Applied Mathematics & Information Sciences
An International Journal

New clone particle swarm optimization-based particle filter algorithm and its application

Yuming Bo[1], *Zhimin Chen*[1], *Jie Zhang*[1], *Jianliang Zhu*[1]

[1]School of Automation, Nanjing University of Science and Technology, Nanjing, 210094, china

Received: Jun 7, 2012; Revised Jul. 15, 2012; Accepted Jul. 22, 2012
Published online: 1 January 2013

Abstract: A new particle filter algorithm based on new Clone particle swarm optimizationNCPSO-PF is presented in this paper in order to solve the problem of low precision and complicated calculation of particle filter based on particle swarm optimization algorithm(PSO-PF). The algorithm enables the particles to fit the environment better and then reach the goal of global optimization through orthogonal initialization, clonal selection and local searching of self-learning. Finally different models are used for simulation experiment and the results indicate that this new algorithm improves the operation speed and precision of practical engineering application.

Keywords: particle filter; particle swarm optimization; Clone; orthogonal

1. Introduction

Particle filtering is a sequential Monte Carlo methodology where the basic idea is the recursive computation of relevant probability distributions using the concepts of importance sampling and approximation of probability distributions with discrete random measures[1]. It is widely applied to positioning and navigation of non-Gaussian noise and non-linear system, fault diagnosis[2], target tracking and mode identification field [3] as its state function and observation function has no non-linear hypothesis. Nevertheless, PF may confront with the problem of weight degradation [4] which if solved by resampling method may result unavoidable particle impoverishment[5-6]. As a result, the filtering precision is influenced.

PF based on Intelligent Optimization Algorithms conduces to significant improvement of particle degradation in PF and great enhancement of precision [7]. Particle swarm optimization-based PF (PSO-PF) is a typical representative of Intelligent Optimization Algorithm which introduces PSO algorithm into PF. Through introduction of the latest measurement value to the sample distribution, along with the utilization of PF algorithm for sampling process optimization and constantly update of particle speed, the sample distribution is inclined to move to the area with higher posterior probability [8]. PSO-PF improves the particle degradation of PF and is easier for actualization. Unfortunately PSO-PF is a process of iterative optimization which will prolong the calculation time because of the high iterative frequency[9]. Moreover, PSO-PF may be easily trapped into local optimization, influencing the precision and stability of practical engineering application [10].

A new particle filter algorithm based on new Clone particle swarm optimizationNCPSO-PFis suggested in this paperThis algorithm utilizes orthogonal strategy and clone particle swarm algorithm for optimization, according enhancing the quality and diversity of particles. The experimental results prove that NCPSO-PF improves the efficiency of particle filter.

2. PSO-PF algorithm

2.1. Basic PSO algorithm

PSO is an intelligence optimization algorithm imitating birds' clustering movement. It is widely applied to target tracking, positioning and navigation, mode identification etc. by virtue of its advantages of simple concept, ease in actualization, fewer parameters, and effectiveness in solving complicated optimization and so on. PSO algorithm can be expressed as follows[11]: to randomly initialize a

* Corresponding author: e-mail: byming@mail.njust.edu.cn

particle swarm whose number is m and dimension is n, in which the ith particle's position is$X_i = (x_{i1}, x_{i2}, \cdots x_{in})$ and its speed is$V_i = (v_{i1}, v_{i2}, \cdots v_{in})$. During each iteration, the particles can renew their own speed and position through partial extremum and global extremum so as to reach optimization. The update formula is:

$$V_i = w \times V_i + c_1 \times Rand() \times (P_i - X_i)$$
$$+ c_2 \times Rand() \times (G - X_i) \quad (1)$$
$$X_{i+1} = X_i + V_i \quad (2)$$

where $Rand$is a random number within interval (0,1), $w$is the inertia coefficient(A larger inertia coefficient means stronger global search ability, while a smaller one stands for stronger local search ability), c_1 and c_2 are learning factors.

2.2. Description of standard PSO-PF algorithm

The importance sampling process of conventional PF is suboptimal, whereas the incorporation of PSO algorithm will optimize the sampling process of PF, allow the weight of particle sets are more inclined to high likelihood region, accordingly solving the problem of particle impoverishment, and conducing to reduction of particle numbers required by PF. PSO method is fused with PF and the key lies in utilizing the optimal state value P_{pbest} experienced by the particles and the state value P_{gbest} of the maximum particle with the greatest objective function value, and updating the speed and position of each particle on a real-time base through equation (5) and (6), accordingly forcing the particles to be closer to the real state.

$$V_k^i = |Randn| \times (P_{\text{pbest}} - X_{k-1}^i)$$
$$+ |randn| \times (P_{\text{gbest}} - X_{k-1}^i) \quad (3)$$
$$X_k^i = X_{k-1}^i + V_{k-1}^i \quad (4)$$

Where $|Randn|$ and $|randn|$ are positive Gaussian distribution random numbers[12].

3. principle of NCPSO-PF

Based on the enlightenments brought by biological immune system and clone mechanism, an orthogonal immune clone partial swarm algorithm is presented here for function solution optimization. First of all, let $A(k) = \{a_1(k), a_2(k), \cdots, a_{size}(k)\}$ denotes the $k$the generation particle population, S_{size} denotes the scale of particle swarm. Each particle may actualize particle swarm update through clone selection and self-learning. Corresponding operators are given as follows:

(1) Immune clone operator For any antibody $a_i = (a_{i,1}, a_{i,2}, \cdots, a_{i,n})$ in particle swarm $A(k)$ of kth generation, $i = 1, 2, \cdots, S_{size}$ performs clone operation with scale q_i, becoming clone q_i of antibody a_i. The q_i clone of antibody a_i is defined as $\overline{a}^i = I_i \times a_i$, wherein $\overline{a}^i$ is the antibody group of $\overline{a}^i$ after cloning, and I_i is the q_i-dimensional vector with element 1.

$$q_i(k) = Int\left[n_c \times \frac{f(a_i)}{\sum_{j=1}^{S_{size}} f(a_j)} \varUpsilon\right], i = 1, 2, \cdots, S_{size} \quad (5)$$

where n_c denotes population clone scale, $n_c > S_{size}$, $Int()$ means rounding up. $f(a_i)$ is the affinity of antibody a_i, while $\varUpsilon$ reflects the affinity of antibody $a_i$with other particles and is defined as:

$$\varUpsilon = \min\{D_{ij}\} = \min\{\exp(\|a_i - a_j\|)\}, \quad (6)$$
$$i \neq j; i, j = 1, 2, \cdots, S_{size}$$

in which $\|x\|$ denotes any of the norms, and in this paper Euclidean distance is employed. D is a symmetric matrix reflecting the diversity of population.

(2) Immune gene operation. Immune gene operation mainly consists of genetic recombination and genetic mutation. In order to facilitate the communication of useful information among individuals and to enhance the uniformity of progeny individuals, orthogonal recombination algorithm is employed here to recombine the cloned individuals by probability.

(1) Genetic recombination. Given the genetic recombination operatorP^c, and $C(k) = \{c^1(k), c^2(k), \cdots, c^{S_{SIZE}}(k)\}$ as the population after recombination.

Taking two parental particles $b_1 = (b_{1,1}, b_{1,2}, \cdots, b_{1,n})$ and $b_2 = (b_{2,1}, b_{2,2}, \cdots, b_{2,n})$ into consideration, a sub-space $[l, u]$ is determined in accordance with the following equation:

$$\begin{cases} l = [\min(b_{1,1}, b_{2,1}), \min(b_{1,2}, b_{2,2}), \cdots, \min(b_{1,n}, b_{2,n})] \\ u = [\max(b_{1,1}, b_{2,1}), \max(b_{1,2}, b_{2,2}), \cdots, \max(b_{1,n}, b_{2,n})] \end{cases} \quad (7)$$

First of all each of the region in Space $[l, u]$ is quantified into Q levels, and the difference between two neighboring levels are always the same. $\beta_i = (\beta_{i,1}, \beta_{i,2}, \cdots, \beta_{i,Q})$ is defined as follows:

$$\begin{cases} \min(l_i, u_i), & j = 1 \\ \min(l_i, u_i) + (j-1) \times (\frac{|l_i - u_i|}{Q-1}), & 2 \leqslant j \leqslant Q-1 \\ \max(l_i, u_i), & j = Q \end{cases} \quad (8)$$

$F - 1$ integer will be generated, $k_1, k_2, \cdots, k_{F-1}$, and $1 < k_1 < k_2 < \cdots < k_{F-1} < n$ is met. Then for any particles$x = (x_1, x_2, \cdots, x_n)$, the following F factors will be generated.

$$f_1 = (x_1, \cdots, x_{k1}), f_2 = (x_{k1+1}, \cdots, x_{k2}), \cdots,$$
$$f_F = (x_{F-1+1}, \cdots, x_n) \quad (9)$$

Q levels of factor f_i are

$$\begin{cases} f_i(1)=(\beta_{k_{i-1}+1,1},\beta_{k_{i-1}+2,1},\cdots,\beta_{k_i,1}) \\ f_i(2)=(\beta_{k_{i-1}+1,2},\beta_{k_{i-1}+2,2},\cdots,\beta_{k_i,2}) \\ \cdots \\ f_i(Q)=(\beta_{k_{i-1}+1,Q},\beta_{k_{i-1}+2,Q},\cdots,\beta_{k_i,Q}) \end{cases} \tag{10}$$

The orthogonal matrix $L_M(Q^F)=[r_{i,j}]M_2\times F$ is used to generate the following M particles:

$$\begin{cases} (f_1(r_{1,1}),f_2(r_{1,2}),\cdots,f_F(r_{1,F}),) \\ (f_1(r_{2,1}),f_2(r_{2,2}),\cdots,f_F(r_{2,F}),) \\ \cdots \\ (f_1(r_{M,1}),f_2(r_{M,2}),\cdots,f_F(r_{M,F}),) \end{cases} \tag{11}$$

Finally, an optimal particle is selected from M particles as the clone progeny subject. Given $Q=3, F=4, M=9$, and orthogonal matrix is marked as $L_9(3^4)$.

(2) Genetic mutation. AEA mutation strategy is used, letting $D(k)=\{d^1(k),d^2(k),\cdots,d^{S_{SIZE}}(k)\}$ the population after mutation, that is,

$$d_j^i=c_j^i-\frac{fix(c_j^i\times 10^l(\bmod 10-md(10))}{10^l} \tag{12}$$

$$i=1,2,\cdots,S_{SIZE}; j=1,2,\cdots,q$$

where t is the random number of $0\sim 15$, $fix()$ is rounded down, $rnd(10)$ a random integer less than 10, whereas the probability of mutation is usually very small.

(3) Clone selection operator. Different from the selection operation in evolutionary operation, clone selection operator is to select the excellent individual from particle clone and to form new populations. Given $\forall i=1,2,\cdots,S_{SIZE}$, then:

$$d_i'(k)=\{d_j^i(k)|\min f(d_j^i), j=1,2,\cdots,q\} \tag{13}$$

For probability $p^k(d_i'(k)\cup a_i(k)\rightarrow a_i(k+1))$,

$$p^k(a_i(k+1)=d_i'(k))=$$
$$\begin{cases} 1, & f(a_i(k))>f(d_i'(k)) \\ \exp(-\frac{f(a_i(k))-f(d_i'(k))}{a}) & f(a_i(k))\leqslant f(d_i'(k))\text{and} \\ & a_i(k)\text{is not the best antibody.} \\ 0 & f(a_i(k))\leqslant f(d_i'(k))\text{and} \\ & a_i(k)\text{is the best antibody.} \end{cases} \tag{14}$$

(4) Self-learning operator. During the clone operation of particle swarm, it searches in a relatively large range, to avoid the loss of optimal solution, self-learning local searching of optimal individuals after clone selection is performed in this paper. For the optimal individual$\hat{a}(k)=(\hat{a}_1(k),\hat{a}_2(k),\cdots,\hat{a}_n(k))$, first of all a self-learning population$L$ is generated with size L_{SIZE}, then L:

$$L=\begin{cases} \hat{a}(k), & i=1 \\ New_i, & \text{others} \end{cases} \tag{15}$$

wherein $New_i=(e_{i,1},e_{i,2},\cdots,e_{i,n})$ is generated according to the following equation:

$$e_{i,k}=\hat{a}_{i,k}\cdot U(1-sradius,1+stradius) \tag{16}$$

$k=1,2,\cdots,n$, in which $sradius\in[0,1]$ denotes searching semi-diameter, $U(1-sradius,1+sradius)$ denotes the random number in interval$[1-sradius,1+sradius]$. Given $Min_i=(m_1,m_2,\cdots,m_n)$ the individual with optimal fitness in$L$, e.g. $Min_i\in L$, and for any individual $L_i\in L$ in a small population, we have $f(L_i)\geqslant f(Min_i)$, if L_i meets $f(L_i)\leqslant f(Min_i)$, then it is a winner and will be preserved in the self-learning population, or otherwise a loser that should die. The eliminated individuals will be occupied by new individuals $New_i'=(e_{i,1}',e_{i,2}',\cdots,e_{i,n}',)$generated by Min_i, wherein P_0 is the probability given.

$$e_{i,k}'=$$
$$\begin{cases} m_k+d\times(m_k-L_{i,k}),k=1,2,\cdots,n. & ifU(0,1)<P_0 \\ (m_1,m_2,\cdots,m_{i_1-1},m_{i_2},m_{i_2-1},\cdots, & else \\ m_{i_1+1},m_{i_1},m_{i_2+2},\cdots,m_n) & \end{cases} \tag{17}$$

where d is the random variable within range $[0,1]$. Finally, Gaussian mutation of the individuals in self-learning population is carried out, and the learned individual is replaced by optimal individual.

4. Process of NCPSO-PF

(1) Take$N$particles $\{x_{0:k}^i, i=1,...,N\}$as samples from importance function at the initial time. The importance density function is expressed in equation(18)

$$x_k^i\sim q(x_k^i|x_{k-1}^i,z_k)=p(x_k^i|x_{k-1}^i) \tag{18}$$

Giving the fitness function:

$$Y=\exp[-\frac{1}{2R_k}(z_{New}-z_{\mathrm{Pred}})] \tag{19}$$

Where z_{New} is the latest observed value, $z_{\mathrm{Pred}}$is the predictive observed value.

(2) calculate the importance value:

$$w_k^i=w_{k-1}^i p(z_k|x_{k-1}^i)=w_{k-1}^i p(z_k|x_k^i) \tag{20}$$

(3) Updating the speed of particle swarm$A(k)$, then we have $A'(k)$.

(4) Replace $T\%$ relatively undesirable individuals with $T\%$ desirable ones in$A'(k)$, and $A(k)\leftarrow A'(k)$.

(5) Perform clone operation against particle swarm$A(k)$, including clone recombination and clone mutation, then $A(k)$.

(6) Perform immune gene operation against particle swarm$A(k)$, including clone recombination and clone mutation, then D.

(7) Perform clone selection against$D$, then $A^*(k)$.

(8) Force the self-learning operator to take effect on the optimal individual of $A^*(k)$ and then to upgrade the optimal individual.

(9) Assuming x_e is the individual extreme value of the current particle, pb_t^i the optimal solution at t time of particle i. $gbest_k$ is the global optimal solution at t time .Compare their fitnessupdate pbandpg:

$$pb_k^i = \begin{cases} pb_k^i, Y_g) < Y(pb_k^i) \\ x_g, Y_g) > Y(pb_k^i) \end{cases} \tag{21}$$

$$pg_k \in \{x_k^1, x_k^2, x_k^3, \cdots x_k^N | Y(x)\} \\ = \max\{Y(x_k^1), Y(x_k^2), Y(x_k^3), \cdots Y(x_k^N)\} \tag{22}$$

(10) When the optimal value of particle complies with the initially-set threshold valueεor algorithm reached maximum iteration times λ, it is indicated that the particles have been already distributed around the true values. By now particle optimization should be stopped. Else jump to step(3).

(11) Calculate the importance weight of the particles after optimization and perform normalization:

$$w_k^i = w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tag{23}$$

$$w_k^i = w_k^i / \sum_{i=1}^{N} w_k^i \tag{24}$$

(12) State output:

$$\widetilde{x} = \sum_{i=1}^{N} w_k^i x_k^i \tag{25}$$

5. Simulation experiment

5.1. Univariate nonstationary growth model

Choosing a univariate nonstationary growth model (UNGM), and the process model and measurement model of the simulated objects are given as follows:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1+[x(t-1)]^2} + 8\cos[1.2(t-1)] + w(t) \tag{26}$$

$$z(t) = \frac{x(t)^2}{20} + v(t) \tag{27}$$

In which, $w(t)$ and $v(t)$ are zero-mean Gaussian noise. Since this system is highly non-linear and the likelihood function presents bimodal, it will be difficult for traditional filtering methods to deal with this system. By using PFPSO-PFNCPSO-PF, state estimation and tracking of this nonlinear system are performed, and the formula of root-mean-square error is

$$RMSE = [\frac{1}{T}\sum_{t=1}^{T}(x_t - \widehat{x}_t)^2]^{1/2} \tag{28}$$

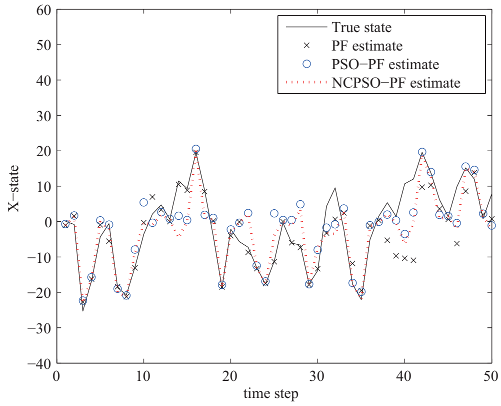


Figure 1 state estimation $(Q = 10)$

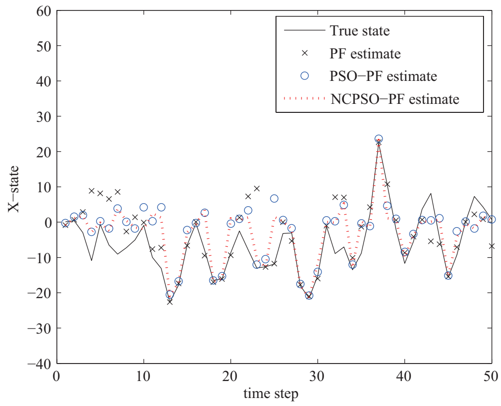


Figure 2 state estimation $(Q = 20)$

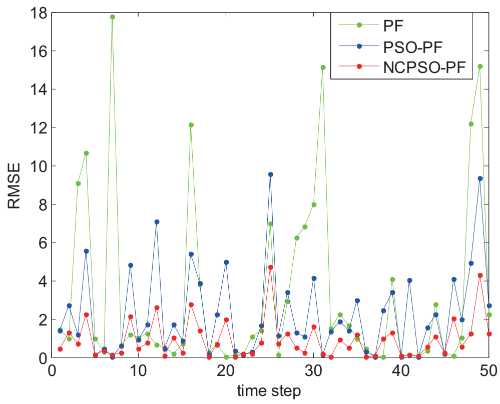


Figure 3 RMSE of different algorithm $(Q = 10)$

Giving the particle number $N = 100$, and process noise variance $Q = 10$ or 20, measurement noise variance $R = 1$, initially-set threshold value is ε and maximum iteration times is λ, simulation is conducted. After single-step simulation by using PF, PSO-PF and NCPSO-PF, the simulation result is presented in figure 5.1. . . 5.4. After 500 times of Monte-Carlo simulation, result is given in Table 1.

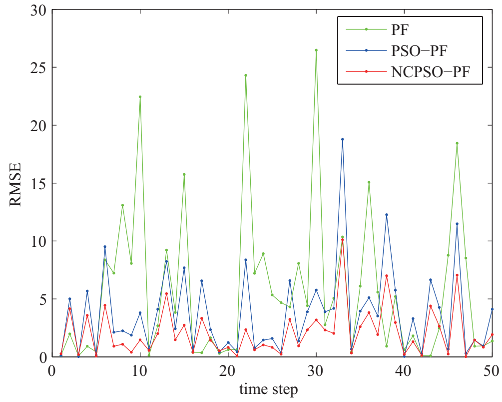


Figure 4 RMSE of different algorithm $(Q = 20)$

Table 1 comparison of simulation parameters by UNGM model

Parameters	Algorithm	RMSE	Time
	PF	2.8652	0.3528
N=100,Q=10,R=1	PSO- PF	1.7063	0.5281
	NCPSO-PF	1.4682	0.4298
	PF	5.7628	0.3401
N=100,Q=20,R=1	PSO- PF	2.8860	0.5328
	NCPSO-PF	2.5645	0.4582

From the experimental result, regardless in precision or speed, NCPSO-PF has been improved by contrast with PF and PSO-PF. This can be ascribed to the fact that the orthogonal cross strategy can reinforce the uniformity of progeny individuals, avoiding loss of optimal solution in the individual neighborhood. At the same time the precocity of particles can be eliminated, and thereby the speed and precision of the algorithm is enhanced. In the above calculation, with respect to both precision and time, NCPSO-PF has the most desirable performance.

5.2. simulation model of white noise deconvolution filter

In practical engineering, the estimation of optimal output of deconvolution white noise is a common-seen problem. In this paper, PF, PSO-PF and NCPSO-PF are used to forecast the white noise of a randomly-selected system that is described as follows:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0.3 & 0.5 \end{bmatrix} x(t) + \begin{bmatrix} -1 \\ 2 \end{bmatrix} w(t) \qquad (29)$$

$$z(t) = [11]x(t) + v(t) \qquad (30)$$

in which $w(t) = b(t)g(t)$ is Gaussian white noise. $b(t)$ is the white noises with values 1 and 2. The probability for value acquisition is:

$$P(b(t)) = \begin{cases} \lambda, b(t) = 1 \\ 1-\lambda, b(t) = 0 \end{cases} \qquad (31)$$

λ is the probability of $w(t)$ with a non-zero value. $g(t)$

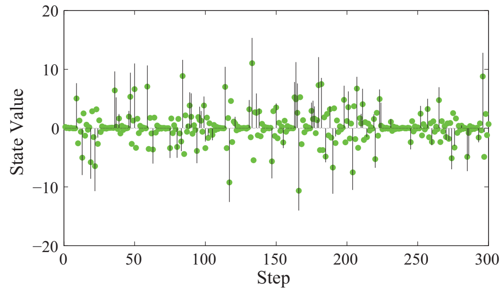


Figure 5 PF estimate

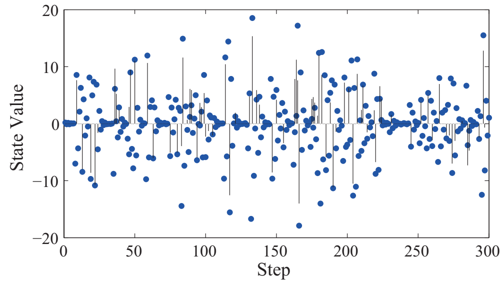


Figure 6 PSO-PF estimate

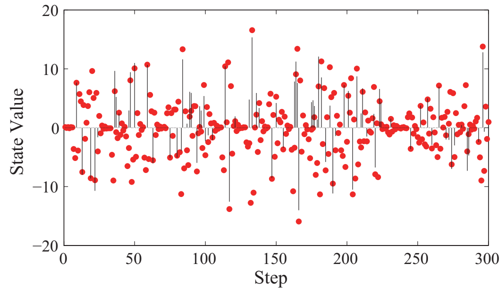


Figure 7 NCPSO-PF estimate

is the Gaussian white noise independent of $b(t)$ with mean value 0 and variance σ_g^2. In the simulation, $\sigma_v^2 = 0.1$, $\sigma_g^2 = 0.1$, and the simulation results are shown as following, in which the vertical axis of the solid line end point, and the vertical axis of round dot represents the estimated value.

Making a comparison between the simulation results shown in the figure, the estimation of white noise by PF displays the lowest precision, and it fails to well predict the value of most of the state points, while the precision of two particle filters by PSO is improved significantly. As for NCPSO-PF, since it successfully preserves the diversity of population, it has the highest precision and can well estimate the values of white noise.

6. Conclusion

This paper presents a new particle filter algorithm based on clone particle swarm which can enhance the convergence speed and grantee the population diversity while enlarge the local searching range, thereby improving the quality of the particles. The experiment results indicate that the algorithm in this paper conduces to enhancement of particle filter precision and speed as well as good robust, thus it is of great application value in practical engineering.

Acknowledgements

This paper was supported by the National Nature Science Foundation of China (No. 61104196), the National defense pre-research fund of China (No. 40405020201), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20113219110027), the Special Plan of Independent research of Nanjing University of Science and Technology of China (No. 2010ZYTS051).

References

[1] Sunny Chauhan and Neeraj Dhiman, Common Theorems on Fuzzy Metric Spaces Relation, Math. Sci. Lett. 1 (2012) 89-96.

[2] Wang Xue-guang, Chen Shu-hong, An Segmentation Algorithm Based on Two- Method, Inf. Sci. Lett. 1 (2012) 77-83.

[3] F. S. JIAN, Y. M. XU, Z. J. YIN. Enhanced multiple model particle filter for maneuvering target tracking. *Control Theory & Applications,* (2010) 1012-1016.

[4] A. Doucet, S. Godsill. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Compuring,* (2000) 197-208.

[5] A. Kong, J. Liu. sequential imputations and Bayesian missing data problems. *Journal American Statistical Association,* (1994) 278-288.

[6] Chen Shu, Wu Guo Qing and Ye Jun Min, A Domain Oriented Approach for Network Software Requirement Modeling, Inf. Sci. Lett. **1** (2012) 21-39.

[7] Zhao,Jing, Li,Zhiyuan. Particle filter based on Particle Swarm Optimization resampling for vision tracking. *Expert Systems with Applications.* **37** (2010) 8910-8914.

[8] Chunlin Ji,Yangyang Zhang,Mengmeng Tong, et al. Particle Filter with Swarm Move for Optimization. *Lecture Notes in Computer Science.* (2008)909-918.

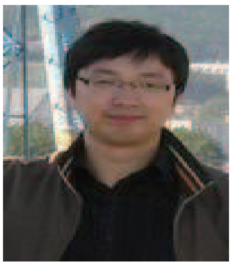
[9] Xin Liu, Mahesan Niranjan. State and parameter estimation of the heat shock response system using Kalman and particle filters. *Bioinformatics.* **28** (2012)1501-1507.

[10] Ying Li, Bendu Bai, Yanning Zhang. Improved particle swarm optimization algorithm for fuzzy multi-class SVM. *Journal of Systems Engineering and Electronics.* **21**(2010) 509-513.

[11] Pan Zhengjun, Kang Lishan. An adaptive evolutionary algorithm for numerical optimization. *Lecture in computer science. Heidelberg, germany: springer-verlag,* **1285**(1997) 27-34.

Yuming Bo received the Master's degree from Nanjing University of Science and Technology in 1987, and the PhD degree in automation from Nanjing University of Science and Technology in 1996. He is currently a professor , PhD supervisor and president of school of automation in Nanjing University of Science and Technology. His research interests are in the areas of Control theory and control applications, information processing, intelligent optimization algorithm and information systems.

Zhimin Chen received Master's degree from Nanjing University of Science and Technology in 2010. He is currently a PhD candidate in Nanjing University of Science and Technology. His research interests are in the areas of Mathematics, intelligent optimization, network architecture and information systems.



Jie Zhang received the Master's degree from Nanjing University of Science and Technology in 2005, and the PhD degree in automation from Nanjing University of Science and Technology in 2006. He is currently an associate professor in Nanjing University of Science and Technology. His research interests are in the areas of particle filter, information processing, information systems and intelligent optimization algorithm .

Jianliang Zhu received Master's degree from Nanjing University of Science and Technology in 2007, and the PhD degree in automation from Nanjing University of Science and Technology in 2012. He is currently a supervisor of master students in Nanjing University of Science and Technology. His research interests are in the areas of Control theory and control applications, information processing, intelligent optimization algorithm and information systems.