* | \mathbf{X}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{X}_k^n) \sum_i p(\mathbf{X}_k^n | \mathbf{X}_{k-1}^i)} \quad (14)$$

where, $p(\mathbf{z}_k | \mathbf{X}_k)$ is the measurement likelihood and given by Equation (9). $p(\mathbf{X}_k^* | \mathbf{X}_{k-1}^i)$ is state transition probability density function and given by Equation (1). Finally, we obtain the multi-target TBD algorithm based on MCMC. The steps of the proposed algorithm are given in the next subsection.

MCMC-based PF-TBD Steps

The steps of the MCMC-based PF TBD algorithm we propose are detailed as follows.

1. Initialize the MCMC sampler: Randomly select a sample \mathbf{X}_{k-1}^i , move all targets in the selected sample according to their motion models $\mathbf{X}_{k|k-1}^i \sim p(\mathbf{X}_k / \mathbf{X}_{k-1}^i) = \prod_{j=1}^E p(\mathbf{x}_k^j / \mathbf{x}_{k-1}^{j,i})$ and use the result as the initial state of the \mathbf{X}_k Markov Chain.
2. MCMC Sampling: Repeat $B + MN$ times:
 - a. Propose a new state by sampling from the proposal density $\mathbf{X}_k^* \sim N(\mathbf{X}_k^*; \mathbf{X}_k^n, \mathbf{S})$.
 - b. Calculate the acceptance ratio (14) for this proposed state.
 - c. Generate a random number from the uniform distribution $u \sim U(0, 1)$.
 - d. If $u \leq a = \min\{1, r\}$, accept \mathbf{X}_k^* as the next state for Markov Chain, i.e., $\mathbf{X}_k^{n+1} = \mathbf{X}_k^*$; otherwise, keep the current state, i.e., $\mathbf{X}_k^{n+1} = \mathbf{X}_k^n$.
3. As an approximation for the current posterior $p(\mathbf{X}_k | \mathbf{Z}^k)$, we return the new sample set $\{\mathbf{X}_k^i\}_{i=1}^N$, obtained by storing every M th sample after the initial B burn-in iterations above.

Therefore, according to MMSE criterion, the estimation of multi-target state is given by the sample set $\{\mathbf{X}_k^i\}_{i=1}^N$

$$\hat{X}_k^{\text{MMSE}} = \int X_k p(X_k | Z^k) dX_k \approx \frac{1}{N} \sum_{i=1}^N X_k^i \quad (15)$$

Simulation Results and Performance Analysis

The simulated data consists of a total of 35 frames, with the three targets existing in the surveillance area. Target 1 is well separated from the else. Target 2 and target 3 are close and parallel. The trajectories are plotted as shown in Figure 1, which also labeled the moving direction and number of the targets.

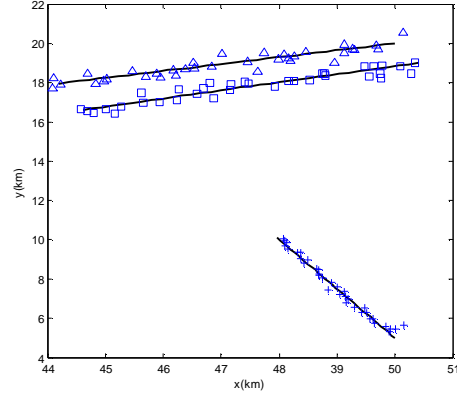
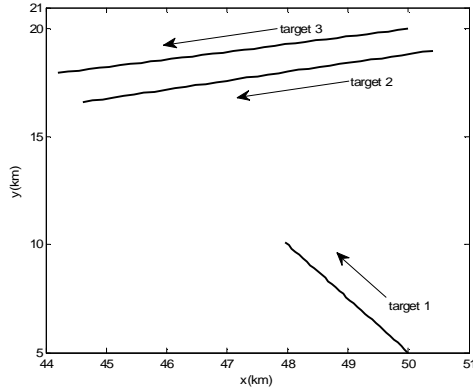


Figure 1. Illustration of trajectories of the simulation scenario Figure 2. Estimated trajectories by MCMC-TBD algorithm

The parameters of the MCMC-based PF-TBD algorithm are set as: $B = 500$, $M = 10$, and $N = 100$. Thus, the total length of the Markov Chain is 1500.

Figure 2 plots the estimated trajectories and the actual paths of the three targets with SNR=10dB by single Monte Carlo trial. It is shown that the tracking performance for target 1 separated from the else is very well. For target 2 and 3 which paths are parallel in close proximity, the tracking performance declines slightly.

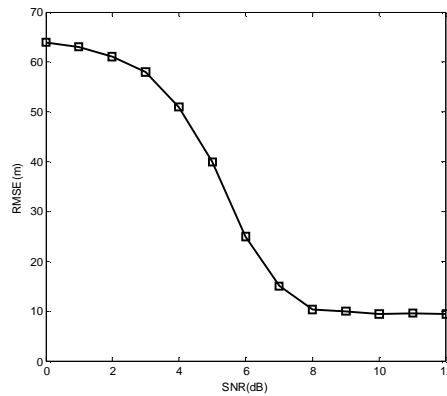


Figure 3. Performance of MCMC-based PF TBD algorithm versus SNR

Figure 3 plots the root-mean-square errors (RMSE) of the position versus SNR. RMSE is given by

$$\text{RMSE} = \sqrt{\frac{1}{IEK} \sum_{i=1}^I \sum_{k=1}^K \sum_{j=1}^E \left((x_{i,k}^j - \hat{x}_{i,k}^j)^2 + (y_{i,k}^j - \hat{y}_{i,k}^j)^2 \right)} \quad (16)$$

where, I is the number of Monte Carlo trials, K is the total number of the frame, $(x_{i,k}^j, y_{i,k}^j)$ is the actual value of the target j at i Monte Carlo trial, $(\hat{x}_{i,k}^j, \hat{y}_{i,k}^j)$ is the estimation.

The curve in Figure 3 shows that the performance of the MCMC-based PF TBD algorithm increases with SNR. When $\text{SNR} \geq 8\text{dB}$, RMSE converges to less than the range resolution.

Conclusion

This paper presents an MCMC-based PF TBD algorithm for detection and tracking multi-target. The proposed algorithm uses an efficient MCMC sampling step to replace the sampling importance resampling step. The simulation results show the proposed algorithm can detect and track multi-target even their paths are close proximity. And, when $\text{SNR} \geq 8\text{dB}$, the performance is very well.

The proposed algorithm here is for a fixed known number of targets. The extension is for a variable unknown number of targets.

Acknowledgements

This work is supported by the Scientific Research Start Project of Introduction Teacher of Leshan Normal University, Project Z1403, Leshan Technology Bureau, Project 14GZD015, and Major Project of Education Department in Sichuan, Project 15ZA0277.

References

- [1] L. Fan, X. Zhang, L. Wei. "TBD algorithm based on improved Randomized Hough Transform for dim target detection", *Progress In Electromagnetics Research C*, 31, pp. 271-285, 2012.
- [2] S. J. Davey, M. G. Rutten, B. Cheung. "A comparison of detection performance for several track-before-detect algorithms", *Eurasip Journal on Advances in Signal Processing*, pp. 22-37, 2007.
- [3] L. Fan, X. L. Zhang, J. Shi. "Track-before-detect procedures for detection of extended object", *EURASIP Journal on Advances in Signal Processing*, 2011:35.
- [4] Y. Boers, J. N. Driessen. "Multitarget particle filter track before detect application", *IEE Proceedings-Radar Sonar and Navigation*, 151(6), pp. 351-357, 2004.
- [5] M. S. Arulampalam, S. Maskell, N. Gordon, et al. "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", *IEEE Transactions on Signal Processing*, 50(2), pp. 174-188, 2002.
- [6] Z. Khan, T. Balch, and F. Dellaert. "MCMC-Based particle filtering for tracking a variable number of interacting targets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11), pp. 1805-1819, 2005.
- [7] P. Green. "Trans-Dimensional Markov Chain Monte Carlo", *Highly Structured Stochastic Systems*, 2003.
- [8] J. Geweke, H. Tanizaki. "Note on the sampling distribution for the Metropolis-Hastings algorithm", *Communication in Statistics Theory and Methods*, 32(4), pp. 775- 789, 2003.
- [9] S. Geman, D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, pp. 721-741, 1984.