3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)
Proceedings of Engineering & Technology (PET)
pp.846-850

A Comparative Study of Particle Filter, PMCMC and Mixture Particle Filter Methods for Tracking in High Dimensional State Spaces

Nesrine Amor[1 *] , Souad Chebbi [2*] and Nidhal Bouaynaya [3]**

[*] *The Laboratory of Technologies of Information and Communication and Electrical Engineering (LaTICE), National School of Engineers of Tunis (ENSIT) ,University of Tunis, Tunis, Tunisia*

[**] *Department of Electrical and Computer Engineering, Rowan University, New Jersey, USA*

e-mails: [1](nisrine.amor@hotmail.fr), [2](chebbi.souad@gmail.com), [3](bouaynaya@rowan.edu)

Abstract- **Nonlinear / non-Gaussian state-space models emerge in many applications in control and signal processing. In the last two decades, the Particle Filter (PF), also known as Sequential Monte Carlo, became a tremendously popular tool to perform tracking in nonlinear/non-Gaussian dynamical systems. Howeverthe computational cost of the PF becomes prohibitive when applied to high dimensional state-spaces. In this context, several authors proposed different strategies to overcome what became to be known as the “curse of dimensionality” of the PF. This paper presents a comparative study between a standard particle filter, the particle Markov chain Monte Carlo and the mixture particle filter, in tracking nonlinear high-dimensional systems.**

Keywords- High dimensional systems, mixture particle filter, nonlinear estimation, particle filtering, Particle Markov Chain Monte Carlo.

I. INTRODUCTION

The tracking problem can generally be considered as a dynamic state estimation problem, where the state is the set of unknown parameters of the target being tracked. The goal is to equentially estimate the state of the target from the noisy observations.

Object tracking has been studied, in recent years, in various applications, including robotics, machine vision, and video analysis. In a Bayesian framework, the tracking problem reduces to the estimation of the posterior probability density function (pdf) of the state given current and past observations [14]. For the linear Gaussian model, the optimal solution is given by the Kalman filter. In general nonlinear/non-Gaussian state-space models, the analytical solution is intractable but several approximations can be used including the extended Kalman filter [1] and the unscented Kalman filter [2].

The particle filter (PF) is a sequential Monte Carlo method to estimate the posterior density of the state, and does not make any assumptions about the pdfs or the linearity of the system model [13]. Specifically, the particle filter approximates the posterior pdf by an ensemble of particles and their associated weights. It has been shown that the particle filter converges asymptotically (as the number of particles tends to infinity) towards the optimal Bayesian filter in the mean square error sense [13] Thus, the PF appears to be the most promising in tracking nonlinear and non-Gaussian systems; however it is inefficient in high dimensional spaces. The number of particles required increases super-exponentially with the dimension of the state [6].

Many authors proposed several approaches to deal with what is known as the curse of dimensionality issue, including Markov Chain Monte Carlo particle filter (PMCMC) [5,6,7], the Mixture Particle Filter (MPF) [8,9,10]. This paper presents a comparative study, in terms of performance and speed, between the standard particle filter, the particle Markov chain Monte Carlo and the mixture particle filter. The performance of the algorithms will be quantified using the minimum square error (MSE).

II. The PARTICLE FILTER

Since their introduction within the signal processing community in the nineties, the particle filter became the tracker of choice to deal with nonlinear and non-Gaussian problems [11]. Its popularity stems from its simplicity, its ease of implementation and performance in highly nonlinear systems.

We consider a discrete-time state-space model defined by the state and measurement equations [3]:

$$x_k = f\left(x_{k-1}\right) + w_k \tag{1}$$

$$y_k = h\left(x_{k-1}\right) + v_k \tag{2}$$

Where x_k and y_k represent, respectively, the unknown system state and measurement at time k; $y_{1:k}$, f_k and h_k are non-linear functions; and w_k and v_k are respectively the system and measurement noises. Let $y_{1:k}$ denote the set of past and current observations up to time *k*. In a Bayesian contect, the estimation of the state x_k relies upon the posterior distribution $P(x_k \mid y_{1:k})$. The optimal state estimate is given by a point estimate, e.g., the mean, of this posterior density. Using Bayes theorem [4,12], the posterior distribution of the state can be written as

ISSN: 2356-5608

$$P(x_k \mid y_{1:k}) = \frac{P(y_k \mid x_k)\, P(x_k \mid y_{1:k-1})}{P(y_k \mid y_{1:k-1})} \tag{3}$$

where $P(y_k \mid y_{1:k-1})$ is the normalizing constant. The prediction of the state distribution is given by the equation:

$$P(x_k \mid y_{1:k-1}) = \int P(x_k \mid x_{k-1}) P(x_{k-1} \mid y_{1:k-1})\, dx_{k-1} \tag{4}$$

The PF uses a set of random particles to estimate the posterior distribution of the state. Specifically, the posterior is approximated by a set of weighted particles $\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}$ [3, 4, 12]:

$$P(x_k \mid y_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_k - x_k^{(i)}\right) \tag{5}$$

where δ is the dirac delta function.

The weights are normalized such that: $\sum_{i=1}^{N} w_k^{(i)} = 1$.

Ideally, the particles need to be sampled from the true posterior, which is not available. Therefore, another distribution, referred to as the *importance distribution* or the *proposal distribution*, $q(x_k / x_{k-1}, y_k)$ is used. Theoretically, the only condition on the importance distribution is that its support includes the support of the posterior distribution. Practically the number of particles is finite and the importance distribution should be chosen to approximate the posterior distribution. The importance weights are given by:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k \mid x_k^{(i)})\; p(x_k^{(i)} \mid x_{k-1}^{(i)})}{q(x_k^{(i)} \mid x_{k-1}^{(i)}, y_{1:k})} \tag{6}$$

For instance, if the importance distribution is given by the prior density, $q(x_k^{(i)} \mid x_{k-1}^{(i)}, y_{1:k}) = p(x_k^{(i)} \mid x_{k-1}^{(i)})$, the weight equation reduces to $w_k^{(i)} = w_{k-1}^{(i)}\, p(y_k \mid x_k^{(i)})$.

Given the discrete approximation to the posterior distribution in (5), the mean of the state at time *k* is:

$$\hat{x}_k = \sum_{i=1}^{N} w_k^{(i)}\, x_k^{(i)} \tag{8}$$

The algorithm of the particle filter method is carried out according to the flowchart in Fig. 1.

III. The PARTICLE MARKOV CHAIN MONTE CARLO

Markov Chain Monte Carlo (MCMC) algorithms were poposed to impove the particle filter performance in high-dimensional state-spaces [5, 7].

A. Markov chain Monte Carlo

The MCMC methods are more effective than the particle filter in high dimensional problems [5,7]. The main idea of MCMC is to construct a Markov chain $(x^1 \quad x^2 \ldots)$ whose stationary distribution corresponds to the distribution of interest *p(x)*. The Metropolis-Hastings algorithm proposes an approach to build the Markov chain, which we adopt in this paper.

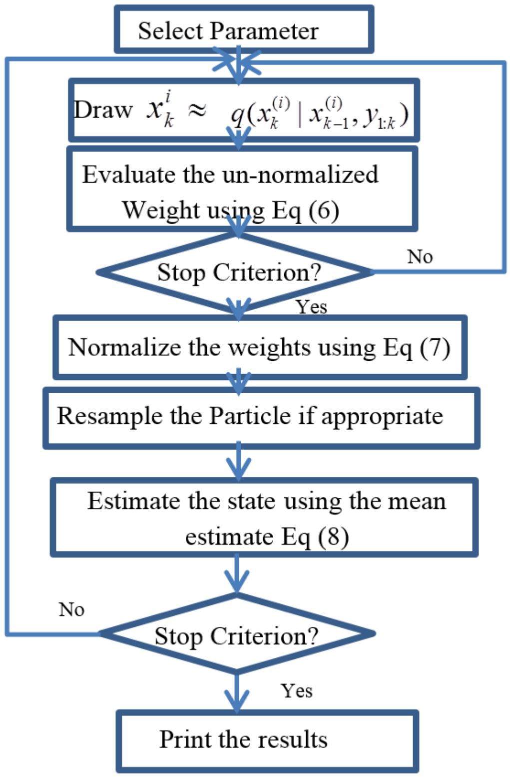

Fig 1. Flowchart of the particle filter

The Metropolis-Hastings algorithm is a two-stage procedure. The first step is to generate a proposal sample, denoted by x^*, from a proposal distribution, q ($x^* \mid x^n$). Usually, $x^* \mid x^n \approx$ Normal (x^n, σ^2)[6,7]. The second step is the accept – reject stage based on calculation of the acceptance probability:

$$\alpha(X^* \mid X_n) = \min\left(1, \frac{p(X^*)\, q(X_n \mid X^*)}{p(X_n)\, q(X^* \mid X_n)}\right), \tag{9}$$

Where *p(x)* denotes the target distribution that we are trying to approximate, here: the posterior density of the state given current and past observations $P(x_k \mid y_{1:k})$.

To implement the acceptance probability, a random number must be generated, between 0 and 1, denoted μ. Then:

If $\mu > \alpha$

  $x^{n+1} = x^n$

else

  $x^{n+1} = x^*$

end if

B. Particule Markov Chain Monte Carlo

The underlying idea of the PMCMC approach is to perform a Metropolis-Hastings (MH) accept-rejection step as a correction for having used a proposal distribution to sample the current state. In this paper, we adopt the development of PMCMC proposed in [5,7].

The algorithm of PMCMC is described as follows:

Step 1. Initialize particle set randomly

For k = 1, . . . , T **do**

Step 2. Propose $x_k^* \approx q\left(x_k \mid x_k^{m-1}\right)$

Step 3. Compute the MH acceptance probability:

$$\alpha\left(x_k^m, x_k^*\right) = \min\left(1, \frac{p\left(x_k^* \mid y_{0:k}\right) q\left(x_k^m \mid x_k^*\right)}{q\left(x_k^* \mid x_k^m\right) p\left(x_k^m \mid y_{0:k}\right)}\right)$$

Step 4. Accept $x_k^m = x_k^*$ with probability $\alpha\left(x_k^m, x_k^*\right)$

Step 5.The new particle set for approximating for $x_k^i = x_k^m$ defined by:

$$P(\hat{x}_k \mid y_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_k - x_k^{(i)}\right)$$

end for

IV. MIXTURE PARTICLE FILTER

The mixture particle filter is based essentially in grouping the particles within components, wich are independately tracked. Each component is assigned a probability that is tracked using another Monte Carlo filter operating at a higher level. The component with the highest probability at any particular time step is considered as the true estimation at that instant. The mixtures of particle filters interact only in the computation of the mixture weights, thus leading to an efficient numerical algorithm. We adopt the mixture model proposed in [8, 9, 10].

The filtering distribution is modeled as an *M*-component mixture model:

$$P(x_t | y_{1:t}) = \sum_{m=1}^{M} \pi_{m,t} \; p_m(x_t | y_t) \tag{10}$$

with $\sum_{m=1}^{M} \pi_{m,t} = 1$.

Note that no parametric model is assumed for the individual mixture components. With a known mixture filtering distribution $P(x_{t-1} \mid y_{t-1})$, and an obtained new prediction distribution, the algorithm leads to predict the distribution for the *m*-th component. Thereby the new prediction distribution is squarely obtained by computing the prediction distribution, each components individually, and combining them in a mixture which it retains the original component weights.

In the mixture particle filter, the filtering distribution $P(x_k | y_k)$ is approximated by:

$$P(x_k | y_{1:k}) = \sum_{m=1}^{M} \pi_{m,k} \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_k - x_k^{(i)}\right) \tag{11}$$

The sum of all the particle weights and the mixture component weights is equal to one:

$$\sum_{m=1}^{M} \pi_{m,k} = 1, \; \sum_{i \in I_m} w_k^{(i)} = 1, \; m = 1, 2, ..., M.$$

New samples are generated from an appropriately chosen proposal distribution, that depends on the new measurement and the old state. $x_k^{(i)} \sim q(x_k \mid x_{k-1}^{(i)}, y_k)$, $i \in I_m$. To maintain a properly weighted sample set, the new particle weights are set to:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k \mid x_k^{(i)}) \; p(x_k^{(i)} \mid x_{k-1}^{(i)})}{q(x_k^{(i)} \mid x_{k-1}^{(i)}, y_{1:k})} \tag{12}$$

The weights are normalized using the following:

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{j \in I_m} \tilde{w}_t^j} \tag{13}$$

The new sample set $\left\{x_t^i, w_t^i\right\}$ $i \in I_m$ is then approximately distributed to $P_m(x_k \mid y_k)$.

The approximation for the new mixture weights given by

$$\pi_{m,k} \approx \frac{\pi_{m,k-1} \tilde{w}_{m.k}}{\sum_{n=1}^{M} \pi_{n,k-1} \tilde{w}_{n.k}}, \qquad \tilde{w}_{m.k} = \sum_{i \in I_m} \tilde{w}_k^i \tag{14}$$

V. SIMULATIONS RESULTS

This section compares the performance of the standard particle filter with the mixture particle filter and PMCMC. We consider the following nonlinear dynamical system, where the state has dimension 5.

$$\begin{aligned} x_{k+1} &= \frac{x_k}{2} + 25 * \frac{x_k}{1 + x_k^2} + 8 * \cos(1.2k) + w_k \\ y_k &= \frac{x_k^2}{20} + v_k \end{aligned} \tag{15}$$

where w_k and v_k are Gaussian white noise. This is a severely nonlinear example in both the system and measurement equations.

We compare the performance of all above algorithms applied to this example. The following figure shows the results of each method.

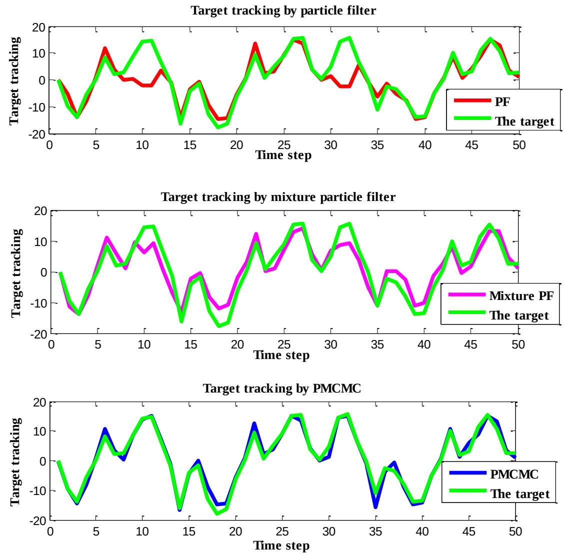

Fig 2. Target tracking of the dynamical system in (15) using the PF (top), micture PF (middle) and PMCMC (buttom).

Figure 3 shows the minimum square error of each algorithm.

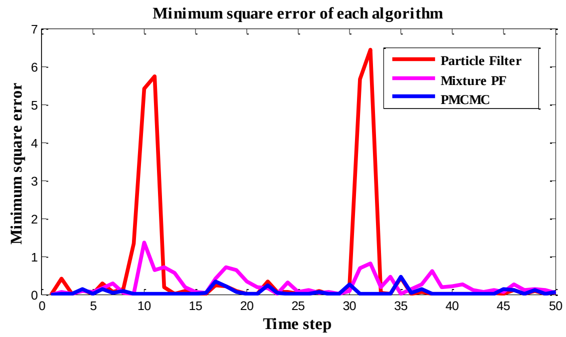

Fig3 Minimum square error of the PF (red), mixtue PF (violet) and PMCMC (blue).

Figures 2 and 3 display 50 time steps. The Standard particle filter and PMCMC use 500 particles. The mixture particle filter was constrained to have a maximum of five components and 100 particles per component.

The comparative results for a typical run with 500 particles are given in Figures 2 and 3. As is clearly shown in Figure 3, the standard particle filter has two error bursts, the first one between 7s and 12s and the second one between 30s and 33s. The mixture particle filter and PMCMC, however, are able to track the target successfully. Note that between PMCMC and mixture particle filter, the PMCMC is more efficient than the mixture PF as illustrated in Figure 3 using MSE. The convergence time of PMCMC is slower than the two other methods PMCMC converged in 2.430s, PF in 1.923s and Mixture PF in 2.276s.

VI. CONCLUSION

This paper presents a comparative study between a standard Particle Filter and two strategies, based on particle filter: the Particle Markov Chain Monte Carlo and Mixture Particle Filter approaches. These methods clearly represent interesting alternatives to the classic particle filter, especially in high-dimensional state-spaces. The simulation results showed that, in terms of MSE performance, convergence and running time, the PMCMC approach exhibits a good tracking performance compared to the particle filter and mixture particle filter; however, it is slower than the other two methods.

VII. REFERENCES

[1] Welch, G. and Bishop, G., "An introduction to the Kalman filter". In Proceedings of SIGGRAPH 2001 Course 8. Computer Graphics. ACM Press/Addison-Wesley-2001.

[2] Julier, S. and Uhlmann, J.," A new extension of the Kalman filter to nonlinear systems". In Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls, pages 182–193, Orlando, FL, USA-1997.

[3] Petia G., Nidhal B., Filipe S.,Lyudmila M. and Lakhmi J.," A Beamformer-Particle Filter Framework for Localization of Correlated EEG Sources" , IEEE journal of biomedical and health informatics.

[4] Xi C., Simo S., Simon G." A Bayesian Particle Filtering Method For Brain Source Localization", Digital Signal Processing, Volume 47, December 2015, Pages 192-204.

[5] C. Andrieu, A. Doucet and R. lenstein," Particle Markov chain Monte Carlo methods" Journal of the royal Statistical. Society. B (2010) 72, Part 3, pp. 269–342.

[6] F. Septier, S. Pang, A. Carmi, S. Godsill,"On MCMC-Based Particle Methods for Bayesian Filtering: Application to Multitarget Tracking" IEEE International Workshop on Computational Advances in Multi Sensor Adaptive Processing (CAMSAP), Dec 2009, pp.1280-1287.

[7] Lyudmila Mihaylova and Avishy Carmi ,"Particle algorithms for filtering in high dimensional state spaces": a case study in group object tracking, ICASSP 2011.

[8] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multi-modality through mixture tracking," in Int. Conf. Computer Vision, 2003, vol. 2, pp. 1110–1116.

[9] Peng Cui, Li-Feng Sun, Fei Wang, and Shi-Qiang Yang, "Contextual Mixture Tracking" IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 11, NO. 2, FEBRUARY 2009.

[10] Lubina P Ibrahim , A.Neela Madheswari, Smitha.S, "Improved object tracking using mixture particle filter for maintaining multi-modality", IRACST – International Journal of Advanced Computing, Engineering and Application (IJACEA), ISSN: 2319-281X, Vol. 2, No. 5, October 2013.

[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] A. Doucet and A. M. Johansen, A tutorial on particle filtering and smoothing: Fifteen years later," in Handbook of Nonlinear Filtering, vol. 12, 2009, pp. 656-704.

[13] A tutorial on particle filtering and smoothing: Fifteen years later," Handbook of Nonlinear Filtering, vol. 12, pp. 656–704, 2009.

[14] Dan Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches, Wiley 2006.