Digital Signal Processing 20 (2010) 561–574

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Interacting MCMC particle filter for tracking maneuvering target

Liu Jing, Prahlad Vadakkepat *

Department of Electrical and Computer Engineering, 4 Engineering Drive 3, National University of Singapore, Singapore 117576

A R T I C L E I N F O

Article history:
Available online 7 August 2009

Keywords:
Particle filter
Markov chain Monte Carlo
Particle swarm
Maneuvering target tracking

A B S T R A C T

In this paper, a new method, named interacting MCMC particle filter, is proposed to track maneuvering target. The particles are sampled from the target posterior distribution via direct interacting MCMC sampling method, which avoids sample impoverishment and increases the robustness of the algorithm. Moreover, the interacting MCMC particle filter algorithm accelerates the MCMC convergence rate via propagating each particle based on both its history information and the information from other particles.

1. Introduction

In the history of development of maneuvering target tracking techniques, single model based adaptive Kalman filtering came into existence first [1–3]. Aidala [1] proposed the adaptive Kalman filtering method based on single motion model of the moving target in 1973. In the proposed method, the target maneuvering is estimated by adjusting the Kalman gain.

Decision-based techniques, which detect the manoeuvre and then cope with it effectively, appeared next. Examples of this approach include the input estimation (IE) techniques [4,5], the variable dimension (VD) filter [6], the two-stage Kalman estimator [7], etc. In addition to basic filtering computation, these techniques require additional effort to detect the target maneuvers.

The decision based techniques are followed by multiple-model algorithms, which describe the motion of a target using multiple sub-filters. The generalized pseudo-Bayesian (GPB) method [8], the interacting multiple model (IMM) method [9, 10], and the adaptive interacting multiple model (AIMM) method [11] are included in this kind of approach. Using the multiple model based methods which use more than one model to describe the motion of the target, performance is enhanced. Among them, interact multiple model algorithm (IMM) is the most common one. The main feature of the IMM algorithm is its ability to estimate the state of a dynamic system with several behavior modes which can "switch" from one to another. In particular, the IMM estimator can be a self-adjusting variable-bandwidth filter, which makes it natural for tracking maneuvering targets.

Remark. In the above methods, the data models are assumed Gaussian and weakly nonlinear, and the Kalman filter/extended Kalman filter (KF/EKF) is used to perform target state estimation. More recently, nonlinear filtering techniques have been gaining more attention. The most common one among them is the particle filter, a state estimate pdf sampling based algorithm, which is more capable of dealing with nonlinear and non-Gaussian system estimation problems.

Particle filter, which uses sequential Monte Carlo methods for on-line learning within a Bayesian framework, can be applied to any state-space models.

* Corresponding author. Fax: +65 6779 1103.
E-mail addresses: elelj20080730@gmail.com (L. Jing), elepv@nus.edu.sg (P. Vadakkepat).

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2009.08.011

The application of particle filter in maneuvering target tracking has been paid attention only in recent years. Several approaches, which use multiple models to describe the changing maneuvering model, have been proposed in the particle filter framework. One of the methods is based on the auxiliary particle filter. In [12], Karlsson used an auxiliary particle filter to track a highly maneuvering target. In this method, each particle is split deterministically into a number of possible maneuver hypotheses with each hypothesis corresponding to a specific model.

Other methods focus on how to switch between different motion models. In [13], Bayesian switching structure is chosen as the principle which determines switching between different models. A set of models are utilized to cope with the unknown maneuver. Moreover, to deal with non-Gaussian noise, Cauchy distribution is used as the system noise distribution. In [14] and [15], the maneuvering target tracking system is treated as a jump Markov linear system. *The MCMC process* is used as the selection scheme to choose the motion model from a set of candidate models at some specific time step.

However, in the above approaches, the possible motion models and transition probability matrices are assumed as known. In practice, *the dynamics* are hard to break up into several different motion models and the model transition probabilities are difficult to obtain. A general model is needed to cope with the wide variety of motions exhibited by the maneuvering target.

For the single model based methods, Karlsson [16] and Ikoma [17] applied optimal recursive Bayesian filters directly to the nonlinear target model. The algorithm may fail due to sample impoverishment when tracking wide variation in maneuvering movements.

The target's state variables, such as the position and velocity, vary quickly and are not restricted to a fixed dynamic model when it performs maneuvering movements. New features of posterior distribution of the target state are encountered during the tracking process. The standard particle filter can not cope with the new feature of the posterior distribution since it provides no opportunity to generate new values for unknown quantities after their initial generation. Consequently, as the posterior distribution drifts away from this initial values, the particle base may degenerate to contain few distinct values of these variables. As a result, most of the particles are assigned with low weights and eliminated via the resampling process. This leads to serious sample impoverishment and then the tracking process fails.

There have been some systematic techniques proposed recently to solve the problem of sample impoverishment. One such technique is regularized particle filter [18], which resamples from a continues approximation of the posterior density $p(x_k|z_{1:k})$, whereas the standard particle filter resamples from the discrete approximation of the posterior density. This approach is frequently found to improve performance, despite a less rigorous derivation. An alternative solution to the same problem is the resample-move algorithm [19]. This technique uses periodic MCMC steps to diversify particles in an importance sampling-based particle filter. It does so in a rigorous manner that ensures the particles asymptotically approximate samples from the posterior. However, the resample-move algorithm can not avoid sample impoverishment due to the existence of resampling procedure at each time step. Moreover, the traditional MCMC sampling needs a lot of iterations to converge to the target posterior distribution, which is very slow and not suitable for real-time tracking.

In this paper, a new method, named interacting MCMC particle filter, is proposed to handle the sample impoverishment problem. The particles are sampled from the target posterior distribution via direct interacting MCMC sampling method, which avoids sample impoverishment effectively.

The proposed algorithm also accelerates the MCMC convergence rate. In the standard MCMC based particle filter method, each particle is propagated independently, however, neglecting the information from other particles. It should be easier and faster to reach the high posterior density area if more information is incorporated. Inspired by the particle swarm algorithm, which is to find optimal regions of complex search spaces through the interaction of individuals in a population of particles, the proposed algorithm propagates each particle based on both its history information and the information from other particles. The proposed algorithm is named interacting MCMC particle filter since it incorporates the interaction of the particles in contrast with the traditional MCMC based particle filter.

Remark. It is well known that all numerical integration approaches perform better when correlation among the components is low [27].

In particular, MCMC algorithm converges rapidly. In the interacting MCMC particle filter method, at each time step the introduction of the interaction of particles reduces the correlation among one particle's history states, which speeds up the MCMC convergence rate.

The rest of the sections are organized as follows, firstly, the basic theory of particle filter and Markov chain Monte Carlo are introduced in Section 2. The interacting MCMC particle filter is presented in Section 3. In Section 4 the proposed algorithm is compared with the resample-move algorithm in two synthetic problems where one maneuvering target is tracked. The paper is summarized in Section 5.

2. Problem formulation

To define the problem of tracking, consider a dynamic system represented by the state sequence $\{x_k,\ k \in \aleph\}$, whose temporal evolution is provided by the state equation:

$$x_k = f(x_{k-1}, v_{k-1}), \tag{1}$$

where f is a nonlinear function and $\aleph$ is the set of natural numbers. $\{v_{k-1},\ k \in \aleph\}$ is the process noise sequence. The objective of tracking is to recursively estimate x_k from a sequence of measurements $\{z_k,\ k \in \aleph\}$,

$$z_k = h(x_k, n_k), \tag{2}$$

where h is a nonlinear function. $\{n_k,\ k \in \aleph\}$ is a sequence of observation noises. From the Bayesian perspective, the tracking problem is to recursively calculate the posterior probability density function (pdf), $p(x_k|z_{1:k})$, where $z_{1:k} = \{z_t,\ t = 1, \ldots, k\}$.

2.1. Basic theory of particle filter

In this paper, *particle filters* are considered to solve the state estimation problem due to its ability in tackling nonlinear and non-Gaussian problems. The posterior pdf, $p(x_k|z_{1:k})$, is approximated by a set of particles with associated weights. The detailed particle filter algorithm is described in the following.

(i) Initialization. Sample initial particles $\{x_0^i,\ i = 1, \ldots, NP\}$ from the prior distribution $p(x_0)$ and set the weights w_0^i to $\frac{1}{NP}$, $i = 1, \ldots, NP$. NP is the number of particles.

(ii) Prediction. Particles at time step $k-1$, $\{x_{k-1}^i,\ i = 1, \ldots, NP\}$, are passed through the system model (1) to obtain the predicted particles at time step k, $\{\hat{x}_k^i,\ i = 1, \ldots, NP\}$:

$$\hat{x}_k^i = f\left(x_{k-1}^i, v_{k-1}^i\right), \tag{3}$$

where v_{k-1}^i is a sample drawn from the probability density function of the system noise $p_v(v)$.

(iii) Update. Once the observation data, z_k, is measured, evaluate the importance weight of each predicted particle (4) and obtain the normalized weight for each particle (5):

$$\tilde{w}_k^i = p\left(z_k|\hat{x}_k^i\right), \tag{4}$$

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{NP} \tilde{w}_k^i}. \tag{5}$$

Thus define a discrete distribution $\{w_k^i\colon\ i = 1, \ldots, NP\}$ over $\{\hat{x}_k^i\colon\ i = 1, \ldots, NP\}$, with probability mass w_k^i associated with element $\hat{x}_k^i$ at time step k. The approximate posterior distribution, $\hat{p}(x_k|z_{1:k})$, can be estimated as

$$\hat{p}(x_k|z_{1:k}) = \sum_{i=1}^{NP} w_k^i \delta\left(x_k - \hat{x}_k^i\right). \tag{6}$$

(iv) Resample. Resample the discrete distribution $\{w_k^i\colon\ i = 1, \ldots, NP\}$ NP times to generate particles $\{x_k^j\colon\ j = 1, \ldots, NP\}$, so that for any j, $Pr\{x_k^j = \hat{x}_k^i\} = w_k^i$. Set the weights w_k^i to $\frac{1}{NP}$, $i = 1, \ldots, NP$, and move to stage (ii).

The resampling step reduces the effects of degeneracy [20], however, it introduces other practical problems. The particles that have high weights are statistically selected many times. This leads to loss of diversity among the particles as the resultant samples will contain many repeated points which results in sample impoverishment. Sample impoverishment leads to failure in tracking since less diverse particles are used to represent the uncertain dynamics of the moving object. Especially, when tracking a maneuvering target, whose position, velocity and acceleration vary quickly, serious sample impoverishment occurs (all particles collapse to a single point within a few iterations) and the tracking algorithm fails. An effective method is required to handle the sample impoverishment problem associated with maneuvering target tracking process.

2.2. Basic theory of Markov chain Monte Carlo process

Let M be a finite set and suppose that $\pi(m)$ is a probability distribution on M from which we wish to draw samples. The Metropolis–Hastings (MH) algorithm [21] is a widely used procedure for drawing approximate samples from π, which works by finding a Markov chain on M with π as stationary distribution and using the well-known fact that after running the chain for a long time it is approximately distributed as π. The pseudocode for the MH algorithm in this context is as follows [22].

Metropolis–Hastings algorithm. Start with an arbitrary initial configuration m_0, then iterate for $\tau = 0, \ldots, B + J$, where B is the length of burn-in period and J is the number of MCMC iterations.

(1) Propose a new assignment m' by sampling from the proposal density function $Q(m'; m_\tau)$.

(2) Sample $\rho \sim U(0,1)$, where $U(0,1)$ is a uniform distribution in the interval $(0,1)$.

(3) Calculate the acceptance ratio,

$$a = \frac{\pi(m')}{\pi(m_\tau)} \frac{Q(m_\tau; m')}{Q(m'; m_\tau)}. \tag{7}$$

(4) If $\rho \leqslant \min\{1, a\}$, then accept move:

$$m_{\tau+1} = m', \tag{8}$$

else reject move:

$$m_{\tau+1} = m_\tau. \tag{9}$$

It is a standard practice to discard a number of initial "burn-in" samples, say B of them, to allow the MH algorithm to converge to a stationary distribution. In the proposed algorithm, the target distribution $\pi(m)$ is chosen as the approximate posterior distribution $\hat{p}(x_k|z_{1:k})$ and at each time step k, the MH algorithm is used to generate a set of samples from $\hat{p}(x_k|z_{1:k})$.

3. Interacting MCMC particle filter

The resample-move method [19] diversifies particles after resampling, which can reduce sample impoverishment but could not avoid it absolutely. The introduction of MCMC to improve importance sampling suggests that MCMC alone could be used to obtain a particle filter that can effectively handle the sample impoverishment problem.

For the traditional MCMC sampling method [19], it is very slow to converge to the target posterior distribution. During the MCMC move iterations, each particle is propagated based on its previous states and the trajectory of each particle is developed independently, neglecting the information from other particles. It is reasonable that incorporating the neighborhood particles' information accelerates the convergence of the MCMC algorithm. For example, when several people search for a piece of gold in a wide area concurrently, each person looks for the gold depending on not only his previous experience but also the information from his partners (where the gold may appear most probably), and adjust the search strategy accordingly. The particle swarm algorithm finds optimal regions of complex search spaces through the interaction of individuals in a population of particles. In the proposed interacting MCMC sampling algorithm, during the MCMC iterations, each particle is propagated based on not only the past information but the information from other particles as well. The proposed algorithm is named as interacting MCMC particle filter since it incorporates the interaction of the particles in contrast with the traditional MCMC based particle filter.

It is well known that all numerical integration approaches perform better when correlation among the components is low. In particular, MCMC algorithm converges rapidly. In the interacting MCMC particle filter method, at each time step the introduction of the interaction of particles reduces the correlation among one particle's history states, which speeds up the MCMC convergence rate.

The basic theory of particle swarm algorithm is introduced in Section 3.1, and the proposed interacting MCMC particle filter algorithm is presented in Section 3.2.

3.1. Particle swarm algorithm

Particle swarm adaptation has been shown to successfully optimize a wide range of functions [23–25]. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of individual vectors, called "particles" as they are conceptualized as moving points in multidimensional space. Each particle is drawn stochastically toward the position of its own previous best performance, pB_i, and the position of the best previous performance of its neighbors, gB, where i denotes the ith particle. The algorithm in pseudocode follows [26].

Particle swarm algorithm.

(1) Initialize population.

(2) Do,
for $i = 1$ to Population Size,

$$V_i = V_i + \varphi_1(pB_i - S_i) + \varphi_2(gB - S_i), \tag{10}$$

$$S_i = S_i + V_i. \tag{11}$$

Next i
Until termination criterion is met.

S_i represents the state vector of the ith particle, and V_i denotes the velocity of S_i. The variables φ_1 and φ_2 are random positive numbers, drawn from a uniform distribution and defined by an upper limit $\varphi_{\max}$, which is a parameter of the system.

3.2. Interacting MCMC particle filter algorithm

Here a different Monte Carlo approximation of the target posterior distribution, in terms of unweighted samples, is proposed based on the interacting MCMC sampling method. In particular, the posterior distribution $p(x_{k-1}|z_{1:k-1})$ at time $k-1$ is represented as a set of NP unweighted samples $p(x_{k-1}|z_{1:k-1}) \approx \{x_{k-1}^i\}_{i=1}^{NP}$. According to the Bayesian theory, the posterior filtering distribution at time step k, $p(x_k|z_{1:k})$, can be represented as

$$p(x_k|z_{1:k}) \approx c \cdot p(z_k|x_k) \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1})\, dx_{k-1}, \tag{12}$$

where c is a constant. Instead of importance sampling, interacting MCMC is used to sample from (12) at each time step. The sampling procedure results in an unweighted particle approximation for the posterior distribution $p(x_k|z_{1:k}) \approx \{x_k^i\}_{i=1}^{NP}$.

In the proposed interacting MCMC particle filter algorithm, at each time step, the particles are propagated based on the dynamic model to obtain the predicted particles. The predicted particles are then chosen as the starting points in the following interacting MCMC sampling procedure. During the interacting MCMC iterations, the proposal function based on the particle swarm algorithm is used to generate a set of samples from the target posterior distribution. The algorithm steps are listed in the following.

Interacting MCMC particle filter algorithm.

(i) Initialization. Sample x_0^i from the prior distribution $p(x_0)$ and set the weights w_0^i to $\frac{1}{NP}$, $i = 1, \ldots, NP$.

(ii) Prediction. Each particle is passed through the system model to obtain the predicted particles:

$$\hat{x}_k^i = f\big(x_{k-1}^i, v_{k-1}^i\big), \tag{13}$$

where v_{k-1}^i is a sample drawn from the probability density function of the system noise $p_v(v)$.

(iii) Update. Once the observation data, z_k, is measured, evaluate the importance weight of each predicted particle in (14) and obtain the normalized weight for each particle as in (15),

$$\tilde{w}_k^i = w_{k-1}^i p\big(z_k|\hat{x}_k^i\big), \tag{14}$$

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}. \tag{15}$$

Thus define a discrete distribution $\{w_k^i\colon\ i = 1, \ldots, NP\}$ over $\{\hat{x}_k^i\colon\ i = 1, \ldots, NP\}$, with probability mass w_k^i associated with element $\hat{x}_k^i$ at time step k. The approximate posterior distribution, $\hat{p}(x_k|z_{1:k})$, can be estimated as

$$\hat{p}(x_k|z_{1:k}) = \sum_{i=1}^{NP} w_k^i \delta\big(x - \hat{x}_k^i\big). \tag{16}$$

(iv) Interacting MCMC move. For each predicted particle $\hat{x}_k^i$, $i = 1, \ldots, NP$, repeat $B + J$ times (B is the length of burn-in period and J is the number of MCMC iterations):

(1) Initialization, set

$$\tau = 0, \tag{17}$$

$$\chi_{k,0}^i = \hat{x}_k^i, \tag{18}$$

$$V_{k,0}^i = 0, \tag{19}$$

$$\xi_{k,0}^i = w_k^i, \tag{20}$$

where τ denotes the τth MCMC iteration.

(2) Search among the weights $\{\xi_{k,\tau}^i,\ i = 1, \ldots, NP\}$ to obtain the largest weight, and identify the particle corresponding to the largest weight, $gB_{k,\tau}$. For each specific particle $\chi_{k,\tau}^i$, search among its history weights, $\{\xi_{k,\lambda}^i,\ \lambda = 0, \ldots, \tau\}$ and obtain the particle with the largest weight, $pB_{k,\tau}^i$.

(3) For each particle $\chi^i_{k,\tau}$, calculate its velocity $V^i_{k,\tau+1}$ (21), and then propagate it to the next position (22),

$$V^i_{k,\tau+1} = V^i_{k,\tau} + \varphi_1 \times \left(gB_{k,\tau} - \chi^i_{k,\tau}\right) + \varphi_1 \times \left(pB^i_{k,\tau} - \chi^i_{k,\tau}\right), \tag{21}$$

$$\hat{\chi}^i_{k,\tau+1} = \chi^i_{k,\tau} + V^i_{k,\tau+1}. \tag{22}$$

(4) Sample $\rho \sim U(0,1)$, where $U(0,1)$ is a uniform distribution in the interval $(0,1)$.
(5) Calculate the acceptance ratio,

$$a = \frac{\hat{p}(\hat{\chi}^i_{k,\tau+1}|z_{1:k})}{\hat{p}(\chi^i_{k,\tau}|z_{1:k})} \frac{Q(\chi^i_{k,\tau}; \hat{\chi}^i_{k,\tau+1})}{Q(\hat{\chi}^i_{k,\tau+1}; \chi^i_{k,\tau})}. \tag{23}$$

(6) If $\rho \leqslant \min\{1, a\}$, then accept move:

$$\chi^i_{k,\tau+1} = \hat{\chi}^i_{k,\tau+1}, \tag{24}$$

else reject move:

$$\chi^i_{k,\tau+1} = \chi^i_{k,\tau}. \tag{25}$$

(7) Calculate the weight of each particle:

$$\xi^i_{k,\tau+1} = p\left(z_k|\chi^i_{k,\tau+1}\right). \tag{26}$$

(8) $\tau = \tau + 1$, then move to step (2).

Finally, obtain the resampled particles,

$$x^i_k = \chi^i_{k,B+M}, \quad i = 1, \ldots, NP, \tag{27}$$

and set the weights w^i_k to $\frac{1}{NP}$, $i = 1, \ldots, NP$, then move to the prediction stage (ii).

4. Simulation results and analysis

The conventional MCMC based particle filter (resample-move algorithm) and interacting MCMC particle filter are compared in the following two examples: a robot equipped with sonar tracks a maneuvering target and a radar tracks an aircraft performing coordinated turn. In the first example, the dynamic model of the maneuvering target is represented as

$$X_k = \Phi X_{k-1} + \Gamma\left[a_{k-1} + m_{k-1}(s,t)\right], \tag{28}$$

where $X_k = [px, vx, py, vy]^T_k$ is the state vector; px and vx are respectively the position and velocity of the moving object along the Cartesian frame x axis; and, py, vy along the y axis. $m_{k-1}(s,t)$ is the maneuver-induced acceleration. s and t are the start and end times of the maneuver. a_{k-1} accounts for the random acceleration of the target. The transition matrix Φ represents the dynamic characteristics of the target. Γ is a unity matrix.

The robot installed with one sonar sensor is positioned at the origin of the plane. The measurement equation is as follows:

$$Z_k = h(X_k) + n_k, \tag{29}$$

where $Z_k = [z_1, z_2]_k$ is the observation vector. z_1 is the distance between the robot and moving object, and z_2 is the bearing angle. The measurement noise $n_k = [n_{z_1}, n_{z_2}]_k$ is a zero mean Gaussian white noise process with variance R: $E[n_k\, n_j] = R\delta_{kj}$, where

$$R = \begin{bmatrix} \sigma^2_{z_1} & 0 \\ 0 & \sigma^2_{z_2} \end{bmatrix}. \tag{30}$$

Eq. (29) is expanded as follows:

$$z_{1,k} = \sqrt{(px_k - x_R)^2 + (py_k - y_R)^2} + n_{z_1,k}, \tag{31}$$

$$z_{2,k} = \tan^{-1}\left(\frac{py_k - y_R}{px_k - x_R}\right) + n_{z_2,k}. \tag{32}$$

Eq. (31) describes the changing distance between the robot and moving object. (x_R, y_R) is the position of the robot in Cartesian coordinates and the robot is assumed to be at the origin of the plane. Eq. (32) describes the object's changing bearing angle.

Table 1
Simulation parameters.

Simulation parameter	Value
Number of particles	200
The variance matrix Q of process noise	$\mathrm{diag}\{1\ 1\ 1\ 1\}_{4\times 4}$
The variance matrix R of observation noise	$\mathrm{diag}\{0.1\ 0.01\}_{2\times 2}$
$\varphi_{\max}$ (the upper limit of parameters φ_1 and φ_2 in particle swarm)	2

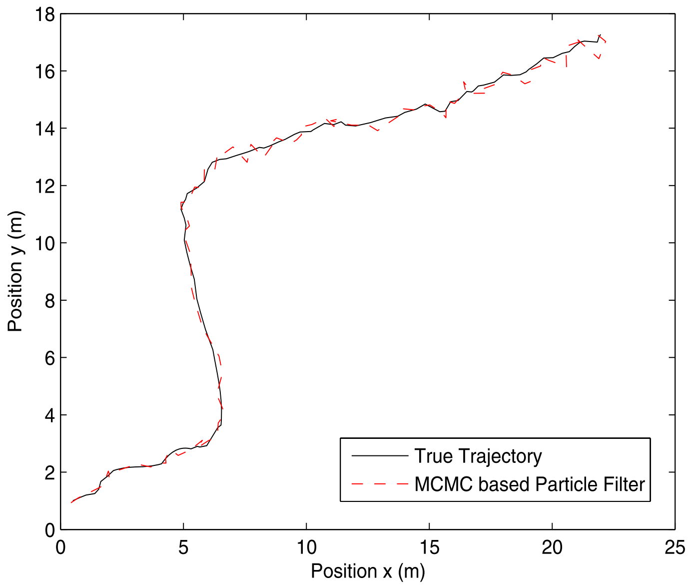


Fig. 1. MCMC based particle filter: tracking trajectory of ground target.

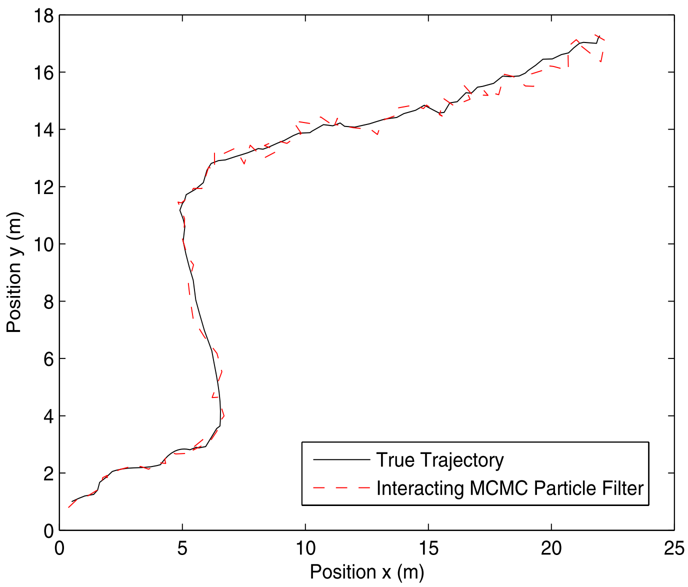


Fig. 2. Interacting MCMC particle filter: tracking trajectory of ground target.

Remark. In this example, the target considered executes a three-leg maneuvering sequence: constant velocity, left turn, right turn, and constant velocity.

It is assumed that the dynamic model of the maneuvering target (28) is unknown and a simple motion model (33) is adopted in the two algorithms that are being compared. The simulation parameters are listed in Table 1;

Table 2
Performance comparison for robot tracking case.

	MCMC iteration number	RMSE (m)	TLR (%)	ET (s)
MCMC based PF	5	0.4243	52	0.15
Interacting MCMC PF	5	0.3318	0	0.2

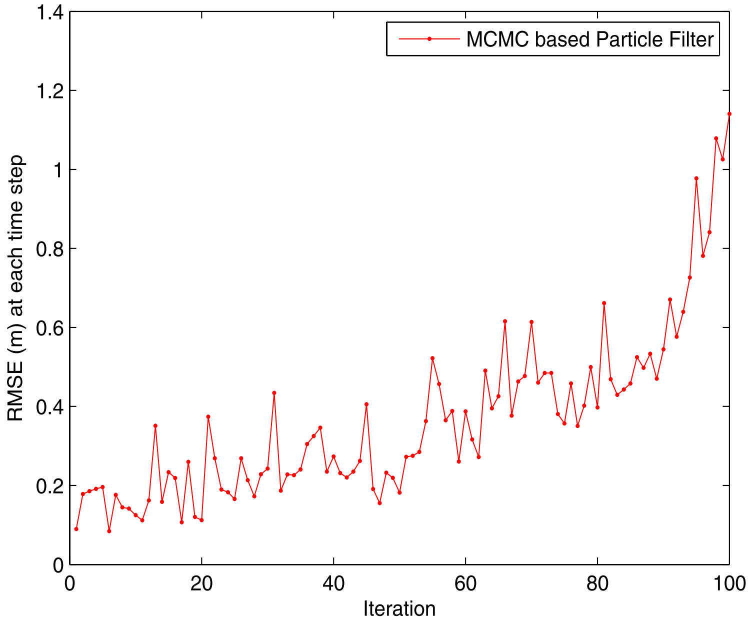


Fig. 3. MCMC based particle filter for ground target tracking: RMSE at each time step.

$$X_k = \Phi X_{k-1} + v_{k-1}. \tag{33}$$

The tracking trajectory of one successful realization performed by the two algorithms respectively are shown in Figs. 1 and 2. One hundred Monte Carlo simulations are carried out. The simulation comparison between the two algorithms is presented as Root Mean Square Error (RMSE) in position (Table 2). It can be seen that the interacting MCMC particle filter gained superior tracking performance than the MCMC based particle filter at the same MCMC iteration number. Also the RMSE for each time is presented in Figs. 3 and 4. The definition of RMSEs can be found in [12].

Remark. The performance of the methods is also compared via the tracking loss rate (TLR) and the executing time (ET), which are listed in Table 2. The tracking loss rate (TLR) is defined as the ratio of the number of simulations, in which the target is lost in track, to the total number of simulations carried out. The executing time (ET) is the CPU time needed to execute one time step in MATLAB 7.1 on a 3 GHz (Mobile) Pentium IV operating under Windows 2000.

From Table 2, it can be seen that the interacting MCMC particle filter (with tracking loss rate 0%) is robust than the MCMC based particle filter (with tracking loss rate 52%). This verifies that the interacting MCMC particle filter is capable of avoiding sample impoverishment, which increases the robustness of the algorithm. Figs. 5–8 show the failure tracking process performed by MCMC based particle filter algorithm. When sample impoverishment began at time step 50 (Fig. 8, all the particles were assigned with zero weights), the x and y tracking trajectory started to diverge at time step 50 (Figs. 6 and 7). It can be concluded that the failure of the tracking is due to sample impoverishment.

The example above is basically designed for tracking systems in which the filters are uncoupled such that, for instance, tracking in the x and y directions are independent. In reality, typical target maneuvers, such as an aircraft performing a coordinated turn, produce motion that is highly correlated across the tracking directions. In the second example, a track-while-scan (TWS) radar tracking an aircraft performing coordinated turn is studied. A coordinated turn model is:

$$px_{k+1} = px_k + \frac{\sin\omega T}{\omega} vx_k - \frac{1-\cos\omega T}{\omega} vy_k, \tag{34}$$

$$py_{k+1} = py_k + \frac{1-\cos\omega T}{\omega} vx_k + \frac{\sin\omega T}{\omega} vy_k, \tag{35}$$

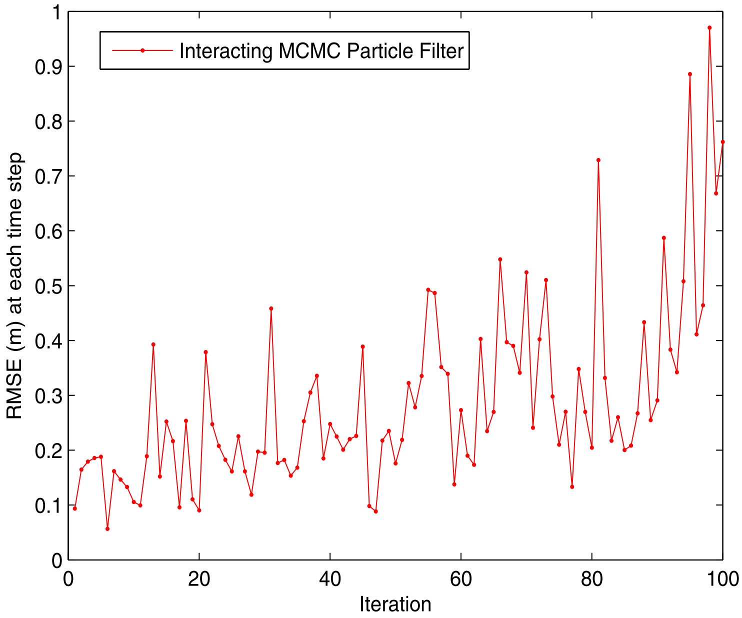


Fig. 4. Interacting MCMC particle filter for ground target tracking: RMSE at each time step.

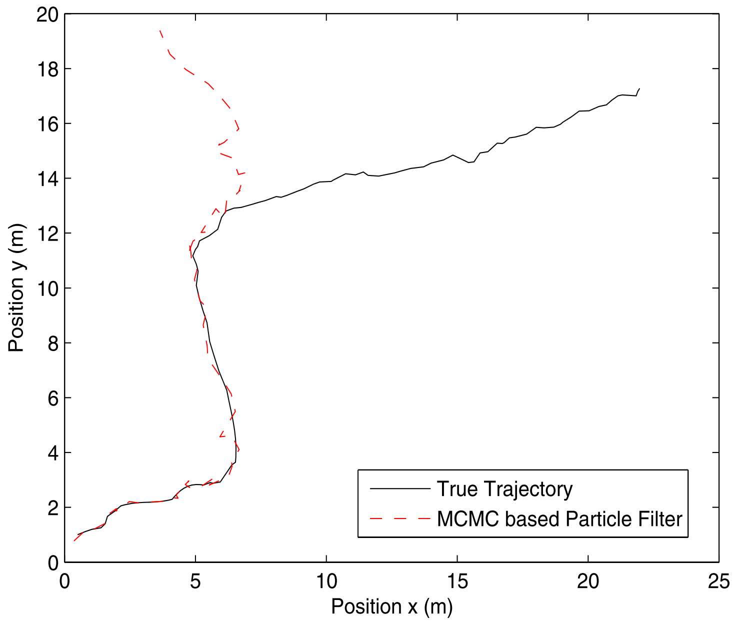


Fig. 5. MCMC based particle filter: failure tracking trajectory.

$$vx_{k+1} = vx_k \cos\omega T - vy_k \sin\omega T, \tag{36}$$

$$vy_{k+1} = vx_k \sin\omega T + vy_k \cos\omega T, \tag{37}$$

where ω denotes the turn rate in radians per second and T denotes the sampling interval.

The measurement equations of the TWS radar are similar to those used in the first example.

In this example, an aircraft executing a three leg maneuvering coordinated turn: constant velocity, $3 \times g$ turn, $-3 \times g$ turn, and finally at constant velocity, is considered. The initial velocity is 75 m/s. No models and transition probability matrix are assumed.

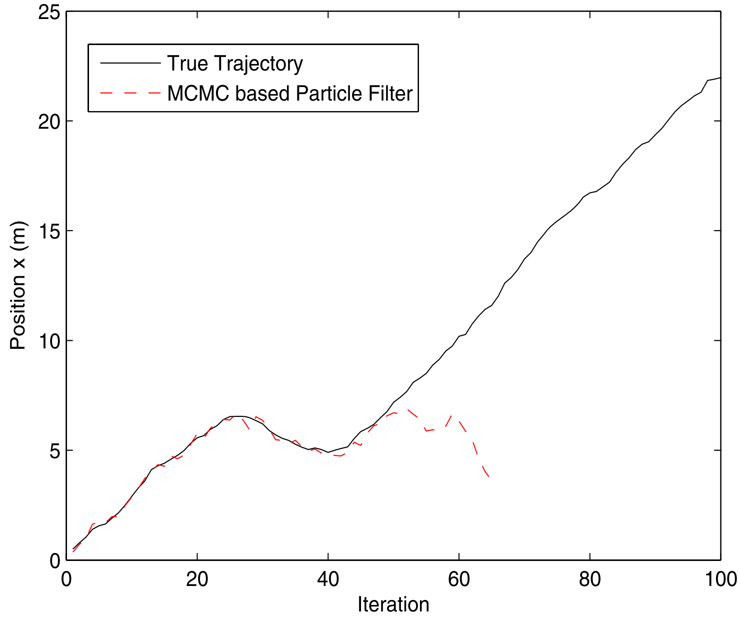


Fig. 6. MCMC based particle filter: failure tracking trajectory (position x).

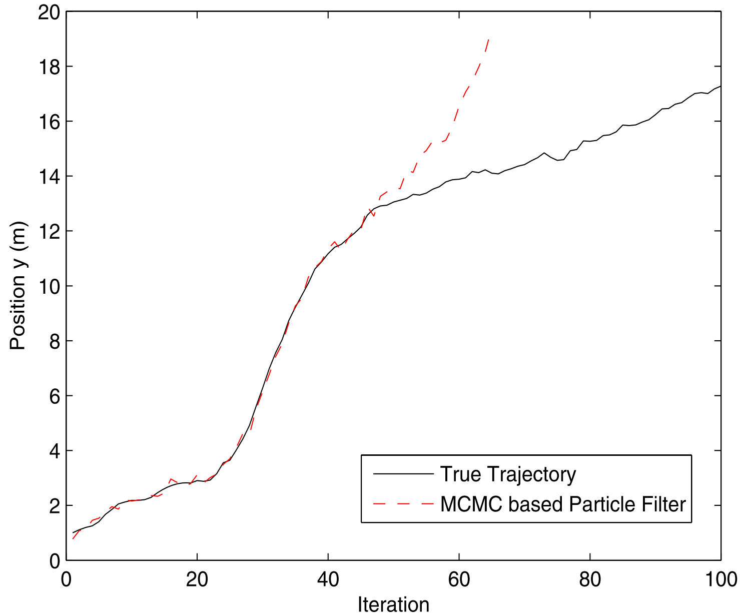


Fig. 7. MCMC based particle filter: failure tracking trajectory (position y).

Table 3
Performance comparison for aircraft tracking case.

	MCMC iteration number	RMSE (m)	TLR (%)	ET (s)
MCMC based PF	5	34.5923	37	0.18
Interacting MCMC PF	5	18.0255	0	0.3

The tracking trajectory of the maneuvering aircraft performed by the two algorithms respectively are shown in Figs. 9 and 10. One hundred Monte Carlo simulations are carried out. The simulation comparison between the two algorithms is presented in Table 3. Also the RMSE for each time step is presented in Figs. 11 and 12. From Table 3, it can be observed

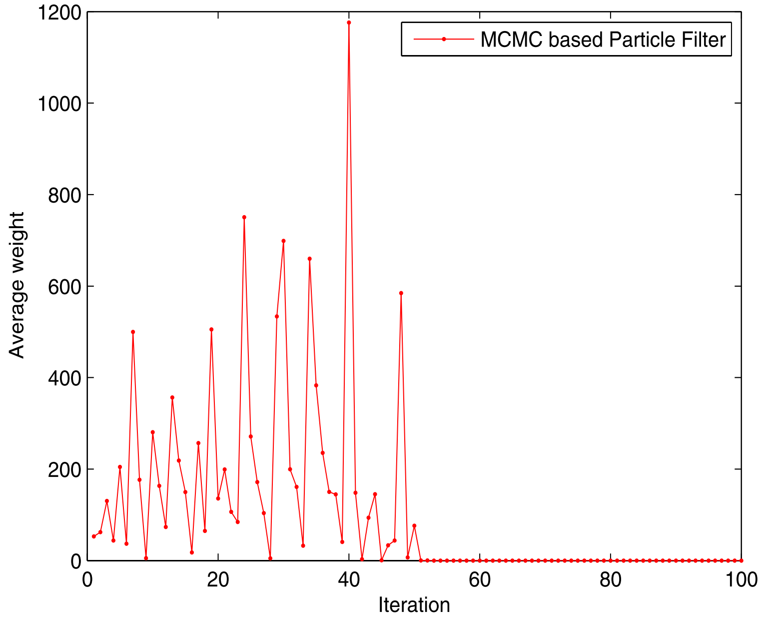


Fig. 8. MCMC based particle filter: failure tracking (weight).

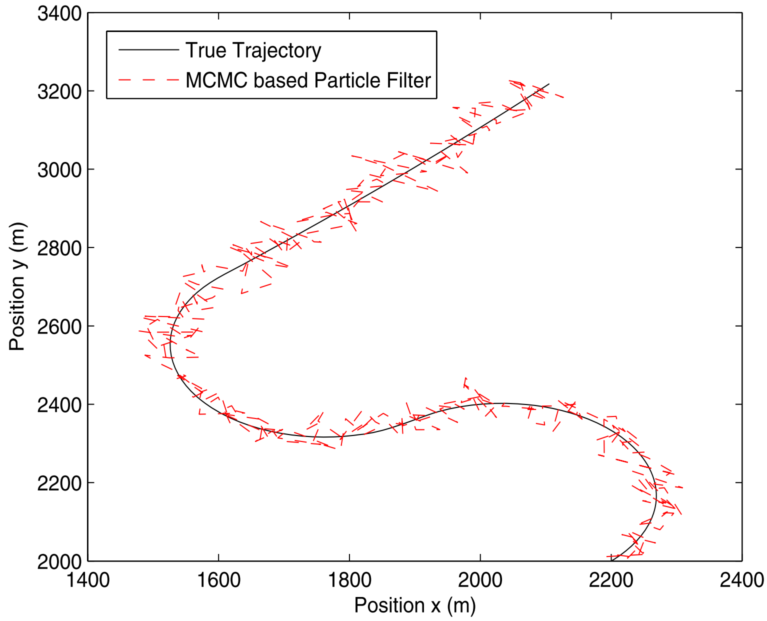


Fig. 9. MCMC based particle filter: tracking trajectory of an aircraft.

that the interacting MCMC particle filter gained superior tracking performance than the MCMC based particle filter both in accuracy and robustness.

5. Conclusions

Particle filter algorithm is used widely in the application of maneuvering target tracking. However, the standard particle filter may fail due to sample impoverishment when tracking wide variation in maneuvering movements. In this paper, a new method, named as interacting MCMC particle filter, is proposed to handle the sample impoverishment problem. The particles are sampled from the target posterior distribution via direct interacting MCMC sampling method, which avoids sample impoverishment and increases the robustness of the algorithm. Moreover, the interacting MCMC particle filter algorithm

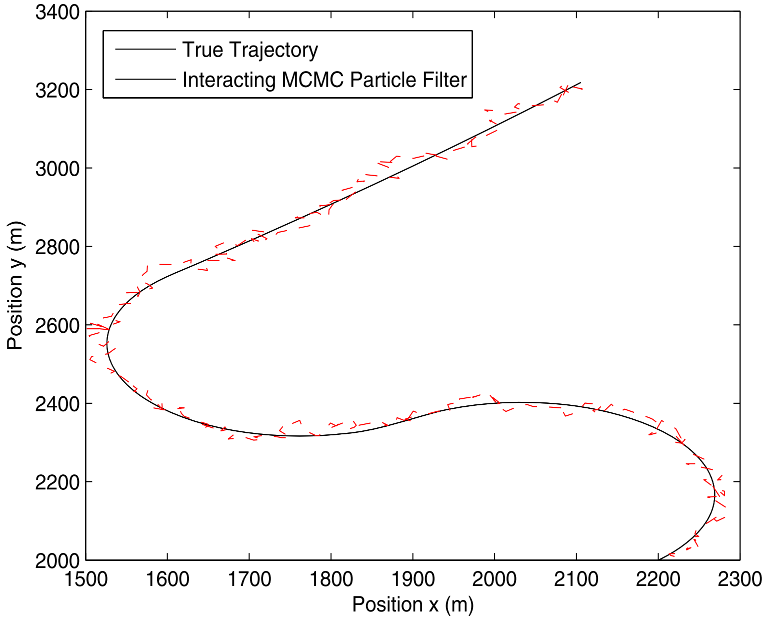


Fig. 10. Interacting MCMC particle filter: tracking trajectory of an aircraft.

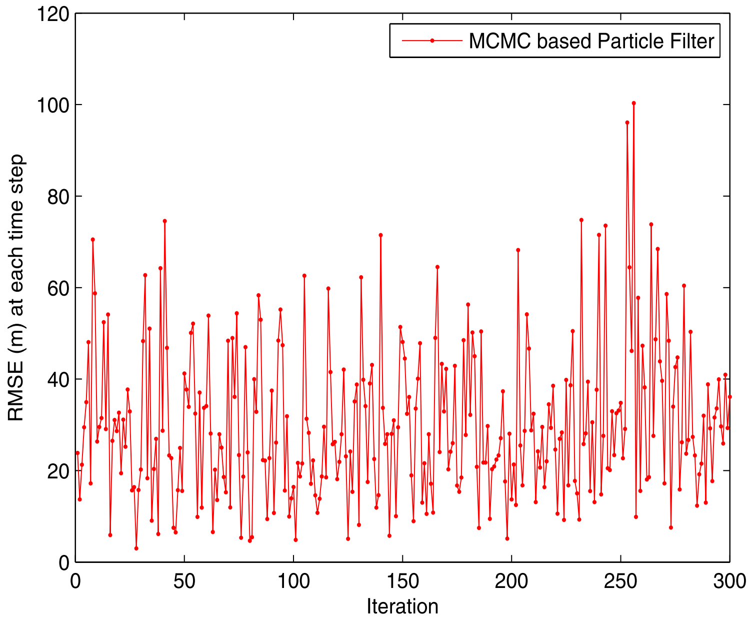


Fig. 11. MCMC based particle filter for aircraft tracking: RMSE at each time step.

accelerates the MCMC convergence rate. Inspired by the particle swarm algorithm, the proposed algorithm propagates each particle based on both its history information and the information from other particles. As a consequence, the particles are herded to the area with high posterior distribution density much faster than the traditional MCMC move. The interacting MCMC particle filter is compared with the conventional MCMC based particle filter in two synthetic examples: a robot equipped with sonar tracks a maneuvering target and a radar tracks an aircraft performing coordinated turn. The simulation results show that the interacting MCMC particle filter gains superior tracking performance than the conventional MCMC based particle filter both in accuracy and robustness.

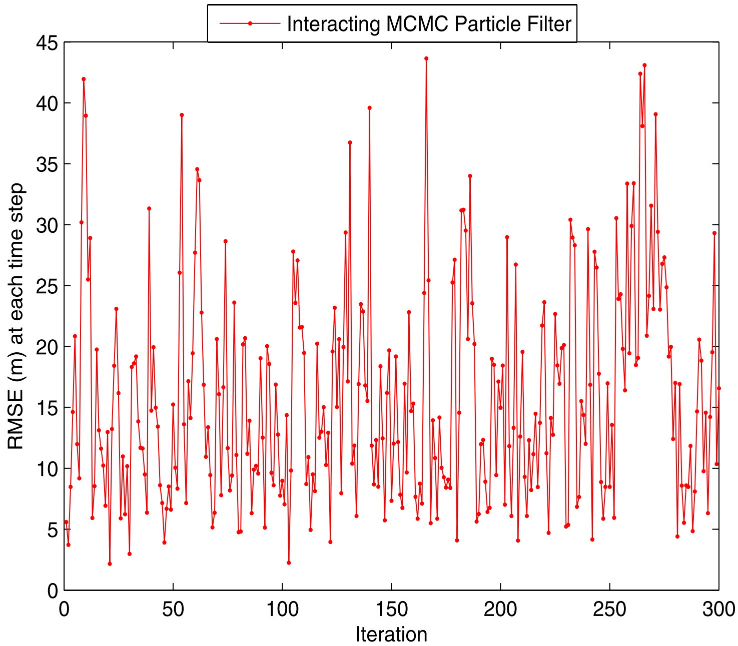
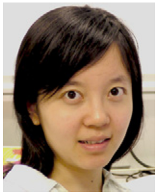


Fig. 12. Interacting MCMC particle filter for aircraft tracking: RMSE at each time step.

References

[1] V. Aidala, J. Davis, The utilization of data measurement residuals for adaptive Kalman filtering, OCEANS 5 (1973) 450–460.
[2] R. Kirlin, A. Moghaddamjoo, Robust adaptive Kalman filtering for systems with unknown step inputs and non-Gaussian measurement errors, in: IEEE International Conference on ICASSP, vol. 10, 1985, pp. 157–160.
[3] M. Efe, D. Atherton, Maneuvering target tracking with an adaptive Kalman filter, in: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 1, 1998, pp. 737–742.
[4] H. Lee, M.-J. Tahk, Generalized input-estimation technique for tracking maneuvering targets, IEEE Trans. Aerospace Electron. Syst. 35 (1999) 1388–1402.
[5] Y.T. Chan, F. Couture, Manoeuvre detection and track correction by input estimation, IEE Proc. Radar Signal Process. 140 (1993) 21–28.
[6] Y.H. Park, J.H. Seo, J.G. Lee, Tracking using the variable-dimension filter with input estimation, IEEE Trans. Aerospace Electron. Syst. 31 (1995) 399–408.
[7] A.T. Alouani, P. Xia, T.R. Rice, W.D. Blair, A two-stage Kalman estimator for tracking maneuvering targets, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, 1991, pp. 761–766.
[8] J. Tugnait, Detection and estimation for abruptly changing systems, Automatica 18 (1982) 607–615.
[9] H.A.P. Blom, Y. Bar-Shalom, The interacting multiple model algorithm for systems with a jump-linear smoothing application, IEEE Trans. Automat. Control 33 (1988) 780–783.
[10] Y. Bar-Shalom, K. Chan, H. Blom, Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm, IEEE Trans. Aerospace Electron. Syst. 25 (1989) 296–300.
[11] A. Munir, P. Artherton, Adaptive interacting multiple model algorithm for tracking a maneuvering target, IEE Proc. Radar Sonar Navig. 142 (1995) 11–17.
[12] R. Karlsson, N. Bergman, Auxiliary particle filters for tracking a maneuvering target, in: Proceedings of the 39th IEEE Conference on Decision and Control, vol. 4, 2000, pp. 3891–3895.
[13] N. Ikoma, T. Higuchi, H. Maeda, Tracking of maneuvering target by using switching structure and heavy-tailed distribution with particle filter method, in: Proceedings of the 2002 International Conference on Control Applications, vol. 2, 2002, pp. 1282–1287.
[14] W. Malcolm, A. Doucet, S. Zollo, Sequential Monte Carlo tracking schemes for maneuvering targets with passive ranging, in: Proceedings of the Fifth International Conference on Information Fusion, vol. 1, 2002, pp. 482–488.
[15] A. Doucet, N. Gordon, V. Kroshnamurthy, Particle filters for state estimation of jump Markov linear systems, IEEE Trans. Signal Process. 49 (2001) 613–624.
[16] R. Karlsson, F. Gustafsson, Range estimation using angle-only target tracking with particle filters, in: Proceedings of American Control Conference, vol. 5, 2001, pp. 3743–3748.
[17] N. Ikoma, N. Ichimura, T. Higuchi, H. Maeda, Maneuvering target tracking by using particle filter, in: IFSA World Congress and 20th NAFIPS International Conference, vol. 4, 2001, pp. 2223–2228.
[18] C. Musso, N. Oudjane, F. LeGland, Improving Regularised Particle Filters in Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001.
[19] W.R. Gilks, C. Berzuini, Following a moving target – Monte Carlo inference for dynamic Bayesian models, J. Roy. Statist. Soc. Ser. B 63 (2001) 127–146.
[20] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2002) 174–188.
[21] W. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97–109.
[22] W.R. Gilks, S. Richardson, D.J. Spiegelhalter, Markov Chain Monte Carlo in Practice, Chapman & Hall, 1996.
[23] P.J. Angeline, Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences, in: V.W. Porto, N. Saravanan, D. Waagen, A.E. Eiben (Eds.), Evolutionary Programming VII, Springer-Verlag, 1998, pp. 601–610.
[24] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: IEEE Int. Conf. Neural Netw., 1995, pp. 1942–1948.

[25] J. Kennedy, The particle swarm: Social adaptation of knowledge, in: Int. Conf. Evolutionary Computation, 1997, pp. 303–308.
[26] M. Clerc, J. Kennedy, The particle swarm: Explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (2002) 58–73.
[27] Philip J. Davis, Philip Rabinowitz, Methods of Numerical Integration, Academic Press, 1984.



Liu Jing received the B.Eng. and M.Eng. degrees in automatic control from the Department of Automatic Control, Northwestern Polytechnical University, Xi'an, China, in 1998 and 2000, respectively. She received the Ph.D. degree in 2006 from the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. Her research interests include multiple-target tracking, maneuvering-target tracking, and data fusion, with particular emphasis on the application of particle filters.

Prahlad Vadakkepat (M'00–SM'05) received the M.Tech. and Ph.D. degrees from the Indian Institute of Technology Madras, India, in 1989 and 1996, respectively. From 1991 to 1996, he was a Lecturer with the Regional Engineering College Calicut (currently the National Institute of Technology Calicut), Kerala, India. From 1996 to 1998, he was with Korea Advanced Institute of Science and Technology, Daejeon, Korea, as a Postdoctoral Fellow. He is currently an Associate Professor with the National University of Singapore, Singapore. His research interests are in humanoid robotics, distributed robotic systems, evolutionary robotics, neuro-fuzzy controllers, and intelligent control techniques. Dr. Vadakkepat is the Founder Secretary of the Federation of International Robot-soccer Association [www.fira.net], where he is currently the FIRA General Secretary. He was appointed as the Technical Activity Coordinator of IEEE Region 10 from 2001 to 2002. He is an Associate Editor of the International Journal of Humanoid Robotics since 2003.