

Adaptive Double-Resampling Particle Filter Algorithm for Target Tracking

Qiang Zhao, Chen Wei, Liang Qi and Wenhua Yuan

Abstract Based on the traditional particle degradation and depleted of particle filter and the number of particle set, which cannot be adaptive to change brought by the filtering accuracy and convergence rate of decline. A new methods of Innovation and resampling particle filter was applied to the paper. This approach can solve the problems mentioned above. The algorithm first uses the observation information to establish the particle distribution program of the resampling. Then to conduct a resampling on the basis of the initial resampling. The second resampling used the particle cross aggregation algorithm. This can improve efficiency of the particles, and avoid the increase of the calculation when using too many particles. The simulation result based on the DR/GPS shows that compared with the traditional PF algorithm, the algorithm can improve the accuracy and stability of the filter.

Keywords Double-resampling · Particle filter · Innovation · Adaptive · Target tracking

1 Introduction

The particle filter uses a weighted particle set to describe the probability distribution of state variable, using Monte Carlo method to realize Recursive Bayesian filter, it is an effective nonlinear non-Gaussian system suboptimal prediction method, and has been successfully applied in the field of target tracking [1–3]. But in order to correctly approximate the posterior probability density, it needs a large quantity of frequency samples. However, a large number of update operations of particle optimization will lead to lower efficiency of the algorithm. It is the main defect of particle filter.

Q. Zhao (✉) · C. Wei · L. Qi · W. Yuan
School of Electronics and Information, Jiangsu University of Science and Technology,
Zhenjiang 212003, Jiangsu, China
e-mail: zhaoqiang_200@163.com

In order to improve the efficiency of particle filter, the most direct and effective way to improve the efficiency of particle filter is to adjust the size of the particle size. It can achieve the purpose of filtering accuracy and reasonable operation speed. In fact, the particle filter has the problems of particle degeneracy and impoverishment, it leads to filter failure because too few effective particles, the most fundamental reason is that the probability distribution description ability is insufficient, the filter is premature convergence. FOX [4] proposed a Kullback-Leibler Distance (KLD) sampling method which realize the adaptive adjustment of particle sets by computing the K-L distance between posterior probability distribution and maximum likelihood probability distribution of particle sets. In addition, some researches about adaptive adjustment method [5] of particle set were proposed to reduce the number of sampled particles [5], but the rationality of particle distribution is not considered.

At the present, the main research means is to improve the micro-ability and adaptability of particle set [4]. Therefore, on the basis of analyzing a large number of literature, this paper puts forward a kind of double sampling adaptive particle filter algorithm. First, the proposed algorithm takes the new observed value of the target state and the residual error of the prediction result as the current observation. Then, according to the new information to reflect the relationship between the accuracy of the target prediction and the uncertainty of the system, the noise variance of the new information estimation system is determined, and use it to determine the sampling of the proposed distribution [6–9]. If predication is accurate, this method can obtain accurate density estimation only using a small amount of particles. When target motion changes greatly, the method uses more samples to ensure accurate tracking. The actual operation is to use a double-resampling when the sample set has more particles. First step, according to the new information of observation to control the number of particles double sampling, it can enhance the ability to predict the particle set. The second step, according to the particle space distribution to control particle size, it can ensure the consistency of particle size and particle spatial scale. This method has the rationality of spatial distribution. The Monte Carlo simulation experiment shows that this method can balance the particle diversity in prediction stage and particle size in update phase. Under the premise of keeping the accuracy of the particle filter, it can effectively reduce the calculation of the particle filter.

2 The Resampling Distribution Scheme Based on the Observed Innovation

Considering the following nonlinear discrete dynamic system is:

$$\begin{cases} x_{t+1} = f_t(x_t, w_t) \\ z_t = h_t(x_t, v_t) \end{cases} \quad (1)$$

In this formula, $x_t \in R^n$ is n dimensional vector state vector at time t. $z_t \in R^m$ is m dimensional observation vector, w_t is system noise, v_t is observation noise, they variance obey $\left[\begin{matrix} (\sigma_x^t)^2 & \\ & (\sigma_z^t)^2 \end{matrix} \right]$ distribution.

Assume \hat{x}_t and x_t^p are the estimation and predication status value in step t respectively. And $e^t = \hat{x}_t - x_t^p$ is observation innovation at current time. Assume that state vector x_k predicts more accurately, then σ_x^t should be small, the new sampling should be located in a small neighborhood prediction area. However, when x_t predicts not accurate, σ_x^t should be taken larger value. Enable the sampling distribution range is large enough to contain the real target state. Thus, the system noise variance can be express as:

$$\sigma_x^t = \begin{cases} \min(e^t, \sigma_{\max}) & \text{if } e^t \geq \sigma_x^{t-1} \\ \max(\alpha e^{t-1}, e^t, \sigma_{\min}) & \text{if } e^t < \sigma_x^{t-1} \end{cases} \quad (2)$$

In order to avoid poor particle phenomena caused by poor system noise, setting the lower limit of the system noise variance is σ_{\min} , and the σ_{\max} is the system maximum noise variance. In order to avoid the estimation error makes the system noise decrease quickly, the algorithm uses the attenuation factor α to control the reduction speed of noise variance, in the experiment $\alpha = 0.9$.

The number of particle is related to system noise, when the system noise is small, using a small amount of particles can approximate the target distribution; and when the system noise is bigger, the particle sampling range will be expanded, so should increase the number of particles. Using Sigmoid function to represents the relationship between the number of particles and uncertainty measure.

$$N_t = N_{\min} + (N_{\max} - N_{\min}) \left(\frac{2}{1 + \exp(-\beta(r_k - r_{\min}))} - 1 \right) \quad (3)$$

In the formula, N_{\min} expresses the min particles, and N_{\max} expresses the max particles. The uncertainty measure $r_t = \sigma_x^t \sigma_y^t$ of the target state in the t moment is estimated according to the state covariance. The lower limit of uncertainty measure is $r_{\min} = \sigma_{x\min} \sigma_{y\min}$, $\beta = 0.01$ is control coefficient.

3 Adaptive Particle Filter Algorithm Based on Double-Resampling

3.1 Particles Sparse Polymerization Double-Resampling

When the target motion changes greatly, or the carrier surrounding environment changes, the algorithm needs more samples to ensure the effectiveness of the filter, and at this time, the spatial distribution of particles also shows clustering trend. At

this time, the particles' space differentiation appears redundant causing that updating particles centralization values consumes too much computing resources. In this paper, before updating the weights of particle set, the algorithm weighted aggregation for particles based on spatial scale mesh of particles, to reduce the size of the particle set. So the algorithm is called particles sparse polymerization resampling [10].

Firstly, this paper gives the definition of a grid dividing the state space:

Definition 1 if dividing the i dimension of n dimensional space S into m_i closed left and right open intervals which are equal length. Thus, it can divide the whole space S into $m_1 \times m_2 \times \dots \times m_n$ disjoint n dimensional grid g_n .

Definition 2 using the sample particles which attach to grid cell g_i on the space to express grid density of g_i , denoted as $\text{den}(g_i)$. As $\text{den}(g_i) = 0$, it says g_i is empty; otherwise, it says g_i is not empty. The current grids and the adjacent grids in its $l(l < n)$ dimension direction compose a corresponding mesh grid set of 3^l scale. This paper uses “#” to mark the grid set relevant variables. The k dimensional grid set corresponding to grid g_i is marked $\#g_i^l$. When $l = n$, $\#g_i^l$ is called the whole dimensional grid set. When $l < n$, $\#g_i^l$ is called the non-whole dimensional grid set. Based on the weights of the particles, taking weighted average with all the particles within the unit space to obtain a polymeric particle, and the unit space is called the polymerization unit. The particle aggregation method with a grid set as polymerization unit implementation is called cross polymerization.

3.2 Particle Cross Polymerization

Assuming that the particles are distributed in a k non-empty grid cell, the grid cell containing N_k particles can be described as $g_k: \{\omega_k^i, x_k^i | i = 1, 2, \dots, N_k\}$, in the formula $k = 1, 2, \dots, K$, x_k^i denotes as the state of particle, ω_k^i denotes as the weight of particle, the corresponding l dimension grid set is $\#g_k^l: \{x_k^i, \omega_k^i | i = 1, 2, \dots, \#N_k\}$, $\#N_k^l$ is the number of particles including in the polymerization unit $\#g_k^l$. Taking the grid set as the polymerization unit, all particles within the polymerization unit weight combination to get the aggregated particles of the central grid. Polymerization equation is:

$$\begin{cases} \hat{x}_t^k = \frac{\sum_{i=1}^{\#N_k^l} x_k^i \omega_k^i}{\sum_{i=1}^{\#N_k^l} \omega_k^i} \\ \hat{\omega}_t^k = \frac{\sum_{i=1}^{\#N_k^l} \omega_k^i}{3^l} \end{cases} \quad (4)$$

In the formula, $(\hat{x}_t^k, \hat{\omega}_t^k)$ expresses polymeric particle corresponding to the k grid. Polymerization achieves consistent with the particle collection size and particle space size. Polymeric particle shows that higher weight of the particle has a greater

impact on spatial distribution, which has a reasonable distribution of computing resources according to the weight of the particle. Each particle average weights to $3l$ polymeric particle (l is the dimension of space), averaging the distribution of particle on the certain extent, it can effectively relieve the weight degradation problem caused by reason that weight is too concentrated to a single particle. Thus, the small weight particles will not be abandoned, and it can avoid the phenomenon of particle diversity scarcity caused by the particle weight sampling, which effectively reduces the weight concentration.

3.3 Steps of Adaptive Double-Resampling Particle Filter Algorithm

Combining particle cross aggregation algorithm, the following gives the specific steps of adaptive double sampling particle filter.

(1) Initialization

According prior distribution $p(x_0)$ to initialize sampling set $\{x_0^i\}_{i=1}^{N_0}$, and ordering the particle weight is $\omega_0^i = \frac{1}{N_0}, i = 1, \dots, N_0, t = 1$.

(2) Sampling

According to the sampling weight ω_{t-1}^i , choosing N_t samples from sampling set $\{x_{t-1}^i\}_{i=1}^{N_{t-1}}$, then a group new sampling set $\{\hat{x}_{t-1}^i\}_{i=1}^{N_t}$ is obtained.

(3) Updating the weights

Each sampling value calculates the new sampling according to the state transition equation $x_t^i = \hat{x}_{t-1}^i + \omega_t, i = 1, \dots, N_t$ in the sampling set $\{\hat{x}_{t-1}^i\}_{i=1}^{N_t}$.

Ordering the weight of sampling i for the confidence coefficient of its corresponding candidate item O_i , the normalization weights is $\omega_t^i = g(O_i, \Omega)$.

(4) Double-resampling

According to the grid scale L , dividing the State grid lattice space $\{g_k | k = 1, 2, \dots, K_t\}$ and calculating the grid density;

(1) According formula (4), taking particles sparse polymerization double-resampling to generate polymeric particles set $\hat{\mathcal{S}}_t = \{x_t^k, \omega_t^k\}_{k=1,2,\dots,N_t}$.

(2) According to the Update Model $p(o_t|x_t)$, updating the weights $\omega_t^k|_{k=1,2,\dots,N_t}$ of polymeric particles.

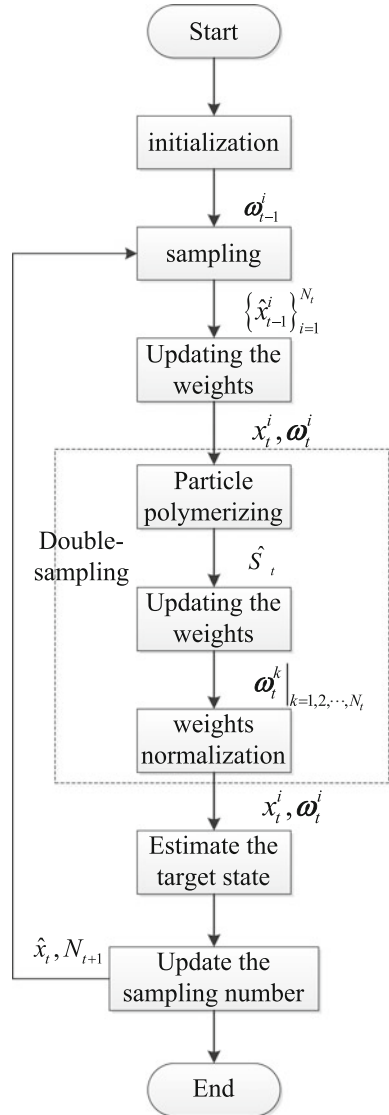
(3) Normalize the particle weights ω_t^i .

(5) Estimate the target state: $\hat{x}_t = \sum_{i=1}^N \omega_t^i x_t^i$.

- (6) Calculate predictive state: $x_t^p = \sum_{i=1}^{N_t} s_t^i$.
- (7) Calculate observation innovation: $e^t = \hat{x}_t - x_t^p$.
- (8) According step 3 to update the number of samples N_{t+1} .
- (9) Return to the step 2, going on the steps until the end of tracking.

The following figure is the algorithm structure (Fig. 1):

Fig. 1 The algorithm structure



4 Simulation Experiment and Analysis

4.1 Establish DR/GPS Integrating System Model

Taking the DR/GPS integrated navigation system as an example, the model of DR/GPS integrated navigation system is established as follows:

$$\begin{cases} X_t = \phi_{t,t-1}X_{t-1} + W_t \\ Z_t = h(X_t) + V_t \end{cases} \quad (5)$$

Here, the state transition matrix, the observation matrix and the noise matrix are respectively such as:

$$\begin{aligned} \phi_{t,t-1} &= \text{diag}[F_e, F_n] \\ F_e &= \begin{bmatrix} 1 & T & (-1 + \alpha_e T + e^{-\alpha_e T})\alpha_e^{-2} \\ 0 & 1 & (1 + e^{-\alpha_e T})\alpha_e^{-1} \\ 0 & 0 & e^{-\alpha_e T} \end{bmatrix}, F_n = \begin{bmatrix} 1 & T & (-1 + \alpha_n T + e^{-\alpha_n T})\alpha_n^{-2} \\ 0 & 1 & (1 + e^{-\alpha_n T})\alpha_n^{-1} \\ 0 & 0 & e^{-\alpha_n T} \end{bmatrix}; \end{aligned} \quad (6)$$

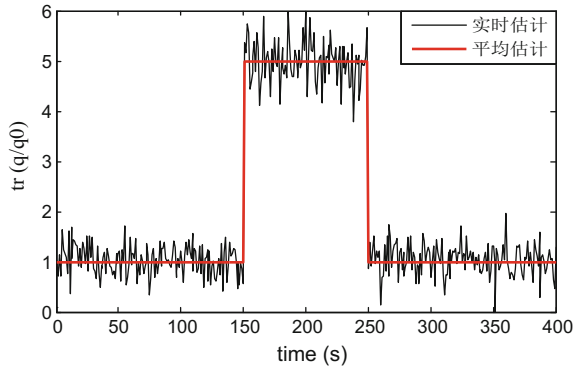
$$h(X_t) = \begin{bmatrix} x_e & x_n & \frac{v_n a_e - v_e a_n}{v_e^2 + v_n^2} & T\sqrt{v_e^2 + v_n^2} \end{bmatrix}^T \quad (7)$$

$$W_t = [0 \ 0 \ w_e \ 0 \ 0 \ w_n]^T; \quad V_t = [v_1 \ v_2 \ v_3 \ v_4]^T.$$

Here, $\alpha_e = 1/\tau_e$, $\alpha_n = 1/\tau_n$, τ_e , τ_n is time constant of vehicle about Eastward and Northward maneuver acceleration respectively. w_e , w_n is white noise sequence which is meeting the $N(0, \sigma_e^2)$ and $N(0, \sigma_n^2)$ distribution respectively. v_1 and v_2 is the observed noise from the output of the GPS receiver in the East position and North position, respectively. They are both approximated as $(0, \sigma_1^2)$ and $(0, \sigma_2^2)$ Gaussian white noise. ε_ω is gyro drift, it is approximated as $(0, \sigma_\varepsilon^2)$ Gaussian white noise. ε_s is the observation noise of the mileage meter, according to the actual situation, it is assumed to be non-Gauss, and the corresponding noise distribution is shown in Fig. 2, and the likelihood probability density function is approximate to the formula (8).

$$\begin{aligned} p(z_t|x_t) &= \frac{1}{2} \exp \left[-\frac{1}{2} (z_t - h(x_{t|t-1}) - \bar{n})^T R^{-1} (z_t - h(x_{t|t-1}) - \bar{n}) \right] \\ &+ \frac{1}{2} \exp \left[-\frac{1}{2} (z_t - h(x_{t|t-1}) + \bar{n})^T R^{-1} (z_t - h(x_{t|t-1}) + \bar{n}) \right] \end{aligned} \quad (8)$$

Fig. 2 Process noise mean trace



4.2 Simulation Experiment and Analysis

In order to analyze the performance of the proposed algorithm, this paper uses Matlab as the development language, simulation experiment is carried out on the platform of DR/GPS integrated navigation system.

The initial conditions of simulation and the related parameters are:

$$T = 1s; \sigma_e^2 = \sigma_n^2 = (0.3m/s^2)^2; a_e = a_n = 1;$$

$$\sigma_1^2 = (15m)^2; \sigma_2^2 = (16m)^2; \sigma_s^2 = (0.3m/s^2)^2;$$

Respectively, initial state value, initial state error variance matrix, initial process noise and measurement noise variance matrix is:

$$x_0 = [0 \quad 0.1 \quad 0 \quad 0 \quad 0.1 \quad 0]^T;$$

$$P_0 = \text{diag}[100, 1, 0.04, 100, 1, 0.04]; R_0 = \text{diag}(15^2, 16^2, 0.005^2, 0.7^2);$$

$$Q_0 = \text{diag}(0.1^2, 0.5^2, 0.3^2, 0.1^2, 0.5^2, 0.3^2)$$

The simulation time is set to 400 s. In order to verify the effectiveness of the proposed algorithm under different noise conditions. In the process of simulation, the vector is assumed to be a variable acceleration motion toward east and north in 150–250 s whose acceleration is $3\sin(t/10)m/s^2$. In this process, the noise variance is constant, and the mean value is enlarged to 5 times. Within the 250–400 s, the carrier is doing the uniformly accelerated motion whose acceleration is 2 m/s². At this point, the observation noise covariance is enlarged to 3 times of the initial value to increase the observation noise.

According to the simulation conditions, we use PF and APF algorithm proposed in this paper to simulate, the simulation results are shown in Figs. 2, 3, 4 and 5 and Table 1.

Fig. 3 Measurement noise covariance trace

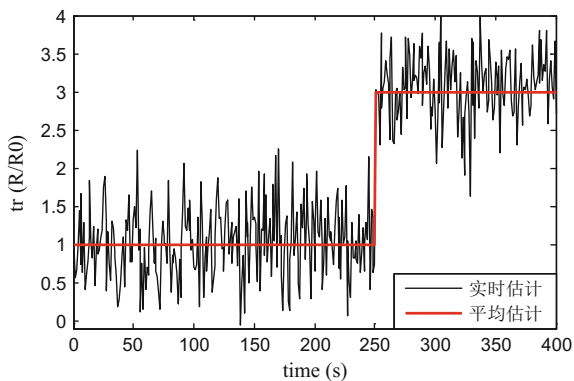


Fig. 4 Comparison of east position errors between PF and APF

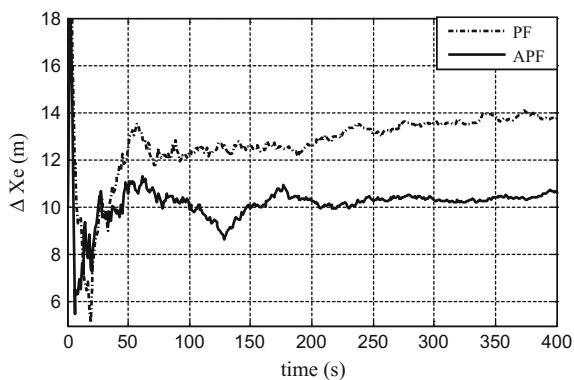


Fig. 5 Comparison of north position errors between PF and APF

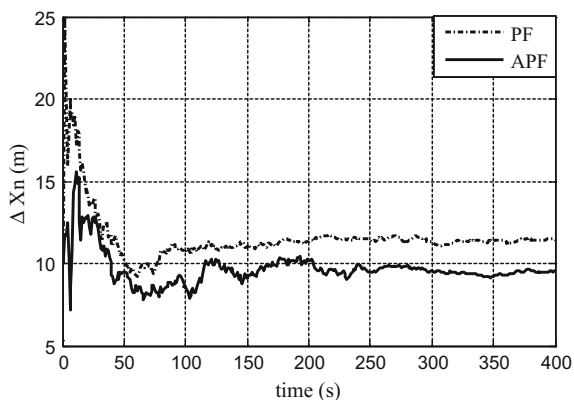


Table 1 Comparison of simulation results under different particle numbers

Algorithm	N	$\eta/\%$	E_{ARMS}	Estimated time/s
PF	100	2.563	0.3901	0.305
	500	1.004	0.1993	0.556
APF	100	6.715	0.1489	0.198
	500	8.811	0.0227	0.287

Figures 2 and 3 describe the variation of the noise in the simulation process. The comparison of the position error about the Eastward and the northward is given in Figs. 4 and 5. From Figs. 4 and 5, due to the irregular change of the noise, we can see that the traditional PF algorithm is less than the APF algorithm in the convergence speed and stability. It is mainly due to the traditional PF algorithm has more or less the particle degeneracy and impoverishment, the number of particles can't be changed adaptively. In this paper, the proposed algorithm can effectively overcome the above problems by changing the particle size and distribution.

In order to further compare the estimation accuracy of the two algorithms, we selected different particle numbers. The PF algorithm and APF algorithm are used to investigate the mean square root mean square error (ARMSE), the filter efficiency and the state estimation time. The simulation results are shown in Table 1.

Filtering efficiency in the Table 1 can be expressed as $\eta = \frac{1}{E_{ARMS}N}$. The higher filtering efficiency shows that the higher filtering accuracy can be obtained with less number of N particles. From the Table 1 we can see that when the number of particles are in the same time, we can choose N = 100, so APF algorithm has higher estimation accuracy and filtering efficiency, and needs shorter state estimation time than PF algorithm. When the number of particles are the same, choosing N = 500, and the E_{ARMS} of the PF algorithm is 0.1993, it shows that the filtering accuracy is improved and the filtering efficiency η and the state estimation time are decreased. This shows that the filtering precision is improved at the expense of filtering efficiency and state estimation time. While under the premise of the E_{ARMS} of APF algorithm is smaller than PF (i.e. filtering precision is high), the filtering efficiency is higher than PF times, the state estimation time is instead smaller than PF. This shows that the APF algorithm can achieve greater efficiency with shorter time and higher accuracy.

5 Conclusion

In this paper, an adaptive particle filter based on adaptive particle filter is proposed. It is aiming at these problems which are occurred in traditional particle filter. These problems are such as particle degradation problem, dilution problem, and the number of particle sets can't be self-changed to bring about the decline of filtering precision and convergence rate. The algorithm can determine the number of particles by real-time detection of the observation information. In the double sampling

process, this particle cross polymerization method is effective to improve the particle degeneracy and impoverishment problem and more effectively improve the filtering precision. The validity of the algorithm is proved by DR/GPS simulation experiment.

Acknowledgements This work is supported by National Natural Science Foundation of China (Grant no. 61503162, 51505193), Natural Science Foundation of Jiangsu Province in China (Grant no. BK20150473).

References

1. Brasnett P, Mihaylova L, Bull D, et al. Sequential Monte Carlo tracking by fusing multiple cues in video sequences [J]. *Image and Vision Computing* (S0262-8856), 2007, 25(8):1217–1227.
2. Zhou S K, Chellappa R, Moghaddam B. Visual tracking and recognition using appearance-adaptive models in particle filters [J]. *IEEE Trans. on Image Process* (S1057-7149), 2004, 13(11):1491–1506.
3. Arulampalam M.S, Maskell S., Gordon N, et al. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking[J]. *IEEE Transactions on Signal* (S1053-587X), 2002, 50(2):174–188.
4. Fox D. Adapting the sample size in particle filters through KLD-sampling[J]. *The International Journal of Robotics Research*, 2003, 22(12):985–1003.
5. Pan P, Schonfeld D. Dynamic proposal variance and optimal particle allocation in particle filtering for video tracking[J]. *IEEE Transaction on Circuits and Systems for Video Technology*, 2008, 18(9):1268–1279.
6. Zheng Zuohu, Wang Shouyong, Wan Yang. Model of Maneuvering Target Track With Adaptive Parameter Adjustment[J]. *Journal of Air Force Radar Academy*, 2011, 25(2):113–116.
7. Zhao Xuemei, Chen Ken, Li Dong. Application of strong tracking Kalman filter in video target tracking[J]. *Computer Engineering and Applications*, 2011, 47(11):128–131.
8. Ba hongxin, He xinyi, Fang Zheng, Li Chunfang. A New Variance Adaptive Filtering Algorithm for Maneuvering Target Tracking[J]. *Journal of Wuhan University of Technology*, 2011, 35(3):448–452.
9. Zhang Kai, Shan Ganlin, Ji Bing, Chen Hai. Improved Variable Dimension Filtering for Attitude Tracking of Target[J]. *Electronics Optics & Control*, 2012, 19(3):40–43.
10. Li Tian-Cheng, Sun Shu-Dong. Double-resampling Based Monte Carlo Localization for Mobile Robot[J]. *Acta Automatica Sinica*, 2010, 36(9):1279–1286.