http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (1): 155-161, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.155.161

Research of Particle Filter Based on Immune Particle Swarm Optimization

Long-Hua Ma, Yu Zhang, Zhe-Ming Lu and Hui Li
School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, 310027, China

Abstract: Particle degradation, as a main limitation of particle filter, can be resolved by making use of common re-sampling method, but it always bring about the problem of sample dilution. The Immune Particle Swarm Optimization (IMPSO) was introduced into particle filter and a new kind of particle filter named IMPSO-based particle filter was proposed. In the IMPSO-based particle filter algorithm, particles are driven to the area with a higher posterior probability density and maintain big particle diversity at the same time. Simulation results show that IMPSO-based particle filter can eliminates the degeneracy phenomenon, avoid the sample dilution problem and guarantee the effectiveness.

Key words: Particle filter, immune particle swarm optimization, particle degradation, sample dilution

INTRODUCTION

Particle Filter (PF) is a nonlinear state estimation method using of a discrete random density of particles and their weights to approximate the relevant probability density distribution and updating the discrete random density according to the algorithm (Oppenheim *et al*., 2008). Compared with the usual nonlinear state estimation method, such as Extended Kalman Filtering and unscented Kalman Filter, the particle filter is applicable to any nonlinear and non-Gaussian situation, gets widespread attention and is widely used in target tracking, image processing, fault detection, etc. (Budhiraja *et al*., 2007).

Particle degradation is the most important negative aspect of PF, that is, after repeated recursive, except for very few particles with large weights, the weights of most particles are close to zero and become invalid particles. For such problem, domestic and foreign scholars have done a lot of research works. Gordon *et al*. (1993) introduced re-sampling link to the particle filter to avoid the particle degradation, but it leads to sample dilution problem which means after many times copy of power value particles, the number of different samples decreases and the diversity of particle set is destroyed. For sample dilution caused by the re-sampling process, domestic and foreign scholars have proposed a variety of improvement strategies. Ronghua and Bingrong (2004) proposed the use of genetic algorithms to improve the re-sampling link to solve the problem of particle degradation and sample dilution. Zhang *et al*. (2006) introduced the idea of particle swarm optimization to re-sampling, avoids the problem of particle degradation and sample dilution effectively, but affects the filtering effect since the particle swarm algorithm is easy to fall into the local minima.

The Immune Particle Swarm Optimization (IMPSO) was introduced into particle filter to optimize the re-sampling link of particle filter, avoiding particle degradation and improving the problem of sample dilution by immune particle swarm optimization in this paper. The simulation results show that the new particle filter algorithm, compared with other improved algorithms, improves the precision of the filter well and at the same time avoids the phenomenon of particle degradation and sample dilution.

The PF algorithm is a kind of technology which is based on Monte Carlo simulation to realize the recursive Bayesian filtering. Random samples in the state space were used to approximate the posterior probability density function and Monte Carlo estimation principle to estimate the state value. Specific processes of basic particle filter algorithm can be found in literature (Tan *et al*., 2002).

The particle filter algorithm takes sequential sampling algorithm as the basic framework. In order to avoid particle degradation, the re-sampling link proposed in literature (Gordon *et al*., 1993) is introduced. Algorithm completes state estimation process through sampling, update, weights calculation, normalization of weights and re-sampling link. The re-sampling link in literature (Gordon *et al*., 1993) is competed through a simple copy of power value particles with small weights particles removed which inevitably would damage the diversity of particles, cause phenomenon of sample dilution. Especially when likelihood function and prior distribution function have smaller overlapping area as

Corresponding Author: Hui Li, Teaching Building 5, Yuquan Campus, Zhejiang University, 38 ZheDa Road, Hangzhou 310027, Peoples' Republic of China Tel/Fax: +86-571-87953349

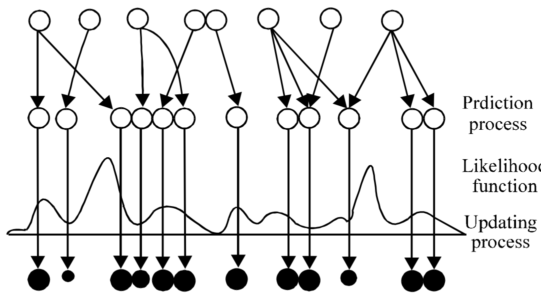

Fig. 1: The overlap region between likelihood function and prior density function is small

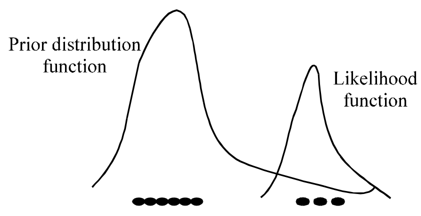

Fig. 2: The likelihood function is at the tail of prior distribution

shown in Fig. 1 or the likelihood function is located in the tail of prior distribution as shown in Fig. 2, only a small part of the particles' weights will increase after updating. After re-sampling link, particle set is composed of particles with only a few differences, inevitably producing sample dilution phenomenon.

PARTICLE SWARM OPTIMIZATION ALGORITHM

Generally speaking, Particle Swarm Optimization (PSO) is a kind of computational algorithm which could optimize a problem by trying to improve a candidate solution with regard to a given measure of quality iteratively. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space which are updated as better positions found by other particles. This is expected to move the swarm toward the best solutions.

PSO is a meta-heuristic algorithm as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered will then come to guide the movements of the swarm. The process is repeated and by doing so as it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

The choice of PSO parameters can have a large impact on optimization performance. Selecting PSO parameters that yield good performance has therefore, been the subject of much research. The PSO parameters can also be tuned by using another overlaying optimizer, a concept known as meta-optimization. Parameters have also been tuned for various optimization scenarios. The basic PSO is easily trapped into a local minimum. This premature convergence can be avoided by not using the entire swarm's best known position but just the best known position of a sub-swarm "around" the particle that is moved. Such a sub-swarm can be a geometrical one, for example "the most nearest particles" or, more often, a social one, i.e., a set of particles that is not depending on any distance. In such a case, the PSO variant is said to be locally best (vs. global best for the basic PSO). If we suppose there is an information link between each particle and its neighbours, the set of these links builds a graph, a communication network, that is called the topology of the PSO variant. A commonly used social topology is the ring, in which each particle has just two neighbours, but there are far more. The topology is not necessarily fixed and can be adaptive.

IMMUNE PARTICLE SWARM OPTIMIZATION ALGORITHM

Concentration adjustment mechanism of immune algorithm: Artificial immune algorithm is a stochastic global search algorithm for a simulation of the natural immune system function. Based on the concentration adjustment mechanism, it simulates the ability of the natural immune system to maintain immune balance.

Through the promotion and inhibition of antibody, immune response maintains a certain level. Antibody with large affinity with the antigen and lower concentrations will be promoted and antibody with small affinity with the antigen and higher concentrations will be suppressed. In the role of this mechanism, it maintains the diversity of antibodies in the immune system (Liu and Chen, 1998).

Immune particle swarm optimization (IMPSO) algorithm: IMPSO algorithm is a hybrid optimization algorithm based on particle swarm optimization algorithm, introducing the self-regulation mechanism based on concentration in immune algorithm. IMPSO combines the particle swarm optimization algorithm's characteristics of fast searching speed and simple realization with immune algorithm's concentration adjustment mechanism to overcome the shortcomings that particle swarm optimization algorithm in the optimization process cannot maintain the diversity of individuals and also cannot be adjusted balanceable between the diversity of individuals and the focus of groups. This improves the diversity of particles, avoids the shortcomings of the particle swarm algorithm easily being trapped into local minima and improves the accuracy of the algorithm.

Definition 1: The concentration of the ith particle (antibody) in immune particle swarm optimization is defined as:

$$C(X_i)=\frac{1}{\sum_{j}^{M+N}\left|\text{fitness}(X_i)-\text{fitness}(X_j)\right|} \tag{1}$$

where, i = 1, 2, 3...M+N, fitness is fitness calculation function.

According to definition 1, the concentration based probability of selection formula is:

$$P(X_i)=\frac{\frac{1}{C(X_i)}}{\sum_{i}^{M+N}\frac{1}{C(X_i)}} \tag{2}$$

It can be seen from Eq. 2, the more similar particles, the smaller probability the particle will be selected, thus avoiding the loss of particles with low concentration but in good condition. IMPSO algorithm uses the concentration-based selection method to ensure the diversity of the antibody theory.

IMPSO-BASED PARTICLE FILTER

In essence, both the particle filter algorithm and immune particle swarm optimization algorithm use random search of particle collection to find the optimal solution in the space, so they have a certain similarity. Immune particle swarm optimization based particle filter (IMPSO-based PF) proposed in this paper is established on the basis of this similarity. The basic idea of the IMPSO-based PF algorithm is taking of the particles as input particles of an IMPSO algorithm to estimate the state value. IMPSO makes particle collection move to the area with large posterior probability density and uses immune concentrations regulation role to ensure the particles' diversity. It fundamentally changes the particle filter algorithm which just simply copies the particles with power value and discards the particles with small weights while re-sampling, avoiding particle degradation while improving sample dilution. Consider the following nonlinear system:

$$\begin{cases}X_k=f(X_{k-1},V_{k-1})\\Y_k=h(X_k,N_k)\end{cases} \tag{3}$$

where, X_k, Y_k separately denote state vector and measurement vector at time k. V_k, N_k separately denote system noise and observation noise. Processes of IMPSO-based PF algorithm are as follows:

Step 1: Select N random particles from f(x, v) and they are recorded as the initial particle collection $P = [X_k, 1/N]$, make k = 1

Step 2: Use system equations to update the particle collection:

$$X^i_k = f(X^i_{k-1}, V_{k-1}) \tag{4}$$

In the equation, i = 1, 2, ..., N, the particle collection at time k+1 is $P1 = [X_k, 1/N]$

Step 3: Calculate weights of each particle and then normalize, get the particle collection at time k+1 is $P2 = [X'_k, w^i_k]$ and take particle collection as the input particles of an IMPSO algorithm. The fitness value of each particle is calculated in accordance with the definition 2, respectively

Weight:

$$w_k{}^i = w_{k-1}{}^i * \frac{P(y_k\,|\,x_k{}^i)*P(x_k{}^i\,|\,x_{k-1}{}^i)}{q(x_k{}^i\,|\,x_{k-1}{}^i, y_k)} \tag{5}$$

Weights normalized:

$$w_k{}^i = w_k{}^i / \sum_{i=1}^{N} w_k{}^i \tag{6}$$

Definition 2: Immune particle swarm optimization fitness function of the particle filter algorithm is defined as:

$$\text{fitness}(X_k^i) = \exp\left(-\frac{1}{2R_k}\left(Y_{new}(X_k^i) - Y_{pred}(X_k^i)\right)^2\right) \quad (7)$$

where, R_k denotes observation noise variance, X_{new} (X^i_k) and Y_{pred} (X^i_k) denote the newest particle observations and predicted observations of the ith particle at time k, respectively.

Step 4: Update each particle's velocity, position and fitness value in particle collection according to immune particle swarm optimization algorithm and save this Gbest as immunological memory factor into memory vault:

$$\begin{cases} V(X_k^i) = W \times V(X_{k-1}^i) + \\ C_1 \times \text{Rand} \times (\text{Pbest} - X_k^i) + \\ C_2 \times \text{Rand} \times (\text{Gbest} - X_k^i) \\ X_k^i = X_{k-1}^i + V(X_k^i) \end{cases} \quad (8)$$

where, W is the inertia coefficient and C_1, C_2 are the learning factors

Step 5: Randomly generate new particles which can meet the requirements and constitute a new set of particles together with the collection of particles generated in step 4 P3 = [X^i_k], i = 1, 2, ..., M+N. Calculate the fitness value of each particle according to definition 5

Step 6: Calculate each particle's concentration and selection probability according to Eq. 1 and 2 based on their fitness values. Arrange them in decreasing order and choose the former N particles to constitute a particle collection P4 = [X^i_k]. Check whether each particle's fitness value meets the threshold requirements, respectively and use immunological memory factor Gbest produced by step 4 to replace particles which does not meet the condition

Step 7: Repeat step 2-6 until the iterative requirements are satisfied

Step 8: Take P4 = [X^i_k] as the output of immune particle swarm optimization re-sampling, update the corresponding weights of each particle according to the Formula 5 and 6, calculate state estimates at time k:

$$X_k \approx \sum_{i=1}^{N} w_k^i \times X_k^i \quad (9)$$

Step 9: Make k = k+1 and return to step 1 until k meets the stop conditions

SIMULATION TESTS AND RESULTS ANALYSIS

For the proposed algorithm, two typical nonlinear models are used to validate its nonlinear state estimation capability, at the same time to calculate and analyze the valid samples and Root Mean Square Error (RMSE). In comparison with other algorithms, IMPSO-based PF algorithm has better filtering accuracy and can avoid particle degradation effectively while improving sample dilution phenomenon.

Nonlinear system state estimation performance analysis and verification:

Case 1: Use the following nonlinear system to verify the state estimation ability of the new algorithm in nonlinear state. The equations of nonlinear system are:

$$\begin{cases} X(k-1) = \frac{1}{2} \times X(k-1) + \frac{25 \times X(k-1)}{1 + X^2(k-1)} + 8 \times \cos(1.2 \times k) + W(k-1) \\ Y(k) = \frac{X^2(k)}{20} + V(k) \end{cases} \quad (10)$$

where, W(k), V(k) denote Gaussian noises with zero mean and 10 and 1 variance, respectively. The conventional Particle Filter (PF), Genetic Particle Filter (GPF), Particle Swarm Optimization Particle Filter (PSOPF) and IMPSO-based PF were used to carry on the simulation, respectively. The particle number N is 150, the simulation time is 80 sec and time interval is 1 sec. For convenient comparison, the simulation curves selected during the time period of t = 40~70 are shown in Fig. 3.

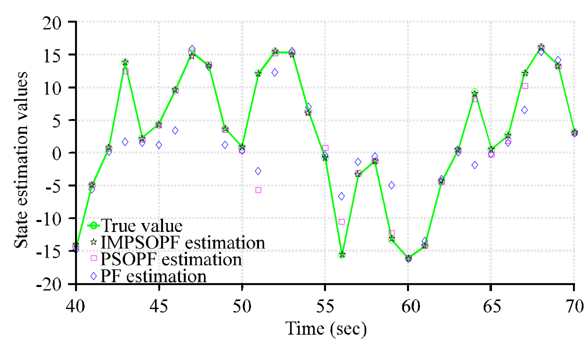

Fig. 3: The comparison of particle filter (PF), particle swarm optimization particle filter (PSOPF) and immune particle swarm optimization based particle filter (IMPSOPF) algorithms

It can be seen from Fig. 3, the estimated value of immune particle swarm optimization particle filter fits a high degree with the true value, showing that the algorithm is effective. Figure 4 shows IMPSO-based PF algorithm state estimates.

Case 2: In order to inspect the estimated capacity of the algorithm to the mutation signal. The following piecewise function was used to test its state estimation capacity under the conditions of the mutant signal. Select piecewise function of the system equations as follows:

$$\begin{cases} X(t) = \begin{cases} 5, t < T/3 \\ 10, T/3 < t < 2T/3 \\ 3, t > 2T/3 \end{cases} \\ Y(t) = X(t) + N(t) \end{cases} \quad (11)$$

In the equation, N(t) denotes Gaussian noise with mean 0 and variance 5. Take the particle number as 40, the simulation time as 80 sec and time interval as 1 sec. Curves of the conventional particle filter and the immune particle swarm particle filter are shown in Fig. 5.

It can be seen from Fig. 5 that the new algorithm has a better estimation effect when estimating a long time invariant. In order to compare these algorithms, RMSE is utilized as the standard measure of estimation accuracy. Monte Carlo simulations of the four algorithms are carried on 10 times for case one and case two, respectively. The average of RMSE are listed into Table 1. It can be seen from Table 1, the IMPSO-based PF algorithm has the smallest RMSE of the mean and is able to improve the estimation accuracy greatly.

Improvement of the particle filter particle degradation and sample dilution: In order to compare the improvement effects of various improved algorithms for particle filter particle degradation, the concept of the effective sample size is introduced.

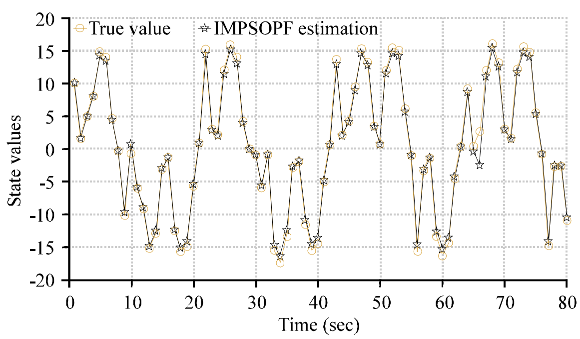

Fig. 4: Result of immune particle swarm optimization based particle filter (IMPSOPF) algorithm state estimates

Definition 3: The effective sample size is defined as follows:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (w(X_k^i))^2} \quad (12)$$

In the Equation, $w(X_k^i)$ indicates the weight of the ith particle at time k after re-sampling. The fewer effective samples, the more heavily it denotes particles degradation; otherwise, it denotes less extent of particles degradation. According to the test functions above, Monte Carlo simulations are carried on 10 times for case one and case two, respectively in the four algorithms. The average results are included in Table 2 and 3, respectively. In Table 2, it is shown that the numbers of effective samples are of the maximum value in average which means that the IMPSO-based PF algorithm is more robust on resisting degradation phenomenon, i.e., sample dilution problem, in nonlinear system as described in case one.

Table 1: Comparison of four algorithms in root mean square error

	Particle filter	Genetic particle filter	Particle swarm optimization particle filter	Immune particle swarm optimization particle filter
Case 1	4.452	4.1370	3.9960	3.3980
Case 2	1.025	0.4250	0.3764	0.3559

Table 2: Comparison of the effective sample size of four algorithms in case one

Num	Particle filter	Genetic particle filter	Particle swarm optimization particle filter	Immune particle swarm optimization particle filter
1	48	100	116	125
2	49	97	114	117
3	44	99	117	121
4	38	94	119	124
5	39	102	116	125
6	46	104	112	123
7	44	95	118	119
8	43	101	115	121
9	41	93	117	122
10	51	105	113	124
Average	44.3	99	115.7	122.1

Table 3: Comparison of the effective sample size of four algorithms in case two

Num	Particle filter	Genetic particle filter	Particle swarm optimization particle filter	Immune particle swarm optimization particle filter
1	11	30	34	36
2	9	28	35	35
3	9	31	32	39
4	10	31	32	39
5	11	33	35	38
6	11	30	30	37
7	8	31	31	40
8	8	32	33	35
9	12	29	31	39
10	9	32	36	40
Average	9.8	30.7	32.9	37.8

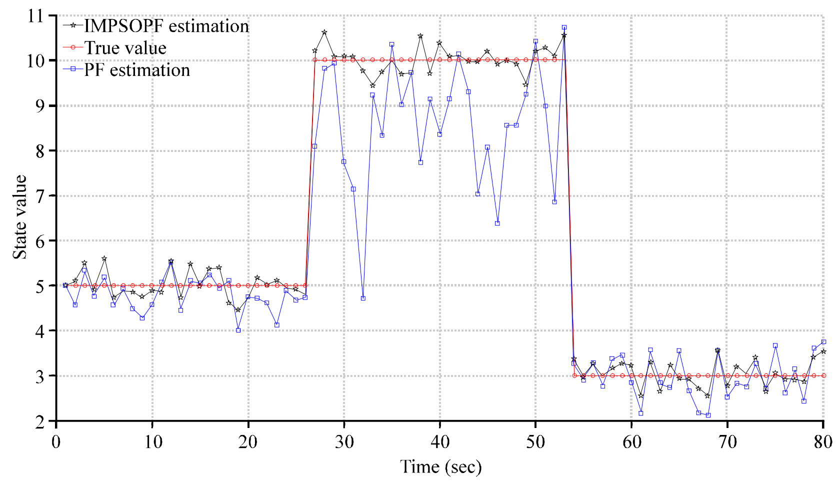

Fig. 5: The comparison of particle filter (PF) and immune particle swarm optimization based particle (IMPSOPF) filter algorithms

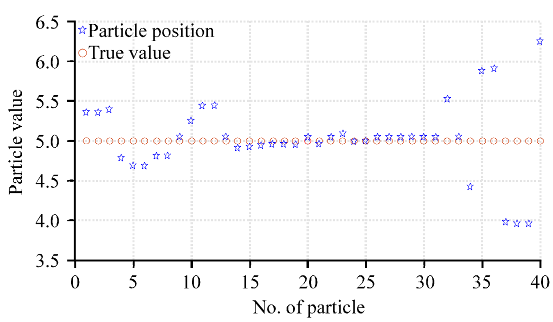

Fig. 6: Particle distribution map of particle filter at t = 15

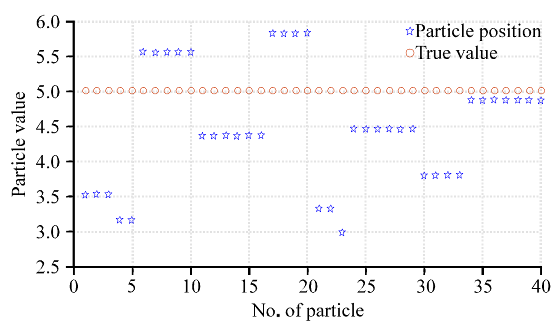

Fig. 7: Particle distribution map of immune particle swarm optimization particle filter at t = 15

Likewise, from the comparisons of the number of valid samples in Table 3, IMPSO-based PF algorithms also shows the improvement on the degradation phenomenon when facing the mutation signal in case two. In order to illustrate the improvement of the new algorithm for the problems of sample dilution, the distributions of particle collection for PF and IMPSO-based PF algorithm were given at time t = 15, as shown in Fig. 6 and 7.

Figure 6 reflects the sample points of PF algorithm. PF have degrade from initial 40 particles to 10 particles at t = 15 which shows that the sample dilution of PF is very serious.

In Fig. 7, IMPSO-based PF keeps about 35 different sample points at t = 15 and has a distinct advantage compared with other algorithms. IMPSO-based PF can avoid particle degradation and at the same time, alleviate the problem of sample dilution.

CONCLUSIONS AND FUTURE WORKS

Since the particle filter is prone to appear particle degradation and sample dilution, this paper proposed an idea of using immune particle swarm optimization to improve the link re-sampling and in essence to avoid the problem of particle degradation and sample dilution. The simulation results show that compared with other commonly used improved algorithms, the immune particle swarm optimization particle filtering algorithm can improve the precision of the filter, at the same time effectively prevent particle degradation and sample dilution. The proposed algorithm is feasible and effective.

IMPSO-based PF algorithm can be used in all domain of PF algorithm, such as target tracking and joint positioning with senses of hearing and seeing. And also it can be used in machine vision, navigation, image processing, bio-information index, failure diagnosis, process control and financial data processing, etc.

Conditional density propagation based PF algorithm is also an effective way to tackle the problem of particle degradation which can reduce the dimension of non-rigid state vector object such as rotation and the degree of tilt by verifying likelihood of the parameters in each particle each. We plan to compare this algorithm with the IMPSO-based PF in the next manuscript.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the financial support from the National Nature Science Foundation of China under grant No. 61070003 and 61071128; Zhejiang Provincial Natural Science Foundation of China under grant No. R1090052 and Y1101184; Project of Zhejiang Department of Education under the grant No. N20100690.

REFERENCES

Budhiraja, A., L. Chen and C. Lee, 2007. A survey of numerical methods for nonlinear filtering problems. Physica D Nonlinear Phenom., 230: 27-36.

Gordon N.J., D.J. Salmond and A.F.M. Smith, 1993. Novel approach to nonlinear and non-Gaussian Bayesian state estimation. Proceedings of the IEEE on Radar and Signal Processing, Volume 140, April, 1993, UK., pp: 107-113.

Liu, J.S. and R. Chen, 1998. Sequential Monte Carlo methods for dynamic systems. J. Am. Statistical Assoc., 93: 1032-1044.

Oppenheim, G., A. Philippea and J. de-Rigal, 2008. The particle filters and their applications. Chemometrics Intelligent Lab. Syst., 91: 87-93.

Ronghua, L. and H. Bingrong, 2004. Coevolution based adaptive Monte Carlo localization (CEAMCL). Int. J. Adv. Rob. Syst., 1: 183-190.

Tan, Y.Z., J. Shen and Z.Z. Lu, 2002. Study of immunity optimization algorithm and its future development. Inform. Control, 31: 385-389.

Zhang, W.M., G. Du, S. Hong and Y.H. Zhang, 2006. Study of nonlinear filter methods: Particle filter. J. Syst. Eng. Electron., 7: 1-5.