

Research Article

Application of the Artificial Fish School Algorithm and Particle Filter Algorithm in the Industrial Process Control Particle Filtering Algorithm for Industrial Process Control

Zhaoxia Huang 

Department of Mathematics and Statistics, Ankang University, Ankang 725000, China

Correspondence should be addressed to Zhaoxia Huang; 2007990011@aku.edu.cn

Received 6 July 2020; Revised 27 July 2020; Accepted 4 August 2020; Published 15 September 2020

Guest Editor: Hussein Abulkasim

Copyright © 2020 Zhaoxia Huang. 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.

The Industrial Internet of Things (IIoT) is of strategic importance in the new era of industrial big data, creating a brand-new industrial ecosystem. Considering the unknown parameters in the IIoT-based industrial process control systems, this paper combines the artificial fish swarm algorithm (AFSA) and the particle filtering (PF) algorithm into the AFSA-PF algorithm based on the self-organizing state space (SOSS) model. The AFSA-PF algorithm not only can estimate the system state but also can make the sampling distribution of the unknown parameter to move the true parameter distribution. Ultimately, the true values of the unknown parameters are identified. In this way, the system model can gradually approximate the actual IIoT-based industrial process control system.

1. Introduction

The Internet of Things (IoT) has entered various areas of our lives, such as smart light bulbs and shared bikes to smart locks [1–4]. The IoT connects up a dazzlingly array of things, making our lives more and more convenient. When it comes to industry, the IoT also has lots to offer: industrial control, automation, metering, condition monitoring, and asset management, to name a few.

Coupled with emerging information technologies (ITs) such as cloud computing and big data, the IoT evolves into the industrial IoT (IIoT). The IIoT is of strategic importance in the new era of industrial big data, creating a brand-new industrial ecosystem. Under this ecosystem, intelligent automated machines, advanced predictive analysis, and human-machine collaboration are deeply integrated to enhance productivity, efficiency, and reliability. The IIoT makes it possible to connect embedded modules and products into largescale systems, laying the basis for intelligent industrial monitoring.

Traditionally, industrial processes are monitored by online systems with alarming function. There are two kinds

of industrial process monitoring systems: the manufacturing execution system (MES) and process control system. The process control system, which is more popular than the MES, monitors the states of industrial processes with configuration software, focusing on the online detection of machine failures and the change trend of key parameters.

The current process control system provides many modeling methods, namely, principal component analysis (PCA), kernel-PCA (KPCA), and support vector machine (SVM). During operation, the process control system collects the operational data continuously and selects the most suitable modeling method to predict faults online and judge whether it is necessary to issue alarms. However, the prediction accuracy of the current process control system is yet to be improved.

The IIoT-based control of industrial processes has better accuracy than the current process control system. But, the fault prediction is still hindered by the heavy presence of unknown parameters in the control system, which adds to the difficulty in mechanical maintenance. To solve the problem, this paper combines the artificial fish swarm algorithm (AFSA) and the particle filtering (PF) algorithm

into the AFSA-PF algorithm based on the self-organizing state space (SOSS) model and verifies the effectiveness of the algorithm through simulation. The artificial fish school algorithm is a random search optimization algorithm that simulates the ecological behavior of fish schools in the natural environment. It includes four adaptive behaviors of fish schools: foraging behavior, grouping behavior, rear-end behavior, and random behavior. It has been used in many engineering fields and achieved good results.

With the dawn of the Fourth Industrial Revolution (Industry 4.0), the IIoT has emerged as a communication network that allows devices to be accessed easily without sacrificing security and reliability. In general, the IIoT consists of four layers, including a sensing layer, site management layer, network layer, and application layer (the copyright of Figure 1 does not belong to the author).

The sensing layer collects information from various devices, e.g., radio-frequency identification (RFID) readers, cameras, and sensors. In this layer, intelligent wireless systems with sensors can automatically perceive and exchange information between devices and control them in a remote manner.

The site management layer provides an external interface for the industrial data and manages the data locally. This layer is equivalent to a local dispatching and management centre.

The network layer mainly transmits messages and processes information. This layer connects everything together, allowing them to share information with each other.

The application layer adopts suitable technologies to optimize the production processes or industrial applications.

The IIoT is generally deployed in four phases: First, smart sensors collect the real-time industrial data anytime, anywhere. Second, the communication network uploads the collected data in real time. Third, the uploaded data are modelled and analysed based on big data, cloud computing, and data mining. Fourth, the industrial production is upgraded through information management and platform integration, making production more efficient, resource utilization more complete, and production cost lower.

The traditional monitoring systems of industrial processes mainly support online monitoring and early warning. The IIoT-based intelligent monitoring system emphasizes on the diagnosis and prediction of abnormalities through modeling the current states of industrial processes. To issue real-time alarms, the IIoT-based system pushes the alarm information to the relevant management and operation terminals via the mobile network. The distributed message queue middleware is adopted to deliver the alarm messages reliably in real time. All alarm messages are sent from the system to the middleware, according to the type and level of the information.

2. Materials and Methods

The materials and methods section should contain sufficient detail so that all procedures can be repeated. It may be divided into headed subsections if several methods are described.

Based on the Monte Carlo (MC) method and recursive Bayesian estimation, the PF provides a filtering method for nonlinear, non-Gaussian systems [5, 6]. During the PF, the

MC method is employed for integral operations in the Bayesian estimation according to the law of large numbers.

The basic idea of the PF algorithm is as follows: First, a set of random samples called particles are generated based on the empirical distribution of system state vectors. Then, the weight and position of the particles are updated continuously according to the measured data. Finally, the initial empirical distribution is corrected as per the updated information.

In essence, the PF algorithm approximates the probability distribution of correlations by discrete random measures, which are composed of particles and their weights, and updates the discrete random degree recursively. If the sample size is large, the MC method will be called to approach the true posteriori probability density function of the state variable.

At present, the PF algorithm has been widely applied in positioning and tracking [7–13], including but not limited to vehicle positioning, aircraft positioning through map matching or terrain aided navigation, integrated navigation, and angle measurement. Some scholars have applied the PF algorithm to single-target air-to-sea passive tracking and multitarget tracking. Other application fields of the algorithm are target recognition, system identification, parameter estimation, automatic control, and fault detection. The wide application is attributable to its excellence in recursive Bayesian estimation of nonlinear, non-Gaussian systems.

For an industrial process monitoring system, the discrete equation of state can be expressed as

$$X_k = \phi_{k,k-1} X_{k-1} + U_k + W_k, \quad (1)$$

where, $X_k = [x_{e(k)}, v_{e(k)}, a_{e(k)}, x_{n(k)}, v_{n(k)}, a_{n(k)}]^T$ is the state variable of the system; W_k is the state noise vector; U_k is the mean system manoeuvre at time k ; and $\phi_{k,k-1}$ is the system state transition matrix.

The discrete observation equation of the system can be defined as

$$Y_k = h(X_k) + V_k,$$

$$Y_k = [e_{\text{GPS}(k)}, n_{\text{GPS}(k)}, \omega_k, s_k]^T,$$

$$h(X_k) = \begin{bmatrix} x_{e(k)} \\ x_{n(k)} \\ \frac{v_{n(k)} a_{e(k)} - v_{e(k)} a_{n(k)}}{v_{e(k)}^2 + v_{n(k)}^2} \\ \xi^T \sqrt{v_{e(k)}^2 + v_{n(k)}^2} \end{bmatrix}, \quad (2)$$

$$V_k = \begin{bmatrix} v_1(k) \\ v_2(k) \\ \varepsilon_{\omega(k)} \\ \varepsilon_s(k) \end{bmatrix},$$

where Y_k is the observed vector; v_1 and v_2 are the observed noises; ε_{ω} is the drift of the gyro; ε_s is the observed error; ξ is the scaling factor; and ξ is an uncertain system parameter.

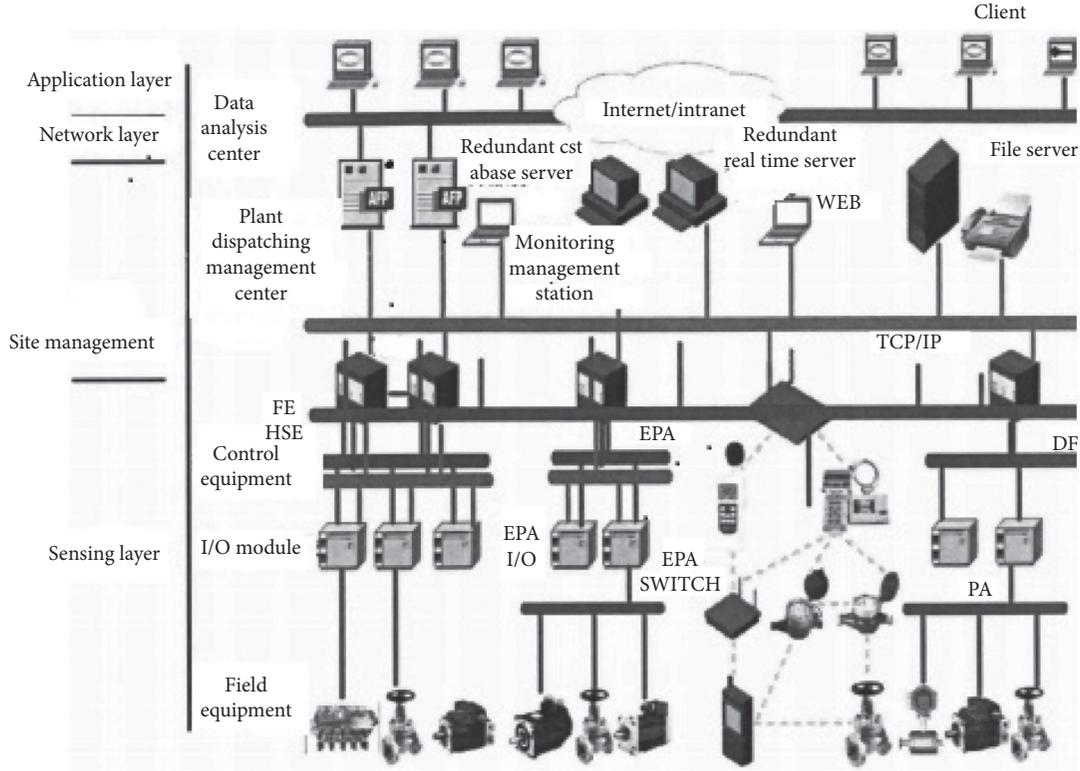


FIGURE 1: The structure of the IIoT.

The general nonlinear equation can be described as

$$x_{k+1} = f(x_k, w_k), \quad (3)$$

$$z_k = h(x_k, v_k), \quad (4)$$

where $x_k \in R^n$ is the system state vector at time $k \in T$; $f: R^n \times R^n \rightarrow R^n$ is the state evolution map; $\{w_k\}_{k \in T}$ is the process noise sequence with independent distribution; $h: R^n \times R^m \rightarrow R^m$ is the measure mapping; and $\{v_k\}_{k \in T}$ is the independent identically distributed sequence of measured noises. The initial state $x_0 \in R^n$ of the system is a random vector.

The goal of process monitoring is to estimate the state based on the measured data $z_{1:k} = \{z_1, \dots, z_k\}$. Suppose $p(x_0)$ is the priori probability distribution of the state of the given system. Then, the posteriori probability distribution $p(x_{0:k} | z_{1:k})$ of the target state x_k at time k can be described as $\{x_{0:k}^i, \omega_k^i\}_{i=1}^N$. This is a set of particles corresponding to the weight $\{\omega_k^i, i = 0, \dots, N\}$ of $\{x_{0:k}^i, i = 0, \dots, N\}$, where $x_{0:k} = \{x_j, j = 0, \dots, k\}$ is a set of states from time 0 to time k .

Normalizing the weight to $\sum_i \omega_k^i = 1$, the posteriori probability distribution of the target state at time k can be discretely weighted to

$$p(x_{0:k} | z_{1:k}) \approx \sum_{i=1}^N \omega_k^i \delta(x_{0:k} - x_{0:k}^i). \quad (5)$$

The weights are selected by importance sampling. If the set $\{x_{0:k}^i\}_{i=1}^N$ can be obtained by the importance density function $q(x_{0:k} | z_{1:k})$, then the weight can be described as

$$\omega_k^i \propto \frac{p(x_{0:k}^i | z_{1:k})}{q(x_{0:k}^i | z_{1:k})}. \quad (6)$$

If the importance density can be resolved as

$$q(x_{0:k} | z_{1:k}) = q(x_k | x_{0:k-1}, z_{1:k}) q(x_{0:k-1} | z_{1:k-1}). \quad (7)$$

Then, a new set $\{x_{0:k}^i\}_{i=1}^N$ can be obtained by the particle $\{x_k^i\}_{i=1}^N$ obtained from $q(x_k | x_{0:k-1}, z_{1:k})$, and the particle set $\{x_{0:k-1}^i\}_{i=1}^N$ can be obtained from $q(x_{0:k-1} | z_{1:k-1})$.

The posteriori probability density function can be expressed as

$$\begin{aligned} p(x_{0:k} | z_{1:k}) &= \frac{p(z_k | x_{0:k}, z_{1:k-1}) p(x_{0:k} | z_{1:k-1})}{p(z_k | z_{1:k-1})} \\ &= \frac{p(z_k | x_{0:k}, z_{1:k-1}) p(x_k | x_{0:k-1}, z_{1:k-1})}{p(z_k | z_{1:k-1})} \\ &\quad \cdot p(x_{0:k-1} | z_{1:k-1}), \\ &= \frac{p(z_k | x_k) p(x_k | x_{k-1})}{p(z_k | z_{1:k-1})} p(x_{0:k-1} | z_{1:k-1}) \\ &\propto p(z_k | x_k) p(x_k | x_{k-1}) p(x_{0:k-1} | z_{1:k-1}). \end{aligned} \quad (8)$$

Substituting (5) and (6) into (7), the importance weight update formula can be obtained as

$$\begin{aligned}\omega_k^i &= \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i) p(x_{0:k-1}^i | z_{1:k-1})}{q(x_k^i | x_{0:k-1}^i, z_{1:k}) q(x_{0:k-1}^i | z_{1:k-1})} \\ &= \omega_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{0:k-1}^i, z_{1:k})}.\end{aligned}\quad (9)$$

If $q(x_k | x_{0:k-1}, z_{1:k}) = q(x_k | x_{k-1}, z_k)$, then the importance density function depends only on x_{k-1} and z_k . In this case, only $\{x_k^i\}_{i=1}^N$ needs to be stored in the calculation, regardless of the set $\{x_{0:k-1}^i\}_{i=1}^N$ and the past measurement $z_{1:k-1}$. The modified weight can be expressed as

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)}.\quad (10)$$

The standard PF algorithm chooses the easiest achievable priori probability density as the importance density function:

$$q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i).\quad (11)$$

Replacing (11) with (10), the importance weight can be reduced to

$$\omega_k^i \propto \omega_{k-1}^i p(z_k | x_k^i).\quad (12)$$

Then, the weight ω_k^i can be normalized:

$$\omega_k^i = \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^i}.\quad (13)$$

The posteriori probability density $p(x_k | z_{1:k})$ can be expressed as

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^N \omega_k^i \delta(x_k - x_k^i).\quad (14)$$

It can be seen that, when $N \rightarrow \infty$, the abovementioned formula can approximate the real posteriori probability $p(x_k | z_{1:k})$ by the law of large numbers.

The pseudocode of the standard PF algorithm is given below (Algorithm 1).

The PF algorithm was improved based on the SOSS model to control the industrial processes based on the IIoT. To begin with, the vector of the unknown parameters in the IIoT-based process control system is denoted as θ and extended to the state vector, such that $Z_k = [X_k, \theta_k]^T$. Let Y_k be a stochastic noise. If the stochastic step-size model $\theta_k = \theta_{k-1} + \gamma_k$ is applied, then the state space of the process control system can be modelled as

$$Z_k = \begin{bmatrix} \varphi_{k,k-1} & 0 \\ 0 & 1 \end{bmatrix} Z_{k-1} + \begin{bmatrix} U \\ 0 \end{bmatrix} + \begin{bmatrix} W_k \\ \gamma_k \end{bmatrix},\quad (15)$$

$$Y_k = H(Z_k) + V_k.$$

We suppose the initial set θ_0^i , $\{i = 1, \dots, N\}$ of the parameter vector θ is sampled from the uniform distribution of (3):

$$\theta_{j,0}^i - U(P_j - r_j, P_j + r_j),\quad (16)$$

where $\theta_{j,0}^i$ is the j -th component of θ_0^i ; $U(P_j - r_j, P_j + r_j)$ is the uniform distribution from $P_j - r_j$ to $P_j + r_j$; P_j is the centre of $(P_j - r_j, P_j + r_j)$; and r_j is a real number chosen by the user according to the range of parameter changes.

Given the priori condition probability $\rho(X_0)$ of the system state, $\{Z_k^i, w_k^i\}_{i=1}^N = (\{X_k^i, \theta_k^i\}_{i=1}^N, \{w_k^i\}_{i=1}^N)$ can be used to describe the posteriori probability distribution $\rho(Z_k | Y_k)$ of the target state Z_k at time k , and $\{Z_k^i, i = 1, \dots, N\}$ can be used to represent the particle set that corresponds to the weight $\{w_k^i, i = 1, \dots, N\}$.

Normalizing the weight to $\sum_{i=1}^N w_k^i = 1$, the pseudocode of the SOSS-based PF algorithm can be programmed as follows (Algorithm 2).

In normal cases, the SOSS-based PF algorithm can effectively filter unknown parameters of the IIoT-based process control system. However, the performance of the algorithm depends heavily on the sampling quality, which is a subproblem of initial sampling. Unfortunately, the initial distribution of unknown parameters is not easy to determine. If the initial samples only have a few values in the concentration zone of posteriori distribution, it would be difficult for the SOSS-based PF algorithm to filter the unknown parameters well.

To solve the problem, the AFSA [14–20], a stochastic search optimization algorithm, was introduced to improve the SOSS-based PF algorithm. Mimicking the behaviors of fish swarms in nature, the AFSA includes four adaptive behaviors: foraging, clustering, chasing, and random walk. This algorithm has been successfully applied in many engineering fields. The AFSA-based optimization of the SOSS-based PF algorithm is implemented as follows.

First, the search space Ω of unknown parameters was divided into several equal-size subspaces $\Omega^1, \Omega^2, \dots, \Omega^M$, each of which is defined as an artificial fish (AF). The AFs simulate the random search process of the fish in nature. As shown in Figure 2, the relationship between each subspace and the original search space can be represented as $\Omega^1 \cup \dots \cup \Omega^M \subseteq \Omega$. Then, the optimal distribution space of unknown parameters was obtained.

Through the abovementioned process, the SOSS-based AFSA-PF algorithm was obtained. This global search algorithm can effectively estimate system states and derive a realistic distribution of unknown parameters. The pseudocode of the SOSS-based AFSA-PF algorithm is given below (Algorithm 3).

```

[ $\{x_k^i, \omega_k^i\}_{i=1}^N$ ] = SIS[ $\{x_{k-1}^i, \omega_{k-1}^i\}_{i=1}^N, z_k$ ]
{
A particle  $\{X_0^i\}_{i=1}^N$  is produced by a priori probability  $p(x_0)$ , and the total weight of all particles is  $1/N$ , making  $k = 1$ .
For  $k = 1, \dots, T$ 
For  $i = 1, \dots, N$ 
Prediction: Use equation of State (1) to predict the unknown state  $x_k^i$ .
Calculate particle weights:
 $\omega_k^i = \omega_{k-1}^i P(z_k | x_k) = \omega_{k-1}^i \rho_{ek}(z_k - h(x_k^i)), i = 1, \dots, N$ 
End For
Normalized weight,
For  $i = 1, \dots, N$ 
 $\omega_k^i = \omega_k^i / \sum_{i=1}^N \omega_k^i$ .
End For
Then, the minimum mean square estimate of the state vector at time  $k$  is obtained as  $\hat{x}_k \approx \sum_{i=1}^N \omega_k^i x_k^i$ . The priori condition probability density of system states is chosen as the important density function  $\rho_{ek}$ .
Resampling: A new set of particles  $\{x_k^i\}_{i=1}^N$  is obtained.
End For
}

```

ALGORITHM 1: The standard PF algorithm.

```

[P] = SOSS( $\{X_0^i\}_{i=1}^N, \{\theta_0^i\}_{i=1}^N, \Omega$ )
{
A particle set  $\theta_0^i \sim U(P - r, P + r), (i = 1, \dots, N)$ , is initialized, and a particle  $\{X_0^i\}_{i=1}^N$  is produced by a priori probability  $\rho(X_0)$ , with  $\{Z_0^i\}_{i=1}^N = (\{X_0^i\}_{i=1}^N, \{\theta_0^i\}_{i=1}^N)$ , and all particles have a weight of  $1/N, k = 1$ .
For  $k = 1, \dots, T$ 
For  $i = 1, \dots, N$ 
Prediction: Use equation of state to predict the unknown state  $Z_k^i$ 
Calculate particle weights:
 $w_k^i = w_{k-1}^i \rho(Y_k | Z_k) = w_{k-1}^i \rho_{ek}(Y_k - H(Z_k^i)), i = 1, \dots, N$ 
End For
Normalized weight,
For  $i = 1, \dots, N$ 
 $w_k^i = w_k^i / \sum_{i=1}^N w_k^i$ .
End For
Then, the minimum mean square estimate of the state vector at time  $k$  is obtained as  $\hat{Z}_k \approx \sum_{i=1}^N w_k^i Z_k^i$ . The priori condition probability density of system states is chosen as the important density function  $\rho_{ek}$ .
Resampling:  $\{Z_k^i\}_{i=1}^N = \text{resample}(\{X_k^i\}_{i=1}^N, \{\theta_k^i\}_{i=1}^N)$ 
End For
}

```

ALGORITHM 2: The SOSS-based PF algorithm.

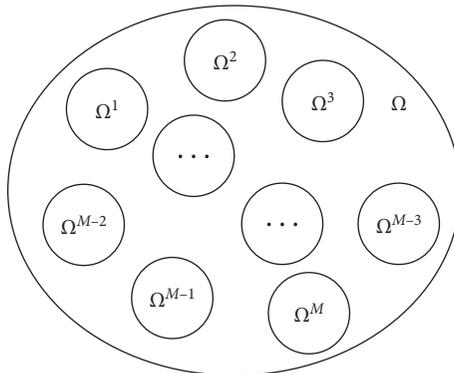


FIGURE 2: The relationship between the original space and subspaces.

```

[P] = ESSOSS({X0i}_{i=1}^N, {θ0i}_{i=1}^N, Ω)
{
For each subset Ωq, (q = 1, ..., M), initialize the particle set θ0q,i ~ U(cq - d, cq + d). Particle {X0i}_{i=1}^N is produced by a Priori
Probability ρ(X0). {Z0q,i}_{i=1}^N = ({X0i}_{i=1}^N, {θ0q,i}_{i=1}^N, Ωq). All the particles have a weight of 1/N, let k = 1.
For k = 1, ..., T
For Ωq = Ω1, ..., ΩM
For i = 1, ..., N
PREDICTION: Using Equation of state (17) to predict unknown states Zkq,i.
Calculate particle weights:
wki = wk-1iρ(Yk | Zk) = wk-1iρek(Yk - H(Zkq,i)), i = 1, ..., N
End For
Normalized weight,
For i = 1, ..., N
wki = wki / ∑i=1N wki.
End For

```

ALGORITHM 3: The SOSS-based AFSA-PF algorithm.

3. Results

The results and discussion may be presented separately, or in one combined section, and may optionally be divided into headed subsections.

To verify its adaptability, the SOSS-based AFSA-PF algorithm was applied to estimate the state and identify the parameters of an IIoT-based process control system with unknown parameters. During the simulation, the error level

of each sensor was adjusted: the observed noise was changed to $\varepsilon_{\omega} \sim N(0, 0.04^2)$, and that of the odometer was kept at $\varepsilon_s \sim t(1)$. The initial state X_0 and initial value P_0 of the error variance matrix of the system were also kept stable. The SOSS-based AFSA-PF algorithm was first adopted to identify the unknown parameters δ_1 , δ_2 , and ξ (Figure 3).

The system states can be described by the following nonlinear mathematical model:

$$\begin{aligned}
 \dot{x}_1(t) &= x_2(t), \\
 \dot{x}_2(t) &= \frac{9.6 \cos(x_1(t)) - \sin(x_1(t))(au(t) + bx_2^2(t)\sin(x_1(t)) - acx_4(t))}{d - b \cos^2(x_1(t))}, \\
 \dot{x}_3(t) &= x_4(t), \\
 \dot{x}_4(t) &= au(t) + b(x_2^2(t)\sin(x_1(t)) - \dot{x}_2(t)\cos(x_1(t))) - acx_4(t),
 \end{aligned} \tag{17}$$

where $x_1(t)$, $x_2(t)$, $x_3(t)$, and $x_4(t)$ are the system state; $u(t)$ is the control quantity of the system; and $a=0.6$, $b=0.020$, $c=0.1$, and $d=0.1865$ are the parameter variables of the system.

In each subset, the particle θ^i , $\{i=1, \dots, N\}$ of the unknown parameter θ is assumed to satisfy uniformly distributed sampling, that is, $\theta^i \sim U(c^q - d, c^q + d)$.

Based on the abovementioned artificial intelligence (AI) model, the proposed algorithm can be implemented in the following steps.

When the unknown parameter belongs to space Ω^q , the least mean square estimate of the state vector at time k can be given as

$$\hat{Z}_k^q \approx \sum_{i=1}^N w_k^i Z_k^{q,i}. \tag{18}$$

The objective function can be expressed as

$$\Psi = \rho_{ek}(Y_k - H(\hat{Z}_k^q)), \tag{19}$$

where ρ_{ek} is the priori condition probability density of system states. The estimated value of unknown parameter θ_k^q in vector \hat{Z}_k^q can be adopted as the new centre of space Ω^q , i.e., $c^q = \theta_k^q$.

3.1. Foraging. Let c^q ($v_{q,l} < \text{visual}$) be the current state of an AFs. Then, the AF chooses a state c^l in its perception range. If $\psi^q < \psi^l$, the AF will move one step in that direction; otherwise, the AF will move one step in a random direction:

$$\left[\begin{array}{l} \text{If } \psi^q < \psi^l, \quad c_k^{q\text{next}} = c^q + \text{rand}(\cdot) \text{Step} \frac{c_k^l - c_k^q}{\|c_k^l - c_k^q\|}, \\ \text{Else,} \quad c_k^{q\text{next}} = c^q + \text{rand}(\cdot) \text{Step}, \end{array} \right] \tag{20}$$

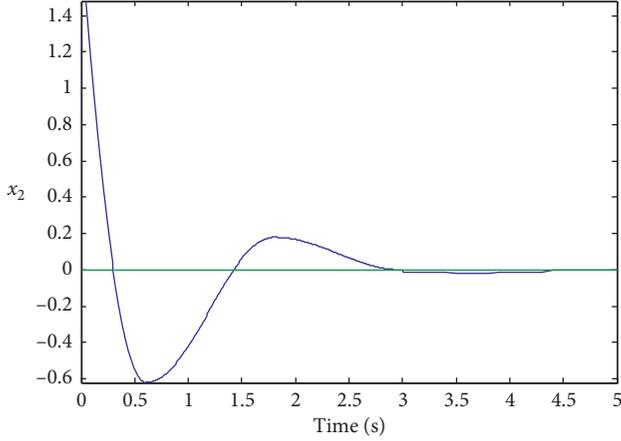


FIGURE 3: The unknown parameters identified by the SOSS-based AFSA-PF algorithm.

where Step is the maximum step length of the AF; $\text{rand}(\cdot)$ is a random number between 0 and 1.

3.2. Clustering. Let c^q be the current state of an AF and n_f be the number of AFs in other clusters in the perception range of the AF. Then, the centre of the neighbourhood of the AF can be described as $c^c = 1/n_f \sum_{l \neq q}^{n_f} c^l$. Then, it is possible to obtain the new unknown parameter Range $\theta^c - U(c^c - d, c^c + d)$, and its distribution $\{Z_{k-1}^{c,i}\}_{i=1}^N = \text{sample}(\{X_{k-1}^{c,i}\}_{i=1}^N, \{\theta_{k-1}^{c,i}\}_{i=1}^N, \Omega^c)$. Through particle weight prediction and calculation, we have $\tilde{Z}_k^c \approx \sum_{i=1}^N w_k^i Z_k^i$, and thus, $\psi^c = \rho_{ek}(Y_k - H(\tilde{Z}_k^c))$. If $\psi^c/n_f > \delta \cdot \psi^q$ (δ is the crowding factor), the centre of the sampling space Ω^q for the particle set θ^i in the next time ($\theta_k^{q,i} - U(c^{q\text{next}} - d, c^{q\text{next}} + d)$) can be calculated by

$$c_k^{q\text{next}} = c_q + \text{rand}(\cdot) \text{Step} \frac{c_k^c - c_c^q}{\|c_k^c - c_c^q\|}. \quad (21)$$

Otherwise, the AF will move as it does in the foraging process.

3.3. Bulletin Board. In our algorithm, a bulletin board is set up to record the optimal value of the current objective function. After each move, the AF compares its objective function Ω^q with the objective function Ω^o of the bulletin board. If the former is better, the objective function Ω^o of the bulletin board will be replaced with the objective function value of the AF. The state vector of the optimal value is regarded as the optimal value of \tilde{Z}_k .

Resampling:

$$\{Z_k^{q,i}\}_{i=1}^N = \text{resample}(\{X_k^i\}_{i=1}^N, \{\theta_k^i\}_{i=1}^N, \Omega^q)$$

End For

End For

The local linearization of the abovementioned nonlinear system is equivalent to the model of the original nonlinear system:

Rule 1 : If $x_1(t)$ is about 0

$$\dot{x}(t) = (A_1 + D\Delta E_{a1})x(t) + (B_1 + D\Delta E_{b1})u(t)$$

Rule 2 : If $x_1(t)$ is about ± 1.57

$$\dot{x}(t) = (A_2 + D\Delta E_{a2})x(t) + (B_2 + D\Delta E_{b2})u(t)$$

The parameters of the correlation matrix are as follows:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 30.174 & 0 & 0 & 0.2326 \\ 0 & 0 & 0 & 1 \\ -0.5957 & 0 & 0 & -0.0842 \end{bmatrix}, \\ B_1 &= \begin{bmatrix} 0 \\ -2.1268 \\ 0 \\ 0.7533 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 20.1702 & 0 & 0 & 0.1709 \\ 0 & 0 & 0 & 1 \\ 0.0165 & 0 & 0 & -0.051 \end{bmatrix}, \\ B_2 &= \begin{bmatrix} 0 \\ -1.6278 \\ 0 \\ 0.6214 \end{bmatrix}, \\ E_{a1} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 6 \end{bmatrix}, \\ E_{b1} &= \begin{bmatrix} 0 \\ 4 \\ 0 \\ 0 \end{bmatrix}, \\ E_{a2} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 6 \end{bmatrix}, \\ E_{b2} &= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \\ D &= 0.01 \times I, \end{aligned} \quad (22)$$

where I is a fourth-order identity matrix.

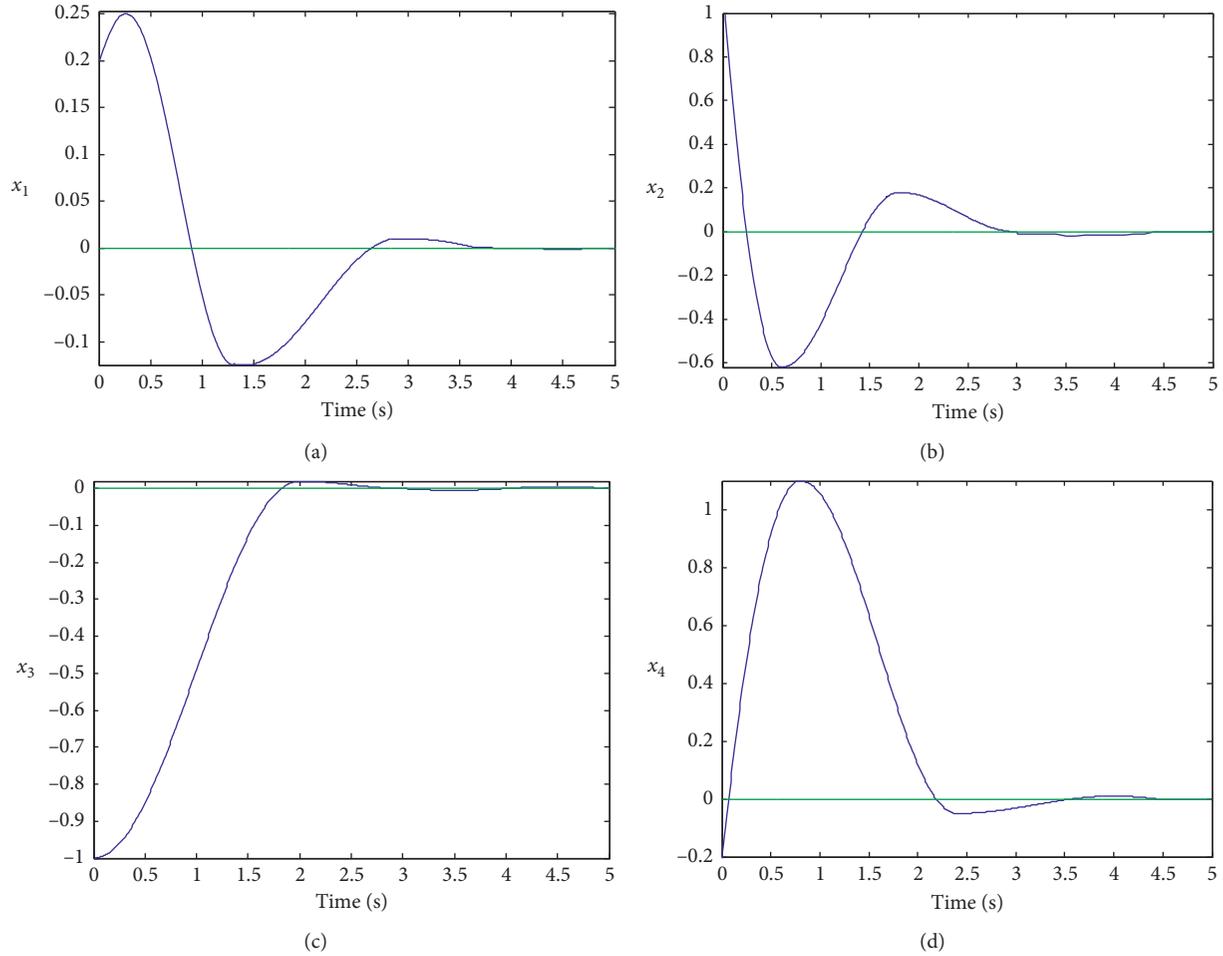


FIGURE 4: The actual state response of the system.

Assuming that the initial state is $x(0) = [0.1, 1, -1, -0.1]^T$, the weight matrix $Q = \text{diag}\{2, 0.02, 2, 0.02\}$ of system performance indices and $R = 0.02$ can be established. Finally, the state feedback matrix of the controller can be obtained as

$$\begin{aligned} F_1 &= [332.1445 \quad 89.3544 \quad 76.3622 \quad 145.9341], \\ F_2 &= [50.5252 \quad 163.3102 \quad 145.1502 \quad 136.9229]. \end{aligned} \quad (23)$$

The upper bound of the corresponding system performance index can be obtained as

$$J^* = x_0^T P x_0 = 22.3245. \quad (24)$$

The designed controller was applied to the model of the original nonlinear system, and the simulation time was set to 5 s. The simulation results are displayed in Figures 4 and 5, where the blue solid lines and green dotted line stand for the actual state and control quantity of the system, respectively.

To verify its antijamming performance, the designed controller was simulated after the state vectors $x_1(t)$ and $x_3(t)$ of the original system were added pulse jamming signals at 1.28 s and 2.79 s, respectively. The simulated results on state variables and control variables are presented in Figures 6 and 7, respectively.

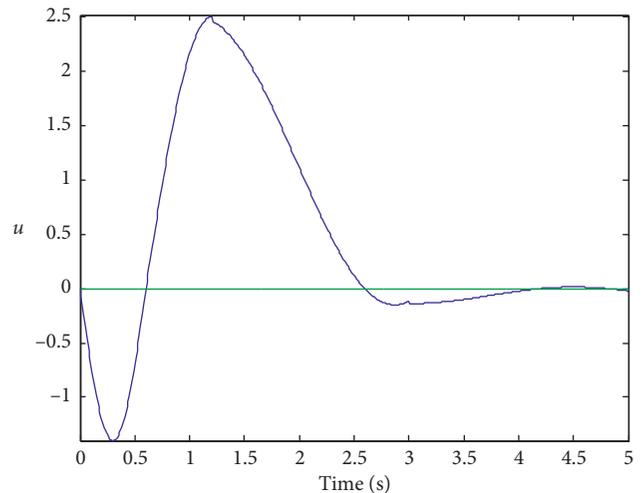


FIGURE 5: The controlled state response of the system.

To verify its robustness to variable system parameters, the designed controller was further simulated after changing system parameters from $b = 0.036$, $c = 0.1$, and $d = 0.365$. The simulated results on state variables and control variables are presented in Figures 8 and 9, respectively.

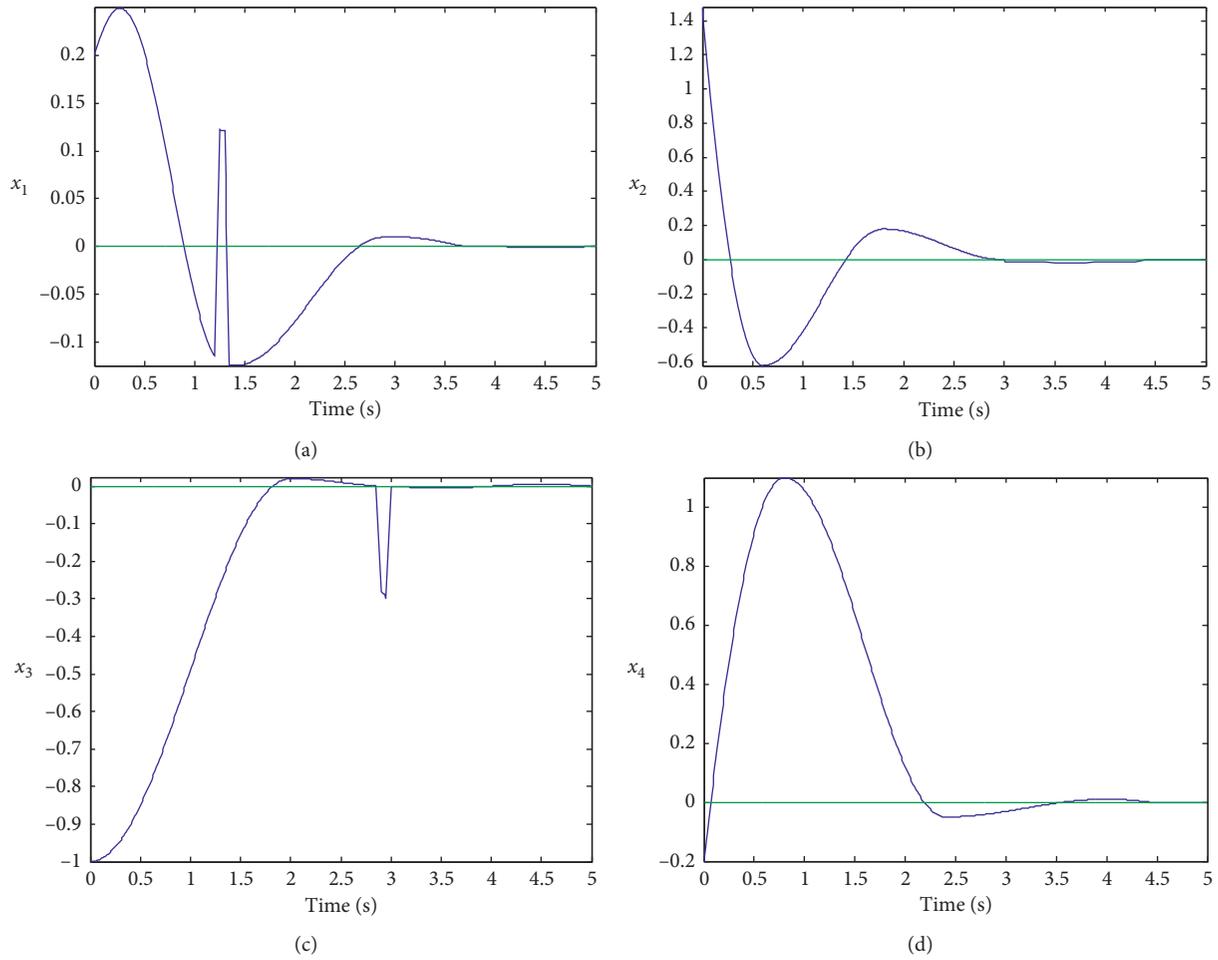


FIGURE 6: The actual state response of the system under jamming signals.

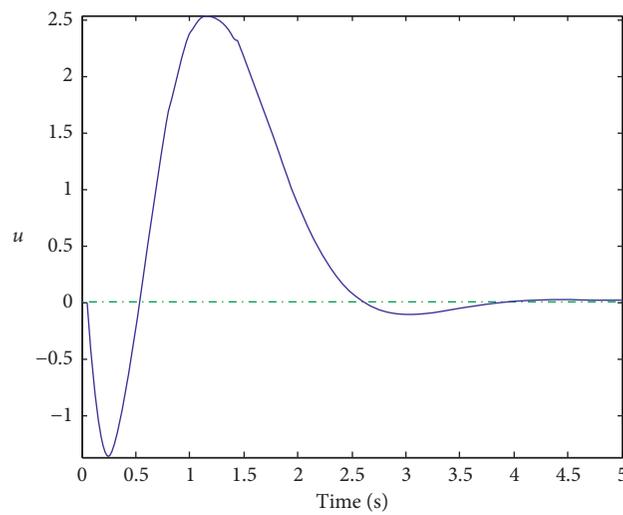


FIGURE 7: The controlled state response of the system under jamming signals.

From the abovementioned simulation results, the designed controller was found to keep the nonlinear system with unknown parameters stable and bring the system back to

equilibrium state, despite the addition of jamming signals and the variation in system parameters. Changing the error level of sensors could affect the system error and also influence the

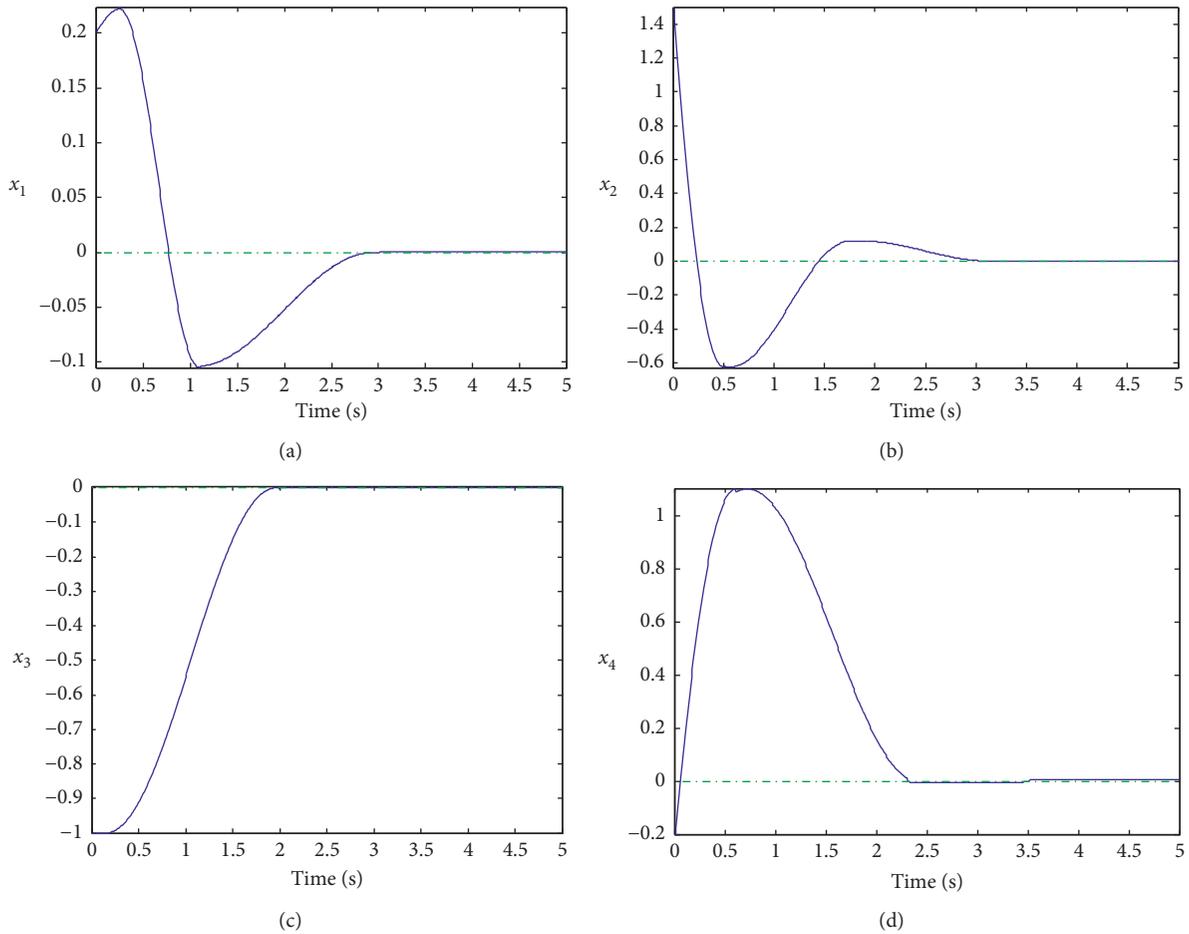


FIGURE 8: The actual state response of the system at $b = 0.035$, $c = 0.1$, and $d = 0.365$.

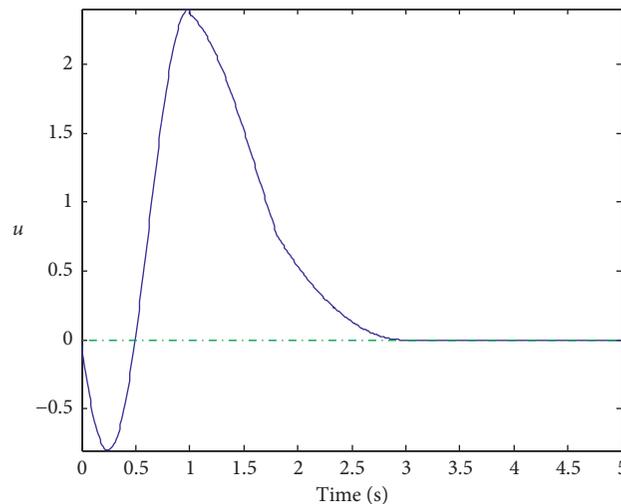


FIGURE 9: The controlled state response of the system at $b = 0.038$, $c = 0.1$, and $d = 0.364$.

identification of unknown parameters. The stronger the sensor noise, the greater the system error. However, the influence on unknown parameters is way smaller than the location error. As a result, the proposed algorithm could accurately estimate the estimation parameters.

4. Conclusions

The existing filtering algorithms for the IIoT-based process control system only work if the system model is deterministic. In real-world scenarios, it is very difficult to find an

IIoT-based process control system that is truly deterministic. The model parameters of the system are often unknown, though their ranges might be identifiable. To eliminate the effect of the unknown parameters on the IIoT-based control of industrial processes, this paper develops a novel AFSA-PF algorithm based on the SOSS model. The SOSS-based AFSA-PF algorithm can effectively estimate system states, approximate the actual distribution of unknown parameters, and disclose the true value of the unknown parameters. With the aid of the proposed algorithm, the system model could reflect the actual states of the IIoT-based process control system. The research findings provide a good reference for applying AI techniques in industrial process control.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by High-Level Talents Foundation of Ankang University (Grant No. 2019AYQDZR01) and Ankang Science and Technology Research and Development Project (Grant No. AK2019SF-09).

References

- [1] R. Fedele, M. Merenda, F. Giammaria, R. Carotenuto, and F. G. D. Corte, "Energy harvesting for IoT road monitoring systems," *Instrumentation Mesure Métrologie*, vol. 18, no. 4, pp. 605–623, 2018.
- [2] A. Rahaman, M. Islam, M. Islam, M. Sadi, and S. Nooruddin, "Developing IoT based smart health monitoring systems: a review," *Revue d'Intelligence Artificielle*, vol. 33, no. 6, pp. 435–440, 2019.
- [3] D. Krčmařík, M. Petrů, and R. Moezzi, "Innovative IoT sensing and communication unit in agriculture," *European Journal of Electrical Engineering*, vol. 21, no. 3, pp. 273–278, 2019.
- [4] W.-T. Sung and Y.-C. Hsu, "Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4522–4529, 2011.
- [5] N. G. Polson, J. R. Stroud, and P. Müller, "Practical filtering with sequential parameter learning," University of Pennsylvania Working Paper, Philadelphia, PA, USA, 2006.
- [6] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [7] A. Ataei and Q. Wang, "Non-linear control of an uncertain hypersonic aircraft model using robust sum-of-squares method," *IET Control Theory & Applications*, vol. 6, no. 2, pp. 203–215, 2012.
- [8] J. Chen, C. P. Pan, and H. C. Zhao, "One near space hypersonic aircraft attack angle observer and controller design," in *Proceedings of the 2011 Chinese Control and Decision Conference (CCDC)*, Mianyang, China, May 2011.
- [9] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [10] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.
- [11] H. H. Fu, J. Xu, H. Zhang, M. Zhang, and X. X. Xu, "A novel video target tracking method based on lie group manifold," *Traitement du Signal*, vol. 35, no. 3-4, pp. 331–340, 2018.
- [12] G. Kitagawa, "Theory and methods," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1203–1215, 1998.
- [13] M. Hürzeler and H. R. Künsch, "Approximating and maximising the likelihood for a general state-space model," *Sequential Monte Carlo Methods in Practice*, Springer, New York, NY, USA, 2001.
- [14] K. Yano, "A self-organizing state space model and simplex initial distribution search," *Computational Statistics*, vol. 23, no. 2, pp. 197–216, 2008.
- [15] M. Jiang and K. Zhu, "Multiobjective optimization by artificial fish swarm algorithm," in *Proceedings of the 2011 IEEE International Conference on Computer Science and Automation Engineering*, Shanghai, China, June 2011.
- [16] D. Li, F. Qian, and J. Gao, "Performance-first control for discrete-time LQG problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2225–2230, 2009.
- [17] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, 2003.
- [18] D. Zhang, J. Tan, H. Tian, Z. Wang, and W. Guo, "Aquifer parameter inversion by artificial fish swarm algorithm based on quantum theory," *Ingénierie des Systèmes d'Information*, vol. 24, no. 1, pp. 29–33, 2019.
- [19] P. Guo, X. Wang, and Y. Han, "The enhanced genetic algorithms for the optimization design," in *Proceedings of the 2010 3rd International Conference on Biomedical Engineering and Informatics*, Yantai, China, October 2010.
- [20] C.-T. Cheng, K. Fallahi, H. Leung, and C. K. Tse, "A genetic algorithm-inspired UUV path planner based on dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1128–1134, 2012.