Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 676289, 10 pages
http://dx.doi.org/10.1155/2015/676289

Research Article

A Novel Strong Tracking Fault Prognosis Algorithm

Qi Zhang, Wei Jiang, Tian-Mei Li, and Jian-Fei Zheng

Unit 302, Xi'an Research Institute of High-Tech, Xi'an 710025, China

Correspondence should be addressed to Qi Zhang; zhangqi6530@163.com

Received 27 November 2014; Revised 21 December 2014; Accepted 21 December 2014

Academic Editor: Gang Li

Copyright © 2015 Qi Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Improving the ability to track abruptly changing states and resolving the degeneracy are two difficult problems to particle filter applied to fault prognosis. In this paper, a novel strong tracking fault prognosis algorithm is proposed to settle the above problems. In the proposed algorithm, the artificial immunity algorithm is first introduced to resolve the degeneracy problem, and then the strong tracking filter is introduced to enhance the ability to track abruptly changing states. The particles are updated by strong tracking filter, and better particles are selected by utilizing the artificial immune algorithm to estimate states. As a result, the degeneracy problem is resolved and the accuracy of the proposed fault prognosis algorithm is improved accordingly. The feasibility and validity of the proposed algorithm are demonstrated by the simulation results of the standard validation model and the DTS200 system.

1. Introduction

Particle filter (PF) is a leading and powerful algorithm for estimating the states of nonlinear or non-Gaussian systems. The past decades have witnessed a wide range of applications, including target tracking [1–4], data detection [5], modeling [6, 7], price forecasting, and fault detection [8–11]. On the other hand, a great number of investigators are interested in enriching particle filtering framework, and many new particle filters are proposed in recent years [12–15]. In these studies, it is found that resolving the degeneracy problem and improving the ability to track abruptly changing states are two difficult problems to particle filter applied to fault prognosis [16–20]. The degeneracy problem means that most particles are assigned to zero weights. As a result, the performance of the particle filter deteriorates because most computational resource is wasted. It is noted that, however, degeneracy can be reduced by resampling or choosing good importance sampling functions. Along this line of research, many resampling algorithms have been proposed for reducing the degeneracy. In these resampling algorithms, sequential importance resampling (SIR) is the representation which largely copies the particles with larger weights to replace the particles with smaller weights [21, 22]. The degeneracy problem is partially addressed, but the sample impoverishment is not fully concerned. Sample impoverishment means that most particles are the same in the set of particles since the particles with larger weights are largely copied. In this circumstance, choosing good importance sampling functions deserves further studies, and many investigators are interested in this question. For example, extended Kalman filter (EKF) was introduced to propose extended particle filter (EPF) by De Freitas et al. [6], and unscented Kalman filter (UKF) was introduced to propose unscented particle filter (UPF) by van der Merwe et al. [23]. Both EPF and UPF can resolve the degeneracy problem, but they cannot track abruptly changing states due to the disadvantages of EKF and UKF.

Strong tracking filter (STF) has good performance for tracking abruptly changing states, and thus it can be used to update particles. In addition, it is known that artificial immunity (AI) can search for the best one from all the range, and thus it can be used to clone and vary particles. Therefore, in this paper STF and AI algorithms are utilized jointly to improve particle filter algorithm. As a result, a novel fault prognosis algorithm based on strong tracking artificial immunity particle filter (STAIPF) is proposed to settle the above discussed problems. In the proposed algorithm, the artificial immunity algorithm is first introduced to resolve the degeneracy problem, and then the strong tracking filter is introduced to enhance the ability to track abruptly changing states. More specifically, the particles are updated by strong tracking filter, and better particles for states estimation are selected by utilizing the artificial immune algorithm to enhance the diversity of samples. Therefore, the degeneracy problem and sample impoverishment are resolved simultaneously, and the accuracy of the proposed fault prognosis algorithm is improved as well. Finally, the feasibility and validity of the proposed algorithm are demonstrated by the simulation results of the standard validation model and the DTS200 system.

The remainder of this paper is structured as follows. In Section 2, the particle filter is introduced. Section 3 provides the artificial immune algorithm. In Section 4, we present a strong tracking filter. A novel strong tracking fault prognosis algorithm is proposed in Section 5. Section 6 provides simulation results. This paper is concluded in Section 7.

2. Particle Filter

Actually, particle filter is a sequential Monte Carlo methodology. Its primary principle is to recursively compute relevant probability distributions by importance sampling and to approximate the probability distributions with discrete random variables. Detailed information of particle filter can be found in [21].

In general, the following state-space and observation equations are considered:

$$\begin{aligned} x_k &= f\left(x_{k-1}, v_{k-1}\right), \\ y_k &= h\left(x_k, n_k\right), \end{aligned} \tag{1}$$

where the subscript k denotes time index, $y_k \in R^{n_y}$ is an observations vector, $x_k \in R^{n_x}$ is a state vector, $v_k \in R^{n_v}$ is a system noise vector, $n_k \in R^{n_n}$ is an observation noise vector, $h(\cdot)$ is a measurement function, and $f(\cdot)$ is a system transition function. The first equation is known as the state equation while the second one is known as the measurement equation. It is usually assumed that the analytical forms of the two functions and the distributions of the two noises in (1) are known. Then, the object is to recursively estimate x_k based on the observations y_k and the above assumptions.

Due to the general nature of model (1) and the impact of non-Gaussian noises, it is difficult to solve the above filtering problem in an analytical manner. In this case, particle filter is an effective alternative. In particle filter, the number of effective particles is commonly denoted by N_{eff}, which is used to weigh the degeneracy degree of the particles. In other words, the smaller N_{eff} implies the worse degeneracy degree. Here, N_{eff} is defined as

$$N_{\text{eff}} = \text{round}\left(\frac{1}{\sum_{i=1}^{N}\left(w_k^i\right)^2}\right), \tag{2}$$

where w_k^i denotes the normalized weighting coefficient of particle i at time k, and round$(\cdot)$ denotes rounding to the nearest integer.

Based on the above descriptions, the SIR algorithm can be summarized as follows [21], which is one of the most popular particle filtering algorithms.

A Procedure Description of SIR

Step 1 (initialization). Draw N particles according to initial importance function, and $\langle x_k^i, 1/N\rangle$ is assumed to denote the ith particle, and then set $k = 1$.

Step 2. Update.

$$x_k^i = f\left(x_{k-1}^i, v_{k-1}\right), \tag{3}$$

where x_k^i denotes the particle i at time k, which is updated according to formula (3).

Step 3. Assign the weighting coefficient of x_k^i according to

$$w_k^i = w_{k-1}^i p\left(y_k \mid x_k^i\right). \tag{4}$$

Step 4. Normalize weighting coefficient by

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^{N} w_k^i}. \tag{5}$$

Step 5. If $N_{\text{eff}} < N/3$, then run system resampling and assign the same weighting coefficient to all particles:

$$w_k^i = \frac{1}{N}. \tag{6}$$

End.

Step 6. Estimate the state at time k according to

$$x_k^* = \sum_{i=1}^{N} x_k^i \times w_k^i. \tag{7}$$

Step 7. Return to Step 2.

3. Artificial Immune Algorithm

Artificial immune algorithm (AIA), heredity algorithm, and evolutionary algorithm are all bionic algorithms which simulate the behaviors of natural organisms. Artificial immune algorithm is characterized by its diversity, self-regulation, clone, and mutation, which is a global searching algorithm based on natural immune systems [24, 25].

Antigen and antibody correspond to target function and possible value of the associated optimization problem, respectively. Generally, affinity has two forms. One is the appetency which denotes the matching degree between antigen and antibody. The other is the repellency which denotes the similar degree of antigen and antibody, and the repellency ensures the diversity of the antibodies. Immune algorithm is based on the memory cell and clone number calculated according to affinity, which can converge to the optimal solution ensured by the above steps. The basic procedure of immune algorithm is summarized as Figure 1.

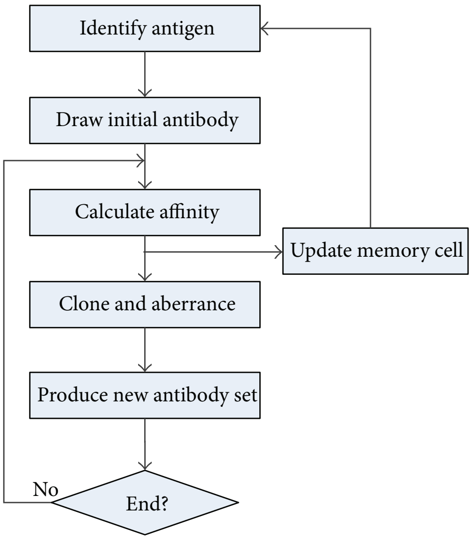


Figure 1: The basic procedure of immune algorithm.

At first, we should identify antigen and draw the initial antibody, then calculate the affinity between the antibodies and antigen, and update the memory cell. Next, clone and mutate the better antibodies to produce the new antibody set. Finally, return to calculate the affinity until the end.

4. Strong Tracking Filter

The usual EKF cannot track abruptly changing states when the system and the filter become steady, which may result in greater tracking error and even divergence. It is known that strong tracking filter has the advantage to track abruptly changing states. In this paper, the adopted strong tracking filter is a suboptimal fading extended Kalman filter (SFEKF) [26, 27].

Consider a kind of systems as follows:

$$\begin{aligned} x(k+1) &= f(k,u(k),x(k)) + \Gamma(k)\,v(k), \\ y(k+1) &= h(k+1,x(k+1)) + e(k+1), \end{aligned} \tag{8}$$

where the integer $k \geq 0$ is a discrete time variable, $x \in R^n$ is a state vector, $u \in R^p$ is an input vector, and $y \in R^m$ is an output vector. $f : R^p \times R^n \to R^n$ and $h : R^n \to R^m$ are nonlinear functions which have first-order continuous partial derivatives with respect to the states. $\Gamma \in R^{n\times q}$ is a known matrix. $v(k)$ is a q-dimension Gaussian white noise vector, and $e(k)$ is an m-dimension Gaussian white noise vector. In this paper, we consider the case that the noises have the following statistical characteristics:

$$\begin{gathered} Ev(k) = Ee(k) = 0, \\ E\left[v(k)\ \ v^T(j)\right] = Q(k)\,\delta_{k,j}, \\ E\left[e(k)\ \ e^T(j)\right] = R(k)\,\delta_{k,j}, \\ E\left[v(k)\ \ e^T(j)\right] = 0, \end{gathered} \tag{9}$$

where $Q(k)$ is a symmetric nonnegative definite matrix, $R(k)$ is a symmetric positively definite matrix, and $x(0) \sim N(x_0, P_0)$ is the initial state. It is further assumed that $x(0)$ is statistically independent of $v(k)$ and $e(k)$.

For the considered system represented by (8), a procedure description of strong tracking filter is summarized as follows [26, 27].

A Procedure Description of Strong Tracking Filter

Step 1. Set $k = 1$, and choose the initial value of $\hat{x}(0 \mid 0)$, $P(0 \mid 0)$ and infirmness gene β.

Step 2. Calculate $\hat{x}(k+1 \mid k)$ according to

$$\hat{x}(k+1 \mid k) = f(k, u(k), \hat{x}(k \mid k)). \tag{10}$$

Step 3. Calculate $H(k+1, \hat{x}(k+1 \mid k))$ and $F(k, \hat{x}(k \mid k))$ according to

$$\begin{aligned} H(k+1,\hat{x}(k+1 \mid k)) &= \left.\frac{\partial h(k+1,x(k+1))}{\partial x}\right|_{x(k+1)=\hat{x}(k+1\mid k)}, \\ F(k,\hat{x}(k \mid k)) &= \left.\frac{\partial f(k,u(k),x(k))}{\partial x}\right|_{x(k)=\hat{x}(k\mid k)}. \end{aligned} \tag{11}$$

Step 4. Calculate $\gamma(k+1)$ according to

$$\gamma(k+1) = y(k+1) - h(k+1, \hat{x}(k+1 \mid k)). \tag{12}$$

Step 5. Calculate the fading factor $\lambda(k+1)$ as

$$V_0(k+1) = \begin{cases} \gamma(1)\,\gamma^T(1), & k = 0, \\ \dfrac{\rho V_0(k) + \gamma(k+1)\,\gamma^T(k+1)}{1+\rho}, & k \geq 1, \end{cases} \tag{13}$$

where the parameter ρ is named forgetting factor, and $0 \leq \rho \leq 1$. In general, we set $\rho = 0.95$.

$$\begin{aligned} N(k+1) &= V_0(k+1) - H(k+1,\hat{x}(k+1 \mid k))\,\Gamma(k) \\ &\quad \times Q(k)\,\Gamma^T(k)\,H^T(k+1,\hat{x}(k+1 \mid k)) \\ &\quad - \beta \cdot R(k+1), \\ M(k+1) &= H(k+1,\hat{x}(k+1 \mid k))\,F(k,u(k),\hat{x}(k \mid k)) \\ &\quad \times P(k \mid k)\,F^T(k,u(k),\hat{x}(k \mid k))\,H^T \\ &\quad \times (k+1,\hat{x}(k+1 \mid k)), \\ \lambda_0 &= \frac{\operatorname{tr}[N(k+1)]}{\operatorname{tr}[M(k+1)]}, \\ \lambda(k+1) &= \begin{cases} \lambda_0, & \lambda_0 \geq 1, \\ 1, & \lambda_0 < 1. \end{cases} \end{aligned} \tag{14}$$

Step 6. Estimate the value of $\hat{x}(k+1 \mid k+1)$ by the following recursive formulas:

$$\begin{aligned} P(k+1 \mid k) = {} & \lambda(k+1) F(k, u(k), \hat{x}(k \mid k)) \\ & \times P(k \mid k) F^T(k, u(k), \hat{x}(k \mid k)) \\ & + \Gamma(k) Q(k) \Gamma^T(k), \\ K(k+1) = {} & P(k+1 \mid k) H^T(k+1, \hat{x}(k+1 \mid k)) \\ & \times \left[H(k+1, \hat{x}(k+1 \mid k)) P(k+1 \mid k) H^T \right. \\ & \left. \times (k+1, \hat{x}(k+1 \mid k)) + R(k+1) \right]^{-1}, \\ \hat{x}(k+1 \mid k+1) = {} & \hat{x}(k+1 \mid k) + K(k+1) \gamma(k+1). \end{aligned} \tag{15}$$

Step 7. Calculate $P(k+1 \mid k+1)$ by

$$\begin{aligned} P(k+1 \mid k+1) = {} & [I - K(k+1) H(k+1, \hat{x}(k+1 \mid k))] \\ & \cdot P(k+1 \mid k) \end{aligned} \tag{16}$$

$k+1 \rightarrow k$, andreturn to Step 2 until the end of circle.

5. Strong Tracking Fault Prognosis Algorithm

A novel strong tracking fault prognosis algorithm is proposed in this section, in which the artificial immunity algorithm is introduced to resolve the degeneracy problem, and the strong tracking filter is introduced to enhance the ability to track abruptly changing states. To be specific, the particles are updated by strong tracking filter, and better particles for estimation are selected by artificial immune algorithm to enhance the diversity of samples. As such, the proposed algorithm can predict fault more accurately.

The following is the detailed idea description of the proposed algorithm. First, draw N particles as the initial antibodies according to the important sampling function. Then, we update the N particles by STF and calculate the appetency and repellency of the antibodies to ascertain the clone number of every antibody. A new antibody set is made up of the initial antibodies and clone antibodies, in which some similar antibodies are discarded by respectively calculating affinities of all the antibodies. By this procedure, the good ones are chosen to estimate the state of next time. Thus, the degeneracy problem and the sample impoverishment are resolved, and the ability of tracking abruptly changing states is much improved because of the introduced STF in the proposed fault prognosis algorithm.

In this paper, we consider the nonlinear dynamic system as (1), and set 1 and $\langle x_k^i, w_k^i \rangle$ as the antigen and antibody, respectively. Based on the STF algorithm and AI algorithm, we propose a novel strong tracking fault prognosis algorithm, which is illustrated step by step in the following.

A Procedure Description of the Proposed Fault Prognosis Algorithm

Step 1 (initialization). Draw N particles according to the importance function at time k, and $\langle x_k^i, 1/N \rangle$ is assumed to denote each particle; then set $k = 1$ and choose the initial value of $\hat{x}(0 \mid 0)$, $P(0 \mid 0)$, and infirmness gene β.

Step 2 (update). Update the particles using the strong tracking filter, which calculates x_k^i according to (10)–(15).

Step 3 (assign weights). Assign the weighting coefficients to x_k^i according to (4) and then normalize weighting coefficients according to (5).

Step 4 (build initial antibodies). The N updated particles are regarded as initial antibodies put in the memory cell.

Step 5 (calculate the affinity). Calculate $\text{fit}_k(i)$ and $\text{off}_k(i, j)$ according to the following formulas which are the appetencies of every antibody and the repellencies between the antibodies.

$$\begin{aligned} \text{fit}_k(i) &= 1 - w_k^i, \\ \text{off}_k(i, j) &= \left| x_k^i - x_k^j \right|. \end{aligned} \tag{17}$$

Step 6 (clone the antibodies). Clone the antibodies according to the appetencies of every antibody. Here the clone number $\text{knum}_k(i)$ is ascertained by

$$\text{knum}_k(i) = \text{round}\left(N \times \cos\left(\frac{\pi}{2} \times \text{fit}_k(i) \right) \right). \tag{18}$$

Step 7 (mutation). Mutate the whole antibodies according to

$$x_k^i = x_k^i + \text{fit}_k(i) \ \text{randn}(\cdot). \tag{19}$$

Step 8 (choose the optimal particles). Calculate $\text{fit}_k(i)$ and $\text{off}_k(i, j)$ of the mutated antibodies and discard one of the two antibodies if $\text{off}_k(i, j) < 0.0001$ which is set according to the value range of particles. Sort the remainder antibodies in terms of $\text{fit}_k(i)$ and choose the anterior N antibodies to replace the memory cell. Return to Step 3 until the ending criterion is satisfied.

Step 9 (estimate). Consider the N antibodies in memory cell as new particles to estimate the states at $k+1$ according to (7).

Step 10. j-step-ahead prediction.

$$x_{k+j|k+j-1}^i = f\left(x_{k+j-1}^i, v_{k+j-1} \right), \quad j \in [1, p]. \tag{20}$$

Step 11. j-step-ahead fault prognosis probability at k time.

$$\text{fault}(j, k) = \sum_{i=1}^{N} w_k^i I\left(x_{k+j|k+j-1}^i \in \omega_0 \right), \tag{21}$$

where ω_0 denotes the fault state.

Step 12. Calculate the weighted fault prognosis probability at $k + j$ time.

$$p_{\text{fault}(k+j)} = \sum_{j=1}^{p} \text{fault}(j, k) \times w_j^*. \tag{22}$$

Step 13. If $N_{\text{eff}} < N/3$, resample the particles.

Step 14. Set $k = k + 1$, return to Step 2.

6. Simulation Experiments and Results Analysis

In this section, two simulation experiments are presented to demonstrate the effectiveness of the proposed algorithm. The first one is a standard validation model that has been widely used to validate the ability to track the states and the ability to resolve the degeneracy problem and sample impoverishment. The second experiment is used to validate whether or not the proposed algorithm can predict the system's fault with abruptly changing states.

6.1. Experiment 1 of the Standard Validation Model. The model used in this section can be found in many studies [28], which is a standard validation model to validate the performance of various particle filters.

Specifically, the state equation and the measurement equation of the standard validation model are represented as

$$\begin{aligned} x_k &= f(x_{k-1}, k) + v_{k-1}, \\ y_k &= \frac{x_k^2}{20} + n_k, \end{aligned} \tag{23}$$

where

$$f_k(x_{k-1}, k) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2k), \tag{24}$$

and v_k is a Gaussian noise with mean 0 and variance 10, and n_k is also a Gaussian noise with mean 0 and variance 1. The normal distribution $N(0, 5)$ is the initial importance function. For implementing the proposed algorithm, we set $x_0 = 0.1$, $\rho = 0.95$, $\beta = 4$, and $N = 100$, where N is the number of particles. In this simulation, we choose cycling 5 times as the terminal criterion. The estimated results of strong tracking artificial immunity particle filter for the standard validation model are shown in Figure 2.

Figure 2 shows that the estimate values of states are almost the same with the real values of states, so STAIPF can track states of the standard validation model and the proposed algorithm is feasible and effective.

The number of effective particles N_{eff} is used to weight the degeneracy degree of the particles. The smaller N_{eff} denotes more serious degeneracy degree. In this simulation, STAIPF and STPF run 10 times and the means of N_{eff} are respectively calculated to validate whether STAIPF can resolve the degeneracy problem or not to a certain extent. Table 1 is the simulation results of effective particles, in which N_{eff_1} and $\widetilde{N}_{\text{eff}_1}$, respectively, denote the number of effective particles and the mean of STAIPF, and N_{eff_2} and $\widetilde{N}_{\text{eff}_2}$, respectively, denote the number of effective particles and the mean of STPF.

Table 1: The number of effective particles.

Time	1	2	3	4	5	6	7	8	9	10
N_{eff_1}	91	78	89	98	53	89	88	79	63	96
N_{eff_2}	34	28	36	42	58	28	15	18	26	49

$\widetilde{N}_{\text{eff}_1} = 82.4.$
$\widetilde{N}_{\text{eff}_2} = 33.4$

Table 1 and the mean of N_{eff} of two algorithms show that $\widetilde{N}_{\text{eff}_1}$ is 2.4 times as much as $\widetilde{N}_{\text{eff}_2}$. Therefore, STAIPF is better than STPF about the ability to resolve the degeneracy problem. As a result, STAIPF can resolve degeneracy problem because of introducing artificial immunity algorithm.

Further, the normalized weighting coefficients of particles at the final time are shown in Figure 3 to demonstrate that STAIPF can resolve the degeneracy problem better than STPF.

We can find that the normalized weighting coefficients of particles at the final time about STAIPF are all greater than zero, but most of the normalized weighting coefficients about STPF equals zero. As a result, STAIPF can resolve the degeneracy problem better than STPF.

6.2. Experiment 2 of the DTS200 System

6.2.1. System Description. A modified mathematical model of the DTS200 system [26] as shown in Figure 4 was produced by the Amira Automation Company in Germany. DTS200 system is composed of three cylinders tank 1 (T_1), tank 2 (T_2), and tank 3 (T_3) with the cross-section area A. The cylinders are connected to each other by pipes with the cross-section area S_n. The outflow valve with circular cross-section area S_n is located at tank 2. The outflowing distilled water collected in a water tank supplies to the pump1 and pump2. A leak valve with the cross-section area S_1 is fixed on the bottom of each tank, which is usually closed.

The pump flows Q_1 and Q_2 denote the input signals, and the liquid heights h_1, h_2, and h_3 of T_1, T_2, and T_3 denote the output signals. The liquid heights h_1 and h_2 measured by 3 piezoresistive difference pressure sensors are controlled by Q_1 and Q_2.

The system is modeled by

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{A}(\mathbf{x}) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y} &= \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T. \end{aligned} \tag{25}$$

With the vectors defined by

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \triangleq \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}, \qquad \mathbf{u} \triangleq \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \\ \mathbf{A}(\mathbf{x}) &= \frac{1}{A} \begin{bmatrix} -Q_{13} \\ Q_{32} - Q_{20} \\ Q_{13} - Q_{32} \end{bmatrix}, \qquad \mathbf{B} = \frac{1}{A} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{26}$$

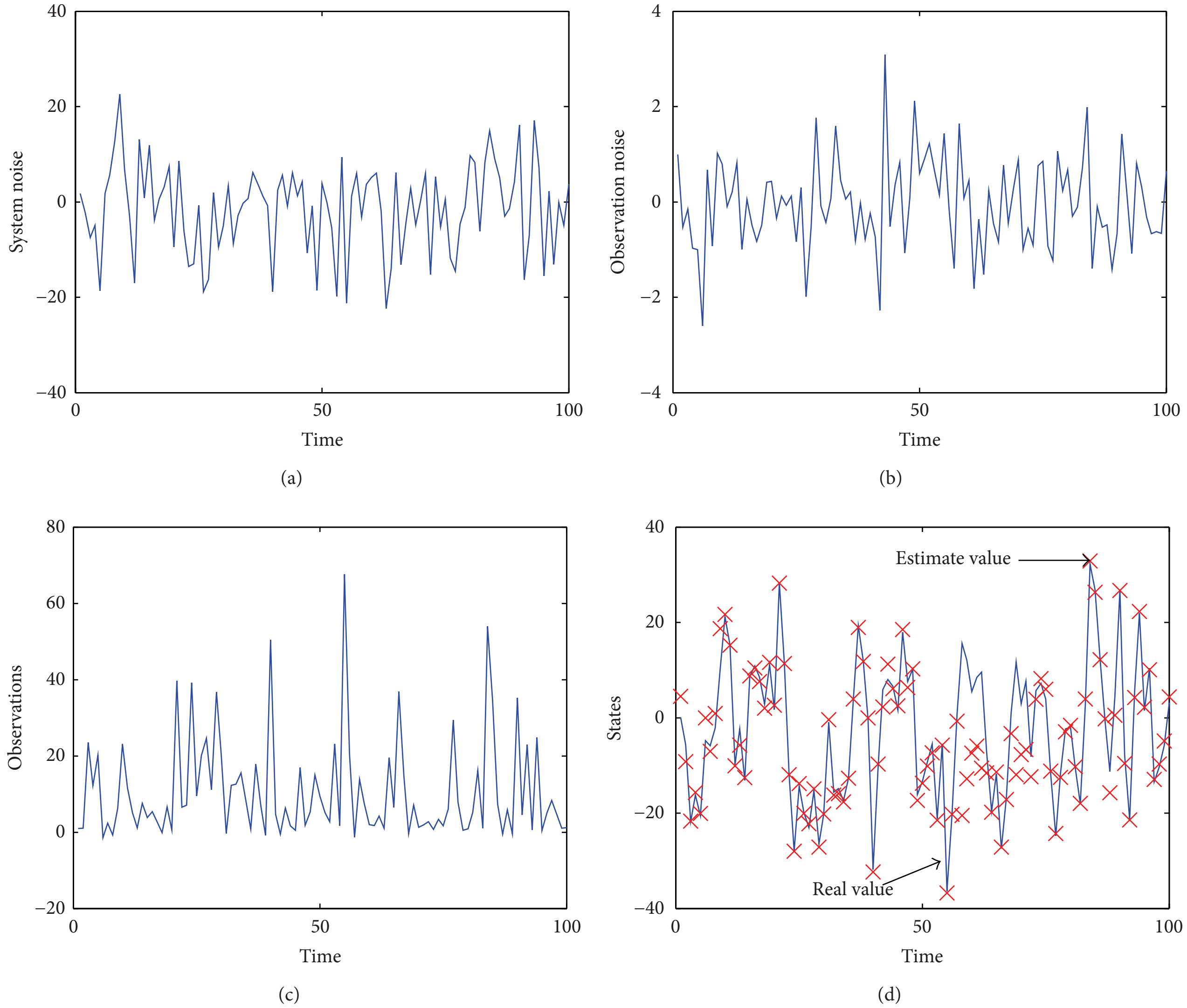


Figure 2: The estimated results of STAIPF for the standard validation model: (a) and (b) about system noise, (c) about the observations, and (d) about states estimate values.

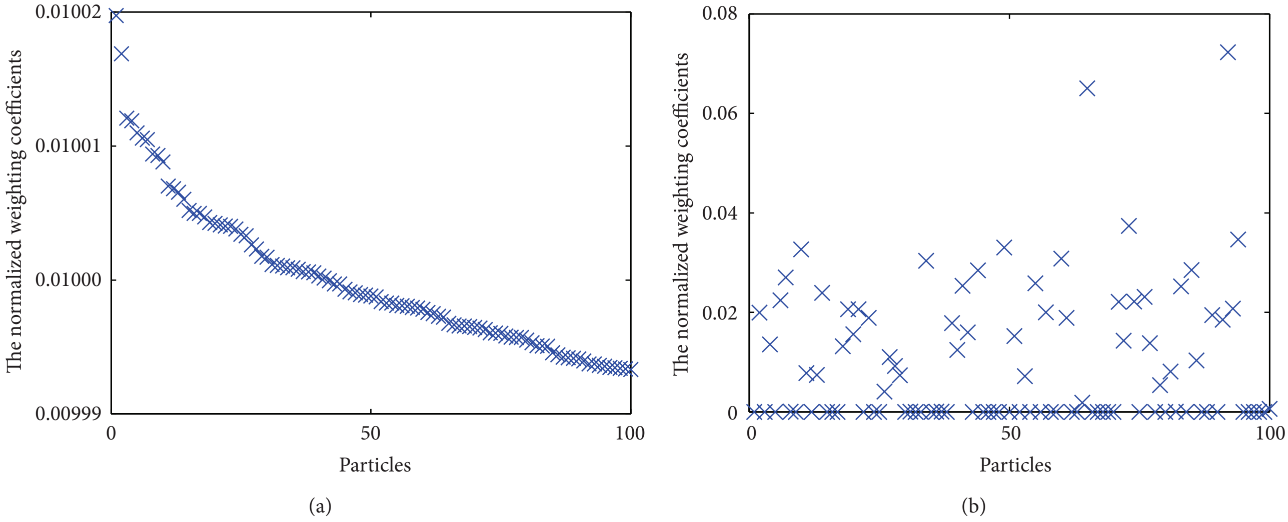


Figure 3: The normalized weighting coefficients of particles at the final time: (a) and (b) about STAIPF and STPF, respectively.

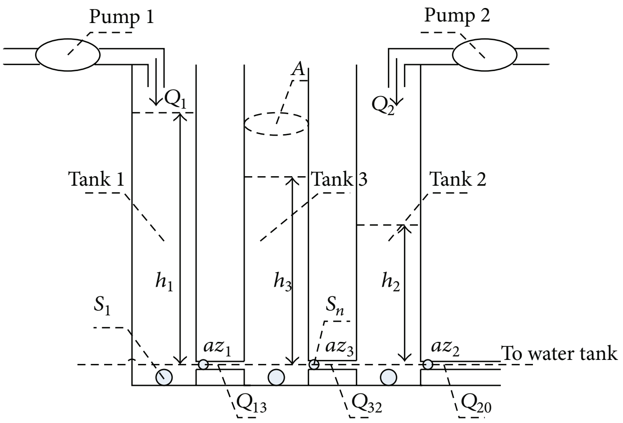


FIGURE 4: A structural schematic diagram of DTS200.

The significance of the above mathematical symbols is summarized as follows:

az_i: outflow coefficients,

h_i: liquid heights [m],

Q_{ij}: flow rates [m^3/s],

Q_1, Q_2: pump flow rates [m^3/s],

A: cross-section area of cylinder [m^2],

S_n: cross-section area of connection pipe [m^2],

where $i = 1, 2, 3$, and $(i, j) \in \{(1, 3); (3, 2); (2, 0)\}$. Consider

$$q = azS_n \operatorname{sgn}(\Delta h)\left(2g\left|\Delta h\right|\right)^{1/2}, \tag{27}$$

where g: earth's acceleration of gravity [$\mathrm{m/s}^2$], sgn(z): sign of the argument z, Δh: liquid height difference between two tanks connected to each other, q: flow in the connecting pipe, and az: outflow coefficient.

The results of the unknown quantities are calculated as

$$\begin{aligned} Q_{13} &= az_1 S_n \operatorname{sgn}(h_1 - h_3)\left(2g\left|h_1 - h_3\right|\right)^{1/2}, \\ Q_{32} &= az_3 S_n \operatorname{sgn}(h_3 - h_2)\left(2g\left|h_3 - h_2\right|\right)^{1/2}, \\ Q_{20} &= az_2 S_n \left(2gh_2\right)^{1/2}. \end{aligned} \tag{28}$$

The technical parameters of the DTS200 system are given as follows:

$$\begin{gathered} A = 0.0154\,\mathrm{m}^2, \qquad S_n = 5 \times 10^{-5}\,\mathrm{m}^2, \\ Q_{1\max} = Q_{2\max} = 100\,\mathrm{mL/s}, \qquad h_{\max} = 0.62\,(\pm 0.01\,\mathrm{m}), \\ g = 9.81\,\mathrm{m/s}^2, \qquad az_1^0 = 0.5, \\ az_2^0 = 0.6, \qquad az_3^0 = 0.5. \end{gathered} \tag{29}$$

For the above concerned system, it can be equivalently modeled with Euler discretization technique as the following nonlinear discrete model:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) + \Delta t \cdot \mathbf{A}(\mathbf{x}) + \Delta t \cdot \mathbf{B} \cdot \mathbf{u}(k) + \mathbf{w}(k), \\ \mathbf{y}(k) &= \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \mathbf{v}(k), \end{aligned} \tag{30}$$

where $\Delta t = 1\,\mathrm{s}$ is the sampling interval, the initial levels of T_1, T_2, and T_3 are $h_1^0 = 0.4\,\mathrm{m}$, $h_2^0 = 0.3\,\mathrm{m}$, $h_3^0 = 0.35\,\mathrm{m}$, respectively, and $Q_1 = Q_2 = 4.5 \times 10^{-5}\,\mathrm{m}^3/\mathrm{s}$.

To achieve fault prognosis, in this experimental study, the fault state ω_0 is defined as

$$\omega_0 \in \left\{\left|\frac{h_k - h_k^0}{h_k^0}\right| \geq 0.1\right\}, \quad k = 1, 2, 3. \tag{31}$$

6.2.2. Simulation Results and Analysis. The abrupt change fault is introduced to the system in order to verify the tracking ability of the proposed algorithm. We suppose that az_2 changes according to the following formula:

$$az_2 = \begin{cases} 0.6, & 0 < k \leq 20, \\ az_2^0 + (k - 20) \times 0.06, & 20 < k \leq 80, \\ 4.2, & 80 < k \leq 100. \end{cases} \tag{32}$$

To implement the proposed algorithm, we set the number of particles $N = 100$, $p = 5$, $\beta = 10$, $\rho = 0.95$, and the process noise and measurement noiseare $w(k) \sim N(0, 2 \times 10^{-4})$ and $v(k) \sim N(0, 1 \times 10^{-4})$, respectively. Further, we suppose that the simulation time is $100\Delta t$ and the initial particle meets the condition as $\mathbf{X}_0 \sim N([h_1^0, h_2^0, h_3^0]^T, 10^{-2})$.

The simulation results of state estimation and fault prognosis probability of the normal system are shown as Figure 5, which are performed by the strong tracking fault prognosis algorithm.

We can find from Figure 5 that the estimated values of the system states are all around the real ones with time and the weighted fault prognosis probability is always zero, and the normalized weighting coefficients of most of particles are not zero. As a result, the proposed algorithm can accurately predict the normal system and resolve the degeneracy problem.

The simulation results of the state estimation and fault prognosis probability of the faulty system are shown as Figure 6, which are performed by the strong tracking fault prognosis algorithm.

In this case, it is found from Figure 6 that the system states deviate from the real value when az_2 changes with time, and the range ability of h_2 is the biggest. The system fault will appear when h_2 deviates from the real value to a certain point. The normalized weighting coefficients of most of particles are not zero. As a result, the proposed algorithm can accurately predict the system fault in time and resolve the degeneracy problem well.

Further, we extract the values of h_2 and fault prognosis probability from Figures 6(a) and 6(b). In this paper, the

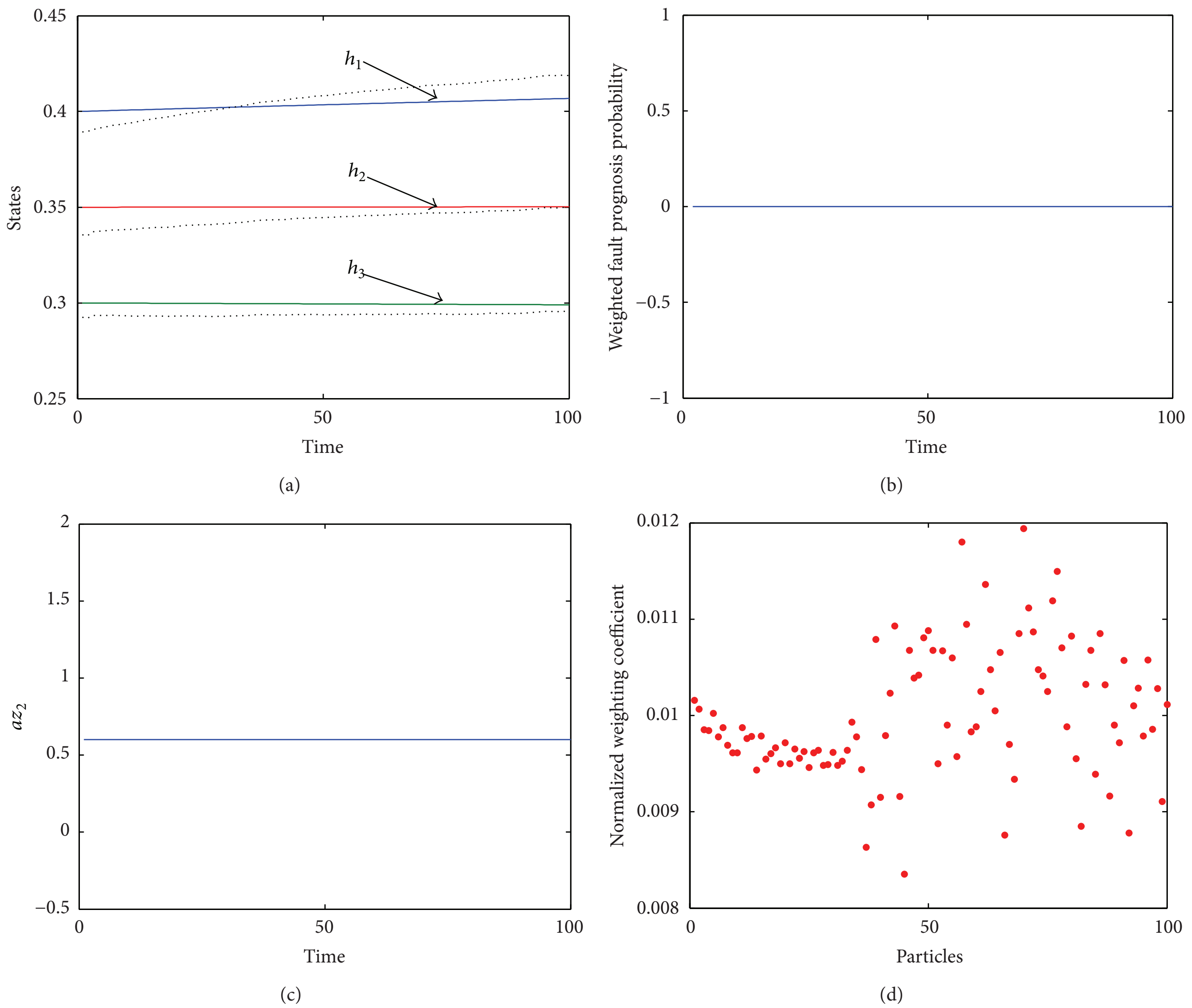


Figure 5: The simulation results of state estimation and fault prognosis probability of normal system: (a) about the results of states estimation, (b) about weighted fault prognosis probability, (c) about az_2 changing with time, and (d) about the normalized weighting coefficient of particles at the final time.

Table 2: The relation between prob and h_2.

Time	66	67	68	69	70	71	72	73	74	75
h_2	0.2798	0.2784	0.2770	0.2762	0.2754	0.2736	0.2716	**0.2698**	**0.2682**	**0.2653**
prob	0	0	0	0.2435	0.4353	**0.5609**	**0.9214**	**1.0000**	**1.0000**	**1.0000**

fault is declared when the fault prognosis probability prob is greater than 0.5. Specifically, the values of h_2 and the weighted fault prognosis probability prob during the period [66, 75] are shown in Table 2 to demonstrate that the proposed algorithm can accurately predict the system fault.

According to the definition of the fault state ω_0 in (31), the system fault will happen when h_2 deviates from the real value to 0.2700 at time 73. It is observed from Table 2 that the proposed algorithm can predict the fault at time 71. As a result, the algorithm can accurately predict the system fault in time.

Together with the above simulation results, we can observe that the degeneracy problem and the sample impoverishment can be resolved simultaneously in the proposed fault prognosis algorithm, and thus the feasibility and validity of the proposed algorithm are demonstrated.

7. Conclusions

Considering that SIR and EPF cannot track abruptly changing states and STPF can track abruptly changing states, a novel strong tracking fault prognosis algorithm is proposed

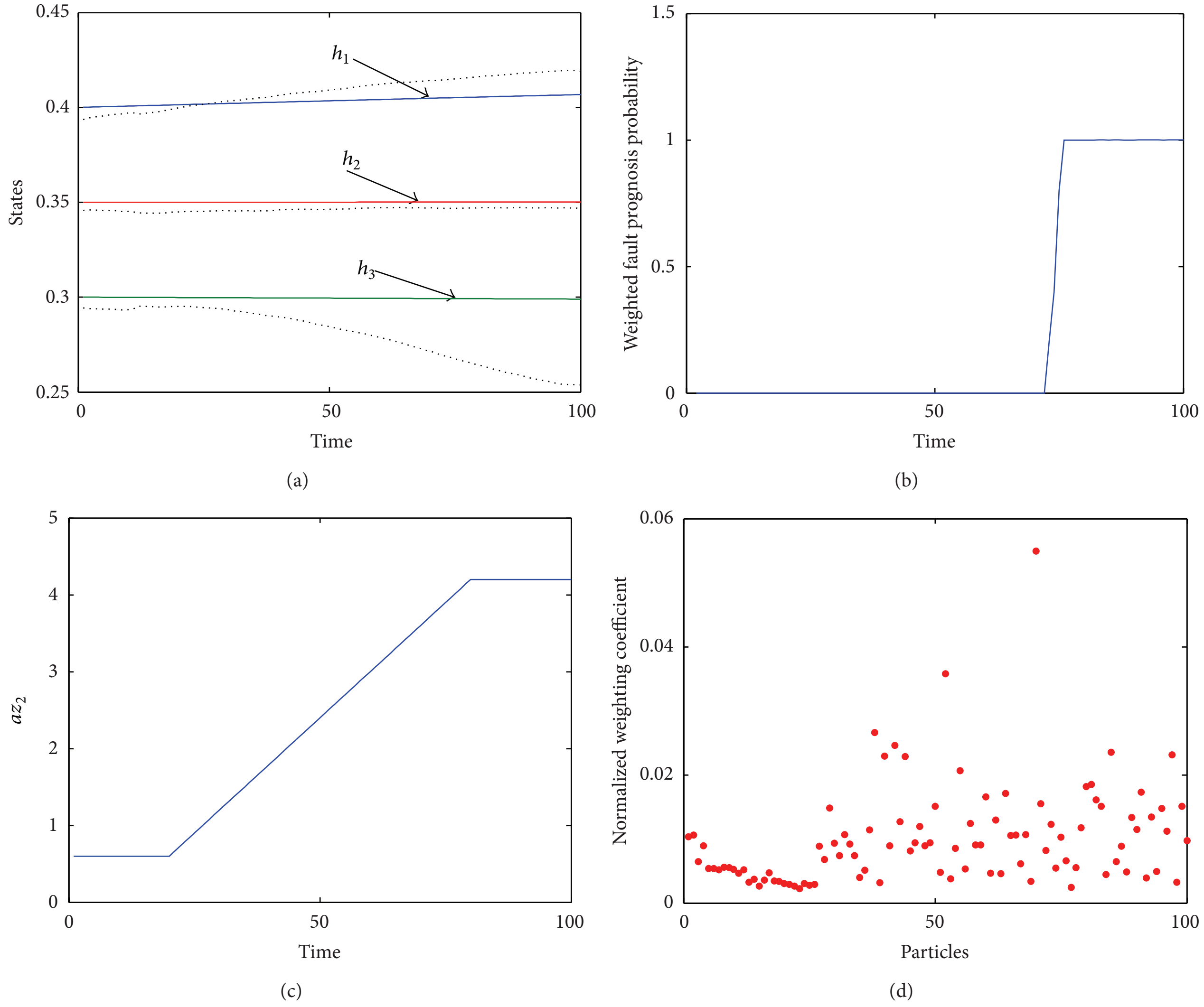


FIGURE 6: The simulation results of the state estimation and fault prognosis probability of the faulty system: (a) about the results of states estimation, (b) about weighted fault prognosis probability, (c) about az_2 changing with time, and (d) about the normalized weighting coefficient of particles at the final time.

to settle the degeneracy problem and the sample impoverishment in particle filter, which is applied to predict the fault of systems with abruptly changing states. The simulation results of the standard validation model show that the strong tracking fault prognosis algorithm can resolve degeneracy problem, and the simulation results of the DTS200 system demonstrate that the proposed algorithm can track abruptly changing states and accurately predict the system fault in time. As a result, the proposed algorithm is feasible and effective for the applications in fault prognosis.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61104223, 61174030, 61304103, 61374126, and 61473094 and the China's Postdoctoral Science Fund 2014M552670.

References

[1] W. Ng, J. Li, S. Godsill, and J. Vermaak, "A review of recent results in multiple target tracking," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 40–45, September 2005.

[2] J. A. Pocock, S. L. Dance, and A. S. Lawless, "State estimation using the particle filter with mode tracking," *Computers and Fluids*, vol. 46, no. 1, pp. 392–397, 2011.

[3] A. Xydes, M. Moline, C. G. Lowe, T. J. Farrugia, and C. Clark, "Behavioral characterization and particle filter localization to improve temporal resolution and accuracy while tracking acoustically tagged fishes," *Ocean Engineering*, vol. 61, no. 15, pp. 1–11, 2013.

[4] C. Chang and R. Ansari, "Kernel particle filter for visual tracking," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 242–245, 2005.

[5] T. Ghirmai, M. F. Bugallo, J. Míguez, and P. M. Djurić, "A sequential Monte Carlo method for adaptive blind timing estimation and data detection," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2855–2865, 2005.

[6] J. F. G. De Freitas, M. Niranjan, A. H. Gee, and A. Doucet, "Sequential Monte Carlo methods to train neural network models," *Neural Computation*, vol. 12, no. 4, pp. 955–993, 2000.

[7] D. Ryu, F. Liang, and B. K. Mallick, "Sea surface temperature modeling using radial basis function networks with a dynamically weighted particle filter," *The Journal of the American Statistical Association*, vol. 108, no. 501, pp. 111–123, 2013.

[8] X. R. Li, C. W. Yu, S. Y. Ren, C. H. Chiu, and K. Meng, "Day-ahead electricity price forecasting based on panel cointegration and particle filter," *Electric Power Systems Research*, vol. 95, no. 2, pp. 66–76, 2013.

[9] T. Wei, Y. F. Huang, and P. Chen, "Particle filtering for adaptive sensor fault detection and identification," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3807–3812, Orlando, Fla, USA, May 2006.

[10] M. Z. Chen and D. H. Zhou, "Particle filter based fault prediction of nonlinear systems," in *IFAC Symposium Proceedings of Safe Process*, pp. 2971–2977, Elsevier Science, Washington, DC, USA, 2001.

[11] F. Cadini, E. Zio, and G. Peloni, "Particle filtering for the detection of fault onset time in hybrid dynamic systems with autonomous transitions," *IEEE Transactions on Reliability*, vol. 61, no. 1, pp. 130–139, 2012.

[12] M. Ades and P. J. van Leeuwen, "An exploration of the equivalent weights particle filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 139, no. 672, pp. 820–840, 2013.

[13] P. M. Stano, Z. Lendek, and R. Babuska, "Saturated particle filter: almost sure convergence and improved resampling," *Automatica*, vol. 49, no. 1, pp. 147–159, 2013.

[14] W. Hassan, N. Bangalore, P. Birch, R. Young, and C. Chatwin, "An adaptive sample count particle filter," *Computer Vision and Image Understanding*, vol. 116, no. 12, pp. 1208–1222, 2012.

[15] S.-H. Park, Y.-J. Kim, and M.-T. Lim, "Novel adaptive particle filter using adjusted variance and its application," *International Journal of Control, Automation and Systems*, vol. 8, no. 4, pp. 801–807, 2010.

[16] X. S. Si, C. H. Hu, X. Y. Kong, and D. H. Zhou, " A residual storage life prediction approach for systems with operation state switches," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6304–6315, 2014.

[17] G. Li, S. J. Qin, Y. Ji, and D. Zhou, "Reconstruction based fault prognosis for continuous processes," *Control Engineering Practice*, vol. 18, no. 10, pp. 1211–1219, 2010.

[18] X.-S. Si, M.-Y. Chen, W. Wang, C.-H. Hu, and D.-H. Zhou, "Specifying measurement errors for required lifetime estimation performance," *European Journal of Operational Research*, vol. 231, no. 3, pp. 631–644, 2013.

[19] J. Zhang, C. Hu, X. Si et al., "Remaining useful life estimation for systems with time-varying mean and variance of degradation processes," *Quality and Reliability Engineering International*, vol. 30, no. 6, pp. 829–841, 2014.

[20] X.-S. Si, W. Wang, C.-H. Hu, M.-Y. Chen, and D.-H. Zhou, "A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation," *Mechanical Systems and Signal Processing*, vol. 35, no. 1-2, pp. 219–237, 2013.

[21] P. M. Djurić, J. H. Kotecha, J. Zhang et al., "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[22] M. 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.

[23] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," Tech. Rep. CUED/F-INFENG/TR 380, Cambridge University, London, UK, 2000.

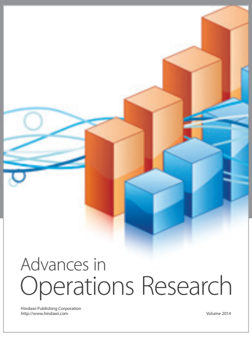
[24] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, Berlin, Germany, 1999.

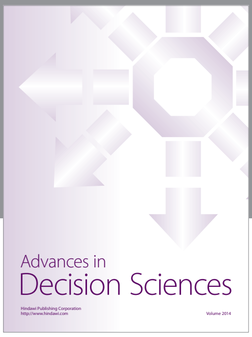
[25] X. W. Zhou, "Evolutionary algorithm and its application in artificial immune system," in *Proceedings of the 2nd International Symposium on Intelligent Information Technology Application (IITA '08)*, pp. 32–36, Shanghai, China, December 2008.

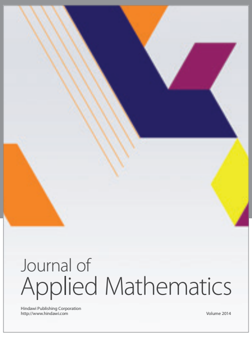
[26] D. H. Zhou and Y. Z. Ye, *Modern Fault Diagnosis and Fault Tolerant Control*, Tsinghua University press, Beijing, China, 2000.

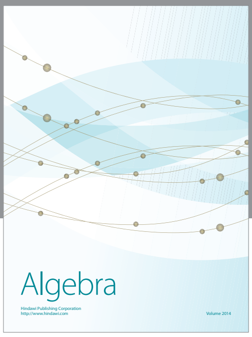
[27] W.-H. Wang, Z. Peng, and D.-H. Zhou, "State estimation of a class of hybrid systems in the presence of unknown mode transition," *Acta Automatica Sinica*, vol. 31, no. 3, pp. 456–457, 2005.

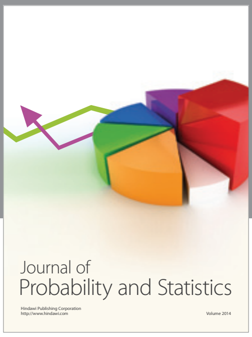
[28] Y. W. Mo and D. Y. Xiao, "Evolutionary particle filter and its application," *Control Theory and Applications*, vol. 22, no. 2, pp. 269–270, 2005.











Hindawi

Submit your manuscripts at
http://www.hindawi.com