

Cubature Particle Filter with MCMC and Applications to Re-entry Ballistic Target Tracking

Jing Mu and Yuan liCai

Abstract—Cubature particle filter with Markovian Chain Monte Carlo process (CPF-MC) is proposed in order to alleviate the degeneracy and impoverishment problems existing in the particle filter, and the CPF-MC algorithm is improved from two aspects. On the one hand, CPF-MC uses the square root cubature Kalman filter (SRCKF) as proposal distribution that integrates the latest measurement into the particle filter and approximates the optimal posterior probability distribution more accurately. On the other hand, after resampling, a SRCKF based Markovian Chain Monte Carlo (MCMC) process is used to make the particles diversity and suppress the impoverish phenomenon. The CPF-MC algorithm is applied to re-entry ballistic target tracking; simulation results demonstrate that the CPF-MC achieves the better performance and is superior to generic particle filter with MCMC (GPF-MC), extended particle filter with MCMC (EPF-MC) and unscented particle filter with MCMC (UPF-MC), furthermore, the CPF-MC run faster.

Index Terms—Markovian chain Monte Carlo, nonlinear filter, particle filter, re-entry ballistic target

I. INTRODUCTION

Particle filters (PFs), or the sequential Monte Carlo filters, are widely used for a wide range of target tracking problems in the nonlinear system since they can efficiently handle non-Gaussian and nonlinearity [1], [2]. The PFs have the superior characteristics of allowing for a complete representation of the posterior distribution of states, so that any statistical estimate, such as the mean, modes, kurtosis and variance, can be easily computed [3]. However, in some case, the state posterior distribution doesn't exit [4]. One method solved is using the proposal distribution. Generic particle filter (GPF) in [2] uses the transition prior as proposal function; a common problem in the GPF is the degeneracy phenomenon because the variance of the importance weights increases stochastically over time [4]. In order to solve the degeneracy problem, one strategy is to choose the proper proposal distribution that can approximate the posterior distribution reasonably well. The extended particle filter (EPF) in [5] and unscented particle filter (UPF) in [6] were proposed in which the extended Kalman filter (EKF) and unscented Kalman filter (UKF) were used to generate the proposal distributions, respectively. The

PF-based including incremental principle component analysis [7], the adaptive Gaussian mixture filter bridging the ensemble Kalman filter and PFs [8], and memory-based PF [9] were proposed. Their combination has shown superior performance in terms of accuracy and robustness, but it suffers from the heavy computational load. Another strategy to reduce the effects of the degeneracy problem is the resampling step, the particles that have high weights are statistically selected many times, this leads to a loss of diversity among the particles, i.e., it introduces sample impoverishment. The sample impoverishment is severe in the case of small process noise, especially, for the case of very small process noise; all particles will collapse to a single point within a few iterations [2]. Schemes, such as Markovian chain Monte Carlo (MCMC) process, exist to improve particles diversity [10].

In general, it is hard to design the proper proposal distribution. Recently, cubature Klamen filter (CKF) uses a third-degree cubature rule to compute integrals numerically and solve high-dimensional nonlinear filtering problems with minimal computational effort; moreover, the square root cubature Kalman filter (SRCKF) was developed to improve the numerical stability [11], [12]. So, the combination of the CKF and PF forms an attractive framework, so cubature particle filter was proposed for continuous system [13]. In the study, we develop cubature particle filter with MCMC (CPF-MC) for discrete nonlinear system that captures the virtues of the SRCKF and PF and performs better than conventional PFs in terms of robustness and accuracy.

II. DEVELOPMENT OF CUBATURE PARTICLE FILTER WITH MCMC

Considering the following non-linear system:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is state, $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is the measurement; $\{\mathbf{w}_{k-1}\}$ and $\{\mathbf{v}_k\}$ are process and measurement Gaussian noise sequences with zero means and covariance \mathbf{Q}_{k-1} and \mathbf{R}_k , respectively, and they are mutually uncorrelated.

A. Cubature Particle Filter with MCMC

Initialization : $k = 0$

Draw the states (particles) $\mathbf{x}_0^{(i)}$ from $p(\mathbf{x}_0)$, calculate $\bar{\mathbf{x}}_0^{(i)} = E[\mathbf{x}_0^{(i)}]$ and $\mathbf{S}_0^{(i)} = \text{Chol}(E[(\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0^{(i)})(\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0^{(i)})^T])$, here, $i = 1, \dots, N$, $\text{Cho}()$ is cholesky decomposition.

Manuscript received July 9, 2014; revised September 18, 2014. This work was supported in part by the Key Foundation of Xi'an Technological University under Grant 605-01001219.

J. Mu is with the Xi'an Technological University, Xi'an, 710032 China (e-mail: mujing1977@163.com).

Y. Cai is with the Xi'an Jiaotong University, Xi'an, 710046 China (e-mail: ylicai@mail.xjtu.edu.cn)

For $k=1,2,\dots$

The particles and the square root of their corresponding covariances at $k-1$ time are $\mathbf{x}_{k-1}^{(1)}, \mathbf{x}_{k-1}^{(2)}, \dots, \mathbf{x}_{k-1}^{(i)}, \dots, \mathbf{x}_{k-1}^{(N)}$ and $\mathbf{S}_{k-1}^{(1)}, \mathbf{S}_{k-1}^{(2)}, \dots, \mathbf{S}_{k-1}^{(i)}, \dots, \mathbf{S}_{k-1}^{(N)}$, respectively.

Importance sampling step

Update the particles ($i=1, \dots, N$) using the SRCKF algorithm

Calculate cubature points here, $\xi_j = \sqrt{m/2}[1]_j$ and $\omega_j = 1/m$ ($j=1, 2, \dots, m=2n_x$) are the basic cubature points and their corresponding weights, the way of $[1]_j$ generated is introduced in [11].

$$\mathbf{X}_{k-1,j}^{(i)} = \mathbf{S}_{k-1}^{(i)} \xi_j + \mathbf{x}_{k-1}^{(i)} \quad (3)$$

Propagate particles into future

$$\mathbf{X}_{k,j}^{(i)*} = \mathbf{f}(\mathbf{X}_{k-1,j}^{(i)}) \quad (4)$$

$$\bar{\mathbf{x}}_k^{(i)} = \sum_{j=1}^m \omega_j \mathbf{X}_{k,j}^{(i)*} \quad (5)$$

$$\bar{\mathbf{S}}_k^{(i)} = \text{Tri}([\chi_k^{*(i)} \mathbf{S}_{Q,k-1}]) \quad (6)$$

Here, $\mathbf{S}_{Q,k-1}$ is a square root of \mathbf{Q}_{k-1} , $\text{Tri}()$ is denoted as a general triangularization algorithm, the weighted, centered matrix $\chi_k^{*(i)}$ is defined as:

$$\chi_k^{*(i)} = 1/\sqrt{m} [\mathbf{X}_{k,1}^{(i)*} - \bar{\mathbf{x}}_k^{(i)} \quad \mathbf{X}_{k,2}^{(i)*} - \bar{\mathbf{x}}_k^{(i)} \quad \dots, \mathbf{X}_{k,m}^{(i)*} - \bar{\mathbf{x}}_k^{(i)}] \quad (7)$$

Incorporate new observation

$$\mathbf{X}_{k,j}^{(i)} = \bar{\mathbf{S}}_k^{(i)} \xi_j + \bar{\mathbf{x}}_k^{(i)} \quad (8)$$

$$\mathbf{Z}_{k,j}^{(i)} = \mathbf{h}(\mathbf{X}_{k,j}^{(i)}) \quad (9)$$

$$\bar{\mathbf{z}}_k^{(i)} = \sum_{j=1}^m \omega_j \mathbf{Z}_{k,j}^{(i)} \quad (10)$$

$$\mathbf{S}_{zz,k}^{(i)} = \text{Tri}([\mathbf{I}_k^{*(i)} \mathbf{S}_{R,k}]) \quad (11)$$

$$\mathbf{P}_{xz,k}^{(i)} = \chi_k^{(i)} \mathbf{I}_k^{*(i)T} \quad (12)$$

$$\mathbf{W}_k = (\mathbf{P}_{xz,k}^{(i)} / \mathbf{S}_{zz,k}^{(i)}) / \mathbf{S}_{zz,k}^{(i)} \quad (13)$$

$$\hat{\mathbf{x}}_k^{(i)} = \bar{\mathbf{x}}_k^{(i)} + \mathbf{W}_k (\mathbf{z}_k - \bar{\mathbf{z}}_k^{(i)}) \quad (14)$$

$$\mathbf{S}_k^{(i)} = \text{Tri}([\chi_k^{(i)} - \mathbf{W}_k \chi_k^{*(i)} \quad \mathbf{W}_k \mathbf{S}_{R,k}]) \quad (15)$$

Here, $\mathbf{S}_{R,k}$ is a square root of \mathbf{R}_k , the matrices $\chi_k^{(i)}$, $\mathbf{I}_k^{*(i)}$ are defined as:

$$\chi_k^{(i)} = 1/\sqrt{m} [\mathbf{X}_{k,1}^{(i)} - \bar{\mathbf{x}}_k^{(i)} \quad \mathbf{X}_{k,2}^{(i)} - \bar{\mathbf{x}}_k^{(i)} \quad \dots, \mathbf{X}_{k,m}^{(i)} - \bar{\mathbf{x}}_k^{(i)}] \quad (16)$$

$$\mathbf{I}_k^{*(i)} = 1/\sqrt{m} [\mathbf{Z}_{k,1}^{(i)} - \bar{\mathbf{z}}_k^{(i)} \quad \mathbf{Z}_{k,2}^{(i)} - \bar{\mathbf{z}}_k^{(i)} \quad \dots, \mathbf{Z}_{k,m}^{(i)} - \bar{\mathbf{z}}_k^{(i)}] \quad (17)$$

Sample $\tilde{\mathbf{x}}_k^{(i)} \sim q(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_k^{(i)}, \mathbf{S}_k^{(i)} \mathbf{S}_k^{(i)T})$

Realize: $\tilde{\mathbf{x}}_k^{(i)} = \hat{\mathbf{x}}_k^{(i)} + \mathbf{S}_k^{(i)} \times \text{randn}(n_x, 1)$ Set

$$\tilde{\mathbf{x}}_{0:k}^{(i)} \triangleq (\mathbf{x}_{0:k-1}^{(i)}, \tilde{\mathbf{x}}_k^{(i)}) \text{ and } \tilde{\mathbf{S}}_{0:k}^{(i)} \triangleq (\mathbf{S}_{0:k-1}^{(i)}, \mathbf{S}_k^{(i)})$$

Evaluate the importance weights for each particle

$$w_k^{(i)} \propto \frac{p(\mathbf{z}_k | \tilde{\mathbf{x}}_k^{(i)}) p(\tilde{\mathbf{x}}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\tilde{\mathbf{x}}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k})} \quad (18)$$

Obtain normalized weights $\tilde{w}_k^{(i)}$

$$\tilde{w}_k^{(i)} = w_k^{(i)} / \sum_{i=1}^N w_k^{(i)} \quad (19)$$

Resampling

Multiply/Suppress particles $(\tilde{\mathbf{x}}_{0:k}^{(i)}, \tilde{\mathbf{S}}_{0:k}^{(i)})$ or $(\tilde{\mathbf{x}}_k^{(i)}, \mathbf{S}_k^{(i)})$ with high/low importance weights $\tilde{w}_k^{(i)}$ respectively, to obtain N random particles $(\tilde{\mathbf{x}}_{0:k}^{(i)}, \tilde{\mathbf{S}}_{0:k}^{(i)})$ or $(\tilde{\mathbf{x}}_k^{(i)}, \tilde{\mathbf{S}}_k^{(i)})$ with weight $1/N$.

SRCKF based MCMC process (this process is elaborately derived in Section 2.2)

Apply a Markov transition kernel with invariant distribution given by $p(\mathbf{x}_{0:k}^{(i)} | \mathbf{z}_{1:k})$ to obtain $(\mathbf{x}_{0:k}^{(i)}, \mathbf{S}_{0:k}^{(i)})$ or $(\mathbf{x}_k^{(i)}, \mathbf{S}_k^{(i)})$.

Output

Obtain the particles and corresponding covariance $(\mathbf{x}_{0:k}^{(i)}, \mathbf{S}_{0:k}^{(i)})$ or $(\mathbf{x}_k^{(i)}, \mathbf{S}_k^{(i)})$, and the estimated state $\hat{\mathbf{x}}_k$ and corresponding covariance \mathbf{P}_k at k time:

$$\hat{\mathbf{x}}_k = 1/N \sum_{i=1}^N \mathbf{x}_k^{(i)} \quad (20)$$

$$\mathbf{P}_k = 1/N \sum_{i=1}^N \mathbf{x}_k^{(i)} \mathbf{x}_k^{(i)T} - \hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T \quad (21)$$

Denoting the particles as $\{\tilde{\mathbf{x}}_k^{(j)}, j=1, \dots, N\}$ after resampling, assuming there exists a Markov transition kernel $K(\mathbf{x}_k^{(i)} | \tilde{\mathbf{x}}_k^{(i)})$ which meets the following constant condition [3]:

$$\int_{\tilde{\mathbf{x}}_k^{(j)}} K(\mathbf{x}_k^{(i)} | \tilde{\mathbf{x}}_k^{(j)}) p(\tilde{\mathbf{x}}_k^{(i)} | \tilde{\mathbf{x}}_{0:k-1}^{(i)}, \mathbf{z}_{1:k}) d\tilde{\mathbf{x}}_k^{(j)} = p(\mathbf{x}_k^{(i)} | \tilde{\mathbf{x}}_{0:k-1}^{(i)}, \mathbf{z}_{1:k}) \quad (22)$$

Applying $K(\mathbf{x}_k^{(i)} | \tilde{\mathbf{x}}_k^{(i)})$ with invariant distribution given by $p(\mathbf{x}_{0:k}^{(i)} | \mathbf{z}_{1:k})$ to obtain $(\mathbf{x}_{0:k}^{(i)}, \mathbf{P}_{0:k}^{(i)})$ and making the particles diversity, the particles $(\mathbf{x}_{0:k}^{(i)}, \mathbf{P}_{0:k}^{(i)})$ can be still approximated by $p(\mathbf{x}_{0:k}^{(i)} | \mathbf{z}_{1:k})$. This is the theory of implementing MCMC process. There are usually two MCMC methods used in practice, such as, the Metropolis-Hastings (M-H) algorithm and Gibbs sampler. The commonly used method is M-H method.

B. SRCKF based MCMC

We obtain $(\tilde{\mathbf{x}}_{0:k}^{(i)}, \tilde{\mathbf{S}}_{0:k}^{(i)})$ after resampling. In fact we only need: $(\tilde{\mathbf{x}}_{k-1}^{(i)}, \tilde{\mathbf{S}}_{k-1}^{(i)})$.

Sample v from a uniform distribution: $v \sim U[0, 1]$

Update the particles with the SRCKF algorithm

Calculate the cubature points and propagate the particle through state equation

$$\tilde{\mathbf{X}}_{k-1,j}^{(i)} = \tilde{\mathbf{S}}_{k-1}^{(i)} \xi_j + \tilde{\mathbf{x}}_{k-1}^{(i)} \quad (23)$$

$$\tilde{\mathbf{X}}_{k,j}^{(i)*} = \mathbf{f}(\tilde{\mathbf{X}}_{k-1,j}^{(i)}) \quad (24)$$

Time update

$$\bar{\tilde{\mathbf{x}}}_k^{(i)} = \sum_{j=1}^m \omega_j \tilde{\mathbf{X}}_{k,j}^{(i)*} \quad (25)$$

$$\bar{\tilde{\mathbf{S}}}_k^{(i)} = \text{Tri}([\tilde{\chi}_k^{*(i)} \mathbf{S}_{Q,k-1}]) \quad (26)$$

where the matrix $\tilde{\mathbf{x}}_k^{*(i)}$ is defined as:

$$\tilde{\mathbf{x}}_k^{*(i)} = 1/\sqrt{m} [\tilde{\mathbf{x}}_{k,1}^{(i)*} - \bar{\mathbf{x}}_k^{(i)} \quad \tilde{\mathbf{x}}_{k,2}^{(i)*} - \bar{\mathbf{x}}_k^{(i)}, \dots, \tilde{\mathbf{x}}_{k,m}^{(i)*} - \bar{\mathbf{x}}_k^{(i)}] \quad (27)$$

Measurement update

$$\tilde{\mathbf{x}}_{k,j}^{(i)} = \tilde{\mathbf{S}}_k^{(i)} \xi_j + \bar{\mathbf{x}}_k^{(i)} \quad (28)$$

$$\tilde{\mathbf{z}}_{k,j}^{(i)} = \mathbf{h}(\tilde{\mathbf{x}}_{k,j}^{(i)}) \quad (29)$$

$$\tilde{\mathbf{z}}_k^{(i)} = \sum_{j=1}^m \omega_j \tilde{\mathbf{z}}_{k,j}^{(i)} \quad (30)$$

$$\tilde{\mathbf{S}}_{zz,k}^{(i)} = \text{Tri}(\tilde{\mathbf{Y}}_k^{(i)} \mathbf{S}_{R,k}) \quad (31)$$

$$\tilde{\mathbf{P}}_{xz,k}^{(i)} = \tilde{\mathbf{x}}_k^{(i)} \tilde{\mathbf{Y}}_k^{(i)T} \quad (32)$$

$$\mathbf{W}_k = (\tilde{\mathbf{P}}_{xz,k}^{(i)} / \tilde{\mathbf{S}}_{zz,k}^{(i)T}) / \tilde{\mathbf{S}}_{zz,k}^{(i)} \quad (33)$$

$$\tilde{\mathbf{x}}_k^{*(i)} = \bar{\mathbf{x}}_k^{(i)} + \mathbf{W}_k (\mathbf{z}_k - \tilde{\mathbf{z}}_k^{(i)}) \quad (34)$$

$$\tilde{\mathbf{S}}_k^{*(i)} = \text{Tri}(\tilde{\mathbf{x}}_k^{(i)} - \mathbf{W}_k \tilde{\mathbf{Y}}_k^{(i)} \quad \mathbf{W}_k \mathbf{S}_{R,k}) \quad (35)$$

where the matrices $\tilde{\mathbf{x}}_k^{(i)}$, $\tilde{\mathbf{Y}}_k^{(i)}$ are defined as:

$$\tilde{\mathbf{x}}_k^{(i)} = 1/\sqrt{m} [\tilde{\mathbf{x}}_{k,1}^{(i)} - \bar{\mathbf{x}}_k^{(i)} \quad \tilde{\mathbf{x}}_{k,2}^{(i)} - \bar{\mathbf{x}}_k^{(i)}, \dots, \tilde{\mathbf{x}}_{k,m}^{(i)} - \bar{\mathbf{x}}_k^{(i)}] \quad (36)$$

$$\tilde{\mathbf{Y}}_k^{(i)} = 1/\sqrt{m} [\tilde{\mathbf{z}}_{k,1}^{(i)} - \bar{\mathbf{z}}_k^{(i)} \quad \tilde{\mathbf{z}}_{k,2}^{(i)} - \bar{\mathbf{z}}_k^{(i)}, \dots, \tilde{\mathbf{z}}_{k,m}^{(i)} - \bar{\mathbf{z}}_k^{(i)}] \quad (37)$$

Sample the candidate

$$\mathbf{x}_k^{*(i)} \sim q(\mathbf{x}_k | \tilde{\mathbf{x}}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) = \mathcal{N}(\tilde{\mathbf{x}}_k^{*(i)}, \tilde{\mathbf{S}}_k^{*(i)} \tilde{\mathbf{S}}_k^{*(i)T})$$

If $v \leq \min\{1, \frac{p(\mathbf{z}_k | \mathbf{x}_k^{*(i)}) p(\mathbf{x}_k^{*(i)} | \tilde{\mathbf{x}}_{0:k-1}^{(i)}) q(\tilde{\mathbf{x}}_k^{(i)} | \tilde{\mathbf{x}}_{0:k-1}^{(i)}, \mathbf{z}_{1:k})}{p(\mathbf{z}_k | \tilde{\mathbf{x}}_k^{(i)}) p(\tilde{\mathbf{x}}_k^{(i)} | \tilde{\mathbf{x}}_{0:k-1}^{(i)}) q(\mathbf{x}_k^{*(i)} | \tilde{\mathbf{x}}_{0:k-1}^{(i)}, \mathbf{z}_{1:k})}\}$ then

accept move

$$\mathbf{x}_{0:k}^{(i)} = (\tilde{\mathbf{x}}_{0:k}^{(i)}, \mathbf{x}_k^{*(i)}), \mathbf{S}_{0:k}^{(i)} = (\tilde{\mathbf{S}}_{0:k}^{(i)}, \mathbf{S}_k^{*(i)})$$

Else reject move

$$\mathbf{x}_{0:k}^{(i)} = \tilde{\mathbf{x}}_{0:k}^{(i)}, \tilde{\mathbf{S}}_{0:k}^{(i)} = \tilde{\mathbf{S}}_{0:k}^{(i)}$$

From the above descriptions, the paper has two contributions. The first contribution is that the CPF-MC uses the SRCKF as proposal distribution that integrates the latest measurement into the particle filter and approximates the optimal posterior probability distribution more accurately. The second contribution is that the SRCKF based MCMC process after resampling is used to make the particles diversity and suppress the impoverish phenomenon. Because the SRCKF is more accurate and numerical stability; furthermore, has less computation cost, so CPF-MC should perform better and its computation cost is decreased.

III. RE-ENTRY BALLISTIC TARGET TRACKING

To demonstrate the superiority of the CPF-MC, we apply it to a practical re-entry ballistic target tracking with a six-dimensional state space. All the simulations were done in MATLAB on a ThinkPad PC with an Intel (R) CORE i5 M480 processor with the 2.67GHz clock speed and 3GB physical memory.

A. Model of Target Motion and of Radar Measurements

Assume that the Earth is spherical and non-rotating; the radar is situated at the surface of the Earth, the relative

location of the re-entry ballistic target and the radar is depicted in [14].

There are two significant forces, gravity and aerodynamic drag, acting on the target in the re-entry phase [15]. With this formulation, the re-entry ballistic target motion is described by the following discrete-time nonlinear dynamic state equation in the East-North-Up coordinates system (ENU-CS):

$$\mathbf{x}_k = \mathbf{\Phi} \mathbf{x}_{k-1} + \mathbf{G} \boldsymbol{\psi}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \quad (38)$$

where the state vector is defined as:

$$\mathbf{x}_k = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k \quad z_k \quad \dot{z}_k]^T$$

$$\mathbf{\Phi} = \text{diag}([\phi \quad \phi \quad \phi])$$

$$\phi = [1 \quad T; 0 \quad 1]$$

$$\mathbf{G} = \text{diag}([\tau \quad \tau \quad \tau])$$

$$\tau = [T^2/2; T]$$

And $\boldsymbol{\psi}(\mathbf{x}_k) = [f_1(\mathbf{x}_{k-1}) \quad f_2(\mathbf{x}_{k-1}) \quad f_3(\mathbf{x}_{k-1})]^T$ is formulated as:

$$f_1(\mathbf{x}_{k-1}) = -\rho(h_{k-1}) V_{k-1} \dot{x}_{k-1} / 2\beta - \mu x_{k-1} / r_{k-1}^3 \quad (39)$$

$$f_2(\mathbf{x}_{k-1}) = -\rho(h_{k-1}) V_{k-1} \dot{y}_{k-1} / 2\beta - \mu y_{k-1} / r_{k-1}^3 \quad (40)$$

$$f_3(\mathbf{x}_{k-1}) = -\rho(h_{k-1}) V_{k-1} \dot{z}_{k-1} / 2\beta - \mu (z_{k-1} + R_e) / r_{k-1}^3 \quad (41)$$

Here:

$$r_{k-1} = \sqrt{x_{k-1}^2 + y_{k-1}^2 + (z_{k-1} + R_e)^2}$$

$$V_{k-1} = \sqrt{\dot{x}_{k-1}^2 + \dot{y}_{k-1}^2 + \dot{z}_{k-1}^2}$$

$$h_{k-1} = r_{k-1} - R_e$$

T (in s) is the time interval between radar measurements, μ is Earth's gravitational constant ($\mu = 3.986005 \times 10^{14} \text{m}^3/\text{s}^2$) and R_e is Earth's radius ($R_e = 6378.137 \text{km}$), respectively.

$\rho(h)$ is the air density (kg/m^3). Below 90km at height, the air density $\rho(h)$ is approximately modeled as an exponentially decaying function of height, i.e. $\rho(h) = c_1 e^{-c_2 h}$, c_1 and c_2 are dimensionless constants, i.e. $c_1 = 1.754$ and $c_2 = 1.49 \times 10^{-4}$ [16]. β is ballistic coefficient (kg/m^2).

Process noise \mathbf{w}_{k-1} is assumed to be a zero-mean white Gaussian process with non-singular covariance matrix $\mathbf{Q}_k = \text{diag}([q\theta \quad q\theta \quad q\theta])$, $\theta = [T^3/3 \quad T^2/2; T^2/2 \quad T]$, parameter q (in m^2/s^3) controls the amount of process noise in target dynamics.

The measurements, collected by the radar for target tracking, are the range R_k , elevation E_k and azimuth A_k . According to the geometry, the measurement equation is:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (42)$$

where $\mathbf{z}_k = [R_k \quad E_k \quad A_k]^T$, and

$$R_k = \sqrt{x_k^2 + y_k^2 + z_k^2} + v_R \quad (43)$$

$$E_k = \arctan(z_k / \sqrt{x_k^2 + y_k^2}) + v_E \quad (44)$$

$$A_k = \arctan(y_k / x_k) + v_A \quad (45)$$

v_k is modeled as zero-mean white Gaussian measurement noise with covariance matrix $N_k = \text{diag}([\sigma_R^2 \sigma_E^2 \sigma_A^2])$, σ_R , σ_E and σ_A are the error standard deviations of range, elevation and azimuth. It is independent of the process noise w_k and initial state x_0 .

B. Simulations and analysis

In the simulation, the parameters used are selected as $T=0.1s$, $q=1m^2/s^3$, the initial position and module of the velocity: $x_0=232km$, $y_0=232km$, $z_0=90km$, and $V_0=3000m/s$; the initial elevation and azimuth angle: $E_0=210^\circ$, and $A_0=60^\circ$. The ballistic coefficient: $\beta=4000kg/m^2$. The ballistic trajectory is generated using fourth-order Runge-Kutta method. Then the true initial state can be obtained as $x_0=[232km-1299m/s \ 232km-2250m/s \ 90km-1500m/s]$, and the corresponding covariance is selected as $P_0=\text{diag}[500^2 \ 500^2 \ 200^2 \ 500^2 \ 200^2]$.

The initial state estimate \hat{x}_0 is chosen randomly from $\hat{x}_0 \sim \mathcal{N}(x_0, P_0)$ in each run. The error standard deviations of the measurements are selected as $\sigma_R=100m$, and $\sigma_E=\sigma_A=1mrad$. All of the particle filters used 400 particles.

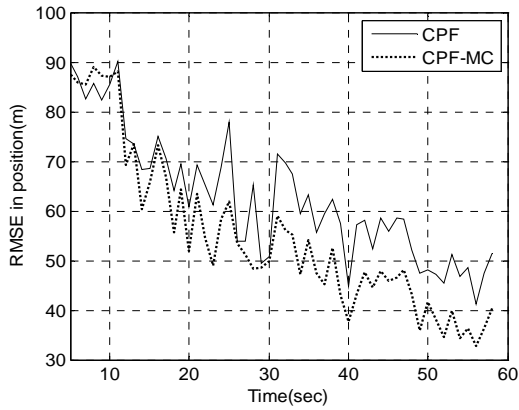


Fig. 1. RMSEs in position for CPF and CPF-MC.

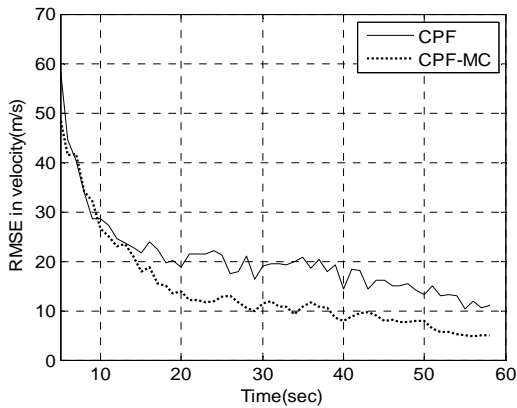


Fig. 2. RMSEs in velocity for CPF and CPF-MC.

To compare the performance of the various PFs, we use the performance metrics of root-mean square error (RMSE) and average accumulated mean-square root error (AMSRE) in the position and velocity which are separately defined in [14]. All following performance curves and figures were obtained by averaging over 100 independent Monte Carlo runs. All filters were initialized with the same condition in each run.

Firstly, we compare the performance of the CPF (no with

MCMC) and CPF-MC on re-entry ballistic target tracking. Fig.1 and Fig. 2 show the RMSEs for the CPF and CPF-MC in position and velocity. The means and variances of the AMSREs of two algorithms in position and velocity are listed in Table I. Table II lists the runtime of the two algorithms.

TABLE I: MEAN AND COVARIANCE OF AMSREs FOR VARIOUS FILTERS IN POSITION AND VELOCITY

Various filters	Position/m		Velocity/m·s ⁻¹	
	mean	variance	mean	variance
CPF	71.2043	8.7973	85.4288	51.9024
CPF-MC	65.0396	9.3895	84.5774	64.8122

TABLE II: MEAN RUNTIME FOR VARIOUS FILTERS

Various filters	Mean runtime/s
CPF	126.90
CPF-MC	268.17

From Fig. 1, Fig. 2 and Table I, we can see the RMSEs of CPF-MC in position and velocity are less than those of CPF because the CPF-MC includes the MCMC process after resampling. Again we see that the performance of the CPF-MC is superior to that of the CPF. Meantime, from Table II, we also see that CPF-MC has large amount of computational load because the runtime of CPF-MC is more than twice that of CPF.

Now the CPF-MC algorithm is compared to the GPF with MCMC (GPF-MC), EPF with MCMC based on EKF (EPF-MC) and UPF with MCMC based on UKF (UPF-MC) for re-entry ballistic target tracking. Fig.3 and Fig.4 show the RMSEs for the four filters in position and velocity. Table III lists the means and variances of AMSREs for the four filters in position and velocity. The runtime of the four filters is listed in Table IV. The GPF-MC algorithm is not listed in Figures and Tables because it is divergent to the re-entry ballistic target tracking.

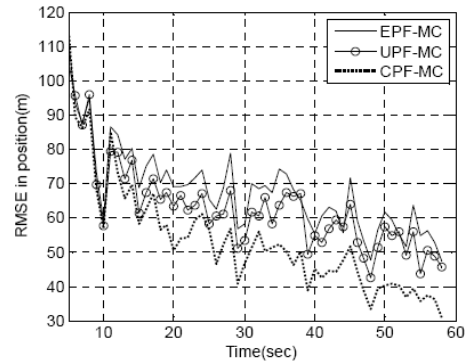


Fig. 3. RMSEs in position for various filters.

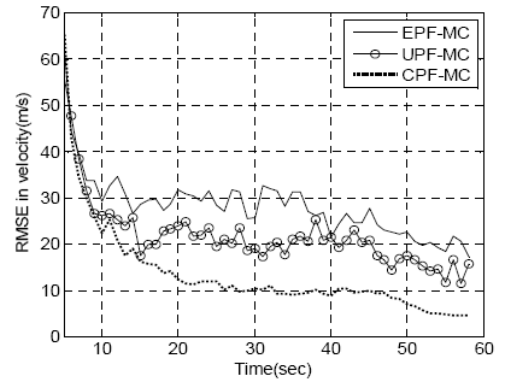


Fig. 4. RMSEs in velocity for various filters.

From Table IV, the runtime of UPF-MC is about three times that of EPF-MC and CPF-MC, and the runtime of CPF-MC is slightly larger than that of EPF-MC. Hence the CPF-MC has better performance on re-entry ballistic target tracking.

TABLE III: MEAN AND VARIANCE OF AMRSES IN POSITION AND VELOCITY FOR VARIOUS ALGORITHMS

Various algorithms	Position/m		Velocity/ $\text{m}\cdot\text{s}^{-1}$	
	mean	variance	mean	variance
EPF-MC	77.4252	8.9509	158.5514	70.3600
UPF-MC	73.0660	15.152	155.9940	61.5161
CPF-MC	66.1665	10.3946	151.2112	57.6589

TABLE IV: MEAN RUNTIME FOR VARIOUS ALGORITHMS

Various algorithms	Mean runtime/s
EPF-MC	268.79
UPF-MC	844.38
CPF-MC	283.33

From Fig. 3, Fig. 4 and Table III, the EPF-MC use EKF algorithm as the proposal distribution, the EKE algorithm uses linearization and Gaussian assumption to introduce the large error, the performance of EPF-MC is poor on re-entry ballistic target tracking. The RMSEs and means and variances of UPF-MC in position and velocity decrease compared to those of EPF-MC. And the RMSEs and means and variances of CPF-MC in position and velocity significantly reduce compared to EPF-MC and UPF-MC algorithms.

From Table IV, the runtime of UPF-MC is about three times that of EPF-MC and CPF-MC, and the runtime of CPF-MC is slightly larger than that of EPF-MC. Hence the CPF-MC has better performance on re-entry ballistic target tracking.

IV. CONCLUSION

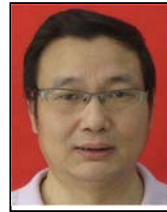
In this study, we develop the CPF-MC algorithm, which uses a SRCKF as proposal distribution, making it better suitable for proposal distribution generation, and the SRCKF based MCMC process after resampling is used to effectively alleviate the degeneracy and impoverishment problems in the particle filter. Applications of reentry ballistic target tracking demonstrate that the CPF-MC is superior to the GPF-MC, EPF-MC and UPF-MC, and the runtime of CPF-MC is less than that of EPF-MC, and is about third of that of UPF-MC.

REFERENCES

- [1] B. Ristic, S. Arulampalam, and N. J. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Application*, Boston: Artech House, 2004.
- [2] A. Doucet and N. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2011.
- [3] P. Fearnhead, "Sequential Monte Carlo methods in filter theory," Ph.D. dissertation, University of Oxford, Oxford, MA, 1998.
- [4] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, March 2000.
- [5] J. F. G. D. Freitas, M. Niranjanm, A. H. Gee, and A. Doucet, "Sequential monte carlo methods to train neural network models," *Neural Computation*, vol. 12, no. 4, pp. 955-993, April 2000.
- [6] R. V. D. Merwe, A. Doucet, N. D. Freitas, and E. Wan, "The unscented particle filter," *Advances in Neural Information Processing Systems*, vol. 13, pp. 584-590, 2001.
- [7] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computation visual*, vol. 77, pp. 125-141, 2008.
- [8] A. S. Stordal, H. A. Karlsen, G. Naevdal, and B. Valles, "Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter," *Computational Geosciences*, vol. 15, no. 2, pp. 293-305, Feb, 2011.
- [9] D. Mikami, K. Otsuka, and J. Yamato, "Memory-based particle filter for face pose tracking robust under complex dynamics," presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami, FL, pp. 999-1006, 2009.
- [10] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb, 2002.
- [11] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. on Automatic Control*, vol. 54, no. 6, pp. 1254-1269, June 2009.
- [12] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Trans. on Signal Processing*, vol. 58, no. 10, pp. 4977-4993, Oct 2010.
- [13] M. Havlicek, J. Jan, M. Brazdil, and V. D. Callhoun, "Nonlinear estimation of BOLD signal based on cubature particle filter," presented at the 20th Biennial International EURASIP Conference in Analysis of Biomedical Signals and Images, Brno, CZ, pp. 328-332, 2010.
- [14] J. Mu and Y. Cai, "Likelihood-based iteration square-root cubature Kalman filter with applicaitons to state estimation of re-entry ballistic target," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 7, pp. 949-958, Oct 2013.
- [15] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking part II: motion models of ballistic and space targets," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 96-119, Jan 2010.
- [16] A. Farina, B. Ristic, and D. Benvenuti, "Tracking a ballistic target: comparison of several nonlinear filters," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 854-867, Feb 2002.



J. Mu received her BSc and MSc degrees in computer science and engineering in 1999 and 2002 for Xi'an Technological university, respectively, and she received her Ph.D degree in control engineering and theory from Xi'an Jiaotong University in 2012. Her research interests include non-linear state estimation techniques for reentry ballistic target.



Yuanli Cai received his B.S, M.S and Ph.D degrees from Northwestern Polytechnical University in 1984, 1987 and 1990, respectively. He is currently a professor in the School of Electronic and Information Engineering and the director of the control engineering institute of Xi'an Jiaotong University. His research interests include flight control, guidance and navigation, complex system modeling and simulation, intelligent information

processing, etc.