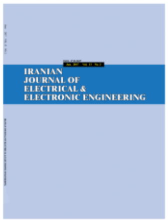
IUST
Iran University of
Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper



[DOI: 10.22068/IJEEE.16.4.449]

A New Modified Particle Filter With Application in Target Tracking

R. Havangi*(C.A.)

Abstract: The particle filter (PF) is a novel technique that has sufficiently good estimation results for the nonlinear/non-Gaussian systems. However, PF is inconsistent that caused mainly by loss of particle diversity in resampling step and unknown a priori knowledge of the noise statistics. This paper introduces a new modified particle filter called adaptive unscented particle filter (AUPF) to overcome these problems. The proposed method uses an adaptive unscented Kalman filter (AUKF) filter to generate the proposal distribution, in which the covariance of the measurement and process of the state are online adjusted by predicted residual as an adaptive factor based on a covariance matching technique. In addition, it uses the genetic operators based strategy to further improve the particle diversity. The results show the effectiveness of the proposed approach.

Keywords: Particle Filter, Genetic Algorithm, Unscented Kalman Filter, Target Tracking.

1 Introduction

THE state estimation is to find the actual values of states of a dynamical system using a sequence of noisy measurements [1, 2]. It plays an important role in many applications, such as target tracking [3], robot navigation [4, 5], etc. The PF is an effective estimator for the nonlinear/non-Gaussian systems [1]. It is a recursive Monte Carlo-based method that constructs probability density function (PDF) using a set of random particles with associated weights [6]. In PF, the particles are evolved over time via a combination of importance sampling and resampling step.

To improve the performance of PF, choosing the proposal distribution and the resampling scheme are important [1]. In standard PF, the state transition is often chosen as the proposal distribution [1, 2, 6]. As this proposal distribution does not include information about the new measurements, most particles get negligible weights and it leads to the degeneracy of particle filter [8, 9]. In addition, the resampling step is performed to reduce the effects of particle degeneracy [1]. In the resampling step, the particles with low-weights are discarded while those with high-weights are copied many times [11]. This will lead to a great loss of diversity in particles and thus bring another problem that is particle impoverishment. In an extreme case, all particles may collapse to a single point within a few iterations [11-13]. Thus, methods have to be introduced to add variability to the resampled particle set to reduce the potential of particle impoverishment [14]. To solve these problems, other forms of proposal distribution that take into account the new observations are proposed [15-17]. For example, the auxiliary particle filter [1], the regularized particle filter [18], the bootstrap particle filter [9-11]. In [19, 20], the extended Kalman filter (EKF) Gaussian approximation is used as the proposal distribution for PF. In [21], a support vector regression particle filter (SVRPF) is proposed to overcome particle degeneracy and impoverishment problems from nonlinear and non-Gaussian environments, especially in regard to narrow observation noise. In addition, the consistency of the SVRPF and Bayes' filtering is demonstrated. In [22], insignificant particles are removed using a binary mask in each stage. The feedback PF is designed based on an ensemble of a controlled stochastic system in [23]. In this method, the evolution of each particle is controlled by its own state and feedback circuit. As the resampling technique in solving the degeneracy problem, some

Iranian Journal of Electrical and Electronic Engineering, 2020.
Paper first received 17 September 2019, revised 07 November 2019, and accepted 09 November 2019.

* The author is with the Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.
E-mail: havangi@birjand.ac.ir.
Corresponding Author: R. Havangi.

researchers' various resampling schemes have been proposed [24, 25]. A novel resampling scheme called branching particle method is discussed in [26]. In this method, four new branchings are introduced that resolved the common complaint of unstable particle number from the branching PF while reducing the computation complexity. Some researchers use optimization methods to improve particle filters [27].In this approach, before the resampling, by modified grey wolf optimizer, the particles in the PF are optimized.

A serious limitation of PF is that it requires a priori knowledge of the noise statistics. However, in real applications, a priori knowledge of the noise statistics is unknown. The problem here is that the performance of PF is closely connected to the quality of these priori noise statistics. Evidences have shown how a poor estimation of the input noise statistics may seriously degrade the filter performance, and even provoke the divergence of the filter. Incorrect priori knowledge of the noise statistics may seriously degrade the performance of PF [28-30]. It can even lead to practical divergence and inconsistency [31, 32]. A classical method for solving this problem is adaptive estimation of priori knowledge. In the field of Kalman filter, different adaptation methods are researched in recent years. The adaptation is based on Sage-Huse methods [33]. However, these methods have an instability problem when the noise covariance is not semi-positive definite. The residual-based adaptive estimation methods are developed to adapt the measurement noise covariance matrices [34, 35]. The main idea of the covariance matching adaptation method is to estimate the covariance of measurement and process noise at every sampling instant by keeping the residual covariance consistent with its theoretical value [36, 37]. Another adaptation method is based on interacting multiple models filtering, which is combined with the EKF or UKF method to deal with nonlinear model uncertainties [38, 39].

To solve the aforementioned problems and improve the performance of PF, a new adaptive PF based on adaptive unscented Kalman filter and Genetic algorithm is proposed, which is called AUPF. Compared with standard PF, the advantage of the proposed method is that it is well robust in terms of accuracy and consistency in different conditions, especially when the statistics of noises are unknown. This is why; first, AUPF uses AUKF to define the proposal density. Since AUKF includes information of the new observations, the proposed particles will be much closer to observations than this generated from the state transition and therefore reducing the particle degeneracy. In addition, at the same time, it adaptively estimates and adjusts measurement noise variance. Second, the genetic operators are incorporated in AUPF before the resampling step. It increases the diversity of particles, and thus reduces the potential of particle impoverishment and leads to a more complete and accurate description of the posterior PDF.

The rest of the paper is organized as follows. The particle filter and genetic algorithm are briefly reviewed in Section 2. The adaptive unscented particle filter is developed and briefly analyzed in Section 3. Simulation results that compare the performances of the existing algorithms are presented in Section 4. Finally, some conclusions are provided in Section 5.

2 Background

2.1 Particle Filter

A dynamic system represented by

$$\begin{aligned} x_t &= f(x_{t-1}) + w_{t-1} \\ y_t &= h(x_t) + v_t \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^n$ is the state vector, $y_t \in \mathbb{R}^m$ is the measurement vector, w_{t-1} and v_t are independent white-noise variables, $f(.)$ and $h(.)$ are known nonlinear functions with appropriate dimensions. The objective of filtering is to estimate the posterior density of the state given the past measurements $p(x_t|y_{1:t})$ [1, 2]. From the perspective of Bayesian filtering, the posterior $p(x_t|y_{1:t})$ can be estimated in two steps: 1) the prediction and 2) the update [1]. These two steps are formulated as (2) and (3), respectively

$$p(x_t \mid y_{1:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1} \tag{2}$$

$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t) p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \tag{3}$$

where $p(y_t|y_{1:t-1})$ is as:

$$p(y_t \mid y_{1:t-1}) = \int p(y_t \mid x_t) p(x_t \mid y_{1:t-1}) dx_t$$

The PF is an implementation of Bayesian filtering by sequential Monte Carlo (SMC) sampling. In PF, by using a set of samples and the corresponding weights at time step $\{(x_t^{(i)}, w_t^{(i)}),\ i = 1...N\}$, the posterior $p(x_t|y_{1:t})$ approximated as [1-2]:

$$p(x_t \mid y_{1:t}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta(x_t - x_t^{(i)}) \tag{4}$$

where $\delta(x_t - x_t^{(i)})$ is Dirac's delta function, and N is the number of particles. The particle filter consists of three steps: sampling, importance weighting, and resampling. In the sampling step, particles are sampled according to the proposal density function $q(x_t^{(i)} \mid x_{t-1}^{(i)}, y_t)$. The weight of particles is updated as follows [2]:

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} \frac{p(y_t \mid x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)})}{q(x_t^{(i)} \mid x_{t-1}^{(i)}, y_t)} \tag{5}$$

2.2 Genetic Algorithm

Genetic algorithm (GA) is a global search technique that mimics aspects of biological evolution. It has been successfully applied to solve different optimization problems [40]. The main idea of GA is based on the evolutionary process of biological organisms. The population of GA develops by selection, crossover, and mutation [41, 42]. Crossover is a genetic operation, which pairs two individuals and mates them. Mutation is another operation that randomly alters the selected individual. In proportion to the fitness values, the individuals are selected to undergo crossover and mutation and search for an optimal solution. In summary, the implementation procedure of GA is as follows:

1. Initialization

   Generate random chromosomes describing the solution, the number corresponds to the size of the population and bounded within the solution space.

2. Iteration

   a) Calculate the fitness of chromosomes;

   b) Select chromosome into an intermediate population according to their fitness;

   c) Perform crossover and mutation to get new individuals into the population.

3. Termination

   If some termination condition is met the iteration generation number met the threshold or the fitness is good enough, and so on. Otherwise, repeat the iterations

3 A New Modified Particle Filter

In this section, a new modified particle filter called AUPF is proposed. As the performance of PF depends on sampling strategy and resampling method, the optimization of PF is done with improving sampling and resampling steps. In AUPF, AUKF is used to define the proposal density of PF and the genetic operators based strategy is used to increase diversity particles.

3.1 Sampling Step

The choice of proposal distribution is one of the most critical issues in the design of particle filter [1, 2]. The most popular choice is the transition prior. However, it can fail when prior distribution is a much broader distribution than the likelihood [1]. To solve this problem, the proposal distribution is generated using AUKF in this paper. This strategy has the advantage of including information from the latest measurement in the proposal distribution. In this algorithm, the proposal distribution is as follows:

$$x^{(i)}_t \sim q(x^{(i)}_t \mid x^{(i)}_{t-1}, y_t) = N(x_t; x^{(i)}_{t|t}, P^{(i)}_{t|t}) \tag{6}$$

where $N(x_t; x^{(i)}_{t|t}, P^{(i)}_{t|t})$ is the Gaussian distribution (with mean $x^{(i)}_{t|t}$ and covariance $P^{(i)}_{t|t}$) obtained by using AUKF. For this purpose, a set of $2n$+1 points is generated by

$$\begin{aligned}
&\chi^{(0)(i)}_{t-1|t-1} = x^{(i)}_{t-1|t-1}\\
&\chi^{(j)(i)}_{t-1|t-1} = x^{(i)}_{t-1|t-1} + \left(\sqrt{(n_x+\lambda)\ P^{(i)}_{t-1|t-1}}\right)_j \qquad j=1,\ldots,n_x\\
&\chi^{(j)(i)}_{t-1|t-1} = x^{(i)}_{t-1|t-1} - \left(\sqrt{(n_x+\lambda)\ P^{(i)}_{t-1|t-1}}\right)_j \qquad j=n+1,\ldots,2n_x
\end{aligned} \tag{7}$$

where $\left(\sqrt{(n+\lambda)\ P^{(i)}_{t-1|t-1}}\right)_j$ is j-th $j-th$ column of the matrix square root of $(n+\lambda)\ P^{(i)}_{t-1|t-1}$, n_x is the dimension of state. The sigma propagated through the nonlinear system and measurement models as:

$$\begin{aligned}
&\bar{x}^{(j)(i)}_{t|t-1} = f(\chi^{(j)(i)}_{t-1|t-1})\\
&\Im^{(j)(i)}_t = h(x^{-(j)(i)}_{t|t-1})
\end{aligned} \tag{8}$$

where $\bar{x}^{(j)(i)}_{t|t-1}$ is its transformed sigma point. The prediction of mean $x^{(i)}_{t|t-1}$ and covariance $P^{(i)}_{t|t-1}$ of the state as well as the prediction of the measurement vector is performed as:

$$\begin{aligned}
&x^{(i)}_{t|t-1} = \sum_{j=0}^{2n} \omega^{(j)}_m \bar{x}^{(j)(i)}_{t|t-1}\\
&\bar{y}^{(i)}_t = \sum_{j=0}^{2n} w^{(j)}_m \Im^{(j)(i)}_t
\end{aligned} \tag{9}$$

$$P^{(i)}_{t|t-1} = \sum_{j=0}^{2n} \omega^{(j)}_c (\bar{x}^{(j)(i)}_{t|t-1} - x^{(i)}_{t|t-1})(\bar{x}^{(j)(i)}_{t|t-1} - x^{(i)}_{t|t-1})^T + Q_t \tag{10}$$

where the weights ω_m and ω_c are as:

$$\begin{aligned}
&\omega^{(0)}_m = \frac{\lambda}{(n+\lambda)} \qquad \omega^{(0)}_c = \frac{\lambda}{(n+\lambda)} + (1-\alpha^2+\beta)\\
&\omega^{(j)}_m = \omega^{(j)}_c = \frac{\lambda}{2(n+\lambda)} \qquad (j=1,\ldots,2n)
\end{aligned}$$

The parameter β is used to incorporate the knowledge of the posterior distribution. Finally, the updated state and covariance are calculated by

$$\begin{aligned}
&x^{(i)}_{t|t} = x^{(i)}_{t|t-1} + K^{(i)}_t (y_t - \bar{y}^{(i)}_t)\\
&P^{(i)}_{t|t} = P^{(i)}_{t|t-1} - K_t P^{(i)}_{\upsilon\upsilon} (K^{(i)}_t)^T
\end{aligned} \tag{11}$$

where

$$\begin{aligned}
&K^{(i)}_t = P^{(i)}_{\delta\upsilon} (P^{(i)}_{\upsilon\upsilon})^{-1}\\
&P^{(i)}_{yy} = \sum_{j=0}^{2n} w^{(j)}_c (\Im^{(j)(i)}_t - \bar{y}^{(i)}_t)(\Im^{(j)(i)}_t - \bar{y}^{(i)}_t)^T + R_t\\
&P^{(i)}_{xy} = \sum_{j=0}^{2n} \omega^{(j)}_c (\bar{x}^{(j)(i)}_{t|t-1} - x^{(i)}_{t|t-1})(\zeta^{(j)(i)}_t - \bar{y}^{(i)}_t)^T
\end{aligned} \tag{12}$$

and y_t is the measurement. From the above equations, it can be seen that the covariance matrices R_t and Q_t will affect the precision of the UPF directly. In target tracking applications, the priori knowledge of the process noise and measurement noise statics are unknown. This is why it is hard to determine process noise and measurement noise covariance.

The use of wrong priori statistics in PF may lead to large estimation errors or even to the divergence. If R_t and/or Q_t are too small at the beginning of the estimation process, the uncertainty tube around the true value will probably tighten and a biased solution might result; if R_t and/or Q_t are too large, filter divergence, in the statistical sense, could occur. So the accuracy of R_t and Q_t has become the key problem that affects the precision and stability of the filter and needs to be tuned online. In this paper, an online adaptive scheme of UPF is presented. The adaptation is in the sense of adaptively adjusting the process noise and measurement noise covariances using the covariance-matching technique. The basic idea behind the covariance-matching technique is to make the actual value of the covariance of the innovation sequence match its theoretical value [12]. When the actual residual is incompatible with the original hypothesis of Q_t and R_t, we should make the new estimates $\hat{Q}_t$ and $\hat{R}_t$ to replace the original assumptions. The residual sequence υ_t is represented as the difference between actual measurement vector y_t and the estimated measurement and is defined as [24, 25]:

$$\upsilon_t = y_t - \hat{y}_t$$

The actual covariance of υ_t can be approximated as:

$$C_t = \frac{1}{m}\sum_{k=0}^{m-1} \upsilon_t \upsilon_t^T \tag{13}$$

where m is the width of the moving window and is chosen empirically to give some statistical smoothing,v is the residual vector. If the window size is too small, the measurement estimation covariance can be noisy. In general, the window size is chosen empirically for statistical smoothing [24, 25]. It should be pointed out that too large a window width suffers from the severe computational inefficiency and too small a window width might lead to large variance. Thus, it is important to choose a reasonable window width. From the measurement equation of the standard UKF, the theoretical covariance of υ_t is computed by

$$E(v_t v_t^T) = \sum_{j=0}^{2n} w_c^{(j)} (\Im_t^{(j)(i)} - \bar{z}_t^{(i)})(\Im_t^{(j)(i)} - \bar{y}_t^{(i)})^T + R_t \tag{14}$$

By comparing (14) and (15), an estimate of the matrix R_t can be given by

$$\hat{R}_{t+1} = C_t - P_{yy} \tag{15}$$

where $\hat{R}_t$ is the estimated measurement noise covariance matrix. The matrix $\hat{Q}_t$ can be given by

$$\hat{Q}_{t+1} = K_{t+1} C_t K_{k+1}^T \tag{16}$$

where $\hat{Q}_t$ is the estimated process noise covariance matrix.

3.2 Importance Weighting

In the importance weighting, the weights are updated as follows:

$$w_t = \frac{p(x_t \mid y_{1:t})}{q(x_t \mid y_{1:t})} \tag{17}$$

3.3 Crossover and Mutation Operators

During the resampling step, the small-weight particles might end up with no children and the large-weight particles have a large number of children [11, 12]. Although the resampling step reduces the effects of the degeneracy problem, it decreases the diversity of particles [13]. In AUPF, to mitigate the problem of particle impoverishment, the crossover and mutation are applied to particles before the resampling step.

In the proposed method, the particles $\{\, x_t^{(i)} \quad i = 1,...,N \,\}$ are regarded as the current population where N is the number of particles, $[x_{t,1}^{(m)},...,x_{t,D}^{(m)}]$ is denote chromosome each candidate solution, which consists of D D-dimensional vector x_t, and the corresponding weights of particles are as the fitness functions. Let $C_t^k = \{x_{t,1}^{k(i)},...,x_{t,D}^{k(i)}\}^{i=1:N}$ denote the current population at iteration k, where $\{x_{t,1}^{k(i)},...,x_{t,D}^{k(i)}\}$ represents candidate solution to the optimization problem at iteration k for a given time step t. GA will generate a new population using crossover and mutation operator. For this purpose, the particles are divided into the small-weight particles C_L and large-weight particles C_H. In order to separate the small-weight particles from other large-weight ones, the weight of each particle is compared with a threshold ω_T.

After separating particles, the crossover operator is performed on small-weight particles. In fact, the small-weight particles are modified using the information of the large-weight particles through the crossover operator. There are many crossover operators. In this paper, arithmetic crossover is selected to modify the small-weight particles. Assume that $x_t^{(L)}$ and $x_t^{(H)}$ respectively represent the particles from C_L and C_H. The modified small-weight particle is denoted by $x_t'^{(L)}$ as follows:

$$x_t'^{(L)} = \eta x_t^{(L)} + (1-\eta) x_t^{(H)} \tag{18}$$

in which l = 1, …, N_L and h = 1, …, N_H. N_L and N_H respectively represent the number of particles contained in C_L and C_H. For each $x_t'^{(L)}$, $x_t^{(H)}$ is randomly selected from C_H. $\eta \in [0, 1]$ is a parameter that determines how much information for $x_t^{(L)}$ transferred to $x_t^{(h)}$. The greater α is the more information will be transferred from $x_t^{(L)}$ to $x_t^{(H)}$. To increase the diversity of particles, a mutation strategy is applied to the particle $x_t'^{(L)}$ according to mutation probability ρ_m:

$$x_t^{(L)} = \begin{cases} x_t'^{(L)} + \gamma & \text{if} \quad \theta_l \le \rho_m \\ x_t'^{(L)} & \text{if} \quad \theta_l > \rho_m \end{cases} \tag{19}$$

where vector γ is taken from a uniform distribution, θ_1 is the random variable for $x_t'^{(L)}$ that is drawn from the uniform distribution on [0, 1] and ρ_m denotes the mutation probability. The crossover and mutation operators can move particles to the region of high likelihood. The choice of ρ_m and η has a direct impact on the moving speed. The weighting of modified of particles $x_t'^{(L)}$ are calculated as:

$$w_t^{(L)} = w_{t-1}^{(L)} \frac{p(y_t \mid x_t'^{(L)})\, p(x_t'^{(L)} \mid x_{t-1}'^{(L)})}{N(x_t^L; x_{t|t}'^{(L)}, P_{t|t}^{(L)})} \tag{20}$$

After crossover and mutation, whenever effective number of particles N_{eff} is bellow a predefined threshold, the resampling procedure is performed in AUPF. The effective number of particles N_{eff} is:

$$N_{eff} = \frac{1}{\sum_{m=1}^{N} \left(\frac{\omega_t^{(m)}}{\sum_{i=1}^{N} \omega_t^{(i)}} \right)^2} \tag{21}$$

In summary, the flowchart of the proposed method is shown in Fig. 1.

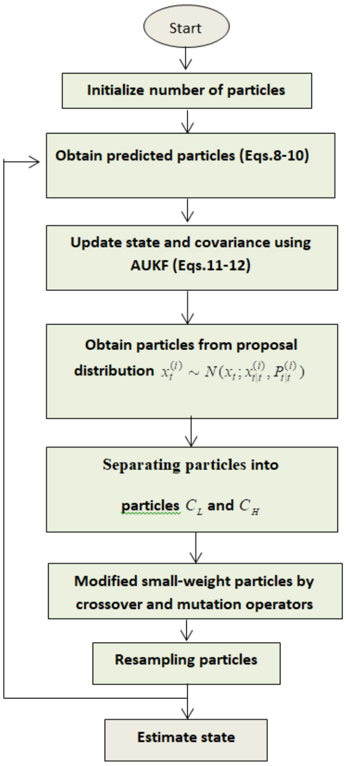

Start
Initialize number of particles
Obtain predicted particles (Eqs.8-10)
Update state and covariance using AUKF (Eqs.11-12)
Obtain particles from proposal distribution $x_t^{(i)} \sim N(x_t; x_{t|t}^{(i)}, P_{t|t}^{(i)})$
Separating particles into particles C_L and C_H
Modified small-weight particles by crossover and mutation operators
Resampling particles
Estimate state

Fig. 1 Flowchart of proposed method.

Table 1 Complexity computation.

Operations	Complexity of algorithms		
	AUPF	UPF	EPF
Sampling	O(M)	O(M)	O(M)
Importance weight	O(M)	O(M)	O(M)
Crossover and mutation operators	O($M^{(L)}$)	–	–
Resampling step	O(M)	O(M)	O(M)

3.4 Computational Complexity

The computational complexity of the proposed algorithm is related to the sampling strategy, calculation of the importance weight, modification of the small-weight particles using crossover and mutation operators of GA, and resampling step. As the computational complexity of UKF is equivalent to EKF, the number of particles M and the number of iterations in GA k determines the complexity of the proposed algorithm. In the proposed algorithm, the complexity of modification of small-weight particles are modified using the information of the large-weight particles through crossover and mutation operator of GA is $O(KM^{(L)})$ that can be approximated to $O(M^{(L)})$ as GA does not need to iterate so many times. This is because searching space is a small area around the position at times step t-1. In addition, computing the proposal distribution, calculating the importance weights is $O(M)$. Also, the complexity of the resampling step is $O(M)$. Table 1 depicts the complexity of individual operations in algorithms. As can be seen, the proposed algorithm has an additional operation. However, the proposed method requires fewer particles to obtain the same estimation accuracy compared to other methods as can be seen from Table 2 and Fig. 5 in the results section. Therefore, the complexity of the proposed algorithm is almost the same as those of other methods.

4 Experimental Results

In this section, two numerical examples are provided to illustrate the effectiveness of the proposed algorithm. One is a single-dimensioned system, which is popular in econometrics and has been used previously in [1-3]. The other example is the target tracking which is multi-dimensioned and is of interest in defense applications.

4.1 Univariate Growth Model

The univariate growth model is a benchmark model that is commonly used in PF testing. This system has been used before in many papers [1-3]. The state-space model of the system is as follows:

$$x_t = 0.5x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + \omega_{t-1}$$

$$y_t = \frac{x_t^2}{20} + v_t$$

where ω_t and v_t are Gaussian white noise signals with zero means and variances $Q_{t\text{-}1}$ and R_t, respectively. In this example, the number of time steps t = 60, the covariance of the measurement noise is R_t = 1, the covariance of the process noise is $Q_{t\text{-}1}$ = 1, and the initial state is x_0 = 0.1. Fifty particles (i.e., N = 50) is used in the related PFs (i.e., EPF, UPF, AUPF). For AUPF, the parameter η is set to 0.1 and the mutation probability ρ_m = 0.5.

At first, the performance of the proposed method is compared with UPF and EPF when the statistical properties of noises are known a priori. In Fig. 2, the tracking performance of the algorithms is depicted. It can be observed that the best state estimation results belong to AUPF.

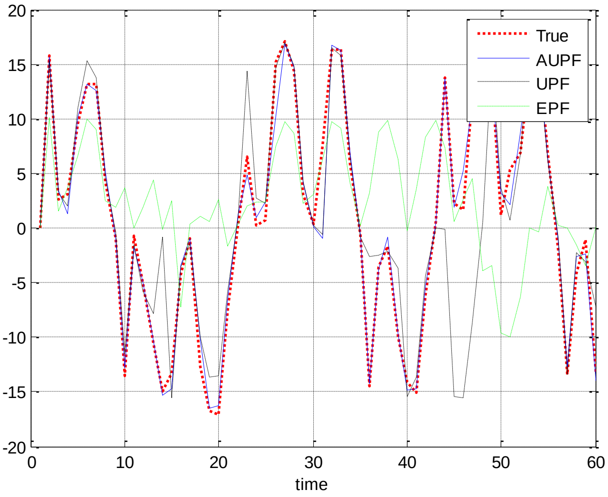


Fig. 2 The state estimation results of EPF, UPF and AUPF.

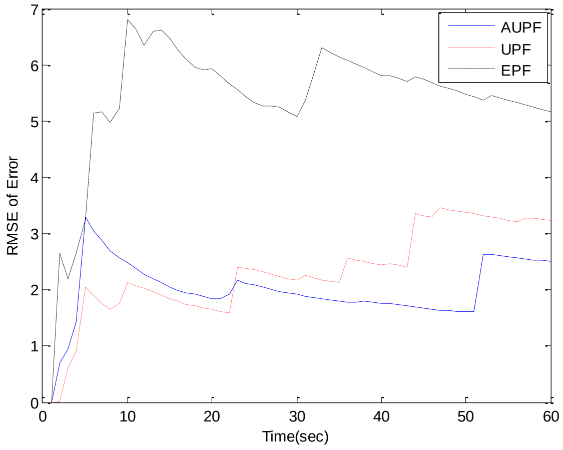


Fig. 3 RMSE of algorithms with respect to time.

To evaluate the accuracy of the estimation, the root mean square error (RMSE) is used for each algorithm. The RMSE is obtained over 100 runs. The RMSE of estimations over time is shown in Fig. 3. It can be seen that AUPF provides a more accurate estimation than EPF and UPF since the diversity of particles is increased in AUPF. In fact, the crossover operator can obviously increase the number of meaningful particles after resampling and enhance the performance of estimation. As a result, it can enrich the particle species and capture the distribution more comprehensively and accurately. The number of distinct particles is shown in Fig. 4. As can be seen, the number of distinct particles in AUPF is more than that of other algorithms. As a result, the consistency of AUPF is increase.

To examine the sensitivity of the performance of algorithms with the number of particles, three different particle numbers of 50, 30, and 10, are used for simulation tests with the covariance of the process noise $Q_{t\text{-}1}$ = 1and measurement noise R_t = 1. As is shown in Fig. 5, in general, when the number of particles

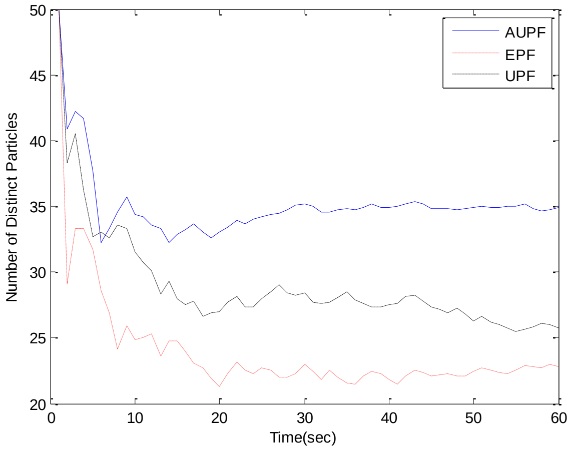


Fig. 4 The number of distinct particles.

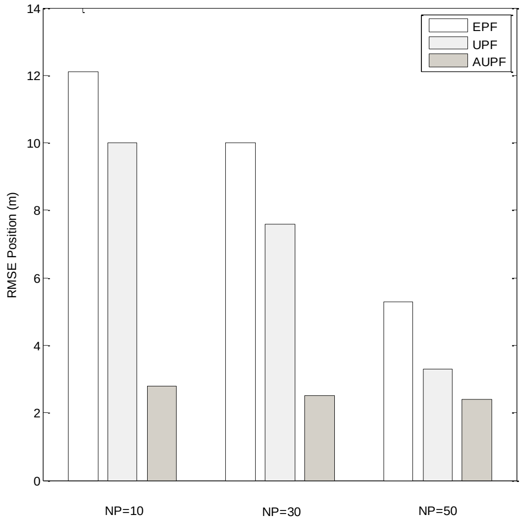


Fig. 5 The performance of algorithms with the number of particles.

increases, the mean of RMSE becomes smaller in all algorithms. However, AUPF gives the least averaged variance with different particles. The effectiveness of AUPF has been demonstrated related to incorporating the crossover and mutation operators, the particle impoverishment encountered in the general PF is mitigated.

The computational cost of algorithms is evaluated using the MATLAB simulations on same PC. Table 2 shows the computational cost of algorithms. The smallest computation time, as expected, belongs to EPF. Compared with UPF and EPF, AUPF have the additional procedure, i.e. performing genetic operators, which consumes part of the computational resource. However, as can be seen from Fig. 5 and Table 2, the proposed method requires less number of particles to obtain the same estimation accuracy compare to EPF and UPF. Therefore, considering the additional operator of AUPF, the computational cost of the proposed algorithm is almost the same as that of UPF.

Finally, the performance of the proposed method is evaluated when the statistical properties of noises are unknown. Fig. 6 shows the comparison of algorithms when the measurement noise and the process noise are considered wrongly as $Q = 5$ and $R = 5$, respectively. From this figure, it can be seen that AUPF performs significantly better than other algorithms. Compared AUPF, the state estimate of UPF and EPF deviated from the true state very large. The RMSE of estimation over time is presented in Fig. 7. It can be seen that the performance of AUPF is better than that of other algorithms. From comparing Figs. 3 and 6, it is clear that the performance of AUPF, in this case, is almost close to the previous case. However, the performance of UPF and EPF, in this case, is worse than that of UPF and EPF in the previous case. This is why AUPF can estimate the unknown measurement noises online as shown in Fig. 8 whereas the UPF and EPF depend on the fixed prior knowledge about the measurement noises. In addition, the AUPF incorporates the crossover

Table 2 Computational cost of EPF, UPF, and AUPF.

Number of particles	Computation time(s)		
	AUPF	UPF	EPF
10	0.4249	0.1831	0.0718
30	0.572	0.4249	0.1839
50	0.725	0.6784	0.2569

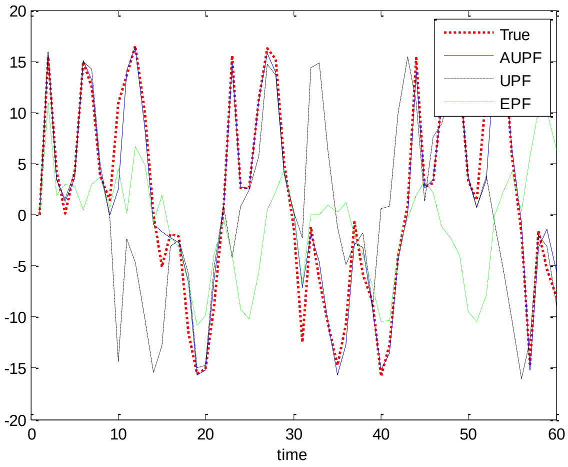


Fig. 6 The state estimation results of EPF, UPF, and AUPF.

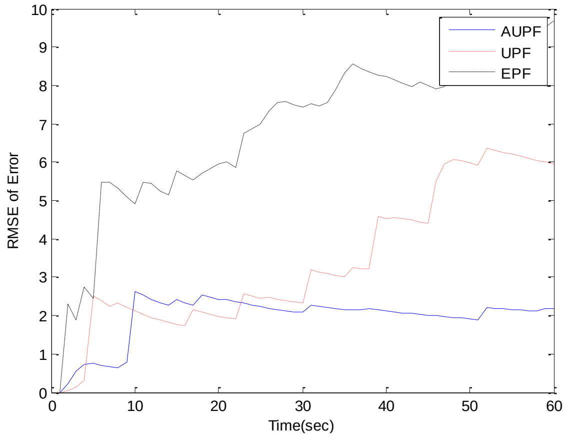
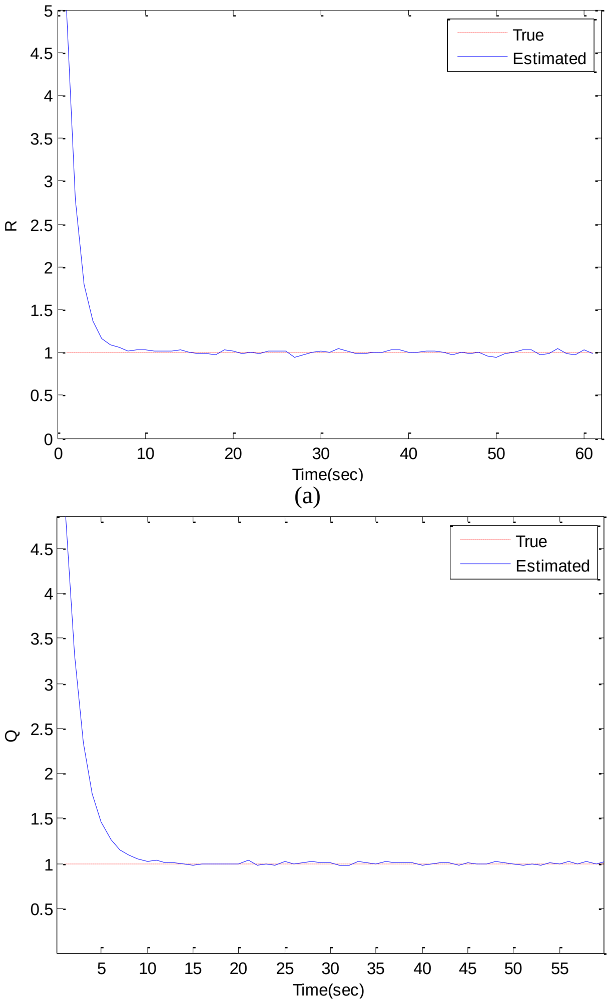


Fig. 7 RMSE of algorithms with respect to time.



(a)

(b)

Fig. 8 Adaptive estimation of Q and R using the proposed method.

operator to mitigate the particle impoverishment problem. When the particle weight is less than a predefined threshold value, the crossover operator is performed. To verify the consistency of AUPF, the average normalized estimation error squared (ANEES) is used:

$$\varepsilon_t = (x_t - \hat{x}_{t|t})^T \hat{P}_{t|t}^{-1} (x_t - \hat{x}_{t|t})$$

where x_t is ground truth. The consistency is evaluated by performing multiple Monte Carlo runs and computing the average the average normalized estimation error squared (ANEES). Given N runs, ANEES is computed as:

$$\bar{\varepsilon}_t = \frac{1}{N}\sum_{i=1}^{N} \varepsilon_{it}$$

The algorithm is consistent if $\bar{\varepsilon}_t$ belongs to $[r_1\ r_2]$ with probability 95% (i.e. $P\{\bar{\varepsilon}_t \in [r_1 \quad r_2]\} = 0.95$). For the 1-dimensional state with 20 Monte Carlo simulations, the two-sided 95% probability concentration region for $\bar{\varepsilon}_t$ is bounded by interval [0.69, 1.35] and the algorithm is consistent if $\bar{\varepsilon}_t$ with probability 95% belong to [0.69, 1.35]. Fig. 9 shows that AUPF method is consistent while other algorithms are is inconsistent. In summary, among these three methods, the proposed algorithm has well robust in terms of accuracy and consistency in a different condition, especially when the statistics of noises are unknown.

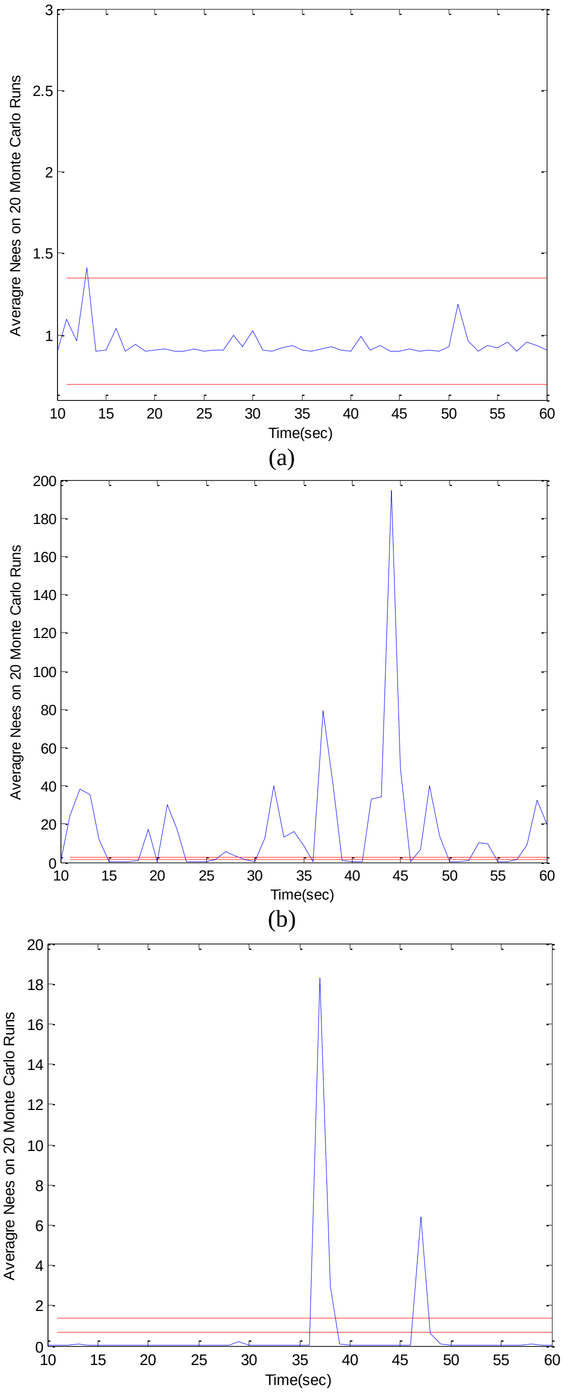
4.2 Target Tracking

To evaluate the performance of AUPF in the case of a multidimensional model, a tracking problem is considered which is of interest in defense applications. The objective of the target tracking is to track the position and the velocity of a moving target using noise-corrupted range and angle measurement. This problem is as follows [43, 44]:

$$x_t = \begin{bmatrix} 1 & \dfrac{\sin\Omega_{t-1}T}{\Omega_{t-1}} & 0 & \dfrac{1-\cos\Omega_{t-1}T}{\Omega_{t-1}} \\ 0 & \cos\Omega_{t-1}T & 0 & -\sin\Omega_{t-1}T \\ 0 & \dfrac{1-\cos\Omega_{t-1}T}{\Omega_{t-1}} & 1 & \dfrac{\sin\Omega_{t-1}T}{\Omega_{t-1}} \\ 0 & \sin\Omega_{t-1}T & 0 & \cos\Omega_{t-1}T \end{bmatrix} x_{t-1}$$

$$+\begin{bmatrix} \dfrac{T}{2} & 0 \\ T & 0 \\ 0 & \dfrac{T}{2} \\ 0 & T \end{bmatrix}\omega_{t-1}, \qquad y_t = \begin{bmatrix} \sqrt{x_t^2 + y_t^2} \\ \arctan(\dfrac{y_t}{x_t}) \end{bmatrix} + v_t$$

where Ω_{t-1} is the turn rate of the vehicle, T is the sampling interval, y_t is measurement, and $x_t = [x_t, \dot{x}_t, y_t, \dot{y}_t]^T$ is the state vector that (x_t, y_t) and $(\dot{x}_t, \dot{y}_t)$ are the position and velocity components, respectively. $\omega_t = [\omega_{tx} \quad \omega_{ty}] \sim N(0, Q_t)$ and $v_t \sim N(0, R_t)$ are independent Gaussian noises where the covariance of process and measurement noises are Q = diag(0.01^2 0.01^2) and R = diag(1^2 1^2), respectively. The tracking position error is defined as



(a)

(b)

(c)

Fig. 9 Consistency: a) proposed method, b) EPF, and c) UPF.

$\sqrt{(x_t-\hat{x}_t)^2+(y_t-\hat{y}_t)^2}$ where $[\hat{x}_t \quad \hat{y}_t]^T$ is the *x*-*y* position estimate of the target at time *t*. To show the effectiveness of AUPF, this method and other algorithms are used to track the same maneuvering target. For this purpose, the simulation scenario of target maneuvering is designed as follows. The target is supposed that is located at the position (0m, 0m). For 150s, the target starts to make a positive turn rate of about 5°/s. Then it turns for 150 s with a negative turn rate of about -5°/s. The true trajectory for a maneuvering target is shown in Fig.10. In all the simulations, 100 Monte Carlo runs have been performed, and the number of particles is 50 with $T = 0.12$s.

The true state and the estimates of the tree filters are shown in Figs. 11 and 12. As shown, for most of the states that are not correctly estimated by EPF and UPF, AUPF offers much more accurate estimation results. From Figs. 11 and 12, it can be observed that PF cannot give the correct state estimation for the states of position and velocity. Compared with EPF and UPF, AUPF offers better state estimation results for states. However, the best state estimation result belongs to AUPF that gives the lowest variance as shown in Figs. 11 and 12.

The RMSE of position over time is shown in Fig. 13. The RMSE is obtained over 100 Monte Carlo runs. It is observed that AUPF has a more accurate estimation than EPF and UPF since the diversity of particles is increased in AUPF. The crossover operator can obviously increase the number of meaningful particles after resampling and enhance the performance of estimation and can enrich the particle species and capture the distribution more comprehensively and accurately.

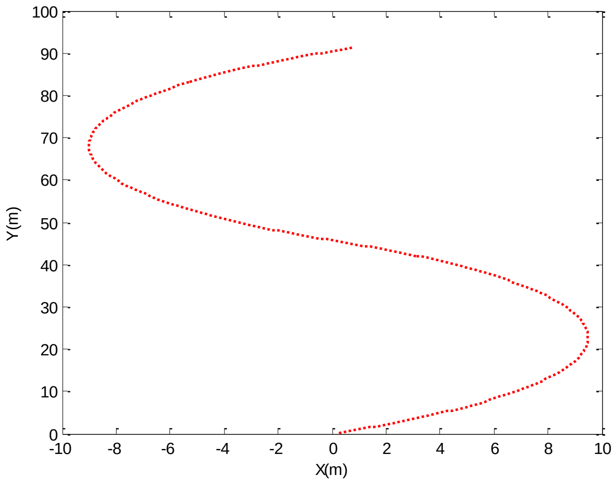


Fig. 10 Actual trajectory.

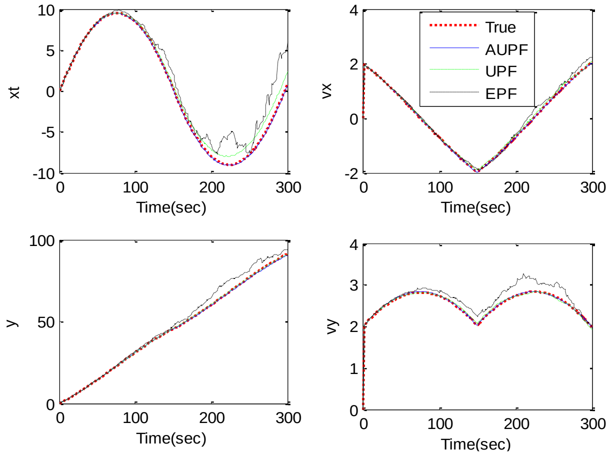


Fig. 11 The state estimation results of EPF, UPF, and AUPF.

5 Conclusion

The particle filter is widely used in many applications, special target tracking problem. However, the particle filter suffers due to the particle impoverishment problem and incorrect a priori knowledge of noises. In order to overcome the particle degradation and non-adjusted online in the traditional particle filter algorithm, AUPF is proposed. The AUPF uses AUKF filter to generate a proposal distribution and the genetic operators to further improve the particle diversity. With this strategy, the small-weight particles are modified to the large-weight ones and finally the

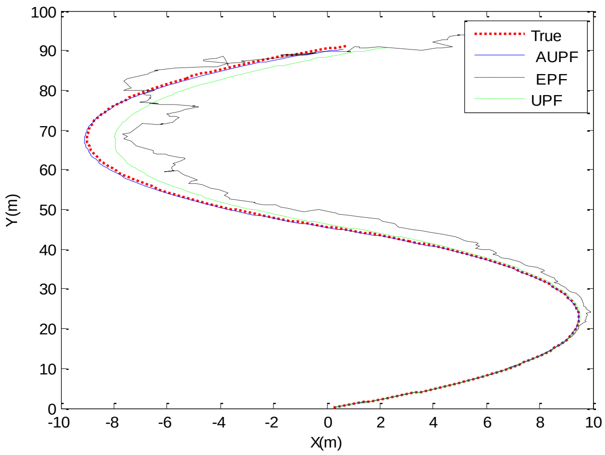


Fig. 12 Actual and estimated trajectories.

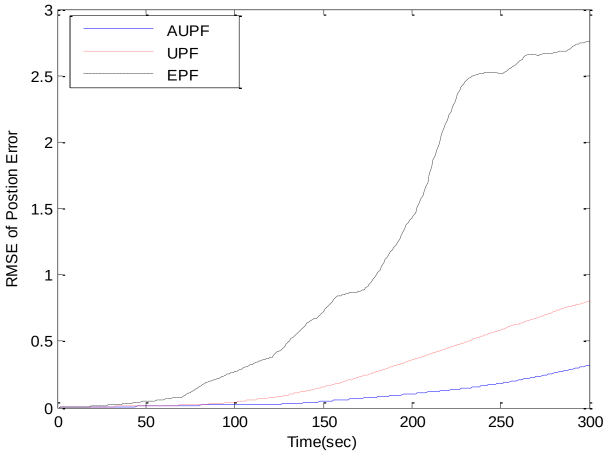


Fig. 13 RMSE of algorithms with respect to time.

particle impoverishment problem is mitigated. Compared with the general PF, the posterior distribution in AUPF can be more sufficiently approximated by particles. The effectiveness of AUPF is demonstrated by using two experiment examples. The simulation results demonstrate the effectiveness of the proposed method. As can be seen from results, the proposed method achieves not only much better performance than the existing methods but also robustness against model uncertainty and therefore it can be especially useful in practical applications.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for Online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*,Vol. 50, No. 2, pp. 174–188, 2002.

[2] M. Sadat Sharifian, A. Rahimi, and N. Pariz, “Classifying the weights of particle filters in nonlinear systems,” *Communications in Nonlinear Science and Numerical Simulation*, Vol. 31, pp. 69–75,2016.

[3] S. S. Dias and M. G. Bruno, “Cooperative target tracking using decentralized particle filtering and RSS sensors,” *IEEE Transactions on Signal Processing*, Vol. 61, No. 14, pp. 3632–3646, 2013.

[4] S. A. Hiremath, G. W. Van Der Heijden, F. K. Van Evert, A. Stein, and C. J. Ter Braak, “Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter,” *Computers and Electronics in Agriculture*, Vol. 100, pp. 41–50, 2014.

[5] M. Atia, J. Georgy, M. Korenberg, and A. Noureldin, “Real-time implementation of mixture particle filter for 3D RISS/GPS integrated navigation solution,” *Electronics Letters*, Vol. 46, No.15, pp. 1083–1084, 2010.

[6] S. Park, J. Hwang, E. Kim, and H. Kang, “A new evolutionary particle filter for the prevention of sample impoverishment,” *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 4, pp. 801–809, 2009.

[7] D. Z. Li, W. Wang, and F. Ismail, “A mutated particle filter technique for system state estimation and battery life prediction,” *IEEE Transactions on Instrumentation and Measurement*, Vol. 63, No. 8, pp. 2034–2043, 2014.

[8] H. Moradkhani, C. M. DeChant, and S. Sorooshian, “Evolution of ensemble data assimilation for uncertainty quantification using the particle filter-Markov chain Monte Carlo method,” *Water Resources Research*, Vol. 48, No. 12, 2012.

[9] N. Papadakis, E. Mémin, A. Cuzol, and N. Gengembre, “Data assimilation with the weighted ensemble Kalman filter,” *Tellus A: Dynamic Meteorology and Oceanography*, Vol. 62, No. 5, pp. 673–697, 2010.

[10] M. Bolic, P. M. Djuric, and S. Hong, “Resampling algorithms and architectures for distributed particle filters,” *IEEE Transactions on Signal Processing*, Vol. 53, No. 7, pp. 2442–2450, 2005

[11] X. Fu and Y. Jia, “An improvement on resampling algorithm of particle filters,” *IEEE Transactions on Signal Processing*, Vol. 58, No. 10, pp. 5414–5420, 2010

[12] T. Li, T. P. Sattar, and S. Sun, “Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters,” *Signal Processing*, Vol. 92, No. 7, pp. 1637–1645, 2012.

[13] C. Fen, M. Wang, and Q. B. Ji, “Analysis and comparison of resampling algorithms in particle filter,” *Journal of System Simulation*, Vol. 21, No. 4, pp. 1101–1105, 2009.

[14] D. A. P. Guingla, R. De Keyser, G. J. De Lannoy, L. Giustarini, P. Matgen, and V. R. Pauwels, “Improving particle filters in rainfall-runoff models: Application of the resample-move step and the ensemble Gaussian particle filter,” *Water Resources Research*, Vol. 49, No. 7, 2013.

[15] H. Wang and Z. Jing, “Adaptive unscented particle filter based on predicted residual,” in *6th IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Vol. 2, pp. 181–184, 2011.

[16] X. Li, G. Shesheng, and Z. Yongmin, “Robust adaptive unscented particle filter, international journal of intelligent mechatronics and robotics,” Vol. 3, No. 2, pp. 55–66, 2013.

[17] N.Zhanga and X. Yang, “Gaussian mixture unscented particle filter with adaptive residual resample for nonlinear model,” in *2nd International Conference on Intelligent Computing and Cognitive Informatics (ICICCI 2015)*, Atlantis Press, 2015.

[18] C. Musso, N. Oudjane, and F. Le Gland, “Improving regularized particle filters,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York, NY, USA: Springer-Verlag, 2001.

[19] J. H. Kotecha and P. M.Djuric, “'Gaussian particle filtering,” *IEEE Transactions on Signal Process*, Vol. 51, No. 10, pp. 2593–2602, 2003.

[20] S. P. Won, W. W. Melek, Senior, and F. Golnaraghi, "A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 5, pp. 1787–1798, May 2010.

[21] X. Qiang, Y.Zhu, and R. Xue, "SVRPF: An improved particle filter for a nonlinear/non-gaussian environment," *IEEE Access*, Vol. 7, 2019.

[22] S. A. Daneshyar and M. Nahvi, "Moving objects tracking based on improved particle filter algorithm by elimination of unimportant particles," *Optik*, Vol. 138, pp. 455–469, 2017.

[23] C. Zhang, A. Taghvaei, and P. G. Mehta, ''Feedback particle filter on Riemannian manifolds and matrix lie groups,'' *IEEE Transactions on Automatic Control*, Vol. 63, No. 8, pp. 2465–2480, Nov. 2018.

[24] M. D. Jenkins, P. Barrie, T. Buggy, and G. Morison, "Selective sampling importance resampling particle filter tracking with multibag subspace restoration" *IEEE Transactions on Cybernetics*, Vol. 48, No. 1, pp. 264–276, 2018.

[25] N. Aunsri and K. Chamnongthai, "Particle filtering with adaptive resampling scheme for modal frequency identification and dispersion curves estimation in ocean acoustics," *Applied Acoustics*, Vol. 154, pp. 90–98, 2019

[26] A. M. Kouritzin, "Residual and stratified branching particle filters," Computational Statistics & Data Analysis, Vol. 111, pp. 145–165, Jul. 2017.

[27] M. Narayana, H. Nenavath, S. Chavanb, and L. Koteswara Raoc "Intelligent visual object tracking with particle filter based on Modifed Grey Wolf Optimizer," *Optik - International Journal for Light and Electron Optics*, Vol. 193, 2019.

[28] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, Vol. 15, No. 2, pp. 175–184, 1970.

[29] R. J. Fitzgerald, "Divergence of the Kalman filters," *IEEE Transactions on Automatic Control*, Vol. 16, No. 6, pp. 736–747, 1971.

[30] C. Hide, T. Moore, and M. Smith, "Adaptive Kalman filtering for low cost INS/GPS," *The Journal of Navigation*, Vol. 56, pp. 143–152, 2003.

[31] C. Lin, Y. Chang, C. Hung, C. Tu, and C. Chuang, "Position estimation and smooth tracking with a fuzzy-logic-based adaptive strong tracking Kalman filter for capacitive touch panels," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 8, 2015.

[32] X. Liu, H. J. Liu, Y. G. Tang, Q. Gao, and Z. M. Chen, "Fuzzy adaptive unscented Kalman filter control of epileptiform spikes in a class of neural mass models," *Nonlinear Dynamics*, Vol. 76, pp. 1291–1299, 2014.

[33] M. Q. Zhu, J. J. Hou, Y. Liu, and J. F. Su, "Target locating estimation algorithm based on adaptive scaled unscented Kalman filter," *ACTA Armamentar II*, Vol. 34, No. 5, pp. 561–566, May 2013.

[34] X. L. Ning, P. P. Huang, J. C. Fang, G. Liu, and S. Z. S. Ge, "An adaptive filter method for spacecraft using gravity assist," *Acta Astronautica*, Vol. 109, pp. 103–111, Apr. 2015.

[35] H. X. Le and S. Matunaga, "A residual based adaptive unscented Kalman filter for fault recovery in attitude determination system of microsatellites," *Acta Astronautica*, Vol. 105, No. 1, pp. 30–39, Dec. 2014.

[36] A. Rahimi, K. D. Kumar, and H. Alighanbari, "Fault estimation of satellite reaction wheels using covariance based adaptive unscented Kalman filter," Acta Astronautica, Vol. 134, pp. 159–169, May 2017.

[37] Y. Meng, S. S. Gao, Y. M. Zhong, G. G. Hu, and A. Subic, "Covariance matching based adaptive unscented Kalman filter for direct filtering in INS/GNSS integration," *Acta Astronautica*, Vol. 120, pp. 171–181, Dec. 2016.

[38] B. B. Gao, S. S. Gao, Y. M. Zhong, G. G. Hu, and C. F.Gu, "Interacting multiple model estimation-based adaptive robust unscented Kalman filter," *International Journal of Control, Automation and Systems*, Vol. 15, No. 5, pp. 2013–2025, Oct. 2017.

[39] L. Wang and S. X. Li, "Enhanced multi-sensor data fusion methodology based on multiple model estimation for integrated navigation system," *International Journal of Control, Automation and Systems*, Vol. 16, No. 1, pp. 295–305, Feb. 2018.

[40] D. E Goldberg and J. H Holland, "Genetic algorithms and machine learning," *Journal of Machine Learning*, pp. 95–99, 1988.

[41] L. Cen, Z. L. Yu, W. Ser, and W. Cen, "Linear aperiodic array synthesis using an improved genetic algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 60, No. 2, pp. 895–902, 2012.

[42] Y. Yoon and Y. H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, Vol. 43, No. 5, pp. 1473–1483, 2013.

[43] M. S. Djogatović, M. J. Stanojević, and N. Mladenović, "A variable neighborhood search particle filter for bearings-only target tracking," *Computers & Operations Research*, Vol. 52, pp. 192–202, 2014.

[44] S. Chena and W. Huang, "Maneuvering target tracking from nautical radar images using particle-Kalman filters," *Journal of Electromagnetic Waves and Applications*, Vol. 27, No. 18, pp. 2366–2378, 2013.

R. Havangi received his M.S. and Ph.D. degrees from the K.N. Toosi University of Technology, Tehran, Iran, in 2003 and 2012, respectively. He is currently an Associate Professor of control systems with the Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. His main research interests are inertial navigation, integrated navigation, estimation and filtering, evolutionary filtering, simultaneous localization and mapping, fuzzy, neural network, and soft computing.