Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1546 – 1558

doi:10.3906/elk-1811-52

Multiellipsoidal extended target tracking with known extent using sequential Monte Carlo framework

Süleyman Fatih KARA*, **Emre ÖZKAN**
Department of Electrical and Electronics Engineering, Faculty of Engineering,
Middle East Technical University, Ankara, Turkey

Received: 10.11.2018 • **Accepted/Published Online:** 11.02.2019 • **Final Version:** 22.03.2019

Abstract: In this paper, we consider a variant of the extended target tracking (ETT) problem, namely the multiellipsoidal ETT problem. In multiellipsoidal ETT, target extent is represented by multiple ellipses, which correspond to the origin of the measurements on the target surface. The problem involves estimating the target's kinematic state and solving the association problem between the measurements and the ellipses. We cast the problem in a sequential Monte Carlo (SMC) framework and investigate different marginalization strategies to find an efficient particle filter. Under the known extent assumption, we define association variables to find the correct association between the measurements and the ellipses; hence, the posterior involves both discrete and continuous random variables. By expressing the measurement likelihood as a mixture of Gaussians we derive and employ a marginalized particle filter for the independent association variables without sampling the discrete states. We compare the performance of the method with its alternatives and illustrate the gain in nonstandard marginalization.

Key words: Extended target tracking, marginalized particle filter, conditional hidden Markov models

1. Introduction

Conventional target tracking methods consider point targets that generate at most one measurement per scan [1, 2]. With the evolution of sensor technology, the focus of the target tracking literature shifted from point target tracking to extended target tracking (ETT) algorithms, which aim at estimating target extent simultaneously with the kinematic state using a set of measurements per scan. The extent of a target can be considered as the region on the target's surface from which the measurements are obtained. Early works on ETT used predefined geometric shapes with a few degrees of freedom to represent target extent, such as rectangles, sticks, circles, or ellipses [3–5].

The most popular ETT model is based on random matrices, which are used to define elliptical target extents [6, 7]. More recent contributions involve detailed representations of complex shapes. These algorithms are based on random hypersurface models that can represent so-called star convex shapes [8, 9]. Recent methods use the Gaussian process-based random hypersurface models to represent target extents [10, 11].

The random matrix models [6, 7] can be extended to multiellipsoidal target extents to obtain a more detailed representation of the extent [12]. In multiellipsoidal target extent models, the true associations of the measurements and the ellipses are not known. Here, we assume that the target extent is known. Therefore, we will focus on the association problem, which will be solved by defining discrete association variables. The joint

*Correspondence: kara.fatih@metu.edu.tr

density of the discrete and the continuous states will be approximated using different PF approaches [13–15]. We will derive and apply a nonstandard marginalization technique, which does not require sampling the discrete variables. We will illustrate the advantages of the nonstandard marginalization approach by comparing it with its alternatives in simulations.

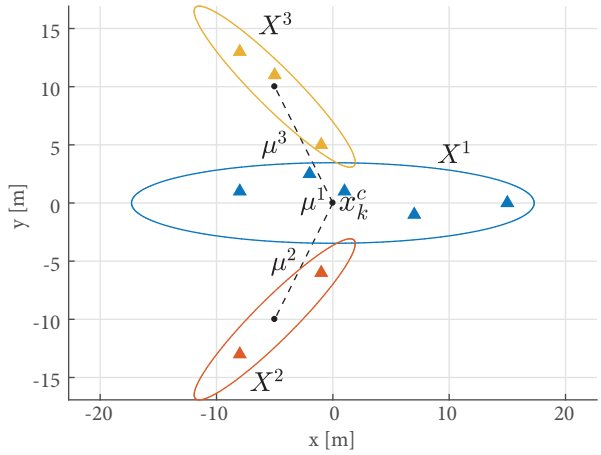

Figure 1. Target extent model with 3 subobjects and measurements assigned to true subobjects.

2. Target extent model

Here, we will assume that each target generates multiple measurements from multiple sources on its extent. The target extent is represented by multiple ellipses. This model can be used to represent various shapes as shown in Figure 1. Each of the ellipses in Figure 1 is called a subobject. Each subobject is associated with a known symmetric positive definite matrix $\{X^\ell\}_{\ell=1}^M \in \mathbb{R}^{d\times d}$ and a mean vector $\{\mu^\ell\}_{\ell=1}^M \in \mathbb{R}^{n_y}$, where M is the number of subobjects and d is the dimension of the extent. Without loss of generality, μ^1 can be assumed to be zero to define one of the subobjects to be the main body of the target (see Figure 1). Suppose m_k measurements $\mathbf{Y}_k = \{y_k^j\}_{j=1}^{m_k} \in \mathbb{R}^{n_y}$ are collected at time k. Under additive Gaussian measurement noise assumption, the measurement equation corresponding to this extent model can be written as a mixture of Gaussians:

$$p(y_k^j|x_k^c) = \sum_{\ell=1}^{M} \pi_\ell \mathcal{N}(y_k^j; x_k^c + \mu^\ell, sX^\ell + R), \tag{1}$$

where x_k^c is the position of the center point, $R \in \mathbb{R}^{n_y\times n_y}$ is the measurement noise covariance matrix, $s \in \mathbb{R}$ is a positive constant scale factor, and $\pi_\ell \in [0,1]$ is the ℓth mixture weight, i.e. the prior probability that a measurement belongs to the ℓth subobject. We define our state vector x_k to be the kinematic state vector that holds the relevant variables:

$$x_k \triangleq [(x_k^c)^T \ (\tilde{x}_k)^T]^T, \tag{2}$$

where $\tilde{x}_k$ denotes the additional state variables. In the Bayesian framework, one can define appropriate priors to compute the posterior distribution of unknown variables. Our aim is to find a recursive update for the posterior distribution:

$$p(x_k|\mathbf{Y}_{1:k}). \tag{3}$$

Unfortunately, this posterior distribution is intractable because, given a set of measurements, we do not know which measurement belongs to which subobject. This creates a combinatorial problem that grows exponentially with time k. In order to solve this assignment problem, we first define an association variable r_k^j for each measurement y_k^j. r_k^j indicates the index of the subobject that measurement y_k^j belongs to, and it takes integer values between 1 and M. The prior probability that a measurement belongs to a certain subobject is assumed to be known:

$$\mathbb{P}(r_k^j = \ell) = \pi_\ell \quad \text{for } \ell = 1, \ldots, M. \tag{4}$$

The augmented vector that holds the prior assignment probabilities of subobjects is denoted by π:

$$\pi \triangleq [\pi_1, \ldots, \pi_M]. \tag{5}$$

3. Inference

The system in this problem can be represented using a conditionally linear Gaussian system (CLGS), which can be written as:

$$x_{k+1} = Fx_k + u_k, \tag{6a}$$

$$y_k^j = Hx_k + v_k^{r_k^j}, \tag{6b}$$

where the Gaussian process noise is expressed as $u_k \sim \mathcal{N}(0, Q)$. The conditional measurement noise can be expressed as $v_k^{r_k^j} \sim \mathcal{N}(0, sX^{r_k^j} + R)$. Our aim is to approximate the joint posterior density of the association variables and the kinematic state:

$$p(x_k, \mathbf{r}_k|\mathbf{Y}_{1:k}), \tag{7}$$

where $\mathbf{r}_k = [r_k^1 \ldots r_k^{m_k}]^T$. To solve this problem we will employ particle filters (PFs) whose details are explained in the following subsections. We use two different marginalization (aka Rao–Blackwellization) approaches to derive these algorithms [14, 15].

3.1. Bootstrap particle filter

The first algorithm that we will consider is the bootstrap particle filter (BPF) [16]. The general flow of the BPF can be described as follows. For each time instant k, the association variables $\mathbf{r}_k$ and the kinematic state should be sampled from their prior densities:

$$x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)}), \tag{8a}$$

$$\mathbf{r}_k^{(i)} \sim \pi, \tag{8b}$$

where the superscript $^{(i)}$ indicates the ith particle. To update the importance weights, the likelihood of the multiple measurements can be expressed as:

$$p(\mathbf{Y}_k|x_k^{(i)}, \mathbf{r}_k^{(i)}) = \prod_{j=1}^{m_k} p(y_k^j|x_k^{(i)}, r_k^{(i),j}), \tag{9}$$

where we assume that each measurement obtained at time k is conditionally independent of the others. Using the likelihood above, the weights are updated as follows:

$$w_k^{(i)} \propto p(\mathbf{Y}_k|x_k^{(i)}, \mathbf{r}_k^{(i)}) w_{k-1}^{(i)}. \tag{10}$$

After calculating the weights, resampling is performed if necessary.

Algorithm 1 Bootstrap particle filter pseudocode.

Initialization at $k = 0$:
for all $i = 1, \ldots, N$ **do**
  Sample $x_0^{(i)} \sim p_0$ and $\mathbf{r}_0^{(i)} \sim \pi_0$
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
end for
Iterations:
for all $k = 1, \ldots, T$ **do**
  for all $i = 1, \ldots, N$ **do**
    Sample $x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$
    Sample $\mathbf{r}_k^{(i)} \sim \pi$
    Update the importance weights according to Eq. (10)
  end for
  Normalize the weights
  Resample, if necessary
end for

3.2. Marginalization

A generic PF approximates the posterior density by weighted random samples. This method is fairly efficient when the state dimension is low. Unfortunately, the efficiency of a PF does not scale up with the dimension of the state; hence, a vast number of particles is needed to approximate the posterior of high-dimensional states. One way to alleviate this problem is to exploit some inherent analytical structures in the target density $p(x_k|y_{1:k})$, if available. Consider the factorization of the target density into conditionally linear and nonlinear parts as follows:

$$p(x_k|y_{1:k}) = p(x_k^l, x_{0:k}^n|y_{1:k}) = p(x_k^l|x_{0:k}^n, y_{1:k}) p(x_{0:k}^n|y_{1:k}). \tag{11}$$

A standard marginalized particle filter (MPF) aims at sampling the nonlinear states and computing analytical expressions for the conditionally linear part of the state (first factor in Eq. (11)). By doing so, the target

density can be represented as

$$p(x_k^l, x_{0:k}^{n,(i)}|y_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{x_{0:k}^{n,(i)}}(x_{0:k}^n) p(x_k^l|x_{0:k}^{n,(i)}, y_{1:k}). \tag{12}$$

As an analogy, KF computes the posterior density as analytical expressions. PF approximates the posterior density with weighted samples. MPF approximates the posterior by using both weighted samples and analytical expressions. This results in sampling efficiency in PF because, depending on the system model, it might be possible to sample a low-dimension nonlinear state x_k^n and compute the posterior of the full state x_k. The MPF requires conditional analytical expressions in the posterior density that exists in our system. In the following subsections, we will investigate two different marginalization approaches for this problem.

3.3. Marginalization by sampling the discrete states

In CLGSs, a standard choice of the target density is the posterior of the continuous and discrete states $p(x_k, \mathbf{r}_{0:k}|\mathbf{Y}_{1:k})$. Consider the following factorization of the target density:

$$p(x_k, \mathbf{r}_{0:k}|\mathbf{Y}_{1:k}) = p(x_k|\mathbf{r}_{0:k}, \mathbf{Y}_{1:k}) p(\mathbf{r}_{0:k}|\mathbf{Y}_{1:k}). \tag{13}$$

Within the MPF framework, the posterior of the discrete states can be approximated using particles:

$$\widehat{p}^N(\mathbf{r}_{0:k}|\mathbf{Y}_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta_{\mathbf{r}_{0:k}^{(i)}}(\mathbf{r}_{0:k}). \tag{14}$$

Conditioned on the discrete states, the system equations become linear and Gaussian. Therefore, the conditional posterior of the continuous state can be written as a Gaussian density $p(x_k|\mathbf{r}_{0:k}^{(i)}, \mathbf{Y}_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}^{(i)}, P_{k|k}^{(i)})$. The mean and covariance of this density are calculated analytically using a KF. Furthermore, the posterior density of the continuous and the discrete states are approximated as

$$p(x_k, \mathbf{r}_{0:k}^{(i)}|\mathbf{Y}_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{\mathbf{r}_{0:k}^{(i)}}(\mathbf{r}_{0:k}) \mathcal{N}(x_k; \hat{x}_{k|k}^{(i)}, P_{k|k}^{(i)}). \tag{15}$$

Iterations of a MPF at each time include:

- *'Measurement Update'* and *'Time Update'* steps for continuous variables;
- *'Sampling'* and *'Weight Update'* steps for discrete variables.

The details of the aforementioned steps are given below.

At any time k, we first propagate the conditional density of the continuous states per particle, which corresponds to the KF's time update equations given by

$$\hat{x}_{k|k-1}^{(i)} = F_k \hat{x}_{k-1}^{(i)}, \tag{16a}$$

$$P_{k|k-1}^{(i)} = F_k P_{k-1}^{(i)} F_k^T + Q. \tag{16b}$$

Then we sample the discrete state $\mathbf{r}_k$ from the importance density:

$$\mathbf{r}_k^{(i)} \sim q(\cdot). \tag{17}$$

Here, if we want to use the bootstrap proposal density, the importance density $q(\cdot)$ is chosen to be equal to the prior density π. Alternatively, one can use the optimal proposal density to increase the performance:

$$\pi_{opt}^{(i),j} \triangleq p(r_k^{(i),j}|y_{1:k}^j, r_{k-1}^{(i),j}) \propto p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j)p(r_k^{(i),j}|r_{k-1}^{(i),j}), \tag{18}$$

where the predictive likelihood $p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j)$ is expressed further as

$$\begin{aligned} p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j) &= \int p(y_k^j|x_k^{(i)}, r_k^{(i),j}, y_{1:k-1}^j)p(x_k^{(i)}|r_k^{(i),j}, y_{1:k-1}^j)dx_k \\ &= \int \mathcal{N}(y_k^j; Hx_k^{(i)} + \mu^{r_k^j}, R + sX^{r_k^j})\mathcal{N}(x_k^{(i)}; x_{k|k-1}^{(i)}, P_{k|k-1}^{(i)})dx_k \\ &= \mathcal{N}(y_k^j; H\hat{x}_{k|k-1}^{(i)} + \mu^{r_k^j}, HP_{k|k-1}^{(i)}H^T + R + sX^{r_k^j}). \end{aligned} \tag{19}$$

The optimal proposal density is calculated using the following expressions for each subobject ℓ:

$$\widetilde{\pi}_{opt,\ell}^{(i),j} = \mathcal{N}\big(y_k^j; H\hat{x}_{k|k-1}^{(i)} + \mu^\ell, HP_{k|k-1}^{(i)}H^T + sX^\ell + R\big)\pi_\ell, \tag{20a}$$

$$\pi_{opt,\ell}^{(i),j} = \frac{\widetilde{\pi}_{opt,\ell}^{(i),j}}{\sum_{n=1}^M \widetilde{\pi}_{opt,n}^{(i),j}}, \tag{20b}$$

where $\hat{x}_{k|k-1}^{(i)}$ and $P_{k|k-1}^{(i)}$ are calculated in the time update step. Next, the particle weights are updated according to the following equation:

$$w_k^{(i)} \propto \prod_{j=1}^{m_k} \frac{p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j)\pi_{r_k^{(i),j}}}{q(r_k^{(i),j})} w_{k-1}^{(i)}. \tag{21}$$

For the continuous state, the measurement update is performed using the KF measurement update equations. Measurement update of the continuous state is performed consecutively for each measurement y_k^j:

$$\hat{x}_k^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k(y_k^j - H\hat{x}_{k|k-1}^{(i)} - \mu^{r_k^j}), \tag{22a}$$

$$P_k^{(i)} = (I - K_kH)P_{k|k-1}^{(i)}, \tag{22b}$$

$$K_k = P_{k|k-1}^{(i)}H^T(HP_{k|k-1}^{(i)}H^T + sX^{r_k^j} + R)^{-1}, \tag{22c}$$

where we assign $\hat{x}_{k|k-1}^{(i)} \leftarrow \hat{x}_k^{(i)}$, $P_{k|k-1}^{(i)} \leftarrow P_k^{(i)}$ and iterate for $j = 1, \ldots, m_k$. As a last step, resampling is employed if necessary. We will refer to the MPF where we can perform the marginalization over the continuous states analytically as the CMPF. The CMPF is summarized in Algorithm 2.

Algorithm 2 Summary of the CMPF for multimeasurement case.

Initialization at $k = 0$:
for all particles $i = 1, \ldots, N$ **do**
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
  Set initial kinematic state of each particle $\hat{x}_0^{(i)} = \bar{x}_0$, $\hat{P}_0^{(i)} = \bar{P}_0$
end for
Iterations:
for time $k = 1, \ldots, T$ **do**
  for all particles $i = 1, \ldots, N$ **do**
    Perform time update of the kinematic state
    Sample the association variables, $\mathbf{r}_k^{(i)} \sim q(\cdot)$
    Update the weights according to Eq. (21)
  end for
  Normalize the weights
  for all particles $i = 1, \ldots, N$ **do**
    for all measurements $j = 1, \ldots, m_k$ **do**
      Perform measurement update of the kinematic states according to Eq. (22)
      Assign $\hat{x}_{k|k-1}^{(i)} \leftarrow \hat{x}_k^{(i)}$, $P_{k|k-1}^{(i)} \leftarrow P_k^{(i)}$ and iterate
    end for
  end for
  Resample, if necessary
end for

3.4. Marginalization by sampling the continuous states

In order to marginalize the discrete state, we slightly modify the target density as $p(x_{0:k}, \mathbf{r}_k|\mathbf{Y}_{1:k})$. In CLGSs, it is possible to obtain an analytical expression for the density of the discrete state when conditioned on the continuous states. Consider the following factorization of the target density:

$$p(x_{0:k}, \mathbf{r}_k|\mathbf{Y}_{1:k}) = p(\mathbf{r}_k|x_{0:k}, \mathbf{Y}_{1:k})p(x_{0:k}|\mathbf{Y}_{1:k}). \tag{23}$$

The posterior of the continuous states can be approximated using particles:

$$\widehat{p}^N(x_{0:k}|\mathbf{Y}_{1:k}) = \sum_{i=1}^{N} w_{0:k}^{(i)} \delta_{x_{0:k}^{(i)}}(x_{0:k}). \tag{24}$$

When $x_{0:k}$ is given, the conditional distribution of r_k can be expressed analytically [15]. Unlike the model given in [15], the discrete states are conditionally independent in our model. We define the mode probabilities as

$$\alpha_k^{(i),j}(\ell) \triangleq \mathbb{P}(r_k^j = \ell|x_{0:k}^{(i)}, y_{1:k}). \tag{25}$$

We will refer to the MPF where we can perform the marginalization over the discrete states analytically as the DMPF. The basic flow of the DMPF for each time instant k can be summarized as follows. Due to independence assumption in our CLGS, we can propagate the mode probabilities according to:

$$\alpha_{k|k-1}^{(i),j}(\cdot) = \alpha_{k-1}^{(i),j}(\cdot), \tag{26}$$

where $\{\alpha_{k-1}^{(i),j}(\ell)\}_{\ell=1}^{M}$ is represented with $\alpha_{k-1}^{(i),j}(\cdot)$ for the sake of simplicity. The continuous state x_k is propagated by sampling from the importance density:

$$x_k^{(i)} \sim q_k(x_k|x_{0:k-1}^{(i)}, \mathbf{Y}_{1:k}). \tag{27}$$

Here, one can use the bootstrap proposal density:

$$\begin{aligned} q(x_k|x_{0:k-1}^{(i)}, y_{1:k}^{j}) =& p(x_k|x_{0:k-1}^{(i)}, y_{1:k-1}^{j}) \\ =& \sum_{\ell=1}^{M} p(x_k|x_{k-1}^{(i)}) \alpha_{k|k-1}^{(i),j}(\ell). \end{aligned} \tag{28}$$

When a new measurement is available, we perform the weight update. Since the continuous state $x_k^{(i)}$ carries information about the discrete state $\mathbf{r}_k$, it serves as an extra measurement [15]. $\alpha_k(\cdot)$ in Eq. (25) can also be expressed in terms of the joint density:

$$\begin{aligned} \alpha_k^{(i),j}(\cdot) &= \mathbb{P}(r_k^j|x_k^{(i)}, x_{0:k-1}^{(i)}, y_k^j, y_{1:k-1}^j) \\ &= \frac{\mathbb{P}(r_k^j, x_k^{(i)}, y_k^j|x_{0:k-1}^{(i)}, y_{1:k-1}^j)}{\mathbb{P}(x_k^{(i)}, y_k^j|x_{0:k-1}^{(i)}, y_{1:k-1}^j)} \\ &\propto \mathbb{P}(r_k^j, x_k^{(i)}, y_k^j|x_{0:k-1}^{(i)}, y_{1:k-1}^j). \end{aligned} \tag{29}$$

For simplicity, we define a mid quantity γ_k:

$$\begin{aligned} \gamma_k^{(i)}(r_k^j) &\triangleq p(r_k^j, x_k^{(i)}, y_k^j|x_{0:k-1}^{(i)}, y_{1:k-1}^j) \\ &= p(y_k^j|x_k^{(i)}, r_k^j) p(x_k^{(i)}|x_{k-1}^{(i)}) \alpha_{k|k-1}^{(i),j}(r_k^j). \end{aligned} \tag{30}$$

Using $\gamma_k(\cdot)$, we can compute $\alpha_k(\cdot)$:

$$\alpha_k^{(i),j}(\ell) \propto \gamma_k^{(i),j}(\ell), \tag{31a}$$

$$\alpha_k^{(i),j}(\ell) = \frac{\gamma_k^{(i),j}(\ell)}{\sum_{m=1}^{M} \gamma_k^{(i),j}(m)}. \tag{31b}$$

The importance weights are updated with the following:

$$w_k^{(i)} \propto \prod_{j=1}^{m_k} \frac{p(x_k^{(i)}, y_k^j|x_{0:k-1}^{(i)}, y_{1:k-1}^j)}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k}^j)} w_{k-1}^{(i)}, \tag{32}$$

where the numerator of Eq. (32) is the marginalization of Eq. (30) over r_k, which is equal to $\sum_{m=1}^{M} \gamma_k^{(i),j}(m)$. Substituting this expression in the numerator of Eq. (32) results in:

$$w_k^{(i)} \propto \prod_{j=1}^{m_k} \frac{\sum_{m=1}^{M} \gamma_k^{(i),j}(m)}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k}^{j})} w_{k-1}^{(i)}. \tag{33}$$

Lastly, we perform resampling, if necessary. The DMPF algorithm is summarized in Algorithm 3. Since the DMPF represents the posterior of the continuous states with particles, its computational requirements increase with the dimension of the continuous state. If the dimension of the continuous state is high, the DMPF will require more particles to represent the corresponding density. Consequently, it will be inefficient and slower.

Algorithm 3 Summary of the DMPF for multimeasurement case.

Initialization at $k = 0$:
for all particles $i = 1, \ldots, N$ **do**
  Sample the initial kinematic state $x_0^{(i)} \sim p_0$
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
end for
Iterations:
for time $k = 1, \ldots, T$ **do**
  for all particles $i = 1, \ldots, N$ **do**
    for all measurements $j = 1, \ldots, m_k$ **do**
      Compute $\alpha_{k|k-1}^{(i),j}(\cdot)$ according to Eq. (26)
    end for
    Sample the kinematic state, $x_k^{(i)} \sim p(x_k|x_{0:k-1}^{(i)}, \mathbf{Y}_{1:k})$
    for all measurements $j = 1, \ldots, m_k$ **do**
      Compute $\gamma_k^{(i),j}(\cdot)$ according to Eq. (30)
      Compute $\alpha_{k|k}^{(i),j}(\cdot)$ according to Eq. (31)
    end for
    Update the weights according to Eq. (33)
  end for
  Normalize the weights
  Resample, if necessary
end for

4. Performance evaluation

We test the algorithms in a simulation of a moving multiellipse object that consists of 3 ellipses (see Figure 1). The object follows a linear path and the number of measurements at each scan is Poisson distributed with an average of 7 measurements. A constant velocity model is used as the kinematic model of the object, which will be explained in the following subsection.

4.1. 2-D constant velocity model

The nonmaneuvering 2-dimensional constant velocity model is one of the most commonly used models in target tracking problems [17]. The target speed is assumed to be nearly constant through a linear path and the system noise accounts for possible accelerations that disturb the speed. The state consists of two-dimensional positions

and velocities $x = [\mathrm{x\ y\ v_x\ v_y}]^T$. The system matrices in Eq. (6) are defined as:

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{34}$$

The process noise covariance matrix Q is chosen as

$$Q = \sigma^2 \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix},$$

where T is the sampling time. The sampling time is set to $T = 1$ s. The above-defined system is the basic 2-D nonmaneuvering constant velocity model, which can be extended by including other kinematic parameters such as heading, jerk, or acceleration.

4.2. Simulation

We simulate the system in 100 MC runs with the same measurement realization to observe the estimation consistency and variance. The numerical results provided here are an average of the MC runs. As a performance metric, RMSE will be used:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{T}(x_k - \widehat{x}_k)^2}. \tag{35}$$

Here, x_k indicates the true value and $\widehat{x}_k$ is the estimated value of the state at time k. In addition to RMSE, the computation times of the algorithms are compared. The algorithms are denoted as follows: BPF, CMPF-B (CMPF with bootstrap proposal density), CMPF-O (CMPF with optimal proposal density), and DMPF.

Since the BPF samples both the association variables $\mathbf{r}_k$ and the kinematic state x_k, it requires an excessive number of particles to provide satisfying results when compared with the other two algorithms. Therefore, the BPF is run with $N = 20{,}000$ particles while the CMPF-B, the CMPF-O, and the DMPF are run with 200 particles.

The prior assignment probabilities are assumed to be equal, which is a reasonable assumption when the measurement rates of the subobjects are unknown. The system noise variance is set to $\sigma^2 = 1$ and the measurement noise covariance is selected to be $R = \mathrm{diag}([10, 10])\mathrm{m}^2$. The scaling factor is taken as $s = 1$. Initial kinematics and their covariance matrix are chosen as $x_0 = [0, 0, 10\mathrm{m/s}, 0]^T$, $P_0 = \mathrm{diag}[400, 400, 100, 100]$, respectively. All simulations are run in MATLAB R2016b on a standard laptop with an Intel Core i5-7200U 2.50 GHz platform with 8 GB of RAM running Windows. The position estimates of all four algorithms are presented in Figure 2.

The average RMSE values for the position estimates are presented in the Table. The BPF has the largest MC variance and RMSE value even if it runs with 100 times more particles than the other algorithms. Therefore, one can claim that using marginalization (if possible) has a positive impact on the performance of the algorithms. We obtained the lowest MC variance and RMSE value with the CMPF-O. The DMPF has the closest estimation performance to the CMPF-O in the sense of RMSE and MC variance. One can also observe the effect of using

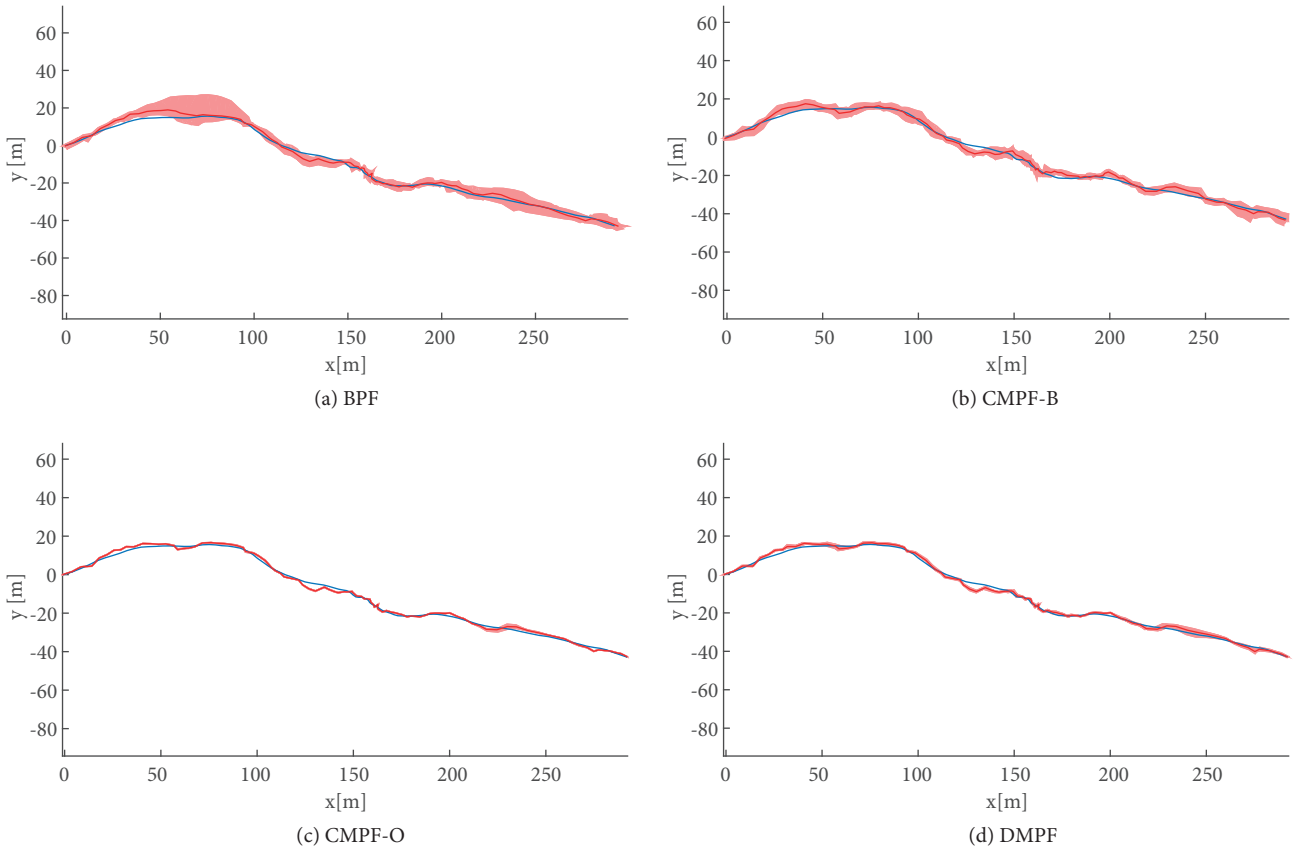

Figure 2. The position estimates of the BPF, CMPF-B, CMPF-O, and DMPF algorithms. The BPF is run with 20,000 particles while other algorithms are run with 200 particles. A total of 100 MC runs are performed for the simulations. The red and blue lines represent the average MC run results and the true positions, respectively. The transparent area shows the upper and lower bounds of the MC estimates.

the optimal proposal density instead of the bootstrap proposal density by comparing the results of the CMPF-B and the CMPF-O. The optimal proposal density has a positive impact on the performance of the algorithm. A second simulation is performed to observe the relationship between the number of particles (N) and RMSE values. In the simulation, all algorithms are run with N =100, 200, 500, 1000, 2000, and 4000 particles. The results are presented in Figure 3. The performances of DMPF and CMPF-O are similar for $N \geq 500$ whereas CMPF-B always has a higher RMSE.

Although CMPF-O achieves smaller RMSE values for a low number of particles, it has one drawback. Since the algorithm includes optimal proposal density computations and KF update equations performed per particle, it requires vast computation effort when compared with the DMPF. Average computation times per update are given in the Table. The DMPF has the smallest computation time since its implementation does not include computationally heavy steps or equations.

4.3. Orientation tracking of a maneuvering target

Lastly, we would like to illustrate the potential of the DMPF on a challenging problem, where the orientation of an extended target varies in time. Since the DMPF method samples the continuous states, it is straightforward to augment the state vector with the heading angle of the target and carry out the nonlinearities without

Table. RMSE of the position estimates and average computation times of the algorithms.

		RMSE [m]	Computation time [s]
BPF	(with 20K particles)	2.91	0.089
CMPF-B	(with 200 particles)	1.59	0.31
CMPF-O	(with 200 particles)	1.33	0.32
DMPF	(with 200 particles)	1.42	0.0019

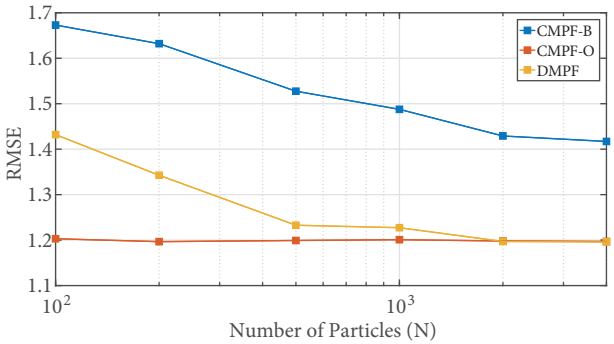

Figure 3. RMSE results versus the number of particles of the three algorithms. The blue line denotes the CMPF-B, the orange line denotes the CMPF-O, and the yellow line denotes the DMPF.

any model approximations. Thanks to this property, a maneuvering target, as shown in Figure 4, can easily be tracked by the DMPF. Note that the DMPF assumes a constant velocity model but the constant velocity assumption is not valid during the maneuvers. The algorithm is robust enough to handle such a mismatch, which is a common problem in most tracking applications. Estimates of the position and heading angle are compared with their true values in Figures 4 and 5, respectively. The simulation is run with 1000 particles and the number of measurements at each scan is Poisson distributed with an average of 15 measurements.

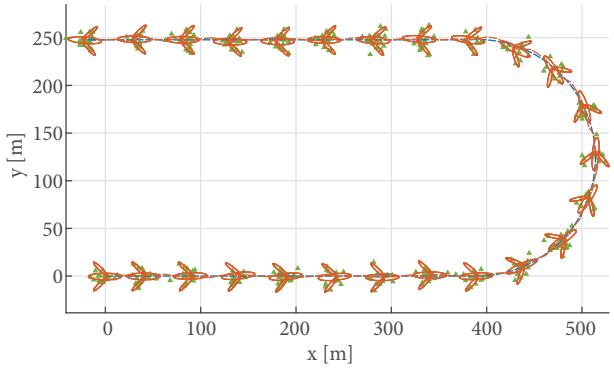

Figure 4. Position estimate of the DMPF with 1000 particles. The orange and blue dashed lines represent the position estimation result and the true value, respectively. The green triangles show the measurements obtained per time step and orange ellipses represent the target extent.

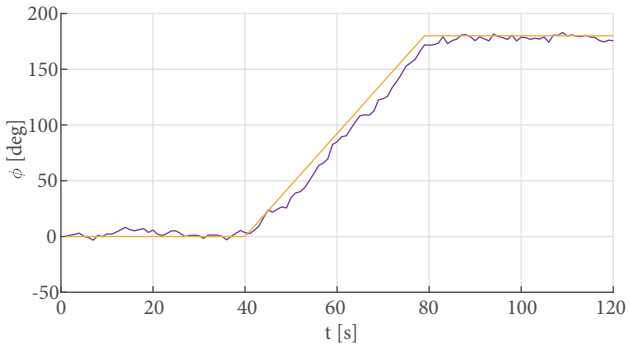

Figure 5. Heading angle estimate of the DMPF with 1000 particles. The purple and yellow lines represent the estimated and the true heading angle, respectively.

5. Conclusion

In this paper, a multiellipsoidal ETT model is studied in the SMC framework under the assumption of known extent. In this model, the posterior involves discrete and continuous random variables. In contrast to standard

approaches, we adapt a nonstandard marginalization in which the posterior density of the continuous states is represented using weighted random samples. In this way, the conditionally analytic part of the resulting PF is dedicated to solving the difficult association problem. In simulations, we compare the performance of the proposed method with its alternatives. The algorithm can achieve a similar performance of a marginalized particle filter that relies on the optimal proposal density, but it requires far less computational power. Furthermore, the proposed method can deal with nonlinearity and non-Gaussianity in ETT models without involving model approximations or linearizations.

References

[1] Blackman S, Popoli R. Design and Analysis of Modern Tracking Systems. Boston, MA, USA: Artech House, 1999.

[2] Bar-Shalom Y, Li XR. Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, CT, USA: YBS Publishing, 1995.

[3] Baum M, Klumpp V, Hanebeck UD. A novel Bayesian method for fitting a circle to noisy point. In: 13th International Conference on Information Fusion; Edinburgh, UK; 2010. pp. 1-6.

[4] Granström K, Lundquist C. On the use of multiple measurement models for extended target tracking. In: 16th International Conference on Information Fusion; İstanbul, Turkey; 2013. pp. 1534-1541.

[5] Granström K, Lundquist C, Orguner U. Tracking rectangular and elliptical extended targets using laser measurements. In: 14th International Conference on Information Fusion; Chicago, IL, USA; 2011. pp. 1-8.

[6] Koch JW. Bayesian approach to extended object and cluster tracking using random matrices. IEEE Transactions on Aerospace and Electronic Systems 2008; 44 (3): 1042-1059. doi: 10.1109/TAES.2008.4655362

[7] Feldman M, Franken D, Koch JW. Tracking of extended objects using random matrices. IEEE Transactions on Signal Processing 2011; 59 (4): 1409-1420. doi: 10.1109/TSP.2010.2101064

[8] Baum M, Hanebeck UD. Random hypersurface models for extended object tracking. In: 2009 IEEE International Symposium on Signal Processing and Information Technology; Ajman, United Arab Emirates; 2009. pp. 178-183.

[9] Baum M, Hanebeck UD. Shape tracking of extended objects and group targets with star-convex RHMs. In: 14th International Conference on Information Fusion; Chicago, IL, USA; 2011. pp. 1-8.

[10] Wahlström N, Özkan E. Extended target tracking using Gaussian processes. IEEE Transaction on Signal Processing 2015; 63 (16): 4165-4178. doi: 10.1109/TSP.2015.2424194

[11] Özkan E, Wahlström N, Godsill J. Rao-Blackwellised particle filter for star-convex extended target models. In: 19th International Conference on Information Fusion; Heidelberg, Germany; 2016. pp. 1193-1199.

[12] Kara SF, Özkan E. Multi-ellipsoidal extended target tracking using sequential Monte Carlo. In: 21st International Conference on Information Fusion; Cambridge, UK; 2018. pp. 1-8.

[13] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing 2000; 10 (3): 197-208. doi: 10.1023/A:1008935410038

[14] Schön T, Gustafsson F, Nordlund PJ. Marginalized particle filters for mixed linear/nonlinear state-space models. IEEE Transactions on Signal Processing 2005; 53 (7): 2279-2289. doi: 10.1109/TSP.2005.849151

[15] Özkan E, Lindsten F, Fritsche C, Gustafsson F. Recursive maximum likelihood identification of jump Markov nonlinear systems. IEEE Transactions on Signal Processing 2015; 63 (3): 754-765. doi: 10.1109/TSP.2014.2385039

[16] Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing 2002; 50 (2): 174-188. doi: 10.1109/78.978374

[17] Li XR, Jilkov VP. Survey of maneuvering target tracking, Part I, Dynamic models. IEEE Transactions on Aerospace and Electronic Systems 2003; 39 (4): 1333-1364. doi: 10.1109/TAES.2003.1261132