Military Technical College
Kobry Elkobbah,
Cairo, Egypt

5th International Conference
on Electrical Engineering
ICEENG 2006

APPLICATION ON SMOOTHING PARTICLE FILTER IN TRACKING A HIGHLY MANEUVERABLE TARGET IN A MULTIPLE-SENSORS NETWORK

Kamel[*] H. and Badawy[**] W.

ABSTRACT

In this paper we apply the smoothing particle filter to track a highly maneuverable target in a multiple-sensors network. We address the scenario of a single highly-maneuverable target moving through a field of stationary sensors with known locations. The target is tracked through the sensors filed using either all sensors or active sensors within a gate around the target. Results have been compared to tracking the same target using conventional particle filter. Smoothing particle filter showed improvement in the performance.

1. INTRODUCTION

A common application of sequential state estimation is the tracking of targets moving through a sensors' filed of view. However, state evolution is often not easily modeled in a predictive fashion. This can happen either when the system being studied is not well understood, or when the random changes in the state are large enough to dominate the predictable changes. The second case is that of a highly maneuverable target [1]. In practical target tracking environment, with the presence of uncertain target models and incomplete observations, nonlinear models in state equation and measurement relations as well as non-Gaussian noise assumptions are more suitable for high performance requirements and some realistic applications. Traditionally, these nonlinear problems are solved using linearizing tracking filters, mainly extended Kalman filters (EKF) [2] but the linearized method is not efficient enough in practice. Several methods for nonlinear non-Gaussian state space model are proposed as in [3], [4], and [5]. They are called "Particle Filters" because of their approximation of non-Gaussian distribution of the state by many numbers of particles in state space. The main advantage of the particle filters is to be able to handle any functional nonlinearity and system of measurement noise of any distribution. But the highly uncertainty and incompleteness of the information in maneuvering target-tracking problem will weaken this advantage. To overcome this weakness, the smoothing particle filter is used [6].

[*]Ph.D. Candidate, Elec. & Comp. Eng. Department, University of Calgary, AB, Canada.
[**]Associate Professor, Elec. & Comp. Eng. Department, University of Calgary, AB, Canada.

In this paper, we are going to address the same scenario mentioned in [7]: a single target moving through a field of arbitrary located sensors. Each sensor can be configured to be active or inactive at each time period. At each scan time, each active sensor outputs a binary observation indicating that the target has or has not been detected. The target trajectory is estimated from the time sequence of target detections by the various sensors.
This paper is organized as follows. We first describe the particle filter equations and algorithm as well as the main idea of the smoothing particle filter. Then, target and sensor models are given. Performance analysis is discussed in section 4. Finally, we state our conclusion.

2. THE BASIC PARTICLE FILTER AND SMOOTHING PARTICLE FILTER

We consider a dynamic system represented by the stochastic process $(X_t) \in R^{n_x}$ whose temporal evolution is given by the state equation:

$$X_t = F_t(X_{t-1}, V_t) \tag{1}$$

The measurement equation is given by:

$$Y_t = H_t(X_t, W_t) \tag{2}$$

The two processes $(V_t) \in R^{n_v}$ and $(W_t) \in R^{n_w}$ are white noises. Moreover, it is to be noted that no linearity hypothesis on F_t and H_t is done. We will denote by $Y_{0:t}$ the sequence of the random variables (Y_0, … , Y_t) and by $y_{0:t}$ one realization of this sequence. The main problem consists in computing at each time t the conditional density L_t of the state X_t given all the observations accumulated up to t, i.e., $L_t = p(X_t \mid Y_0 = y_0, \ldots, Y_t = y_t)$ and also in estimating any functional of the state $g(X_t)$ by the expectation $E[g(X_t) \mid Y_{0:t}]$. The Recursive Bayesian filter resolves exactly this problem in two steps at each time t: prediction step and correction step.
Suppose we know L_{t-1}. The prediction step is done according to the following equation:

$$p(X_t = x_t \mid Y_{0:t-1} = y_{0:t-1}) = \int_{R^{n_x}} p(X_t = x_t \mid X_{t-1} = x) L_{t-1}(x)\, dx \tag{3}$$

Using (1), we can calculate $p(X_t = x_t \mid X_{t-1} = x)$:

$$p(X_t = x_t \mid X_{t-1} = x) \int_{R^{n_v}} p(X_t = x_t \mid X_{t-1} = x, V_t = v) p(V_t = v \mid X_{t-1} = x)\, dv \tag{4}$$

$$= \int_{R^{n_v}} \delta(x_t - F_t(x, v))\, p(V_t = v)\, dv \tag{5}$$

where $\delta(x)$ denotes the Dirac distribution. Then, in the correction step, the Baye's rule enables us to compute L_t:

$$L_t(x_t) = \frac{p(Y_t = y_t \mid X_t = x_t) p(X_t = x_t \mid Y_{0:t-1} = y_{0:t-1})}{p(Y_t = y_t \mid Y_{0:t-1} = y_{0:t-1})} \tag{6}$$

Applying (2), we can rewrite $p(Y_t = y_t \mid X_t = x_t)$ as:

$$p(Y_t = y_t \mid X_t = x_t) = \int_{R^{n_w}} \delta(Y_t - H_t(x_t, w))\, p(W_t = w) dw \tag{7}$$

As well, the denominator in (6) could be expressed as follows:

$$p(Y_t = y_t \mid Y_{0:t-1} = y_{0:t-1}) = \int_{R^{n_x}} p(Y_t = y_t \mid X_t = x)\, p(X_t = x \mid Y_{0:t-1} = y_{0:t-1}) dx \tag{8}$$

The particle filter, named also *Sampling Importance Resampling* (SIR) as mentioned previously, proposes to approximate the densities $(L_t)_t$ by a finite weighted sum of N-Dirac densities centered on elements of R^{n_x}, named "*Particles*". The application of the particle filter requires that one knows how:

- to sample from initial prior marginal $p(X_0)$,
- to sample from $p(V_t)$ for all t, and
- to compute $p(Y_t = y_t \mid X_t = x_t)$ for all t through known function l_t such that $l_t(y;x) \propto p(Y_t = y_t \mid X_t = x_t)$ where missing normalization should not depend on x.

The first particle set S_0 is created by drawing N independent realizations from $p(X_0)$ and assigning uniform weight $1/N$ to each of them. Then, suppose we dispose at time t-1 of the particle set $S_{t-1} = \left(s_{t-1}^n, q_{t-1}^n\right)_{n=1,\ldots,N}$ where $\sum_{n=1}^{N} q_{t-1}^n = 1$. Posteriori marginal L_{t-1} is then estimated by the probability density $L_{S_{t-1}} = \sum_{n=1}^{N} q_{t-1}^n \delta_{s_{t-1}^n}$. Then, the weight of each particle is updated during the correction step. Up to a constant, equation (6) comes down to adjust the weight of predictions by multiplying it by the likelihood $p(y_t|x_t)$.

In practice, the particle set is finite and the major drawback of this algorithm is the degeneracy of the particle set: only few particles keep high weights and the others have very small ones. The resampling is a good way to remedy this drawback because it cancels the particles of smallest weight. To measure the degeneracy of the algorithm, the effective sample size N_{eff} has been introduced in [9] and [10] in more general context in "*importance resampling*". We can estimate this quantity by:

$$\hat{N}_{eff} = 1 \Big/ \sum_{n=1}^{N} \left(q_t^n\right)^2 \tag{9}$$

which measures the number of meaningful particles. The resampling is then done only if $\hat{N}_{eff} < N_{threshold}$. It enables the particle set to better learn the process and to keep its memory during the interval where no resampling occurs. More details can be found in [9], [10], and [11] and we summarize the whole particle filter in Algorithm 1.

However, due to the highly uncertainty and incompleteness of the information in maneuvering target-tracking problem will weaken this advantage. To overcome this weakness, the smoothing particle filter was used [6]. The smoothing particle filter is applied for maneuvering target-tracking problems. Its algorithm combines the particle filter; which tackles the nonlinear non-Gaussian peculiarities, and smoothing of the PDF of system modes; which settles the maneuverability of the target. The algorithm is shown in Algorithm 2.

3. TARGET AND SENSOR MODELS

3.1. Target Model

The target is constrained to 2D motion. The target state is composed of the target's position and velocity at time t_k. A discrete-time linear system driven by white Gaussian noise, to model the target dynamics, is used. The target state vector is given by:

$$X_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{bmatrix} \tag{10}$$

where x_k and y_k represent the target's position at time k; as well $\dot{x}_k$ and $\dot{y}_k$ represent the target's velocity.

Consider the scan rate is T. The maneuvering target system dynamics are given by:

$$X_{k+1} = F\, X_k + W_k \tag{11}$$

where W_k is a vector Gaussian white noise process The transition matrix F is defined as follows:

(1) Constant velocity model:

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

(2) Coordinated turn model:

$$F = \begin{bmatrix} 1 & 0 & \dfrac{sin(\dot{\theta}T)}{\dot{\theta}} & \dfrac{cos(\dot{\theta}T)-1}{\dot{\theta}} \\ 0 & 1 & \dfrac{1-cos(\dot{\theta}T)}{\dot{\theta}} & sin(\dot{\theta}T) \\ 0 & 0 & cos(\dot{\theta}T) & -\dfrac{sin(\dot{\theta}T)}{\dot{\theta}} \\ 0 & 0 & sin(\dot{\theta}T) & cos(\dot{\theta}T) \end{bmatrix} \tag{13}$$

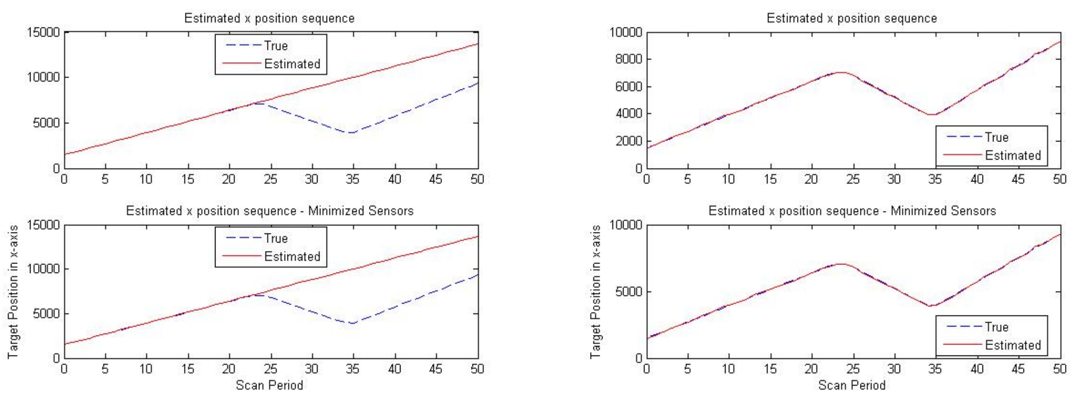
where $\dot{\theta}$ denotes the turn rate in radians/second [12].

3.2. Sensor Model

We've applied the same sensor model as mentioned in [7]. In our scenario, N sensors are located at known locations in the target plane. At each time k, each sensor configured to be active or inactive. The received signal at each sensor is the sum of noise and the signal emitted (or reflected) by the target (if present). The sensor activation depends on an adaptive algorithm in which only sensors within a given distance from the target are activated. The detailed algorithm is mentioned in [7].

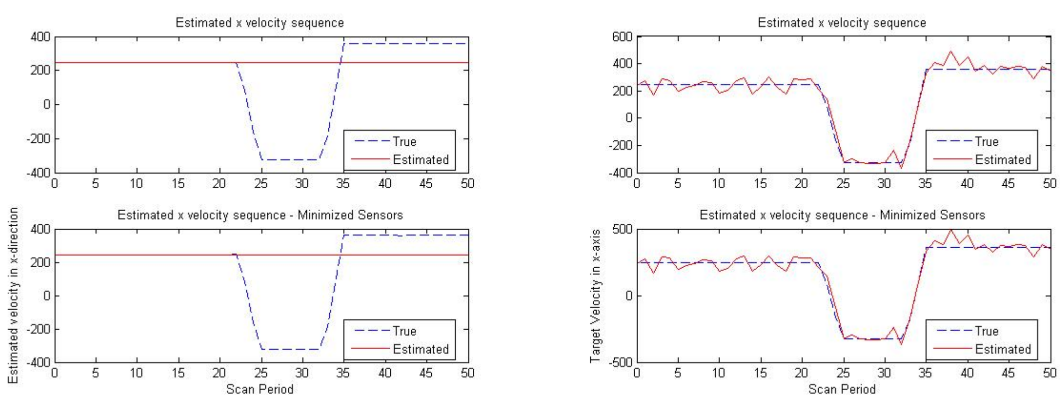
4. PERFORMANCE ANALYSIS

We've applied the smoothing particle filter to track a highly-maneuverable target which makes a Z-shape trajectory. The target was tracked during 50 scan periods. The initial target speed was chosen to be 265 m/s and the turn rate $\pi/4$. The performance of the simulation was evaluated using two sets of 200 Monte Carlo runs. In the first set, we tracked the target using conventional particle filter. Then, we tracked the same target using the smoothing particle filter. Fig. 1 shows the performance of both particle filter and smoothing particle filter tracking the *x*-axis position of a highly-maneuverable target in a multiple-sensors network; meanwhile Fig. 2 shows their performances in tracking the target's velocity. In Fig. 3, we show the performance of both filters in tracking the trajectory of the target flying in a field of stationary sensors with known location distributed randomly. In the three figures, we can notice that the particle filter fails to track the maneuverable target when the maneuver starts. On the other hand, we can notice that the smoothing particle filter can track the same target successfully.



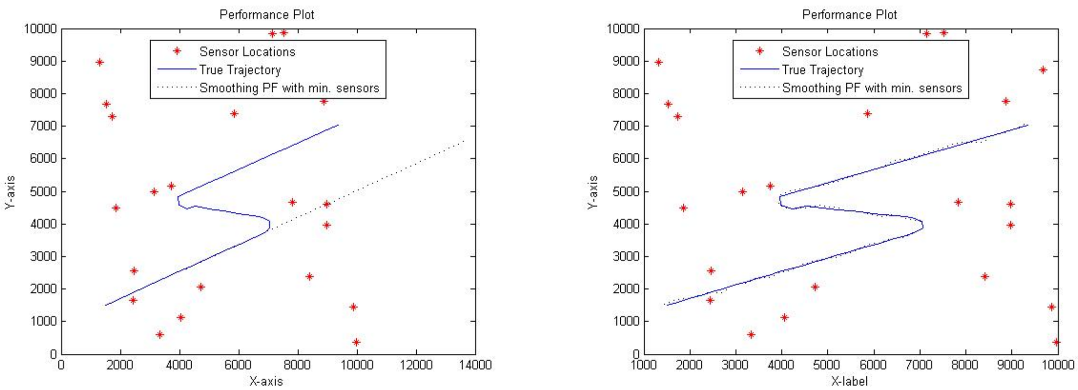
(a) (b)

Fig. 1 True and estimated target position in *x*-axis using (a) particle filter and (b) smoothing particle filter with both all and minimized sensors



(a) (b)

Fig. 2 True and estimated target velocity in *x*-axis using (a) particle filter and (b) smoothing particle filter with both all and minimized sensors



(a) (b)

Fig. 3 True and estimated target trajectory using (a) particle filter and (b) smoothing particle filter with both all and minimized sensors

5- Conclusion

The simulation runs showed the failure of the particle filter tracking a highly-maneuverable target at the turn point; meanwhile, the smoothing particle filter tracked the highly-maneuverable target successfully in multiple-sensors network with both all and minimized sensors.

APPENDICES

Algorithm 1 Basic particle filter with adaptive resampling

for n=1,…,N **do**

Generate a random sample v_t^n from $p(V_t)$.

Compute $s_{t|t-1}^n = F_t\left(s_{t-1}^n, v_t^n\right)$

end for

Correction

for n=1,…,N **do**

Compute $q_t^n = \dfrac{l_t\left(y_t; s_{t|t-1}^n\right)q_{t-1}^n}{\sum_{n=1}^{N} l_t\left(y_t; s_{t|t-1}^n\right)q_{t-1}^n}$

end for

Estimation

Estimate $E\{x_t\}$ by $\hat{E}\{x_t\} = \sum_{n=1}^{N} q_t^n \, g\left(s_{t|t-1}^n\right)$

Effective size estimation

Calculate $\hat{N}_{eff} = 1 \Big/ \sum_{n=1}^{N}\left(q_t^n\right)^2$

Resampling

if $\hat{N}_{eff} < N_{threshold}$ **then**

for n=1,…,N **do**

Draw s_t^n from $\sum_{k=0}^{N} q_t^k \, \delta_{s_{t|t-1}^k}$

Set $q_t^n = 1/N$

end for

else

for n=1,…,N **do**

$s_t^n = s_{t|t-1}^n$

end for

end if

Algorithm 2 Smoothing particle filter

for i=1,…, N **do**

simulate a sample x_1^j from $p(x_1)$ with equal weights.

set initial mode probabilities $p(m_1^i \mid Y_1) = p(m_j)$, j=1,…,N.

end for

for i=1,…, N **do**

compute $x_{t+1}^i = F(x_t^i, m_t^i, w_t)$ where m_t^i is a sample drawn from the system mode set M with distribution $\{p(m_t^i \mid Y_t)\}_{j=1,\dots,N}$ and w_t is a sample drawn from the white noise PDF.

compute $\bar{x}_{t+1}^i = E\{x_{t+1}^i\}$ and $N_{m_t^i}$

Compute the posterior smoothed mode probabilities $p(m_t^i \mid Y_{t+1}) = \frac{1}{c} \sum_{j=1}^{N_{m_t^i}} p(y_{t+1} \mid \bar{x}_{t+1}^{k_j}) \pi_{t+1}^{k_j}$, where c is the normalizing factor. Using the posterior mode probabilities, predict the particles again as in second step.

Calculate the likelihood weights and normalize $\pi_{t+1}^i = \frac{1}{c'} p(y_{t+1} \mid x_{t+1}^i)$, where c' is used to normalize the weights sum to 1.

Calculate probabilities of the system mode at time step t+1: $p(m_{t+1}^i \mid Y_{t+1}) \propto \sum_{j=1}^{N_s} p(m_{t+1}^i \mid m_t^i)\, p(m_t^j \mid Y_{t+1})$, i = 1,…,N.

Perform resampling and roughening procedure of the set $(x_{t+1}^i, \pi_{t+1}^i)_{i=1,\dots,N}$ to overcome the poorness of the particle filter.

end for

Increase t and iterate to second step.

REFERENCES

[1] Schell, C., Linder, S., and Zeidler, J., "Tracking Highly Maneuverable Targets with Unknown Behaviour," *Proc. of IEEE*, vol. 92, No. 3, pp. 558-574, March 2004.

[2] Julier, S., and Uhlmann, J.K., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proc. of AeroSense: The 11th Int. Symp. On Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[3] Gordon, N., Salmond, D., and Smith, A.F., "*Novel Approach to Nonlinear / Non-Gaussian Bayesian State Estimation*," IEE Proceedings-F, 140, No.2, pp.107-113, 1993.

[4] Isard, M., and Blake, A., "*CONDENSATION – Conditional Density Propagation for Visual Tracking*," Journal of Computer Vision, 29, No.1, pp. 5-28, 1998.

[5] Kitagawa, G., "*Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models*," Journal Computational and Graphical Statistics 5, No.1, pp. 1-25, 1996.

[6] Li, Y.Q., Shen, Y., and Liu, Z.Y., "A New Smoothing Particle Filter to Track A Maneuvering Target," *Proceedings of the Second International Conference on Machine Learning and Cybernetics,* Wan, pp. 1004-1008, 2-5 November, 2003.

[7] Stevens, T.J. and Morrell, D., "Minimization of Sensor Usage for Target Tracking in a Network of Irregular Spaced Sensors," *Statistical Signal Processing, 2003 IEEE Workshop on*, pp. 539 – 542, 28 Sept.-1 Oct. 2003.

[8] Hue, C., Jean-Pierre Le Cadre, and Pérez, C., "*Suivi des Objets Multiples par Filtrage Particuliaire*," *Rapport de recherché no. 4033,* Unité de Recherche INRIA, Rennes, October 2000.

[9] Kong, Liu, J.S., and Wong, W.H., "*Sequential Imputation Method and Bayesian Missing Data Problems*," J. Amer Statist. Assoc., 89:278-288, 1994.

[10] Liu, J.S., "Metropolized Independent Sampling with Comparison to Rejection Sampling and Importance Sampling," *Statistics and Computing*, 6:113-119, 1996.

[11] Lurie, D. and Hartley, H.O., "Machine-Generation of Order Statistics for Monte Carlo Computations," *The American Statistics*, 26:26-27, February 1972.

[12] Malcolm, W., Doucet, A., and Zollo, S., "Sequential Monte Carlo Tracking Schemes for Maneuvering Targets with Passive Ranging," *Proceedings of FUSION 2002, the 5th International Conference on Information Fusion,* Annapolis, Marylands, USA, to appear Jul 2002.