# MANEUVERING TARGET TRACKING WITH SIMPLIFIED COST REFERENCE PARTICLE FILTERS

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

Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11794, USA
e-mail: shaxu,monica,djuric@ece.sunysb.edu

**ABSTRACT**

In this paper, we investigate different variants of the recently proposed cost-reference particle filters (CRPFs) and study their application to the problem of tracking of a high-speed maneuvering target in the two-dimensional space. CRPFs drop all probabilistic assumptions required by conventional particle filters and, as a consequence, lead to practically more robust algorithms. We introduce some suitable and natural modifications of CRPFs in order to increase their efficiency and reduce their computational complexity. Computer simulations are provided to illustrate the performance of the new alternatives.

## 1. INTRODUCTION

The online tracking of a maneuvering target is a highly nonlinear and challenging problem that involves, at every time instant, the estimation not only of the unknown state (composed of position, velocity and acceleration of the target) in the dynamic model that describes the evolution of the target, but also the underlying model that accounts for the regime of movement [1]. This problem has been recently addressed by using a new family of sequential Monte Carlo (SMC) methods called cost reference particle filters (CRPFs) [2].

The CRPFs, unlike standard particle filters (SPFs) [3], aim at the estimation of the system state from the available observations without a priori knowledge of any probability density function [4]. The statistical reference 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 conventional particle filters, whose theory is based on probabilistic assumptions.

In this paper, we present some variations of the original CRPFs that result from, on one hand, the theoretical interpretation of the new methodology as a generalization of the standard particle filters (SPFs) and, on the other hand, the search for more efficient and computationally less demanding algorithms. The proposed algorithms are carefully designed to cope with the multiple models involved in the maneuvering target tracking problem. The starting point is the analytical relationship of CRPFs with SPFs. The conditions for which a SPF becomes a special case of the CRPF are discussed and lead to a first simplified version of the method. Other issues like resampling and estimation strategies are also addressed and new schemes are proposed to further simplify the algorithms in their design and implementation.

This work has been supported by the National Science Foundation under Awards CCR-0220011 and CCF-0515246 and the Office of Naval Research under Award N00014-06-1-0012.

The remaining of this paper is organized as follows. The problem formulation of tracking a maneuvering target moving along the two-dimensional space is described in Section 2. The fundamentals of the CRPFs' family are briefly introduced in Section 3. We discuss the conditions that relate CRPFs and SPFs and present some variants of the CRPFs in Sections 4 and 5, respectively. Computer simulation results that illustrate the validity of the proposed algorithms are presented in Section 6. Finally, Section 7 is devoted to the conclusions.

## 2. PROBLEM FORMULATION

The system state consists of the target position, $\mathbf{p}_t = [p_{x,t}, p_{y,t}]^\top$ (m), velocity, $\mathbf{v}_t = [v_{x,t}, v_{y,t}]^\top$ (m/s), and acceleration, $\mathbf{a}_t = [a_{x,t}, a_{y,t}]^\top$ (m/s$^2$), in the $xy$-plane [5]. We collect these magnitudes in a single state vector of the form $\mathbf{x}_t = [\mathbf{p}_t^\top, \mathbf{v}_t^\top, \mathbf{a}_t^\top]^\top \in \mathbb{R}^6$ and represent the dynamic system as

$$\mathbf{x}_t = \mathbf{A}_{m_t}\mathbf{x}_{t-1} + \mathbf{B}_{m_t}\mathbf{u}_{m_t,t} \quad m_t = 1, 2, \tag{1}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{w}_t. \tag{2}$$

The state equation (1) is allowed to switch between two different modes of operation, i.e., $m_t \in \{1, 2\}$. The nearly constant velocity model, $m_t = 1$, is described by

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{I}_2 & T_s\mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix}, \qquad \mathbf{B}_1 = \begin{bmatrix} \frac{1}{2}T_s^2\mathbf{I}_2 \\ T_s\mathbf{I}_2 \\ \mathbf{0}_2 \end{bmatrix}$$

while the accelerating model, $m_t = 2$, designed to track occasional maneuvering motion is given by

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{I}_2 & T_s\mathbf{I}_2 & \frac{1}{2}T_s^2\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 & T_s\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}, \qquad \mathbf{B}_2 = \begin{bmatrix} \frac{1}{6}T_s^3\mathbf{I}_2 \\ \frac{1}{2}T_s^2\mathbf{I}_2 \\ T_s\mathbf{I}_2 \end{bmatrix}$$

where $T_s$ is the sampling period, and $\mathbf{I}_2$ and $\mathbf{0}_2$ represent the $2 \times 2$ identity matrix and zero matrix, respectively. Switching between models occurs randomly, according to the transition probability matrix

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$$

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 $t$, $i, j = 1, 2$, and the initial model probabilities are set to $p(m_t = 1) = p$ and $p(m_t = 2) = 1 - p$. Finally, the state noise $\mathbf{u}_{m_t,t} \in \mathbb{R}^2$ is a zero-mean white Gaussian process used to model small acceleration turbulence,

$$\mathbf{u}_{m_t,t} \sim \mathcal{N}(\mathbf{0}, \sigma_{m_t}^2\mathbf{I}_2), \quad m_t = 1, 2$$

where the variances $\sigma^2_{m_t}$ depend on the model.

The observation function $h(\cdot)$ has four components. An emitter on the moving target transmits a signal with initial power $P_0$ through a fading channel with attenuation coefficient $\alpha$. The transmitted signal power at three reference points and the relative angle between the target and one of the references (which acts as a fusion center) are measured, i.e.,

$$\begin{aligned} h_j(\mathbf{x}_t) &= 10\log_{10}\left(\frac{P_0}{\|\mathbf{r}_j - \mathbf{p}_t\|^\alpha}\right) \quad j = 1,2,3 \\ h_4(\mathbf{x}_t) &= \angle(\mathbf{p}_t) \end{aligned}$$

where $\mathbf{r}_j = [r_{x,j}\ r_{y,j}]^\top$, $j = 1,2,3$, denote the position of the reference points, and $\|\mathbf{z}\| = \sqrt{\mathbf{z}^\top\mathbf{z}}$ is the norm of vector $\mathbf{z}$. The observation noise, $\mathbf{w}_t$, is also modeled as a Gaussian noise with independent components, $w_{j,t} \sim \mathcal{N}(0, \sigma^2_{w_j})$.

The objective is the adaptive estimation of the target state $\mathbf{x}_{0:t}$ given the sequence of measurements $\mathbf{y}_{1:t}$.

## 3. COST REFERENCE PARTICLE FILTERING

In order to estimate the state of the system from the available observations without use of any probabilistic assumptions about the model (1)-(2), we define a real *cost* function

$$\mathcal{C}_t = \mathcal{C}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}, \lambda) = \lambda\mathcal{C}(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) + \triangle\mathcal{C}(\mathbf{x}_t|\mathbf{y}_t)$$

which measures the quality of the state sequence $\mathbf{x}_{0:t}$ given the sequence of observations, $\mathbf{y}_{1:t}$. A recursive structure is assumed and the cost of a sequence at time $t$ is calculated using the cost up to time $t-1$ weighted by a forgetting factor, $\lambda$, plus a cost increment, $\triangle C(\mathbf{x}_t|\mathbf{y}_t)$, obtained from the state and observation vectors at time $t$. An additional *risk* function given by

$$\mathcal{R}(\mathbf{x}_{t-1}|\mathbf{y}_t) = \triangle\mathcal{C}\left(f_x(\mathbf{x}_{t-1})|\mathbf{y}_t\right),$$

represents a prediction or estimate of the cost increment, $\triangle\mathcal{C}(\mathbf{x}_t|\mathbf{y}_t)$ obtained before $\mathbf{x}_t$ is actually propagated.

Risk and cost are the main concepts in the design of the CRPFs, which proceed sequentially in a manner similar to the SPFs [3]. The essential structure of the algorithm is outlined in Table 1, where we have included some changes with respect to the original method in [4] to account for the multiple models existing in the maneuvering target tracking problem [2]. Specifically, given $M$ particles at time $t$, $2M$ risks are evalued at time $t+1$ (one for each model and each particle), but only $M$ trajectories survive the resampling step[1]. Note that a monotonically decreasing function $\mu(\cdot) : \mathbb{R} \rightarrow [0, +\infty]$ is necessary to accomplish both the resampling and estimation steps (see Section 6 for the specific formulation considered in the maneuvering target tracking problem). We will see that several strategies can be used to avoid working with it, and therefore reduce the complexity of the algorithms. More detailed guidelines that assist the algorithm designer, as well as sufficient conditions for the asymptotic convergence of the propagation step, are described in [2, 4].

[1]Note that the trajectory estimates in step 3 are obtained by weighted average over the two models. An alternative is to use estimates based on the model that has maximum posterior probability or to simply state the estimates based on each of the two models together with the posterior probabilities of the models.

**Initialization**
For $i = 1, ..., M$
$\quad \mathbf{x}_0^{(i)} \sim \mathcal{U}(I_{\mathbf{x}_0})$
$\quad \mathcal{C}_0^{(i)} = 0$
$\quad \sigma_0^{2,(i)}$, this variance is not updated until $t > 10$

**Recursive update**
For $t = 1$ to $T$
*1. Selection of the promising trajectories*
For $i = 1, ..., M$
$\quad \mathcal{R}_{t+1}^{(i,m_{t+1})} = \lambda\mathcal{C}_t^{(i)} + \mathcal{R}(\mathbf{x}_t^{(i)}|\mathbf{y}_{t+1}, m_{t+1}) \quad m_{t+1} = 1, 2$
$\quad \hat{\pi}_{t+1}^{(i,m_{t+1})} \propto \mu(\mathcal{R}_{t+1}^{(i,m_{t+1})})$
Resample to obtain $\left\{\hat{\mathbf{x}}_t^{(i)}, \hat{\mathcal{C}}_t^{(i)}, m_{t+1}^{(i)}\right\}_{i=1}^M$

*2. Random propagation*
For $i = 1, ..., M$
$\quad \mathbf{x}_{t+1}^{(i)} \sim p_{t+1}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_t^{(i)}, m_{t+1}^{(i)})$
$\quad$ If $t > 10$, update the propagation variance
$\quad \mathcal{C}_{t+1}^{(i)} = \lambda\hat{\mathcal{C}}_t^{(i)} + \triangle\mathcal{C}_{t+1}(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{t+1})$
*3. State estimation*
$\tilde{\pi}_{t+1}^{(i)} = \mu(\mathcal{C}_{t+1}^{(i)})$
$\pi_{t+1}^{(i)} = \frac{\tilde{\pi}_{t+1}^{(i)}}{\sum_{j=1}^M \tilde{\pi}_{t+1}^{(j)}}$
$\mathbf{x}_{t+1}^{est} = \sum_{i=1}^M \mathbf{x}_{t+1}^{(i)}\pi_{t+1}^{(i)}$

**Table 1**. Multiple model CRPF

## 4. RELATIONSHIP BETWEEN THE CRPF AND THE SPF

As mentioned in the introduction, CRPFs do not use the usual probabilistic assumptions common to SPFs regarding the knowledge of the *a priori* probability density funtion (pdf) of the state signal, $p(\mathbf{x}_0)$, and the noise densities, $p(\mathbf{u}_{m_t,t})$ and $p(\mathbf{w}_t)$. They also do not use the information regarding the transition probability matrix $\mathbf{H}$ needed for the multiple model system. CRFPs are indeed a generalization of the SPFs and the conditions that relate both families can be easily found. Specifically, if the recursive cost of a CRPF is computed by

$$\mathcal{C}(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t}, \lambda) = \lambda\mathcal{C}(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1}, \lambda) + \triangle\mathcal{C}(\mathbf{x}_t^{(i)}|\mathbf{y}_t) \quad (3)$$

and $\lambda = 1$, the conditions are:

(a) $\triangle\mathcal{C}(\mathbf{x}_t^{(i)}|\mathbf{y}_t) = -\ln p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$,

(b) the CRPF and the SPF use the same proposal pdf, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and

(c) the CRPF assigns weights to the streams of particles according to $w_{0:t}^{(i)} \propto e^{-\mathcal{C}_{0:t}^{(i)}}$.

With minor additional adjustments of the CRPF, we claim that the CRPF and the SPF will produce identical results. Thus, we can say that the SPF that resamples at every time instant is a special case of the CRPF.

The other extreme would be if we set $\lambda = 0$. In this case, the corresponding SPF algorithm has to perform resampling at every time instant (as would the CRPF algorithm), and

$$\mathcal{C}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) = \triangle\mathcal{C}(\mathbf{x}_t^{(i)}|\mathbf{y}_t). \quad (4)$$

| CRPF$_{\lambda=0}$ | CRPF$_{\lambda=0,nr}$ | CRPF$^{min}_{\lambda=0,nr}$ |
|---|---|---|
| **1. Select:**<br>For $i=1,\cdots,M, \quad m_{t+1}=1,2$<br>$\mathcal{R}_{t+1}^{(i,m_{t+1})}=\mathcal{R}(\mathbf{x}_t^{(i)}\|\mathbf{y}_{t+1},m_{t+1})$<br>$\hat{\pi}_{t+1}^{(i,m_{t+1})}\propto\mu(\mathcal{R}_{t+1}^{(i,m_{t+1})})$<br>Resample according to $\hat{\pi}_{t+1}^{(i,m_{t+1})}$<br><br>**2. Propagate:**<br>For $i=1,\cdots,M$<br>$\mathbf{x}_{t+1}^{(i)}\sim p_{t+1}(\mathbf{x}\|\hat{\mathbf{x}}_t^{(i)},m_{t+1}^{(i)})$<br>$\mathcal{C}_{t+1}^{(i)}=\triangle\mathcal{C}_{t+1}(\mathbf{x}_{t+1}^{(i)}\|\mathbf{y}_{t+1})$<br><br>**3. Estimate:**<br>$\pi_{t+1}^{(i)}\propto\mu(\mathcal{C}_{t+1}^{(i)}) \quad i=1,\dots,M$<br>$\mathbf{x}_{t+1}^{mean}=\sum_{i=1}^{M}\mathbf{x}_{t+1}^{(i)}\pi_{t+1}^{(i)}$ | **1. Select:**<br>For $i=1,\cdots,M, \quad m_{t+1}=1,2$<br>$\mathcal{R}_{t+1}^{(i,m_{t+1})}=\mathcal{R}(\mathbf{x}_t^{(i)}\|\mathbf{y}_{t+1},m_{t+1})$<br>Sort in increasing order the $\mathcal{R}_{t+1}^{(i,m_{t+1})}$<br>Replicate the $\frac{M}{N}$ trajectories<br><br>**2. Propagate:**<br>For $i=1,\cdots,M$<br>$\mathbf{x}_{t+1}^{(i)}\sim p_{t+1}(\mathbf{x}\|\hat{\mathbf{x}}_t^{(i)},m_{t+1}^{(i)})$<br>$\mathcal{C}_{t+1}^{(i)}=\triangle\mathcal{C}_{t+1}(\mathbf{x}_{t+1}^{(i)}\|\mathbf{y}_{t+1})$<br><br>**3. Estimate:**<br>$\pi_{t+1}^{(i)}\propto\mu(\mathcal{C}_{t+1}^{(i)}) \quad i=1,\dots,M$<br>$\mathbf{x}_{t+1}^{mean}=\sum_{i=1}^{M}\mathbf{x}_{t+1}^{(i)}\pi_{t+1}^{(i)}$ | **1. Select:**<br>For $i=1,\cdots,M, \quad m_{t+1}=1,2$<br>$\mathcal{R}_{t+1}^{(i,m_{t+1})}=\mathcal{R}(\mathbf{x}_t^{(i)}\|\mathbf{y}_{t+1},m_{t+1})$<br>Sort in increasing order the $\mathcal{R}_{t+1}^{(i,m_{t+1})}$<br>Replicate the $\frac{M}{N}$ trajectories<br><br>**2. Propagate:**<br>For $i=1,\cdots,M$<br>$\mathbf{x}_{t+1}^{(i)}\sim p_{t+1}(\mathbf{x}\|\hat{\mathbf{x}}_t^{(i)},m_{t+1}^{(i)})$<br>$\mathcal{C}_{t+1}^{(i)}=\triangle\mathcal{C}_{t+1}(\mathbf{x}_{t+1}^{(i)}\|\mathbf{y}_{t+1})$<br><br>**3. Estimate:**<br>$\mathbf{x}_{t+1}^{min}=\arg\min\{\mathcal{C}_{t+1}^{(i)}\}$ |

**Fig. 1**. Alternative implementations to the multiple model CRPF.

Note that $\lambda = 0$ not only affects the calculation of the costs but also the risks. The CRPF and the SPF will yield identical results if conditions (a) and (b) from above still hold, and $w_t^{(i)} \propto e^{-\mathcal{C}_t^{(i)}}$. Again, we claim that SPF is a particular case of CRPF.

Since our objective it to simplify the algorithms, we study the case $\lambda = 0$. Therefore, we update the costs using exclusively the incremental costs, i.e., according to (4). This method will be denoted as CRPF$_{\lambda=0}$.

## 5. ALTERNATIVE CRPFS

In this section we discuss some alternatives in the implementation of the CRPF in order to reduce its computational complexity maintaining the same efficiency. As mentioned in Section 3, one of the critical steps in the complexity of the algorithm outlined in Table 1 is the calculation of the function $\mu(\cdot)$. Here we address solutions that do not use function $\mu(\cdot)$.

### 5.1. Avoidance of resampling

In standard particle filtering, resampling must always be applied to avoid weight degeneracy, the only choice being the frequency of its use [6, 7]. It may be applied after the processing of every observation, periodically (with period greater than one), or when necessary (where the necessity is determined by an appropriately selected criterion). The last two options are of interest when the particle filters are run on general purpose computers that implement the particle filtering sequentially. However, resampling is a major obstacle for efficient implementation of SPF in parallel VLSI hardware devices, because it creates full data dependencies among processing units [6]. Although some promising methods have been recently proposed [6], parallelization of resampling algorithms remains an open problem.

The selection step in CRPFs is much less restrictive than resampling in conventional SPFs [4]. Specifically, while resampling methods in SPFs must ensure that the probability distribution of the resampled population is an unbiased and unweighted approximation of the original distribution of the particles [8], selection in CRPFs is only aimed at ensuring that the particles are close to the locations that produce cost function minima. This issue has already been exploited to propose a *local resampling* scheme suitable for a straightforward implementation using parallel VLSI hardware [4].

Here we propose an alternative approach where the proposed $CRPF_{\lambda=0}$ scheme is modified to avoid resampling, which is replaced by simple ordering of the obtained risks in step 1 of the algorithm and replicating the corresponding $\frac{M}{N}$ (where $N$ can be $2,3,\cdots$) particles with lowest risks. Therefore, in the first step we avoid both the calculation of $\mu(\cdot)$ and the resampling procedure. This method will be denoted by $CRPF_{\lambda=0,nr}$.

### 5.2. Different estimation procedures

According to Table 1, the estimation step in the CRPF would require the calculation of a probability mass function (pmf) $\mu(\cdot)$. The purpose of this pmf is to somehow normalize the obtained risks. In order to avoid this operation, we modify the previously introduced $CRPF_{\lambda=0,nr}$ by choosing as estimate of the state the particle with minimum cost. The resulting algorithm is symbolized by $CRPF^{min}_{\lambda=0,nr}$.

A summary of all the proposed algorithms is shown in Figure 1.

## 6. COMPUTER SIMULATIONS

In this section, we present simulation results that illustrate the performance of the various tracking algorithms discussed in the previous sections. In our experiment, the target started from an unknown position close to $[0, 0]$, and evolved according to the motion equation (1) for 10 minutes with sampling period $T_s = 5$ seconds. The state noise variances were set to $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 0.01$ and the observation noise components were modeled as $w_{1,t} \sim \mathcal{N}(0, 10)$, $w_{2,t} \sim \mathcal{N}(0, 10)$, $w_{3,t} \sim \mathcal{N}(0, 10)$, and $w_{4,t} \sim \mathcal{N}(0, 1)$. The transition probability matrix, used for switching between models in (1), was

$$\mathbf{H} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix},$$

with initial model probabilities $p(1) = 0.9$ and $p(2) = 0.1$.

The cost increment and risk functions used in the implementation of the CRPFs were

$$\begin{aligned}\triangle\mathcal{C}(\mathbf{x}_t|\mathbf{y}_t) &= \|\mathbf{y}_t - h(\mathbf{x}_t)\|^2 \\ \mathcal{R}(\mathbf{x}_t^{(i)}|\mathbf{y}_{t+1}, m_{t+1}) &= \|\mathbf{y}_{t+1} - h(\mathbf{A}_{m_{t+1}}\mathbf{x}_t^{(i)})\|^2, \quad m_{t+1} = 1, 2;\end{aligned}$$

and the function $\mu$ in the selection step[2], if used, was defined as

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

with $\delta = \frac{1}{10}\sqrt{var(\mathcal{R}_{t+1}^{(i,m_{t+1})})}$ and $\beta = 2$. For the original CRPF, the forgetting factor, $\lambda$ was set to 0.95. The propagation mechanism was a Gaussian density of the form

$$\mathbf{x}_{t+1}^{(i)} \sim \mathcal{N}\left(\mathbf{A}_{m_{t+1}^{(i)}}\hat{\mathbf{x}}_t^{(i)},\ \sigma_t^{2,(i)}\mathbf{B}_{m_{t+1}^{(i)}}\mathbf{B}_{m_{t+1}^{(i)}}^\top\right),$$

where the variance, $\sigma_t^{2,(i)}$, was adaptively computed as

$$\sigma_t^{2,(i)} = \frac{t-1}{t}\sigma_{t-1}^{2,(i)} + \frac{1}{tL_u}\left\|\mathbf{B}_{m_t^{(i)}}^\top\left(\mathbf{x}_t^{(i)} - \mathbf{A}_{m_t^{(i)}}\hat{\mathbf{x}}_{t-1}^{(i)}\right)\mathbf{B}^{(m_t^{(i)})}\right\|^2$$

with initial value $\sigma_0^{2,(i)} = 0.5$.

For comparison purposes, we also implemented the multiple-model SPF [1], which used the same model propagation scheme as the CRPFs. All the particle filters ran using $M = 1000$ particles; the CRPF$_{\lambda=0,nr}$ was run with two different values to discriminate the particles, $N = 2$ and $N = 4$.

The performance of the algorithms was compared by means of the percentage of track loss. At a given simulation run, a track loss was confirmed when the root-mean-square (RMS) position error exceeded a threshold $\gamma_{max}$ within at least 10 consecutive sampling periods. The RMS position error was computed according to

$$RMS_t = \sqrt{(x_{1,t}^{est} - x_{1,t})^2 + (x_{2,t}^{est} - x_{2,t})^2}$$

where $[x_{1,t}\ x_{2,t}]^\top$ was the true position of the target at time $t$ in that run, and $[x_{1,t}^{est}\ x_{2,t}^{est}]^\top$ was the corresponding estimate obtained by the filter. The percentages were calculated averaging over 500 independent simulation trials.

Figure 2 shows the track loss of the algorithms with respect to different values of the threshold $\gamma_{max}$. We can see that the proposed CRPFs work similarly as the SPF. Note that the generation of the particles was done according to the probabilistic models used by the SPF. If there were discrepancies between the model assumptions of the state/measured data and the true models, the differences in performance in favor of the CRPF could have been significant [2, 4].

Finally, Figure 3 depicts the execution time of the compared algorithms. It can be seen that the CRPF without resampling is faster than the SPF and the original CRPF by five times.

## 7. CONCLUSIONS

We have investigated several modifications to the originally presented cost-reference particle filters (CRPFs) and we have applied them to the problem of tracking a maneuvering target in the two-dimensional space. The proposed variations avoid the resampling step (bottleneck in standard particle filtering) and the use of the monotonically decreasing function $\mu(\cdot)$. As a result, computationally less demanding algorithms are obtained with similar accuracy.

[2] An analogous formulation of $\mu(\mathcal{C}_t^{(i)})$ was used.

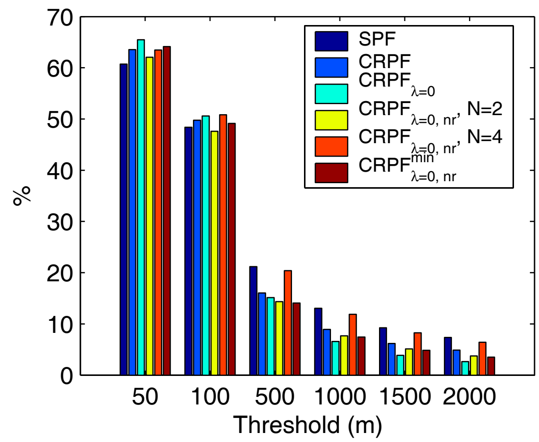


**Fig. 2**. Percentages of track loss.

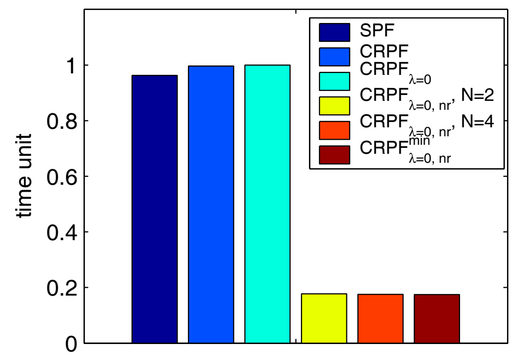


**Fig. 3**. Execution time of the algorithms.

## 8. REFERENCES


[1] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 1006–1012, 2000.

[2] M. F. Bugallo, S. Xu, J. Míguez, and P. M. Djurić, "Maneuvering target tracking using cost reference particle filtering," in *IEEE Proceedings of ICASSP*, 2004, vol. 3.

[3] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[4] J. Míguez, M. F. Bugallo, and P. M. Djurić, "A new class of particle filters for random dynamical systems with unknown statistics," *EURASIP Journal of Applied Signal Processing*, 2004.

[5] F. Gustafsson, F. Gunnarsson, N. Bergman, et al., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[6] M. Bolić, P. M. Djurić, and S. Hong, "New resampling algorithms for particle filters," in *IEEE Procddings of ICASSP*, 2003.

[7] A. Doucet, S. Godsill, and C. Anderieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[8] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on signal processing*, vol. 50, no. 3, pp. 736–746, 2002.