A STUDY ON MULTITARGET TRACKING WITH ADAPTIVE LOCAL LINEARISATION PARTICLE FILTERS

George Kravaritis, Bernard Mulgrew

Institute for Digital Communications
School of Engineering and Electronics, University of Edinburgh
The King's Buildings, Mayfield Road, EH9 3JL, UK
Email: {G.Kravaritis, B.Mulgrew}@ed.ac.uk

ABSTRACT

Over the last decade much research has been conducted on exploiting particle filtering techniques in the field of target tracking. Although the major body of the work in that area concerns tracking a single target, algorithms have also been proposed for the multiple target case. In most multitarget algorithms the state estimator is a basic particle filter (e.g. a sequential importance resampling filter) used in a complex multitarget structure. The contribution of this paper is the use of the more powerful local linearisation particle filter as the basic estimation tool, for the multitarget problem.

1. INTRODUCTION

The particle filter (PF) is a recursive Bayesian estimator which was introduced by Gordon et al. in [1]. The basic PF (sequential importance resampling, SIR) samples randomly the state distributions with a set of samples or 'particles', which when predicted using the system dynamics, characterise the *posterior* estimates. Theoretically the PF converges to the optimal nonlinear/non-Gaussian state estimator, when the number of particles tends to infinity.

When multiple targets exist, modifications to the standard single-target algorithms should be made. In order to built target tracks a data association scheme (e.g. [2], [3]) should be adopted so as to assign the new observations to the targets. After the association a separate tracking filter for each of the targets can be used.

Another approach is to augmented the state space with the states of all targets and to use the resulting multitarget probability distribution (e.g. [4], [5]) for computing the targets' states. The latter is appropriate if it is not necessary to form the individual tracks, but we are interested just in the states of the targets at any given time.

This research was supported by the Strategic Alliance Agreement ZN1217 in Signal and Information Systems between BAE Systems and the University of Edinburgh.

In the multitarget particle filtering literature most of the developed techniques utilise a basic SIR estimator in conjunction with a more sophisticated multitarget algorithmic structure. In this paper we study the benefits of using a local linearisation particle filter (LLPF) [6] as the main estimation tool, concentrating in parallel on a computational efficient filter implementation. The LLPF approximates the use of the optimal sampling density employing a set of extended Kalman filters (EKF) to initially predict the particles and to form the *a-priori* state estimates. Compared with the SIR it has been shown that it improves the tracking performance.

The structure of the paper is as follows. Section 2 discusses the multitarget tracking problem, section 3 outlines the tracking algorithms, section 4 presents the simulation results and section 5 summarizes.

2. TRACKING MULTIPLE TARGETS

This section describes the multitarget tracking problem. A comprehensive introduction to tracking can be found in textbook [7]. In the scenario to be studied a static radar monitors L targets moving with constant velocity (CV). Their position and velocity are perturbed with process noise **s**. The model equation of a single target is given below:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{s}_{k-1} \tag{1}$$

where the linear transition matrix is

$$\mathbf{F} = \begin{bmatrix} 1 & T_r & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_r \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

and T_r is the radar update time. The state vector:

$$\mathbf{x}_k = [x_k \; y_k \; \dot{x}_k \; \dot{y}_k]^T \tag{3}$$

consists of the position and velocity of the target in the Cartesian x-y plane. We let $\mathbf{s}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$, where $\mathbf{Q}_k$ is the diagonal process noise covariance matrix.

The radar lies in the origin of the plane and measures the azimuth angle and the range of the targets:

$$\mathbf{z}_k = \begin{bmatrix} \theta_k \\ R_k \end{bmatrix} \tag{4}$$

The measurement equation for a single target is given next.

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{5}$$

where h is the nonlinear function that transforms the target's position coordinates from Cartesian to polar:

$$h(\mathbf{x}_k) = \begin{bmatrix} \arctan(y_k/x_k) \\ \sqrt{x_k^2 + y_k^2} \end{bmatrix} \tag{6}$$

For the measurement noise we let $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$, where $\mathbf{R}_k$ is the diagonal measurement noise covariance matrix.

For simplicity in our analysis we consider 2 targets. Suppose that $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ are the state vectors of the two targets. To implement a multitarget filter we augment the state space and we create the *multitarget* state which is

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^{(1)} \\ \mathbf{x}_k^{(2)} \end{bmatrix} \tag{7}$$

Likewise we augment the noise vectors $\mathbf{s}_k = [\mathbf{s}_k^{(1)}\ \mathbf{s}_k^{(2)}]^T$ and $\mathbf{v}_k = [\mathbf{v}_k^{(1)}\ \mathbf{v}_k^{(2)}]^T$, from equations (1) and (5). The augmented nonlinear function h in (6) becomes

$$h(\mathbf{x}_k) = \begin{bmatrix} \arctan(y_k^{(1)}/x_k^{(1)}) \\ \sqrt{x_k^{(1)2} + y_k^{(1)2}} \\ \arctan(y_k^{(2)}/x_k^{(2)}) \\ \sqrt{x_k^{(2)2} + y_k^{(2)2}} \end{bmatrix} \tag{8}$$

The transition and noise matrices $\mathbf{F}$, $\mathbf{Q}_k$ and $\mathbf{R}_k$ are also augmented: $\mathbf{F} = diag(\mathbf{F}, \mathbf{F})$, $\mathbf{Q}_k = diag(\mathbf{Q}_k^{(1)}, \mathbf{Q}_k^{(2)})$ and $\mathbf{R}_k = diag(\mathbf{R}_k^{(1)}, \mathbf{R}_k^{(2)})$. For multitarget tracking we apply equations (1) and (5) using the augmented quantities. The above analysis can be generalised for the L-target case, if we augment the state and measurement spaces so as to include all L targets.

The existence of multiple targets necessitates the use of a data association logic, so as to assign at each scan the new observations to the targets' tracks. For that purpose we associate each measurement with the track which is *nearest* to it. For our 2-target scenario, we use as a metric the Euclidean distance, which is simple and computational cheap to implement. When the number of targets is large, it is more appropriate to use a more sophisticated statistical distance for the associations.

Consider again that we have two targets. In scan k we get two measurements $\mathbf{z}'_k$ and $\mathbf{z}''_k$. Since we do not know at this point the correct association we construct the two multitarget measurement update candidates:

$$\mathbf{z}_k^a = [\mathbf{z}'_k\ \mathbf{z}''_k]^T,\ \mathbf{z}_k^b = [\mathbf{z}''_k\ \mathbf{z}'_k]^T \tag{9}$$

We then (*a-priori*) predict the multitarget state and compute the Euclidean distances d_a and d_b between the predicted targets' position and measurements $\mathbf{z}_k^a$ and $\mathbf{z}_k^b$ respectively. We select the measurement whose distance from the targets is smaller:

$$\mathbf{z}_k = \begin{cases} \mathbf{z}_k^a, & \text{if } d_a < d_b. \\ \mathbf{z}_k^b, & \text{if } d_a > d_b. \end{cases} \tag{10}$$

After the measurement association, we use the particle filters to estimate the final posterior state. If the number of the targets is large an optimisation scheme (like the Viterbi algorithm) would be more computational efficient, than computing all distances from the different measurement candidates.

We note here that first we predict the particles so as to compute the *a-priori* state, and then we associate the observations. When using an SIR this is straightforward, since the filter predicts the particles using just the model equation (1) with random noise samples. Care however should be taken when using a LLPF, given that the prediction step incorporates the measurements as well (whose association at this point is unknown). In what follows we will address this problem and we will describe in detail the two algorithms.

3. PARTICLE FILTER ALGORITHMS

In this section we present the multitarget SIR and LLPF algorithms. An introduction to particle filtering and a detailed description of the single-target SIR and LLPF can be found in book [8].

3.1. Multitarget SIR

The SIR algorithm is considered to be the basic particle filter. Consider that the set $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{Ns}$ is a weighted sampled approximation of the multitarget state posterior probability density function $p(\mathbf{x}_k|\mathbf{z}_k)$, where $\mathbf{x}_k^i$ is a set of N_s samples (or particles) with associated weights w_k^i, $\mathbf{x}_k$ is the multitarget state vector and $\mathbf{z}_k$ is the associated measurement vector. The multitarget SIR consists of a prediction, an association and an update step.

First the algorithm predicts the particles $\mathbf{x}_{k-1}^i$ one step ahead by generating N_s process noise samples $\mathbf{s}_{k-1}^i$ and setting

$$\mathbf{x}_k^i = f_{k-1}(\mathbf{x}_{k-1}^i, \mathbf{s}_{k-1}^i) \tag{11}$$

where f_{k-1} is the state dynamics function at $k-1$.

The association step starts by computing the *a-priori* state estimate using the mean of the predicted particles:

$$\bar{\mathbf{x}}_k = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{x}_k^i \tag{12}$$

Then the distances between the predicted position (obtained from $\bar{\mathbf{x}}_k$) and the candidate augmented measurements are calculated. For the 2-target case we form $\mathbf{z}_k^a$ and $\mathbf{z}_k^b$ from (9) and we transform them to the Cartesian plane using the following forms:

$$\begin{aligned} \mathbf{z}_{C,k}^a &= \begin{bmatrix} \mathbf{z}'_{C,k} \\ \mathbf{z}''_{C,k} \end{bmatrix} = \begin{bmatrix} R'_k \cos\theta'_k \\ R'_k \sin\theta'_k \\ R''_k \cos\theta''_k \\ R''_k \sin\theta''_k \end{bmatrix} \\ \mathbf{z}_{C,k}^b &= \begin{bmatrix} \mathbf{z}''_{C,k} \\ \mathbf{z}'_{C,k} \end{bmatrix} = \begin{bmatrix} R''_k \cos\theta''_k \\ R''_k \sin\theta''_k \\ R'_k \cos\theta'_k \\ R'_k \sin\theta'_k \end{bmatrix} \end{aligned} \tag{13}$$

The predicted position is given after truncating $\bar{\mathbf{x}}_k$ keeping just its position elements:

$$\bar{\mathbf{x}}_{p,k} = [\bar{x}_k^{(1)}\ \bar{y}_k^{(1)}\ \bar{x}_k^{(2)}\ \bar{y}_k^{(2)}\ ]^T \tag{14}$$

The distances d_a and d_b between $\bar{\mathbf{x}}_{p,k}$ and $\mathbf{z}_{C,k}^a, \mathbf{z}_{C,k}^b$ are given by the well known Euclidean metric or ℓ^2-norm:

$$d_a = |\bar{\mathbf{x}}_{p,k} - \mathbf{z}_{C,k}^a|_2,\ d_b = |\bar{\mathbf{x}}_{p,k} - \mathbf{z}_{C,k}^b|_2 \tag{15}$$

For the association we use relation (10).

The update phase assigns a normalised weight w_k^i to each particle $\mathbf{x}_k^i$ proportional to its likelihood function:

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{j=1}^{N_s} \tilde{w}_k^j} \tag{16}$$

where

$$\tilde{w}_k^i = p(\mathbf{z}_k|\mathbf{x}_k^i) = \mathcal{N}(\mathbf{z}_k; h_k(\mathbf{x}_k^i), \mathbf{R}_k) \tag{17}$$

The *a-posteriori* state estimate $\hat{\mathbf{x}}_k$ is given then by the weighted sum of the particles:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N_s} w_k^i \mathbf{x}_k^i \tag{18}$$

The final step is to resample with replacement the posterior state distribution, using a scheme like systematic resampling (SR) [8]. Firstly we construct the cumulative distribution function (cdf):

$$c_1 = 0 \text{ and } c_i = c_{i-1} + w_k^i,\ \text{where } i \in (1 \dots N_s) \tag{19}$$

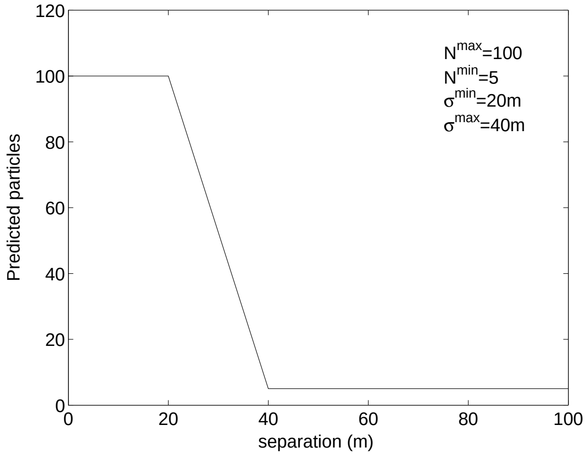

Fig. 1. Number of predicted particles over measurement separation.

and we draw a uniformly distributed starting point:

$$u_i \sim \mathrm{U}(0, N_s^{-1}) \tag{20}$$

For every particle $\{\mathbf{x}_k^j\}_{j=1}^{N_s}$ we move along the cdf:

$$u_j = u_1 + N_s^{-1}(j-1) \tag{21}$$

and while $u_j > c_i$ we force $i = i + 1$. When $u_j < c_i$ we assign the resampled particle $\mathbf{x}_k^{j*} = \mathbf{x}_k^i$. With the resampling step we eliminate particles with small weights and we robustify the particle filter.

3.2. Adaptive Multitarget LLPF

Doucet at al. [6] suggested to use linearisation techniques so as to incorporate the measurements into the prediction phase of the filter. The proposed LLPF uses EKFs to estimate the *a-priori* particles. In the multitarget case the association of the measurements is not known before the initial prediction. Therefore one should construct all possible association hypotheses and predict the states using these hypotheses. In the 2-target case for example, since there are two different association hypotheses ($\mathbf{z}_k^a$, $\mathbf{z}_k^b$), one should predict twice the multitarget state vector. The first estimate will be computed using $\mathbf{z}_k^a$ in the EKFs and the second using $\mathbf{z}_k^b$. The measurements are associated based on the minimum distance of the two *a-priori* estimates from their measurements.

Let us concentrate on the 2-target case. The association logic as described before, requires to predict twice the N_s particles of the filter. For L targets we have to predict $L \cdot N_s$ particles. Since for a large L this would be computationally inefficient, we should use an adaptive number of predicted particles based on an 'association difficulty' metric. For our algorithm we will vary the number of the predicted particles according to the Euclidean distance σ_k between $\mathbf{z}_k^a$ and $\mathbf{z}_k^b$. When the distance is large, the targets are widely separated and the association is therefore 'easy'. For that case we need less precision in estimating the *a-priori* states and therefore fewer particles could be predicted. Of course after the association the remaining particles of the chosen hypothesis will also be predicted.

For varying the number of predicted particles (N_p), a function $g(\sigma_k) = N_p(k)$ is introduced:

$$g(\sigma_k) = \begin{cases} N^{max}, & \text{if } \sigma_k < \sigma^{min}. \\ n(\sigma_k), & \text{if } \sigma^{min} < \sigma_k < \sigma^{max}. \\ N^{min}, & \text{if } \sigma_k > \sigma^{max}. \end{cases} \tag{22}$$

where

$$n(\sigma_k) = \frac{N^{min} - N^{max}}{\sigma^{max} - \sigma^{min}} \cdot (\sigma_k - \sigma^{min}) + N^{max} \tag{23}$$

We use N^{max} predicted particles when the measurement separation is smaller than σ^{min}, and N^{min} particles when the separation is bigger than σ^{max}.

Another implementation issue concerns the use of the EKFs in the prediction step of the LLPF. When predicting a particle $\mathbf{x}_{k-1}^i$ we use firstly an EKF for computing its mean and covariance one step ahead, and then we sample from the resulting density to obtain the particle $\mathbf{x}_k^i$. Since resampling eliminates small-weighted particles replacing them with others, $N_r \in [1, \dots, N_s]$ resampled particles have multiplicity $m_k^i \geq 1$. Therefore a computational efficient prediction approach is to use EKFs just for these N_r particles $\mathbf{x}_{k-1}^i$, drawing for each m_{k-1}^i samples from the predicted densities (we use thus $N_s - N_r$ fewer EKFs).

The algorithm (for the 2-target case) starts by constructing the measurement candidates with equation (9), transforming them to the Cartesian plane using (13) and computing their distance σ_k:

$$\sigma_k = |\mathbf{z}_{C,k}^a - \mathbf{z}_{C,k}^b|_2 \tag{24}$$

Using σ_k we calculate the number of the predicted particles N_p from equations (22) and (23).

We then choose randomly N_p from the N_s particles $\mathbf{x}_{k-1}^i$ and we predict them, once for each measurement candidate, using the EKFs in the way described before:

$$\begin{aligned} [\hat{\mathbf{x}}_k^{a,i}, \hat{\mathbf{P}}_k^{a,i}] &= \mathbf{EKF}[\mathbf{x}_{k-1}^i, \mathbf{P}_{k-1}^i, \mathbf{z}_k^a] \\ [\hat{\mathbf{x}}_k^{b,i}, \hat{\mathbf{P}}_k^{b,i}] &= \mathbf{EKF}[\mathbf{x}_{k-1}^i, \mathbf{P}_{k-1}^i, \mathbf{z}_k^b] \end{aligned} \tag{25}$$

where $\mathbf{P}_k^i$ is the state covariance. By sampling the resulting densities we obtain two set of particles:

$$\mathbf{x}_k^{a,i} \sim \mathcal{N}(\mathbf{x}_k^{a,i}; \hat{\mathbf{x}}_k^{a,i}, \hat{\mathbf{P}}_k^{a,i}),\ \mathbf{x}_k^{b,i} \sim \mathcal{N}(\mathbf{x}_k^{b,i}; \hat{\mathbf{x}}_k^{b,i}, \hat{\mathbf{P}}_k^{b,i}) \tag{26}$$

The two candidate *a-priori* state estimates are

$$\bar{\mathbf{x}}_k^a = \frac{1}{N_p}\sum_{i=1}^{N_p} \mathbf{x}_k^{a,i},\ \bar{\mathbf{x}}_k^b = \frac{1}{N_p}\sum_{i=1}^{N_p} \mathbf{x}_k^{b,i} \tag{27}$$

which we truncate keeping just their position elements:

$$\begin{aligned} \bar{\mathbf{x}}_{p,k}^a &= [\bar{x}_k^{a,(1)}\ \bar{y}_k^{a,(1)}\ \bar{x}_k^{a,(2)}\ \bar{y}_k^{a,(2)}\ ]^T \\ \bar{\mathbf{x}}_{p,k}^b &= [\bar{x}_k^{b,(1)}\ \bar{y}_k^{b,(1)}\ \bar{x}_k^{b,(2)}\ \bar{y}_k^{b,(2)}\ ]^T \end{aligned} \tag{28}$$

The distances d_a and d_b between the position estimates and their measurements are given by

$$d_a = |\bar{\mathbf{x}}_{p,k}^a - \mathbf{z}_{C,k}^a|_2,\ d_b = |\bar{\mathbf{x}}_{p,k}^b - \mathbf{z}_{C,k}^b|_2 \tag{29}$$

These distances determine the final association:

$$\{\mathbf{x}_k^i, \mathbf{P}_k^i, \mathbf{z}_k\}_{i=1}^{N_p} = \begin{cases} \{\mathbf{x}_k^{a,i}, \mathbf{P}_k^{a,i}, \mathbf{z}_k^a\}_{i=1}^{N_p}, & \text{if } d_a < d_b. \\ \{\mathbf{x}_k^{b,i}, \mathbf{P}_k^{b,i}, \mathbf{z}_k^b\}_{i=1}^{N_p}, & \text{if } d_a > d_b. \end{cases} \tag{30}$$

The remaining $N_s - N_p$ particles are predicted with equations (25) and (26) using the associated $\mathbf{z}_k$.

We continue by assigning a normalised weight w_k^i to each particle $\mathbf{x}_k^i$ using equations (16) and (17). The posterior state estimate $\hat{\mathbf{x}}_k$ is obtained from the weighted sum of the particles (18). We resample using equations (19)-(21) assigning to each resampled particle its associated covariance.

This algorithm for convenience will be called adaptive multitarget LLPF (A-MLLPF). In the simulations for comparison we also implemented an algorithmic variation without the adaptive scheme (MLLPF).

4. SIMULATION RESULTS

In the studied scenario a static radar monitored two targets. The radar update rate was 1sec, the measurement noise was 2^o and 1.2m and the track process noise was 0.9m and 0.2m/s (for both x and y axes). For tracking we used the SIR, the LLPF and the adLLPF, all employing 100 particles. We also set $\sigma^{min} = 20$m, $\sigma^{max} = 40$m, $N^{min} = 5$ and $N^{max} = 100$. We created 800 different crossing tracks (figure 2) each lasting 120sec. The initial x-axis target distance varied uniformly from 10m to 250m. For every track we performed 50 Monte Carlo simulations, each with different measurements. We evaluated the mean values for every track and then we average them to obtain the overall simulation results.

As we can see from table 1, both proposed algorithms MLLPF and A-MLLPF had about 34% fewer disassociations than the MSIR, 40% smaller RMS position error and 11%-16% less track swaps. Comparing the suboptimal A-MLLPF with the MLLPF, we see that the degradation on performance is small: 4%-5% for the disassociation and

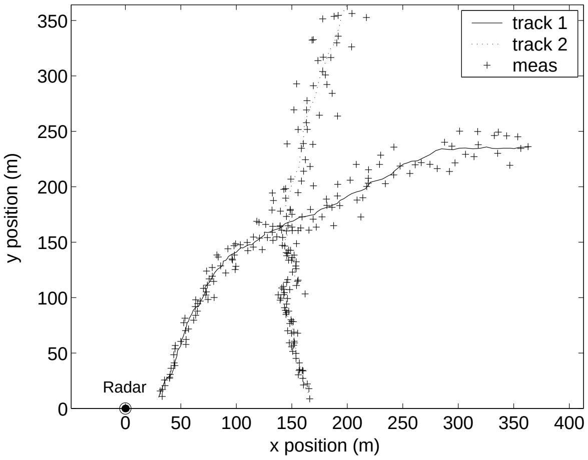

Fig. 2. An example of two crossing tracks. The initial targets' distance is 130m.

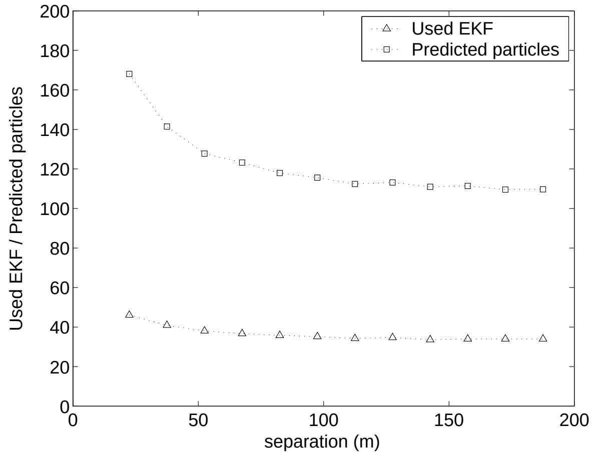

Fig. 3. Number of predicted particles and EKFs over average track measurement separation, after 800 tracks x 50 runs.

track swaps, and negligible (less than 0.4%) for the RMS error. The main advantage of the A-MLLPF is of course its computational efficiency, having on average about 113 particles predicted and using just 35 EKFs (figure 3).

5. SUMMARY

This work introduced a novel multitarget algorithm which uses as its basis the LLPF. The algorithm employs an adaptive mechanism which varies the computational load according to the difficulty of the measurements' association. Compared with the equivalent multitarget SIR filter, the proposed algorithm gives on average fewer disassociations and significantly smaller RMS position error. Future work includes using gating techniques before the association for decreasing the computational load and developing features for clutter rejection and for handling a varying number of targets.

6. REFERENCES

[1] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proc.-F*, vol. 140, no. 2, 1993.

[2] K. Fukunaga and T.E. Flick, "An optimal Global Nearest Neighbor metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 3, pp. 314–318, May 1984.

[3] T.E. Fortmann et al, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. OE-8, pp. 173–184, July 1983.

[4] C. Kreucher, K. Kastella, and A.O. Hero III, "Tracking mutliple targets using a particle filter representation of the Joint Multitarget Probability Density," in *SPIE Annual Meeting*, San Diego, California, August 2003.

[5] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. 6th International Conference in Information Fusion*, Australia, July 2003.

[6] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[7] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, 1999.

[8] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications*, Artech House, Norwood, 2004.

	MSIR	MLLPF	A-MLLPF
disassociations	0.80601	0.52324	0.54563
track swaps	0.05906	0.04980	0.05234
RMSE (m)	42.1578	25.4899	25.6264
predicted particles	-	200	113.429
EKFs	-	200	34.6048

Table 1. Simulation results, averaged per track after 800 tracks x 50 runs.