MANEUVERING TARGET TRACKING USING COST REFERENCE PARTICLE FILTERING

Mónica F. Bugallo[†], *Shanshan Xu*[†], *Joaquín Míguez*[‡], *Petar M. Djurić*[†]

[†]Department of Electrical and Computer Engineering, Stony Brook University, NY 11794, USA
[‡]Dept. Electrónica e Sistemas, Univ. da Coruña, Campus de Elviña s/n, 15071 A Coruña, Spain
e-mail: monica,shaxu,djuric@ece.sunysb.edu jmiguez@udc.es

ABSTRACT

Target tracking is a highly nonlinear problem that has been successfully addressed in recent years using sequential Monte Carlo (SMC) methods, usually called particle filters. In this paper, we investigate the application of a new class of SMC techniques, termed cost-reference particle filters (CRPFs), to tracking of a high-speed maneuvering target. The new CRPF methodology drops all probabilistic assumptions (i.e., prior probabilities, knowledge of noise distributions and likelihood functions) that are common to conventional particle filters and, as a consequence, leads to practically more robust algorithms. The advantage of the proposed CRPF over the standard SMC filter in the context of maneuvering target tracking is illustrated through computer simulations.

1. INTRODUCTION

In recent years particle filtering has attracted significant attention in the signal processing community [2, 5] and one of its most outstanding and successful applications has been target tracking [3, 6, 4]. The primary reason is the flexibility and accuracy of particle filters in resolving very difficult nonlinear problems where the underlying models are represented by dynamic state-space equations. In these models, one equation describes the stochastic Markovian evolution of the system states in time, while the other equation quantifies the observations as (possibly nonlinear) functions of the unobserved states. All existing particle filtering methods require a mathematical representation of the dynamics of the system evolution that includes assumptions of probabilistic models.

In this paper, we investigate the application to target tracking of a new class of particle filtering methods introduced in [7]. The main feature of the new techniques is that they are not based on any particular probabilistic assumptions regarding the dynamical model of the target. 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. Hence, methods within this class are termed Cost Reference Particle Filters (CRPFs), in contrast to conventional Statistical Reference Particle Filters (SRPFs).

The CRPF methodology is applied to the problem of tracking maneuvering targets. Standard tracking algorithms are 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 dynamic models, they can maneuver, which implies that tracking cannot be successful if it is done with one single model. By the inclusion of several models [6], coping with maneuvering becomes possible, but the overall tracking is much more challenging. From a signal processing point of view, this is a very interesting problem because we have to estimate the underlying model at every time instant, as well as its unknown state.

The fundamentals of the CRPF approach are introduced in Section 2. In Section 3, we apply the proposed algorithms to tracking of a maneuvering target. Finally, brief concluding remarks are made in Section 4.

J. Míguez acknowledges the support of *Ministerio de Ciencia y Tecnología* of Spain and *Xunta de Galicia* (project TIC2001-0751-C04-01). S. Xu, M. F. Bugallo and P. M. Djurić acknowledge the support of the National Science Foundation (Award CCR-0082607).

2. COST REFERENCE PARTICLE FILTERING

2.1. Nonlinear state-space dynamic systems

Many problems in signal processing can be stated in terms of estimation of an unobserved discrete-time random signal in a dynamic system of the form

$$\mathbf{x}_t = f_x(\mathbf{x}_{t-1}) + \mathbf{u}_t \quad \text{state equation} \tag{1}$$
$$\mathbf{y}_t = f_y(\mathbf{x}_t) + \mathbf{v}_t, \quad \text{observation equation} \tag{2}$$

where $t = 1, 2, ...$ denotes discrete time, $\mathbf{x}_t \in \mathbb{R}^{L_x}$ is the signal of interest, that represents the system state at time t; $f_x : \mathbb{R}^{L_x} \rightarrow I_x \subseteq \mathbb{R}^{L_x}$ is a (possibly nonlinear) state transition function; $\mathbf{u}_t \in \mathbb{R}^{L_x}$ is the state perturbation or system noise at time t; $\mathbf{y}_t \in \mathbb{R}^{L_y}$ is the vector of observations collected at time t, which depends on the system state; $f_y : \mathbb{R}^{L_x} \rightarrow I_y \subseteq \mathbb{R}^{L_y}$ is a (possibly nonlinear) transformation of the state; and $\mathbf{v}_t \in \mathbb{R}^{L_y}$ is the observation noise vector at time t, assumed statistically independent from the system noise $\mathbf{u}_t$.

The ultimate aim is the online estimation of the sequence of system states from the available observations, i.e., we intend to estimate $\mathbf{x}_t|\mathbf{y}_{1:t}$, $t = 0, 1, 2, ...$

2.2. Sequential Algorithm

In order to estimate $\mathbf{x}_{0:t}$ from $\mathbf{y}_{1:t}$ without the need of any probabilistic assumption on model (1)-(2), we consider a user-defined real *cost* function,

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

that measures the quality of the state sequence $\mathbf{x}_{0:t}$ given the sequence of observations, $\mathbf{y}_{1:t}$. We assume a recursive structure where the cost of a sequence up to time $t-1$ can be updated by looking solely at the state and observation vectors at time t, $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively, which are used to compute the cost increment, $\triangle C(\mathbf{x}_t|\mathbf{y}_t)$. The parameter λ is a forgetting factor that avoids attributing an excessive weight to *old* observations when a long series of data are collected, hence allowing for potential adaptivity. We also consider a one-step *risk* function $\mathcal{R}(\mathbf{x}_{t-1}|\mathbf{y}_t)$ that measures the adequacy of the state at time $t-1$ given the new observation, $\mathbf{y}_t$. It is convenient to view the risk function, $\mathcal{R}(\mathbf{x}_{t-1}|\mathbf{y}_t)$, as a prediction or estimate of the cost increment, $\triangle\mathcal{C}(\mathbf{x}_t|\mathbf{y}_t)$, that can be obtained before $\mathbf{x}_t$ is actually propagated. Hence, a natural choice of the risk function is

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

The proposed estimation technique proceeds sequentially in a manner similar to the standard SMC filter [2]. Given a set of M state trajectories and associated costs up to time t, $\left\{\mathbf{x}_t^{(i)}, \mathcal{C}_t^{(i)}\right\}_{i=1}^M$ where $\mathcal{C}_t^{(i)} = \mathcal{C}(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t}, \lambda)$ and $\mathbf{x}_{0:t}^{(i)}$ is the sequence of state vectors leading to the i-th particle $\mathbf{x}_t^{(i)}$, the grid of state trajectories is randomly propagated when $\mathbf{y}_t$ is observed by taking the following steps:

1. **Selection of the most promising trajectories (resampling).** For $i = 1, 2, ..., M$, let

$$\begin{aligned} \mathcal{R}_{t+1}^{(i)} &= \lambda\mathcal{C}_t^{(i)} + \mathcal{R}(\mathbf{x}_t^{(i)}|\mathbf{y}_{t+1}) \\ \hat{\pi}_{t+1}^{(i)} &\propto \mu(\mathcal{R}_{t+1}^{(i)}) \end{aligned}$$

   where $\mu : \mathbb{R} \to [0, +\infty)$ is a monotonically decreasing function and $\hat{\pi}_{t+1} : \{1, ..., M\} \to [0, 1)$ is a probability mass function (pmf). A new particle filter is obtained by resampling the trajectories $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^M$ according to the pmf $\hat{\pi}_{t+1}^{(i)}$ and we denote it as $\left\{\hat{\mathbf{x}}_t^{(i)}, \hat{\mathcal{C}}_t^{(i)}\right\}_{i=1}^M$.

2. **Random propagation.** For $i = 1, ..., M$, let

$$\begin{aligned} \mathbf{x}_{t+1}^{(i)} &\sim p_{t+1}(\mathbf{x}|\hat{\mathbf{x}}_t^{(i)}) \\ \mathcal{C}_{t+1}^{(i)} &= \lambda\hat{\mathcal{C}}_t^{(i)} + \triangle\mathcal{C}_{t+1}^{(i)} \end{aligned}$$

   where $\triangle\mathcal{C}_{t+1}^{(i)} = \triangle\mathcal{C}_{t+1}(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_t)$ and p_{t+1} is a probability density function (pdf) chosen by the designer, which must verify $E_{p_{t+1}(\mathbf{x}|\hat{\mathbf{x}}_t^{(i)})}\mathbf{x}_{t+1}^{(i)} = f_x\left(\hat{\mathbf{x}}_t^{(i)}\right)$.

3. **Estimation of the state.** Let $\pi_{t+1}^{(i)} \propto \mu(\mathcal{C}_{t+1}^{(i)})$ for $i = 1, \ldots, M$. The function π_{t+1} is a pmf and we can obtain time $t+1$ state estimates in several ways, e.g.,

$$\mathbf{x}_{t+1}^{mean} = \sum_{i=1}^M \mathbf{x}_{t+1}^{(i)}\pi_{t+1}^{(i)}. \quad (3)$$

The general procedure described above is referred to as the CRPF. More detailed guidelines that assist the algorithm designer, as well as sufficient conditions for the asymptotic convergence of the propagation step, are described in [7].

3. MANEUVERING TARGET TRACKING

In this section we investigate the application of the CRPF method to the problem of tracking a maneuvering target along a 2-dimensional space. The target trajectory is characterized by a constant velocity with short periods of acceleration that correspond to maneuvers.

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. 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

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}_m\mathbf{x}_{t-1} + \mathbf{u}_t, \quad m = 1, 2, & (4) \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{w}_t. & (5) \end{aligned}$$

The state equation (4) is allowed to switch between two different modes of operation, m_1 and m_2, given by the 6×6 matrices

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

respectively, where T_s is the sampling period, and $\mathbf{I}_2$ and $\mathbf{0}_2$ represent the 2×2 identity matrix and zero matrix. Model m_2 is designed to track occasional maneuvering motion. Switching between models occurs randomly, according to the transition probability matrix

$$\mathbf{H} = \begin{bmatrix} 0.95 & 0.05 \\ 0.9 & 0.1 \end{bmatrix},$$

where $H_{ij} = p(m_j|m_i)$ is the probability of the system to switch from model m_i to model m_j, $i, j = 1, 2$, and the initial model probabilities are set to $p(m_1) = 0.9$ and $p(m_2) = 0.1$. Finally, the state noise process $\mathbf{u}_t$ is modeled as a 6×1 vector of independent mixture Gaussian random variables, i.e.,

$$\begin{aligned} u_{i,t} &\sim 0.1\mathcal{N}(0, 10) + 0.3\mathcal{N}(0, 1) + 0.6\mathcal{N}(0, 10^{-3}) \\ u_{j,t} &\sim 0.1\mathcal{N}(0, 1) + 0.3\mathcal{N}(0, 10^{-2}) + 0.6\mathcal{N}(0, 10^{-4}) \\ u_{k,t} &\sim 0.1\mathcal{N}(0, 10^{-3}) + 0.3\mathcal{N}(0, 10^{-6}) + 0.6\mathcal{N}(0, 10^{-8}) \end{aligned}$$

where $i = 1, 2$, $j = 3, 4$ and $k = 5, 6$.

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 α. At a fixed reference point, $\mathbf{r} = [r_x\ r_y]^\top$, the transmitted signal power is collected, as well as the relative angle between the target and the reference, the relative velocity and the direction of movement, i.e.,

$$\begin{aligned} h_1(\mathbf{x}_t) &= 10\log_{10}\left(P_0/||\mathbf{r} - \mathbf{p}_t||^\alpha\right) \\ h_2(\mathbf{x}_t) &= \angle\left(\mathbf{p}_t, \mathbf{r}_t\right) \\ h_3(\mathbf{x}_t) &= \sqrt{v_{x,t}^2 + v_{y,t}^2} \\ h_4(\mathbf{x}_t) &= \angle\left(\mathbf{v}_t\right) \end{aligned}$$

where $||\mathbf{z}|| = \sqrt{\mathbf{z}^\top\mathbf{z}}$ is the norm of $\mathbf{z}$. The observation noise, $\mathbf{w}_t$, is also modeled as a mixture Gaussian vector with independent components,

$$\begin{aligned} w_{1,t} &\sim 0.1\mathcal{N}(0, 5) + 0.4\mathcal{N}(0, 1) + 0.5\mathcal{N}(0, 10^{-2}) \\ w_{2,t} &\sim 0.1\mathcal{N}(0, 1) + 0.4\mathcal{N}(0, 0.1) + 0.5\mathcal{N}(0, 10^{-2}) \\ w_{3,t} &\sim 0.1\mathcal{N}(0, 1) + 0.4\mathcal{N}(0, 10^{-2}) + 0.5\mathcal{N}(0, 10^{-4}) \\ w_{4,t} &\sim 0.1\mathcal{N}(0, 1) + 0.4\mathcal{N}(0, 0.1) + 0.5\mathcal{N}(0, 10^{-3}). \end{aligned}$$

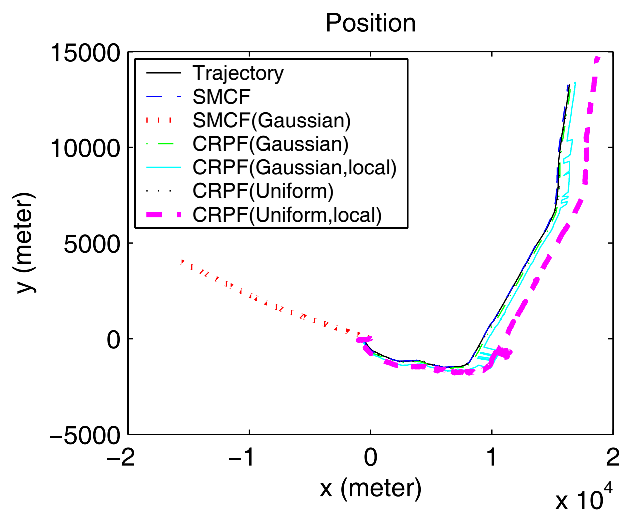

Fig. 1. CRPF vs. bootstrap filter with $M = 300$ particles. Sample trajectory.

We have applied the proposed CRPF to the adaptive estimation of the target track, $\mathbf{x}_{0:t}$, given the collected observations, $\mathbf{y}_{1:t}$. The CRPF, as described in subsection 2.2, is specified by

$$\begin{aligned} \mathcal{C}(\mathbf{x}_0) &= 0 \\ \triangle\mathcal{C}(\mathbf{x}_t|\mathbf{y}_t) &= \| \mathbf{y}_t - h(\mathbf{x}_t) \|^2 \\ \mathcal{R}_m(\mathbf{x}_t|\mathbf{y}_{t+1}) &= \| \mathbf{y}_{t+1} - h\left(\mathbf{A}_m\mathbf{x}_t\right)\|^2, \quad m = 1, 2 \\ \mu(\mathcal{C}_t^{(i)}) &= \frac{1}{\left(\mathcal{C}_t^{(i)} - \min_k \left\{\mathcal{C}_t^{(k)}\right\} + \delta\right)^{\beta}} \end{aligned}$$

where $\delta = 0.01$ and $\beta = 2$. The subindex in the risk function indicates that the two system models, m_1 and m_2, are explored in the selection step (note that the transition matrix $\mathbf{H}$ is part of the probabilistic model and, therefore, unknown to the CRPF). In particular, given M particles at time t, $2M$ risks are computed at time $t+1$ (one for each model and each particle), but only M trajectories are stochastically chosen at the selection step. Thus, both the trajectory to be propagated and the model, m_1 or m_2, to be used when generating the new particle are selected according to their risks. As for the propagation density (p_{t+1} in Section 2.2), we have considered both Gaussian and Uniform multidimensional pdfs (labeled *CRPF (Gaussian)* and *CRPF (Uniform)*, respectively) with independent components and adaptively chosen variances, i.e,

$$\sigma_{l,t}^{2\ (i)} = \frac{t-1}{t}\hat{\sigma}_{l,t-1}^{2\ (i)} + \frac{\left(\left[\mathbf{x}_t^{(i)}\right]_l - \left[f_x(\hat{\mathbf{x}}_{t-1}^{(i)})\right]_l\right)^2}{tL_x}, \qquad (6)$$

where $[\mathbf{z}]_l$ denotes the l-th element of vector $\mathbf{z}$, the ˆ indicates that the particles are already selected (i.e., resampled) and $l = 1, ..., 6$. The initial values are $\sigma_{1:2,0}^{2\ (i)} = 5$ for position, $\sigma_{3:4,0}^{2\ (i)} = 0.1$ for velocity and $\sigma_{5:6,0}^{2\ (i)} = 10^{-2}$ for acceleration. Table 1 summarizes the details of the considered algorithm where the forgetting factor, λ, is set to 0.95.

We have also considered two resampling schemes: the standard multinomial resampling [1] and an additional simple resampling technique (labeled as *CRPF (local)*) where resampling only occurs among *neighbor* particles (see [7] for details), which leads to a straightforward parallelization of the algorithm using an array of processors connected in a ring configuration.

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, for $i = 1, ..., M$
$\quad \mathcal{R}_{m,t+1} = \lambda\mathcal{C}_t^{(i)} + \left\|\mathbf{y}_{t+1} - h(\mathbf{A}_m\mathbf{x}_t^{(i)})\right\|^2, m = 1, 2$
$\quad \hat{\pi}_{m,t+1} \propto \mu(\mathcal{R}_{m,t+1}), m = 1, 2$
$\quad$ Resample using $\hat{\pi}_{m,t+1}$ to yield:
$\quad \left\{\hat{\mathbf{x}}_t^{(i)}, \hat{\mathcal{C}}_t^{(i)}, m^{(i)}\right\}_{i=1}^{M}, m^{(i)} \in \{1, 2\}$
$\quad \mathbf{x}_{t+1}^{(i)} \sim p_{t+1}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_t^{(i)}, m^{(i)})$
$\quad$ If $t > 10$,
$\quad\quad$ For each state component, update the propagation variance according to eq. (6).
$\quad \mathcal{C}_{t+1}^{(i)} = \lambda\hat{\mathcal{C}}_t^{(i)} + \left\|\mathbf{y}_{t+1} - h(\mathbf{x}_{t+1}^{(i)})\right\|^2$
State estimation
$\quad \tilde{\pi}_t^{(i)} = \mu(\mathcal{C}_t^{(i)})$
$\quad \pi_t^{(i)} = \frac{\tilde{\pi}_t^{(i)}}{\sum_{j=1}^{M} \tilde{\pi}_t^{(j)}}$
$\quad \mathbf{x}_t^{est} = \sum_{i=1}^{M} \mathbf{x}_t^{(i)}\pi_t^{(i)}$

Table 1. Sequential CRPF algorithm

Finally, for comparison and benchmarking purposes, we have implemented the popular SMC filter (SMCF) as proposed in [2, Chapter 23] for maneuvering target tracking. The SMCF has an algorithmic structure (resampling, importance sampling and state estimation) very similar to the proposed sequential CRPF family. We have considered two alternatives in the implementation of the SMCF: the algorithm obtained when the SMCF uses the true mixture Gaussian density (labeled as *SMCF*), and the *mismatched* algorithm that models the state noise processes with Gaussian densities: $\mathcal{N}(0, 1)$ for position, $\mathcal{N}(0, 10^{-2})$ for velocity, and $\mathcal{N}(0, 10^{-6})$ for acceleration (labeled as *SMCF (Gaussian)*). Table 2 summarizes the details of these algorithms.

Figure 1 shows the system trajectory in a single simulation run and the estimates corresponding to the SMCF and the CRPF algorithms. The trajectory starts in an unknown position close to $(0, 0)$ and evolves for 1 hour, with sampling period $T_s = 5$ s. It is apparent that all the algorithms, except the mismatched SMCF, remain locked to the vehicle position during the whole simulation interval.

The performance of the tracking algorithms is measured in terms of the mean absolute deviations obtained by averaging 50 independent simulation trials. The deviation signals are computed as

$$e_{k,t} = \frac{1}{50}\frac{1}{2}\sum_{j=1}^{50}\left(|x_{k,t,j} - x_{k,t,j}^{est}| + |x_{k+1,t,j} - x_{k+1,t,j}^{est}|\right)$$

where $k = 1$ for position, $k = 3$ for velocity and $k =$

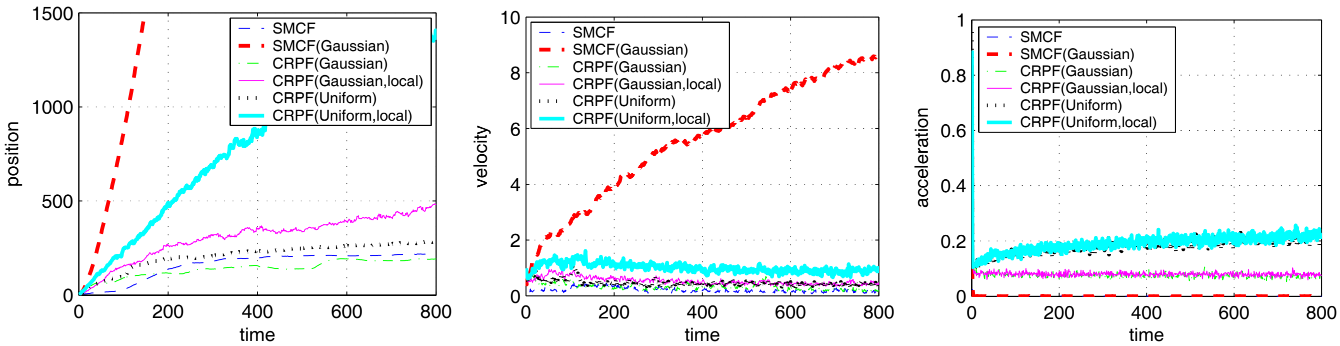

Fig. 2. CRPF vs. bootstrap filter with $M = 300$ particles. Left: Mean absolute deviation of position. Middle: Mean absolute deviation of velocity. Right: Mean absolute deviation of acceleration.

Initialization
For $i = 1, ..., M$
$\mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$m_0^{(i)} \sim P_0$
$w_0^{(i)} = \frac{1}{M}$
Recursive update
For $t = 1$ to T, for $i = 1, ..., M$
$\hat{m}_t^{(i)} \sim p(m|m_{t-1}^{(i)})$
$\hat{\mathbf{x}}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \hat{m}_t^{(i)})$
sample $\kappa_i = k$ with probability $p(\mathbf{y}_t|\hat{\mathbf{x}}_t^{(k)})$
$\mathbf{x}_t^{(i)} = \hat{\mathbf{x}}_t^{(\kappa_i)}$
$m_t^{(i)} = \hat{m}_t^{(\kappa_i)}$
$w_t^{(i)} = \frac{1}{M}$
State estimation
For $t = 1$ to T
$\mathbf{x}_t^{est} = \sum_{i=1}^{M} \mathbf{x}_t^{(i)} w_t^{(i)}$

Table 2. SMCF for the 2-dimensional tracking system.

5 for acceleration; j is the simulation number, hence $\mathbf{x}_{t,j} = [x_{1,t,j}, \ldots, x_{6,t,j}]^\top$ is the true state containing the position, velocity and acceleration at time t in j-th run, and $\mathbf{x}_{t,j}^{est} = [x_{1,t,j}^{est}, \ldots, x_{6,t,j}^{est}]^\top$ is the corresponding estimate obtained with a particle filter.

Figure 2 shows from left to right: (a) the mean absolute deviation in the estimated position, (b) the mean absolute deviation in the estimated velocity, and (c) the mean absolute deviation in the estimated acceleration. It is clear from the plots that the performance of the CRPF method is hardly affected by the choice of propagation density. On the contrary, the use of the correct statistical information is critical for the performance of the SMCF. Note that the CRPF algorithm labeled as *CRPF (Gaussian)* also draws the state particles from a Gaussian sequence of densities (the same as the mismatched SMCF), but it attains a superior performance compared to the SMCF. Also the CRPF with local resampling shows performance close to the SMCF with perfect knowledge of the noise statistics. Although it presents a slight degradation with respect to the CRPF with multinomial resampling, the feasibility of a simple parallel implementation makes the local resampling method extremely appealing.

4. CONCLUSIONS

We have investigated the use of a new class of particle filters, called CRPF [7], for tracking a maneuvering target in 2-dimensional space. CRPFs allow to drop the probabilistic assumptions required by conventional particle filters, hence leading to more flexible and robust algorithms. Computer simulations are provided to illustrate the performance of the CRPF approach when compared with the classical SMC filter.

5. REFERENCES

[1] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, March 2002.

[2] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, New York (USA), 2001.

[3] N. Gordon, "A hybrid bootstrap filter for target tracking in clutter," *IEEE Transactions on Aerospace Electronic Systems*, vol. 33, pp. 353–358, January 1997.

[4] C. Hue, J. P. Le Cadre, and P. Pérez, "Sequential monte carlo methods for multiple target tracking and data fusion," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 309–325, February 2002.

[5] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, September 1998.

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

[7] J. Míguez, M. F. Bugallo, and P. M. Djurić, "A new class of particle filters for random dynamical systems with unknown statistics," *Submitted to EURASIP Journal of Applied Signal Processing*, 2003.