

Available online at www.sciencedirect.com



Digital Signal Processing 17 (2007) 774-786

Digital Signal Processing

www.elsevier.com/locate/dsp

# Performance comparison of EKF and particle filtering methods for maneuvering targets

Mónica F. Bugallo\*, Shanshan Xu, Petar M. Djurić

Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794-2350, USA

Available online 25 October 2006

#### Abstract

Online tracking of maneuvering targets is a highly nonlinear and challenging problem that involves, at every time instant, the estimation not only of the unknown state in the dynamic model describing the evolution of the target, but also the underlying model accounting for the regime of movement. In this paper we review and compare several sequential estimation procedures, that use appropriate strategies for coping with various models that account for the different modes of operation. We focus on the application of the recently proposed cost-reference particle filtering (CRPF) methodology, which aims at the estimation of the system state without using probability distributions. The resulting method has a more robust performance when compared to standard particle filtering (SPF) algorithms or the interactive multiple model (IMM) algorithm based on the use of the well known extended Kalman filter (EKF). Advantages and disadvantages of the considered algorithms are illustrated and discussed through computer simulations.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Maneuvering target tracking; Sequential estimation; Cost-reference particle filtering

# 1. Introduction

In recent years the problem of target tracking has attracted significant attention in the signal processing community [1–7]. Most of the standard algorithms for target tracking have been focused on targets that do not change their regimes of movement while they are tracked. If the targets have several regimes of movement, i.e., different movement models, they can maneuver, which implies that tracking cannot be successful if it is done with one model. By the inclusion of several models, coping with maneuvering becomes possible, but the overall tracking is then much more challenging [8,9]. From a theoretical point of view this is a very interesting nonlinear problem, which includes the estimation of the underlying model at every time instant as well as its unknown states and parameters [9,10]. In this paper, we review, analyze, and compare the application of different techniques to the problem of tracking a single maneuvering target.

Maneuvering target tracking has often been addressed by the interactive multiple model (IMM) algorithm [11–13] and combined with different techniques like the probabilistic data association (PDA) [14] or the various versions of the Kalman filter [11,15]. These methods have become almost the standard approach to maneuvering target tracking

\* Corresponding author.

1051-2004/\$ – see front matter  $\,$  © 2006 Elsevier Inc. All rights reserved. doi:10.1016/j.dsp.2006.10.001

E-mail address: monica@ece.sunysb.edu (M.F. Bugallo).

[16,17]. However, they have some major limitations due to the fact that their performance depends on the elemental filters they are based on. If the underlying filters do not fit the considered problem well, the performance of the algorithms gets severely degraded.

Maneuvering target tracking has also been tackled by particle filtering schemes [8,9,15,18–21]. The primary reason to use these methods is their flexibility and accuracy in resolving very difficult nonlinear problems where the underlying models are represented by dynamic state-space equations. Existing standard particle filtering (SPF) methods require a mathematical representation of the dynamics of the system evolution that includes assumptions of probabilistic models.

Cost-reference particle filtering (CRPF), unlike SPF, aims at the estimation of the system state from the available observations without a priori knowledge of any distribution function [22]. The reference derived from the assumed probability distributions is substituted by a user-defined cost function that measures the quality of the state signal estimates according to the available observations. The resulting techniques present a more robust performance than the one achieved by SPF methods whose theory is based on probabilistic assumptions. The basic CRPF method and its variants have already been successfully applied to the problem of tracking maneuvering targets [10,20,23].

In this paper, we review the CRPF method which uses a deterministic approach for model proposals, a problem that is inherent to maneuvering target tracking [23]. The algorithm is compared with the multiple model auxiliary SPF which uses the same strategy for coping with the multiple models in the system [20]. We consider two different formulations of the CRPF, one of which has been proven to be equivalent to that of the auxiliary SPF [24]. For performance comparison, we also include the IMM algorithm based on the extended Kalman filter (EKF).

The remaining of this paper is organized as follows. Section 2 introduces the dynamic system that describes the problem of tracking of maneuvering targets moving along a two-dimensional space. The fundamentals of the CRPF family and its application to the considered problem are described in Section 3. We discuss other algorithms, the IMM-EKF and the multiple model auxiliary SPF, which are used for comparison with the CRPF in Section 4. Computer simulation results that illustrate the performance of the different algorithms are presented in Section 5. Finally, brief concluding remarks are included in Section 6.

#### 2. A dynamic system describing a maneuvering target

A maneuvering target trajectory is characterized by a constant velocity regime with short periods of turning that correspond to maneuvers. The dynamics of this type of objects can be described by a Markovian switching structure of the form [8]

$$\tilde{\mathbf{x}}_t = \mathbf{A}(\omega_{t-1})\tilde{\mathbf{x}}_{t-1} + \mathbf{B}\mathbf{u}_t,$$

where  $\tilde{\mathbf{x}}_t = [x_{1,t} \ x_{2,t} \ \dot{x}_{1,t} \ \dot{x}_{2,t}]^\top \in \Re^4$  contains the Cartesian coordinates of the target position (m) and velocity (m/s) in the *xy*-plane. The state transition matrices are given by

$$\mathbf{A}(\omega_t) = \begin{pmatrix} 1 & 0 & \frac{\sin(\omega_t T_s)}{\omega_t} & -\frac{1-\cos(\omega_t T_s)}{\omega_t} \\ 0 & 1 & -\frac{1-\cos(\omega_t T_s)}{\omega_t} & \frac{\sin(\omega_t T_s)}{\omega_t} \\ 0 & 0 & \cos(\omega_t T_s) & -\sin(\omega_t T_s) \\ 0 & 0 & \sin(\omega_t T_s) & \cos(\omega_t T_s) \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{pmatrix}$$

where  $T_s$  is the sampling period (s) and  $\omega_t$  is a turn rate variable. The maneuver behavior depends on the velocity and is modeled as

$$\omega_t = \frac{a_t}{\sqrt{\dot{x}_{1,t-1}^2 + \dot{x}_{2,t-1}^2}} + u_{\omega,t}$$

where  $a_t$  represents the typical maneuvering acceleration at time instant t, which randomly switches among three possible values,  $\mathcal{A} = \{a^{(1)}, a^{(2)}, a^{(3)}\}$ , and defines the nature of the movement, i.e., straight motion, right turn or left turn. Therefore, we have three different system models, and we use  $m_t \in \{1, 2, 3\}$  to denote the system model at time instant t (e.g.,  $m_t = 1$  corresponds to  $a_t = a^{(1)}$ ). Switching the movement regime occurs according to the transition probability matrix,

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix},$$

where  $h_{ij} = p(m_t = j | m_{t-1} = i)$  is the probability of the system to switch from model *i* at time instant t - 1 to model *j* at time instant *t*, *i*, *j* = 1, 2, 3, and the initial model probabilities are set to guarantee that  $\sum_{i=1}^{3} p(m_t = i) = 1$ . The overall state vector is given by

$$\mathbf{x}_t = \begin{bmatrix} \tilde{\mathbf{x}}_t & \boldsymbol{\omega}_t \end{bmatrix}^\top,\tag{1}$$

with state noises  $\mathbf{u}_t \in \Re^2$  and  $u_{\omega,t} \in \Re$ .

The observation equation is defined as

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t,\tag{2}$$

where  $\mathbf{v}_t$  is the observation noise with independent components  $v_{j,t}$ , j = 1, 2, and the function  $h(\cdot)$  has two components. An emitter on the moving target transmits a signal with power  $P_0$  through a fading channel with attenuation coefficient  $\alpha$ . The received signal power and the relative angle between the target and a reference point with fixed location  $\mathbf{r} = [r_x r_y]^{\top}$  are measured, i.e.,

$$h_1(\mathbf{x}_t) = 10 \log_{10} \left( \frac{P_0}{\|\mathbf{r} - \mathbf{p}_t\|^{\alpha}} \right), \qquad h_2(\mathbf{x}_t) = \angle (\mathbf{p}_t - \mathbf{r}),$$

where  $\mathbf{p}_t = [x_{1,t} \ x_{2,t}]^\top \in \Re^2$  denotes the position of the target and  $\|\mathbf{z}\| = \sqrt{\mathbf{z}^\top \mathbf{z}}$  is the norm of the vector  $\mathbf{z}$ . The objective is the sequential estimation of the target state,  $\mathbf{x}_{0:t}$ , given the sequence of measurements,  $\mathbf{y}_{1:t}$ .

# 3. The CRPF method

The CRPF methodology allows for recursive estimation of unobserved states of dynamic systems without assumptions about the probability distributions of the noises in the state and observation equations [22]. The main feature of these filters is that the statistical reference provided by the a posteriori state probability density function (pdf),  $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ , is substituted by a user-defined cost function that measures the quality of the state signal estimates according to the available observations. The cost-reference particle filter at time *t* is described by a set of *M* state samples or particles, which represent the state,  $\mathbf{x}_t$ , and associated costs up to that time instant. To account for the multiple regimes existing in the maneuvering target tracking problem, the filter also includes the maneuvering model,  $m_t$ . The cost-reference particle filter at time instant *t* is given by the measure [10]

$$\Xi_t = \{\mathbf{x}_t^{(i)}, \mathcal{C}_t^{(i)}, m_t^{(i)}\}_{i=1}^M$$

The particles  $\mathbf{x}_t^{(i)}$ , i = 1, ..., M, are generated using a selected propagation density, which is constructed under a set of mild conditions [22]. In general, the cost function is formulated by means of a recursive additive structure,

$$\mathcal{C}_{t}^{(i)} = \mathcal{C}\left(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t},\lambda\right) = \lambda \mathcal{C}\left(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1},\lambda\right) + \Delta \mathcal{C}\left(\mathbf{x}_{t}^{(i)}|\mathbf{y}_{t}\right),\tag{3}$$

where the term on the right-hand side of the expression represents the cost up to time t - 1 weighted by a forgetting factor,  $\lambda$ , plus a cost increment,  $\Delta C(\mathbf{x}_t^{(i)}|\mathbf{y}_t)$ , obtained from the state and observation vectors at time t. Different functions for the cost function as well as for the cost increment can be proposed under some defined design conditions [22]. In this paper, we use as cost increment

$$\Delta \mathcal{C}(\mathbf{x}_t^{(i)}|\mathbf{y}_t) = \|\mathbf{y}_t - h(\mathbf{x}_t^{(i)})\|^q, \tag{4}$$

where q is either 1 or 2.

The measure,  $\Xi_t$ , is randomly propagated when  $\mathbf{y}_{t+1}$  is observed and an updated measure,  $\Xi_{t+1}$ , is built. The multiple model CRPF method for the stated maneuvering target tracking problem consists of the following steps:

(1) Selection or resampling: Given M particles at time t, 3M predictions or risks are evaluated at time t + 1 (one for each model and each particle). These predictions are calculated using the particles from the previous time instant,  $\{\mathbf{x}_{t}^{(i)}\}_{i=1}^{M}$  and the current observation,  $\mathbf{y}_{t+1}$ . They represent anticipated costs of the propagated particles. A possible expression for the risks is [22]

$$\mathcal{R}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_t^{(i)} + \Delta \mathcal{C} \big( \mathbf{x}_t^{(i)} | \mathbf{y}_{t+1}, m_{t+1} \big), \quad m_{t+1} = 1, 2, 3,$$

where  $\Delta C(\mathbf{x}_t^{(i)}|\mathbf{y}_{t+1}, m_{t+1})$  is obtained by first predicting a new state according to the model  $m_{t+1}$ , and then applying the incremental cost function given by Eq. (4) [20]. Once predictions are generated, we replicate particles with low risks while high-risk particles are discarded. To perform this selection, we construct a probability mass function (pmf) from the risk  $\mathcal{R}_{t+1}$ , i.e.,

$$\hat{\pi}_{t+1}^{(i,m_{t+1})} \propto \mu \left( \mathcal{R}_{t+1}^{(i,m_{t+1})} \right)$$

where  $\mu(\cdot): \Re \to [0, +\infty)$  is a monotonically decreasing function. As shown in [22], the choice of  $\mu(\cdot)$  has a direct impact on the performance of the algorithm. In this paper we chose  $\mu(\cdot)$  as

$$\mu\left(\mathcal{R}_{t+1}^{(i,m_{t+1})}\right) = \frac{1}{\left(\mathcal{R}_{t+1}^{(i,m_{t+1})} - \min_{i} \{\mathcal{R}_{t+1}^{(i,m_{t+1})}\} + \delta\right)^{\beta}}$$

where  $\delta = \frac{1}{10} \sqrt{\operatorname{var}\{\mathcal{R}_{t+1}^{(i,m_{t+1})}\}_{i=1}^{3M}}$  with  $\operatorname{var}(\cdot)$  denoting variance, and  $\beta = 2$ . Other methods for performing particle selection have also been explored [22,24].

It is important to remark that only M trajectories survive the resampling step. Specifically, we select  $\hat{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(k)}$ with probability  $\hat{\pi}_{t+1}^{(k,m_{t+1})}$ , and build an intermediate particle filter,

$$\hat{\Xi}_{t+1} = \{\hat{\mathbf{x}}_{t}^{(i)}, \hat{\mathcal{C}}_{t}^{(i)}, m_{t+1}^{(i)}\}_{i=1}^{M}$$

where  $\hat{C}_{t}^{(i)} = C_{t}^{(k)}$  if, and only if,  $\hat{\mathbf{x}}_{t}^{(i)} = \mathbf{x}_{t}^{(k)}$ . (2) *Particle propagation*: Using a selected propagation density [22], new particles are generated by

$$\mathbf{x}_{t+1}^{(i)} \sim p_{t+1} (\mathbf{x}_{t+1} | \hat{\mathbf{x}}_{t}^{(i)}, m_{t+1}^{(i)}),$$

and the associated costs are updated according to

$$\mathcal{C}_{t+1}^{(i)} = \lambda \hat{\mathcal{C}}_t^{(i)} + \Delta \mathcal{C} \left( \mathbf{x}_{t+1}^{(i)} | \mathbf{y}_{t+1} \right)$$

As a result, we obtain the updated representation of the particle filter,

$$\Xi_{t+1} = \left\{ \mathbf{x}_{t+1}^{(i)}, \mathcal{C}_{t+1}^{(i)}, m_{t+1}^{(i)} \right\}_{i=1}^{M}$$

(3) *Estimation*: One possibility to obtain an estimate of the state is by assigning a pmf  $\pi_{t+1}^{(i)}$ , i = 1, 2, ..., M, to the particles in  $\Xi_{t+1}$ . This pmf can be constructed analogously as  $\hat{\pi}_{t+1}^{(i)}$  in step (1), substituting the risks by costs. Using this pmf, useful estimates of the state based on the mean value can be calculated according to

$$\mathbf{x}_{t+1}^{\text{mean}} = \sum_{i=1}^{M} \pi_{t+1}^{(i)} \mathbf{x}_{t+1}^{(i)}.$$
(5)

Other alternatives for estimation include obtaining estimates based on the model that has maximum "posterior probability," assigning estimates based on each of the three models together with the posterior probabilities of the models, or applying methods that do not require pmf calculation [22,24].

The multiple model CRPF algorithm is outlined in Table 1. The initialization and update of distribution parameters for particle generation have been previously addressed in [22].

## 4. Review of the IMM-EKF and the multiple model SPF methods

For comparison and benchmarking purposes, we have also considered the IMM algorithm based on the use of the well known EKF and the algorithm known as auxiliary SPF [8]. The latter has an algorithmic structure (resampling, importance sampling and state estimation) very similar to the CRPF family.

Table 1	
The multiple mod	lel CRPF algorithm

 $\begin{aligned} & \text{Initialization} \\ & \text{For } i = 1, \dots, M, \text{ initialize:} \\ & \text{The particle } \mathbf{x}_{0}^{(i)} \sim \mathcal{U}(I_{\mathbf{x}_{0}}) \\ & \text{The cost } \mathcal{C}_{0}^{(i)} = 0 \\ & \text{The parameters for the proposal function, } p_{t+1}(\cdot) \\ & \text{Recursive update} \\ & \text{For } t = 1 \text{ to } T \\ & (1) \text{ Selection or resampling (for } i = 1, \dots, M) \\ & \text{Compute:} \\ & \mathcal{R}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})\|^{q}, \quad m_{t+1} = 1, 2, 3, \\ & \text{ where } \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})\|^{q}, \quad m_{t+1} = 1, 2, 3, \\ & \text{ where } \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})\|^{q}, \quad m_{t+1} = 1, 2, 3, \\ & \text{ where } \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})\|^{q}, \quad m_{t+1} = 1, 2, 3, \\ & \text{ where } \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})\|^{q}, \quad m_{t+1} = 1, 2, 3, \\ & \text{ where } \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})} = \lambda \mathcal{C}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})\|_{t+1} \\ & \hat{\pi}_{t+1}^{(i,m_{t+1})} \approx \mu (\mathcal{R}_{t+1}^{(i,m_{t+1})}) = \frac{1}{(\mathcal{R}_{t}^{(i,m_{t+1})}, \hat{\pi}_{t+1}^{(i)})} \\ & \text{ Resample to obtain } \quad \hat{\mathcal{L}}_{t+1} = \{ \hat{\mathbf{x}}_{t}^{(i)}, \hat{\mathcal{C}}_{t}^{(i)}, m_{t+1}^{(i)} \} \\ & \text{ If } t > 10, \text{ update parameters for the proposal function, } p_{t+1}(\cdot) \\ & \text{ Calculate } \mathcal{C}_{t+1}^{(i)} = \lambda \hat{\mathcal{C}}_{t}^{(i)} + \|\mathbf{y}_{t+1} - h(\mathbf{x}_{t+1}^{(i)})\|^{q} \\ & \text{ (3) Estimation} \\ & \text{ Compute:} \\ & \tilde{\pi}_{t}^{(i)} = \mu (\mathcal{C}_{t}^{(i)}) = \frac{1}{(\mathcal{C}_{t}^{(i)} - \min_{t} (\mathcal{C}_{t}^{(i)}) + \delta)\beta} \\ & \pi_{t+1}^{(i)} = \sum_{j=1}^{m} \pi_{t+1}^{(j)} \\ & \text{ (for } i = 1, \dots, M) \\ & \mathbf{x}_{t+1}^{\text{mean}} = \sum_{j=1}^{m} \pi_{t+1}^{(i)} \mathbf{x}_{t+1}^{(i)} \\ \end{array}$ 

## 4.1. The IMM-EKF method

The EKF is a suboptimal estimation method suitable for systems with nonlinearities [1]. It proceeds by continually updating a linearization around the previous state estimate and by approximating the state densities by Gaussian densities. For the maneuvering target tracking problem described in Section 2, the EKF is combined with the IMM scheme in order to deal with several models in the system [25]. Therefore, three EKFs, one per model, are implemented in parallel. As a result, a mixture density is obtained and approximated by a single Gaussian which matches the first and second moments.

The IMM algorithm is summarized in Table 2 and each of its cycles consists of the following steps [1]:

(1) Calculation of mixing probabilities: Given that model j is in effect at time instant t and conditioned on the observations up to time instant t - 1,  $\mathbf{y}_{1:t-1}$ , the probability  $\mu_{i|j}(t - 1|t - 1)$ , i = 1, 2, 3, that the *i*th model was in effect at time instant t - 1 is calculated.

(2) *Combination of mixing probabilities*: Given the outputs of the EKFs at time instant t - 1,  $\hat{\mathbf{x}}_{t-1|t-1}^{i}$ , i = 1, 2, 3, the mixed initial condition for the filter match to model j,  $\hat{\mathbf{x}}_{t-1|t-1}^{0j}$ , and the corresponding covariance,  $\mathbf{P}_{t-1|t-1}^{0j}$ , are obtained.

- (3) *Model-matched filtering*: The estimates in step (2) are used as inputs to the filter matched to model *j*, which calculates the output  $\hat{\mathbf{x}}_{t|t}^{j}$  and updates the covariance matrix  $\mathbf{P}_{t|t}^{j}$  using the observation  $\mathbf{y}_{t}$ .
- (4) Update of model probabilities: The *j*th model probability at time *t*,  $\mu_j(t)$ , is updated using the likelihood of the *j*th filter.
- (5) *Estimation*: The model-conditioned estimates,  $\hat{\mathbf{x}}_{t|t}^{j}$ , j = 1, 2, 3, are combined to get a state estimate.

Table 2
The IMM-EKF algorithm
Initialization
For $j = 1, 2, 3$ (each model), initialize:
The estimate $\hat{\mathbf{x}}_{0 0}^{j} = E[\mathbf{x}_{0}]$
The covariance matrix $\mathbf{P}_{0 0}^{j} = E\left[(\mathbf{x}_{0} - E[\mathbf{x}_{0}])(\mathbf{x}_{0} - E[\mathbf{x}_{0}])^{H}\right]$
The model probability $\mu_1(0) = p(1), \mu_2(0) = p(2), \mu_3(0) = p(3)$
Recursive update
For $t = 1$ to $T$
(1) Calculation of mixing probabilities (for $i, j = 1, 2, 3$ )
Compute $\mu_{i j}(t-1 t-1) = \frac{1}{c_i}h_{ij}\mu_i(t-1)$ , with $\bar{c}_j = \sum_{i=1}^3 h_{ij}\mu_i(t-1)$
(2) Combination of mixing probabilities (for $j = 1, 2, 3$ )
Calculate:
$\hat{\mathbf{x}}_{t-1 t-1}^{0j} = \sum_{i=1}^{3} \hat{\mathbf{x}}_{t-1 t-1}^{i} \mu_{i j}(t-1 t-1)$
$\boldsymbol{\epsilon}^{ij}$ , $\boldsymbol{\epsilon}^{ij}$ , $\boldsymbol{\epsilon}^{ij}$ , $\boldsymbol{\epsilon}^{ij}$ , $\boldsymbol{\epsilon}^{ij}$ , $\boldsymbol{\epsilon}^{ij}$ , $\boldsymbol{\epsilon}^{ij}$
$\mathbf{P}^{0j} = \sum_{i=1}^{3} \mu_{i} (t_{i} + t_{i}) \times (\mathbf{P}^{i} + \epsilon^{ij} + \epsilon^{ij})$
$\mathbf{r}_{t-1 t-1} = \sum_{i=1}^{t} \mu_i  j(t-1 t-1) \times (\mathbf{r}_{t-1 t-1} + \mathbf{e}_{t-1 t-1}\mathbf{e}_{t-1 t-1})$
(3) Model-matched filtering (for $j = 1, 2, 3$ )
Use $\hat{\mathbf{x}}_{t-1 t-1}^{o_j}$ and $\mathbf{P}_{t-1 t-1}^{o_j}$ as input to EKF <sub>j</sub>
Estimate $\hat{\mathbf{x}}_{t t}^{j}$ and $\mathbf{P}_{t t}^{j}$
Obtain the likelihood $\Lambda_j(t) = p(\mathbf{y}_t   j, \hat{\mathbf{x}}_{t-1 t-1}^{0j}, \mathbf{P}_{t-1 t-1}^{0j})$
(4) Update of model probabilities (for $j = 1, 2, 3$ )
Compute $\mu_j(t) = \frac{1}{c} \Lambda_j(t) \bar{c}_j$ , with $c = \sum_{i=1}^3 \Lambda_j(t) \bar{c}_j$
(5) Estimation
Obtain $\hat{\mathbf{x}}_{t t} = \sum_{j=1}^{3} \hat{\mathbf{x}}_{t t}^{j} \mu_{j}(t)$

#### 4.2. The multiple model auxiliary SPF method

The SPF uses samples and associated weights to approximate the posterior distribution of the estimated states. Unlike the CRPF, samples/weights are generated/updated based on the available probabilistic knowledge of the system of interest. For the multiple model case, the particle set is given by [9]

$$\Upsilon_t = \{\mathbf{x}_t^{(i)}, w_t^{(i)}, m_t^{(i)}\}_{i=1}^M,$$

where  $w_t^{(i)}$  denotes the *i*th weight. We focus on the auxiliary SPF [26, Chapter 23], which has an analogous basic structure as the original CRPF [22]. Although several strategies can be applied to cope with the multiple models in the maneuvering target tracking problem [9,20], we adopt the same scheme as the one described for the CRPF in Section 3. The resulting multiple model auxiliary SPF implementation is an extension of the auxiliary SPF algorithm, which uses the prior density as the proposal function and proceeds, at time instant t + 1, as follows:

(1) *Resampling*: Using the particles from the previous time instant,  $\{\mathbf{x}_{t}^{(i)}\}_{i=1}^{M}$ , and the three possible models, 3M particles are proposed. Resampling is carried using a multinomial distribution obtained from the weights

$$\tilde{w}_{t+1}^{(i,m_{t+1})} = w_t^{(i)} p(m_{t+1}|m_t^{(i)}) p(\mathbf{y}_{t+1}|\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})}), \quad m_{t+1} = 1, 2, 3$$

where  $p(\mathbf{y}_t | \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})})$  is obtained by first predicting a new state according to the model  $m_{t+1}$ ,  $\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})}$ , and then obtaining the likelihood. *M* new particles are sampled out of the 3*M* proposed particles to obtain the transitional filter

$$\hat{\Upsilon}_{t+1} = \{\hat{\mathbf{x}}_{t}^{(i)}, \hat{w}_{t}^{(i)}, m_{t+1}^{(i)}\}_{i=1}^{M}$$

(2) Particle propagation: It is carried out using the state transition distribution and the model sample, i.e.,

$$\mathbf{x}_{t+1}^{(i)} \sim p(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t}^{(i)}, m_{t+1}^{(i)}).$$

Table 3Multiple model auxiliary SPF algorithm

 $\begin{aligned} & \text{Initialization} \\ & \text{For } i = 1, \dots, M, \text{ initialize:} \\ & \text{The model } m_0^{(i)} \sim p(m_0) \\ & \text{The particle } \mathbf{x}_0^{(i)} = \mathbf{n} \\ & \mathbf{x}_0^{(i)} = \mathbf{n} \\ & \mathbf{Recursive update} \\ & \text{For } t = 1 \text{ to } T, \text{ for } i = 1, \dots, M \\ & (1) \text{ Resampling (for } i = 1, \dots, M) \\ & \text{Calculate } \tilde{w}_{t+1}^{(i,m_{t+1})} = w_t^{(i)} p(m_{t+1}|m_t^{(i)}) p(\mathbf{y}_{t+1}|\bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})}), \quad m_{t+1} = 1, 2, 3, \\ & \text{ where } \bar{\mathbf{x}}_{t+1}^{(i,m_{t+1})} \text{ is a prediction of the state based on the model} \\ & \text{Resample to obtain } \hat{T}_{t+1} = \{ \hat{\mathbf{x}}_t^{(i)}, \hat{w}_t^{(i)}, m_{t+1}^{(i)} \}_{i=1}^M \\ & (2) \text{ Particle propagation (for } i = 1, \dots, M) \\ & \text{ Obtain:} \\ & \mathbf{x}_{t+1}^{(i)} \sim p(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_t^{(i)}, m_{t+1}^{(i)}) \\ & \tilde{w}_{t+1}^{(i)} \propto \frac{p(\mathbf{y}_{t+1}|\hat{\mathbf{x}}_t^{(i)})}{p(\mathbf{y}_{t+1}|\hat{\mathbf{x}}_t^{(i)})} \\ & w_{t+1}^{(i)} = \frac{\tilde{w}_{t+1}^{(i)}}{\sum_{j=1}^{M} \tilde{w}_{t+1}^{(j)}} \\ & (3) \text{ Estimation} \\ & \text{ Compute } \mathbf{x}_{t+1}^{\text{mean}} = \sum_{i=1}^{M} w_{t+1}^{(i)} \mathbf{x}_{t+1}^{(i)} \end{aligned}$ 

The number of particles must be large enough to ensure an adequate representation of the posterior density. Once the particles are propagated, weights are calculated and normalized [21].

(3) *Estimation*: The procedure is completely analogous as that of the CRPF. The mean square estimate of the state,  $\mathbf{x}_{t+1}^{\text{mean}}$ , is calculated using the particles and their weights.

A summary of this algorithm is provided in Table 3.

It is well known that particle filtering is a computationally expensive methodology. The SPF and CRPF algorithms have approximately the same computational complexity whereas the Kalman filtering based methods are much less demanding. Also, the more particles we use for tracking with particle filtering, the slower the processing of the data is. However, particle filtering can be implemented in parallel and savings in computing times of the order of magnitude can be achieved [27].

## 5. Computer simulations

In this section, we present computer simulations that illustrate the performance of the various tracking algorithms described in the previous sections. We compared the following methods:

- (1) the IMM-EKF (labeled in the figures as IMM-EKF);
- (2) the multiple model auxiliary SPF with correct knowledge of noise distributions (labeled MM-SPF);
- (3) the multiple model auxiliary SPF with mismatched information about the observation noise distribution (labeled MM-SPFM);
- (4) the CRPF with parameters q = 1 and  $\lambda = 0.95$  (labeled CRPF);
- (5) the CRPF with parameters q = 1 and  $\lambda = 0$  (labeled CRPF0);
- (6) the CRPF with parameters q = 2 and  $\lambda = 0.95$  (labeled CRPF2);
- (7) the CRPF with parameters q = 2 and  $\lambda = 0$  (labeled CRPF20).

For all the particle filters, the number of particles was initially set to M = 500. The CRPF with  $\lambda = 0$  is of special interest since it has been proven to be equivalent to the multiple model auxiliary SPF [20,24].

## 5.1. Fixed trajectory

In the first experiment we studied the performance of the algorithms over a single fixed trajectory. We considered a scenario where the target started from an initial position close to (0, 0), and with an initial constant velocity of 55 m/s. It moved for 5 min and measurements were made with a sampling time of  $T_s = 5$  s. The magnitude of the velocity was constrained within the interval (50, 60) m/s for the whole motion course. The maneuvering model variable,  $a_t$ , switched among three discrete values  $\{0, 5, -5\}$ . The model,  $m_t$ , was changing according to the Markovian transition probability matrix,

$$\mathbf{H} = \begin{pmatrix} 0.70 & 0.15 & 0.15 \\ 0.20 & 0.70 & 0.10 \\ 0.20 & 0.10 & 0.70 \end{pmatrix},$$

and the state noises were modeled as zero-mean white Gaussian processes,

$$\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 \mathbf{I}_2), \qquad u_{\omega,t} \sim \mathcal{N}(\mathbf{0}, \sigma_{u_{\omega}}^2),$$

where  $\sigma_u^2 = 10^{-2}$  and  $\sigma_{u_{\omega}}^2 = 10^{-4}$  were the respective variances and  $\mathbf{I}_2$  represents the 2 × 2 identity matrix [7, Chapter 6].

Measurements were collected at the reference point, following Eq. (2) with  $P_0 = 1$ . The observation noise was modeled as a mixture Gaussian distribution,

 $\mathbf{v}_t \sim 0.7 \mathcal{N}(\mathbf{0}, 4 \cdot \mathbf{I}_2) + 0.3 \mathcal{N}(\mathbf{0}, 25 \cdot \mathbf{I}_2).$ 

The selected target trajectory and the reference point are shown in Fig. 1.

Figure 2 shows the system trajectory in a single run and the estimates corresponding to the algorithms. For clarity in the presentation, we plotted the results of the four CRPF methods on the left, and the results for one of the CRPF algorithms and the rest of the algorithms on the right. It can be seen that all the CRPF algorithms and the multiple model auxiliary SPF method with perfect knowledge of the distributions of the noises performed similarly and remained locked to the true trajectory of the target during the whole tracking interval. However, the IMM-EKF, which had to use a Gaussian assumption for the observation noise and considered  $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ , and the multiple model auxiliary SPF, which also used that same wrong Gaussian assumption, were not able to track properly. Note that the CRPF algorithms were assuming the same distribution as the IMM-EKF for propagation of particles and this choice did not affect their performance. Therefore the CRPF algorithms, which do not rely on any prior probabilistic information, showed a more robust performance.

We also compared the algorithms by means of the root-mean-square (RMS) error of the position and the percentage of track loss. Let  $\mathbf{p}_t = [x_{1,t} \ x_{2,t}]^{\top}$  denote the true position of the target at time instant *t*, and  $\hat{\mathbf{p}}_t = [\hat{x}_{1,t} \ \hat{x}_{2,t}]^{\top}$  be the



Fig. 1. Target trajectory and the reference location.



Fig. 2. Trajectory estimates. Left: Different CRPF algorithms. Right: Comparison of the CRPF method with the rest of the methods.



Fig. 3. Left: RMS error of the position. Right: Percentage of track loss for various values of  $\gamma$ .

corresponding estimate obtained by a particular filter. The RMS error at time instant t was calculated by averaging over N = 500 independent simulation trials,

RMS<sub>t</sub> = 
$$\sqrt{\frac{1}{N} \sum_{i=1}^{N} [(\hat{x}_{1,t,i} - x_{1,t,i})^2 + (\hat{x}_{2,t,i} - x_{2,t,i})^2]},$$

where the subscript *i* denoted the trial number. For a particular simulation run, a track loss was confirmed when the position error,  $e_t = \sqrt{(\hat{x}_{1,t} - x_{1,t})^2 + (\hat{x}_{2,t} - x_{2,t})^2}$ , exceeded a threshold  $\gamma$  within at least 10 consecutive sampling periods. The percentage was obtained by averaging over *N* independent runs. Figure 3 depicts the results. Again the IMM-EKF and the multiple model auxiliary SPF with incorrect noise distributions showed much worse performance than the other filters.

## 5.2. Effect of the reference point location

In the next experiment, we moved the reference point away from the trajectory area from (-2741, -2161) to (-6000, -5000). This is equivalent to reducing the signal-to-noise ratio (SNR). To combat the low SNR, we increased the number of particles for the particle filtering algorithms, and we set it to M = 1000. The results are shown in Fig. 4. We can draw similar conclusions as from the previous experiment.



Fig. 4. Effect of location of reference point. Top: Trajectory estimates. Bottom: RMS error of the position and percentage of track loss for various values of  $\gamma$ .

## 5.3. Effect of the sampling period

We also checked the sensitivity of the algorithms to variations in the sampling period,  $T_s$ . Using the same parameters as in the first experiment, we set  $T_s = 10$  s. The results for the RMS error of the position and percentage of track loss are shown in Fig. 5. It can be seen that the RMS error increased for all the methods. The IMM-EKF performed the worst. A possible reason is that the maneuver onset and/or offset instant could be missed, and the statistical relation between adjacent samples was weakened.

## 5.4. Effect of the incomplete information

The performance of the multiple model auxiliary SPF strongly depends on the availability of correct model information, which includes noise distributions, model switching probabilities, and model parameters. We previously showed that the performance of this algorithm severely deteriorated when incorrect information about the noise distributions was used. In another set of experiments we tested the robustness of the considered algorithms with respect to the model transition probabilities and the models' parameters.

First, we considered the following model switching probability matrices:

$$\mathbf{H}_1 = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}, \qquad \mathbf{H}_2 = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.4 & 0.2 & 0.4 \\ 0.4 & 0.4 & 0.2 \end{pmatrix}.$$



Fig. 5. Effect of the sampling interval. Left: RMS error of the position. Right: Percentage of track loss for various values of  $\gamma$ .



Fig. 6. Effect of availability of switching probability matrices. Left: RMS error of the position. Right: Percentage of track loss for various values of  $\gamma$ .

The trajectory was generated by using the matrix  $\mathbf{H}$  in Section 5.1 but the algorithms assumed the previous matrices for calculation of the likelihoods. Figure 6 shows the results. We can see that the performance of the multiple model auxiliary SPF became worse in the absence of such information.

We next supposed that the typical maneuvering model value was unknown, and we analyzed the performance of the algorithms by considering two sets of possible values for the acceleration variable,  $a_t \in A_1 = \{0, 3, -3\}$  and  $a_t \in A_2 = \{0, 7, -7\}$  instead of  $\{-5, 0, 5\}$ . Figure 7 shows the results. Again, it is clear that the CRPF is more robust when this information is not available.

#### 5.5. Different trajectories

We applied the considered algorithms on 500 randomly generated trajectories. The results shown in Fig. 8 are similar to the previous experiments.

#### 6. Conclusions

In this paper, we have discussed and compared the application of different techniques to the problem of tracking a maneuvering target in the two-dimensional space. We have focused on the cost-reference particle filtering (CRPF) methodology, which does not use probabilistic assumptions required by standard particle filtering (SPF), leading to a more robust performance. The multiple model problem is solved by a deterministic model proposal process that



Fig. 7. Effect of availability of model value. Left: RMS error of the position. Right: Percentage of track loss for various values of  $\gamma$ .



Fig. 8. Different trajectories. Left: RMS error of the position. Right: Percentage of track loss for various values of  $\gamma$ .

allows for exploring the different regimes of movement in each step of the algorithm. We have compared the resulting algorithm with the extended Kalman filter (EKF) and the multiple model auxiliary SPF method. The advantages and disadvantages of the considered algorithms were illustrated and discussed through computer simulations.

Future work includes the study of different cost functions and their relevance to attain better performance and the extension of the algorithms to deal with the problem of tracking of multiple targets.

# Acknowledgments

This work has been supported by the National Science Foundation under the Award CCF-0515246 and by the Office of Naval Research under Award N00014-06-1-0012.

## References

- Y. Bar-Shalom, X.R. Li, T. Kirubarajan, Estimation with Application to Tracking and Navigation: Theory, Algorithms, and Software, Wiley, New York, 2001.
- [2] N.J. Gordon, A hybrid bootstrap filter for target tracking in clutter, IEEE Trans. Aerospace Electron. Syst. 33 (1) (1997) 353–358.
- [3] F. Gustafsson, F. Gunnarsson, N. Bergman, et al., Particle filters for positioning, navigation, and tracking, IEEE Trans. Signal Process. 50 (2) (2002) 425–437.
- [4] C. Hue, J.P.L. Cadre, P. Perez, Tracking multiple objects with particle filtering, IEEE Trans. Aerospace Electron. Syst. 38 (3) (2002) 791-812.
- [5] D. Li, K.D. Wong, Y.H. Hu, A.M. Sayeed, Detection, classification, and tracking of targets, IEEE Signal Process. Mag. 19 (2) (2002) 17–29.

- [6] X. Li, V. Jilkov, Survey of maneuvering target tracking—Part I: Dynamic systems, IEEE Trans. Aerospace Electron. Syst. 39 (4) (2003) 1333–1364.
- [7] B. Ristic, S. Arulampalam, N. Gordon, Beyond Kalman Filter: Particle Filter for Tracking Applications, Artech House Publishers, 2004.
- [8] R. Karlsson, N. Bergman, Auxiliary particle filters for tracking a maneuvering target, in: Proceedings of IEEE Conference on Decision and Control, vol. 4, 2000, pp. 3891–3895.
- [9] S. McGinnity, G.W. Irwin, Multiple model bootstrap filter for maneuvering target tracking, IEEE Trans. Aerospace Electron. Syst. 36 (3) (2000) 1006–1012.
- [10] M.F. Bugallo, S. Xu, J. Míguez, P.M. Djurić, Maneuvering target tracking using cost reference particle filtering, in: Proceedings of the 29th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '2004), Montreal, Canada, 2004.
- [11] H.A.P. Blom, Y. Bar-Shalom, The interacting multiple model algorithm for systems with Markovian switching coefficients, IEEE Trans. Automat. Control 33 (8) (1988) 780–783.
- [12] X. Li, V. Jilkov, A survey of maneuvering target tracking—Part V: Multiple-model methods, IEEE Trans. Aerospace Electron. Syst. 41 (4) (2003) 1255–1321.
- [13] E. Mazor, A. Averbuch, Y. Bar-Shalom, J. Dayan, Interacting multiple model methods in target tracking: A survey, IEEE Trans. Aerospace Electron. Syst. 34 (1) (1998) 103–123.
- [14] T. Kirubarajan, Y. Bar-Shalom, D. Lerro, Bearing-only tracking of maneuvering targets using a batch-recursive estimator, IEEE Trans. Aerospace Electron. Syst. 37 (3) (2001) 770–780.
- [15] B. Ristic, M.S. Arulampalam, Tracking a maneuvering target using angle-only measurements: Algorithms and performance, Signal Process. 83 (6) (2003) 1223–1238.
- [16] W. Blair, G. Watson, IMM algorithm for solution to benchmark problem for tracking maneuvering targets, in: Proceedings of SPIE, vol. 2221, SPIE, 1994.
- [17] V. Jilkov, D. Angelova, T. Semerdjiev, Design and comparison of mode-set adaptive IMM algorithms for maneuvering target tracking, IEEE Trans. Aerospace Electron. Syst. 35 (1) (1999) 343–350.
- [18] M.S. Arulampalam, N. Gordon, M. Orton, B. Ristic, A variable structure multiple model particle filter for GMTI tracking, in: Proceedings of the 2002 International Conference on Information Fusion, Annapolis, MD, 2002.
- [19] M.R. Morelande, S.S. Challa, Manoeuvring target tracking in clutter using particle filters, IEEE Trans. Aerospace Electron. Syst. 41 (1) (2005) 252–270.
- [20] M.F. Bugallo, S. Xu, P.M. Djurić, Comparison of EKF- and PF-based methods in tracking maneuvering targets, in: Proceedings of the Aerospace Conference 2006, Big Sky, MT, 2006.
- [21] A. Doucet, N. Gordon, V. Krishnamurthy, Particle filters for state estimation of jump Markov linear systems, IEEE Trans. Signal Process. 49 (3) (2001) 613–624.
- [22] J. Míguez, M.F. Bugallo, P.M. Djurić, A new class of particle filters for random dynamical systems with unknown statistics, EURASIP J. Appl. Signal Process. 15 (2004) 2278–2294.
- [23] S. Xu, M.F. Bugallo, P.M. Djurić, Maneuvering target tracking with simplified cost reference particle filters, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France, 2006.
- [24] M.F. Bugallo, J. Míguez, P.M. Djurić, Positioning by cost reference particle filters: Study of various implementations, in: Proceedings of the 2005 International Conference on "Computer as a Tool" (EUROCON), Belgrade, Serbia and Montenegro, 2005.
- [25] X.R. Li, Y. Bar-Shalom, Design of an interacting multiple model algorithm for air traffic control tracking, IEEE Trans. Control Syst. 1 (3) (1993) 186–194.
- [26] A. Doucet, N. de Freitas, N.J. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, 2001.
- [27] M. Bolić, P.M. Djurić, S. Hong, Resampling algorithms and architectures for distributed particle filters, IEEE Trans. Signal Process. 53 (7) (2005) 2442–2450.

**Mónica F. Bugallo** received the Ph.D. degree in computer engineering from the University of A Coruña, Spain, in 2001. From 1998 to 2000 she was with the Departamento de Electrónica y Sistemas at the Universidade da Coruña, Spain, where she worked in interference cancellation applied to multiuser communication systems. In 2001, she joined the Department of Electrical and Computer Engineering at Stony Brook University where she is currently Assistant Professor and teaches courses in digital communications and information theory. Her research interests lie in the area of statistical signal processing and its applications to different disciplines including communications and biology.

**Shanshan Xu** received her B.S. in electrical engineering from Tsinghua University, Beijing, China, in 2000 and the M.S. degree in electrical engineering from Stony Brook University, New York, in 2003. She is now a Ph.D. student in the Department of Electrical and Computer Engineering, Stony Brook University. Her research interests are in the field of statistical signal processing and its applications.

**Petar M. Djurić** received his B.S. and M.S. degrees in electrical engineering from the University of Belgrade in 1981 and 1986, respectively, and his Ph.D. degree in electrical engineering from the University of Rhode Island in 1990. From 1981 to 1986 he was Research Associate with the Institute of Nuclear Sciences, Vinca, Belgrade. Since 1990 he has been with Stony Brook University, where he is Professor in the Department of Electrical and Computer Engineering. He works in the area of statistical signal processing, and his primary interests are in the theory of modeling, detection, estimation, and time series analysis and its application to a wide variety of disciplines including wireless communications and bio-medicine.