Journal of Control Science and Engineering

Fault Diagnosis and Application to Modern Systems

Guest Editors: Xiao He, Zidong Wang, Gang Li, Zhijie Zhou, and Youqing Wang

Fault Diagnosis and Application to Modern Systems

Journal of Control Science and Engineering

Fault Diagnosis and Application to Modern Systems

Guest Editors: Xiao He, Zidong Wang, Gang Li, Zhijie Zhou, and Youqing Wang

Editorial Board

Contents

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 1026759, 3 pages
https://doi.org/10.1155/2017/1026759

Editorial

Fault Diagnosis and Application to Modern Systems

Xiao He,[1] Zidong Wang,[2] Gang Li,[3] Zhijie Zhou,[4] and Youqing Wang[5]

[1]*Department of Automation, TNList, Tsinghua University, Beijing 100084, China*
[2]*Department of Computer Science, Brunel University London, London UB8 3PH, UK*
[3]*School of Instrument Science and Opto-Electronics Engineering, Beihang University, Beijing 100191, China*
[4]*Xi'an High-Tech Institute, Xi'an 710025, China*
[5]*College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China*

Correspondence should be addressed to Xiao He; hexiao@tsinghua.edu.cn

Received 27 April 2017; Accepted 27 April 2017; Published 24 May 2017

With the increasing demand for higher performance and higher safety and reliability standards, fault diagnosis (FD) for modern control systems has become an active field of research over the past decades. FD is a theory and technology that can utilize the analytical redundancy relationship of the control system and provide the whole system with accurate information of the fault that occurred by using the input and output of the system. Both theoretical challenge and practical application demands of many kinds of real-time control systems motivate the investigation of FD. A typical FD process consists of a fault detection unit, a fault isolation strategy, and a fault estimation technique. The task of fault detection problem is to construct a residual signal which is then compared with a predefined threshold. When the residual exceeds the threshold, an alarm is generated. The scope of the fault isolation problem is to locate the true fault from all possible faults. The objective of the fault estimation problem is to determine the fault amplitude as well as the emergence time of the fault. The precise understanding of the fault that occurred by using FD technology is a prerequisite for the next fault accommodation process and thus is very important for system safety.

The existing FD techniques can be generally divided into two main categories: the model-based method and the data-driven approach. In the case where a mathematical model can be obtained for an objective system, a model-based FD method can provide exact decoupling or maximum attenuation to the factors except the target fault. On the other hand, when it is hard to obtain a mathematical model but enough historical data of a system can be obtained, the data-driven approach is more applicable and may get a better FD result. Recently, analytical-based FD techniques for modern systems have received more and more attention and there have been an increasing number of results reported in the literature for the topics of FD. This special issue aims to bring together some typical recent developments in fault diagnosis technology and its application to modern systems. We have solicited submissions to this special issue from scholars from all over the world and have got 26 submissions. After rigorous peer-review processes, 9 papers have been selected to be published in this special issue.

Some works in this special issue focus on the model-based FD and Fault-Tolerant Control (FTC) techniques. In the paper entitled "Detection of Intermittent Fault for Discrete-Time Systems with Output Dead-Zone: A Variant Tobit Kalman Filtering Approach" by J. Huang and X. He, a fault detection method for discrete-time systems with output dead-zone is proposed. Most of the present research on fault diagnosis focuses on input dead-zone instead of output dead-zone, where the dead-zone is usually treated as an unknown input. The Tobit Kalman filtering approach is employed in the design of the residual generator. Compared with the traditional Kalman filter, the Tobit Kalman filter avoids the estimation bias that is brought about by ignoring the correlation between the state and the measurement noise in the dead-zone. Due to the obvious discontinuity occurring in thresholds of the dead-zone, the gradient does not exist, which makes the Extended Kalman filter hard to apply. Also, when the output dead-zone is located between the sigma points, there will be a bias in the measurement noise covariance that causes a bias in the estimation using Unscented Kalman filter. By contrast, the above conditions

can be averted by utilizing the Tobit Kalman filter that provides the optimal estimation with minimum variance when the output dead-zone appears. Even though the particle filter can give an accurate estimation in the output dead-zone, the Tobit Kalman filter has similar computational expenses to the traditional Kalman filter to avoid the collapse that may well happen in applying a particle filter in the high dimensional systems. Therefore, the fault detection method proposed in this paper has higher practicability in modern control systems, especially the ones with limited computational power, like embedded systems. Meanwhile, the Tobit Kalman filter performs well in tracking so that the residual generator in this paper is capable of quickly and accurately detecting the appearance and disappearance of faults.

In the paper entitled "Fault-Tolerant Control of a Nonlinear System Actuator Fault Based on Sliding Mode Control" by J. He et al., a fault-tolerant control scheme is presented for a class of nonlinear systems with actuator faults and unknown input disturbances. The sliding mode control law is designed based on the reaching law method. Then, in view of the unpredictable state variables and unknown information in the control law, the original system is transformed into two subsystems through a coordinate transformation. One subsystem only has actuator faults, and the other subsystem has both actuator faults and disturbances. A sliding mode observer is designed for the two subsystems, respectively, and the equivalence principle of the sliding mode variable structure is used to realize the accurate reconstruction of the actuator faults and disturbances. The observation value and the reconstruction value are used to carry out an online adjustment to the designed sliding mode control law, and fault-tolerant control of the system is realized. Simulation results show the effectiveness of the proposed method.

Some works in this special issue pay attention to data-driven FD. Traditional data-driven fault diagnosis techniques are not applicable in the case with big data generated from the modern complicated industry process. In the paper entitled "A Novel Multimode Fault Classification Method Based on Deep Learning" by F. Zhou et al., a fault diagnosis method based on a hierarchical deep neural network is proposed. The hierarchical deep neural network consists of three hierarchies (each one is a traditional unsupervised multilayer neural network), which are utilized to calculate the running mode, to identify the fault component, and to classify the fault level of the research object, respectively. This method guarantees the accuracy of fault diagnosis. The proposed neural network has been applied to real sampled data generated by the Case Western Reserve University Bearing Data Center and the results demonstrate its efficiency compared with those of the hierarchical back propagation neural network, the support vector machine, and the deep neural network. Furthermore, the proposed algorithm is robust for its ability to handle both the big data and the sample in small scale, which overmatches the deep neural network for data in big scale.

Due to the complexity of the industrial process, the fault data usually has time-correlated characteristic and its distribution law is hard to obtain. This renders the traditional back propagation single-hidden-layer neural network unable to guarantee the accuracy of the fault classification. To solve this problem, in the paper entitled "Research on Fault Diagnosis Method Based on Rule Base Neural Network" by Z. Ni et al., a neural network fault diagnosis method based on rule base is proposed. It utilizes the known factors causing the faults to regularize the experts' experience by fuzzy logic and applies its rule base to the multi-hidden-layer neural network. This treatment can make full use of a priori knowledge, so it can guarantee the accuracy of neural network based fault diagnosis techniques. The proposed algorithm is robust to the size of the test data and is also applicable in the case where only data in small scale can be obtained. It can extract the effective samples without the assumption of the large-scale samples with identical distribution. The proposed algorithm not only is able to isolate a single fault but also can hold composite faults. By applying fuzzy logic, uncertain information is processed effectually. Thus, compared with other neural network based fault diagnosis approaches, the proposed method in this paper can obtain a more precise fault rule base and a higher classification accuracy.

As a widely used computing service method for large-scale resource scheduling, cloud computing has gained momentum recently. In order to assess the reliability of cloud computing accurately and effectively, in the paper entitled "Reliability Assessment of Cloud Computing Platform Based on Semiquantitative Information and Evidential Reasoning" by H. Wei and P.-L. Qiao, a novel assessment approach for the cloud computing platform is presented. A new model of the reliability assessment based on ER rule is proposed, which can combine both of the qualitative knowledge and the quantitative data. In particular, the quantitative data include the Mean Time to Failures (MTTF), frequency of network attack, fault-tolerant rate of cloud computing platform, and Mean Time to Repair (MTTR) of service, and qualitative knowledge is associated with the subjective quality of a situation or phenomenon, such as scalability of cloud system, controllability of access terminal, completeness of service, and stability of service. A four-level reliability attributes structure of cloud computing platform is established, including "excellent," "good," "common," and "bad." In addition, the reliability of the cloud computing platform is categorized in four distinct aspects, that is, network reliability, hardware reliability, software reliability, and service reliability. A case study for assessing the reliability of an actual cloud computing platform is given, and its good performance on assessing the reliability of a real cloud computing platform sheds light on the effectiveness and rationality of the proposed method.

Considering that the partial least squares (PLS) model is based on the assumption of a single operating mode, in the paper entitled "An Efficient Quality-Related Fault Diagnosis Method for Real-Time Multimode Industrial Process" by K. Peng et al., a novel multimode PLS model is developed using the well-known Gaussian mixture model and PLS quality-related process monitoring problem is investigated for the multimode process based on finite Gaussian mixture models and PLS method. The advantage of original PLS is subsequently followed to achieve the quality-related monitoring goals. A Gaussian mixture model is implemented for training data to conduct the multimode division and estimate

the model parameter set. Then, posterior probabilities of each monitored sample belonging to all Gaussian modes are calculated through the Bayesian inference strategy. Based on the posterior probabilities, a comprehensive statistics index is defined and a combined index was developed for the fault detection purpose. Finally, a new index named gradient contribution rate is proposed to measure the contribution to the combined index and find out the variable that should be in charge of fault in quality. Performances of the whole proposed scheme are verified in a real industrial hot strip mill process compared with some existing methods.

In practical industry applications, the collected process variables always have different units. In order to obtain a consistent result, traditional Principal Component Analysis (PCA) methods need to normalize the collected data to zero mean and unit variance first, which actually neglects some useful information between different variables. To shorten such a gap, in the paper entitled "Fault Diagnosis Method Based on Information Entropy and Relative Principal Component Analysis" by X. Xu and C. Wen, relative-transformation PCA is implemented based on information entropy instead of prior information to form a new fault diagnosis method called information-entropy relative-transformation PCA. The algorithm calculates the information entropy for each characteristic variable in the original dataset based on the information gain algorithm. According to the information entropy, it allocates the weight for each standardized characteristic variable. Relative-transformation PCA is utilized to obtain the features for fault diagnosis. Simulation experiments based on Tennessee Eastman process and Wine datasets demonstrate the feasibility and effectiveness of the new method.

Some works focus on the application of FD techniques for some specific real-time systems. For example, in the paper entitled "Neural Network Based Fault Detection and Diagnosis System for Three-Phase Inverter in Variable Speed Drive with Induction Motor" by F. Asghar et al., the fault detection and isolation problem is considered for three-phase inverters based on features extraction and neural networks. Six different single switching device open faults and six different double switching device open faults are considered. In order to present decorrelation between different kinds of faults, Clarke transformation is used to extract the main features of output currents of a three-phase inverter. Then, the extracted features are used to diagnose faults by using artificial neural networks. The proposed method is able to detect and isolate both single faults and multiple faults. Compared with the previous approaches, more accurate fault diagnosis results can be obtained by using the proposed method due to the application of feature extraction. Finally, simulations and experimental results demonstrate the effectiveness of the fault diagnosis scheme. The main novelty of the paper lies in the application of the Clarke transformation and the conduction of the experiment.

In the paper entitled "Intelligent Vehicle Embedded Sensors Fault Detection and Isolation Using Analytical Redundancy and Nonlinear Transformations" by N. Pous, a fault detection and isolation scheme is proposed with intelligent vehicle embedded sensors based on analytical redundancy and nonlinear transformations. Four kinds of faults, that is, additive and stuck odometric distance faults and multiple and stuck odometric velocity faults, are considered. The analytical redundancy models of the foregoing sensors are developed. A measure estimation method is proposed by using a measurement, which is sensitive to fault. By using the estimated measurements and the real-time measurements, the residual is designed based on a nonsymmetrical Gaussian transformation. Then, fault detection and isolation can be achieved by comparing the residuals and a predefined threshold. Experimental results illustrate the effectiveness of the fault diagnosis scheme.

Xiao He
Zidong Wang
Gang Li
Zhijie Zhou
Youqing Wang

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 3583610, 14 pages
https://doi.org/10.1155/2017/3583610

Research Article

A Novel Multimode Fault Classification Method Based on Deep Learning

Funa Zhou,[1] Yulin Gao,[1] and Chenglin Wen[2]

[1]*School of Computer and Information Engineering, Henan University, Kaifeng, China*
[2]*School of Automation, Hangzhou Dianzi University, Hangzhou, China*

Correspondence should be addressed to Yulin Gao; gaoyulinhn@163.com and Chenglin Wen; wencl@hdu.edu.cn

Received 23 December 2016; Accepted 22 February 2017

Academic Editor: Youqing Wang

Due to the problem of load varying or environment changing, machinery equipment often operates in multimode. The data feature involved in the observation often varies with mode changing. Mode partition is a fundamental step before fault classification. This paper proposes a multimode classification method based on deep learning by constructing a hierarchical DNN model with the first hierarchy specially devised for the purpose of mode partition. In the second hierarchy , different DNN classification models are constructed for each mode to get more accurate fault classification result. For the purpose of providing helpful information for predictive maintenance, an additional DNN is constructed in the third hierarchy to further classify a certain fault in a given mode into several classes with different fault severity. The application to multimode fault classification of rolling bearing fault shows the effectiveness of the proposed method.

1. Introduction

Rolling bearing is a very pivotal component in rotating machines, which are widely used in large-scale automated industrial equipment. Mechanical failure caused by rolling bearings may cause abnormality of the rotating machinery system, resulting in huge economic losses, and even cause some unnecessary casualties [1–5]. Therefore, timely and precisely classification is critical for bearing monitoring.

The methods for mechanical equipment fault classification can be divided into qualitative model based method, quantitative model based method, and data-driven based method [6, 7]. Qualitative model and quantitative model based methods require precise mathematical model or a large amount of expert knowledge of the system, which will inevitably limit its application in fault classification field. In the recent two decades, data-driven method is widely used in fault detection of complex system. Instead of much more prior knowledge, data-driven approach can detect fault only through the measured data of the complex system [8–11]. The most common used data-driven fault classification methods are statistical feature extraction based methods and machine learning based methods. However, the method based on statistical feature extraction can only realize fault detection and it is unable to realize fault classification. For fault classification, we had better use machine learning method such as Support Vector Machine (SVM) and artificial neural network (ANN).

In the field of mechanical system fault classification, because of the sensitivity of vibration spectrum to equipment failure, vibration signals are usually used as the data source for fault classification of mechanical equipment. Due to mechanical equipment's characteristics of being nonstable, nonlinear, large-scale, high-dimensional, and noise polluted, it is usually very difficult for precise fault feature extraction which is the most critical factor of the accuracy of mechanical equipment monitoring [12–14]. Some scholars have put forward some feature extraction methods that combine signal processing technology with machine learning method for fault classification of mechanical equipment. Widodo and Yang extract the frequency-domain feature as the data source of SVM to detect the machinery fault [13]. When the number of samples is small and the signals are nonstationary, Yu et al. proposed a bearing fault classification method by combining SVM

and Empirical Mode Decomposition (EMD) [10]. Hu et al. extracted the energy of each wavelet packet transform (WPT) node as the preextracted feature to develop a combined WPT-SVM based method for more accurate bearing fault classification [15]. Wang et al. also used WPT to extract nonstationary characteristics of the bearing's vibration signal as the preextracted feature of ANN [16]. The method uses the nonlinear learning classification ability and self-organizing ability of ANN to classify and diagnose bearing faults. Yang and Tang proposed a method combining expert system and back propagation neural network (BPNN) [17]. This method makes full use of the advantages of expert system and ANN to successfully detect the bearing failure. Since bearing vibration signals are susceptible to Gaussian noise, Jiang et al. used high level statistics as the feature vector of BPNN to improve the performance of BPNN in bearing fault classification [18]. However, SVM and BPNN share the shortcomings of shallow learning method: SVM is an algorithm of two classifiers, and it is inefficient in multiclassification especially in the case when the sample number of observation is very large. Selecting the appropriate kernel function and scale parameter usually needs a wealth of experience. ANN also suffers from many defects, such as the following: (1) ANN has a slow convergence rate and can easily converge to the local optimum and (2) ANN is ineffective in feature learning of complex nonlinear data and usually results in poor classification accuracy. In summary, SVM and BPNN as the shallow learning methods could not well extract the data feature involved in the high-dimensional unsteady data [19]. With the load varying, bearing can work in different steady state, which is called "multimode" phenomenon. Current research work on machine learning based classification did not take multimode problem into account.

For multimode process, the data feature of each mode is different [20], but current research on bearing fault classification usually regards it as a single mode for simplicity of data processing which will result in inaccurate classification result since feature extracted is inaccurate [21–23]. Therefore, mode partition should be implemented before fault feature extraction of a separate mode for accurate feature extraction. Zhang et al. proposed an improved k-means clustering algorithm based on existing modal partition method [20]. Song et al. studied the issue to distinguish stability mode from transition mode without the number of modes known in advance [24]. Zhao et al. separated multiple modalities according to the diversity analysis in operational phases and established online monitoring method along multiple batch directions [25]. Zhang et al. used modal subspace separation method to deal with multimode monitoring problems [26]. By using various characteristics of the subspace, different mode can well be separated, which can provide chance for more accurate multimode fault classification.

Unfortunately, mode partition and corresponding fault monitoring method for certain multimode processes are only specially developed for a specific industrial process [20, 24–27]. It is required to develop a more universal method. Deep learning is a promising ubiquitous feature extraction tool which has attracted wide attention by scholars from various fields [21, 28–30]. Comparing to shallow learning, deep learning can well process the feature extraction and the issue of nonlinear big data by constructing a deep network [31, 32]. Through the unsupervised layer-by-layer greedy training algorithm and BP-based global parameter fine-tuning, deep neural network (DNN) can not only avoid the local optimization problem, but also solve the problem of limitation in number of labeled samples and the limitation in generalization ability. Deep learning method was firstly proposed by Hinton and Salakhutdinov in 2006 [22]. In view of its excellent feature extraction capabilities, it also attracts the attention of fault classification experts. Lu et al. successfully used the better feature extraction ability of deep neural network to diagnose the bearing fault [33]. The proposed method overcomes the shortcomings that the traditional feature extraction method could not discover the unknown type fault timely and effectively. Jia et al. used deep neural network to monitor the failure of bearings [34]. Gan et al. proposed a fault classification method based on hierarchical neural network [11]. By constructing a two-layer neural network, the method not only could locate the position of bearing fault but also effectively mines the fault size of the bearing in the same position. Deep learning, as one of the most popular machine learning methods, has brought a subversive revolution to the field of artificial intelligence. However, application about the deep learning is still in infancy, during the application process; there are also many issues demanding improvement. For example, the data in [11] are derived from a single mode, without considering the multimode observation caused by load varying problem. Therefore, it cannot fully extract the fault feature involved in the observation of different mode which is essential for the accuracy of multimode fault classification.

To solve the above-mentioned problems, this paper presents a multimode fault classification method based on deep learning. First, a DNN model is constructed, and the trained network is used to mode partition; then, a new set of DNNs are constructed for observation data of each mode, and the trained networks are used to determine which component fails to implement fault location recognition; finally, for a certain fault in a given mode, another DNN is constructed to classify those observation data with different fault size.

The remainder of this paper is as follows: Section 2 overviews the theory of deep learning. Section 3 develops a multimode fault classification method based on DNN by hierarchically constructing DNN models with different purpose. In Section 4, effectiveness of the proposed multimode fault classification method is demonstrated by experiments analysis. Section 5 concludes this paper.

2. Theory of Deep Learning

Deep learning is a method based on unsupervised feature learning. We use deep learning theory to construct DNN. DNN training process consists of two steps: (1) using the unsupervised learning algorithm to pretrain the network layer by layer, which is helpful for DNN to efficiently mine features from raw data; (2) using the back propagation algorithm to fine-tune the parameters of the whole network,

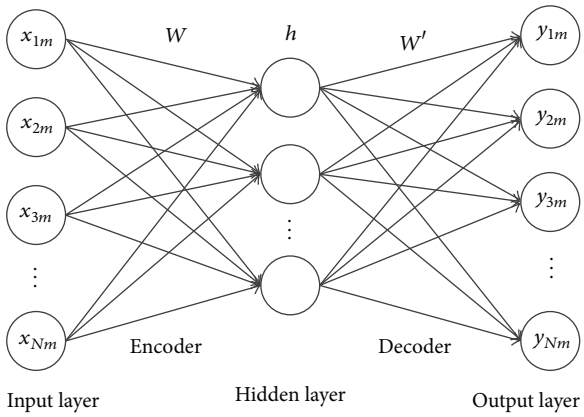


Figure 1: The model of AutoEncoder.

optimizing the performance of DNN to mine raw feature. In this paper, DNN is pretrained by multistacking AutoEncoder (AE).

2.1. AutoEncoder. AutoEncoder is an unsupervised machine learning structure, and it can be viewed as a three-layer forward artificial neural network, as shown in Figure 1. It consists of the input layer, the hidden layer, and the output layer. AutoEncoder is a very special neural network with single hidden layer, whose output is equal to the input. AutoEncoder network parameters can be adjusted by repeated training process, such that the reconstructed output is an approximation with high accuracy of the input. AutoEncoder is composed of two parts: encoder and decoder. The encoder network encodes the input data from the high-dimensional space into low-dimensional space; then the low-dimensional space data is mapped into high-dimensional space through decoder network which realized the reconstruction process from output to input. Therefore, the low-dimensional space data can be used as the characteristic representation of the input data.

Given an unlabeled dataset $\{x_{nm}\}$, $(n = 1, 2, \ldots, N;\ m = 1, 2, \ldots, M)$ consisting of N observation features or variables, each observation variable has M samples. The encoder network encodes the sample $x_m = [x_{1m}, x_{2m}, \ldots, x_{Nm}]^T$ to the hidden activate value h with an activation function f_θ. The encoder process is described as follows:

$$h = f_\theta(x_m) = \sigma(Wx_m + b), \tag{1}$$

where f_θ is the encoder function, Sigmoid function σ is usually taken as the activation function in the encoder process, W is the weight matrix of the network between input layer and the hidden layer, b is the bias vector generated by the encoder network, and $\theta = \{W, b\}$ is the connection parameter between the input layer and the hidden layer. The Sigmoid function can be depicted via

$$\sigma(x) = \frac{1}{[1 + \exp(-x)]}. \tag{2}$$

Similarly, for the decoder network, the feature matrix h obtained from encoder network is used to reconstruct y_m through the decoder network such that the reconstructed y_m is equal to the input x_m. The decoder process is described as follows:

$$y_m = g_{\theta^T}(h) = \sigma\left(W^T h + d\right), \tag{3}$$

where g_{θ^T} is the decoder function, σ is the activation function of the decoder process, W^T represents the weight matrix between the hidden layer and the output layer of the network, and d is the bias vector generated by the decoder process.

The essence of AE training process is to optimize the network parameters θ and θ^T. In order to make the output y_m as close as possible to the input x_m, we characterize the degree of approximation between input and output by minimizing the reconstruction error $J_{(\theta,\theta^T)}(x, y; W, b)$. The optimization process is described below:

$$J_{(\theta,\theta^T)}(x, y; W, b) = \frac{1}{m}\left\|y - x\right\|^2. \tag{4}$$

In each training process, the gradient descent method is used to update the training parameters θ and θ^T of the AE network. The processes of network parameter update are as follows:

$$\begin{aligned} W_l &= W_l - \alpha\frac{\partial}{\partial W_l}J_{(\theta,\theta^T)}(x, y; W, b), \quad l = 1, 2, \\ b_l &= b_l - \alpha\frac{\partial}{\partial b_l}J_{(\theta,\theta^T)}(x, y; W, b), \quad l = 1, 2, \end{aligned} \tag{5}$$

where α represents the learning rate and partial derivatives $(\partial/\partial W_l)J_{(\theta,\theta^T)}(x, y; W, b)$ and $(\partial/\partial b_l)J_{(\theta,\theta^T)}(x, y; W, b)$ can be calculated with back propagation algorithm.

DNN can be simply viewed as a multihidden layers neural network formed by stacking many AutoEncoders. This model uses the bottom-up method of unsupervised learning, extracting the features layer by layer. Then supervised learning method is applied to fine-tune the whole network parameters, which can extract the most essential characteristics from original signals. The structure of DNN is shown in Figure 2.

First of all, pretrain the DNN by using the unsupervised layer-by-layer greedy training algorithm. Firstly, the first AutoEncoder AE1 is trained by giving an unlabeled dataset x as the input of encoder network. The encoded feature h_1 is the hidden layer of AE1. The training parameter θ_1 is obtained by designing the unique x as the output of AE1. Then, use h_1 as the input of the second AutoEncoder (AE2) and train AE2 to acquire the network training parameter θ_2. h_2 is the hidden layer of AE2 which can be viewed as the characteristics of AE2. After that, choose h_2 as the input of the third AutoEncoder (AE3). Repeat the process to get the hidden layer features h_N of the Nth AutoEncoder (AE N) and the corresponding network training parameter θ_N.

Secondly, a classifier is added in the top layer of DNN. The feature information is extracted by using the unsupervised learning method in the pretraining process of DNN. However, DNN does not have the ability of classifying; a classifier should be added in the top of DNN. In this paper, Softmax

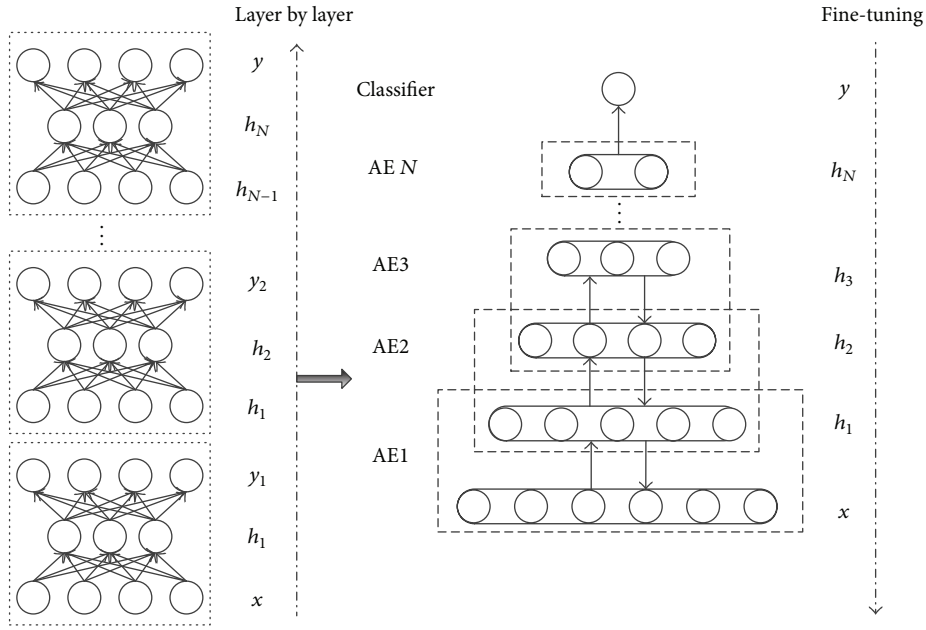


Figure 2: The structure of DNN.

classifier is used as the output layer of DNN. We suppose the training dataset is $\{x_m\}$ $(m = 1, 2, \ldots, M)$, the label is $u_m \in \{1, 2, \ldots, k\}$, and the probability $p(u = i \mid x)$ for each category i $(i = 1, 2, \ldots, k)$ can be calculated via the following hypothesis function:

$$h_\theta(x_m) = \begin{bmatrix} p(u_m = 1 \mid x_m; \theta) \\ p(u_m = 2 \mid x_m; \theta) \\ \vdots \\ p(u_m = k \mid x_m; \theta) \end{bmatrix} = \frac{1}{\sum_{i=1}^{k} e^{\theta_i^T x_m}} \begin{bmatrix} e^{\theta_1^T x_m} \\ e^{\theta_2^T x_m} \\ \vdots \\ e^{\theta_k^T x_m} \end{bmatrix}, \tag{6}$$

where θ is the model parameter of Softmax. Similarly to the AE model, in order to guarantee the performance of the classifier, the classifier model parameter is trained by minimizing the cost function J_θ. The cost function of Softmax training process is shown in (7), where the top network parameter θ_{N+1} is obtained from minimizing $J_\theta(x_m)$.

$$J_\theta(x_m) = -\frac{1}{M}\left[\sum_{m=1}^{M}\sum_{i=1}^{k} 1\{u_m = i\} \log \frac{e^{\theta_i^T x_m}}{\sum_{i=1}^{k} e^{\theta_i^T x_m}}\right]. \tag{7}$$

Finally, fine-tune. In order to guarantee the accuracy of feature extraction and the classification effectiveness of output layer, the whole DNN training parameters are fine-tuned by using a supervise algorithm of back propagation with some limited number of sample labels. The process of fine-tuning is completed by minimizing the reconstruction error $E(\theta)$. The procedures for parameter update are as follows:

$$E(\theta) = \frac{1}{M}\sum J_\theta\left(Y'_m, u_m; \theta\right), \qquad \theta = \theta - \alpha\frac{\partial E(\theta)}{\partial \theta}, \tag{8}$$

where Y_m represents the actual output value, θ is a parameter set generated from the whole network training, $\theta = \{\theta_1, \theta_2, \ldots, \theta_N, \theta_{N+1}\}$, back propagation algorithm is used to update the network parameter θ, and α is the learning rate in the process of deep learning. The fine-tuning process uses the labeled data to improve the performance of DNN.

2.2. DNN-Based Classification. In order to accurately extract the essential characteristics of the mechanical equipment health conditions by DNN modeling, the following steps are required. Firstly, the original vibration signals should be preprocessed. Since frequency-domain signals are more sensitive to mechanical equipment faults, the original time-domain signals are converted into frequency-domain signals in the first step. Secondly, use the preprocessed data as the input of the DNN model to extract features of mechanical equipment health conditions with unsupervised layer-by-layer pretraining. Last but not least, the whole network parameter θ can be updated by using the back propagation algorithm to fine-tune the DNN structure when limited number of labeled samples is available. In this way we can get an effective feature extraction result for fault classification. The preprocessed datasets are divided into training data and testing data. The training data is used to construct DNN model to obtain the training parameter θ, and the

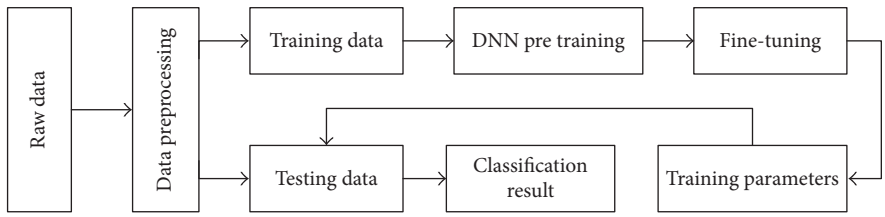


Figure 3: Framework of fault classification based on DNN.

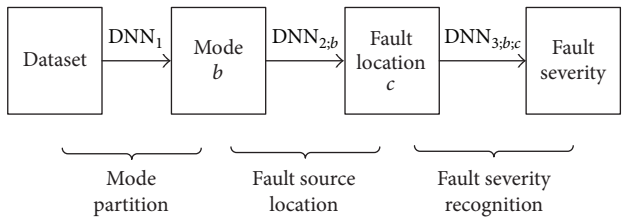


Figure 4: A hierarchical DNN framework of three layers.

testing network initialized with training parameter θ is used to verify its effectiveness. Misclassification rate is used as an accuracy indicator of the DNN-based fault classification method. Detailed steps of DNN for mechanical system fault classification are shown in Figure 3.

3. Multimode Fault Classification Model Based on Deep Learning

There are a number of multimode processes in practical system. For multimode process, the potential feature extracted from the observation of each steady mode also varies. So it is necessary to separate the observation into several operation modes for accuracy data feature extraction.

Therefore, mode partition is a fundamental step before fault classification. In this paper, this problem is solved by constructing a hierarchical DNN model with the first hierarchy specially devised for the purpose of mode partition. By this means, it can make an effective mode partition for multimode process, which can increase the accuracy of DNN-based fault classification. Framework of three-layer DNN is shown in Figure 4.

The detailed steps for multimode fault classification are as follows.

Step 1 (mode partition). In this step, we focus on building a DNN model to determine the mode label of each sample. The whole datasets are used as the input of the multimode classification model. The mode partition process can be illustrated in detail as follows.

(1) Construct a new DNN_1 with N hidden layers AE descripted in (9), and initialize the training parameters of DNN_1.

$$\begin{aligned}&\left[\mathrm{Net}_1, \mathrm{Tr}_1\right]\\&\quad = \mathrm{Feedforward}\left(\theta_{11}'; H_{11}, H_{12}, \ldots, H_{1N}; S_1\right),\end{aligned} \tag{9}$$

where $\theta_{11}' = \{W_1, b_1\}$, where W_1 is the weight matrix and b_1 is the bias vector. $H_{11}, H_{12}, \ldots, H_{1N}$ are the numbers of hidden layer neurons in DNN_1. The network configuration can be represented by Tr_1. S_1 denotes the training dataset. We use M_{11} in (10) to represent the number of neurons in the input layer of DNN_1.

$$M_{11} = \mathrm{size}\left(S_1\right). \tag{10}$$

The parameters of DNN_1 can be initialized via

$$\begin{aligned}W_1 &= \mathrm{rand}\left(H_{11}, M_{11}\right),\\ b_1 &= \mathrm{zeros}\left(H_{11}, 1\right).\end{aligned} \tag{11}$$

(2) Training of DNN_1 to obtain the net parameter θ_1'. Unsupervised layer-by-layer feature extraction based on the training dataset S_1 is implemented to the N-level AE defined in (9).

$$\begin{aligned}h_1 &= f_{\theta_{11}'}\left(S_1\right) = \sigma\left(W_1 \cdot S_1 + b_1\right),\\ h_N &= f_{\theta_{1N}'}\left(h_{N-1}\right).\end{aligned} \tag{12}$$

Add a Softmax classifier on the top of DNN_1. Limited number of training labels sets ρ_1 is used to fine-tune DNN_1 and update the training parameter θ_1' via

$$\begin{aligned}E\left(\theta_1'\right) &= \min \frac{1}{m} \sum J_{\theta_1'}\left(Y_1, \rho_1; \theta_1'\right),\\ \theta_1' &= \theta_1' - \alpha_1 \frac{\partial E\left(\theta_1'\right)}{\partial \theta_1'},\end{aligned} \tag{13}$$

where $\theta_1' = \{\theta_{11}', \theta_{12}', \ldots, \theta_{1N}', \theta_{1(N+1)}'\}$, with $\theta_{1(N+1)}'$ calculated by (6)-(7), and m is the number of samples. Y_1 denotes the output of DNN_1, and α_1 is learning rate in fine-tuning process.

(3) Mode partition uses the trained DNN_1. Once test sample S_1' is obtained, compute the probability of each test sample via the trained Net_1. Then use (14) to divide the test sample into different modes:

$$\begin{aligned}&\mathrm{Mode}\,(m)\\&\quad = \arg\max_b P\left(Y_1\,(m) = b \mid S_1'\,(m); \theta_1'; \mathrm{Tr}_1\right),\end{aligned} \tag{14}$$

where $m = 1, 2, \ldots, M$ and b ($b = 1, 2, \ldots, B$) is the mode type of sample. Mode(m) denotes the mode label of the mth test sample.

Compare the mode partition label Mode(m) with the actual mode label Label(m) to determine the misclassification number as

$$e_1 = \text{size}\left(S\_\text{miss}\right), \tag{15}$$

where size is the operation to characterize the size of a set and S_miss is the misclassification set defined by

$$S\_\text{miss} = \left\{ x_m \mid \arg_{x_m} \text{Mode}\left(m\right) \neq \text{Label}\left(m\right) \right\}. \tag{16}$$

Step 2 (fault source location). For a certain mode partitioned in Step 1, We can further locate the fault source. The procedure in Step 2 is analogous to Step 1, which is described below.

(1) According to the mode partition result, we build the second hierarchy of the model which comprises a set of B DNNs, and $S_{2;b}$ ($b = 1, 2, \ldots, B$) denotes the training dataset in DNN_2.

$$\begin{aligned}&\left[\text{Net}_{2;b}, \text{Tr}_{2;b}\right]\\ &\quad = \text{Feedforward}\left(\theta'_{2;b1}; H_{2;b1}, H_{2;b2}, \ldots, H_{2;bN}; S_{2;b}\right).\end{aligned} \tag{17}$$

Parameter initialization mechanism of DNN_2 is the same as Step 1.

(2) Train DNN_2 to obtain the net parameters $\theta'_{2;b}$. Similarly, for detailed calculation process, one can refer to (12)–(13).

(3) Determine the fault location by the trained $\text{Net}_{2;b}$.

The test dataset $S'_{2;b}$ is used to predict the unknown fault locations based trained $\text{Net}_{2;b}$. Assume that each mode has c ($c = 1, 2, \ldots, C$) different fault locations; fault location label for the mth sample of the bth mode can be calculated with prediction formula via

$$\begin{aligned}&\text{Location}_{2;b}\left(m\right)\\ &\quad = \arg\max_c P\left(Y_{2;b}\left(m\right) = c \mid S'_{2;b}\left(m\right); \theta'_{2;b}; \text{Tr}_{2;b}\right).\end{aligned} \tag{18}$$

Compute the misclassification number $e_{2;b}$ ($b = 1, 2, \ldots, B$) of the bth mode. And then the misclassification of this classification step can be computed via

$$e_2 = \sum_{b=1}^{B} e_{2;b}. \tag{19}$$

Step 3 (fault severity recognition). In order to identify the fault severity, the third hierarchy is devised with the intention to distinguish the fault severity. Construct the third deep network $\text{Net}_{3;b;c}$, $S_{3;b;c}$ is the training dataset in DNN_3, and $S'_{3;b;c}$ is the test dataset. Parameter training process is similar to Step 2. The severity classification label of the mth sample in $S'_{3;b;c}$ can be determined by

$$\begin{aligned}&\left[\text{Net}_{3;b;c}, \text{Tr}_{3;b;c}\right] = \text{Feedforward}\left(\theta'_{3;b;c1}; H_{3;b;c1}, H_{3;b;c2},\right.\\ &\left.\quad \ldots, H_{3;b;cN}; S_{3;b;c}\right)\end{aligned} \tag{20}$$

$$\begin{aligned}&\text{Severity}_{3;b;c}\left(m\right)\\ &\quad = \arg\max_d P\left(Y_3\left(m\right) = d \mid S'_{3;b;c}\left(m\right); \theta'_{3;b;c}; \text{Tr}_{3;b;c}\right).\end{aligned} \tag{21}$$

The misclassification number for a given fault in a certain mode can be computed via

$$\begin{aligned}e_{3;c} &= \sum_{b=1}^{B} e_{3;b;c},\\ e_3 &= \sum_{c=1}^{BC} e_{3;c},\end{aligned} \tag{22}$$

where $e_{3;b;c}$ is the misclassification number of the cth fault location in the bth mode, $e_{3;c}$ is the misclassification number of all C modes, and e_3 is the misclassification number in this step.

Step 4 (accuracy computation of the whole multimode classification network). In this paper, the classification accuracy of the hierarchical DNN is measured by the numbers of misclassifications. The final accuracy is calculated by the ratio of the total number of the misclassifications to the total number of samples. The procedure of calculation is as follows:

$$\text{Correct\_rate} = \left(1 - \frac{e_3}{M}\right) \times 100\%. \tag{23}$$

Combining (21) with (22), the final accuracy of the proposed multimode fault classification based on DNN can be formulated as

$$\text{Correct\_rate} = \left(1 - \frac{\sum_{b=1}^{B} \sum_{c=1}^{C} e_{3;b;c}}{M}\right) \times 100\%, \tag{24}$$

where N is the number of total samples, and the flow chart of the proposed multimode fault classification method based on three-layer DNN is depicted in Figure 5.

4. Application to Rolling Bearing Fault Classification

Rolling bearings play an important role for rotating machinery. The health condition of the bearing directly affects the reliability and stability in the whole system. Rolling bearing as the experimental platform is used to verify the effectiveness of the hierarchical DNN multimode fault classification method, and the performance of the proposed method is compared with the traditional method such as DNN, BPNN, SVM, hierarchical BPNN, and hierarchical SVM, which is listed in detail in Section 4.3.

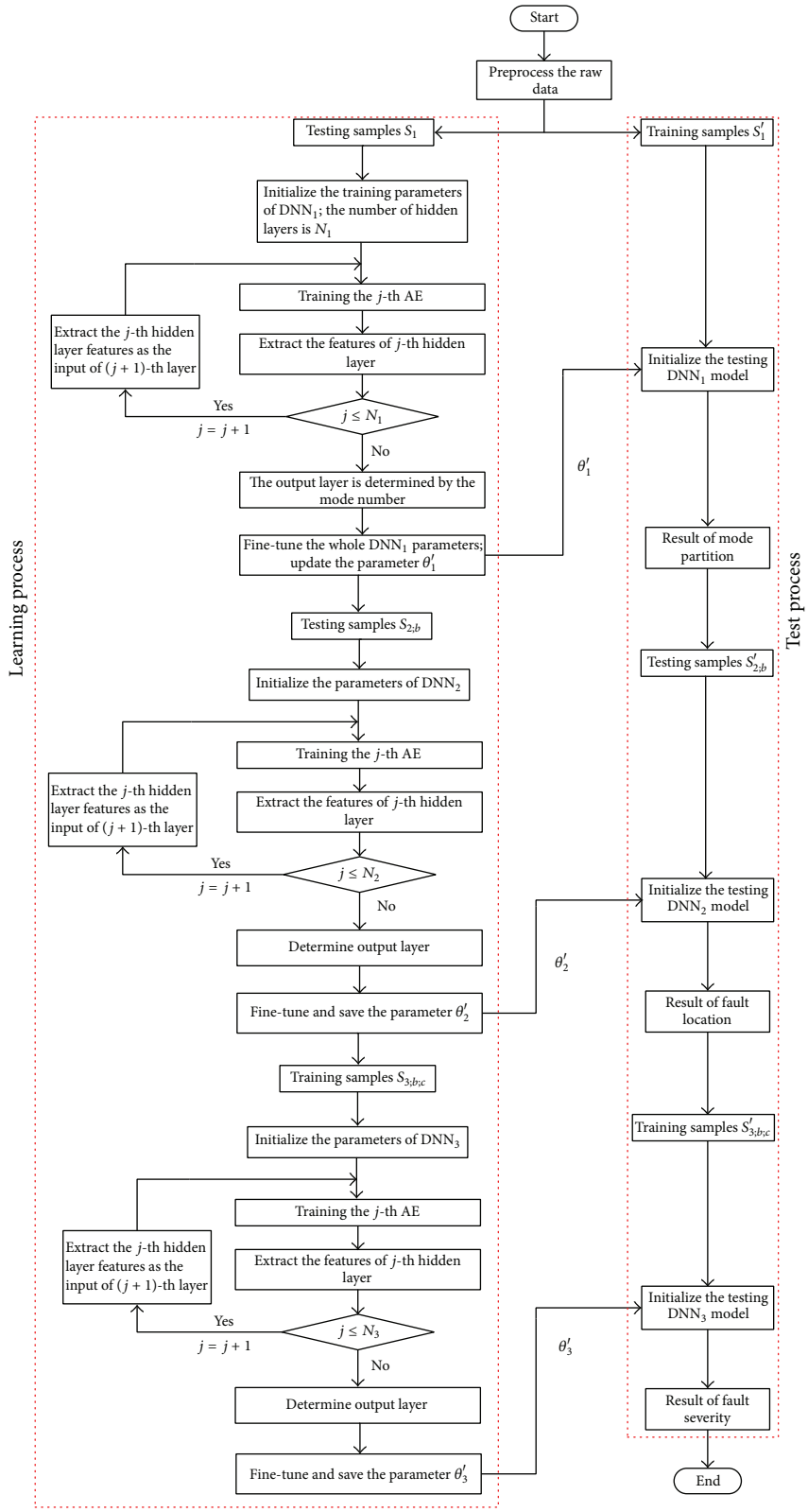


Figure 5: Flow chart of multimode classification based on DNN.

Table 1: Four modes of rolling bearing.

Mode	Load (hp)	Rotating speed (rpm)
Mode 1	0	1797
Mode 2	1	1772
Mode 3	2	1750
Mode 4	3	1730

Table 2: Data description of in dataset for a given mode.

Dataset for different fault location	Dataset for different fault severity	Sample number More/fewer	Fault type	Fault size /mm
		200/100	Normal (N)	0.000
$S_{b;1}$	$S_{b;1;1}$	200/100	Inner race fault (IF)	0.007
	$S_{b;1;2}$	200/100	Inner race fault (IF)	0.014
	$S_{b;1;3}$	200/100	Inner race fault (IF)	0.021
$S_{b;2}$	$S_{b;2;1}$	200/100	Outer race fault (OF)	0.007
	$S_{b;2;2}$	200/100	Outer race fault (OF)	0.014
	$S_{b;2;3}$	200/100	Outer race fault (OF)	0.021
$S_{b;3}$	$S_{b;3;1}$	200/100	Roller fault (RF)	0.007
	$S_{b;3;2}$	200/100	Roller fault (RF)	0.014
	$S_{b;3;3}$	200/100	Roller fault (RF)	0.021

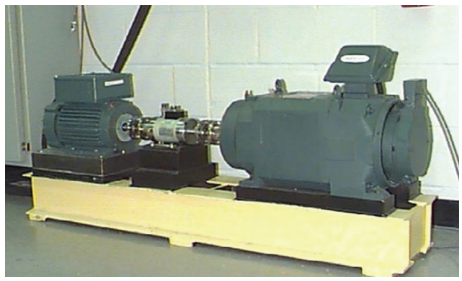


Figure 6: Experimental platform for acquiring the vibration signals of rolling bearing.

4.1. Experimental Platform. The experimental datasets are obtained from the Case Western Reserve University Bearing Data Center in the United States [35]. The experimental platform is shown in Figure 6. It can be seen that the experimental platform consists of a 2 hp motor, a power meter, an electronic controller, a torque sensor, and a load motor. The vibration signals of the drive end of the motor are collected by the acceleration sensor as the experimental datasets for bearing fault classification. In this experiment, we use acceleration sensor to collect the vibration signals with the load of 0 hp, 1 hp, 2 hp, and 3 hp, respectively, and the sampling frequency is 48 kHz. There are four types of bearing health condition: (1) normal condition; (2) inner race fault; (3) outer race fault; (4) roller fault. The sizes of the bearing fault were 0.007 mm, 0.014 mm, and 0.021 mm, respectively.

4.2. Data Description. In this case, we collect the vibration signals of the bearing drive end at different loading. The dataset collected contains 4 kinds of modes; the motor load is 0 hp, 1 hp, 2 hp, and 3 hp, respectively, and 4 modes are shown in Table 1. In each mode, there are four states of inner race fault, outer race fault, roller fault, and normal, with 3 different fault sizes in each fault state, that is to say 10 different fault types in a single mode. This paper selects 200 samples in each fault type; each sample contains 2048 observation points. 100 samples are randomly selected as the training data, and the other 100 samples as the testing data. We use Fast Fourier Transform (FFT) for each sample to get 2048 Fourier coefficients. Because of the symmetry of the Fourier coefficients, we take the first 1024 coefficients as the new samples; that is to say the dataset contains 8000 samples. In order to compare the proposed method of hierarchical network with single-layer network and explore the effect of different sample numbers on network, for a given mode, the sample number of each DNN is listed in Table 2. In addition, we present the original time-domain waveforms of the 10 fault types in mode 1 under A, as shown in Figure 7.

4.3. Results of Fault Classification. The proposed hierarchical DNN structure is applied to bearing fault classification; there are 8000 samples, 4 different modes, 4 fault positions in each mode, and totally 40 health conditions in dataset A. The health conditions of rotating machinery system under multimode, multicondition, multifault type, and large sample data are simulated which demonstrated the performance with

Table 3: DNN model parameters.

Training parameter	DNN_1	$DNN_{2;b}$	$DNN_{3;b;c}$
Hidden layers	5	4	3
Number of neurons	512/400/300/200/100	512/400/200/100	512/256/100
Max number of epochs	500	300	300

Table 4: Accuracy of classification in time domain and frequency domain.

Method	Time domain data	Frequency domain data
HDNN	80.65	99.96

Table 5: Fault severity classification result comparison after mode partition.

Method	Accuracy of fault classification	Accuracy of fault severity recognition
HDNN	99.79	99.52
DNN	97.06	96.38
HSVM	82.82	77.00
SVM	65.74	58.40
HBPNN	81.28	71.68
BPNN	68.11	62.42

the proposed method. To reduce the effect of randomness, the experiment was repeated 20 times. In this paper, the initialized parameters in the DNN pretraining process are shown in Table 3.

The network training uses stochastic gradient descent method; on each hierarchicy the maximum number of iterations of DNN is 500, 300, and 300, respectively. Simulation of three tradition methods, BPNN, SVM, and DNN, is compared with simulation of the proposed multimode fault classification approach to verify its effectiveness. In addition, hierarchical BPNN (HBPNN) and hierarchical SVM (HSVM) are also compared with hierarchical DNN (HDNN). BPNN uses the gradient descent method to update the network weights and bias parameter; one-to-one training mechanism is used to train a SVM with radial basis. The training mechanism of HBPNN and HSVM is the same as HDNN.

Table 4 compares the fault classification accuracies in time domain and frequency domain. It can be seen from Table 4 that rotation machinery fault is more sensitive in frequency domain. So we use FFT as a tool to preprocess the original data.

Table 5 compares the fault classification results after mode partition. It can be seen from line 2 and line 3 that HDNN can obtain more accurate classification either for fault source location or for fault severity recognition which tells us that mode partition is a critical step in multimode fault classification.

The hierarchical model for the case of BPNN and SVM also confirms this conclusion. Comparing line 2 with line 4 and line 6, we can see that HDNN is significantly superior to other hierarchical machine learning models because of the fact that HDNN can get better mode partition accuracy which is shown in Table 6. On the other hand, we can draw another conclusion that the performance of traditional BPNN method is superior to the traditional SVM method in the large sample case, but the accuracy of HSVM is higher than that of HBPNN due to the fact that SVM does well in small sample learning.

Table 6: The classification results by the second hierarchical of the proposed model.

Method	Mode partition
HDNN	99.96
HBPNN	90.45
HSVM	89.73

In order to demonstrate the performance of the proposed multimode classification method, the hierarchical machine learning methods are employed in this paper. As can be seen from Table 6, the accuracy of mode partition with proposed HDNN method can reach 99.96%, and we can naturally find that the performance of HDNN is superior to HBPNN and HSVM in mode partition procedure.

In view of the excellent performance of the proposed multimode classification method, we found that the performance

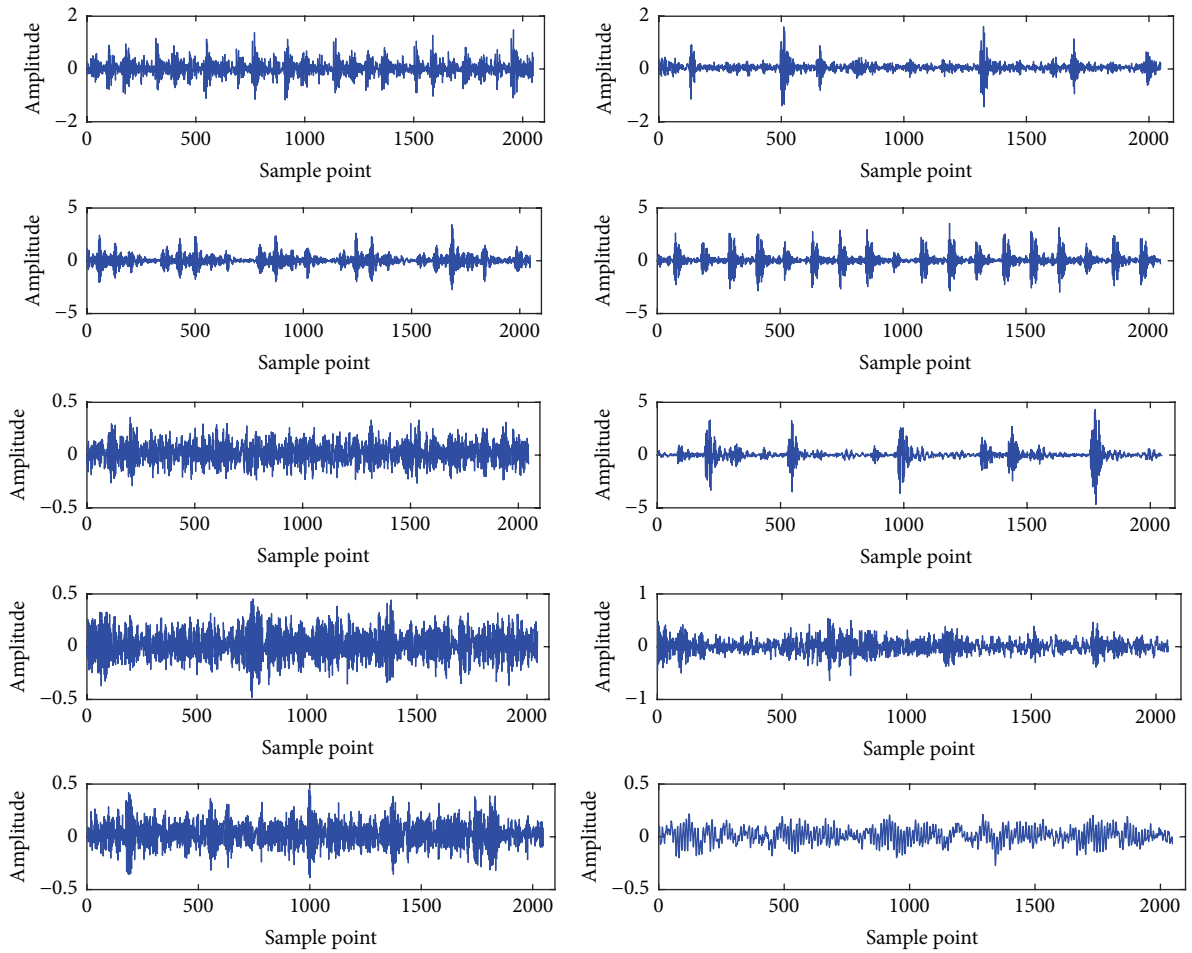


Figure 7: Observation of original signals corresponding to 10 fault types.

of classification was influenced by accurate feature extraction. In order to verify the effectiveness of HDNN based feature extraction method, scatter plots of the feature extracted are demonstrated in Figures 8–10. As shown in Table 3, in each training process, the number of neurons in the last hidden layer is 100; that is to say, the feature dimension is 100, which is too large to be visualized. Therefore, PCA is used as a data compression tool to reduce the feature dimension. In this paper, we use the first three key principal components to plot the scatter chart of the fault source location feature extracted by HDNN, as shown in Figure 8. Figure 8 is the scatter plots for fault feature extracted by HDNN after mode partition, while Figure 9 shows the scatter plots for fault feature extracted by DNN without mode partition. From Figures 9 and 10, we can see that some fault features are overlapped, which result in an unsatisfactory fault classification result.

Figure 10 is the scatter plot of the feature extracted for different modes. We can see from Figure 10 that HDNN does well in multimode fault feature extraction which will greatly affect the accuracy of the successive fault classification.

In summary, the proposed multimode classification method can accurately extract the different fault features based on its strong nonlinear characterization ability.

In general, efficiency of the fault classification method is affected by sample number of the train data. Figure 11 displays the fault classification accuracy of DNN and HDNN in two cases. Red line denotes the classification accuracy of the case when more samples are used as the training data. Black line denotes the classification accuracy of the case when fewer samples (only 1/2 of the first case) are used as the training data. In addition, the line with "$*$" is the simulation result of HDNN and the line with "□" is the simulation result of traditional DNN.

From Figure 11, it can be clearly seen that (1) fault classification accuracy of HDNN does not vary much for the two cases, while the fault classification accuracy of DNN is greatly affected by the number of training data used and (2) in both cases fault classification accuracy of HDNN is much better than DNN. So we can come to the conclusion that HDNN is a more robust fault classification for multimode bearing fault classification in the case when fewer number of training data are available.

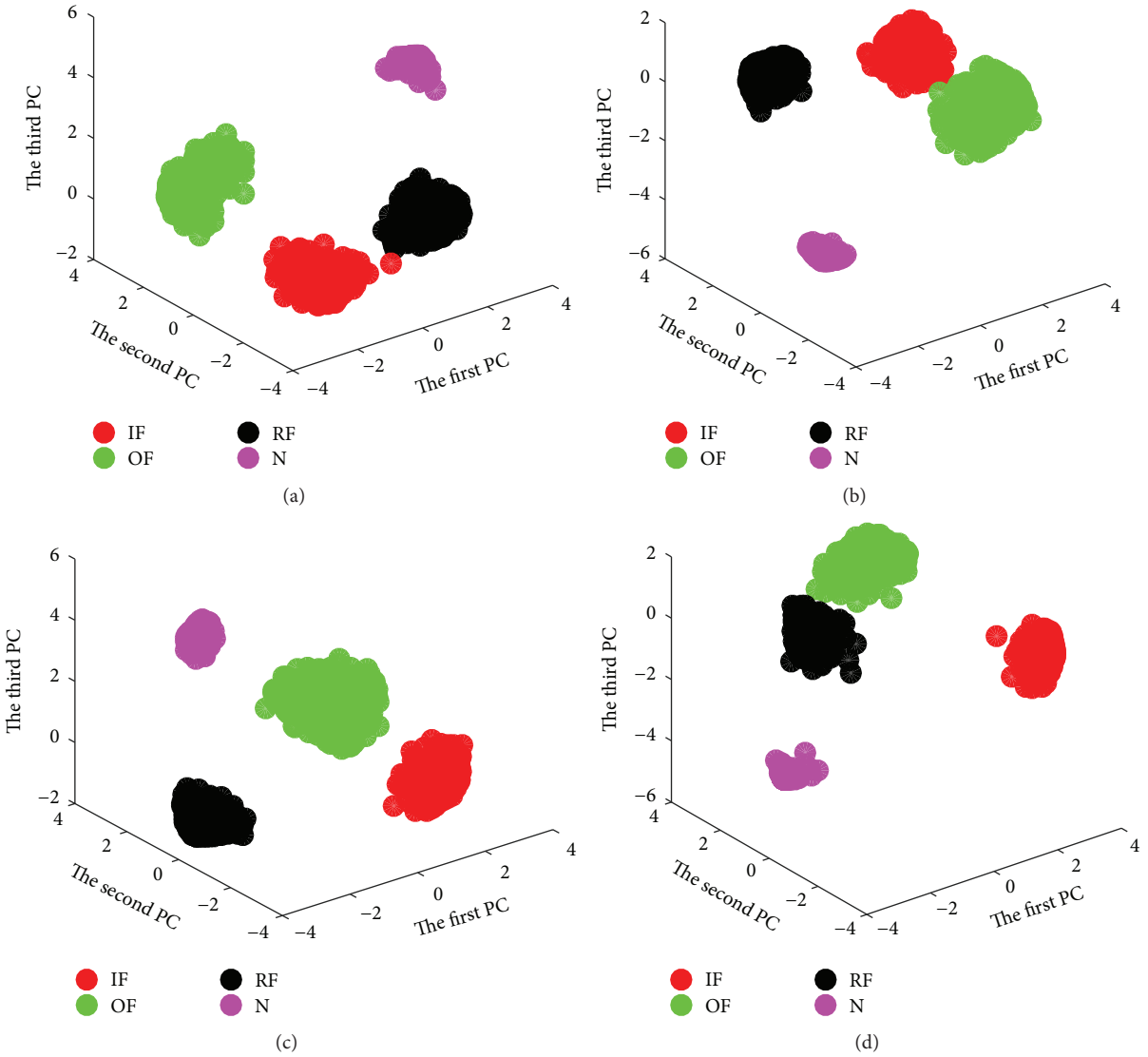


Figure 8: Scatter plots of principal components for the feature of fault classification; (a)–(d) represent four modes: corresponding to Mode 1, Mode 2, Mode 3, and Mode 4, respectively.

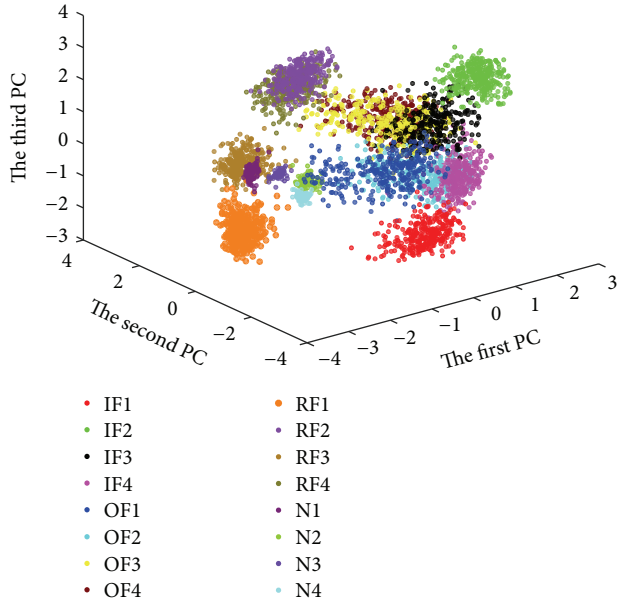


Figure 9: Scatter plots of principal components for fault features with traditional DNN method.

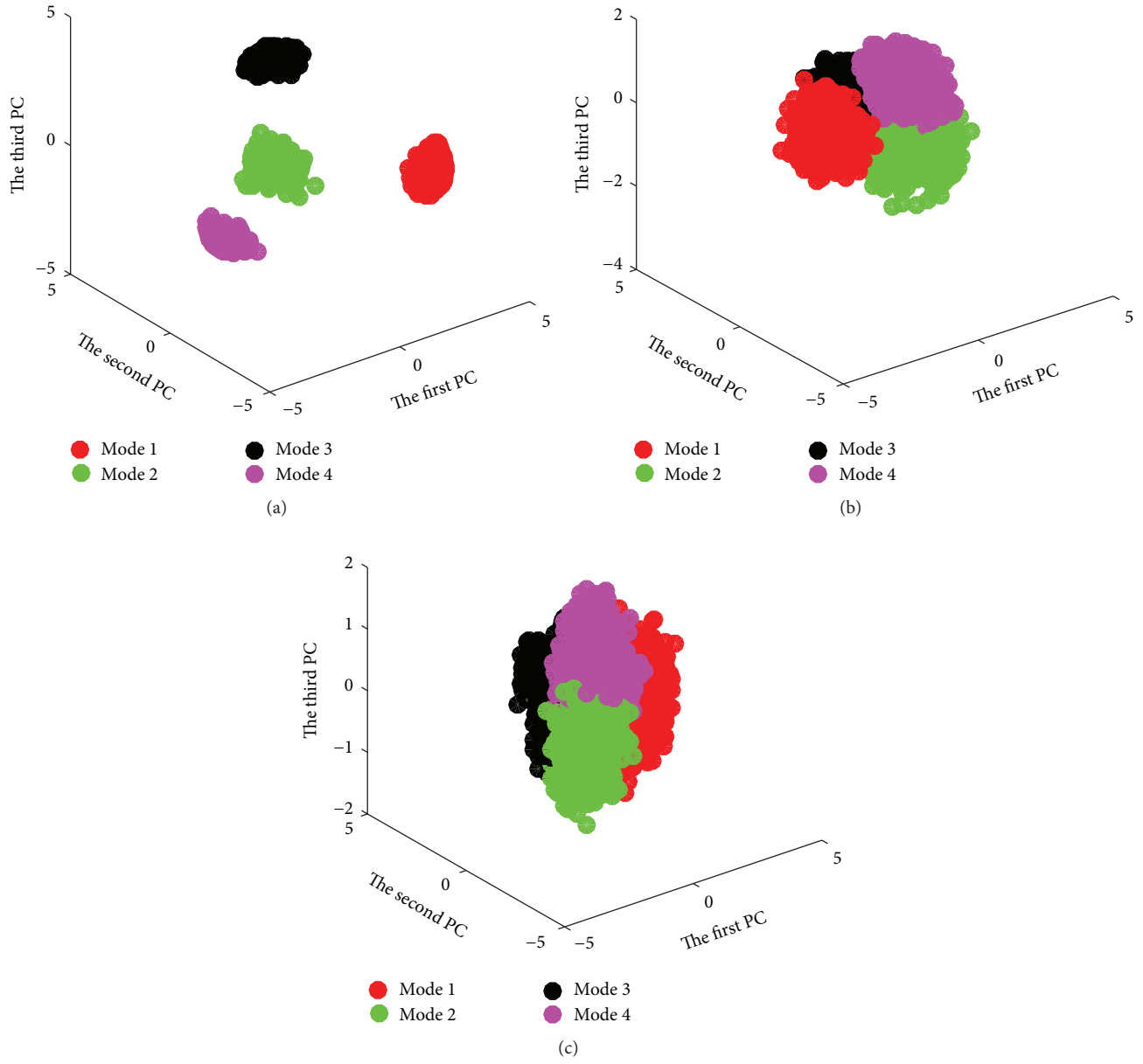


FIGURE 10: Scatter plots of principal components for mode partition features; (a)–(c) represent mode partition result corresponding to HDNN, HBPNN, and HSVM, respectively.

5. Conclusions

In this paper, a novel multimode fault classification method based on DNN is developed. The main idea is to construct a hierarchical DNN model with the first hierarchy specially devised for the purpose of mode partition. The second hierarchical model comprising a set of DNNs is devised to extract feature separately of different modes and precisely diagnose the fault source. Another set of DNNs is devised to distinguish the severity of a certain fault in a given mode, which is helpful for predictive maintenance of the machinery equipment. Rolling bearing is the experiment platform to verify the efficiency of the proposed method.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported in part by the Natural Science Fund of China (Grant nos. U1604158, U1509203, and

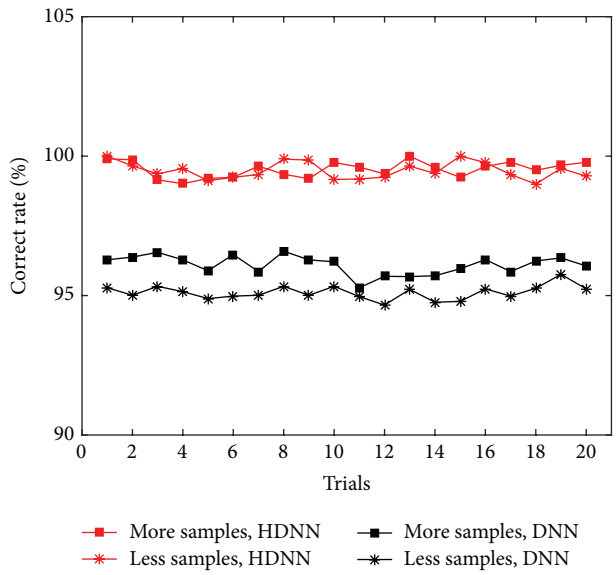


Figure 11: Robustness of the fault classification method to the sample number of training data with 20 trials.

61333005) and Technical Innovation Talents Scheme of Henan Province (Grant no. 2012HASTIT005).

References

[1] D.-H. Zhou, Y. Liu, and X. He, "Review on fault diagnosis techniques for closed-loop systems," *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1933–1943, 2013.

[2] F. N. Zhou, J. H. Park, and Y. J. Liu, "Differential feature based hierarchical PCA fault detection method for dynamic fault," *Neurocomputing*, vol. 202, pp. 27–35, 2016.

[3] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis," *Neurocomputing*, vol. 168, pp. 119–127, 2015.

[4] X. He, Z. Wang, Y. Liu, and D. H. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1670–1679, 2013.

[5] D. Zhao, D. Shen, and Y. Q. Wang, "Fault diagnosis and compensation for two-dimensional discrete time systems with sensor faults and time-varying delays," *International Journal of Robust and Nonlinear Control*, 2017.

[6] H. Li and D. Y. Xiao, "Surver on data driven fault classification methods," *Control and Decision*, vol. 26, no. 1, pp. 1–16, 2011.

[7] R. M. An and Y. Gao, "Spacecraft fault classification based on hierarchical neural network," *Spacecraft Environment Engineering*, vol. 30, no. 2, pp. 203–208, 2013.

[8] F. N. Zhou, C. L. Wen, Y. B. Leng, and Z. G. Chen, "A data-driven fault propagation analysis method," *Journal of Chemical Industry and Engineering (China)*, vol. 61, no. 8, pp. 1993–2001, 2010.

[9] H. Ji, X. He, and D. Zhou, "On the use of reconstruction-based contribution for fault diagnosis," *Journal of Process Control*, vol. 40, pp. 24–34, 2016.

[10] D. J. Yu, M. F. Chen, J. S. Cheng, and Y. Yang, "A fault classification approach for rotor systems based on empirical mode decomposition method and support vector machines," *Proceedings of the Chinese Society for Electrical Engineering*, vol. 26, no. 16, pp. 162–167, 2006.

[11] M. Gan, C. Wang, and C. A. Zhu, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 92–104, 2016.

[12] G. F. Bin, J. J. Gao, X. J. Li, and B. S. Dhillon, "Early fault diagnosis of rotating machinery based on wavelet packets—empirical mode decomposition feature extraction and neural network," *Mechanical Systems and Signal Processing*, vol. 27, no. 1, pp. 696–711, 2012.

[13] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[14] D. Zhao, Z. Lin, and Y. Wang, "Integrated state/disturbance observers for two-dimensional linear systems," *IET Control Theory & Applications*, vol. 9, no. 9, pp. 1373–1383, 2015.

[15] Q. Hu, Z. He, Z. Zhang, and Y. Zi, "Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 688–705, 2007.

[16] L. Y. Wang, W. G. Zhao, and Y. Liu, "Rolling bearing fault diagnosis based on wavelet packet- neural network characteristic entropy," *Advanced Materials Research*, vol. 108–111, no. 1, pp. 1075–1079, 2010.

[17] Y. Yang and W. Tang, "Study of remote bearing fault diagnosis based on BP Neural Network combination," in *Proceedings of the 7th International Conference on Natural Computation (ICNC '11)*, pp. 618–621, IEEE, Shanghai, China, July 2011.

[18] L. Jiang, Q. Li, J. Cui, and J. Xi, "Rolling bearing fault diagnosis based on higher-order cumulants and BP neural network," in *Proceedings of the 27th Chinese Control and Decision Conference (CCDC '15)*, pp. 2664–2667, IEEE, Qingdao, China, May 2015.

[19] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted

Boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, 2014.

[20] S. M. Zhang, F. L. Wang, S. Tan, and S. Wang, "A fully automatic onine mode identiflcation method for multi-mode processes," *Acta Automatica Sinica*, vol. 42, no. 1, pp. 60–80, 2016.

[21] J. Schmidhuber, "Deep Learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[23] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '13)*, pp. 7962–7966, IEEE, Vancouver, Canada, May 2013.

[24] B. Song, S. Tan, and H. B. Shi, "Key principal components with recursive local outlier factor for multimode chemical process monitoring," *Journal of Process Control*, vol. 47, pp. 136–149, 2016.

[25] L. Zhao, C. Zhao, and F. Gao, "Inter-batch-evolution-traced process monitoring based on inter-batch mode division for multiphase batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 138, pp. 178–192, 2014.

[26] Y. Zhang, C. Wang, and R. Lu, "Modeling and monitoring of multimode process based on subspace separation," *Chemical Engineering Research and Design*, vol. 91, no. 5, pp. 831–842, 2013.

[27] F.-N. Zhou, C.-L. Wen, T.-H. Tang, and Z.-G. Chen, "DCA based multiple faults diagnosis method," *Acta Automatica Sinica*, vol. 35, no. 7, pp. 971–982, 2009.

[28] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering & System Safety*, vol. 115, pp. 124–135, 2013.

[29] R. Huang, C. Liu, G. Li, and J. Zhou, "Adaptive deep supervised autoencoder based image reconstruction for face recognition," *Mathematical Problems in Engineering*, vol. 2016, Article ID 6795352, 14 pages, 2016.

[30] H. Liu, L. Li, and J. Ma, "Rolling bearing fault diagnosis based on STFT-deep learning and sound signals," *Shock and Vibration*, vol. 2016, Article ID 6127479, 12 pages, 2016.

[31] P. L. Wang and C. J. Xia, "Fault detection and self-learning identification based on PCA-PDBNs," *Chinese Journal of Scientific Instrument*, vol. 36, no. 5, pp. 1147–1154, 2015.

[32] R. Pang, Z. B. Yu, W. Y. Xiong, and H. Li, "Faults recognition of high-speed train bogie based on deep learning," *Journal of Railway Science and Engineering*, vol. 12, no. 6, pp. 1283–1288, 2015.

[33] C. Lu, Z. Y. Wang, W. L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, pp. 377–388, 2017.

[34] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 303–315, 2016.

[35] Bearing Data Centre, Case Western Reserve University, http://csegroups.case.edu/bearingdatacenter/home.

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 9560206, 13 pages
https://doi.org/10.1155/2017/9560206

Research Article

An Efficient Quality-Related Fault Diagnosis Method for Real-Time Multimode Industrial Process

Kaixiang Peng, Bingzheng Wang, and Jie Dong

Key Laboratory for Advanced Control of Iron and Steel Process, Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

Correspondence should be addressed to Jie Dong; dongjie@ies.ustb.edu.cn

Received 22 December 2016; Accepted 13 February 2017; Published 12 March 2017

Academic Editor: Zhijie Zhou

Focusing on quality-related complex industrial process performance monitoring, a novel multimode process monitoring method is proposed in this paper. Firstly, principal component space clustering is implemented under the guidance of quality variables. Through extraction of model tags, clustering information of original training data can be acquired. Secondly, according to multimode characteristics of process data, the monitoring model integrated Gaussian mixture model with total projection to latent structures is effective after building the covariance description form. The multimode total projection to latent structures (MTPLS) model is the foundation of problem solving about quality-related monitoring for multimode processes. Then, a comprehensive statistics index is defined which is based on the posterior probability of the monitored samples belonging to each Gaussian component in the Bayesian theory. After that, a combined index is constructed for process monitoring. Finally, motivated by the application of traditional contribution plot in fault diagnosis, a gradient contribution rate is applied for analyzing the variation of variable contribution rate along samples. Our method can ensure the implementation of online fault monitoring and diagnosis for multimode processes. Performances of the whole proposed scheme are verified in a real industrial, hot strip mill process (HSMP) compared with some existing methods.

1. Introduction

With modern industrial processes getting increasingly complex and large, prevention monitoring and fault diagnosis have become the key to ensure safe operation, improve product quality, and gain economic benefits. Due to the complex operation mechanism, sheer size, complex conditions, chaotic environment, and vague boundary conditions in complex industrial systems, it is quite tough to implement effective process monitoring. As a result, the data-driven process monitoring technology has become one of the research hotspots in the field of fault diagnosis. The core idea of this technique is to establish the data model by means of using historical data, mining useful information, and getting the features of normal and fault operation mode, so as to realize process monitoring. In the last decades, basic multivariate statistical monitoring techniques, such as principal component analysis (PCA) and partial least squares (PLS), have been established and successfully applied in practice [1].

However, PCA or PLS model is established with data which follow the basis hypothesis of data subject to stable single Gaussian mode. Due to the reasons of fluctuation of raw materials, product specifications, and differences among batches, process data show the characteristic of multimode in actual industrial processes especially for batch processes. Considering the problems existing in the multimode process, traditional fault detection methods and their improved algorithms are difficult to be applied directly; otherwise, the performance of data model in process monitoring will be reduced.

Many scholars have studied a lot and made some progress on those problems [1]. Hwang and Han proposed a hierarchical clustering based on the PCA modeling method [2]. Lane et al. proposed a pooled principal component analysis method [3]. However, the ensemble modeling methods, in which the common feature of subspace in each mode is extracted as a unified model, are unable to fully or accurately depict all operation models. Particularly, when there are many

differences among various modes, the model characterization in their methods is often biased. Chen and Liu used the heuristic smoothing clustering algorithm to classify data automatically, which can get multiple operating modes [4]. Zhao et al. applied multiple PCA and multiple PLS method to fault monitoring for multimode processes [5], in which the similarity index between different operating models was established and used to analyze the shift between the models. In view of stage division, Doan and Srinivasan modeled different stages of the process, respectively, for fault monitoring [6]. Dealing with the multimode problem of the process, the former divided the process data using the clustering method and then established independent models, so as to make fault monitoring more targeted. However, the above independent modeling methods are often complex, have large calculating quantity, and are usually based on the experience of mode division. Whether the division is reasonable or not will directly affect the quality of monitoring results. All the above increase the difficulties of online monitoring.

Considering the unique advantages in dealing with non-Gaussian data, the Gaussian mixture model (GMM) has not been explored in multimode process monitoring until recently. Choi et al. integrated PCA and DA with GMM to detect and isolate the faults in a process with nonlinearity, multistates, or dynamics [7]. Yoo et al. applied a similar strategy into multiway PCA to monitor biological batch processes [8]. However, these methods ignore the possibility that the monitored sample may come from other Gaussian components of lower posterior probabilities, which may lead to biased monitoring results. Yu and Qin proposed a new method that combines finite mixture Gaussian models with Bayesian inference to characterize different operation modes through Gaussian components and then realized fault detection [9]. In recent years, many scholars had proposed different methods to solve multimode monitoring [9].

The main contribution of this paper is summarized as follows. (1) An efficient method for multimode process monitoring based on finite Gaussian mixture models is proposed. (2) A gradient contribution rate is proposed to measure the contribution to the combined index and find out the variable which should be in charge of the fault in quality. This rate can better show the changes of variables contribution rate over time after fault occurrence.

The remainder of this paper is organized as follows. In Section 2, the descriptions of traditional PCA and PLS models in covariance form are provided, and then the covariance description form of the total projection to latent structures (TPLS) model is derived. Multimode information is extracted from the principal component space by GMM and a new multimode total projection to potential structure (MTPLS) model is established in Section 3. A unified monitoring framework based on MTPLS in combination with Bayesian inference is constructed and quality-related fault monitoring is implemented using a combined index in Section 4. In Section 5, a hot strip mill process is taken as an example to verify the superiority of our new method in fault monitoring and diagnosis over traditional methods. The conclusions and future works are given in Section 6.

2. Multivariate Statistical Theory

2.1. PCA and Covariance Description Form. Principal component analysis model is one of the most basic projection models in multivariate statistical analysis. Let $\mathbf{X} \in \mathbf{R}^{N*m}$ be the dataset of m-dimensional process variables, where N stands for the number of samples. Matrix $\mathbf{X}$ can be decomposed into a score matrix and a loading matrix as follows [10]:

$$\begin{aligned} \mathbf{X} &= \widehat{\mathbf{X}} + \mathbf{E} = \mathbf{T}\mathbf{P}^T + \mathbf{E}, \\ \mathbf{T} &= \mathbf{X}\mathbf{P}, \end{aligned} \tag{1}$$

where $\mathbf{T} \in \mathbf{R}^{N*A}$ and $\mathbf{P} \in \mathbf{R}^{m*A}$ stand for score matrix and loading matrix, respectively, and A is the number of principal components. The covariance matrix of normalized data can be defined as follows:

$$\mathbf{\Sigma}_{\mathbf{X}} \approx \frac{1}{N-1}\mathbf{X}^T\mathbf{X}. \tag{2}$$

The PCA loading matrix $\mathbf{P}$ can be obtained by eigenvalue decomposition on the covariance matrix $\mathbf{\Sigma}_{\mathbf{X}}$.

Based on the projection model, monitoring statistics indexes T^2 and SPE can be constructed. Let $\mathbf{x}_{\text{new}} \in R^m$; the indexes can be designed as follows:

$$\begin{aligned} T^2 &= \mathbf{x}_{\text{new}}^T \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T\mathbf{x}_{\text{new}} \le T_\alpha^2, \\ \text{SPE} &= \left\| \left(\mathbf{I} - \mathbf{P}\mathbf{P}^T\right)\mathbf{x}_{\text{new}} \right\|^2 \le \delta_\alpha^2, \end{aligned} \tag{3}$$

where $\mathbf{\Lambda}$ denotes the principal component covariance matrix and T_α^2 and δ_α^2 are the control limit with the confidence level of α.

When the residual error is subject to normal distribution, Jackson and Mudholkar pointed out that the control limit can be calculated as follows:

$$\delta_\alpha^2 = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 \left(h_0 - 1\right)}{\theta_1^2} \right)^{1/h_0}, \tag{4}$$

where $h_0 = 1 - 2\theta_1\theta_3/3\theta_1^2$, $\theta_i = \sum_{j=A+1}^{m} \lambda_j^i$ $(i = 1, 2, 3)$, c_α represents the threshold of standard normal distribution under the confidence level of α, and λ_j represents the eigenvalue of covariance matrix $\mathbf{\Sigma}_{\mathbf{X}}$.

Similarly, in order to apply the sample covariance information into the monitoring index, the principal component covariance matrix can be expressed as

$$\mathbf{\Lambda} = \frac{1}{N-1}\mathbf{T}^T\mathbf{T} = \frac{1}{N-1}\mathbf{P}^T\mathbf{X}^T\mathbf{X}\mathbf{P} = \mathbf{P}^T\mathbf{\Sigma}_{\mathbf{X}}\mathbf{P}. \tag{5}$$

2.2. PLS and Covariance Description Form. In the actual industrial production, the changes of quality variables $\mathbf{Y}$ are of more concern, especially for the faults which can cause the change of quality variables. PLS model uses the quality variables to guide the decomposition of sample space.

PLS decomposition of $\mathbf{X}$ and $\mathbf{Y}$ results in the following:

$$\begin{aligned}\mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E},\\ \mathbf{Y} &= \mathbf{T}\mathbf{Q}^T + \mathbf{F},\end{aligned} \tag{6}$$

where $\mathbf{X} \in \mathbf{R}^{N*m}$, $\mathbf{Y} \in \mathbf{R}^{N*l}$, and score matrix $\mathbf{T}$ can be formulated with $\mathbf{X}$ as $\mathbf{T} = \mathbf{X}\mathbf{R}$.

Parameter matrix $\mathbf{R}$ can be obtained by the loading matrix $\mathbf{P}$ and weight matrix $\mathbf{W}$ in PLS iterative calculation, $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$.

According to the iterative process of the PLS model, Peng et al. proposed a model construction method using data covariance information [11], in which the covariance matrix of data was introduced into the iterative process, and model parameter matrices can be obtained at the same time. Compared with conventional PLS, the model construction method using data covariance information reduced the calculation amount although the intrinsic properties essence was not changed.

Different from the PCA projection model, the decomposition structure of space $\mathbf{X}$ in PLS is defined by two matrices $\mathbf{P}$ and $\mathbf{R}$, and an oblique projection structure is induced in input space. It is the quality that guides the decomposition of sample space, so that the principal component space is changed. The covariance matrix of the principal component space can be expressed as

$$\mathbf{\Lambda} = \frac{1}{N-1}\mathbf{T}^T\mathbf{T} = \frac{1}{N-1}\mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R} = \mathbf{R}^T\mathbf{\Sigma}_{\mathbf{X}}\mathbf{R}. \tag{7}$$

Similar to PCA model monitoring, the monitoring sample statistics can be constructed by using the covariance matrix of the above formula as follows:

$$\begin{aligned}T^2 &= \mathbf{x}_{\text{new}}^T\mathbf{R}\mathbf{\Lambda}^{-1}\mathbf{R}^T\mathbf{x}_{\text{new}} \le T_\alpha^2,\\ Q &= \left\|\left(\mathbf{I} - \mathbf{P}\mathbf{R}^T\right)\mathbf{x}_{\text{new}}\right\|^2 \le \delta_\alpha^2.\end{aligned} \tag{8}$$

The control limit of the residual statistic can be calculated as follows:

$$\delta_\alpha^2 = g\chi_{h,\alpha}^2, \tag{9}$$

where $g = S/2\mu$, $h = 2\mu^2/S$, μ represents the sample mean of residual statistic Q, S represents the sample variance of Q, and $g\chi_{h,\alpha}^2$ is the threshold of χ^2 variables with scale factor g and free degree h.

3. TPLS Monitoring Model

3.1. TPLS. PLS algorithm uses two variable spaces to describe process change. However, the main component of samples contains the part which is orthogonal to $\mathbf{Y}$, and this part cannot reflect the variations related to $\mathbf{Y}$. On the other hand, PLS decomposition structure makes the residual in $\mathbf{X}$ remain very large, which is not suitable to be monitored by index Q. Therefore, Li et al. proposed a kind of total projection algorithm [12], which is based on traditional PLS decomposition. The original latent variable space is decomposed into one subspace relevant to quality variables directly and another subspace orthogonal to quality variables. At the same time, the residual space is decomposed into subspaces with large variance and residual subspace containing noise only, using the PCA orthogonal projection technique.

By further decomposition, we can model $\mathbf{X}$ and $\mathbf{Y}$ as follows:

$$\begin{aligned}\mathbf{X} &= \mathbf{X}_y + \mathbf{X}_o + \mathbf{X}_r + \mathbf{E}_r,\\ \mathbf{Y} &= \mathbf{T}_y\mathbf{Q}_y^T + \mathbf{F},\end{aligned} \tag{10}$$

where $\mathbf{X}_y = \mathbf{T}_y\mathbf{P}_y^T$, $\mathbf{X}_o = \mathbf{T}_o\mathbf{P}_o^T$, and $\mathbf{X}_r = \mathbf{T}_r\mathbf{P}_r^T$. $\mathbf{X}_y$ stands for the part which is relevant to $\mathbf{Y}$ directly in $\widehat{\mathbf{X}}$, $\mathbf{X}_o$ stands for the part which is orthogonal to $\mathbf{Y}$ in $\widehat{\mathbf{X}}$, and $\mathbf{X}_r$ stands for the part with large variance component in $\mathbf{E}$.

At the same time, based on the structure of PLS projection, Li et al. also performed a detailed analysis of the space structure of TPLS and drew a good conclusion [12]. Similar to PLS, TPLS also exhibits an oblique projection, but TPLS projects $\mathbf{x}$ to four different spaces, which reflect different relationship among quality variables.

For a new measurement of sample $\mathbf{x}_{\text{new}}$, the corresponding score and residual part can be calculated as follows [12]:

$$\begin{aligned}\mathbf{t}_{y\text{new}} &= \mathbf{Q}_y^T\mathbf{Q}\mathbf{R}^T\mathbf{x}_{\text{new}},\\ \mathbf{t}_{o\text{new}} &= \mathbf{P}_o^T\left(\mathbf{P}\mathbf{R}^T - \mathbf{P}_y\mathbf{R}_y^T\right)\mathbf{x}_{\text{new}},\\ \mathbf{t}_{r\text{new}} &= \mathbf{P}_r^T\left(\mathbf{I} - \mathbf{P}\mathbf{R}^T\right)\mathbf{x}_{\text{new}},\\ \mathbf{x}_{rr\text{new}} &= \left(\mathbf{I} - \mathbf{P}_r\mathbf{P}_r^T\right)\left(\mathbf{I} - \mathbf{P}\mathbf{R}^T\right)\mathbf{x}_{\text{new}}.\end{aligned} \tag{11}$$

Compared with PLS, TPLS model is easy to be explained and suitable for process monitoring. Similar to PLS in monitoring strategy, TPLS uses two statistic indexes T^2 and Q in process monitoring. In TPLS, $\mathbf{X}_y$, $\mathbf{X}_o$, and $\mathbf{X}_r$ represent the main variation in the process, and thus they are suitable for T^2 statistic, and $\mathbf{E}_r$ represents the residual part of the process which is suitable to be monitored by using statistic Q.

3.2. The Covariance Description of TPLS. The four spaces in TPLS can get a more detailed description of the different relationships between $\mathbf{X}$ and quality variables $\mathbf{Y}$. Based on the covariance matrix of the PLS model, the parameter matrices $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{W}$ will be obtained. Then, parameter matrix $\mathbf{R}$ is calculated by $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$.

Combining with the covariance description form of PCA and PLS model, we can do space decomposition in the following form.

In PCA decomposition of $\widehat{\mathbf{Y}}$, characteristic vectors of the covariance matrix $\mathbf{\Sigma}_{\widehat{\mathbf{Y}}}$ are extracted to construct $\mathbf{Q}_y$. $\mathbf{\Sigma}_{\widehat{\mathbf{Y}}}$ can be expressed as

$$\mathbf{\Sigma}_{\widehat{\mathbf{Y}}} = \frac{1}{N-1}\widehat{\mathbf{Y}}^T\widehat{\mathbf{Y}} = \mathbf{Q}\mathbf{R}^T\mathbf{\Sigma}_{\mathbf{X}}\mathbf{R}\mathbf{Q}^T. \tag{12}$$

Similarly, in PCA decomposition of $\widehat{\mathbf{X}}_o$ and $\mathbf{E}$, we can extract characteristic vectors of each covariance matrix to

form a loading matrix in corresponding space. Covariance matrices can be expressed as

$$\begin{aligned}\boldsymbol{\Sigma}_{\widehat{\mathbf{X}}_o} &= \frac{1}{N-1}\left(\widehat{\mathbf{X}} - \mathbf{T}_y\mathbf{P}_y^T\right)^T\left(\widehat{\mathbf{X}} - \mathbf{T}_y\mathbf{P}_y^T\right)\\ &= \boldsymbol{\Sigma}_{\widehat{\mathbf{X}}} - \boldsymbol{\Sigma}_{\widehat{\mathbf{X}}\widehat{\mathbf{Y}}}\mathbf{Q}_y\mathbf{P}_y^T - \mathbf{P}_y\mathbf{Q}_y^T\boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}\widehat{\mathbf{X}}} + \mathbf{P}_y\mathbf{Q}_y^T\boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}}\mathbf{Q}_y\mathbf{P}_y^T,\\ \boldsymbol{\Sigma}_{\mathbf{E}} &= \frac{1}{N-1}\left(\mathbf{X} - \widehat{\mathbf{X}}\right)^T\left(\mathbf{X} - \widehat{\mathbf{X}}\right)\\ &= \boldsymbol{\Sigma}_{\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}\mathbf{P}^T - \mathbf{P}\mathbf{R}^T\boldsymbol{\Sigma}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\widehat{\mathbf{X}}},\end{aligned} \tag{13}$$

where

$$\begin{aligned}\boldsymbol{\Sigma}_{\widehat{\mathbf{X}}} &= \mathbf{P}\mathbf{R}^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}\mathbf{P}^T,\\ \boldsymbol{\Sigma}_{\widehat{\mathbf{X}}\widehat{\mathbf{Y}}} &= \mathbf{P}\mathbf{R}^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}\mathbf{Q}^T,\\ \boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}\widehat{\mathbf{X}}} &= \mathbf{Q}\mathbf{R}^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}\mathbf{P}^T.\end{aligned} \tag{14}$$

According to the score and the residual structure model of new measurement samples, let

$$\begin{aligned}\mathbf{R}_y^T &= \mathbf{Q}_y^T\mathbf{Q}\mathbf{R}^T,\\ \mathbf{R}_o^T &= \mathbf{P}_o^T\left(\mathbf{P}\mathbf{R}^T - \mathbf{P}_y\mathbf{R}_y^T\right),\\ \mathbf{R}_r^T &= \mathbf{P}_r^T\left(\mathbf{I} - \mathbf{P}\mathbf{R}^T\right).\end{aligned} \tag{15}$$

It can be easily proved that this form is equivalent to the standard one.

The following part shows the calculation process of TPLS model using covariance information.

Covariance Description Form of TPLS Algorithm. Obtain $\boldsymbol{\Sigma}_{\mathbf{X}}$ and $\boldsymbol{\Sigma}_{\mathbf{XY}}$:

(1) Use GMM-PLS algorithm, and obtain parameter matrix: $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_A] \in R^{m*A}$, $\mathbf{W} = [w_1, \ldots, w_A] \in R^{m*A}$, $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_A] \in R^{l*A}$, $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$.

(2) Calculate PCA decomposition of $\widehat{\mathbf{Y}}$: do an eigenvalue decomposition on $\boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}}$; obtain the loading matrix $\mathbf{Q}_y = [\mathbf{q}_{y1}, \ldots, \mathbf{q}_{yA_y}] \in R^{m*A_y}$ and principal component number: $A_y = \mathrm{rank}(\mathbf{Q})$.

(3) $\mathbf{P}_y^T = (\mathbf{Q}_y^T\boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}}\mathbf{Q}_y)^{-1}\mathbf{Q}_y^T\boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}\widehat{\mathbf{X}}}$.

(4) Calculate PCA decomposition of $\widehat{\mathbf{X}}_o$: do an eigenvalue decomposition on $\boldsymbol{\Sigma}_{\widehat{\mathbf{X}}_o}$; obtain the loading matrix $\mathbf{P}_o = [\mathbf{p}_{o1}, \ldots, \mathbf{p}_{oA_o}] \in R^{m*A_o}$ and principal component number: $A_o = A - A_y$.

(5) Calculate PCA decomposition of $\mathbf{E} = \mathbf{X} - \widehat{\mathbf{X}}$: do an eigenvalue decomposition on $\boldsymbol{\Sigma}_{\mathbf{E}}$; obtain loading matrix $\mathbf{P}_r = [\mathbf{p}_{r1}, \ldots, \mathbf{p}_{rA_r}] \in R^{m*A_r}$ and principal component number A_r: based on the PCA method.

4. Multimode Process Monitoring and Fault Diagnosis

4.1. Mode Division of Principal Components. According to industrial process data with the characters of multimode, we need to determine a mixed model based on historical data firstly and then design a monitoring framework. Considering covariance information required for the statistical model, multimode modeling data can be processed by GMM. It is the assumption that data are made up of different Gaussian distributions. That is, for any sample data $\mathbf{x}$, it is possible to take a certain probability from K different Gaussian distributions. As a result, global probability distribution can be expressed by the mixed model of the K Gaussian elements. It can be expressed as

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{i=1}^{K} w_i p(\mathbf{x} \mid \boldsymbol{\theta}_i), \tag{16}$$

where K is the number of mixture components, w_i denotes the weight of the ith Gaussian component, and $\sum_{i=1}^{K} w_i = 1$, $\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$ represents the statistical parameters. Parameters estimation usually adopts EM iterative algorithm. The corresponding multivariate Gaussian density function for the ith component is given by

$$p(\mathbf{x} \mid \boldsymbol{\theta}_i) = \frac{1}{(2\pi)^{m/2}\left|\boldsymbol{\Sigma}_i\right|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right]. \tag{17}$$

According to the rule of Bayes inference, the posterior probability of $\mathbf{x}$ belonging to the ith Gaussian component is

$$p(\boldsymbol{\theta}_i \mid \mathbf{x}) = \frac{w_i p(\mathbf{x} \mid \boldsymbol{\theta}_i)}{\sum_{k=1}^{K} w_k p(\mathbf{x} \mid \boldsymbol{\theta}_k)}. \tag{18}$$

However, due to factors such as production flow, batch, and specification, the quality variables of the final products have some certain degree of difference in real production processes. It may be the root cause that process data is with multimode and multistage features. Therefore, considering that the PLS algorithm is with the space decomposition under guidance of quality variables, this paper first performs mode division with principal component space $\mathbf{T}$ and acquires the mode label C_k of $\mathbf{t}_i$. This method carried out with the projection of training data can highlight the influence of quality variables better.

Based on advantages of GMM in processing multimode problems, we deal with principal components matrix $\mathbf{T}$ with GMM for acquiring $\sum_{i=1}^{K} w_{ti} = 1$ and $\boldsymbol{\theta}_{ti} = \{\boldsymbol{\mu}_{ti}, \boldsymbol{\Sigma}_{ti}\}$. The total number of estimated parameters is $K((1/2)A^2+(3/2)A+1)-1$, where A is the number of the principal components. Usually, A is far less than process variables number m, which can reduce the number of estimated parameters greatly and speed up the calculation.

After mode division, principal component space model based on GMM is established, where each Gauss component corresponds to different mode characteristics. For training

samples, $\mathbf{x}_i$ can be divided into the modes whose principal variable belongs to

$$C_k = P\left(\boldsymbol{\theta}_{tk} \mid \mathbf{t}_i\right) = \frac{w_{tk} p\left(\mathbf{t}_i \mid \boldsymbol{\theta}_{tk}\right)}{\sum_{i=1}^{K} w_{ti} p\left(\mathbf{t}_i \mid \boldsymbol{\theta}_{ti}\right)}. \tag{19}$$

Taking process variables $x \in R^m$ and output variables $\mathbf{y} \in \mathbf{R}^l$ into account, we construct a new vector $\mathbf{z}$ which stands for the process information as follows:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \in \mathbf{R}^{(m+l)}. \tag{20}$$

Assuming that variable $\mathbf{z}$ is satisfied with mixed Gauss distribution, the distribution parameters $\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$ can be acquired by mean($\mathbf{z}$) and cov($\mathbf{z}$) directly; prior probabilities ω_i are the same as principal space distribution ω_{ti}.

$$\boldsymbol{\Theta}_{\mathrm{GMM}} = \left\{\left\{w_1, \boldsymbol{\mu}_z^{(1)}, \boldsymbol{\Sigma}_z^{(1)}\right\}, \ldots, \left\{w_K, \boldsymbol{\mu}_z^{(K)}, \boldsymbol{\Sigma}_z^{(K)}\right\}\right\}. \tag{21}$$

Divide $\boldsymbol{\mu}_z^{(k)}$ and $\boldsymbol{\Sigma}_z^{(k)}$ into the forms of [11]

$$\begin{aligned} \boldsymbol{\mu}_z^{(k)} &= \begin{bmatrix} \boldsymbol{\mu}_x^{(k)} \\ \boldsymbol{\mu}_y^{(k)} \end{bmatrix}, \\ \boldsymbol{\Sigma}_z^{(k)} &= \begin{bmatrix} \boldsymbol{\Sigma}_x^{(k)} & \boldsymbol{\Sigma}_{xy}^{(k)} \\ \boldsymbol{\Sigma}_{yx}^{(k)} & \boldsymbol{\Sigma}_y^{(k)} \end{bmatrix}. \end{aligned} \tag{22}$$

As above, it can be noted that mode classification will be under the guidance of quality variables. Then, because the number of principal components is far less than that of process variables, this has a great advantage in the treatment of estimated parameters calculation. In addition, after the mode division of original training data, multimode information such as covariance matrices can be directly calculated, which reduce the amount of calculation and improve calculation accuracy.

4.2. Multimode TPLS Based Fault Detection. According to the principle of building PLS and TPLS, the essence is to use data information, variance, and covariance to represent process characteristics. As far as PLS is concerned, the modeling process is to maximize the covariance of linear combinations of process variables and quality variables, so the modeling process can be converted into a covariance form through the initial data $\mathbf{X}$ and $\mathbf{Y}$. Therefore, in order to adapt to the multimode characteristic of industrial process data better, we can extend multivariate statistical methods to multimode scope by covariance strategy which will improve the performance of the monitoring model.

Based on the above analysis, we can make a rational division of training data to obtain the multimode information in the process of fault monitoring. When the sample is collected and is ready for being monitored, it can be divided into corresponding models with the probability, using Bayes classification ability under the data pretreatment. Then, we can calculate the monitoring statistic of the sample to justify which mode it belongs to. We treat the posterior probability of the monitoring sample belonging to each Gauss component as the membership degree of the corresponding model.

By using data information of probability w_i and parameters $\boldsymbol{\theta}$ to monitor the process, the comprehensive monitoring index is constructed, which can be used to monitor the fault reasonably.

For a new monitoring sample $\mathbf{x}_{\mathrm{new}} \in \mathbf{R}^m$, the probability of sample data belonging to different modes is $P(\boldsymbol{\theta}_k \mid \mathbf{x}_{\mathrm{new}})$.

4.3. Comprehensive Monitoring Index. According to PCA decomposition of $\widehat{\mathbf{Y}}$ in TPLS, $\mathbf{T}_y = \widehat{\mathbf{Y}}\mathbf{Q}_y$, the covariance matrix of principal components in space $\mathbf{X}_y$ can be expressed as [13]

$$\boldsymbol{\Lambda}_y = \frac{1}{N-1}\mathbf{T}_y^T\mathbf{T}_y = \frac{1}{N-1}\mathbf{Q}_y^T\widehat{\mathbf{Y}}^T\widehat{\mathbf{Y}}\mathbf{Q}_y = \mathbf{Q}_y^T\boldsymbol{\Sigma}_{\widehat{\mathbf{Y}}}\mathbf{Q}_y. \tag{23}$$

Available by $\widehat{\mathbf{Y}} = \mathbf{T}\mathbf{Q}^T$ and $\mathbf{T} = \mathbf{X}\mathbf{R}$,

$$\boldsymbol{\Lambda}_y = \mathbf{Q}_y^T\mathbf{Q}\mathbf{R}^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}\mathbf{Q}^T\mathbf{Q}_y = \mathbf{R}_y^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}_y. \tag{24}$$

In the same way, the covariance matrices of principal components in spaces $\mathbf{X}_o$ and $\mathbf{X}_r$ can be done as in the above proof:

$$\begin{aligned} \boldsymbol{\Lambda}_o &= \mathbf{R}_o^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}_o, \\ \boldsymbol{\Lambda}_r &= \mathbf{R}_r^T\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{R}_r. \end{aligned} \tag{25}$$

In order to realize the multimode fault monitoring, the monitoring index based on the MTPLS model is obtained by using the probability information and Bayesian inference:

$$\begin{aligned} &T_{y\mathrm{new}(k)}^2 \\ &\quad = \left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right)^T \mathbf{R}_{y(k)}\boldsymbol{\Lambda}_{y(k)}^{-1}\mathbf{R}_{y(k)}^T\left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right), \\ &Q_{r\mathrm{new}(k)} = \left\|\mathbf{x}_{r\mathrm{new}}\right\|^2 \\ &\quad = \left\|\left(\mathbf{I} - \mathbf{P}_{r(k)}\mathbf{P}_{r(k)}^T\right)\left(\mathbf{I} - \mathbf{P}_{(k)}\mathbf{R}_{(k)}^T\right)\left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right)\right\|^2. \end{aligned} \tag{26}$$

Similarly,

$$\begin{aligned} &T_{o\mathrm{new}(k)}^2 \\ &\quad = \left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right)^T \mathbf{R}_{o(k)}\boldsymbol{\Lambda}_{o(k)}^{-1}\mathbf{R}_{o(k)}^T\left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right), \\ &T_{r\mathrm{new}(k)}^2 \\ &\quad = \left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right)^T \mathbf{R}_{r(k)}\boldsymbol{\Lambda}_{r(k)}^{-1}\mathbf{R}_{r(k)}^T\left(\mathbf{x}_{\mathrm{new}} - \boldsymbol{\mu}_x^{(k)}\right). \end{aligned} \tag{27}$$

The threshold can be inferred by the setting in standard TPLS.

In summary, we make use of covariance information mainly to calculate and then to achieve process monitoring in MTPLS. Compared with standard TPLS, the covariance model is more suitable for monitoring multimode processes and making full use of data information in the process of model construction and fault monitoring. Avoiding direct classification on data, the covariance model reduces the effect of classification on the final performance monitoring of the process.

4.4. Quality-Related Combined Index. In TPLS based process monitoring, space $\mathbf{X}_y$ represents the change part related to quality variable, while space $\mathbf{E}_r$ represents the uncertain parts related to quality variable. They reflect two different kinds of quality-related faults. Therefore, it is necessary to observe two subspaces at the same time. In practice, a unified monitoring index is more popular than the two separate ones. In PCA based fault detection, Yue and Qin proposed a combined index [14]. Li et al. proposed a combined one for TPLS based process monitoring [12]. Similarly, a combined index which incorporates T_y^2 and Q_r is proposed in a way as follows:

$$\varphi_y = \frac{T_y^2}{\delta_y} + \frac{Q_r}{\delta_{rr}} = \mathbf{x}^T \mathbf{\Phi} \mathbf{x}, \tag{28}$$

where

$$\mathbf{\Phi} = \frac{\mathbf{R}_y \Lambda_y^{-1} \mathbf{R}_y^T}{\delta_y} + \frac{\left(\mathbf{I} - \mathbf{R}\mathbf{P}^T\right)\left(\mathbf{I} - \mathbf{P}_r\mathbf{P}_r^T\right)\left(\mathbf{I} - \mathbf{P}\mathbf{R}^T\right)}{\delta_{rr}}. \tag{29}$$

ζ^2 is the threshold of this combined index which can be obtained by approximate distribution $\zeta^2 = g\chi_{h,\alpha}^2$. It is supposed that there is no fault in the process when the monitoring result is $\varphi_y < \zeta^2$.

Scale factor g and free degree h are calculated in

$$\begin{aligned} g &= \frac{\operatorname{tr}(\mathbf{S}\mathbf{\Phi})^2}{\operatorname{tr}(\mathbf{S}\mathbf{\Phi})}, \\ h &= \frac{[\operatorname{tr}(\mathbf{S}\mathbf{\Phi})]^2}{\operatorname{tr}(\mathbf{S}\mathbf{\Phi})^2}, \end{aligned} \tag{30}$$

where $\mathbf{S} = \operatorname{cov}(\mathbf{x}) = \mathbf{\Sigma}_\mathbf{X}$, which is the covariance matrix of process variable $\mathbf{x}$. Using this combined index, we can simultaneously monitor the anomalies in the two subspaces and thus monitor the faults associated with the quality variables $\mathbf{Y}$.

4.5. Gradient Contribution Rate for Fault Diagnosis. It is necessary to isolate the faulty variables after a fault is detected. As a common fault separation method, the contribution plot assumes that the variables which have greater contribution to the monitoring statistics are very likely to be faulty variables. According to the description framework of complete decomposition of contribution proposed by Alcala and Qin, contribution to the combined index can be described as the following form [15]:

$$\begin{aligned} \operatorname{con}\mathbf{t}_i &= \left(\boldsymbol{\gamma}_i \mathbf{x}\right)^2 = \left(\boldsymbol{\xi}_i^T \mathbf{\Phi}^{1/2} \mathbf{x}\right)^2, \\ \text{index} &= \left\|\mathbf{\Phi}^{1/2}\mathbf{x}\right\|^2 = \sum_{i=1}^{m} \operatorname{con}\mathbf{t}_i, \end{aligned} \tag{31}$$

where $\boldsymbol{\gamma}_i$ represents the ith row of matrix $\mathbf{\Phi}^{1/2}$, $\boldsymbol{\xi}_i$ represents the ith row of identity matrix, and m represents the number of variables in one sample.

Traditional contribution plot method is used for analyzing a specific sample when the fault is detected, which shows the contribution value of each variable to one monitoring index in bar chart. After that, the variables with greater contribution will be selected as the possible cause of fault. Westerhuis et al. put forward a generalized contribution to statistics form and a method of obtaining the control limits for variable contributions [16]. Choi et al. proposed specific statistical methods to set the upper limit of the variable contribution to the four monitoring statistics [7]. Li et al. proposed a kind of contribution plot based on TPLS, which describes the contribution of all variables to monitoring index T_y^2 and Q_r in a unified way [12].

For the fault diagnosis method based on traditional contribution figure for one single sample after fault occurrence, there are some flaws that cannot well describe fault source and the change of other malfunction variables caused by fault source. In order to combine the idea of analyzing the contribution rate of faulty variables along the time coordinates with the change of the variable itself, reducing the impact of variable magnitude of value on the contribution rate, we refer to the gradient contribution rate to solve the fault variable analysis.

First, we introduce a mathematical symbol $\odot$ and a scale factor vector $\mathbf{v} = [\nu_1, \ldots, \nu_m]^T$, where $\odot$ indicates element product. $\mathbf{x} \odot \mathbf{v} = [\mathbf{x}_1\nu_1, \ldots, \mathbf{x}_i\nu_i, \ldots, \mathbf{x}_m\nu_m]^T$, and $\mathbf{x}_i\nu_i$ indicates the change of variable $\mathbf{x}_i$. As can be segmented, if $\nu_i > 1$, then $|\mathbf{x}_i\nu_i| > |\mathbf{x}_i|$; if $\nu_i = 1$, then $\mathbf{x}_i\nu_i = \mathbf{x}_i$; if $\nu_i < 1$, then $|\mathbf{x}_i\nu_i| < |\mathbf{x}_i|$. So, equation $\varphi(\mathbf{v} \odot \mathbf{x})|_{\mathbf{v}=\mathbf{1}_m} = \varphi(\mathbf{x})$ can be established.

It can be seen from the first-order Taylor series expansion of $\varphi(\mathbf{v} \odot \mathbf{x})$ near $\mathbf{v} = \mathbf{1}_m$ that

$$\varphi(\mathbf{x} \odot \mathbf{v}) \approx \varphi(\mathbf{x}) + \sum_{i=1}^{m} \left.\frac{\partial \varphi(\mathbf{x} \odot \mathbf{v})}{\partial \nu_i}\right|_{\mathbf{v}=\mathbf{1}_m} (\nu_i - 1). \tag{32}$$

Based on the above conclusion, the contribution rate may be defined as follows.

For a monitoring sample $\mathbf{x}$, $C(\mathbf{x}, i) \triangleq |(\partial\varphi(\mathbf{x}\odot\mathbf{v})/\partial\nu_i)|_{\mathbf{v}=\mathbf{1}_m}|$ indicates the contribution rate of the ith variable to index φ.

As described above, the contribution rate represents the gradient of each variable to detection index under the same abnormal changes. Variables which are with great contribution will be considered with great influence to index φ, the same to quality variable.

For a new monitoring sample $\mathbf{x}_{\text{new}}$, the contribution rate of the ith variable can be calculated as

$$\left| \left.\frac{\partial \varphi(\mathbf{x}_{\text{new}} \odot \mathbf{v})}{\partial \nu_i}\right|_{\mathbf{v}=\mathbf{1}_m} \right|. \tag{33}$$

As a result, the gradient contribution rate based on comprehensive monitoring index φ_y can be expressed as follows:

$$\begin{aligned} C(\mathbf{x}_{\text{new}}, i) &= \left| \left.\frac{\partial\left(\boldsymbol{\xi}_i^T \mathbf{\Phi}^{1/2}(\mathbf{x}_{\text{new}} \odot \mathbf{v})\right)^2}{\partial \nu_i}\right|_{\mathbf{v}=\mathbf{1}_m} \right| \\ &= \left|2\boldsymbol{\xi}_i^T \mathbf{\Phi}^{1/2} \mathbf{x}_{\text{new}} \mathbf{x}_{\text{new},i}\right|, \end{aligned} \tag{34}$$

where $\mathbf{x}_{\text{new},i}$ represents the value of the ith variable in monitoring sample $\mathbf{x}_{\text{new}}$.

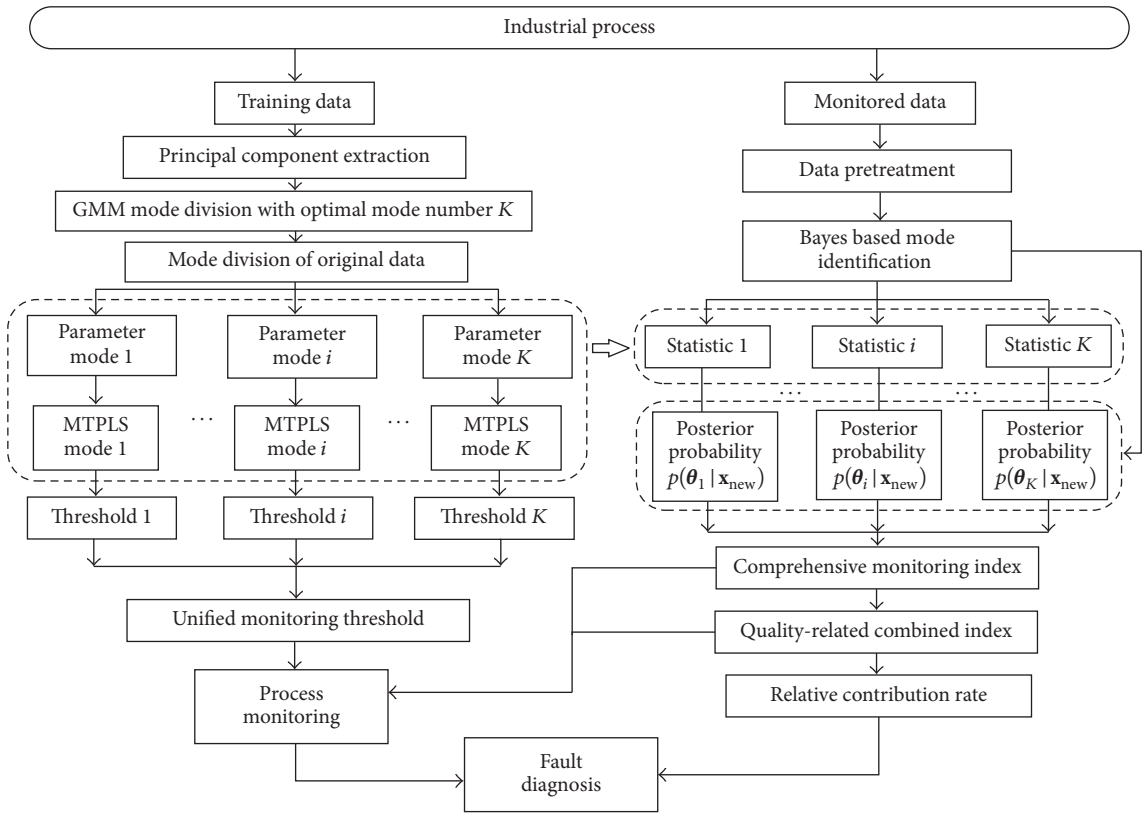


Figure 1: Schematic diagram of the proposed PLS-MTPLS and Bayesian-based process monitoring and diagnosis method.

Due to the diffusion effect of fault, the method of setting absolute control limits using absolute value of variable contribution for fault diagnosis is not with good effect. Therefore, we use relative contribution rate; namely,

$$C_r\left(\mathbf{x}_{\text{new}}, i\right) = \frac{C\left(\mathbf{x}_{\text{new}}, i\right)}{\sum_{i=1}^{m} C\left(\mathbf{x}_{\text{new}}, i\right)}, \tag{35}$$

where relative contribution rate satisfies

$$\sum_{i=1}^{m} C_r\left(\mathbf{x}_{\text{new}}, i\right) = 1. \tag{36}$$

As described above, in index φ_y based quality-related fault diagnosis, the contribution rate can reflect contribution gradients of variables to the monitoring index. Therefore, those variables which have a larger contribution rate are able to affect combined index and quality variables significantly.

4.6. Framework of Fault Detection and Diagnosis. The schematic diagram of the proposed process monitoring and diagnosis is shown in Figure 1. Detailed procedures for multimode process detection can be summarized below:

(1) Collect a set of historical training data under all possible operating modes and determine the number of modes.

(2) Use EM algorithm to learn the Gaussian mixture model of principal component space and estimate the model parameter set $\boldsymbol{\Theta}_{\mathbf{T}}$ based on the iterative steps.

(3) Do multimode division and multimode information acquisition of process data according to C_k. Then, for each monitored sample $\mathbf{x}_{\text{new}}$, compute its posterior probabilities belonging to all Gaussian components through Bayesian inference strategy.

(4) Calculate local monitoring statistics for the monitored sample $\mathbf{x}_{\text{new}}$ within each Gaussian component and integrate them into the comprehensive index with probabilities.

(5) Integrate the quality-related monitoring statistics into a quality-related combined index φ_y.

(6) Specify a confidence level $(1-\alpha)100\%$ for determining control threshold ζ^2 and generate the monitoring plot for all the monitored samples.

(7) Detect the abnormal operating condition at the monitored samples satisfying $\varphi_y > \zeta^2$ which is helpful for fault diagnosis.

(8) Calculate the relative contribution rate of variables to the combined index φ_y before and after fault occurrence and generate the contribution rate plot for fault diagnosis analysis.

5. Application to HSMP

5.1. Hot Strip Mill Process. HSMP (hot strip mill process) is an extremely complex industrial production process. In

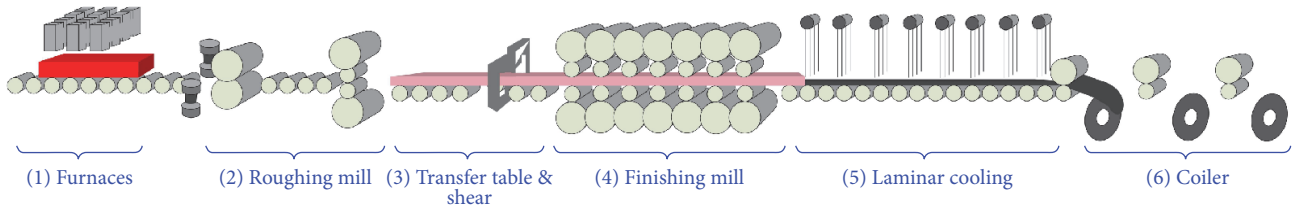
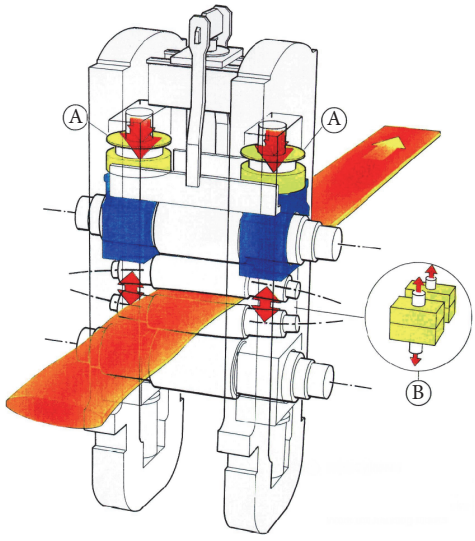


Figure 2: The schematic of HSMP.



Ⓐ HGC cylinder
Ⓑ Work roll bending blocks

Figure 3: The structure of mill stand.

the process of production, improving the quality of products can bring about higher economic and social benefits for the factory. Typical HSM machine production line is mainly composed of reheating furnace, roughing mill, transfer table, crop shear, finishing mill group, run-out table cooling, and coiler. Figure 2 shows the whole process flow chart. The reheating furnace can ensure the temperature of the strip reaches 1200 degrees Celsius before roughing mill. A slab of thickness of 100~200 mm is sent to the roughing mill group after cutting off the scales, eventually forming 28~45 mm thick middle slab through several times of rolling. Through the transport of transfer table and in turn with insulation cover, crop shear, and high pressure water descaling, the slab runs into seven stands of finishing mill group. In order to enhance the performance of the final product, the steel plate needs to go through laminar cooling. This paper focuses on fault monitoring in the part of finishing mill process (FMP).

As shown in Figure 2, FMP consists of seven stands. Every stand contains two working rolls and backup rolls, which are driven by their own power drive units. The distance between two working rolls is called roll gap, which can be adjusted by the hydraulic device. A detailed structure diagram of the finishing roll is shown in Figure 3. This means that the strip will go through all the seven stands during the finishing mill process.

In whole FMP, it is noted that the stands are actually not working independently but are coupled with each other by different control schemes. The thickness in the exit of the last stand is the key factor which directly affects the quality of products. Whole finishing mill process is controlled by automatic thickness control system. It can be seen that there is an obvious hysteresis control of the exit thickness. Not until the abnormal value of the exit thickness is detected, caused by some fault of front stands, can the thickness control system be started. Therefore, establishing real-time acquisition of the relationship between the process variables and exit thickness and then monitoring the thickness by real-time measuring process variables become very meaningful [14].

5.2. Fault Detection Simulation Analysis. Production specification can be determined by different thicknesses of the steel strip in HSMP which should meet different industrial demands. We select the steel plate data of two specifications for modeling: one is the thickness of 2.70 mm and the other is 3.95 mm. The sampling interval for the variable is 0.01 s and 4000 samples are used for training modeling.

In the actual finishing mill process, we can collect the data information including roll gap, milling force between working rolls, and bending force in every stand. Generally speaking, the exit thickness has more relationships with roll gap and milling force than with bending force. Using data collected under normal operating conditions, GMM iterative learning is performed in principal component space which is under the guidance of quality variables. With the model division result, the process of multimode parameters calculation of the original data is carried out. Figure 4 shows the clustering distribution of two kinds of normal production. In this part, the clustering numbers are $K1 = 3$ and $K2 = 5$ which are fixed according to Yu et al. The K-means algorithm is applied to roughly calculate $\boldsymbol{\mu}_i^k$. Randomly initialize the value of $\boldsymbol{\Sigma}_i^k$ before GMM iterative learning. Then, we establish the proposed PLS-MTPLS model. Variables concerned in FMP are as follows.

Process Variables

$\mathbf{x}_1 \sim \mathbf{x}_7$: average gap of F_i stand, $i = 1, \ldots, 7$, μm

$\mathbf{x}_8 \sim \mathbf{x}_{14}$: the force between supporting and working roll in F_i stand, $i = 1, \ldots, 7$, KN

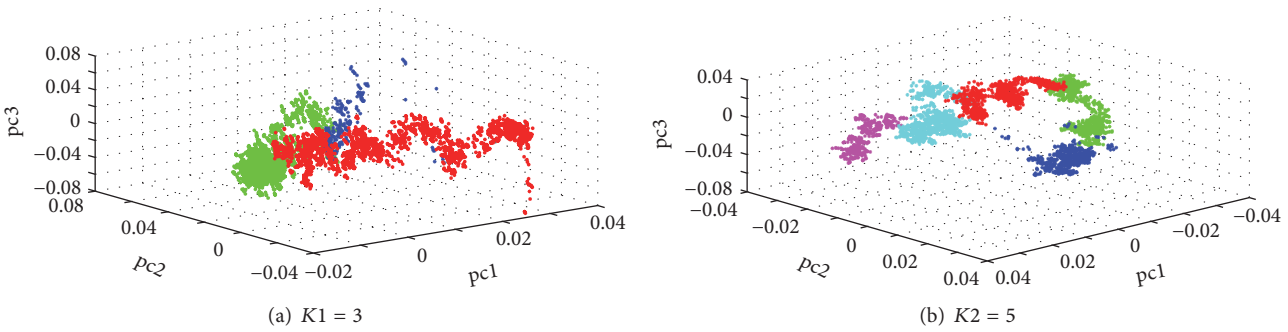


FIGURE 4: Principal components clustering distribution where $K1 = 3$ and $K2 = 5$ (pc: principal component).

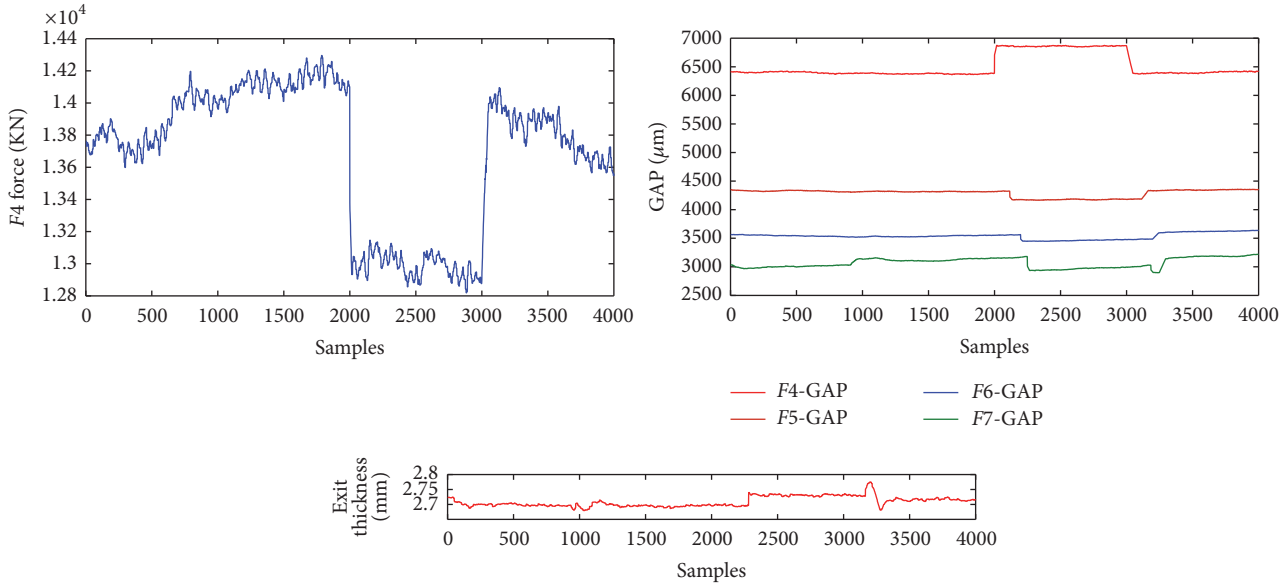


FIGURE 5: Curve of fault source $\mathbf{x}_4$ and affected variables.

$\mathbf{x}_{15} \sim \mathbf{x}_{20}$: the bending force in the working roll of F_i stand, $i = 1, \ldots, 7$, KN

Quality Variable

$\mathbf{y}$: thickness of strip at the exit of FMP, mm

For different types of faults that may occur in FMP, we select three encountered faults as a detected object in this section which are shown in Table 1.

According to the exit thickness value of the strip steel under three types of fault condition, it is obvious that fault 1 and fault 3 are quality-related, while fault 2 is quality-unrelated. As Figures 6 and 9 show, the method based on PLS-MTPLS gives a higher fault detection rate for fault 1 and fault 3. And for the quality-unrelated fault 2, PLS-MTPLS inherits the effect of space division in traditional TPLS method, making the monitoring index ${T_y}^2$ which is directly related to the quality have a relatively low rate of false alarm.

To examine the advantages of our proposed approach, a comparison research has been done using two evaluating indexes: FDR and FAR.

FAR

$$= \frac{\text{Number of samples}\left(\varphi_y > \zeta^2 \mid \text{quality is normal}\right)}{\text{total fault} - \text{free samples}}, \quad (37)$$

FDR

$$= \frac{\text{Number of samples}\left(\varphi_y > \zeta^2 \mid \text{quality is faulty}\right)}{\text{total faulty samples}}.$$

False detection rates and false alarm rates are counted for three types of fault and statistical results are shown in Table 2. It shows that PLS-MTPLS method performs better.

Fault 1 represents the failure of hydraulic roll gap control structure. Fault occurs at about 20 s, namely, the 2000th monitoring sample. The values of roll gap $\mathbf{x}_4$ in the fourth

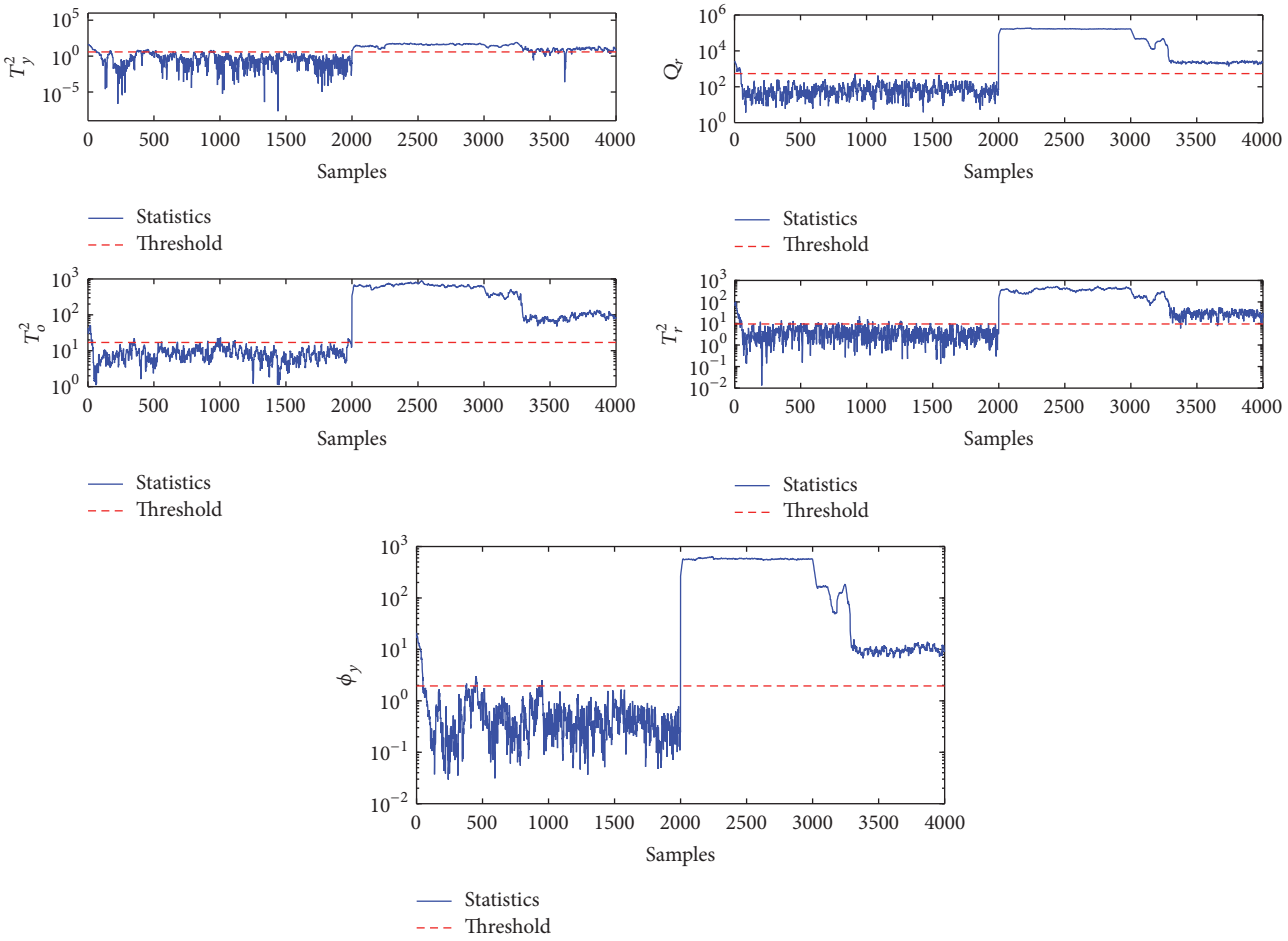


Figure 6: Detection result of fault 1.

Table 1: Faults that occurred in FMP.

Fault number	Fault description	Duration	Type
1	Malfunction of gap control loop in $F4$ stand	20 s–30 s	Quality-related
2	Fault of roll bending force measuring sensor in $F5$ stand	10 s–20 s	Quality-unrelated
3	10% stiction of the cooling valve between $F2$ and $F3$ stands	10 s–20 s	Quality-related

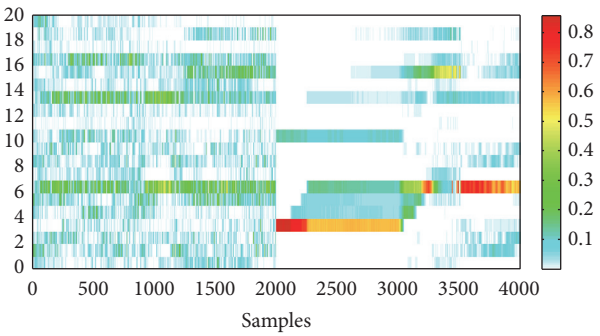


Figure 7: Diagnosis result of fault 1.

stand are directly affected, and then the sampling values of milling force $\mathbf{x}_{11}$ in the fourth stand have also been affected. Because of the influence of feedback control system, roll gaps and milling forces will be changed in the following stands, and then finally the exit thickness is affected. As shown in Figure 5, there is a delay for change of exit thickness value, namely, quality variables with respect to fault occurrence. But for fault detection results, as shown in Figure 6, there is almost no delay. From the point of view of this analysis, detection results can be a good reference for field staffs, in order to take response measurements timely.

Figure 7 gives the observation of change of relative contribution rate for fault 1. As is shown, we can clearly see that many contribution rate values of related variables have changed since the 2000th monitoring sample. When the fault is detected, according to the observation of relative contribution rate, variable $\mathbf{x}_4$ which has the largest relative contribution rate is diagnosed firstly. As a result, we can conclude that roll gap of $F4$ stand is the source of fault. At the same

Table 2: Detection performance comparison.

Fault number	Type	MPLS (T^2)	TPLS (φ_y)	MTPLS ($T_y{}^2$)	MTPLS (φ_y)	PLS-MTPLS ($T_y{}^2$)	PLS-MTPLS (φ_y)
1	FDR	0.7968	0.8977	0.8620	0.9995	0.9435	0.9995
2	FAR	0.3357	0.4335	0.1335	0.2610	0.0080	0.2605
3	FDR	0.7780	0.8510	0.8848	0.9729	0.9092	0.9735

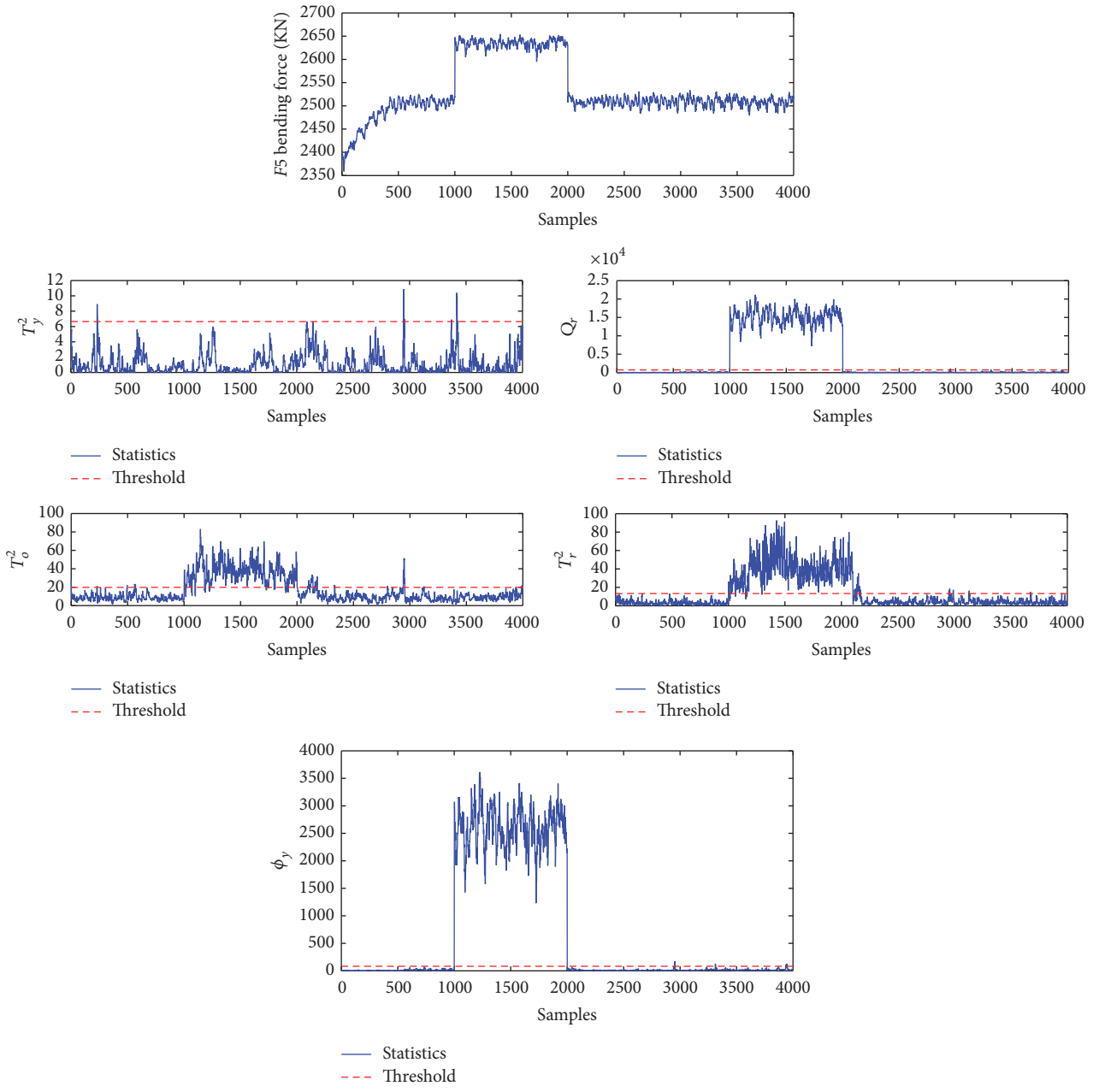


Figure 8: Detection result of fault 2.

time, variables $\mathbf{x}_{11}$ and $\mathbf{x}_5 \sim \mathbf{x}_7$ are subsequently diagnosed which are affected by the fault source, thereby causing fault propagation. Figure 7 shows curves of variables change in real data of fault source variable and variables affected. The diagnostic analysis is in accordance with the actual production situation. As a result, the relative contribution rate can not only diagnose fault variables, but also show the order of fault variables transmission. Then, it can help in finding out the real source of fault with causal relationship among these variables.

Fault 2 represents the fault of sampling value of bending force in $F5$ stand, which is a kind of step transition. When the fault occurs, the value of variable $\mathbf{x}_{18}$ will increase greatly.

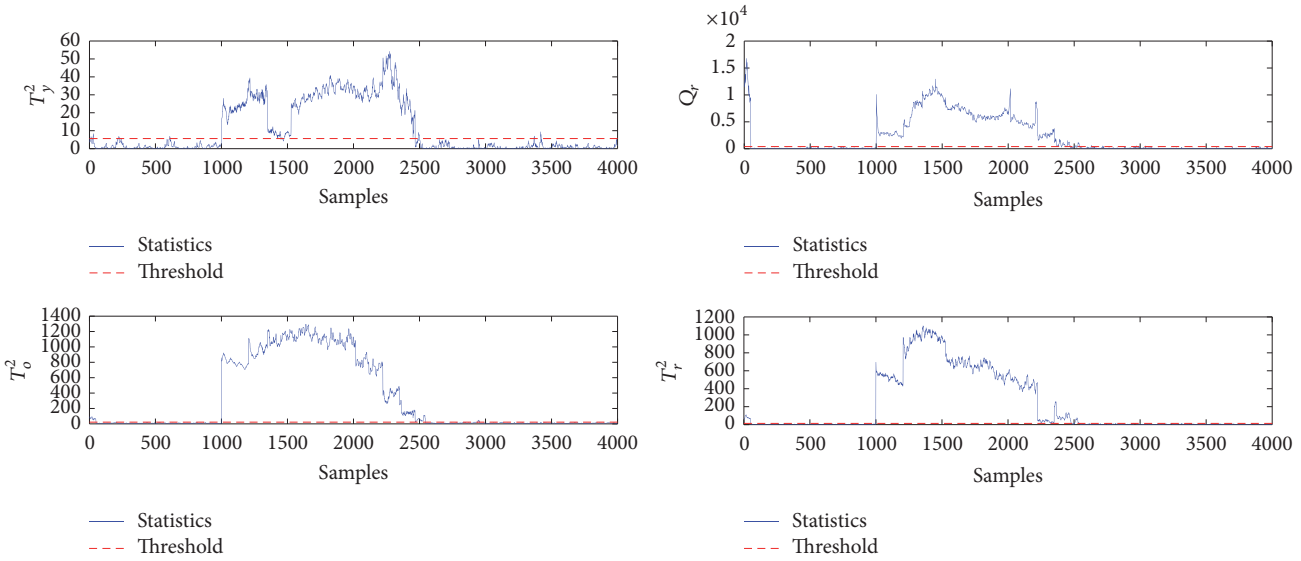


Figure 9: Detection result of fault 3.

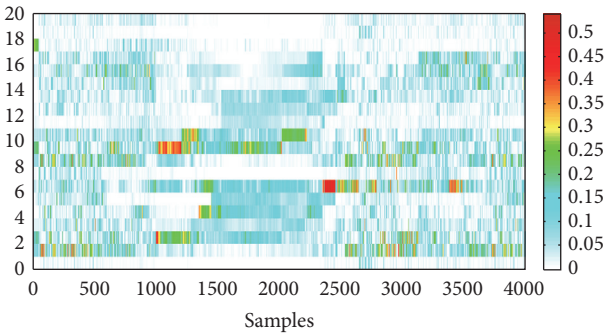


Figure 10: Diagnosis result of fault 3.

Then, with feedback regulation of automatic control system, the bending force value in $F6$ and $F7$ stands will be changed correspondingly. But this kind of fault will only cause the change of strip plate, not thickness as shown in Figure 8.

Fault 3 is a kind of fault in cooling valve between $F2$ and $F3$ stands, which is usual in the process of finishing mill. It will make the rolling force and roll gap in stands following $F3$ stand be changed. Based on the monitoring results of comprehensive index, fault 3 can be detected timely as shown in Figure 9. The change of relative contribution rate is shown in Figure 10. It can be noted that variable $\mathbf{x}_3$ is affected firstly, followed by variable $\mathbf{x}_{10}$ and others. From the above analysis, we can draw a conclusion that the fault diagnosis method based on the relative contribution rate can be applied to FMP effectively.

In this section, we focus on the research of exit thickness of the strip. Twenty variables among measured variables in FMP were selected for building PLS-MTPLS model. Based on it, a kind of comprehensive monitoring index and a kind of relative contribution rate were established for fault monitoring and diagnosis, respectively, for three common faults. Results of monitoring and diagnosis verified that PLS-MTPLS has higher FDR and lower FAR than traditional multivariate statistics methods shown in Table 2. In addition, compared with MTPLS which clusters with original data directly, this method is with better monitoring effects in statistics T_y^2 of principal component which can be seen in Table 2.

6. Conclusion

In this paper, a new PLS-MTPLS method is proposed on the basis of covariance descriptions of PCA and PLS algorithm for multimode process monitoring. After mode division of quality-related principal components, multimode information is embedded into the monitoring model by integrating GMM with TPLS, which avoids the direct use of process training data for modeling. Based on the quality-related multimode monitoring model PLS-MTPLS, a kind of comprehensive monitoring index is applied to execute real-time online monitoring. Then, a combined index is constructed for improving monitoring efficiency and extended to fault diagnosis by relative gradient contribution rate calculation.

The efficiency and superiority of PLS-MTPLS are demonstrated through application to the monitoring of HSMP. As can be seen from the comparison and analysis, the proposed approach can reduce computational complexity and be more suitable for multimode processes.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grants 61473033 and 61673032 and by Beijing Natural Science Foundation (4142035), China.

References

[1] L. H. Chiang, R. D. Braatz, and E. L. Russell, *Fault Detection and Diagnosis in Industrial Systems*, Springer, London, UK, 2001.

[2] D.-H. Hwang and C. Han, "Real-time monitoring for a process with multiple operating modes," *Control Engineering Practice*, vol. 7, no. 7, pp. 891–902, 1999.

[3] S. Lane, E. B. Martin, R. Kooijmans, and A. J. Morris, "Performance monitoring of a multi-product semi-batch process," *Journal of Process Control*, vol. 11, no. 1, pp. 1–11, 2001.

[4] J. Chen and J. Liu, "Mixture principal component analysis models for process monitoring," *Industrial and Engineering Chemistry Research*, vol. 38, no. 4, pp. 1478–1488, 1999.

[5] S. J. Zhao, J. Zhang, and Y. M. Xu, "Monitoring of processes with multiple operating modes through multiple principle component analysis models," *Industrial and Engineering Chemistry Research*, vol. 43, no. 22, pp. 7025–7035, 2004.

[6] X.-T. Doan and R. Srinivasan, "Online monitoring of multiphase batch processes using phase-based multivariate statistical process control," *Computers and Chemical Engineering*, vol. 32, no. 1-2, pp. 230–243, 2008.

[7] S. W. Choi, J. H. Park, and I.-B. Lee, "Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis," *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1377–1387, 2004.

[8] C. K. Yoo, K. Villez, I.-B. Lee, C. Rosén, and P. A. Vanrolleghem, "Multi-model statistical process monitoring and diagnosis of a sequencing batch reactor," *Biotechnology and Bioengineering*, vol. 96, no. 4, pp. 687–701, 2007.

[9] J. Yu and S. J. Qin, "Multiway gaussian mixture model based multiphase batch process monitoring," *Industrial and Engineering Chemistry Research*, vol. 48, no. 18, pp. 8585–8594, 2009.

[10] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice-Hall, New York, NY, USA, 1998.

[11] K. Peng, K. Zhang, B. You, and J. Dong, "Quality-related prediction and monitoring of multi-mode processes using multiple PLS with application to an industrial hot strip mill," *Neurocomputing*, vol. 168, pp. 1094–1103, 2015.

[12] G. Li, S.-Z. Qin, Y.-D. Ji, and D.-H. Zhou, "Total PLS based contribution plots for fault diagnosis," *Acta Automatica Sinica*, vol. 35, no. 6, pp. 759–765, 2009.

[13] K. Zhang, H. Hao, Z. Chen, S. X. Ding, and K. X. Peng, "A comparison and evaluation of key performance indicator-based multivariate statistics process monitoring approaches," *Journal of Process Control*, vol. 33, pp. 112–126, 2015.

[14] H. H. Yue and S. J. Qin, "Reconstruction-based fault identification using a combined index," *Industrial and Engineering Chemistry Research*, vol. 40, no. 20, pp. 4403–4414, 2001.

[15] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring," *Automatica*, vol. 45, no. 7, pp. 1593–1600, 2009.

[16] J. A. Westerhuis, S. P. Gurden, and A. K. Smilde, "Generalized contribution plots in multivariate statistical process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 95–114, 2000.

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 8595960, 13 pages
https://doi.org/10.1155/2017/8595960

Research Article

Fault-Tolerant Control of a Nonlinear System Actuator Fault Based on Sliding Mode Control

Jing He,[1] Lin Mi,[1] Songan Mao,[2] Changfan Zhang,[1] and Houguang Chu[1]

[1] *College of Electrical and Information Engineering, Hunan University of Technology, Zhuzhou, Hunan 412007, China*
[2] *School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA*

Correspondence should be addressed to Changfan Zhang; zhangchangfan@263.net

Received 7 December 2016; Revised 28 January 2017; Accepted 5 February 2017; Published 9 March 2017

Academic Editor: Xiao He

This paper presents a fault-tolerant control scheme for a class of nonlinear systems with actuator faults and unknown input disturbances. First, the sliding mode control law is designed based on the reaching law method. Then, in view of unpredictable state variables and unknown information in the control law, the original system is transformed into two subsystems through a coordinate transformation. One subsystem only has actuator faults, and the other subsystem has both actuator faults and disturbances. A sliding mode observer is designed for the two subsystems, respectively, and the equivalence principle of the sliding mode variable structure is used to realize the accurate reconstruction of the actuator faults and disturbances. Finally, the observation value and the reconstruction value are used to carry out an online adjustment to the designed sliding mode control law, and fault-tolerant control of the system is realized. The simulation results are presented to demonstrate the approach.

1. Introduction

In recent years, theoretical research in fault-tolerant control has made great progress in the practical applications [1–8]. Fault-tolerant control scheme is widely studied in linear systems. Tao and Xu studied the fault-tolerant control of known and unknown parameters of high-speed train dynamics model [9, 10]. Yu and Jiang proposed an innovative strategy for compensating the actuator faults to optimize system performance [11]. Shen et al. presented an integrated design method of adaptive robust control for a linear system with adaptive fault identification [12]. And Zhao et al. studied an adaptive sliding mode control for damage problems [13]. However, most of the actual objects are nonlinear, and some of the working points in the linear systems will enter the nonlinear region when it has a fault. For systems of actuator with random failures and uncertain parameters, Fan et al. studied its stabilization and tracking problem, but they did not consider the disturbance of unknown inputs [14]. Yin et al. presented a fault-tolerant control system scheme for real-time performance optimization, but they did not consider the effect of actuator faults [15]. In view of uncertain overdrive systems, Zhang et al. proposed a robust control allocation algorithm based on pseudoinverse, which is compensated for the negative influence of the failure and stuck fault [16]. Hu et al. presented an adaptive terminal sliding mode control method, found the finite time control of the attitude tracking, and solved the problem of actuator control input saturation [17]. However, these two methods did not obtain accurate fault values.

The concept of sliding mode variable structure control is to design the switching hyperplane of the system according to the expected dynamic characteristics of the system. The variable structure controller is used to drive the system state from the initial state to the switching hyperplane in a finite time and then maintain its state at the switching hyperplane. Once the system state reaches the switching hyperplane, the control function will ensure that the system travels along the switching hyperplane to reach the origin of the system. The system characteristics and parameters are entirely dependent on the designed switching hyperplane but are unrelated to external disturbances. Hence, the sliding mode variable structure control is extremely robust and has been widely applied in studies of nonlinear systems.

Based on the work of Zhao et al. about the adaptive sliding mode control in the literature [13], considering a class of nonlinear systems with actuator faults and unknown input disturbances, this paper innovatively presents precisely a fault-tolerant control method with disturbance and fault reconstruction. To fulfill the above scheme, a fault-tolerant control law with sliding mode control is first proposed here. A fault diagnosis and reconstruction method is used to carry out accurate reconstruction of unknown information, and finally the observation value and the reconstruction value are used to carry out the corresponding adjustment of the control law. The simulation results show that the proposed control method can meet the requirements of control accurately and reliably.

The remainder of the paper is organized as follows. Section 2 describes the mathematical model of nonlinear systems with actuator faults and unknown input disturbances. Section 3 presents a fault-tolerant control law based on the sliding mode control. The design of observer is discussed in Section 4. Then the fault reconstruction and the disturbance estimation are discussed in Section 5. Section 6 presents an online adjustment of the fault-tolerant control law. Simulation results are presented in Section 7, and conclusion is in Section 8.

2. Problem Description

We have an uncertain nonlinear system affected by actuator faults and unknown disturbances:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \Phi(x,u) + Ef_a(t) + Dd(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{1}$$

where $x \in R^n$ is the state variables, $u \in R^r$ is the system inputs, and $y \in R^p$ is the outputs. Assume the nonlinear continuous term $\Phi(x,u) \in R^n$ is known. The unknown nonlinear term $d(t) \in R^q$ models the lumped uncertainties and disturbances experienced by the system, which is assumed to be bounded; that is, a positive constant γ_1 exists such that $\|d(t)\| \le \gamma_1$. The unknown nonlinear term $f_a(t) \in R^q$ represents actuator faults, supposed to be norm bounded; that is, there exists a constant γ_2 such that $\|f_a(t)\| \le \gamma_2$. $A \in R^{n\times n}$, $B \in R^{n\times r}$, $C \in R^{p\times n}$, $D \in R^{n\times q}$, and $E \in R^{n\times q}$ are known constant matrices with $n > p > q$.

3. A Fault-Tolerant Control Law

The active fault-tolerant control for faulty system (1) can be described as follows: If there is no fault, that is, $f_a(t) = 0$, it is considered to be a nominal system; any appropriate control law is designed to achieve the gradual stability of the nominal system. If there is a fault, that is, $f_a(t) \neq 0$, an additional control law u_f is designed to address and fix the fault. Thus, for a faulty system, the control law can be designed:

$$u(t) = u_n(t) + u_f(t), \tag{2}$$

where $u_n(t)$ is the nominal control law for the corresponding fault-free system.

If the specified system state command is defined as $r(t)$, $r(t) \in R^n$, the system error can be represented as

$$e_x = r(t) - x(t). \tag{3}$$

Therefore, the sliding mode surface is represented as

$$s_0 = Ge_x, \tag{4}$$

where G is a matrix that will be designed.

From (4), the following equation can be obtained:

$$\begin{aligned} \dot{s}_0 &= G\dot{e}_x = G(\dot{r} - \dot{x}) \\ &= G\dot{r} \\ &\quad - G\left(Ax(t) + \Phi(x,u) + Ef_a(t) + Dd(t) + Bu(t)\right). \end{aligned} \tag{5}$$

The control law is designed as

$$\begin{aligned} u = (GB)^* &\left(G\dot{r} - GAx - G\Phi(x,u) - GEf_a(t)\right. \\ &\left. - GDd(t) + ks_0 + \varepsilon \operatorname{sgn}(s_0)\right), \end{aligned} \tag{6}$$

where parameters k and ε are constant to be designed and $(GB)^*$ is the generalized inverse matrix of (GB). If there is no fault, that is, $f_a(t) = 0$, the control law of the nominal system is

$$\begin{aligned} u_n(t) = (GB)^* &\left(G\dot{r} - GAx - G\Phi(x,u) - GDd(t) + ks_0\right. \\ &\left. + \varepsilon \operatorname{sgn}(s_0)\right). \end{aligned} \tag{7}$$

If there is a fault, that is, $f_a(t) \neq 0$, the additional control law u_f is

$$u_f(t) = (GB)^* \left(-GEf_a(t)\right). \tag{8}$$

Lemma 1. *For the nonlinear system described in (1), the designed fault-tolerant control is shown in (6). When $k > 0$ and $\varepsilon > 0$, the system is in the stability condition of a sliding mode.*

Proof. Consider the following Lyapunov function:

$$V_0 = \frac{1}{2}{s_0}^2. \tag{9}$$

The time derivative of V_0, along with (5), is

$$\begin{aligned} \dot{V}_0 = s_0\dot{s}_0 = s &\left(G\dot{r} - G\left(Ax(t) + \Phi(x,u) + Ef_a(t)\right.\right. \\ &\left.\left. + Dd(t) + Bu(t)\right)\right). \end{aligned} \tag{10}$$

The following equation can be obtained by taking the substitution of (6) into (10):

$$\dot{V}_0 = -k{s_0}^2 - \varepsilon\left|s_0\right|. \tag{11}$$

When $k > 0$ and $\varepsilon > 0$, the following equation can be obtained:

$$\dot{V}_0 \le 0. \tag{12}$$

This completes the proof. □

It can be seen from the above analysis that the system can satisfy the asymptotic stability requirement; namely, it will be driven to the corresponding sliding mode surface in a finite period of time.

4. Observer Design

Equation (6) contains the system state variable x, the disturbance d, and the actuator fault f_a, where these variables are often unknown in control law. Therefore, we carry out the system state observations x through the method of "designing observer," where the disturbance d and actuator fault f_a are reconstructed accurately again on this basis to obtain the corresponding state observation value $\widehat{x}$ and reconstruction values $\widehat{d}$ and $\widehat{f}_a$ of the disturbance and fault, which are substituted into (6) to determine the control.

Assumption 2. D is a column full rank matrix and $\mathrm{rank}(CD) = \mathrm{rank}(D)$.

Remark 3. If the disturbance distribution matrix D is not a column full rank matrix, for example, $\mathrm{rank}(D) = q_1 < q$, then a rank decomposition $D = D_1' D_2'$ could be applied, where D_1' is a column full rank matrix and $d_1'(t) = D_2' d_2'(t)$ could be considered as a new unknown disturbance [18].

Assumption 4. The matrix pair $(A,\ C)$ is observable.

The matrix is partitioned for (1) to obtain the following equation:

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix}\\ &= \begin{bmatrix}A_{11}A_{12}A_{21}A_{22}\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}\Phi_1(x,u)\\ \Phi_2(x,u)\end{bmatrix}\\ &\quad + \begin{bmatrix}E_1\\ E_2\end{bmatrix} f(t) + \begin{bmatrix}D_1\\ D_2\end{bmatrix} d(t) + \begin{bmatrix}B_1\\ B_2\end{bmatrix} u(t),\end{aligned} \tag{13}$$

where $x_1(t) \in R^{n-q}$, $x_2(t) \in R^q$, $A_{11} \in R^{(n-q)\times(n-q)}$, $A_{12} \in R^{(n-q)\times q}$, $A_{21} \in R^{q\times(n-q)}$, $A_{22} \in R^{q\times q}$, $\Phi_1(x,u) \in R^{n-q}$, $\Phi_2(x,u) \in R^q$, $E_1 \in R^{n-q}$, $E_2 \in R^q$, $D_1 \in R^{(n-q)\times q}$, and $D_2 \in R^{q\times q}$ is a nonsingular matrix.

Based on Assumption 2, two transformation matrices, namely, T and S, exist [19] such that

$$x(t) = T^{-1}z(t) = T^{-1}\begin{bmatrix}z_1(t)\\ z_2(t)\end{bmatrix} \quad y(t) = S^{-1}\begin{bmatrix}v_1(t)\\ v_2(t)\end{bmatrix}. \tag{14}$$

Therefore, (1) could be converted as follows:

$$\begin{aligned}\dot{z}(t) &= \begin{bmatrix}\dot{z}_1(t)\\ \dot{z}_2(t)\end{bmatrix}\\ &= TAT^{-1}z(t) + T\Phi(z,u) + TEf_a(t) + TDd(t)\\ &\quad + TBu(t),\\ v(t) &= \begin{bmatrix}v_1(t)\\ v_2(t)\end{bmatrix} = SCT^{-1}z(t),\end{aligned} \tag{15}$$

where

$$SCT^{-1} = \begin{bmatrix}C_{11} & 0\\ 0 & C_{22}\end{bmatrix} \tag{16}$$

and C_{22} is a nonsingular matrix.

The following nonsingular transformation matrix T is then constructed [19]:

$$T = \left[I_{n-q} - D_1 D_2^{-1} 0 I_q\right]. \tag{17}$$

Therefore, the coefficient matrices in (15) are as follows:

$$\begin{aligned}TAT^{-1} &= \overline{A} = \left[\overline{A}_{11}\overline{A}_{12}\overline{A}_{21}\overline{A}_{22}\right],\\ TB &= \overline{B} = \begin{bmatrix}\overline{B}_1\\ \overline{B}_2\end{bmatrix},\\ TD &= \overline{D} = \begin{bmatrix}0\\ \overline{D}_2\end{bmatrix},\\ TE &= \overline{E} = \begin{bmatrix}\overline{E}_1\\ \overline{E}_2\end{bmatrix},\\ T\Phi(x,u) &= \overline{\Phi}(x,u) = \begin{bmatrix}\overline{\Phi}_1(x,u)\\ \overline{\Phi}_2(x,u)\end{bmatrix},\end{aligned} \tag{18}$$

where $\overline{A}_{11} = A_{11} - D_1 D_2^{-1} A_{21}$, $\overline{A}_{12} = (A_{11} - D_1 D_2^{-1} A_{21}) D_1 D_2^{-1} + A_{12} - D_1 D_2^{-1} A_{22}$, $\overline{A}_{21} = A_{21}$, $\overline{A}_{22} = A_{21} D_1 D_2^{-1} + A_{22}$, $\overline{\Phi}_1(x,u) = \Phi_1(x,u) - D_1 D_2^{-1}\Phi_2(x,u)$, $\overline{\Phi}_2(x,u) = \Phi_2(x,u)$, $\overline{E}_1 = E_1 - D_1 D_2^{-1} E_2$, $\overline{E}_2 = E_2$, $\overline{D}_2 = D_2$, $\overline{B}_1 = B_1 - D_1 D_2^{-1} B_2$, $\overline{B}_2 = B_2$, $z_1(t) \in R^{n-q}$, $z_2(t) \in R^q$, $\overline{A}_{11} \in R^{(n-q)\times(n-q)}$, $\overline{A}_{12} \in R^{(n-q)\times q}$, $\overline{A}_{21} \in R^{q\times(n-q)}$, $\overline{A}_{22} \in R^{q\times q}$, $\overline{\Phi}_1(x,u) \in R^{n-q}$, $\overline{\Phi}_2(x,u) \in R^q$, $\overline{E}_1 \in R^{n-q}$, $\overline{E}_2 \in R^q$, $\overline{D}_2 \in R^q$, $C_{11} \in R^{(p-q)\times(n-q)}$, $C_{22} \in R^{q\times q}$, $v_1 \in R^{p-q}$, and $v_2 \in R^q$.

System (15) is then transformed into the following subsystems:

$$\begin{aligned}\dot{z}_1(t) &= \overline{A}_{11}z_1(t) + \overline{A}_{12}z_2(t) + \overline{\Phi}_1(x,u) + \overline{E}_1 f_a(t)\\ &\quad + \overline{B}_1 u(t),\end{aligned} \tag{19}$$

$$v_1(t) = C_{11}z_1(t),$$

$$\begin{aligned}\dot{z}_2(t) &= \overline{A}_{21}z_1(t) + \overline{A}_{22}z_2(t) + \overline{\Phi}_2(x,u) + \overline{E}_2 f_a(t)\\ &\quad + \overline{D}_2 d(t) + \overline{B}_2 u(t),\end{aligned} \tag{20}$$

$$v_2(t) = C_{22}z_2(t). \tag{21}$$

Two subsystems, (19) and (20), could be obtained from system (1) by matrix transformation. Subsystem (19) is free from any uncertainties but is subjected to system actuator faults; subsystem (20) has actuator faults and uncertainties. The detail design of two sliding mode observers corresponding to the above subsystems will be presented as follows.

Assumption 5. $(\overline{A}_{11}, C_{11})$ and $(\overline{A}_{22}, C_{22})$ are observable.

Assumption 6. The nonlinear functions $\overline{\Phi}_1$ and $\overline{\Phi}_2$ are assumed to be known and satisfied the Lipschitz conditions, such that

$$\begin{aligned}\left\|\overline{\Phi}_1(x,u)-\overline{\Phi}_1(\widehat{x},u)\right\| &\le \overline{\gamma}_3\|x-\widehat{x}\| \\ &= \overline{\gamma}_3\left\|T^{-1}\right\|\|z-\widehat{z}\| = \gamma_3\|z-\widehat{z}\|, \\ \left\|\overline{\Phi}_2(x,u)-\overline{\Phi}_2(\widehat{x},u)\right\| &\le \overline{\gamma}_4\|x-\widehat{x}\| \\ &= \overline{\gamma}_4\left\|T^{-1}\right\|\|z-\widehat{z}\| = \gamma_4\|z-\widehat{z}\|,\end{aligned} \tag{22}$$

where γ_3 and γ_4 are known Lipschitz positive constants.

Based on the transformed systems (19) and (20), the present study proposes the following two sliding mode observers:

$$\begin{aligned}\dot{\widehat{z}}_1(t) &= \overline{A}_{11}\widehat{z}_1(t)+\overline{A}_{12}\widehat{z}_2(t)+\overline{\Phi}_1(\widehat{x},u)+\overline{E}_1 r_1(t) \\ &\quad +\overline{B}_1 u(t)+L_1\left(v_1(t)-\widehat{v}_1(t)\right), \\ \widehat{v}_1(t) &= C_{11}\widehat{z}_1(t), \\ \dot{\widehat{z}}_2(t) &= \overline{A}_{21}\widehat{z}_1(t)+\overline{A}_{22}\widehat{z}_2(t)+\overline{\Phi}_2(\widehat{x},u)+\overline{E}_2 r_2(t) \\ &\quad +\overline{B}_2 u(t)+L_2\left(v_2(t)-\widehat{v}_2(t)\right), \\ \widehat{v}_2(t) &= C_{22}\widehat{z}_2(t),\end{aligned} \tag{23}$$

where superscript "$\widehat{\ }$" indicates an estimated value and $r_1(t)$ and $r_2(t)$ represent the input signals of the sliding mode observers:

$$\begin{aligned}r_1(t) &= \begin{cases}-\rho_1\dfrac{F_1\left(\widehat{v}_1(t)-v_1(t)\right)}{\left\|F_1\left(\widehat{v}_1(t)-v_1(t)\right)\right\|} & \text{if } \widehat{v}_1(t)-v_1(t)\\ \neq 00 & \text{if } \widehat{v}_1(t)-v_1(t)=0,\end{cases} \\ r_2(t) &= \begin{cases}-\rho_2\dfrac{F_2\left(\widehat{v}_2(t)-v_2(t)\right)}{\left\|F_2\left(\widehat{v}_2(t)-v_2(t)\right)\right\|} & \text{if } \widehat{v}_2(t)-v_2(t)\\ \neq 00 & \text{if } \widehat{v}_2(t)-v_2(t)=0,\end{cases}\end{aligned} \tag{24}$$

where F_1 and F_2 are the observer gains and ρ_1 and ρ_2 are positive scalars that will be designed.

Assumption 7. The existing arbitrary matrices F_1 and F_2 and symmetric positive definite matrices P_1 and P_2 will satisfy the following equations:

$$\begin{aligned}P_1\overline{E}_1 &= C_{11}^T F_1^T, \\ P_2\overline{E}_2 &= C_{22}^T F_2^T.\end{aligned} \tag{25}$$

From Assumption 5 we know that the existing matrices L_1 and L_2 will make A_{10} and A_{20} become stable matrices:

$$\begin{aligned}\overline{A}_{11}-L_1C_{11} &= A_{10}, \\ \overline{A}_{22}-L_2C_{22} &= A_{20}.\end{aligned} \tag{26}$$

If the state estimation errors are defined as $e_1 = z_1(t)-\widehat{z}_1(t)$, $e_2 = z_2(t)-\widehat{z}_2(t)$, and $e = \begin{pmatrix}e_1 & e_2\end{pmatrix}^T$ and the output estimation errors are defined as $e_{v1} = v_1(t)-\widehat{v}_1(t) = C_{11}e_1(t)$ and $e_{v2} = v_2(t)-\widehat{v}_2(t) = C_{22}e_2(t)$, then, from (19), (20), and (23), the state estimation errors dynamical systems are described by

$$\begin{aligned}\dot{e}_1(t) &= \left(\overline{A}_{11}-L_1C_{11}\right)e_1(t)+\overline{A}_{12}e_2(t)+\overline{\Phi}_1(z,u) \\ &\quad -\overline{\Phi}_2(\widehat{z},u)+\overline{E}_1\left(f_a(t)-r_1(t)\right),\end{aligned} \tag{27}$$

$$\begin{aligned}\dot{e}_2(t) &= \left(\overline{A}_{22}-L_2C_{22}\right)e_1(t)+\overline{A}_{21}e_1(t)+\overline{\Phi}_2(z,u) \\ &\quad -\overline{\Phi}_2(\widehat{z},u)+\overline{E}_2\left(f_a(t)-r_2(t)\right) \\ &\quad +\overline{D}_2 d(t).\end{aligned} \tag{28}$$

Lemma 8. *Considering the error dynamics system (27) and (28) and Assumptions 6 and 7, if the following LMI is satisfied*

$$\begin{bmatrix}A_{10}^TP_1+P_1A_{10}+\xi\gamma^2I_{n-q} & \overline{A}_{21}^TP_2+P_1\overline{A}_{12} & P_1 & 0 \\ \overline{A}_{12}^TP_1+P_2\overline{A}_{21} & A_{20}^TP_2+P_2A_{20}+\xi\gamma^2I_q & 0 & P_2 \\ P_1 & 0 & -\xi I_{n-q} & 0 \\ 0 & P_2 & 0 & -\xi I_q\end{bmatrix}<0 \tag{29}$$

and the parameters ρ_1 and ρ_2 satisfy

$$\begin{aligned}\rho_1 &> \gamma_2, \\ \rho_2 &> \gamma_4+\frac{\left\|\overline{D}_2\right\|}{\left\|\overline{E}_2\right\|}\gamma_1\end{aligned} \tag{30}$$

then e_1 and e_2 are asymptotically convergent; that is,

$$\begin{aligned}\lim_{t\to\infty}e_1(t) &= 0, \\ \lim_{t\to\infty}e_2(t) &= 0,\end{aligned} \tag{31}$$

where ξ is a positive constant and I_q is a q-dimensional identity matrix.

Proof. Consider the following Lyapunov function:

$$V_1 = e_1^T P_1 e_1 + e_2^T P_2 e_2. \tag{32}$$

The derivative of V along with the error dynamic systems (27) and (28) is

$$\begin{aligned}\dot{V}_1 &= e_1^T \left(A_{10}^T P_1 + P_1 A_{10}\right) e_1 + 2e_1^T P_1 \left(\overline{A}_{12} e_2 \right.\\ &\quad \left. + \overline{\Phi}_1 (z,u) - \overline{\Phi}_1 (\hat{z},u) + \overline{E}_1 f_a (t) - \overline{E}_1 r_1 (t)\right)\\ &\quad + e_2^T \left(A_{20}^T P_2 + P_2 A_{20}\right) e_2 + 2e_2^T P_2 \left(\overline{A}_{21} e_1 \right.\\ &\quad \left. + \overline{\Phi}_2 (z,u) - \overline{\Phi}_2 (\hat{z},u) + \overline{E}_2 f_a (t) + \overline{D}_2 d (t)\right.\\ &\quad \left. - \overline{E}_2 r_2 (t)\right).\end{aligned} \tag{33}$$

Let

$$\begin{aligned}\overline{A} &= \left[\overline{A}_{11} - L_1 C_{11} \overline{A}_{12} \overline{A}_{21} \overline{A}_{22} - L_2 C_{22}\right],\\ P &= \left[P_1 00 P_2\right];\end{aligned} \tag{34}$$

then

$$\begin{aligned}\dot{V}_1 &= e^T \left(\overline{A}^T P + P\overline{A}\right) e + 2e^T P \left(\overline{\Phi} (z,u) - \overline{\Phi} (\hat{z},u)\right)\\ &\quad + 2e_1^T P_1 \left(\overline{E}_1 f_a (t) - \overline{E}_1 r_1 (t)\right)\\ &\quad + 2e_2^T P_2 \left(\overline{E}_2 f_a (t) + \overline{D}_2 d (t) - \overline{E}_2 r_2 (t)\right).\end{aligned} \tag{35}$$

Since the inequality $2X^T Y \le (1/\xi) X^T X + \xi Y^T Y$ is true for any scalar $\xi > 0$, then

$$\begin{aligned}&2e^T P \left(\overline{\Phi} (z,u) - \overline{\Phi} (\hat{z},u)\right)\\ &\quad \le \frac{1}{\xi} e^T P^2 e\\ &\qquad + \xi \left(\overline{\Phi} (z,u) - \overline{\Phi} (\hat{z},u)\right)^T \left(\overline{\Phi} (z,u) - \overline{\Phi} (\hat{z},u)\right)\\ &\quad \le \frac{1}{\xi} e^T P^2 e + \xi \gamma^2 \|e\|^2.\end{aligned} \tag{36}$$

From Assumption 7, the following equation can be obtained:

$$\begin{aligned}&2e_1^T P_1 \overline{E}_1 f_a (t) - 2e_1^T P_1 \overline{E}_1 r_1 (t)\\ &\quad = 2e_1^T P_1 \overline{E}_1 f_a (t) - 2\rho_1 e_1^T P_1 \overline{E}_1 \frac{F_1 C_{11} e_1}{\|F_1 C_{11} e_1\|}\\ &\quad \le 2 \|F_1 C_{11}\| \|e_1\| (\gamma_2 - \rho_1) \le 0,\\ &2e_2^T P_2 \overline{E}_2 f_a (t) + 2e_2^T P_2 \overline{D}_2 d (t) - 2e_2^T P_2 \overline{E}_2 r_2 (t)\\ &\quad = 2e_2^T P_2 \overline{E}_2 f_a (t) + 2e_2^T P_2 \overline{D}_2 d (t)\\ &\qquad - 2\rho_2 e_2^T P_2 \overline{E}_2 \frac{F_2 C_{22} e_2}{\|F_2 C_{22} e_2\|}\\ &\quad \le 2 \|F_2 C_{22}\| \|e_2\| \left(\gamma_2 + \frac{\|\overline{D}_2\|}{\|\overline{E}_2\|} \gamma_1 - \rho_2\right) \le 0.\end{aligned} \tag{37}$$

Equations (36) to (37) are substituted into (33) to obtain

$$\dot{V}_1 \le e^T \left(\overline{A}^T P + P\overline{A} + \frac{1}{\xi} P^2 + \xi \gamma^2 I_n\right) e. \tag{38}$$

$\dot{V}_1$ turns out to be negative definite by imposing

$$\overline{A}^T P + P\overline{A} + \frac{1}{\xi} P^2 + \xi \gamma^2 I_n < 0. \tag{39}$$

The linear matrix inequality is satisfied

$$\left[\overline{A}^T P + P\overline{A} + \xi \gamma^2 I_n P P - \xi I_n\right] < 0 \tag{40}$$

such that

$$\begin{bmatrix} A_{10}^T P_1 + P_1 A_{10} + \xi\gamma^2 I_{n-q} & \overline{A}_{21}^T P_2 + P_1 \overline{A}_{12} & P_1 & 0\\ \overline{A}_{12}^T P_1 + P_2 \overline{A}_{21} & A_{20}^T P_2 + P_2 A_{20} + \xi\gamma^2 I_q & 0 & P_2\\ P_1 & 0 & -\xi I_{n-q} & 0\\ 0 & P_2 & 0 & -\xi I_q \end{bmatrix} < 0, \tag{41}$$

so that $e(t)$ will make a global asymptotic convergence to zero; that is,

$$\begin{aligned}\lim_{t\to\infty} e_1 (t) &= 0,\\ \lim_{t\to\infty} e_2 (t) &= 0.\end{aligned} \tag{42}$$

This completes the proof. □

Remark 9. Lemma 8 implies that e_1 and e_2 are bounded; that is, t_0 will exist if $t > t_0$

$$\begin{aligned}\|e_1\| &\le \delta_1,\\ \|e_2\| &\le \delta_2,\\ \|e\| &\le \delta,\end{aligned} \tag{43}$$

where δ_1, δ_2, and δ are three finite positive scalars.

Lemma 10. *Choose the sliding mode surfaces $s_1 = F_1 e_{v1}$ and $s_2 = F_2 e_{v2}$. Suppose Assumptions 6 and 7 and inequality (43) hold; besides, ρ_i $(i = 1, 2)$ is chosen sufficiently large to satisfy*

$$\rho_1 \geq \frac{\left(\left\|\overline{A}_{11} - L_1 C_{11}\right\| + \gamma_3\right)\delta_1 + \left(\left\|\overline{A}_{12}\right\| + \gamma_3\right)\delta_2 + \left\|\overline{E}_1\right\|\gamma_2 + K_1}{\lambda_{\min}\left(\overline{E}_1^T P_1 \overline{E}_1\right)},$$
$$\rho_2 \geq \frac{\left\|F_2 C_{22}\right\|\left(\left(\left\|\overline{A}_{22} - L_2 C_{22}\right\| + \gamma_4\right)\delta_2 + \left(\left\|\overline{A}_{21}\right\| + \gamma_4\right)\delta_1 + \left\|\overline{E}_2\right\|\gamma_2 + \left\|D_2\right\|\gamma_1\right) + K_2}{\lambda_{\min}\left(\overline{E}_2^T P_2 \overline{E}_2\right)}. \tag{44}$$

Then the error system (27) and (28) will be driven to the corresponding sliding mode surface $s_i = 0$ $(i = 1, 2)$ in finite time. K_1 and K_2 are positive constants.

Proof. (1) We select the Lyapunov function as follows:

$$V_2 = \frac{1}{2} s_1^T s_1. \tag{45}$$

The time derivative of V_2 along the trajectories of system (27) is given by

$$\begin{aligned}
\dot{V}_2 &= s_1^T\left(F_1 C_{11}\dot{e}_1\right) = s_1^T F_1 C_{11}\left(\left(\overline{A}_{11} - L_1 C_{11}\right)e_1(t)\right.\\
&\quad + \overline{A}_{12} e_2(t) + \overline{\Phi}_1(z,u) - \overline{\Phi}_1(\hat{z},u) + \overline{E}_1 f_a(t)\\
&\quad \left. - \overline{E}_1 r_1(t)\right) = s_1^T\left(F_1 C_{11}\dot{e}_1\right)\\
&= s_1^T F_1 C_{11}\left(\left(\overline{A}_{11} - L_1 C_{11}\right)e_1(t) + \overline{A}_{12} e_2(t)\right.\\
&\quad \left. + \overline{\Phi}_1(z,u) - \overline{\Phi}_1(\hat{z},u) + \overline{E}_1 f_a(t)\right) - F_1 C_{11}\overline{E}_1 r_1(t)\\
&= s_1^T F_1 C_{11}\left(\left(\overline{A}_{11} - L_1 C_{11}\right)e_1(t) + \overline{A}_{12} e_2(t)\right.\\
&\quad \left. + \overline{\Phi}_1(z,u) - \overline{\Phi}_1(\hat{z},u) + \overline{E}_1 f_a(t)\right) - \rho_1 s_1^T \overline{E}_1^T P_1 \overline{E}_1\\
&\quad \cdot \frac{s_1}{\left\|s_1\right\|}.
\end{aligned} \tag{46}$$

From $\lambda_{\min}(\overline{E}_1^T P_1 \overline{E}_1)\|s_1\|^2 \leq s_1^T \overline{E}_1^T P_1 \overline{E}_1 s_1$ and (43), it follows that

$$\begin{aligned}
\dot{V}_2 &\leq s_1^T F_1 C_{11}\left(\left(\overline{A}_{11} - L_1 C_{11}\right)e_1(t) + \overline{A}_{12} e_2(t)\right.\\
&\quad \left. + \overline{\Phi}_1(z,u) - \overline{\Phi}_1(\hat{z},u) + \overline{E}_1 f_a(t)\right)\\
&\quad - \rho_1 \lambda_{\min}\left(\overline{E}_1^T P_1 \overline{E}_1\right)\left\|s_1\right\| \leq \left\|s_1\right\|\left(\left\|F_1 C_{11}\right\|\right.\\
&\quad \cdot \left(\left(\left\|\overline{A}_{11} - L_1 C_{11}\right\| + \gamma_3\right)\delta_1 + \left(\left\|\overline{A}_{12}\right\| + \gamma_3\right)\delta_2\right.\\
&\quad \left.\left. + \left\|\overline{E}_1\right\|\gamma_2\right) - \rho_1 \lambda_{\min}\left(\overline{E}_1^T P_1 \overline{E}_1\right)\right).
\end{aligned} \tag{47}$$

We design ρ_1 which satisfies

$$\rho_1 \geq \frac{\left(\left\|\overline{A}_{11} - L_1 C_{11}\right\| + \gamma_3\right)\delta_1 + \left(\left\|\overline{A}_{12}\right\| + \gamma_3\right)\delta_2 + \left\|\overline{E}_1\right\|\gamma_2 + K_1}{\lambda_{\min}\left(\overline{E}_1^T P_1 \overline{E}_1\right)}; \tag{48}$$

then

$$\dot{V}_2 \leq -K_1\left\|s_1\right\|. \tag{49}$$

(2) Consider a Lyapunov function candidate

$$V_3 = \frac{1}{2} s_2^T s_2. \tag{50}$$

The time derivative of V_3 along the trajectories of system (28) is given by

$$\begin{aligned}
\dot{V}_3 &= \left(F_2 C_{22} e_2\right)^T\left(F_2 C_{22}\dot{e}_2\right) = \left(F_2 C_{22} e_2\right)^T\\
&\quad \cdot F_2 C_{22}\left(\left(\overline{A}_{22} - L_2 C_{22}\right)e_2(t) + \overline{A}_{21} e_1(t)\right.\\
&\quad + \overline{\Phi}_2(z,u) - \overline{\Phi}_2(\hat{z},u) + \overline{E}_2 f_a(t) + D_2 d(t)\\
&\quad \left. - \overline{E}_2 r_2(t)\right) = s_2^T F_2 C_{22}\left(\left(\overline{A}_{22} - L_2 C_{22}\right)e_2(t)\right.\\
&\quad + \overline{A}_{21} e_1(t) + \overline{\Phi}_2(z,u) - \overline{\Phi}_2(\hat{z},u) + \overline{E}_2 f_a(t)\\
&\quad \left. + D_2 d(t)\right) - \rho_2 s_2^T \overline{E}_2^T P_2 \overline{E}_2 \frac{s_2}{\left\|s_2\right\|}.
\end{aligned} \tag{51}$$

By the same reasoning as in the proof of the previous steps, it follows that

$$\begin{aligned}
\dot{V}_3 &\leq s_2^T F_2 C_{22}\left(\left(\overline{A}_{22} - L_2 C_{22}\right)e_2(t) + \overline{A}_{21} e_1(t)\right.\\
&\quad \left. + \overline{\Phi}_2(z,u) - \overline{\Phi}_2(\hat{z},u) + \overline{E}_2 f_a(t) + D_2 d(t)\right)\\
&\quad - \rho_2 \lambda_{\min}\left(\overline{E}_2^T P_2 \overline{E}_2\right)\left\|s_2\right\| \leq \left\|s_2\right\|\left(\left\|F_2 C_{22}\right\|\right.\\
&\quad \cdot \left(\left(\left\|\overline{A}_{22} - L_2 C_{22}\right\| + \gamma_4\right)\delta_2 + \left(\left\|\overline{A}_{21}\right\| + \gamma_4\right)\delta_1\right.\\
&\quad \left.\left. + \left\|\overline{E}_2\right\|\gamma_2 + \left\|D_2\right\|\gamma_1\right) - \rho_2 \lambda_{\min}\left(\overline{E}_2^T P_2 \overline{E}_2\right)\right).
\end{aligned} \tag{52}$$

We design ρ_2 which satisfies

$$\rho_2 \geq \frac{\|F_2C_{22}\|\left(\left(\left\|\overline{A}_{22}-L_2C_{22}\right\|+\gamma_4\right)\delta_2+\left(\left\|\overline{A}_{21}\right\|+\gamma_4\right)\delta_1+\left\|\overline{E}_2\right\|\gamma_2+\|D_2\|\gamma_1\right)+K_2}{\lambda_{\min}\left(\overline{E}_2^T P_2 \overline{E}_2\right)}; \tag{53}$$

then

$$\dot{V}_3 \leq -K_2\|s_2\|. \tag{54}$$

This completes the proof. □

This result shows that the sliding mode reachability condition is satisfied. As a consequence, based on the sliding mode equivalent principle, an ideal sliding motion will take place on the surface $s_i = 0$ $(i = 1, 2)$ and after some finite time.

5. Fault Reconstruction and Disturbance Estimation

When the system reaches the sliding mode surface, $s_i = \dot{s}_i = 0$ $(i = 1, 2)$ according to the sliding mode equivalent principle [20]. The following equations are obtained:

$$F_1e_{v1} = F_1\dot{e}_{v1} = F_1C_{11}\dot{e}_1 = 0, \tag{55}$$

$$F_2e_{v2} = F_2\dot{e}_{v2} = F_2C_{22}\dot{e}_2 = 0. \tag{56}$$

5.1. The Reconstruction of Actuator Fault $f_a(t)$. From (55), we obtain the following:

$$F_1C_{11}\left(\left(\overline{A}_{11}-L_1C_{11}\right)e_1(t)+\overline{A}_{12}e_2(t)+\overline{\Phi}_1(z,u) - \overline{\Phi}_1(\hat{z},u)+\overline{E}_1f_a(t)-\overline{E}_1r_1(t)\right)=0. \tag{57}$$

From (42), it follows that

$$f_a(t) \approx r_1(t) = \rho_1 \operatorname{sgn}\left(F_1e_{v1}\right). \tag{58}$$

The sigmoid function was used to replace sgn(s) to weaken the chattering problem of the sliding mode, where the reconstructed value of the actuator fault is

$$\hat{f}_a = \rho_1\left(\frac{2}{1+e^{-a_1F_1e_{v1}}}-1\right), \tag{59}$$

where a_1 is a positive constant to be designed.

5.2. The Estimation of the Unknown Input Disturbance $d(t)$. From (56), we obtain the following:

$$F_2C_{22}\left(\left(\overline{A}_{22}-L_2\overline{C}_{22}\right)e_2(t)+\overline{A}_{21}e_1(t)+\overline{\Phi}_2(x,u) - \overline{\Phi}_2(\hat{x},u)+\overline{E}_2f_a(t)-\overline{E}_2r_2(t)+\overline{D}_2d(t)\right)=0. \tag{60}$$

From (42), it follows that

$$d(t) \approx \overline{D}_2^{-1}\left(\overline{E}_2r_2(t)-\overline{E}_2f_a(t)\right) = \overline{D}_2^{-1}\overline{E}_2\left(\rho_2\operatorname{sgn}\left(F_2e_{v2}\right)-\rho_1\operatorname{sgn}\left(F_1e_{v1}\right)\right). \tag{61}$$

The sigmoid function was used to replace sgn(s) to weaken the chattering problem of the sliding mode, where the reconstruction value of the disturbance is

$$\hat{d}(t) = \overline{D}_2^{-1}\overline{E}_2\left(\rho_2\left(\frac{2}{1+e^{-a_2F_2e_{v2}}}-1\right) - \rho_1\left(\frac{2}{1+e^{-a_1F_1e_{v1}}}-1\right)\right), \tag{62}$$

where a_2 is a positive constant to be designed.

6. Adjustment of the Fault-Tolerant Control Law

In view of the control law u designed for the faulted system in (1), the observed value $\hat{x}$ of the system state, the reconstructed values $\hat{d}$ from (62), and $\hat{f}_a$ from (59) of the disturbance and fault, respectively, were substituted into (6). The adjusted control law becomes

$$u = (GB)^*\left(G\dot{r}-GA\hat{x}-G\Phi(\hat{x},u)-GE\hat{f}_a(t) - GD\hat{d}(t)+ks_0+\varepsilon\operatorname{sgn}(s_0)\right). \tag{63}$$

At this point, the control law in (7) adjusted to a nominal system is

$$u_n(t) = (GB)^*\left(G\dot{r}-GA\hat{x}-G\Phi(\hat{x},u)+ks_0 + \varepsilon\operatorname{sgn}(s_0)-GD\hat{d}(t)\right). \tag{64}$$

The additional control law in (8) is adjusted to

$$u_f(t) = (GB)\left(-GE\hat{f}_a(t)\right). \tag{65}$$

Stability analysis is as follows.

Consider the following Lyapunov function:

$$V_4 = \frac{1}{2}{s_0}^2. \tag{66}$$

The time derivative of V_4 along with (5) and (63) is

$$\begin{aligned}\dot{V}_4 &= s_0\dot{s}_0 = s_0\left(G\dot{r}-GAx-GDd-GEf_a\right.\\ &\quad -G\Phi(x,u)-GB(GB)^*\left(G\dot{r}-GA\hat{x}-G\Phi(\hat{x},u)\right.\\ &\quad \left.\left.-GE\hat{f}_a(t)-GD\hat{d}(t)+ks_0+\varepsilon\operatorname{sgn}(s_0)\right)\right)\\ &= s_0\left[-ks_0-\varepsilon\operatorname{sgn}(s_0)-GA(x-\hat{x})-G(\Phi(x,u)\right.\\ &\quad \left.-\Phi(\hat{x},u))-GE\left(f_a-\hat{f}_a\right)-GD\left(d-\hat{d}\right)\right]\end{aligned}$$

$$= s_0 \left[-ks_0 - \varepsilon \operatorname{sgn}(s_0) - GAT^{-1}e - G(\Phi(x,u) - \Phi(\hat{x},u)) - GE\left(f_a - \hat{f}_a\right) - GD\left(d - \hat{d}\right)\right]. \tag{67}$$

It follows from (43) that

$$\dot{V}_4 \le -k{s_0}^2 - \varepsilon|s_0| + |s_0|\left(\left\|GAT^{-1}\delta\right\| + \left\|GD\overline{D}_2^{-1}\overline{E}_2(\rho_2+\rho_1)\right\| + \|GE\rho_1\| + \|G\gamma\delta\|\right). \tag{68}$$

Finally, if the positive constant ε is chosen such that

$$\varepsilon \ge \delta\left\|GAT^{-1} + (\rho_2+\rho_1)\right\|\left\|GD\overline{D}_2^{-1}\overline{E}_2\right\| + \rho_1\|GE\| + \gamma\delta\|G\| \tag{69}$$

we obtain

$$\dot{V}_4 \le -k{s_0}^2. \tag{70}$$

It can be seen from the above analysis that the system can satisfy the asymptotic stability.

7. Simulation Example

Consider a robot arm whose motion equation is as follows [21]:

$$\begin{aligned} &M(q)\ddot{q} + B(q,\dot{q})\dot{q} + G(q) = u,\\ &M(q) = [J_1 0 0 J_m],\\ &B(q,\dot{q}) = [F_n 0 0 F_m],\\ &G(q) = [mgl\sin(q_1) + a(q_1 - q_2) - a(q_1 - q_2)], \end{aligned} \tag{71}$$

where q_1 and q_2 are the link displacement and the rotor displacement, respectively. The link inertia J_1, motor rotor inertia J_m, elastic constant a, link mass m, gravity g, connecting rod length l, and viscous friction coefficients F_n and F_m are all positive constant parameters. The control u is the torque delivered by the motor. The robot parameters are F_n = 0.5 Nm/(rad/s), J_m = 1 Nm^2, k = 2 Nm/rad, F_m = 1, J_1 = 2 Nm^2, m = 0.15 kg, g = 9.8 m/s^2, and l = 0.3 m.

From (1), the parameter matrixes are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k}{J_1} & \frac{-F_1}{J_1} & \frac{k}{J_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_m} & 0 & \frac{-k}{J_m} & \frac{-F_m}{J_m} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & -0.25 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & -2 & -1 \end{bmatrix},$$

$$f(x,u,t) = \begin{bmatrix} 0 \\ -0.2205\sin x_1 \\ 0 \\ 0 \end{bmatrix},$$

$$E = [00240021],$$

$$D = [10001002],$$

$$B = [00100001],$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{72}$$

The transformational matrix T is

$$T = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{73}$$

The transformational matrix S is

$$S = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{74}$$

Set the matrices as

$$\begin{aligned} &P_1 = [2001],\\ &P_2 = [1001],\\ &F_1 = [0204],\\ &F_2 = [0201],\\ &L_1 = [3 - 123.25],\\ &L_2 = [4 - 12 - 1],\\ &G = \begin{bmatrix} 5 & 1 & 0 & 0 \\ 0 & 0 & 5 & 1 \end{bmatrix}. \end{aligned} \tag{75}$$

In the control law of (63), the constants are taken to be k = 5 and ε = 2. The robotic arm system (71) is a dual input system, where the input signal 1 is $r_1 = 2\cos(\pi t)$ and the input

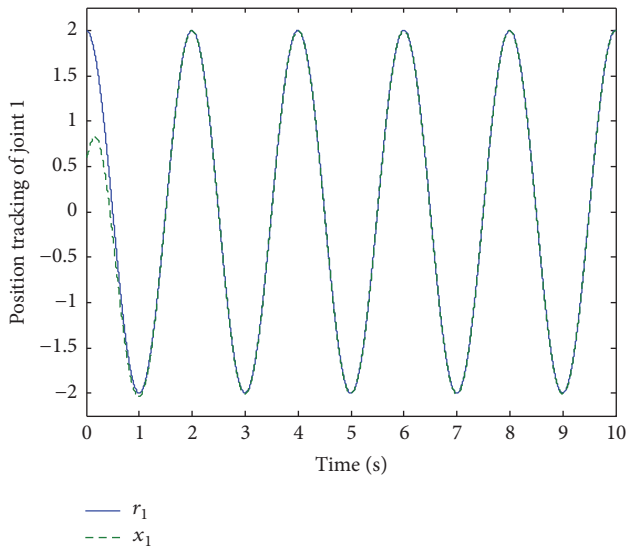


Figure 1: Actual value x_1 and tracking state of input signal 1.

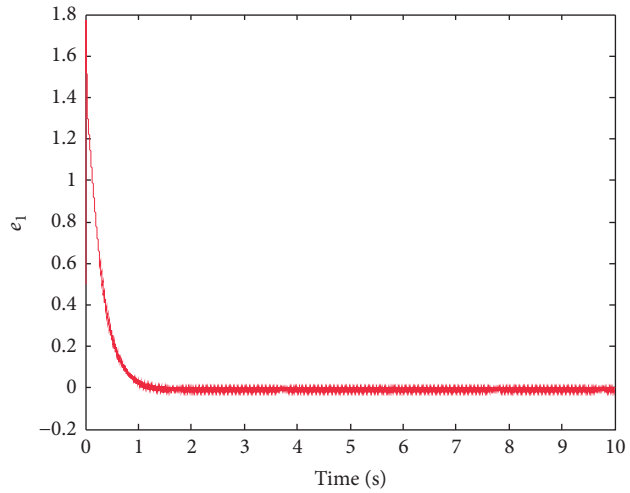


Figure 2: Tracking error e_1 of input signal 1.

signal 2 is $r_2 = 5\sin(\pi t)$. Two square signals were chosen as the "unknown" input disturbance, where one had a period of 1 s and an amplitude of 1 and the other had a period of 2 s with an amplitude of 2. The initial values of the system state variable $x(t)$ were taken as 0.6, 0.3, −1.5, and 0.5, respectively. For faultless systems, the simulated experimental results are as follows.

Figures 1 and 3 show two states of x_1 and x_3 used to separately track the two input signals in the case of different initial values; the given input values were reached quickly, and they remained stable. The errors, e_1 and e_2, shown in Figures 2 and 4, prove the point that the tracking effect is good. Figures 5 and 6 are two output signals of the control law, respectively.

When a system contains an actuator fault, a sinusoidal signal and a cosine signal were chosen to, respectively, simulate the fault: for example, $f_{a1}(t) = \sin(5t)$ and $f_{a2}(t) = \cos(5t)$.

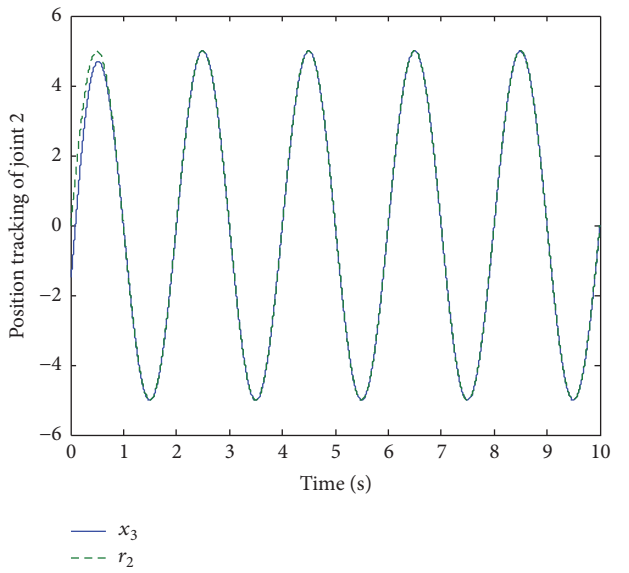


Figure 3: Actual value x_3 and tracking state of input signal 2.

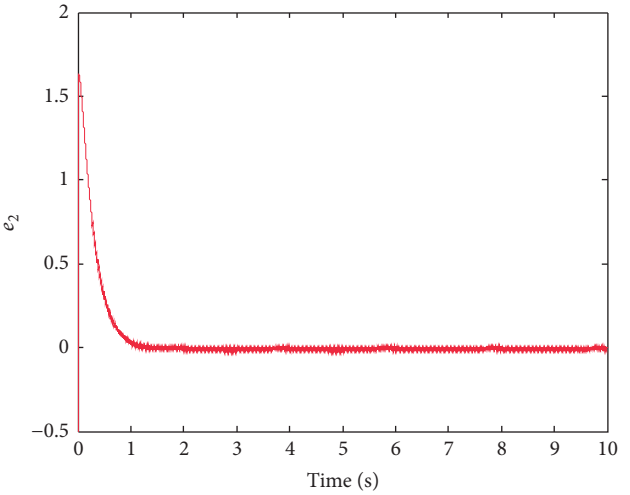


Figure 4: Tracking error e_2 of input signal 2.

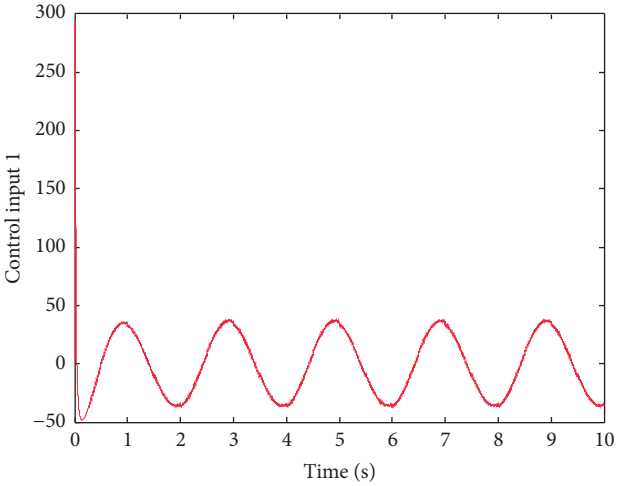


Figure 5: Control input signal u_1.

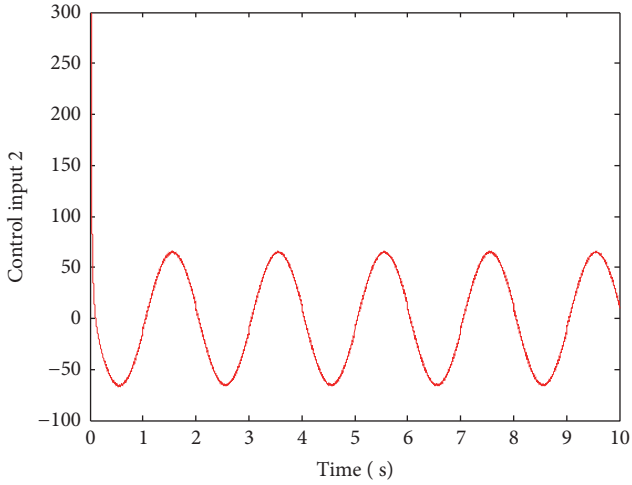


Figure 6: Control input signal u_2.

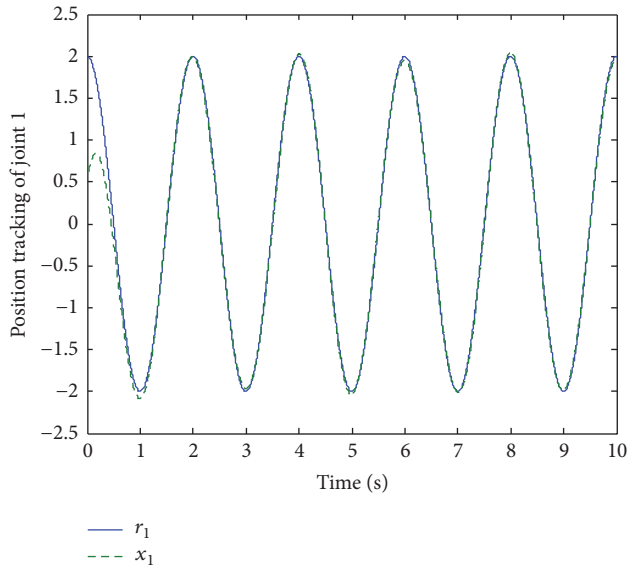


Figure 7: Actual value x_1 and tracking state of input signal 1.

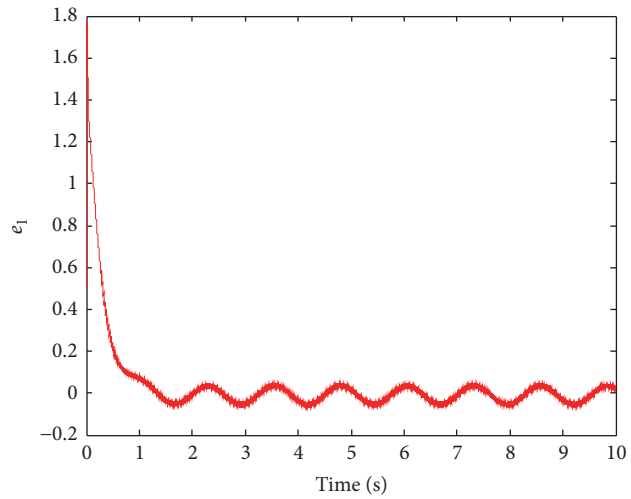


Figure 8: Tracking error e_1 of input signal 1.

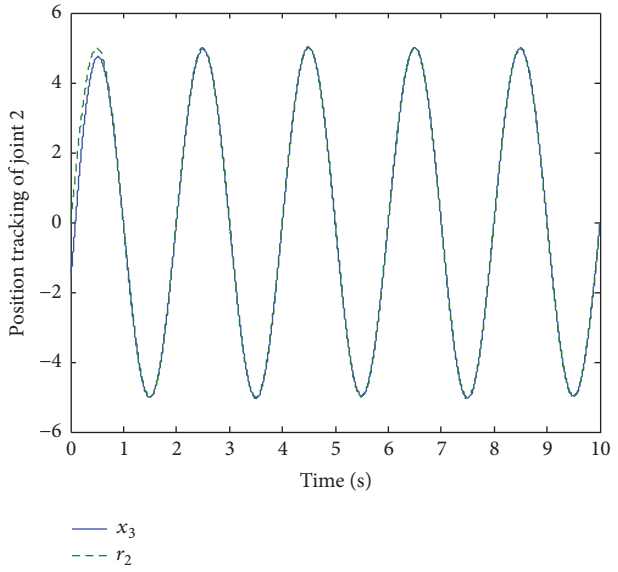


Figure 9: Actual value x_3 and tracking state of input signal 2.

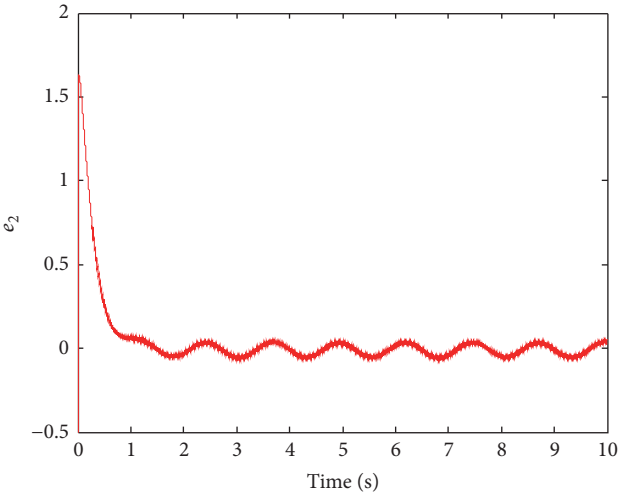


Figure 10: Tracking error e_2 of input signal 2.

Thus the fault-tolerant control was not carried out; namely, the additional control law u_f was not added at this point. Figures 7 and 9 show the result of two input signals tracking, and Figures 8 and 10 show the result of two tracking errors.

It can be seen from Figures 7, 8, 9, and 10 that the input signals cannot be accurately tracked, and the tracking errors converge to 0. When the additional control law u_f is added for fault-tolerant control, the input signal tracking results are shown in Figures 15 and 17, and the results of two tracking errors are shown in Figures 16 and 18. The reconstruction results are shown in Figures 11, 12, 13, and 14.

Figures 11 and 12 display the unknown disturbances d_1 and d_2 and their estimated values, respectively. Meanwhile, Figures 13 and 14 show the actuator faults f_{a1} and f_{a2} and their reconstruction values separately. The simulation

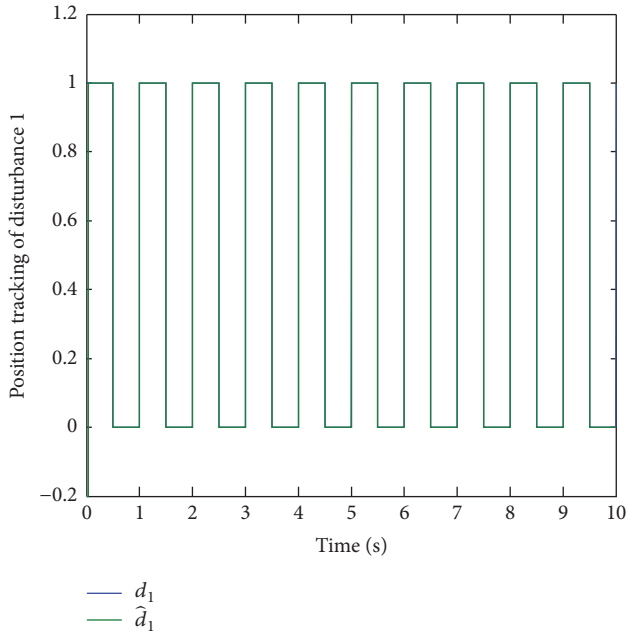


Figure 11: Unknown input disturbance d_1 and its estimation value.

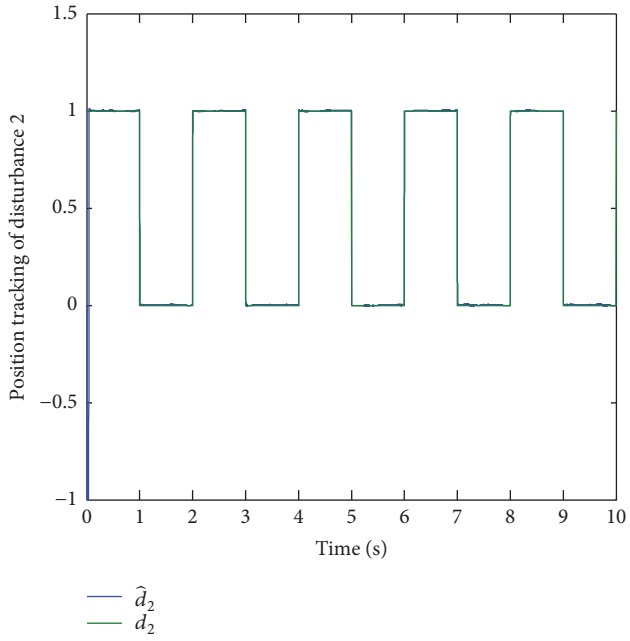


Figure 12: Unknown input disturbance d_2 and its estimation value.

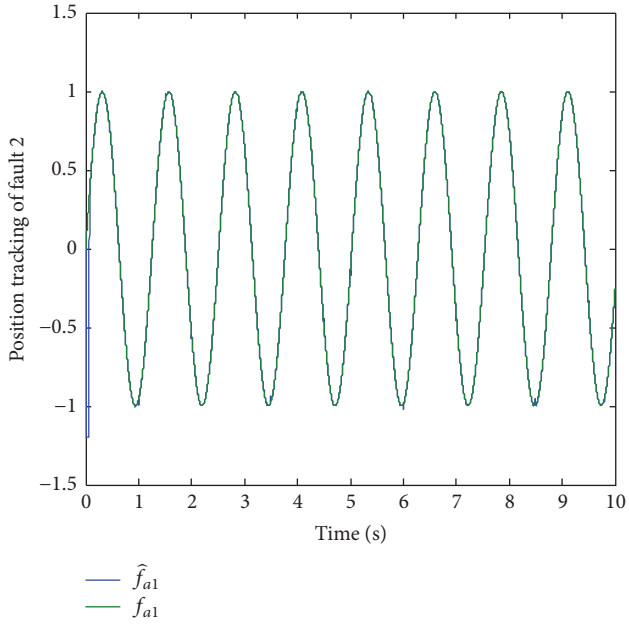


Figure 13: Actuator fault f_{a1} and its reconstruction value.

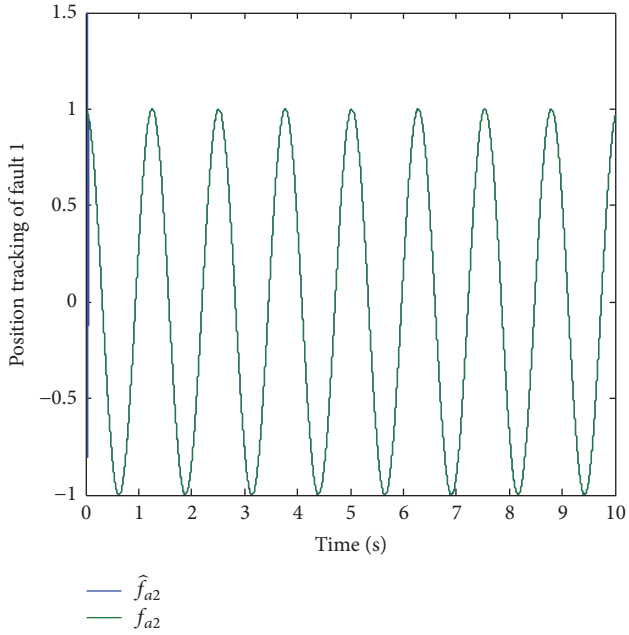


Figure 14: Actuator fault f_{a2} and its reconstruction value.

results indicate that the method can estimate the unknown disturbances and reconstruct the actuator faults accurately. And the tracking effect is good.

Compared with Figures 7, 8, 9, and 10, it can be seen from Figures 15, 16, 17, and 18 that the input signals can be accurately tracked after the additional control law u_f is added, and the tracking errors can converge to 0. Figures 19 and 20 are two output signals of the control law, respectively.

8. Conclusions

In this paper, we proposed a fault-tolerant control scheme based on fault reconstruction for a class of nonlinear systems with actuator faults and unknown input disturbances. It can obtain information from the unknown system state and can restrain the influence of the fault and disturbance. Therefore, this method has very strong engineering practicability. From the simulated experimental results applied on a mechanical

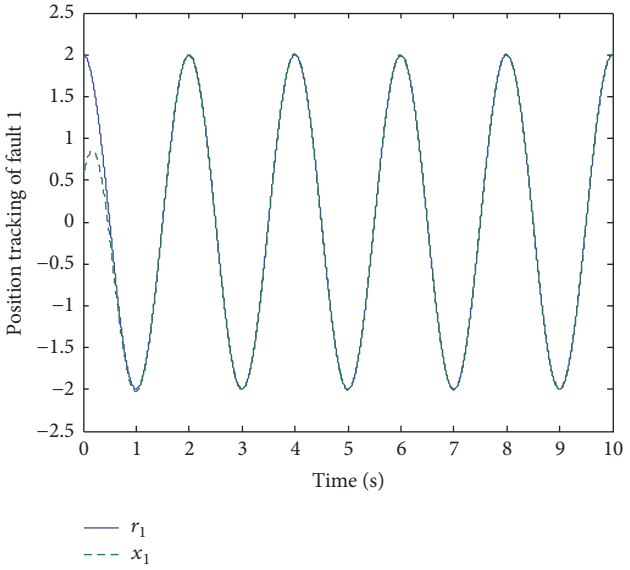


Figure 15: Actual value x_1 and tracking state of input signal 1.

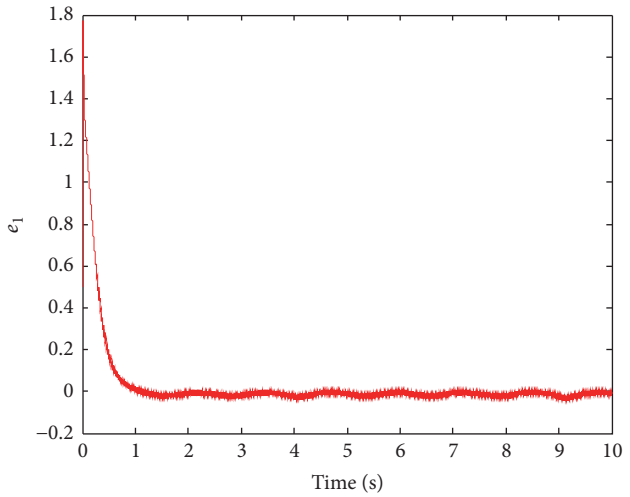


Figure 16: Tracking error e_1 of input signal 1.

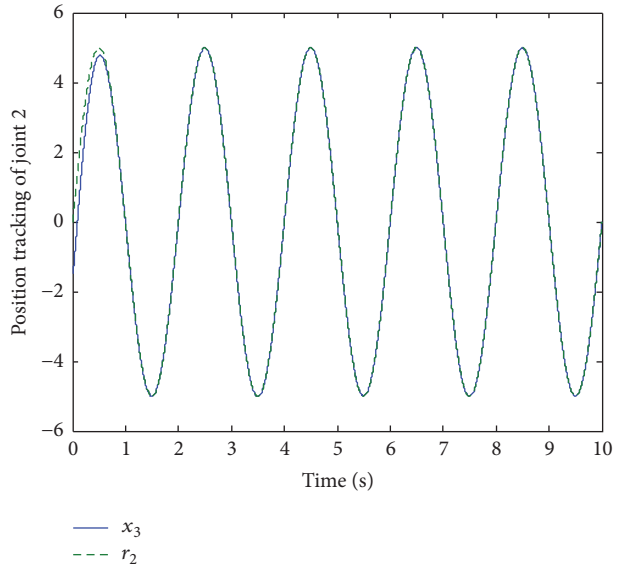


Figure 17: Actual value x_3 and tracking state of input signal 2.

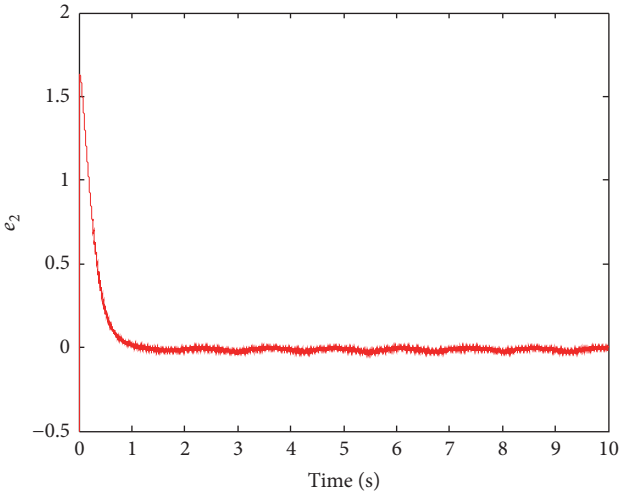


Figure 18: Tracking error e_2 of input signal 2.

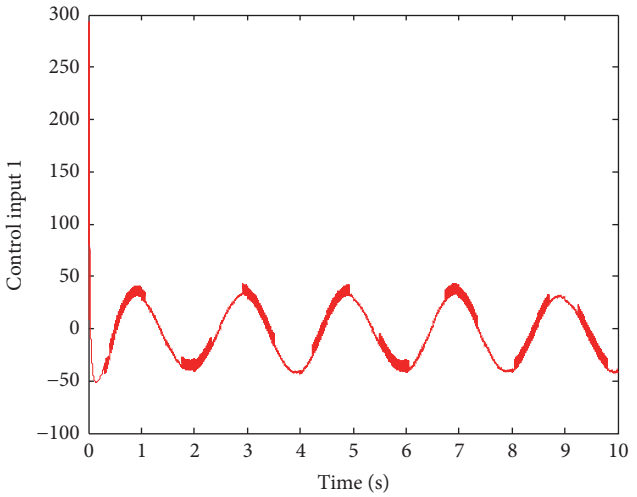


Figure 19: Control input signal u_1.

are, our method can maintain the system asymptotic stability and can achieve a precise control purpose when it contains an actuator fault and an unknown input disturbance, which demonstrates that our proposed method provides a new way to control nonlinear systems with actuator faults and unknown disturbances in practical engineering applications.

Competing Interests

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

Acknowledgments

This work was supported by the Natural Science Foundation of China (nos. 61473117 and 61273157) and Hunan Provincial

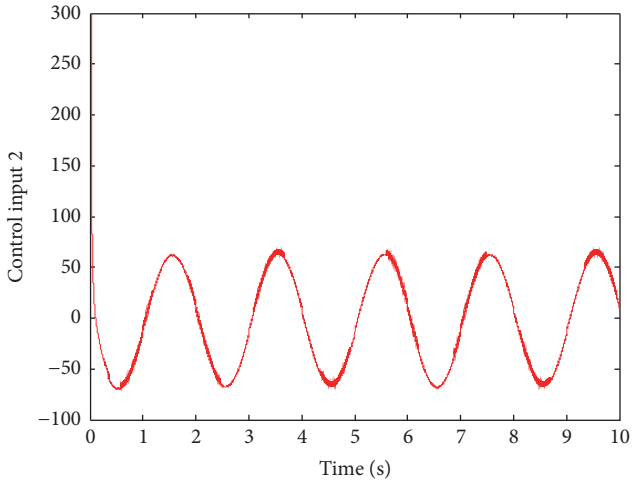


FIGURE 20: Control input signal u_2.

Natural Science Foundation of China (nos. 2016JJ5007 and 2015JJ5011).

References

[1] B. Mogens, K. Michel, L. Jan, and S. Marcel, *Diagnosis and Fault-Tolerant Control*, Springer, Berlin, Germany, 2nd edition, 2006.

[2] Z. Zhou, C. Hu, H. Fan, and J. Li, "Fault prediction of the nonlinear systems with uncertainty," *Simulation Modelling Practice and Theory*, vol. 16, no. 6, pp. 690–703, 2008.

[3] X. He, Z. Wang, and D. H. Zhou, "Network-based robust fault detection with incomplete measurements," *International Journal of Adaptive Control and Signal Processing*, vol. 23, no. 8, pp. 737–756, 2009.

[4] M. A. Djeziri, R. Merzouki, B. O. Bouamama, and M. Ouladsine, "Fault diagnosis and fault-tolerant control of an electric vehicle overactuated," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 986–994, 2013.

[5] R. Loureiro, S. Benmoussa, Y. Touati, R. Merzouki, and B. Ould Bouamama, "Integration of fault diagnosis and fault-tolerant control for health monitoring of a class of MIMO intelligent autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 30–39, 2014.

[6] D. Zhang, W.-A. Zhang, L. Yu, and Q.-G. Wang, "Distributed fault detection for a class of large-scale systems with multiple incomplete measurements," *Journal of the Franklin Institute*, vol. 352, no. 9, pp. 3730–3749, 2015.

[7] D. Zhang, P. Shi, W. Zhang, and L. Yu, "Energy-efficient distributed filtering in sensor networks: a unified switched system approach," *IEEE Transactions on Cybernetics*, pp. 1–12, 2016.

[8] X. He, Z. D. Wang, L. G. Qin, and D. H. Zhou, "Active fault-tolerant control for an internet-based networked three-tank system," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 2150–2157, 2016.

[9] T. Tao and H. Xu, "Fault-tolerant control of high-speed trains with uncertain parameters and actuator failures using control reallocation and adaptive control," *Journal of Information and Computational Science*, vol. 10, no. 7, pp. 1959–1977, 2013.

[10] T. Tao and H. Z. Xu, "Fault-tolerant control of high-speed trains with parameter uncertainty," *Applied Mechanics and Materials*, vol. 325-326, pp. 1099–1105, 2013.

[11] X. Yu and J. Jiang, "Hybrid fault-tolerant flight control system design against partial actuator failures," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 871–886, 2012.

[12] Y. Shen, L. Liu, and E. H. Dowell, "Adaptive fault-tolerant robust control for a linear system with adaptive fault identification," *IET Control Theory & Applications*, vol. 7, no. 2, pp. 246–252, 2013.

[13] J. Zhao, B. Jiang, P. Shi, and Z. He, "Fault tolerant control for damaged aircraft based on sliding mode control scheme," *International Journal of Innovative Computing, Information and Control*, vol. 10, no. 1, pp. 293–302, 2014.

[14] H. Fan, B. Liu, Y. Shen, and W. Wang, "Adaptive failure compensation control for uncertain systems with stochastic actuator failures," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 808–814, 2014.

[15] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, 2014.

[16] X. Y. Zhang, B. Jiang, and K. Zhang, "A reliable tracking control for the 3-DOF hovering system of quadrotor with multi-actuator faults," *Chinese Conference of Decision and Control*, pp. 4961–4966, 2013.

[17] Q. Hu, X. Huo, B. Xiao, and Z. Zhang, "Robust finite-time control for spacecraft attitude stabilization under actuator fault," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 226, no. 3, pp. 416–428, 2012.

[18] W. Chen and F. N. Chowdhury, "Design of sliding mode observers with sensitivity to incipient faults," in *Proceedings of the 16th IEEE International Conference on Control Applications (CCA '07)*, pp. 795–800, Singapore, October 2007.

[19] M. Corless and J. A. Y. Tu, "State and input estimation for a class of uncertain systems," *Automatica*, vol. 34, no. 6, pp. 757–764, 1998.

[20] M. R. Soltanpour, B. Zolfaghari, M. Soltani, and M. H. Khooban, "Fuzzy sliding mode control design for a class of nonlinear systems with structured and unstructured uncertainties," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 7, pp. 2713–2726, 2013.

[21] X. Zhang, T. Parisini, and M. Polycarpou, "Sensor bias fault isolation in a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 370–376, 2005.

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 2697297, 8 pages
https://doi.org/10.1155/2017/2697297

Research Article

Fault Diagnosis Method Based on Information Entropy and Relative Principal Component Analysis

Xiaoming Xu and Chenglin Wen

School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Chenglin Wen; wencl@hdu.edu.cn

Received 19 December 2016; Accepted 22 January 2017; Published 20 February 2017

Academic Editor: Xiao He

In traditional principle component analysis (PCA), because of the neglect of the dimensions influence between different variables in the system, the selected principal components (PCs) often fail to be representative. While the relative transformation PCA is able to solve the above problem, it is not easy to calculate the weight for each characteristic variable. In order to solve it, this paper proposes a kind of fault diagnosis method based on information entropy and Relative Principle Component Analysis. Firstly, the algorithm calculates the information entropy for each characteristic variable in the original dataset based on the information gain algorithm. Secondly, it standardizes every variable's dimension in the dataset. And, then, according to the information entropy, it allocates the weight for each standardized characteristic variable. Finally, it utilizes the relative-principal-components model established for fault diagnosis. Furthermore, the simulation experiments based on Tennessee Eastman process and Wine datasets demonstrate the feasibility and effectiveness of the new method.

1. Introduction

In the process of the industry manufacturing, there is a large amount of variables that is highly correlative; these variables contain the essential information that would be helpful to judge the status of the system. As a result, it is an important problem to find and predict the fault through this information to ensure that the equipment always works in a safe and reliable way [1, 2].

However, during the industry manufacturing, the collected characteristic variables have different units; this raises a problem that we may come out with the different result only due to the unit difference; hence, we have to standardize the unit. What is more, after the standardization, it is inevitable to lose the diversity among different variables and present the property of distribution uniformity in the perspective of geometry which makes it hard to extract the principle component for compression and diagnosis. As to overcome these problem, some methods have been proposed recently [3–8]. Shi et al. use the Mahalanobis distance for relative transformation to reduce the effect of the dimension standardization [4]. Tang et al. propose a relative transformation principal component analysis to reduce the data noise for the transformation oil breakdown voltage prediction [5]. Yi et al. introduce a relative transformation operator to change the original variables in the spatial distribution and eigenvalues of the covariance matrix in the feature space [6]. Wen et al. propose a method called Relative Principle Component Analysis (RPCA); it introduces weighting for each variable based on the prior information of the system to eliminate the false information due to standardizing the variable units [7, 8], but the shortage of this method is that it needs a large amount of prior information from the system which is hard to gain in real engineering application.

In order to solve the problem, this paper introduces the concept of information entropy and proposes a new fault diagnosis method that combines the information entropy and relative transformation PCA which is called information entropy relative transformation PCA (InEnRPCA). The information entropy is put forward by Shannon in 1948 [9]; it indicates that the redundancy exists among any information and can be measured based on the symbol in the information such

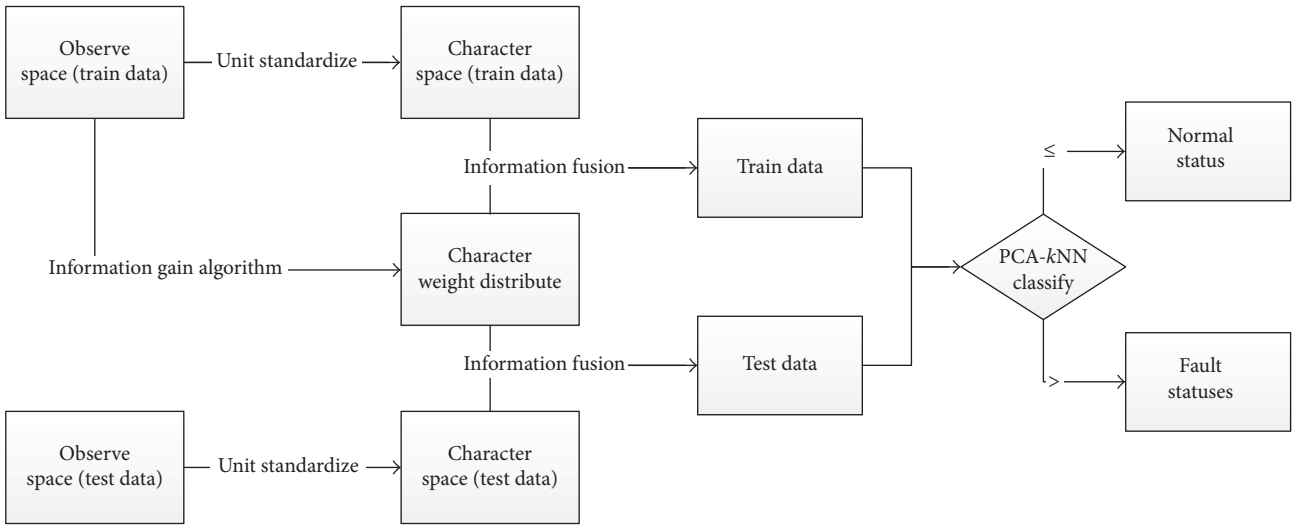


Figure 1: InEnRPCA algorithm process.

as the number, alphabet, and words. With the development of the information theory, the information entropy has become an effective method to measure the degree of importance for each feature in the sample and achieved a wide application in many areas. For instance, while using the decision-making tree for classification, Hu et al. use the information entropy to calculate the significance for each feature and then prune the decision-making tree and reduce the false alarm [10]. Y. Y. Chen and Y. M. Chen use the information entropy to measure the uncertainty on the data in the attribute reduction algorithm [11]. Wang et al. use the information entropy to balance the weight of each sememe in the area of natural language processing [12].

As to the problem we are facing, we start in the perspective of information theory and use the information gain algorithm to extract information entropy as the heuristic knowledge from the original dataset, then use it to calculate the relative transformation factor and allocate the weight for each standardized characteristic variable, and, finally, use the corresponding RPCA method for fault diagnosis.

The rest of this paper is organized as follows: in Section 2, we review the information entropy and information gain definition and algorithm. The original relative transformation PCA method is given in Section 3. Our simulation experiments on Tennessee Eastman process and the Wine dataset from UCI are stated in Section 4, where we make a comparisons between PCA, USPCA, and improved InEnRPCA with thirteen datasets to demonstrate the effectiveness of the new method. Finally, Section 5 gives conclusions and some discussions.

2. Overview of Our Approach

A brief overview of our fault diagnosis approach is given in this section. As the framework in Figure 1 shows, the proposed approach consists of two parts: calculating the relative transformation operators based on information entropy and fault diagnosis based on RPCA-kNN.

In the part for calculating the relative transformation operators, information entropy and information gain algorithm are applied on train data to get the relative transformation operators. Then, the operators are combined with the original data to get the relative transformation matrix.

In fault diagnosis part, the RPCA are used to deal with processed data for dimension reduction, and kNN are used for classification training. After the above process, a model is built for further fault diagnosis.

3. Information Entropies

In the subject of the information theory, probability, and statistics, entropy is used to describe the uncertainty of random variable and can be used to show the reduced degree of the information uncertainty for set Y after getting the character X.

3.1. The Definition of the Information Entropy and Information Gain

Theorem 1 (information entropy). *Assume M is Lebesgue measure set generated by measurable set S with algebra σ and measure μ where $\mu(M) = 1$, also M can be shown in the form of the incompatible sets with $A = A_i$, that is, $M = U_{i=1}^{n} A_i$ and $A_i \cup A_j = \Phi$, $\forall i \neq j$ [13]. Then, we can come to a conclusion:*

$$H(A) = -\sum_{i=0}^{n} \mu(A_i) \log \mu(A_i), \tag{1}$$

where $\mu(A_i)$ is the measure for A_i.

Theorem 2 (information gain). *Given the train set D and the corresponding feature A, the empirical entropy $H(D)$ means the*

uncertainty level to classify set D, and the conditional empirical entropy $H(D \mid A)$ means the uncertainty level to classify feature A in the condition of set D. Then, the difference between them is the information gain which stands for the reduced level of the uncertainty to classify the set D by given feature A.

Obviously, as to the dataset D, the information gain depends on the features and different features have different information gain, the bigger the information gain, the stronger the ability it has for classifying.

3.2. The Information Gain Algorithm. Assume the train dataset is D; $|D|$ denotes the sample capacity which is equal to the number of the samples. There are K classes in set D with each name being C_k $(k = 1, 2, \ldots, K)$ and $|C_k|$ is equal to the number of the samples which belongs to C_k; that is, $\sum_{k=1}^{K} |C_k| = |D|$. Also, assume there are n values in feature A called $\{a_1, a_2, \ldots, a_n\}$ and then divide set D into n parts with each name being D_i $(i = 1, 2, \ldots, n)$ and $|D_i|$ is equal to the number of the samples that belongs to D_i; that is, $\sum_{i=1}^{n} |D_i| = |D|$. Based on the above, define the set D_{ik} as the intersection of class C_k and subset D_i, that is, $D_{ik} = D_i\ \forall D_k$, and $|D_{ik}|$ is the number of D_{ik} [14]. Then, the information gain algorithm can be shown as follows: input: train dataset D and feature A; output: the information gain of set D by feature A defined as $g(D, A)$.

Step 1. Calculate the empirical entropy of dataset $H(D)$:

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}. \tag{2}$$

Step 2. Calculate the empirical entropy of the feature A in the condition of set D:

$$\begin{aligned} H(D \mid A) &= \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) \\ &= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}. \end{aligned} \tag{3}$$

Step 3. Calculate the information gain which is also the relative transformation operator M_A:

$$\begin{aligned} g(D, A) &= H(D) - H(D \mid A), \\ M_A &= g(D, A). \end{aligned} \tag{4}$$

Step 4. Repeat the above processes from Steps 1 to 3 for each feature i in the sample and get the relative transformation operator with each feature:

$$M_i = g(D, i). \tag{5}$$

4. Relative Transformation Principle Component Analysis

After the standardization for the data dimensions by the traditional PCA method, it may bring in some fake information to the principle element due to the uniform distribution. Therefore, we use the relative transformation PCA method to solve the problem [15]. Below is the brief RPCA process.

Assume $X \in R^{a \times b}$, where a corresponds to the number of the samples and b is the number of features.

Step 1. Transfer the original data to the standardized data with mean zero and variance one.

The mean for each column is

$$m = \frac{1}{a} \left(x^0\right)^T l_a, \tag{6}$$

where $l_a = [1, 1, \ldots, 1]^T \in R^a$; then,

$$X = \left(X^0 - l_a m^T\right) \Sigma^{-1}, \tag{7}$$

where Σ_i $(i = 1, \ldots, b)$ is the standard deviation for each column.

We set the relative transformation:

$$\begin{aligned} X^R &= XM \\ &= \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(N) \\ x_2(1) & x_2(2) & \cdots & x_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ x_n(1) & x_n(2) & \cdots & x_n(N) \end{bmatrix} \\ &\quad \times \begin{bmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_n \end{bmatrix} \\ &= \begin{bmatrix} x_1^R(1) & x_1^R(2) & \cdots & x_1^R(N) \\ x_2^R(1) & x_2^R(2) & \cdots & x_2^R(N) \\ \vdots & \vdots & \ddots & \vdots \\ x_n^R(1) & x_n^R(2) & \cdots & x_n^R(N) \end{bmatrix}, \end{aligned} \tag{8}$$

where X^R is relative transformation matrix from X and M is a diagonal matrix in which each of the variables M_i is the relative transformation operator that can be achieved from Section 3.2.

Step 2. Compute the covariance matrix R_{X^R} from X_R:

$$R_{X^R} = \frac{X^T X}{a - 1}. \tag{9}$$

Step 3. Compute the eigenvalue λ and the corresponding eigenvector of the covariance matrix R_{X^R}. Assume all eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \cdots$, where λ_i stands for the i' eigenvalue and p_i is the corresponding eigenvector to λ_i.

$$\begin{aligned} &\left|\lambda I - R_{X^R}\right| = 0, \\ &\left|\lambda_i I - R_{X^R}\right| p_i = 0; \quad i = 1, 2, \ldots, b. \end{aligned} \tag{10}$$

Table 1: The datasets descriptions and parameter constitution.

Dataset	Numbers		Dimensions	Classes
	Train	Test		
TE (1–4)	800	400	52	4
TE (5–8)	800	400	52	4
TE (9–12)	800	400	52	4
TE (13–16)	800	400	52	4
TE (17–20)	800	400	52	4
TE (1–3)	600	300	52	3
TE (4–6)	600	300	52	3
TE (7–9)	600	300	52	3
TE (10–12)	600	300	52	3
TE (13–15)	600	300	52	3
TE (16–18)	600	300	52	3
TE (19–21)	600	300	52	3
Wine	148	30	14	3

Step 4. Select the number of the relative principle element based on the accumulative contribution rate:

$$\frac{\sum_1^i \lambda_i}{\sum_1^n \lambda_i} > 85\%. \tag{11}$$

Step 5. Use kNN for classifying.

5. Application Experiments

In order to verify the universal applicability and effectiveness of the new method, firstly, we test it by thirteen experiments based on Tennessee Eastman process dataset and the Wine dataset [16] which are different in numbers of samples N, dimensions D, and classes C; in addition, we divide all the classes in three or four classes in sequence to avoid the deliberate choosing; secondly, experiments are also performed on original PCA and traditional DPPCA, respectively. The datasets descriptions and experimental settings are shown in Table 1.

5.1. TE Process Example. We use the Tennessee Eastman process data as the testing samples which are obtained from the document. The dataset includes twenty-one classes and each class in the experiment has two hundred train samples and one hundred test samples; besides, each sample has 52 features. The different types in TE process are overlapped and difficult to classify in observed space.

After the data preprocessing, the number of the principle elements is determined by the sum of elements being more than 85%. For the twelve experiments based on Tennessee Eastman process, we choose one of the experiments based on TE classes 5, 6, 7, and 8 labeled by F1, F2, F3, and F4, respectively, for detail introduction and demonstration. Its relative transformation operator for each feature to the TE process can be seen in Table 2 and the experimental results with the three approaches are shown in Figures 2(a)–2(c).

As we can see from the figures, PCA and DSPCA have poor performance in pattern classification where four classes are overlapped. In Table 4, their classification accuracy is 31.5% and 75.3%, respectively. Compared with them, the information entropy RPCA can help identify each class higher and its classification accuracy is 82.3%. Clearly, the InEnRPCA is a suitable feature extraction and classification method for fault diagnosis.

5.2. Wine Dataset. We select the Wine data from the machine learning database UCI, which includes three classes with the number of the each class being 59, 61, and 58; besides, each of the samples has fourteen features. The types in Wine are overlapped and difficult to classify in observed space.

The relative transformation operator for each feature in the Wine dataset can be seen in Table 3 and the experimental results with the three approaches are shown in Figures 3(a)–3(c).

From Figure 2(b) and Table 2, we can see that PCA and DSPCA do not have a perfect performance in pattern classification with the accuracy rate is 66.7% and 96.7%, respectively. Compared with them, the InEnRPCA can identify each class accurately, and the testing data can also be classified obviously, and its classification accuracy rate is 100%.

5.3. Discussion. From the experiments above, it is obvious that the proposed methods can successfully distinguish different fault types and implement fault pattern recognition and diagnosis in most times, via utilizing information entropy to calculate relative transformation operator for each feature and combining with relative PCA approach.

Meanwhile, we also find that although the proposed method can improve the fault classification performance in most times (nearly 80%), the method is sometimes inferior to DSPCA; the reason may be due to the overfitting for the train set and the dissimilarity between the train set and test set; however, we can make sure that utilizing the information

Table 2: Weight for each feature in the TE dataset.

Feature	Weight
1	1.26
2	0.83
3	1.16
4	1.06
5	1.15
6	0.82
7	1.25
8	0.94
9	0.89
10	1.29
11	1.25
12	0.96
13	1.26
14	1.12
15	0.90
16	1.23
17	0.80
18	1.06
19	1.37
20	1.35
21	1.14
22	1.23
23	1.27
24	0.78
25	1.25
26	0.58
27	0.67
28	1.27
29	1.32
30	0.94
31	1.30
32	0.54
33	0.67
34	1.31
35	1.23
36	1.23
37	0.59
38	1.30
39	0.60
40	0.65
41	0.78
42	1.17
43	1.25
44	1.26
45	1.53
46	1.34
47	1.31
48	0.96
49	0.90
50	1.37
51	1.22
52	1.41

Table 3: Weight for each feature in the Wine dataset.

Feature	Weight
1	0.53
2	0.28
3	0.13
4	0.28
5	0.20
6	0.42
7	0.71
8	0.17
9	0.28
10	0.59
11	0.43
12	0.57
13	0.59
14	1.56

Table 4: The classifying accuracy rate for each method.

Dataset	PCA (%)	DSPCA (%)	InEnRPCA (%)
TE (1–4)	50.5	95.8	97.3
TE (5–8)	31.5	75.3	82.3
TE (9–12)	34.0	41.8	47.5
TE (13–16)	38.3	54.5	53.8
TE (17–20)	47.3	77.0	78.0
TE (1–3)	68.7	100.0	100.0
TE (4–6)	62.0	64.0	59.0
TE (7–9)	47.0	73.3	77.3
TE (10–12)	45.7	58.7	59.3
TE (13–15)	49.0	59.3	38.3
TE (16–18)	62.0	83.3	85.0
TE (19–21)	32.7	55.3	59.3
Wine	66.7	96.7	100

entropy to calculate the weight for each feature to avoid distribution equally between different features is reasonable; we will continue in-depth research for better fault diagnosis.

6. Conclusion

This paper has analyzed the PCA method in the perspective of information theory and proposed a kind of fault diagnosis research based on information entropy and Relative Principle

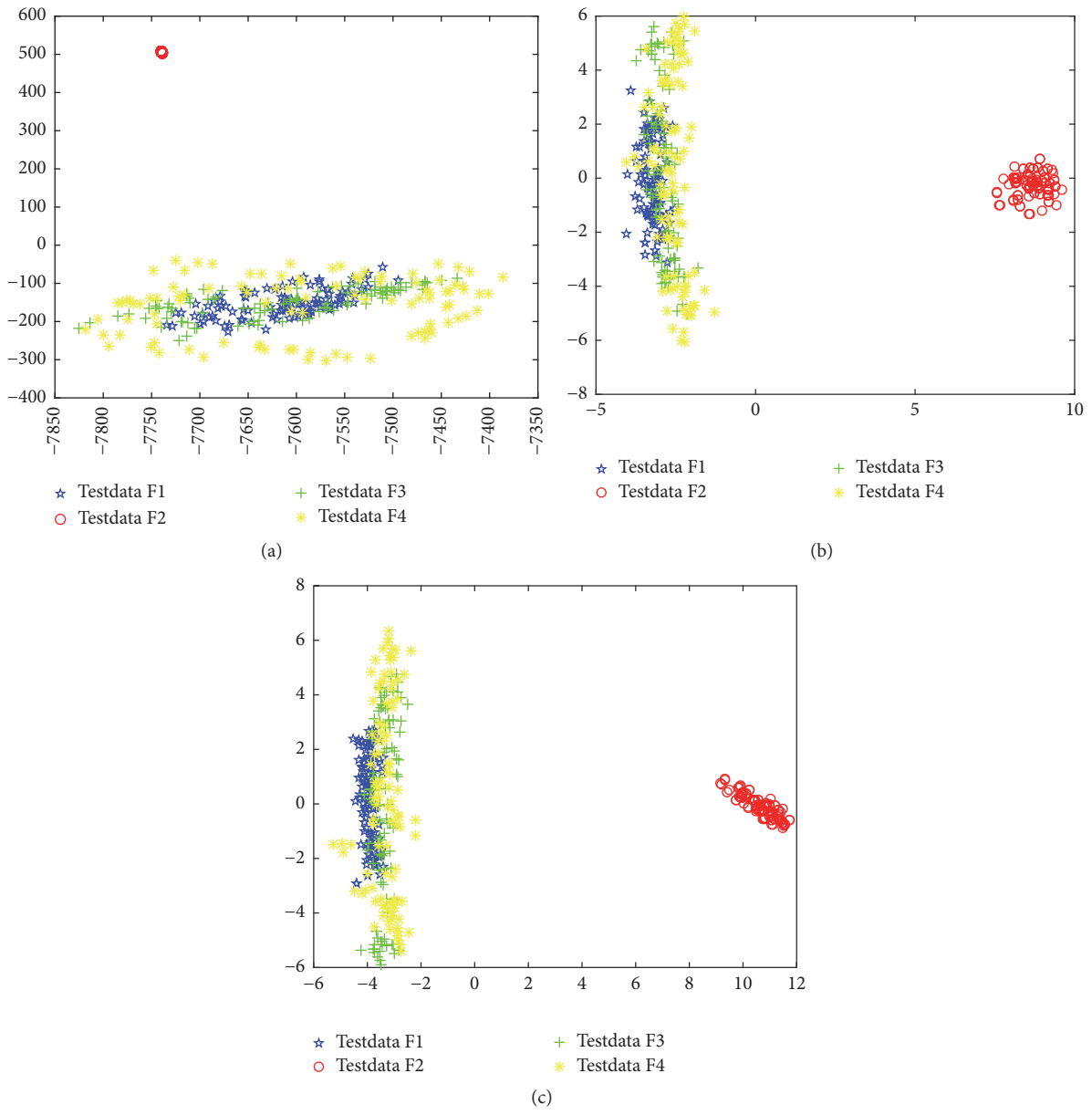


Figure 2: Feature extraction to TE process: (a) PCA. (b) DSPCA. (c) InEnRPCA.

Component Analysis. This proposed method can solve the following problems effectively: (1) the different result only due to calculating in different perspective of dimensions; (2) being uniformly distributed between the characteristic variables after dimension standardization; (3) how to calculate the weight for each variable. Both theoretical analysis and simulation experiments on Tennessee Eastman process and the Wine dataset from UCI demonstrate the feasibility and effectiveness of the new approach.

It is worth noting that the idea of using the information entropy to determine the certainty of the features is not new. However, the idea of combining information entropy with RPCA for faulty detection, to our knowledge, should have no previous publication. The proposed method should be seen as an alternative fault diagnosis method. It is not superior to the other methods in all cases.

There is some interesting future work:

(i) To exploit the way to make the energy conservation the same before and after the relative transformation: in the perspective of the energy conservation, the system energy before and after the relative transformation may usually not be the same. If we can consider the energy conservation during the relative transformation, the effect may be much better.

(ii) To apply the relative transformation by information entropy as data preprocessing in other methods: the proposed method is not limited to RPCA method; it

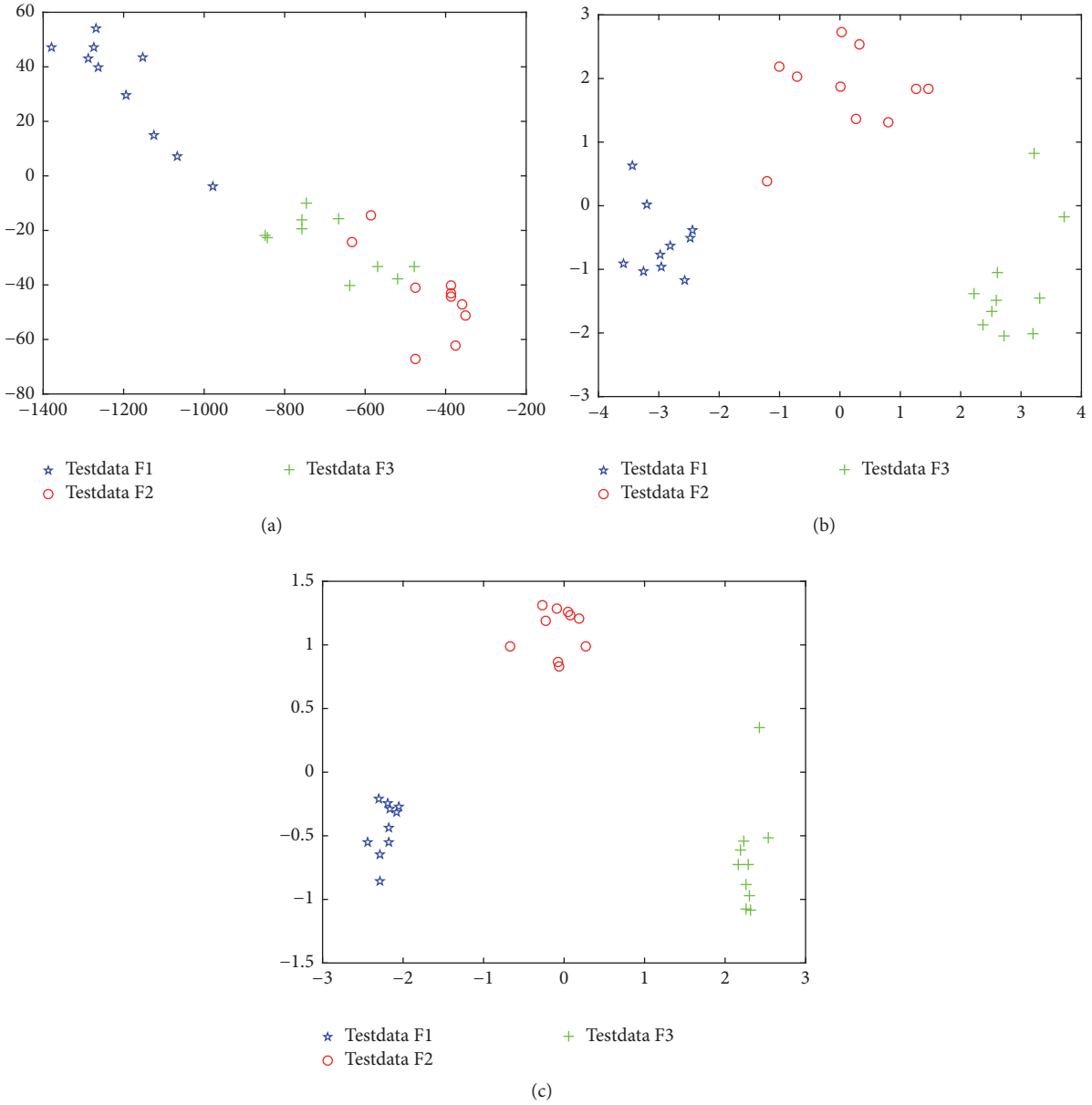


FIGURE 3: Feature extraction to Wine dataset: (a) PCA. (b) DSPCA. (c) InEnRPCA.

is an effective data preprocessing way that can also be applied to other fault diagnosis methods.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by National Natural Science Foundation (NNSF) of China under Grant nos. U1509203, 61333005, U1664264, and 61490701.

References

[1] N.-Y. Lu, F.-L. Wang, F.-R. Gao, and S. Wang, "Statistical modeling and online monitoring for batch processes," *Acta Automatica Sinica*, vol. 32, no. 3, pp. 400–410, 2006.

[2] T. Z. Wang, *Intelligent Integration of Data Mining Method and Its Application*, Shanghai Maritime University, 2006.

[3] G. H. Wen, "Relative transformation for machine learning," *Journal of Computer Research and Development*, vol. 45, no. 4, pp. 612–618, 2008.

[4] H. T. Shi, J. C. Liu, and Y. W. Zhang, "Principle analysis faulty diagnosis based on relative transformation," *Journal of Huazhong University of Science and Technology*, vol. 37, supplement 1, pp. 70–73, 2009.

[5] Y. Tang, T. Peng, Y. Xiong, and F. Jiang, "Breakdown voltage prediction method for transformer oil based on relative transformation principal component analysis," *Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument*, vol. 36, no. 7, pp. 1640–1645, 2015.

[6] J. Yi, D. Huang, S. Fu, H. He, and T. Li, "Optimized relative transformation matrix using bacterial foraging algorithm for process fault detection," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2595–2605, 2016.

[7] W. Cheng-Lin, H. Jing, W. Tian-Zhen, and C. Zhi-Guo, "Relative PCA with applications of data compression and fault diagnosis," *Acta Automatica Sinica*, vol. 34, no. 9, pp. 1128–1139, 2008.

[8] J. Hu, *Relative PCA and Its Application Research*, Henan University, 2008.

[9] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[10] Q. Hu, X. Che, L. Zhang, D. Zhang, M. Guo, and D. Yu, "Rank entropy-based decision trees for monotonic classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 2052–2064, 2012.

[11] Y. Y. Chen and Y. M. Chen, "Attribute reduction algorithm based on information entropy and ant colony optimization," *Journal of Chinese Computer System*, vol. 36, no. 3, pp. 586–590, 2015.

[12] X. L. Wang, L. Y. Lu, and W. P. Tai, "Research of a new algorithm of words similarity based on information entropy," *Computer Technology and Development*, vol. 25, no. 9, pp. 119–122, 2015.

[13] Q. S. Meng, *Information Theory*, Xi'an Jiaotong University Press, 1986.

[14] H. Li, *Statistical Learning Methods*, Tsinghua University Press, 2012.

[15] T. Wang, Y. Liu, T. Tang, and Y. Chen, "Dynamic data window fault detection method based on relative principal component analysis," *Transactions of China Electrotechnical Society*, vol. 28, no. 1, pp. 142–148, 2013.

[16] C. Blake, "UCI repository of machine learning databases," in *Neural Information Processing Systems*, 1998.

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 7849841, 9 pages
https://doi.org/10.1155/2017/7849841

Research Article

Detection of Intermittent Fault for Discrete-Time Systems with Output Dead-Zone: A Variant Tobit Kalman Filtering Approach

Jie Huang and Xiao He

Department of Automation, TNList, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Xiao He; hexiao@tsinghua.edu.cn

Received 10 September 2016; Accepted 9 January 2017; Published 7 February 2017

Academic Editor: William MacKunis

This paper is concerned with the intermittent fault detection problem for a class of discrete-time linear systems with output dead-zone. Dead-zone phenomenon exists in many real practical systems due to the employment of low-cost commercial off-the-shelf sensors. Two Bernoulli random variables are utilized to model the dead-zone effect and a variant formation of Tobit Kalman filter is brought forward to generate a residual that can indicate the occurrence of an intermittent fault. A numerical example is presented to demonstrate the effectiveness and applicability of the proposed technique. The statistical performance of the technique is illustrated as well.

1. Introduction

Due to the uncertainties in the external environment or the abrupt changes in signals, there may exist different kinds of faults yielding unacceptable or intolerable behaviors for the whole system. Since a fault can lead to bad influences or even disastrous consequences on the performance of systems, an accurate fault detection plays a significant role in designing a safe and reliable system. For the past years, many researchers have been investigating this problem and they have established numerous methods, including the model-free fault detection approaches [1–7] and model-based fault detection approaches [8]. Since system mathematical mode indicating evaluation of the system state can be acquired, model-based fault detection approaches occupy an important place in the practical application of diagnosis technology. The mode-based fault detection approaches can be further classified as observer-based approaches [9–13], parity equations approaches [14, 15], and parameter estimation methods [16, 17]. For the model-based approaches, a fault detection observer or filter is designed to detect the fault signal through generating a residual signal and then comparing the residual signal with a threshold [18–20]. The Kalman filter [21] is frequently used for residual signal generation by estimating the states to study the differences between predicted measurements and actual measurements.

At present, most researchers focus on detection of permanent faults and transient faults. Nevertheless, as the electronic technology and computer science have been rapidly developing, a special kind of faults, intermittent faults, comes into sight. Compared with permanent faults, the occurrence of intermittent faults is periodic, intermittent, and recurrent. Differed from transient faults, the intermittent faults can recur in the same component and disappear after changing component. With the cumulative effect, intermittent faults will turn to permanent faults, which are menaces for system performance and equipment safety [22]. There are a few literatures using quantitative analysis methods to investigate the detection problem of intermittent faults. Reference [23] considered the detection of scalar intermittent faults in continuous linear stochastic dynamic systems. Reference [24] looked into the intermittent fault detection problem for networked systems with unknown input and multiple state delays. In [25], a robust fault detection method was proposed to detect intermittent faults for linear stochastic systems in the presence of time-varying parametric perturbations and noises.

In real control systems, especially those making use of low-cost commercial sensors with poor calibration, dead-zone is one of the common sources of measurement nonlinearity. It can seriously limit the performance of systems and bring challenges to engineers. The model of dead-zone with input $v(t)$ and output $w(t)$ can be described by

$$w(t) = D(v(t)) = \begin{cases} h_r(v(t) - \tau_r), & v(t) \geq \tau_r, \\ 0, & \tau_l < v(t) < \tau_r, \\ h_l(v(t) - \tau_l), & v(t) \leq \tau_l, \end{cases} \quad (1)$$

where h_l and h_r are the left and right slopes with $h_l > 0$ and $h_r > 0$; τ_l and τ_r are the left and right break points with $\tau_l < 0$ and $\tau_r > 0$. The researches on the approaches to control the dead-zone can be traced back to Tao and Kokotovic [26] who constructed a continuous-time adaptive dead-zone inverse. Then, they extended it to a discrete-time formulation for linear systems with measurable dead-zone output [27]. After that, [28] presented an asymptotically adaptive elimination of an unknown dead-zone whose input and output are available. Around the beginning of the 21st century, a fuzzy dead-zone precompensator was established in [29] and neural network was applied to the construction of a precompensator in [30].

The Tobit model was first coming forward in [31] as a hybrid of probit analysis and multiple regression for household expenditure with censoring data. Although this model has been widely used in the fields of economics and medicine, it has not been well concerned in control engineering. Reference [32] presented a formulation of Kalman filter, named Tobit Kalman filter, which provided an efficient method to tackle the system with censoring data. In [32], a new definition of innovation was introduced by employing the Tobit regression.

A lot of methods have been proposed for estimation of nonlinear systems with censoring measurements. The extended Kalman filter (EKF) is a commonly used substitution of the Kalman filter when the nonlinear systems are encountered. However, when EKF is not convergent, its performance will deteriorate and become unstable [33]. The unscented Kalman filter (UKF) was devised to be an alternative of the EKF by improving convergence and linearization. Nevertheless, while data are censored, discontinuities will locate between the sigma points resulting in the biased measurement noise covariance [33]. Among those approaches, the particle filter in [34] is accurate but also most computationally expensive. It can cause difficulty in implement of the systems with limitations on computational power, like embedded systems [32]. Another defect is that the posterior weights will go to "collapse" as the particle filter is employed in some very large scale systems [35]. Some methods on fault-tolerant control of systems with dead-zone have been proposed recently [36, 37]. Compared with the aforementioned methods, the Tobit Kalman filter not only has less computational burden, but also has good performance while operating in the nonlinear system, which makes it more practical. This paper considers the fault detection for the discrete-time systems with output dead-zone. Since the dead-zone model does not fall under the category of Tobit model types, the Tobit Kalman filter here is a variant formulation which still maintains the performance of the original one. This variant Tobit Kalman filter will be used for designing the fault detection filter.

So far, there exist fairly rare researches on fault detection, especially the intermittent fault detection, for systems with output dead-zone. This paper has first proposed an intermittent fault detection method for a class of discrete-time systems with output dead-zone via the Tobit Kalman filtering approach, which has less computational expense and higher practicability. Also, the statistical performance will be illustrated in this paper. The remainder of this paper will be divided into five sections. Section 2 states the problem under consideration and the preliminaries of Tobit regression for the data with dead-zone. In Section 3, the variant Tobit Kalman filter will be derived. Section 4 is the part of designing the fault detection filter. Section 5 presents the simulation results and statistical performance. Finally, the conclusion is drawn in Section 6.

2. Problem Formulation and Preliminaries

2.1. Problem Formulation. The discrete-time system with faults to be detected is described as

$$\begin{aligned} x_{k+1} &= Ax_k + w_k + Bf_k, \\ y_k^* &= Cx_k + v_k, \\ y_k &= D(y_k^*), \end{aligned} \quad (2)$$

where $x_k \in \mathbb{R}^n$ is the state vector; $y_k^* \in \mathbb{R}^m$ is the latent measurement vector; $y_k \in \mathbb{R}^m$ is the observed measurement vector with a dead-zone; $f_k \in \mathbb{R}^p$ is the fault vector; $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ represent the Gaussian random vectors with zero mean and covariance Q and $R = \sigma^2$, respectively. The system matrices A, B, and C are constant and deterministic with proper dimensions.

The measurement with dead-zone is to be defined as

$$\begin{aligned} y_k &= D(y_k^*) \\ &= \begin{cases} h_r(y_k^* - T_r) = h_r(Cx_k + v_k - T_r), & y_k^* \geq T_r, \\ 0, & T_l < y_k^* < T_r, \\ h_l(y_k^* - T_l) = h_l(Cx_k + v_k - T_l), & y_k^* \leq T_l, \end{cases} \end{aligned} \quad (3)$$

where $T_l \in \mathbb{R}^m$ is a negative vector with elements $\tau_l(i)$s, representing the left breakpoint; $T_r \in \mathbb{R}^m$ is a positive vector with elements $\tau_r(i)$s, representing the right breakpoint; h_l and h_r are positive constants, representing the left and right slopes, separately. As the statement in [26], the above dead-zone model is a static simplification of different physical phenomena with ignorable fast dynamics.

In order to model the occurrence of dead-zone, two Bernoulli random variables are introduced.

$$\gamma_k(i) = \begin{cases} 1, & y_k^*(i) \geq \tau_r(i), \\ 0, & y_k^*(i) < \tau_r(i), \end{cases}$$
$$\eta_k(i) = \begin{cases} 1, & y_k^*(i) \leq \tau_l(i), \\ 0, & y_k^*(i) > \tau_l(i). \end{cases} \quad (4)$$

At any time step, the measurement y_k can be expressed as a combination of $Cx_k(i) + v_k(i)$ with probability $E(\gamma_k(i))$ or $E(\eta_k(i))$. When $\gamma_k(i) = 1$ or $\eta_k(i) = 1$, the latent measurements can be observed. When $\gamma_k(i) = 0$ and $\eta_k(i) = 0$, the data become latent values. Here, i is the index of elements in the measurement vector, where $i = 1, 2, \ldots, m$.

It should be noticed that the Bernoulli random matrices should be diagonal; that is, $\Gamma_k, H_k \in \mathbb{R}^{m\times m}$.

Consequently, the measurements can be rewritten as

$$y_k = \Gamma_k\left(h_r\left(Cx_k + v_k - \tau_r\right)\right) + H_k\left(h_l\left(Cx_k + v_k - \tau_l\right)\right). \quad (5)$$

2.2. Preliminaries. The fault detection filter in this paper is designed by using a variant Tobit Kalman filter. One of the significant points of the Tobit Kalman filter is introducing the innovation through the Tobit regression. As the Tobit regression is used for reference in the design of the fault detection filter with output dead-zone, some preliminaries will be introduced in this section. Notice that all the equations and values are scalars in this section.

Let $\Phi(y)$ represent the value of cumulative probability density function of unit-normal distribution at y.

Suppose that the values of $(Cx_k + v_k)$ and the limits τ_r, τ_l are known, and v_k follows the normal distribution with zero mean and standard deviation, σ; then

$$\begin{aligned} \Pr\left(y_k > y > 0\right) &= \Pr\left(h_r\left(Cx_k + v_k - \tau_r\right) > y\right) \\ &= \Pr\left(h_r v_k > y - h_r\left(Cx_k - \tau_r\right)\right) \\ &= \Phi\left(\frac{h_r\left(Cx_k - \tau_r\right) - y}{h_r\sigma}\right), \\ \Pr\left(y_k = 0\right) &= \Pr\left(\tau_l < y_k^* < \tau_r\right) \\ &= \Pr\left(\tau_l - Cx_k < v_k < \tau_r - Cx_k\right) \\ &= \Phi\left(\frac{\tau_r - Cx_k}{\sigma}\right) - \Phi\left(\frac{\tau_l - Cx_k}{\sigma}\right), \\ \Pr\left(y_k < y < 0\right) &= \Pr\left(h_l\left(Cx_k + v_k - \tau_l\right) < y\right) \\ &= \Pr\left(h_l v_k < y - h_l\left(Cx_k - \tau_l\right)\right) \\ &= \Phi\left(\frac{y - h_l\left(Cx_k - \tau_l\right)}{h_l\sigma}\right). \end{aligned} \quad (6)$$

The cumulative density function of y_k can be obtained according to (6):

$$F(y_k) = \begin{cases} \Phi\left(\dfrac{h_r\left(Cx_k - \tau_r\right) - y_k}{h_r\sigma}\right), & y_k > 0, \\ \Phi\left(\dfrac{\tau_r - Cx_k}{\sigma}\right) - \Phi\left(\dfrac{\tau_l - Cx_k}{\sigma}\right), & y_k = 0, \\ \Phi\left(\dfrac{y_k - h_l\left(Cx_k - \tau_l\right)}{h_l\sigma}\right), & y_k < 0. \end{cases} \quad (7)$$

The corresponding probability density function is

$$f(y_k) = \begin{cases} \dfrac{1}{h_r\sigma}\phi\left(\dfrac{h_r\left(Cx_k - \tau_r\right) - y_k}{h_r\sigma}\right), & y_k > 0, \\ \dfrac{1}{h_l\sigma}\phi\left(\dfrac{y_k - h_l\left(Cx_k - \tau_l\right)}{h_l\sigma}\right), & y_k < 0, \end{cases} \quad (8)$$

where $\phi(y)$ is the value of probability density function at y.

The expected value of y_k with a dead-zone is

$$\begin{aligned} E(y_k) &= \int_0^{+\infty} \frac{y}{h_r\sigma}\phi\left(\frac{h_r\left(Cx_k - \tau_r\right) - y}{h_r\sigma}\right)dy \\ &\quad + \int_{-\infty}^{0} \frac{y}{h_l\sigma}\phi\left(\frac{y - h_l\left(Cx_k - \tau_l\right)}{h_l\sigma}\right)dy \\ &= h_r\left(Cx_k - \tau_r\right)\Phi\left(\frac{Cx_k - \tau_r}{\sigma}\right) \\ &\quad + h_r\sigma\phi\left(\frac{Cx_k - \tau_r}{\sigma}\right) \\ &\quad + h_l\left(Cx_k - \tau_l\right)\Phi\left(\frac{\tau_l - Cx_k}{\sigma}\right) \\ &\quad - h_l\sigma\phi\left(\frac{\tau_l - Cx_k}{\sigma}\right). \end{aligned} \quad (9)$$

The variance of y_k with a dead-zone is

$$\begin{aligned} \mathrm{Var}\left(y_k\right) &= E\left(y^2\right) - \left(E\left(y\right)\right)^2 = \sigma^2\left(h_r^2\Psi\left(\frac{Cx_k - \tau_r}{\sigma}\right)\right. \\ &\quad + h_l^2\Psi\left(\frac{\tau_l - Cx_k}{\sigma}\right) \\ &\quad \left. + 2h_r h_l\chi\left(\frac{Cx_k - \tau_r}{\sigma}, \frac{\tau_l - Cx_k}{\sigma}\right)\right), \end{aligned} \quad (10)$$

where

$$\begin{aligned} \Psi(\alpha) &= \alpha^2\left(\Phi(\alpha)\left(1 - \Phi(\alpha)\right)\right) \\ &\quad + \alpha\left(\phi(\alpha)\left(1 - 2\Phi(\alpha)\right)\right) + \Phi(\alpha) \\ &\quad - \phi(\alpha)^2, \\ \chi(\alpha, \beta) &= \alpha\beta\Phi(\alpha)\Phi(\beta) + \alpha\Phi(\alpha)\phi(\beta) \\ &\quad + \beta\Phi(\beta)\phi(\alpha) + \phi(\alpha)\phi(\beta). \end{aligned} \quad (11)$$

3. Variant Tobit Kalman Filter

3.1. Time Update. With the updating of the time indices, the estimation of state before y_k taken into account is expressed as

$$\begin{aligned} x_{k|k-1} &= E\left(x_k \mid y_{1:k-1}\right) = E\left(\left(Ax_{k-1} + w_{k-1}\right) \mid y_{1:k-1}\right) \\ &= Ax_{k-1|k-1}, \end{aligned} \tag{12}$$

where $x_{k-1|k-1}$ is the estimate of state at time $k-1$ with all the measurements up to time $k-1$ being given.

The state error covariance matrix can be written as

$$\begin{aligned} P_{k|k-1} &= \operatorname{cov}\left(x_k - x_{k|k-1}\right) \\ &= \operatorname{cov}\left(Ax_{k-1} + w_{k-1} - Ax_{k-1|k-1}\right) \\ &= \operatorname{cov}\left(A\left(x_{k-1} - x_{k-1|k-1}\right) + w_{k-1}\right) \\ &= AP_{k-1|k-1}A^T + Q, \end{aligned} \tag{13}$$

where $P_{k-1|k-1}$ is corresponding state error covariance matrix of $x_{k-1|k-1}$ and x_k is the true value of the state at time k.

3.2. Measurement Update. The stage of measurement update is to rectify the estimate of state using the new information. As all measurements up to time k are given, the equation of the state is written as

$$x_{k|k} = x_{k|k-1} + K_k\left(y_k - E\left(y_k \mid x_{k|k-1}\right)\right). \tag{14}$$

The state error covariance matrix is

$$\begin{aligned} P_{k|k} &= \operatorname{cov}\left(x_k - x_{k|k}\right) \\ &= \operatorname{cov}\left(x_k - x_{k|k-1} - K_k\left(y_k - E\left(y_k \mid x_{k|k-1}\right)\right)\right), \end{aligned} \tag{15}$$

where $E(y_k \mid x_{k|k-1})$ is the expectation of measurement at time k, whose scalar value can be calculated by (8). In the rest of this paper, $E(y_k \mid x_{k|k-1})$ will be denoted as $E(y_k)$ for convenience.

The state error covariance matrix can be written by substituting (5) into (15).

$$\begin{aligned} P_{k|k} &= \operatorname{cov}\left(x_k - x_{k|k-1} - K_k\left(\Gamma_k\left(h_r\left(Cx_k + v_k - T_r\right)\right)\right.\right. \\ &\quad \left.\left. + H_k\left(h_l\left(Cx_k + v_k - T_l\right)\right) - E\left(y_k\right)\right)\right) = E\left(\left(x_k\right.\right. \\ &\quad \left.\left. - x_{k|k-1} - K_k\widetilde{y}_k\right)\left(x_k - x_{k|k-1} - K_k\widetilde{y}_k\right)^T\right) = P_{k|k-1} \\ &\quad - R_{\widetilde{x}\widetilde{y}_k}K_k^T - K_kR_{\widetilde{x}\widetilde{y}_k} + K_kR_{\widetilde{y}\widetilde{y}_k}K_k^T, \end{aligned} \tag{16}$$

where

$$\begin{aligned} \widetilde{y}_k &= y_k - E\left(y_k\right) \\ &= \Gamma_k\left(h_r\left(Cx_k + v_k - T_r\right)\right) \\ &\quad + H_k\left(h_l\left(Cx_k + v_k - T_l\right)\right) - E\left(y_k\right), \end{aligned} \tag{17}$$

$$R_{\widetilde{x}\widetilde{y}_k} = E\left(\left(x_k - x_{k|k-1}\right)\widetilde{y}_k^T\right), \tag{18}$$

$$R_{\widetilde{y}\widetilde{y}_k} = E\left(\widetilde{y}_k\widetilde{y}_k^T\right). \tag{19}$$

Take the trace of the state error covariance matrix described in (16) and then set the deviation of the trace equal to zero. Then, the optimal Kalman gain can be found so as to minimize the state error covariance.

$$\frac{d\operatorname{tr}\left(P_{k|k}\right)}{dK_k} = -2R_{\widetilde{x}\widetilde{y}_k} + 2K_kR_{\widetilde{y}\widetilde{y}_k} = 0, \tag{20}$$

$$K_k = R_{\widetilde{x}\widetilde{y}_k}R_{\widetilde{y}\widetilde{y}_k}^{-1}. \tag{21}$$

Substituting (17) into (18) leads to

$$\begin{aligned} R_{\widetilde{x}\widetilde{y}_k} &= E\left(\left(x_k - x_{k|k-1}\right)\widetilde{y}_k^T\right) = E\left(\left(x_k - x_{k|k-1}\right)\right. \\ &\quad \cdot\left(\Gamma_k\left(h_r\left(Cx_k + v_k - T_r\right)\right)\right. \\ &\quad \left.\left. + H_k\left(h_l\left(Cx_k + v_k - T_l\right)\right) - E\left(y_k\right)\right)^T\right) \\ &= E\left(x_k\left(Cx_k + v_k\right)^T h_r\Gamma_k - x_kT_r^Th_r\Gamma_k + x_k\left(Cx_k\right.\right. \\ &\quad \left.\left. + v_k\right)^T h_lH_k - x_kT_l^Th_lH_k - x_{k|k-1}\left(Cx_k + v_k\right)^T h_r\Gamma_k \right. \\ &\quad + x_{k|k-1}T_r^Th_r\Gamma_k - x_{k|k-1}\left(Cx_k + v_k\right)^T h_lH_k \\ &\quad \left. + x_{k|k-1}T_l^Th_lH_k\right). \end{aligned} \tag{22}$$

Since the expected value of a Bernoulli random variable equals success probability, then

$$\begin{aligned} E\left(\gamma_k\left(i,i\right)\right) &= \Pr\left(y_k^*\left(i\right) \geq \tau_r\left(i\right)\right) \\ &= \Phi\left(\frac{Cx_k\left(i\right) - \tau_r\left(i\right)}{\sigma\left(i\right)}\right), \\ E\left(\eta_k\left(i,i\right)\right) &= \Pr\left(y_k^*\left(i\right) \leq \tau_l\left(i\right)\right) \\ &= \Phi\left(\frac{\tau_l\left(i\right) - Cx_k\left(i\right)}{\sigma\left(i\right)}\right). \end{aligned} \tag{23}$$

In principle, the value of true state should be applied in the calculation. The assumptions in [32] will be used to reduce the constraints.

Assumption 1 (see [32]). For small estimation errors, the prediction of state can be used to obtain a sufficiently accurate estimate of the success probability; that is,

$$\begin{aligned} E\left(\gamma_k\left(i,i\right)\right) &= \Phi\left(\frac{Cx_k\left(i\right) - \tau_r\left(i\right)}{\sigma\left(i\right)}\right) \\ &\approx \Phi\left(\frac{Cx_{k|k-1}\left(i\right) - \tau_r\left(i\right)}{\sigma\left(i\right)}\right), \\ E\left(\eta_k\left(i,i\right)\right) &= \Phi\left(\frac{\tau_l\left(i\right) - Cx_k\left(i\right)}{\sigma\left(i\right)}\right) \\ &\approx \Phi\left(\frac{\tau_l\left(i\right) - Cx_{k|k-1}\left(i\right)}{\sigma\left(i\right)}\right). \end{aligned} \tag{24}$$

Remark 2. As Assumption 1 holds true, the state x_k can be considered as independent of the Bernoulli variables Γ_k and H_k.

Assumption 3 (see [32]). In most applications, the R matrices are diagonal, which means that the measurement noise is independent in the measurements.

According to the assumptions above,

$$\begin{aligned} E(\Gamma_k) = \operatorname{diag}\Bigg(&\Phi\left(\frac{Cx_{k|k-1}(1)-\tau_r(1)}{\sigma(1)}\right),\\ &\Phi\left(\frac{Cx_{k|k-1}(2)-\tau_r(2)}{\sigma(2)}\right),\ldots,\\ &\Phi\left(\frac{Cx_{k|k-1}(m)-\tau_r(m)}{\sigma(m)}\right)\Bigg),\\ E(T_k) = \operatorname{diag}\Bigg(&\Phi\left(\frac{\tau_l(1)-Cx_{k|k-1}(1)}{\sigma(1)}\right),\\ &\Phi\left(\frac{\tau_l(2)-Cx_{k|k-1}(2)}{\sigma(2)}\right),\ldots,\\ &\Phi\left(\frac{\tau_l(m)-Cx_{k|k-1}(m)}{\sigma(m)}\right)\Bigg). \end{aligned} \tag{25}$$

$R_{\tilde{x}\tilde{y}_k}$ can be written in terms of Assumption 1:

$$\begin{aligned} R_{\tilde{x}\tilde{y}_k} &= E\left(x_k x_k^T\right) C^T h_r E(\Gamma_k) + E(x_k) h_r E(v_k\Gamma_k)\\ &\quad - E(x_k) T_r^T h_r E(\Gamma_k) + E\left(x_k x_k^T\right) C^T h_l E(H_k)\\ &\quad + E(x_k) h_l E(v_k H_k) - E(x_k) T_l^T h_l E(H_k)\\ &\quad - x_{k|k-1} E\left(x_k^T\right) C^T h_r E(\Gamma_k) - x_{k|k-1} h_r E(v_k\Gamma_k)\\ &\quad + x_{k|k-1} T_r^T h_r E(\Gamma_k)\\ &\quad - x_{k|k-1} E\left(x_k^T\right) C^T h_l E(H_k)\\ &\quad - x_{k|k-1} h_l E(v_k H_k) + x_{k|k-1} T_l^T h_l E(H_k)\\ &= P_{k|k-1} C^T h_r E(\Gamma_k) + P_{k|k-1} C^T h_l E(H_k). \end{aligned} \tag{27}$$

Compute $R_{\tilde{y}\tilde{y}_k}$ by Assumptions 1 and 3.

$$\begin{aligned} R_{\tilde{y}\tilde{y}_k} &= E(\Gamma_k) h_r C P_{k|k-1} C^T h_r E(\Gamma_k)\\ &\quad + E(\Gamma_k) h_r C P_{k|k-1} C^T h_l E(H_k)\\ &\quad + E(H_k) h_l C P_{k|k-1} C^T h_l E(H_k)\\ &\quad + E(H_k) h_l C P_{k|k-1} C^T h_r E(\Gamma_k) + \operatorname{Var}(y_k), \end{aligned} \tag{28}$$

where

$$\begin{aligned} \operatorname{Var}(y_k) = \operatorname{diag}\big(&\operatorname{Var}(y_k(1)), \operatorname{Var}(y_k(2)),\ldots,\\ &\operatorname{Var}(y_k(m))\big). \end{aligned} \tag{29}$$

Substituting (27), (28), and (21) into (16) yields

$$P_{k|k} = \left(I - K_k\left(E(\Gamma_k) h_r + E(H_k) h_l\right) C\right) P_{k|k-1}. \tag{30}$$

The full view of the variant Tobit Kalman filter is

$$\begin{aligned} x_{k|k-1} &= A x_{k-1|k-1},\\ P_{k|k-1} &= A P_{k-1|k-1} A^T + Q,\\ K_k &= R_{\tilde{x}\tilde{y}_k} R_{\tilde{y}\tilde{y}_k}^{-1},\\ x_{k|k} &= x_{k|k-1} + K_k\left(y_k - E(y_k)\right),\\ P_{k|k} &= \left(I - K_k\left(E(\Gamma_k) h_r + E(H_k) h_l\right) C\right) P_{k|k-1}, \end{aligned} \tag{31}$$

where $R_{\tilde{x}\tilde{y}_k}$, $R_{\tilde{y}\tilde{y}_k}$, $E(\gamma_k)$, and $E(\eta_k)$ are defined as (27), (28), (25), and (26), respectively.

4. Fault Detection

In the fault detection, it is expected that the reconstructed process variables derived by the filter will follow the corresponding real values of the fault-free operating states. To get information on whether a fault occurs, the measured variables will be compared with their estimates delivered by the filter. The difference between the measurements and their estimates is defined as a residual. Therefore, a residual generation is the most significant procedure for a successful fault detection [8].

After the estimation of the states, the estimates of outputs $\hat{y}_k$ are created as shown in

$$\begin{aligned} \hat{y}_k &= D(\hat{y}_k^*)\\ &= \begin{cases} h_r(\hat{y}_k^* - T_r) = h_r(Cx_{k|k-1} - T_r), & \hat{y}_k^* \geq T_r,\\ 0, & T_l < \hat{y}_k^* < T_r,\\ h_l(\hat{y}_k^* - T_l) = h_l(Cx_{k|k-1} - T_l), & \hat{y}_k^* \leq T_l. \end{cases} \end{aligned} \tag{32}$$

Then, the residual vector is built as the difference between the measurements and their estimates:

$$r_k = \hat{y}_k - y_k. \tag{33}$$

The residual evaluation function is used for differentiating the fault from disturbance and uncertainties. This procedure of postprocessing the residuals takes out the information about the fault of interest from the residual signals. After calculating the residual vector, substitute it into the specified evaluation function and compare the evaluation value with the preset threshold. If the residual evaluation value is larger than the threshold, an alarm of fault will be built.

Consider the time-windowed root-mean-square (RMS) norm as the evaluation function:

$$\|r_k\|_e = \sqrt{\frac{1}{N}\sum_{j=0}^{N-1} r_{k-j}^T r_{k-j}}. \tag{34}$$

Then, choose a threshold of the following form:

$$J_{\text{th}} = \sup_{f_k=0} \mathbb{E}\left[\|r_k\|_e\right]. \tag{35}$$

Remark 4. In the practical applications, the threshold is considered as the maximum value of the residual evaluation function in the fault-free case through the Monte Carlo method.

Remark 5. Making choice of the threshold should be compromised based on the actual situation. As the value of threshold is increasing, the false alarm rate will reduce whereas the missing alarm rate will rise up. With the decrease of the threshold value, the false alarm rate will go up but the missing alarm rate will be lower. Hence, to choose a threshold needs to consider overall interests.

The relationship between the value of residual evaluation function and the threshold should be satisfied as

$$\begin{aligned} \|r_k\|_e > J_{\text{th}} &\Longrightarrow \text{fault detected}, \\ \|r_k\|_e \le J_{\text{th}} &\Longrightarrow \text{no faults}. \end{aligned} \tag{36}$$

The algorithm of the fault detection filter is summarized as follows.

Algorithm 6 (fault detection using a variant Kalman filter). The initial conditions $x_{0|0}$ and $P_{0|0}$ are given.

Step 1. Compute $x_{k|k-1}$ and $P_{k|k-1}$ using (12) and (13).
Step 2. Compute $E(\Gamma_k)$, $E(H_k)$, $E(y_k)$, and $\text{Var}(y_k)$ using (25), (26), (9), and (10).
Step 3. Compute $R_{\bar{x}\bar{y}_k}$ and $R_{\bar{y}\bar{y}_k}$ using (27) and (28).
Step 4. Compute K_k using (21).
Step 5. Compute $x_{k|k}$ and $P_{k|k}$ using (14) and (30).
Step 6. Compute $\widehat{y}_k$ using (32) and then obtain r_k through (33).
Step 7. Evaluate r_k using (34) and then compare it with the threshold using the rule (36).
Step 8. Return to Step 1.

5. Simulation Results

In this section, the simulation result will illustrate the applicability of the fault detection filter using the variant Tobit Kalman filtering.

Consider the tracking system of ballistic roll rates explored in [32], which has the dynamic model of (2) with the state-space matrices:

$$\begin{aligned} A &= \alpha \begin{bmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix}. \end{aligned} \tag{37}$$

In this example, $\alpha = 1$; the frequency $\omega = 0.005 \times 2\pi$; the standard deviation of process noise w_k is 0.05; the variance of measurement noise v_k is 1.

The dead-zone of measurement is given as

$$y_k = D\left(y_k^*\right) = \begin{cases} 0.2\left(Cx_k + v_k - 3\right), & y_k^* \ge 3, \\ 0, & -6 < y_k^* < 3, \\ 0.3\left(Cx_k + v_k + 6\right), & y_k^* \le -6; \end{cases} \tag{38}$$

that is, $h_r = 0.2$, $h_l = 0.3$, $T_r = 3$, and $T_l = -6$.

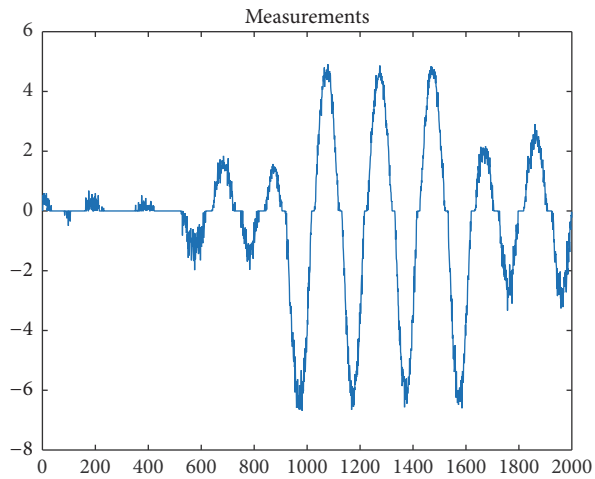


Figure 1: Observed measurements of system.

The initial conditions are set as

$$\begin{aligned} x_0 &= \begin{bmatrix} 5 \\ 0 \end{bmatrix}, \\ P_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{39}$$

Then, an intermittent fault is added to the system, which is described as follows:

$$f_k = \begin{cases} 1, & 500 \le k \le 600, \\ 1.2, & 880 \le k \le 1000, \\ 1, & 1600 \le k \le 1700, \\ 0, & \text{otherwise}, \end{cases} \tag{40}$$

with coefficient matrix

$$B = \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix}. \tag{41}$$

The simulation results are shown in Figures 1, 2, and 3.

Figure 1 is the figure of the measurements with the dead-zone. It can be seen that some parts of measurements have distortion or cannot be observed because of the dead-zone. Figure 2 presents the estimates of the processing states. The blue curve represents the true states and the red dash curve represents the estimates of states using the variant Kalman filter. Even though the data are absent or distorted, the estimation values of states using the variant Tobit Kalman filter can closely track the true states in fault-free case. When the faults occur, the difference between the true states and their estimates is obvious. Accordingly, the outputs of the variant Tobit Kalman filter provide accurate estimation of the measurements to generate the residual signal.

Figure 3 illustrates f_k, residual response, and residual evaluation function response varying with time k. The faults occur in the time intervals $[500, 600]$, $[880, 1000]$,

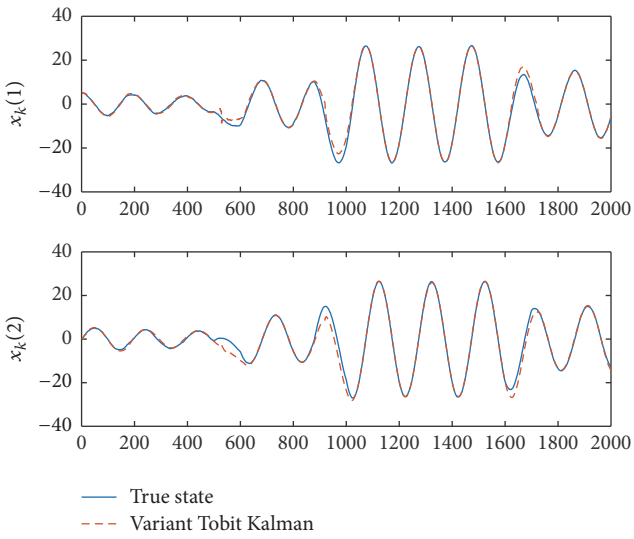


Figure 2: Estimation of states.

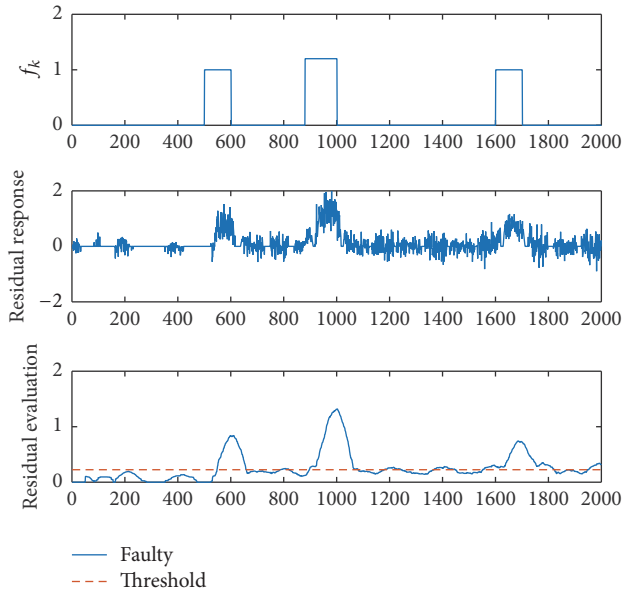


Figure 3: f_k, residual response and residual evaluation.

and [1600, 1700] with the magnitude and coefficient matrix defined as (40) and (41). In the figure of residual evaluation, the blue curve is the response of faulty systems and the red dash line represents the preset threshold. Figure 3 indicates that the response raises an alarm opportunely when the faults happen at time $k = 500$, $k = 880$, and $k = 1600$. When the faults disappear at times $k = 600$, $k = 1000$, and $k = 1700$, the alarms go down rapidly. These simulation results demonstrate that the residual can detect the intermittent faults accurately.

The Monte Carlo method is applied to obtain the statistical performance of the fault detection method proposed in this paper. The statistical performance is shown in Table 1. Fault detection time (FDT) represents the time when the occurrence of fault is detected and the alarm raises. False alarm rate (FAR) is the probability of false detection and missing alarm rate (MAR) is the probability of missing detection. It can be seen that the value of threshold can influence a lot the FDT, FAR, and MAR. As the value of threshold is becoming larger, it takes more time to detect the fault and raise alarm. The smaller threshold can reduce the MAR. However, the FAR is increasing at the same time. When threshold is chosen as a large value, although there will be less or even no false alarms, more faults will be failed to detect. Therefore, it is significant to choose a proper value of threshold and make a tradeoff among the FDT, FAR, and MAR.

Table 1: Statistical performance.

Threshold	0.1846	0.2144	0.2223	0.2625	0.3492
FDT	688	694	699	718	761
FAR (%)	61.67	13	9	1.67	0
MAR (%)	3.67	5.67	7.67	8.67	42

6. Conclusion

In this paper, we have investigated the problem of fault detection for the discrete-time systems with output dead-zone. Two Bernoulli random variables are introduced to describe the dead-zone of measurements. The variant Tobit Kalman filter is utilized for designing a recursive fault detection filter under the situation of dead-zone measurements. A simulation example of the occurrence of intermittent faults is provided. The simulation results have illustrated that the performance of fault detection filter is satisfied, where the fault can be recognized quickly and accurately. Also, the statistical performance of the fault detection method illustrated the relationship of threshold, FDT, FAR, and MAR. The dead-zone is a common issue arising in the engineering applications. The fault detection filter using the variant Tobit Kalman filter shows its practical value, with the good performance and lower computational expense. Furthermore, it can be extended to the fault detection for two-dimension systems like vision-based systems with occlusion region.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61473163, 61522309, and 61490701 and Tsinghua University Initiative Scientific Research Program.

References

[1] X. Sun, H. J. Marquez, T. Chen, and M. Riaz, "An improved PCA method with application to boiler leak detection," *ISA Transactions*, vol. 44, no. 3, pp. 379–397, 2005.

[2] D. Diallo, M. E. H. Benbouzid, D. Hamad, and X. Pierre, "Fault detection and diagnosis in an induction machine drive: a pattern recognition approach based on concordia stator mean current vector," *IEEE Transactions on Energy Conversion*, vol. 20, no. 3, pp. 512–519, 2005.

[3] J. Liu, "Shannon wavelet spectrum analysis on truncated vibration signals for machine incipient fault detection," *Measurement Science and Technology*, vol. 23, no. 5, Article ID 055604, 2012.

[4] S. M. El-Shal and A. S. Morris, "A fuzzy expert system for fault detection in statistical process control of industrial processes," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 30, no. 2, pp. 281–289, 2000.

[5] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," *Mechanical Systems and Signal Processing*, vol. 18, no. 3, pp. 625–644, 2004.

[6] L. B. Jack and A. K. Nandi, "Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms," *Mechanical Systems and Signal Processing*, vol. 16, no. 2-3, pp. 373–390, 2002.

[7] P. V. J. Rodríguez, M. Negrea, and A. Arkkio, "A simplified scheme for induction motor condition monitoring," *Mechanical Systems and Signal Processing*, vol. 22, no. 5, pp. 1216–1236, 2008.

[8] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Springer, London, UK, 2nd edition, 2013.

[9] Z. Gao, X. Liu, and M. Z. Q. Chen, "Unknown input observer-based robust fault estimation for systems corrupted by partially decoupled disturbances," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2537–2547, 2016.

[10] L. Wu, X. Yao, and W. X. Zheng, "Generalized H_2 fault detection for two-dimensional Markovian jump systems," *Automatica*, vol. 48, no. 8, pp. 1741–1750, 2012.

[11] S. Tong, B. Huo, and Y. Li, "Observer-based adaptive decentralized fuzzy fault-tolerant control of nonlinear large-scale systems with actuator failures," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 1, pp. 1–15, 2014.

[12] S. Tong, T. Wang, and Y. Li, "Fuzzy adaptive actuator failure compensation control of uncertain stochastic nonlinear systems with unmodeled dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 563–574, 2014.

[13] H. J. Yang, Y. Q. Xia, and B. Liu, "Fault detection for T-S fuzzy discrete systems in finite-frequency domain," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 4, pp. 911–920, 2011.

[14] J. Blesa, P. Jiménez, D. Rotondo, F. Nejjari, and V. Puig, "An interval NLPV parity equations approach for fault detection and isolation of a wind farm," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3794–3805, 2015.

[15] S. Tornil-Sin, C. Ocampo-Martinez, V. Puig, and T. Escobet, "Robust fault diagnosis of nonlinear systems using interval constraint satisfaction and analytical redundancy relations," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 1, pp. 18–29, 2014.

[16] R. Isermann, "Fault diagnosis of machines via parameter estimation and knowledge processing—tutorial paper," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 29, no. 4, pp. 815–835, 1993.

[17] J.-A. Jiang, J.-Z. Yang, Y.-H. Lin, C.-W. Liu, and J.-C. Ma, "An adaptive PMU based fault detection/location technique for transmission lines part I: theory and algorithms," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 486–493, 2000.

[18] X. He, Z. Wang, Y. Liu, and D. H. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1670–1679, 2013.

[19] X. He, Z. Wang, X. Wang, and D. H. Zhou, "Networked strong tracking filtering with multiple packet dropouts: algorithms and applications," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1454–1463, 2014.

[20] X. He, Z. Wang, L. Qin, and D. Zhou, "Active fault-tolerant control for an internet-based networked three-tank system," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2150–2157, 2016.

[21] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[22] D.-H. Zhou, J.-T. Shi, and X. He, "Review of intermittent fault diagnosis techniques for dynamic systems," *Acta Automatica Sinica*, vol. 40, no. 2, pp. 161–171, 2014.

[23] M. Chen, G. Xu, R. Yan, S. X. Ding, and D. Zhou, "Detecting scalar intermittent faults in linear stochastic dynamic systems," *International Journal of Systems Science*, vol. 46, no. 8, pp. 1337–1348, 2015.

[24] X. He, Y. Hu, and K. Peng, "Intermittent fault detection for uncertain networked systems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 282168, 10 pages, 2013.

[25] R. Yan, X. He, and D. Zhou, "Detection of intermittent faults for linear stochastic systems subject to time-varying parametric perturbations," *IET Control Theory & Applications*, vol. 10, no. 8, pp. 903–910, 2016.

[26] G. Tao and P. V. Kokotovic, "Adaptive control of plants with unknown dead-zones," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 39, no. 1, pp. 59–68, 1994.

[27] G. Tao and P. V. Kokotović, "Discrete-time adaptive control of systems with unknown deadzones," *International Journal of Control*, vol. 61, no. 1, pp. 1–17, 1995.

[28] H. Cho and E.-W. Bai, "Convergence results for an adaptive dead zone inverse," *International Journal of Adaptive Control and Signal Processing*, vol. 12, no. 5, pp. 451–466, 1998.

[29] F. L. Lewis, W. K. Tim, L.-Z. Wang, and Z. X. Li, "Deadzone compensation in motion control systems using adaptive fuzzy logic control," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 6, pp. 731–742, 1999.

[30] R. R. Selmic and F. L. Lewis, "Deadzone compensation in motion control systems using neural networks," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 45, no. 4, pp. 602–613, 2000.

[31] J. Tobin, "Estimation of relationships for limited dependent variables," *Econometrica*, vol. 26, no. 1, pp. 24–36, 1958.

[32] B. Allik, C. Miller, M. J. Piovoso, and R. Zurakowski, "The Tobit Kalman filter: an estimator for censored measurements," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 365–371, 2016.

[33] B. Allik, C. Miller, M. J. Piovoso, and R. Zurakowski, "Nonlinear estimators for censored data: a comparison of the EKF, the UKF and the Tobit Kalman filter," in *Proceedings of the 2015 American Control Conference (ACC '15)*, pp. 5146–5151, Chicago, Ill, USA, July 2015.

[34] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 64, no. 4, pp. 827–836, 2002.

[35] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008.

[36] M. Chen, B. Jiang, and W. W. Guo, "Fault-tolerant control for a class of non-linear systems with dead-zone," *International Journal of Systems Science*, vol. 47, no. 7, pp. 1689–1699, 2016.

[37] M. Chen and G. Tao, "Adaptive fault-tolerant control of uncertain nonlinear large-scale systems with unknown dead zone," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1851–1862, 2016.

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 1763934, 10 pages
https://doi.org/10.1155/2017/1763934

Research Article

Intelligent Vehicle Embedded Sensors Fault Detection and Isolation Using Analytical Redundancy and Nonlinear Transformations

Nicolas Pous,[1] Denis Gingras,[1] and Dominique Gruyer[2]

[1]*Laboratory of Intelligent Vehicles (LIV), Sherbrooke University, Sherbrooke, QC, Canada*
[2]*IFSTTAR-IM-LIVIC, Versailles-Satory, France*

Correspondence should be addressed to Nicolas Pous; pous.nicolas@gmail.com

Received 11 August 2016; Accepted 6 December 2016; Published 16 January 2017

Academic Editor: Gang Li

This work proposes a fault detection architecture for vehicle embedded sensors, allowing to deal with both system nonlinearity and environmental disturbances and degradations. The proposed method uses analytical redundancy and a nonlinear transformation to generate the residual value allowing the fault detection. A strategy dedicated to the optimization of the detection parameters choice is also developed.

1. Introduction

Safety in intelligent vehicle is a key issue. In order to insure driver safety, it is of a high importance that all the information given by embedded sensors is more reliable. Indeed, a large number of applications are based on data fusion of information coming from several sensors, especially in vehicle localization application, as depicted in [1–3].

In the aerospace domain, physical redundancy and a voting system are often used, consisting in the direct comparison of the information provided by at least three identical sensors or systems and then the validation of the recorder data.

However, in industrial fields as the automotive industry, duplicating sensors correspond to a loss of profit and so solutions permitting the verification of the sensors confidence without any supplementary sensors have to be developed.

A large number of fault detectors have been developed during the past decades, to deal with complex systems [4–14]. A large number of them are based on system model [15–17] consisting in the comparison of the predicted behavior of the system and the information generated by the sensors, allowing to determine the system current state. This kind of method needs a perfect knowledge and model of the system behavior to work efficiently. In the studied case, the vehicle manoeuvers could present strong nonlinearity which will add complexity to the system modelling. Some solutions have been proposed to deal with this problems [18–20] but in the case of an automobile application strong and unpredictable environmental interaction could be added to behavior nonlinearity, and model-based solutions will be less efficient.

Considering this context, a solution using analytical redundancy seems a valuable alternative. Analytical redundancy allows comparing the estimation of a chosen metric from sensors of different types in order to detect and identify deviant comportment [21]. Huang and Su have proposed such a solution in [22] with the use of a set of extended Kalman filters to compare the estimated state of an ego-vehicle from different parallel filters, but this solution still needs to determine a system model to work optimally.

Our proposed solution is based on analytical redundancy using nonlinear transformations to generate residual signals in order to detect sensors faults. The use of nonlinear transformations allows improving detection robustness. The paper will be shared in five parts: first, the presentation of the architecture and some generalities are found in Section 2 and then the nonlinear transformation used will be studied in Section 3 and finally the decision process will be discussed in the fourth part. Some simulation results will be presented in Section 5 and, finally, Section 6 will conclude this paper and will provide some future works.

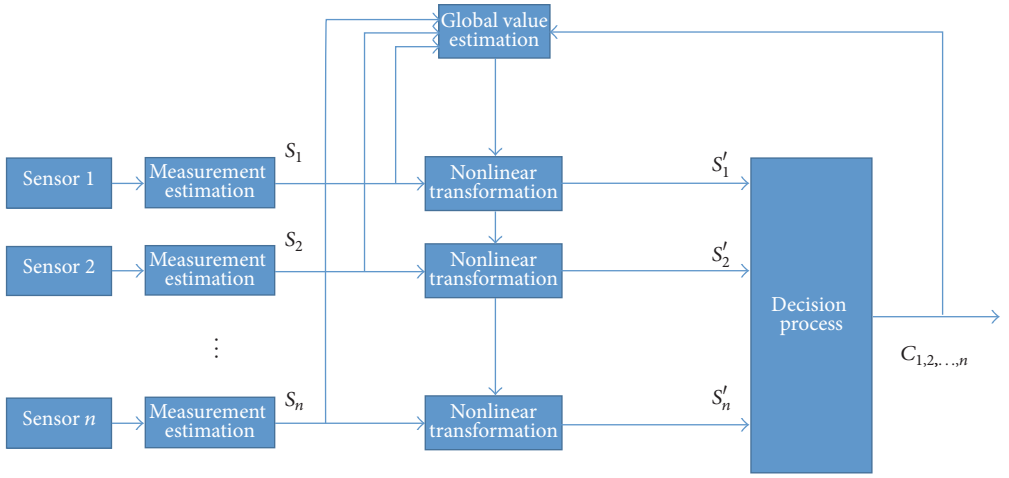


Figure 1: Proposed FDI architecture.

2. Generalities and Architecture

2.1. Architecture. The proposed architecture (Figure 1) is divided into two consecutive transformations which are dedicated to the generation of the residual signal S'. The first transformation allows obtaining a common measurement and the nonlinear transformation (TNL), giving the residual quantity. The second transformation consists in the decision process. The architecture can so be generalized for any type of sensor, according to a measurement under test S, which has to be estimated from every sensor.

The C values correspond to the decision for each sensor, taken into account in the global value estimation. $C = 1$ means a sensor presents a faulty behavior ($C = 0$ corresponding to a nominal behavior). The first necessary step to apply this method is to define the measurements used and apply the transformation to ensure the analytical redundancy. If more than one measurement is chosen to complete the fault detection, this architecture will be applied on every measurement used in the fault detection, and a sensor will be considered faulty if at least one of its corresponding decision value is 1.

$$C_S = C_{s,1} \bullet C_{s,2} \bullet \cdots C_{s,M}, \tag{1}$$

where $C_{s,m}$ represents the Boolean decision values for the measurement m and the sensor s. It is so primordial to define which measurements have to be tested according to the monitored sensors. Measurements have to be generated by at least three different sensors or sets of sensors and need to ensure the faults observability. These two requirements will be discussed in the two following sections for a specific example.

2.2. Analytical Redundancy. In order to implement this method, we need, in the first time, to determine the measurements which will allow us to make the comparison in order to study the tested sensor. In our case, we chose to study proprioceptive sensors, usually used to predict an ego-vehicle positioning state, to get inertial information from the

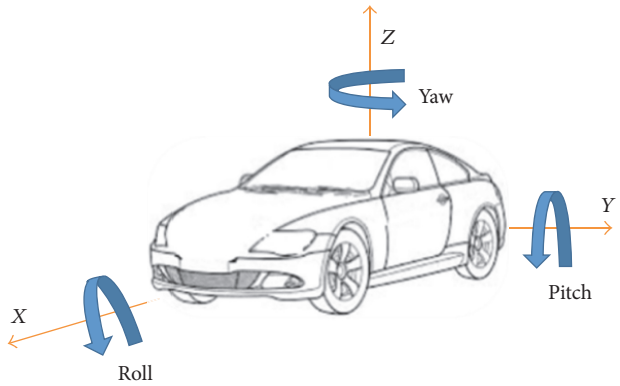


Figure 2: Vehicle axes presentation.

Inertial Navigation System (INS) and to provide odometric data needed to get vehicle speed (longitudinal or wheel speed). Odometers permit us to measure each wheel speed and distance travelled, when the INS will be used to measure vehicle acceleration and yaw rates on the 3 axes (see Figure 2).

Using only this set of sensors, it is possible to determine vehicle yaw rate and longitudinal acceleration, respectively, named $V\theta$ and Acc_x. This information is directly given by the INS sensor, when it has to be deduced from the odometric speed and distance. Using each left-right pair of odometers independently, it is possible to determine both measurements with the front and back couples of sensors. The longitudinal acceleration is then given by the following when the yaw rate is given by (5):

$$Acc_x(t) = \frac{dV_{\text{veh}}(t)}{dt} = \frac{V_{\text{veh}}(t) - V_{\text{veh}}(t-T)}{T}, \tag{2}$$

where V_{veh} is the vehicle speed determined with (3) with $V_{O.L}$ and $V_{O.R}$ being, respectively, the left and right odometric speeds, and T is a sampling period.

$$V_{\text{veh}}(t) = \frac{V_{O.R}(t) - V_{O.L}(t)}{2}. \tag{3}$$

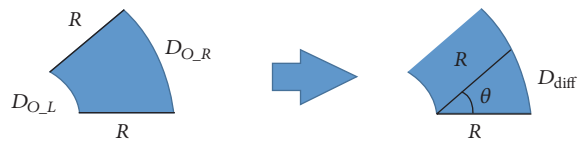


Figure 3: Yaw rate determination.

Now using the odometric distances D_{O_L} and D_{O_R}, it is possible to approximate the yaw rate by calculating the differential distance, D_{Diff}, which can be approximate as the difference between the right and left distances. In Figure 3, R represents the distance between the two wheels, usually known, depending on the vehicle characteristics.

It is then possible to determine the angle θ using (4) and then the yaw rate (5).

$$\theta(t) = \frac{D_{O\_R} - D_{O\_L}}{R} = \frac{D_{\text{Diff}}(t)}{R}, \tag{4}$$

$$V\theta_z(t) = \frac{d\theta(t)}{dt} = \frac{\theta(t)}{T}. \tag{5}$$

2.3. Observability. In order to be efficient, the transformation stage has to ensure that fault on one sensor can still be noticed on the transformed measurement. Basically, it consists in verifying that the measurement during a faulty behavior S_F is different than the nominal behavior S_{NF}. In order to analyze the observability capability, we use faults models presented in Qi et al. [23], which consider that sensors noises and errors can be classified in four categories:

(i) An additive bias:

$$y_F(t) = y_{\text{NF}}(t) + \Delta. \tag{6}$$

(ii) A scale factor:

$$y_F(t) = \alpha * y_{\text{NF}}(t). \tag{7}$$

(iii) An aberrant error:

$$y_F(t) = y_{\text{NF}}(t) + a\delta(t - t_0). \tag{8}$$

(iv) A total loss, which corresponds to an output value of the sensor stuck at 0 or registering only a stochastic process.

Knowing these four faults levels and the transformation applied on the sensors data, it is now possible to determine the efficiency of the proposed architecture. Concerning the INS, every type of fault is observable as there is no transformation, but it has to be determined for the odometers.

First, a bias applied on only one odometric distance (or speed) will be mostly masked by the estimation of the acceleration. If the bias is a constant value, then it will generate no acceleration. However, during the appearance of the fault, there will be a large instantaneous acceleration which can be seen as a Dirac impulsion. The bias on odometric speed will so generate an abnormal aberrant behavior on the acceleration. It will also generate a bias on the yaw rate $V\theta z$ (positive or negative depending on the affected odometer) which is easier to detect. Here, the bias is applied on the right odometer measurement. Knowing these four faults nature and the transformation used, it is now possible to determine the efficiency of the proposed architecture. Concerning the INS, every kind of fault is observable as there is no transformation, but it has to be determined for the odometers. In the following equations, the nonfaulty measurements will be noted as $V\theta_{zNF}$ and Acc_{xNF} when the faulty measurements will be $V\theta_{zF}$ and Acc_{xF}, respectively, for the yaw rate and the longitudinal acceleration. Our objective here is to extract the nonfaulty value and observe the deviation introduced by the injected fault.

$$V\theta_{zF}(t) = \frac{\left(D_{O_R} + \Delta\right) - D_{O_L}}{R * T} = \frac{\Delta}{R * T} + V\theta_{z\text{NF}}(t). \tag{9}$$

Now applying a scale factor to the right odometric speed, the acceleration will be affected (10), but the resulting fault appears as a gain and a bias which is depending on the nonfaulty value from the other odometer. The yaw rate will be affected in the same manner (11).

$$Acc_{xF}(t) = \frac{\left(\alpha V_{O\_R}(t) + V_{O\_L}(t)\right) - \left(\alpha V_{O\_R}(t - T) + V_{O\_L}(t - T)\right)}{2T} \tag{10}$$

$$Acc_{xF}(t) = \alpha Acc_{x\text{NF}}(t) - \frac{(\alpha - 1)\left(V_{O\_L}(t) - V_{O\_L}(t - T)\right)}{2T}$$

$$V\theta_{zF}(t) = \alpha V\theta_{z\text{NF}}(t) + \frac{(\alpha - 1) D_{O\_L}(t)}{RT} \tag{11}$$

A total loss, usually represented by the sensor's output stuck at 0, can be noticed on both acceleration (12) and yaw rate (13).

$$Acc_{xF}(t) = \frac{\left(0 + V_{O\_L}(t)\right) - \left(0 + V_{O\_L}(t - T)\right)}{2T}, \tag{12}$$

$$Acc_{xF}(t) = Acc_{x\text{NF}}(t) - \frac{V_R(t) - V_R(t - T)}{2T}.$$

Here, the acceleration obtains an additive term corresponding to the real acceleration of the affected wheel divided by two ($V_R(t)$ representing the real right wheel speed). The yaw rate will also be affected by an additive term proportional to the real distance travelled by the affected wheel, D_R, as follows:

$$V\theta_{zF}(t) = \frac{0 - D_{O\_L}}{R * T} = V\theta_{z\text{NF}}(t) - \frac{D_R}{R * T}. \tag{13}$$

An aberrant error will also lead to an aberrant error on both measurements, which means high measurements values during a brave time delay.

3. Nonlinear Transformation

Once the first transformation is done, it is possible to generate the residual value S' using a nonlinear transformation. The

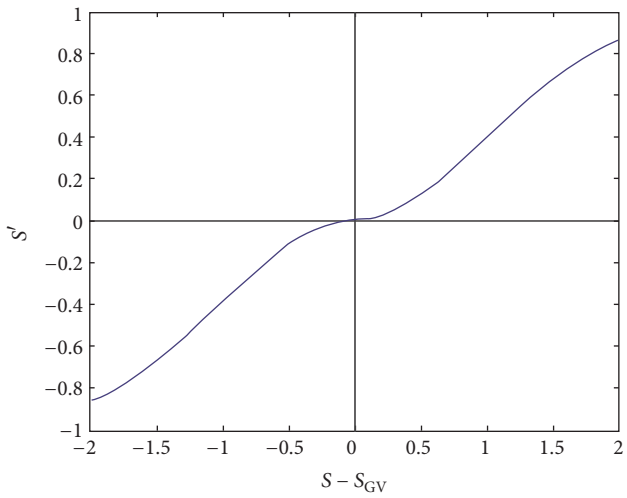


Figure 4: Nonlinear transformation (TNL) example.

attributes of a residual value are generally as described in [24], zero-centered during a nominal behavior, and presenting a noncentered distribution with the appearance of a fault on the corresponding measurement. In order to distinguish the faulty source efficiently, the residual has to be insensitive to fault appearance on other measurements. The chosen transformation (Figure 4) consists in a nonsymmetrical Gaussian transformation, centered on an estimation of the measurement global value, S_{GV}. The σ value permits the adjustment of the sensitivity of the transformation.

$$S' = 1 - \left[\left(\frac{1}{0.4 * (2 * pi)^{0.5}}\right) * \exp^{-0.5*((S-S_{GV})/\sigma)^2}\right] \quad \text{for } S \in [S_{GV}, \infty[,$$
$$S' = \left[\left(\frac{1}{0.4 * (2 * pi)^{0.5}}\right) * \exp^{-0.5*((S-S_{GV})/\sigma)^2}\right] - 1 \quad \text{for } S \in]-\infty, S_{GV}]. \tag{14}$$

In order to estimate the measurement global value, we need to develop an estimation method, which will use all the inputs measurements but has to be insensitive to fault. The chosen method consists in a weighted mean value (15), where each normalized weight W_i depends on the previous variation, as described in (16) to (19). The S_i value corresponds to the measurement for the sensor i.

$$S_{GV}(t) = \sum_{i=1}^{N} W_i(t) * S_i(t), \tag{15}$$

$$W_i = Wo_i * \frac{1}{\sum_{j=1}^{N} Wo_j}, \tag{16}$$

where N is the total number of measurements. The normalization is done to obtain a sum of weights equal to 1. The original weight W_o is calculated using a parameter r reflecting the past and the current deviation between the connected measurement and the estimated global value.

$$Wo_i = \frac{1}{r_i}, \tag{17}$$

$$r_i(t) = g * [r_i(t-1)] + (1-g) * \left[\sqrt{(S(t) - S_{GV}(t-1))^2}\right], \tag{18}$$

where g is a coefficient allowing giving more importance to the past values rather than the current deviation from the global value. Replacing $\sqrt{(S(t) - S_{GV}(t-1))^2}$ in the equation by the term ε, it is possible to generalize the calculation using the initial deviation r_{init}.

$$r_i(t) = g^{N+1} r_{init} + \sum_{i=0}^{N} \left[g^i (1-g) * \varepsilon(t-i)\right]. \tag{19}$$

This estimation method will be evaluated and compared to two others estimations in the simulation section. The residual generation will also be evaluated in Section 5.

4. Decision Process

The decision process is done by comparison with a threshold which has to be defined. Usually it is possible to optimally determine the threshold value by using statistical tools [25, 26], knowing information about signals characteristics, a priori probabilities, and so forth, but the currently studied case does not allow knowing all the information needed. Some alternative solutions such as the Neyman-Pearson criterion have been developed to deal with only one part of the information [27] but it still does not allow optimizing decisions considering several faults natures.

Also the sensitivity of the nonlinear transformation will need to be adjusted in order to optimize the detection. The quality of detection is usually determined using the false alarm and the missed detection rates, but other parameters can be used to evaluate the test. For instance, the maximum error during a missed detection or the cost depending on that value and the probability of appearance can also be used as quality criteria. In order to optimize the decision process, we need in the first time to run simulations representing nominal behavior. Then, failures have to be virtually added according to the description made in Section 2.1. The simulation process will be presented in the next section.

Using this database, the detection process will be done (described in Figure 5), varying both sensitivity and threshold in order to compare results for different cases (faults nature and importance, sensor affected...). Then using quality criteria, the parameters determination will be possible. The developed method to optimize the decision process consists in the choice of a first priority criterion. This criterion will be adjusted by the user. Then every configuration allowing encountering this criterion is determined and every other configuration is removed. The second stage consists of choosing a second criterion for which the optimal value will be found and so the optimal parameters set can be defined.

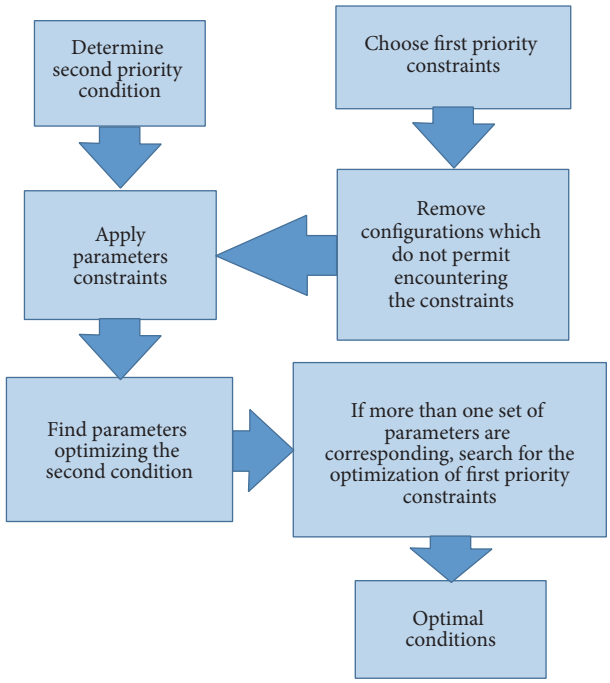


Figure 5: Decision process optimization strategy.

For example, it is possible to limit the false alarm rate to 5% as a first priority constraint and optimize the missed detection rate, and then we obtain the optimal parameter set for the chosen constraints. This method is so dependent of the chosen strategy, and different cases will be presented during the simulation section.

5. Simulations Tests, Evaluation, and Validation

5.1. Transformations Stages. Simulations have been run in two steps. First, vehicle and sensors nominal behaviors have been simulated using industrial version of pro-SiVIC simulation platform, allowing the generation of driving scenarios on different tracks, with speed and direction variations, to get feedback on the vehicle state taking into account its dynamic. The research version of pro-SiVIC has already been used in the development of different ADAS systems.

This software also models the sensors behavior. In order to validate the proposed method, it is primordial to ensure the failure detection function whatever the vehicle dynamic is, so the proposed scenario will present a complex trajectory with various dynamic cases, with speed changes both in curves and straight lines and also constant speed periods; all these driving conditions and dynamic states will allow having results representative of a classic driving scenario involving only one vehicle. As we focus on longitudinal acceleration and yaw rate, both of them are represented in the Figure 6 for the complete scenario.

As the method is working identically for both measurements, we will focus in the first time on the acceleration. First of all, we will evaluate the proposed global value estimation. The results will be compared to two other estimation

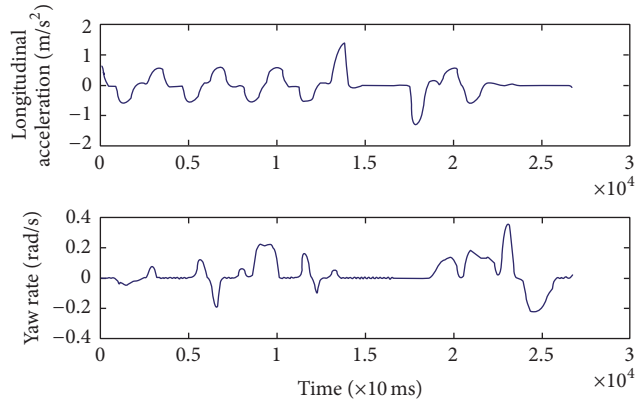


Figure 6: Longitudinal acceleration and yaw rate simulation profiles.

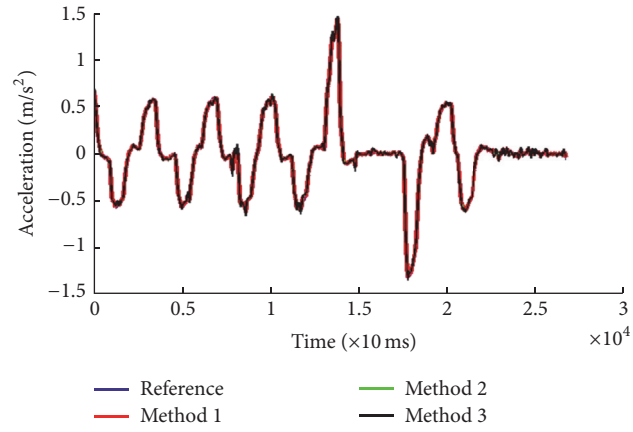


Figure 7: Nonfaulty global value estimation.

methods: a simple arithmetical mean value calculation (called method 1 in the results of Figure 7) and an estimation by Kalman filtering (method 2). The proposed approach will be referred to as method 3. First, the global value is directly estimated from the three (INS and both front and back odometers) measurements using the real acceleration as a reference (Figure 7).

The three estimation methods seem to work efficiently on a nominal case of study. The next step is to virtually add faults on measurements. Figure 8 represents estimations with one of the three measurements directly affected by a bias (left), a scale factor (center), and a punctual loss, represented by the measurement blocked at the previous registered value (right). The time interval affected by the default is highlighted in orange.

The estimation is visibly affected by injected errors, but this perturbation will depend on fault nature and value. Table 1 is presenting the mean quadratic error value during the exposure time for the three types of errors and for a set of both bias and gain fault values.

Except for a small scale factor, the proposed method (method 3) consisting in a weighted mean value calculation is always equal to or better than the two others studied method. Once the estimation method is validated, the nonlinear transformation will be evaluated.

Table 1: Mean quadratic error value for different fault characteristics depending on the estimation method.

Simulation parameters	Method 1	Method 2	Method 3
No faults	$3 * 10^{-4}$	$\mathbf{4 * 10^{-4}}$	$\mathbf{4 * 10^{-4}}$
Bias, 0.1 m/s^2 on one measurement	$1 * 10^{-3}$	$\mathbf{6 * 10^{-4}}$	$\mathbf{6 * 10^{-4}}$
Bias, 0.3 m/s^2	$1 * 10^{-2}$	$1 * 10^{-3}$	$\mathbf{8 * 10^{-4}}$
Bias, 0.5 m/s^2	$3 * 10^{-2}$	$2 * 10^{-3}$	$\mathbf{1 * 10^{-3}}$
Scale factor, 0.9	$\mathbf{0.9 * 10^{-3}}$	$1.02 * 10^{-3}$	$1.02 * 10^{-3}$
Scale factor, 0.7	$2.4 * 10^{-3}$	$1.1 * 10^{-3}$	$\mathbf{1.05 * 10^{-3}}$
Scale factor, 0.5	$5.2 * 10^{-3}$	$1.3 * 10^{-3}$	$\mathbf{1.2 * 10^{-3}}$
Stuck at the last recorded value	$2 * 10^{-2}$	$\mathbf{1 * 10^{-3}}$	$\mathbf{1 * 10^{-3}}$

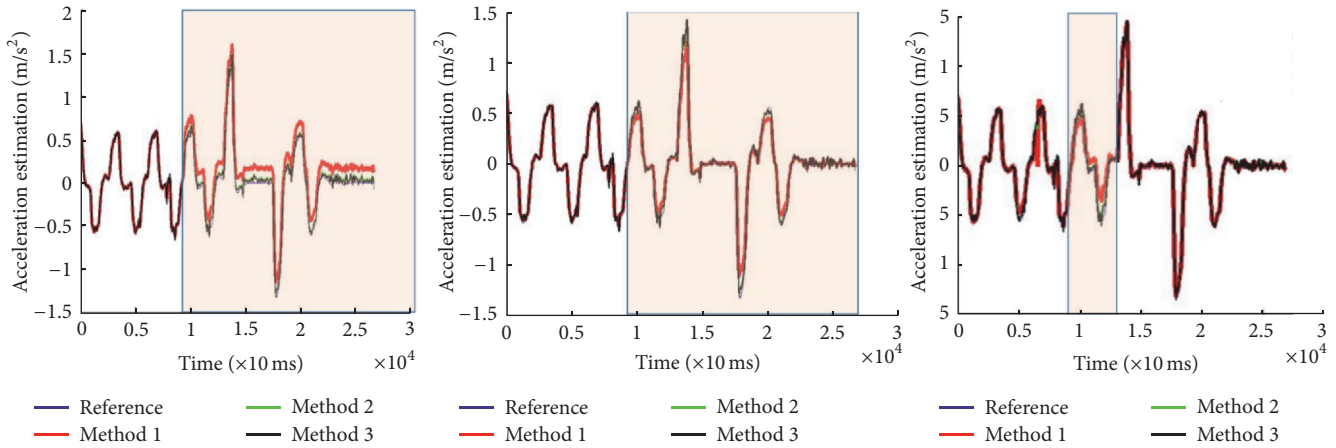


Figure 8: Impact of a fault on the global value estimation.

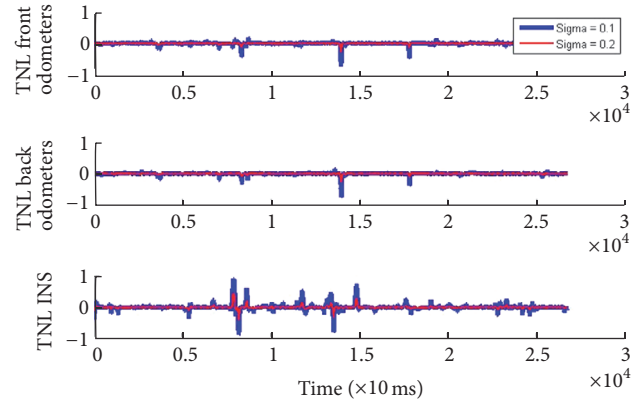


Figure 9: TNL results for different sensitivity values, without fault.

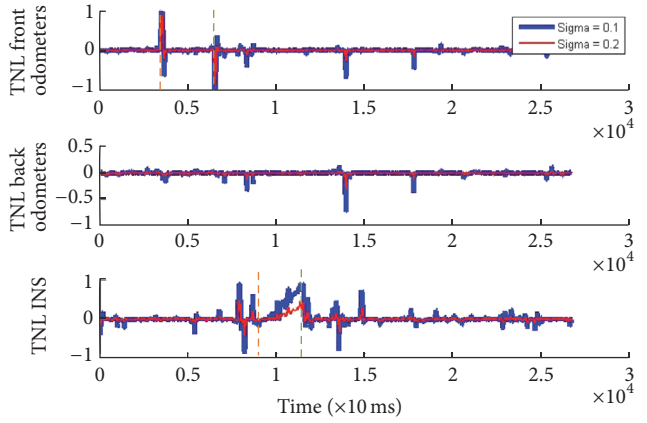


Figure 10: TNL result with the injection of a bias.

Using pro-SiVIC data presented earlier, we compute in the first time the nonlinear transformation on the three measurements as described in Section 2, in a nominal behavior, for two different σ values, 0.1 and 0.2. We are supposed to observe zero-centered signal for each measurement presenting a standard deviation depending on the noise level and the configured sensitivity.

Results obtained represent perfectly the expected behavior (Figure 9). The sigma value has to be set in order to make the method more robust, keeping in mind that it has to be small in order to detect the smallest fault values. All the faults models presented in Section 2 have been virtually added to measurements in order to observe their impacts on the TNL results and are presented in Figures 10–13. In all the figures, the appearance of a fault is represented by an orange line and the disappearance by a green line on the related sensor. When a bias is added on the odometers, it is added on the speed measurement and not directly on the acceleration estimation. The aberrant error is simulated here by the addition of an impulsion presenting an important value (at least ten times the current measurement value) during one sampling period when a total loss is simulated by a blocked value at zero.

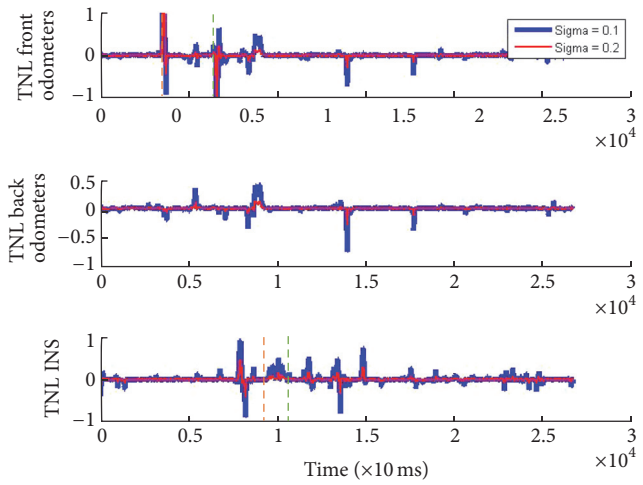


Figure 11: TNL result with the injection of a scale factor.

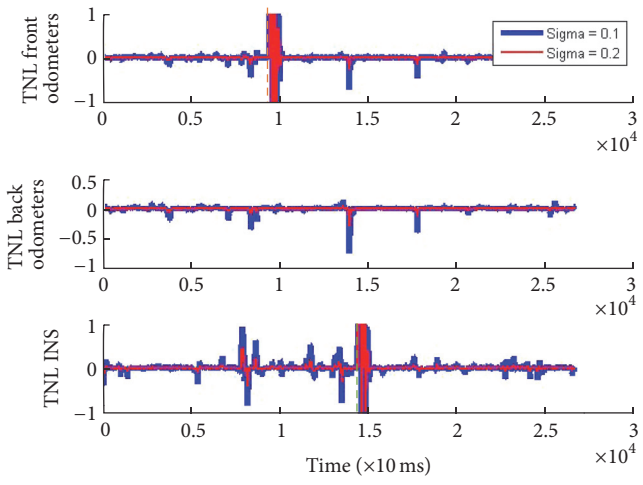


Figure 12: TNL result with the injection of aberrant errors.

Figures 10, 11, 12, and 13 present, respectively, results obtained with the addition of bias, scale factor, aberrant error, and total loss on measurements.

As expected for a bias or a scale factor on the odometric measurement, the acceleration measurement shows a punctual perturbation during the appearance (and disappearance) event but a nonprominent one for the duration of the disturbance. The yaw rate measurement will be more effective for this type of fault.

An aberrant error will generate a high intensity perturbation on the nonlinear transformation.

These simulations allow visual validation of the impact of each kind of fault on the nonlinear transformation, but it is also possible to generalize the results, considering a perturbation ΔS whatever the fault nature is, virtually added to the measurement under test.

$$S_F(t) = S_{NF}(t) + \Delta S, \tag{20}$$

where S_F and S_{NF} are, respectively, the faulty and nonfaulty measurements. Considering the same constant value as an input for all the three measurements with the addition of

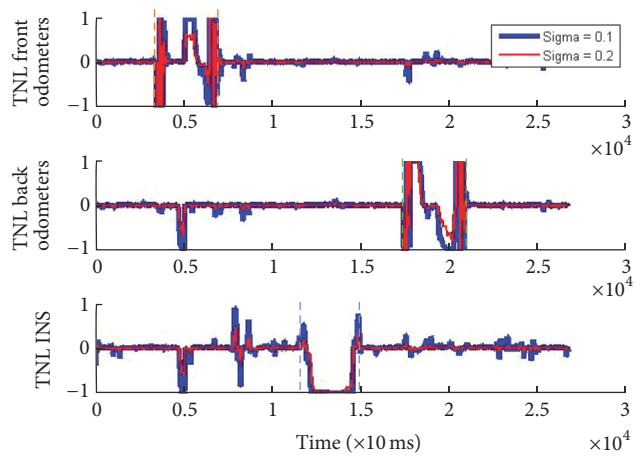


Figure 13: TNL result with the injection of a total loss.

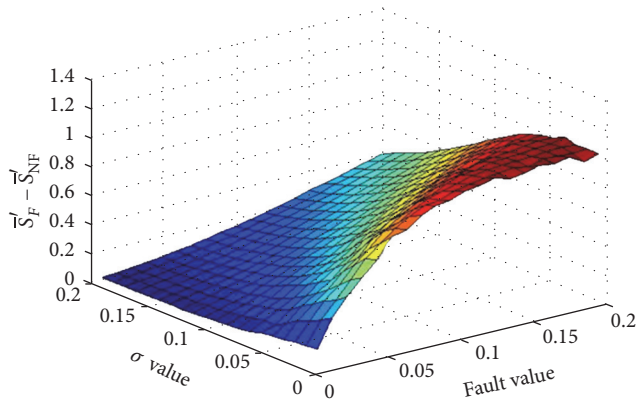


Figure 14: Output sensitivity according to the input sensitivity and the fault value.

noncorrelated stochastic processes on each of them in order to simulate measure noises, we compute in the first time the TNL, which will present the behavior observed in Figure 9. Then, the exact same conditions are used with the addition of a fault on one measurement. The simulation is repeated with different sensitivities and fault values. Then, we compute for each configuration the residual mean value $\overline{S}'$ and subtract the nonfaulty mean value from the faulty one ($\overline{S}'_F - \overline{S}'_{NF}$). The resulting value can be seen as the transformation output sensitivity to a fault according to the σ value. Figure 14 illustrates the obtained results according to fault and σ values. As expected, the highest output sensitivity is observed for the smallest σ values and the strongest faults.

Knowing that the proposed transformations are working efficiently, the focus is now put on the decision process optimization.

5.2. Decision Process. As depicted earlier, the decision is made by comparison between the residual and a threshold. The sensor is depicted as faulty if the absolute value of the corresponding residual is higher than the threshold. The sensor is still tested after the initial decision, but the corresponding measurement will not be used in the global value estimation as long as the residual is higher than the

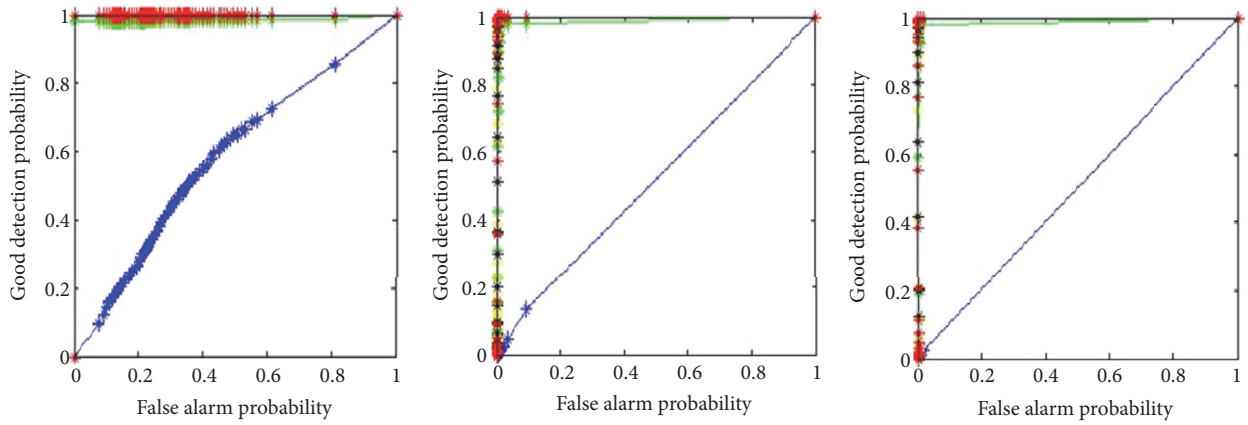


Figure 15: ROC curves for a bias on INS acceleration measurement, σ at 0.02 (left), 0.4 (middle), and 0.8 (right).

Table 2: Colors meaning on the ROC curves.

Color	Bias value (m/s^2)
Blue	0.02
Green	0.2
Yellow	0.4
Black	0.6
Red	0.8

threshold. A recovery is still possible as the sensor remains under test, and if its behavior leads its residual to decrease below the threshold value, the decision value C will return to the initial state 0.

Following this strategy, in order to optimize the decision process, it is required, in the first time, to run simulation using previous results for faulty and nonfaulty measurements and applying the decision, varying both TNL sensitivity and threshold values to establish ROC curves.

ROC curves (Receiver Operating Characteristic) allow evaluating hypothesis test quality by comparing good decisions (P_D) to false alarm (P_F) rates [28]. An example of representation is given in Figure 15 and Table 2 for a bias on the INS acceleration measurement. The different colors correspond to the injected fault value, to which will depend the decision results. As the sensitivity has also an important role to play in the decision process, the results are presented for three different σ values.

The simulation is done for every kind of fault, varying threshold, sensitivity, and fault values. In order to simplify, the example of acceleration measurement for the INS sensor is presented, but the proposed method can be easily applied on every sensor and measurement. As the acceleration measurement presents a large amount of values around 0, scale factor and total loss represented by a stuck at zero measurement will present a high missed detection rate, as the faulty measurement will present values similar to the real values. This kind of error does not present a high risk for the system function, as perturbed measurements will be the same as the real measurement value.

With the obtained information, the optimal parameter can be determined. The decision process optimization objectives are to determine the sensitivity and threshold values allowing obtaining the best performances according to the chosen criteria. As described in Section 4, a method has been developed to choose these two parameters. As the false alarm and missed detection (P_M, $P_M = 1 - P_D$) rates have been determined for each parameters couple, and also the maximal measurement error resulting from each missed detection, the optimization process can be realized. It is possible in the first time to choose characteristics allowing the lowest error rate, but considering the current situation, it is more important to limit the encountered maximum error due to a wrong decision. So different strategies results can be compared to evaluate the best solution.

> First chosen strategy is as follows: 1st priority, false alarm rate limited to 5%, 2nd priority, and maximum error minimization.

In order to realize this optimization strategy, the determination of the maximum error is needed. So, for every wrong decision, the error absolute value between the faulty measurement and the real measurement has to be determined (21); then the maximum error value is connected to each configuration and then the optimization method can be applied.

$$\mathrm{Er}(t) = \sqrt{S_i(t) - S_{\mathrm{real}}(t)}. \tag{21}$$

As the simulation is realized for every kind of fault, the error will be determined for each fault nature. As only one set of parameters has to be configured, the error corresponding to each couple sensitivity/threshold can be assimilated as the mean maximum error value.

$$\mathrm{ErG} = \frac{\mathrm{Er}_{\mathrm{Bias}} + \mathrm{Er}_{\mathrm{SFactor}} + \mathrm{Er}_{\mathrm{TLoss}}}{3}. \tag{22}$$

Using these two criterion, the results mentioned in Tables 3 and 4 are obtained, taking into account every nature of fault with the same occurrence probability.

Table 3: Obtained results for the first chosen strategy.

Maximum false alarm rate specified	0.05
Minimum error value in the remaining parameters set	0.0928
Threshold	*0.47*
Sensitivity	*0.06*
Corresponding false alarm rate	0.0496
Corresponding missed detection rate	0.551

Table 4: Obtained results for the second chosen strategy.

Maximum error accepted	0.05
Minimum false alarm rate in the remaining parameters set	0.169
Threshold	*0.78*
Sensitivity	*0.02*
Corresponding maximum observed error	0.0494
Corresponding missed detection rate	0.5121

The missed detection rate is high, as predicted, but the maximal error encountered is lower than 0.1 m/s^2.

> Second strategy is as follows: 1st priority, Maximum acceptable error set at 0.05 m/s^2, 2nd priority, false error rate minimization.

This second strategy takes into account the same parameters but changing the priorities. In a system where we want the smallest error possible between the measurement and the real value, this will probably be the best alternative.

Here the false alarm rate is higher (around 17%) meaning sensors will frequently be isolated. In a system where measurements can be analytically generated with the help of other sensors, the temporary isolation does not present an important inconvenience. This remains a strategical choice to be made by the user, depending on wanted characteristics.

6. Conclusion

In order to deal with strong system nonlinearity and environmental perturbations, a sensors fault detection algorithm has been developed, using analytical redundancy and nonlinear transformation. After the concept and architecture presentation, the residual generation consisting of the comparison of a common measurement nonlinear transformation (TNL) has been depicted, during which a global value estimation has been proposed in order to remove the measurement common part. A strategy to make the decision process optimization has then been discussed.

The different part of the proposed algorithm has then been evaluated through different simulations, using data from pro-SiVIC software, allowing the simulation of a vehicle dynamic behavior and the embedded sensor responses. In the first time, the global value estimation has been compared to two other estimation methods and shows better results than them for almost all the configuration tested. Then the behavior of the nonlinear transformation has been studied, according to different fault nature and sensitivity configuration. Then, as this parameter and the threshold value have to be determined in order to ensure a quality decision, the proposed method for the parameter choices has been evaluated. In this last part, two different strategies have been compared, according to the user preferences, allowing limiting or reducing test results characteristics.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is part of CooPerCom, a 3-year international research project (Canada-France). The authors would like to thank the National Science and Engineering Research Council (NSERC) of Canada and the Agence Nationale de la Recherche (ANR) in France for supporting the Project STP 397739-10. This work is done in the LIV research group (Laboratory of Intelligent Vehicles), in the Electrical and Computer Engineering Department of Sherbrooke University, (http://www.gel.usherbrooke.ca/LIV/public/). The authors want to thank the other researchers of the LIV for their cooperation.

References

[1] C. Bai-Gen, W. Jian, Y. Qin, and L. Jiang, "A GNSS based slide and slip detection method for train positioning," in *Proceedings of the Asia-Pacific Conference on Information Processing (APCIP '09)*, pp. 450–453, July 2009.

[2] S.-B. Kim, J.-C. Bazin, H.-K. Lee, K.-H. Choi, and S.-Y. Park, "Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data," *IET Radar, Sonar & Navigation*, vol. 5, no. 8, pp. 814–823, 2011.

[3] A. Bak, D. Gruyer, S. Bouchafa, and D. Aubert, "Multi-sensor localization—visual odometry as a low cost proprioceptive sensor," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC '12)*, pp. 1365–1370, IEEE, Anchorage, Alaska, USA, September 2012.

[4] R. J. Patton, J. Chen, and H. Y. Zhang, "Modelling methods for improving robustness in fault diagnosis of jet engine system," in *Proceedings of the 31st IEEE Conference on Decision and Control*, vol. 2, pp. 2330–2335, Tucson, Ariz, USA, December 1992.

[5] J. Chen and R. J. Patton, "Robust fault diagnosis of stochastic systems with unknown disturbances," in *Proceedings of the International Conference on (CONTROL '94). Part 2 (of 2)*, pp. 1340–1345, March 1994.

[6] R. J. Patton, J. Chen, and G. P. Liu, "Robust fault detection of dynamic systems via genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '95)*, pp. 511–516, Sheffield, UK, September 1995.

[7] M. Saif and Y. Guan, "A new approach to robust fault detection and identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 685–695, 1993.

[8] Y. Zhongming and W. Bin, "A review on induction motor online fault diagnosis," in *Proceedings of the 3rd International Power Electronics and Motion Control Conference (IPEMC '00)*, vol. 3, pp. 1353–1358, Beijing, China, 2000.

[9] G. Heredia, B. V. Remu, and A. Ollero, "Actuator fault detection in autonomous helicopters," in *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '04)*, pp. 1–6, Lisbon, Portugal, 2004.

[10] G. Heredia, F. Caballero, I. Maza, L. Merino, A. Viguria, and A. Ollero, "Multi-Unmanned Aerial Vehicle (UAV) cooperative fault detection employing Differential Global Positioning (DGPS), inertial and vision sensors," *Sensors*, vol. 9, no. 9, pp. 7566–7579, 2009.

[11] G. Heredia, A. Ollero, M. Bejar, and R. Mahtani, "Sensor and actuator fault detection in small autonomous helicopters," *Mechatronics*, vol. 18, no. 2, pp. 90–99, 2008.

[12] G. Heredia and A. Ollero, "Detection of sensor faults in small helicopter UAVs using Observer/Kalman filter identification," *Mathematical Problems in Engineering*, vol. 2011, Article ID 174618, 20 pages, 2011.

[13] G. Heredia, A. Ollero, R. Mahtani, M. Béjar, V. Remuss, and M. Musial, "Detection of sensor faults in autonomous helicopters," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 2229–2234, IEEE, Barcelona, Spain, April 2005.

[14] G. Heredia and A. Ollero, "Sensor fault detection in small autonomous helicopters using observer/Kalman filter identification," in *Proceedings of the IEEE International Conference on Mechatronics (ICM '09)*, IEEE, Malaga, Spain, April 2009.

[15] R. J. Patton, P. Frank, and R. Clark, *Fault Diagnosis in Dynamic Systems, Theory and Application*, Prentice Hall, 1989.

[16] R. J. Patton and J. Chen, "Detection of faulty sensors in aero jet engine systems using robust model-based methods," in *Proceedings of the IEE Colloquium on Condition Monitoring for Fault Diagnosis*, pp. 2/1–2/22, IET, October 1991.

[17] S. Simani, C. Fantuzzi, and R. J. Patton, *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, Springer, 2000.

[18] S. I. Roumeliotis, G. Sukhatme, and A. Bekey George, "Fault detection and identification in a mobile robot using multiple-model estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2223–2228, Leuven, Belgium, May 1998.

[19] S. I. Roumeliotis, G. Sukhatme, and G. A. Bekey, "Sensor fault detection and identification in a mobile robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1383–1388, Victoria, Canada, October 1998.

[20] H. Sun and M. E. Cannon, *Relibaility Analysis of an ITS Navigation System*, Department of Geomatics Engineering; The University of Calgary, Calgary, Canada, 1998.

[21] S. Anwar and L. Chen, "An analytical redundancy-based fault detection and isolation algorithm for a road-wheel control subsystem in a steer-by-wire system," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2859–2869, 2007.

[22] W. Huang and X. Su, "Design of a fault detection and isolation system for intelligent vehicle navigation system," *International Journal of Navigation and Observation*, vol. 2015, Article ID 279086, 19 pages, 2015.

[23] X. Qi, D. Theilliol, J. Qi, Y. Zhang, and J. Han, "A literature review on fault diagnosis methods for manned and unmanned helicopters," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS '13)*, pp. 1114–1118, IEEE, Atlanta, Ga, USA, May 2013.

[24] R. J. Patton and J. Chen, "Robust fault detection using eigenstructure assignment: a tutorial consideration and some new results," in *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 3, pp. 2242–2247, December 1991.

[25] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part 1*, Wiley, New York, NY, USA, 1968.

[26] M. Barkat, *Signal Detection and Estimation*, Artech House, 2nd edition, 2005.

[27] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer, Berlin, Germany, 1988.

[28] Y. Cai, M.-Y. Chow, W. Lu, and L. Li, "Evaluation of distribution fault diagnosis algorithms using ROC curves," in *Proceedings of the IEEE PES General Meeting (PES '10)*, Providence, RI, USA, July 2010.

Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 8132528, 7 pages
https://doi.org/10.1155/2017/8132528

Research Article

Research on Fault Diagnosis Method Based on Rule Base Neural Network

Zheng Ni,[1] Zhang Lin,[1] Wang Wenfeng,[1,2,3] Zhang Bo,[1] Liu Yongjin,[1] and Zhang Dajiang[1]

[1]*Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China*
[2]*The 302ed Institute of the Tenth Academy of China Aerospace Science & Industry Corporation, Guiyang, Guizhou 550002, China*
[3]*School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Wang Wenfeng; wwf981612@163.com

Received 10 June 2016; Revised 15 September 2016; Accepted 16 October 2016; Published 12 January 2017

Academic Editor: Xiao He

The relationship between fault phenomenon and fault cause is always nonlinear, which influences the accuracy of fault location. And neural network is effective in dealing with nonlinear problem. In order to improve the efficiency of uncertain fault diagnosis based on neural network, a neural network fault diagnosis method based on rule base is put forward. At first, the structure of BP neural network is built and the learning rule is given. Then, the rule base is built by fuzzy theory. An improved fuzzy neural construction model is designed, in which the calculated methods of node function and membership function are also given. Simulation results confirm the effectiveness of this method.

1. Introduction

In recent years, with the increasing development of science technology and automation, weapon equipment systems have been updated constantly. The fault occurrence probability is higher with the development tendency of large scale, complexity, and nonlinearity. The complexity of weapon system and the nonlinearity between fault cause and fault phenomenon reduce the diagnosis sensitivity of weapon system and improve the error report ratio, which increase the difficulty of fault diagnosis [1–4].

The BP neural network is the most extensive used neural network model. It not only has strong self-learning ability and self-organization, but also can process nonlinear problem. So BP neural network has a great superiority in processing nonlinear fault. Reference [5] uses the method of [6] to improve the BP algorithm of three-layer perceptron and apply it to the fault diagnosis of steam turbine, which optimizes the network step by step and accelerates the convergence speed of the net by this method. Reference [7] applies BP neural network into the fault diagnosis of gear in vehicle transmissions. Reference [8] constructs the fault diagnosis method of solid rocket motor based on fuzzy neural network, which combines BP network and fuzzy inference. Reference [9] synthesizes rule base, Bayesian belief network, and neural network into integrated software while fault prognosis. Reference [10] promotes an intelligent fault diagnosis expert system based on rule base, which merges together with fuzzy theory and enhances the transparence of fault diagnosis.

While ensuring learning samples, traditional rule base relies on expert experience and knowledge mainly and manifests by IF-THEN formation. A formal produced rule can manifest the fault whose causality is distinct, but it cannot manifest the fault whose causality is not distinct and relationship between phenomenon and reason is nonlinear. Fuzzy theory has the ability to process unsure and inexact information well. So this paper combines fuzzy theory with neural network, applies fuzzy logic into the description of high-rise logic frame, and uses neural network to process data, which enhances the accuracy of fault ruling base and acquires exact training samples. The efficiency of fault diagnosis is improved by applying the method to fault diagnosis.

2. Problem Description

On the basis of known data and fault model, the neural network can acquire mapping relationship between data and fault model by the studying of samples. Neural network can

acquire complete and precise message from the incomplete and unsafe ones and realize the analogue to human thinking model in aspect of signal processing and learning ability [9].

In the reality procedure, mean square error can be approximated by $G(x)$:

$$G(x) = (t-a)^T (t-a). \tag{1}$$

Among these, x is weighted value W or biasing b, t is target output, and a is realistic output.

The essence of the neural network training process is an adjusting process of weights W and bias b constantly. In order to guarantee the convergence of nervous system, the steepest method is usually used to adjust the weights W and bias b:

$$\begin{aligned} W^m(k+1) &= W^m(k) - \alpha(k) \times \frac{\partial G(x)}{\partial W^m} \\ &= W^m(k) - \alpha(k) \times \frac{\partial G(x)}{\partial n^m} \times \left(a^{m-1}\right)^T, \\ b^m(k+1) &= b^m(k) - \alpha(k) \times \frac{\partial G(x)}{\partial b^m} \\ &= b^m(k) - \alpha(k) \times \frac{\partial G(x)}{\partial n^m}, \end{aligned} \tag{2}$$

where k is the number of iterations and α is velocity of learning.

$G(x)$ is minimum or its value is small to a certain value ε through continuous recursion and iteration. Due to the existence of the iterative process, the influence of original samples uncertainty to the final result of the fault diagnosis is enhanced. It also leads to the following problems in the process of application of neural network:

(1) The difficulty of acquiring samples: neural network needs a lot of samples training before fault diagnosis, but the presence of sample cannot be ensured and the sample distribution whether equal or not cannot be judged because of the random. It can hardly guarantee the training effect and the accuracy. At the same time, in actual engineering equipment, abundant and distributed equally fault samples are difficult to acquire [11].

(2) The underutilization of experienced knowledge: in the territory of fault diagnosis, the using of experienced knowledge is important. Some faults can be found the fault reasons by experienced knowledge. But neural network acquires experienced knowledge from learning samples and this experienced knowledge has great boundedness. So, the learning method of neural network reduces the accuracy of fault diagnosis.

We can see that the accuracy of fault diagnosis has an intimate relationship with the quality of training sample. Under the condition of unsure message, the paper applies fuzzy theory into neural network to deal with the related fault better and acquire more typical samples, which makes advantages of known experienced knowledge to realize a precise location. Fuzzy theory is used to process unsure message to acquire a more objective fault rule base. Then the improved neural network is used to diagnose fault. The accuracy of fault diagnosis is improved.

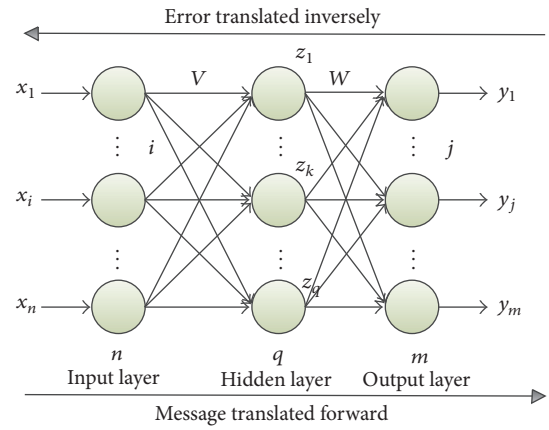


Figure 1: The structure of BP neural network.

3. The Design of BP Neural Network Fault Diagnosis Model Based on Rule-Learning

The general steps of fault diagnosis based on neural network include confirming network structure, then having learning training to network based on known fault phenomenon and reason, adjusting weight and threshold, and finally acquiring result of diagnosis using trained network and input fault phenomenon. So, the fault diagnosis method based on neural network can be divided into two main parts: BP neural network training and BP neural network fault diagnosis.

In order to improve the accuracy of fault diagnosis based on neural network, the belief rule base which is constructed by the fuzzy theory in the basis of existent experience knowledge is used to preprocess original samples before the neural network training. The processed samples are used to train the neural network. Finally, the trained neural network can be used to diagnose fault and obtain the fault diagnosis results. So the key to improve the neural network fault diagnosis is how to construct belief rule based on the fuzzy theory.

3.1. Structure of Classical Neural Network. From the angle of mapping, the procedure of diagnosis can be seen as a mapping from fault eigenvectors set to fault set. So the fault diagnosis is to find the mapping relationship between fault sign and reason. The neural network is to construct the fault mapping relationship through neural network structure. Continuous function of random closed interval can be approximated by three-layer BP neural networks, which include input layer, hidden layer, and output layer. The hidden layer can contain one or more layers [12]. The classical BP neural network's structure is shown in Figure 1.

V and W are linking weighted matrices. Input layer mainly takes charge of receiving various fault messages, and every unit manifests a fault characteristic parameter. The

main functions of hidden layer are learning the fault message of input layer and saving threshold and linking weighted value which make network approximation. Output layer mainly exports the result of diagnosis, which uses Sigmoid or hard-approximation function usually and the number of neurons is decided by the kinds of faults [13].

3.2. Establishment of the Belief Rule Base. Suppose $U = \{U_i;\ i = 1, 2, \ldots, T\}$ is the set of rule premise attribute, and $A_i = \{A_{ij};\ j = 1, 2, \ldots, J_i = |A_i|\}$ $(i = 1, 2, \ldots, T)$ is the reference value set of the premise attribute U_i; then, the form of belief rule is as follows.

If U_1 is A_{1n_1} and U_2 is A_{2n_2} and so on and is A_{Tn_T}, then

$$U_T \cdots \left\{\left(D_1, \overline{\beta_{1k}}\right)\left(D_2, \overline{\beta_{2k}}\right), \ldots, \left(D_N, \overline{\beta_{Nk}}\right)\right\}, \quad (k = 1, 2, \ldots, L), \tag{3}$$

where $A_{in_i} \in A_i$ $(n_i = 1, 2, \ldots, J_i)$ is the reference value i, $\overline{\beta_{ik}}$ $(i = 1, 2, \ldots, N)$ is the result, the degree of confidence belonging to the output reference value D_i meets $\sum_{i=1}^{N} \overline{\beta_{ik}} \le 1$, and L is the number of rules in rule base.

The activated weight of rule K is

$$\omega_k = \frac{\theta_k \alpha_k}{\sum_{i=1}^{L} \theta_i \alpha_i},$$
$$\alpha_k = \prod_{i=1}^{T_k} \left(\alpha_k^i\right)^{\overline{\delta_{ki}}}, \tag{4}$$
$$\overline{\delta_{ki}} = \frac{\delta_{ki}}{\max_{i=1,2,\ldots,T_k}\{\delta_{ki}\}},$$

where θ_k is the relative weight of rule k, α_k^i is the match degree of input i to reference value set A_i in rule k, δ_{ki} is the relative weight of i in rule k, a_i is the overall compatibility of input variable relative to the premise attribute U in rule i, and θ_i is the weight of rule i.

If $\omega_k > 0$, the rule k is activated, otherwise not activated.

Because the belief degree, weight, promise attribute, and reference value are given by initial expert knowledge, the error is easy to be produced. So the generation processes of the rule base need to be optimized. The processes are as follows.

Step 1. Determine the rule belief degree, weight, premise attribute, and output result based on initial expert knowledge, and establish the initial belief rule base.

Step 2. Convert the input data into distributed form.

Step 3. Forward inference to get the simulation output based on the belief rule base and input data, and then calculate the error.

Step 4. If the error is within the allowable range, end the training optimization. Otherwise, turn to Step 5.

Step 5. According to the training data, use the FMINCON function in the MATLAB to achieve the optimization of the training parameters, save the optimization results, update the belief rule base, and then turn to Step 3.

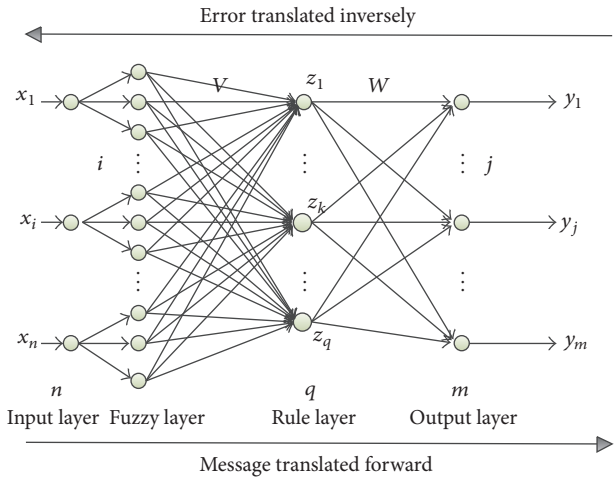


Figure 2: The structure of fuzzy neural network.

3.3. Structure of Improved Fuzzy Neural Network. In order to take advantage of existing experienced knowledge, we use fuzzy logic to reform neural network. The construct fuzzy neural network is constructed according to expert language rule. A 4-layer fuzzy neural network is formed by input layer, fuzzy layer, rule layer, and output layer. The structure is shown in Figure 2.

The description of every node's function is as follows.

(1) *Input Layer*. Input variable is original fault sign one, and the number of nodes is the number of sign variable.

(2) *Fuzzy Layer*. The number of nodes is the sum of all fuzzy intervals of sign variable. The number of fuzzy logic and initial membership functions is defined according to the distribution condition of every fuzzy variable and the significance of each fuzzy variable to system. Then the normalized difference variable is translated to many fuzzy sets. Then each neuron of every layer is correspondence to a fuzzy set. Its output manifests the membership to this fuzzy set.

(3) *Rule Layer*. Every node corresponds to a fuzzy control rule, the prerequisite of completing a fuzzy logic rule is matching operation, and the input is membership and the output is matching degree.

(4) *Output Layer*. According to the sum of value of fuzzy logic value and linking weight, the output is acquired.

There are three kinds of adjustable variables: one is the weighted coefficient of third and fourth layer, which manifests rule parameter; the second and third are mean value and standard deviation of the function.

3.4. Confirmation of Improved Neutral Network Training Parameters. The training and learning of neural network are mainly adjustment of interlayer linking weighted value and the core and width of affiliate function.

Table 1: Training samples table.

Series number	Model	$\overline{E}_{30}$	$\overline{E}_{31}$	$\overline{E}_{32}$	$\overline{E}_{33}$	$\overline{E}_{34}$	$\overline{E}_{35}$	$\overline{E}_{36}$	$\overline{E}_{37}$	$\widehat{g}_3$	$\widehat{g}_4$
1	Imbalance	0.0509	0.522	0.450	0.538	0.036	0.505	0.169	0.152	0.231	0.572
2		0.727	0.643	0.639	0.780	0.057	0.602	0.113	0.021	0.154	0.263
3		0.318	0.358	0.278	0.415	0.028	0.334	0.084	0.245	0.125	0.241
4		0.857	0.533	0.538	0.757	0.066	0.549	0.100	0.364	0.265	0.452
5	Misaligned	0.293	0.185	0.147	0.155	0.018	0.182	0.380	0.285	0.341	0.214
6		0.533	0.520	0.370	0.487	0.052	0.548	0.924	0.214	0.741	0.285
7		0.166	0.171	0.131	0.172	0.020	0.189	0.298	0.623	0.321	0.241
8		0.467	0.247	0.143	0.272	0.042	0.277	0.653	0.214	0.854	0.365
9	Looseness	0.304	0.058	0.051	0.136	0.062	0.156	0.072	0.247	0.280	0.157
10		0.617	0.124	0.197	0.257	0.107	0.222	0.090	0.352	0.264	0.514
11		0.302	0.042	0.077	0.114	0.058	0.110	0.046	0.452	0.153	0.214
12		0.667	0.075	0.127	0.214	0.106	0.161	0.059	0.632	0.352	0.241
13	Rub-impact	0.108	0.286	0.245	0.161	0.053	0.211	0.020	0.214	0.152	0.741
14		0.302	0.806	0.595	0.614	0.095	0.732	0.102	0.145	0.215	0.541
15		0.120	0.428	0.246	0.352	0.045	0.402	0.063	0.362	0.384	0.562
16		0.297	0.001	0.023	0.076	0.053	0.057	0.026	0.264	0.274	0.247

Table 2: Testing samples.

Serial number	$\overline{E}_{30}$	$\overline{E}_{31}$	$\overline{E}_{32}$	$\overline{E}_{33}$	$\overline{E}_{34}$	$\overline{E}_{35}$	$\overline{E}_{36}$	$\overline{E}_{37}$	$\widehat{g}_3$	$\widehat{g}_4$
1	1.181	0.821	0.708	0.881	0.068	0.955	0.228	0.245	0.365	0.241
2	0.9	0.461	0.357	0.341	0.040	0.44	1.055	0.258	0.241	0.854
3	0.798	0.238	0.209	0.364	0.188	0.382	0.159	0.218	0.368	0.247
4	0.309	0.547	0.465	0.292	0.107	0.401	0.026	0.285	0.421	0.548

Table 3: Output result.

Serial number	d_1	d_2	d_3	d_4	Expected output
1	1.0021	0.0035	0.0106	0.0202	1000
2	0.0180	1.0142	0.0240	0.0012	0100
3	0.0028	0.0032	1.0421	0.0507	0010
4	0.0109	0.0207	0.0253	1.0437	0001

(1) *Linking Weighted Value*. Between fuzzy layer and rule layer, the value is 1 while linking and 0 while not linking. We only train the linking weighted value between rule layer and output layer.

(2) *Confirmation of Affiliate Function*. The input data need to be normalized difference because of the characteristic of neural network and the mainly transform function. While BP network is applied into solving problem, the condition of convergence slow or little always appears. A good way to normalize difference can have an improvement on the property of network. In this paper, fuzzy membership normalized difference is used. Because the relationship between fault and sign is fuzzy, whether fault exists does not have distinct bounds and the fault obeys the distribution of some membership function. In a word, the kind of membership we choose is as follows: the value of membership function increases in some method with the increase of independent variable. The expression is

$$\mu(x) = \frac{kx^2}{1 + kx^2}. \tag{5}$$

And ensured self-Cauchy membership function satisfies the above demand.

(3) *Learning Velocity (Step-Size)* l_r. Set the initial value as 0.01, and it can be satisfied according to the following formula:

$$l_r = \begin{cases} 1.05 l_r, & E(k) < E(k-1), \\ 0.7 l_r, & E(k) > E(k-1) \cdot 1.04, \\ l_r, & \text{others}. \end{cases} \tag{6}$$

(4) *Momentum Factor mc*. Set the initial value as 0.95, and it can be satisfied according to the following formula:

$$l_r = \begin{cases} 0, & E(k) < E(k-1), \\ 0.95, & E(k) > E(k-1) \cdot 1.04, \\ \text{mc}, & \text{others}. \end{cases} \tag{7}$$

(5) *Error*. It is a very small number and can be ascertained according to specific condition.

Table 4: Two-mixed fault training.

Serial number	$\overline{E}_{30}$	$\overline{E}_{31}$	$\overline{E}_{32}$	$\overline{E}_{33}$	$\overline{E}_{34}$	$\overline{E}_{35}$	$\overline{E}_{36}$	$\overline{E}_{37}$	$\hat{g}_3$	$\hat{g}_4$
1	0.0809	0.622	0.550	0.468	0.236	0.305	0.469	0.352	0.431	0.672
2	0.127	0.543	0.439	0.560	0.347	0.302	0.513	0.421	0.514	0.663
5	0.523	0.432	0.642	0.334	0.676	0.184	0.295	0.693	0.073	0.114
6	0.518	0.398	0.579	0.299	0.702	0.165	0.249	0.703	0.081	0.125

Table 5: Training result.

Serial number	d_1	d_2	d_3	d_4	Expected output	Model
1	1.0034	0.9924	0.0305	0.0043	1100	Imbalance and misaligned fault
2	1.0200	1.0324	0.0130	0.0031	1100	
3	0.0028	0.0032	1.0421	0.9507	0011	Looseness and rub-impact fault
4	0.0098	0.0110	1.0136	0.9473	0011	

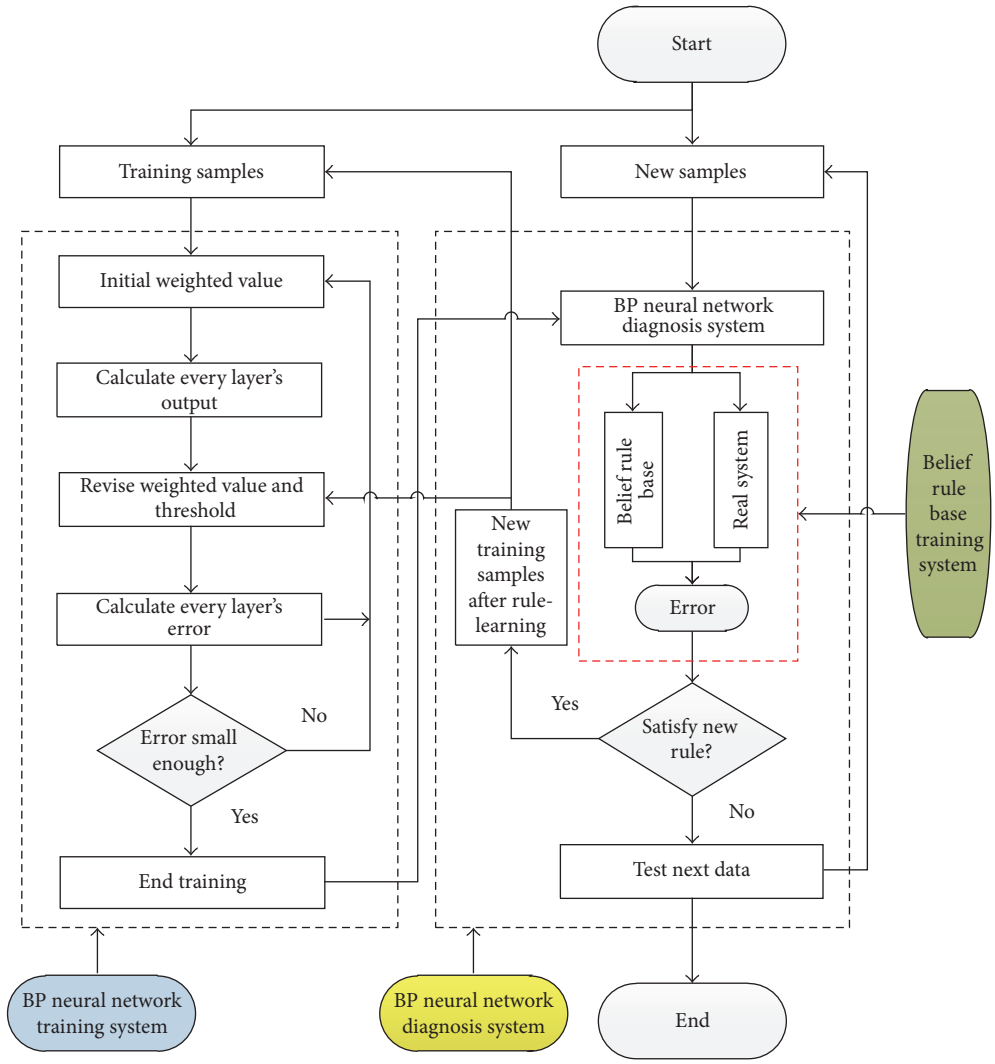


Figure 3: Procedure of rule-learning base BP neural network fault diagnosis.

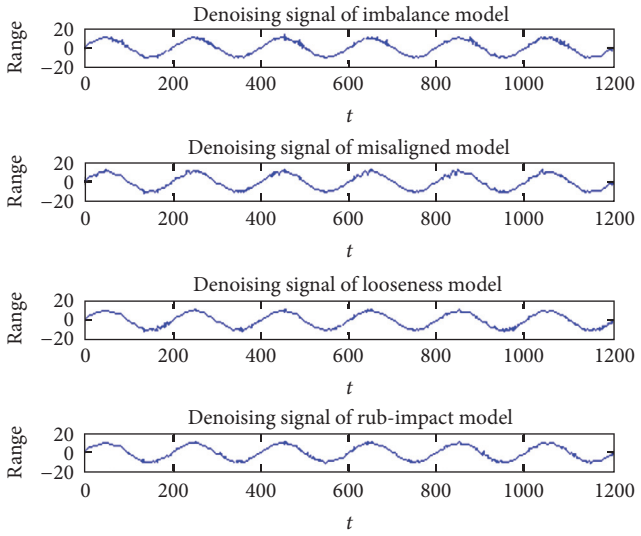


Figure 4: Denoise effect on figure of sampling signal.

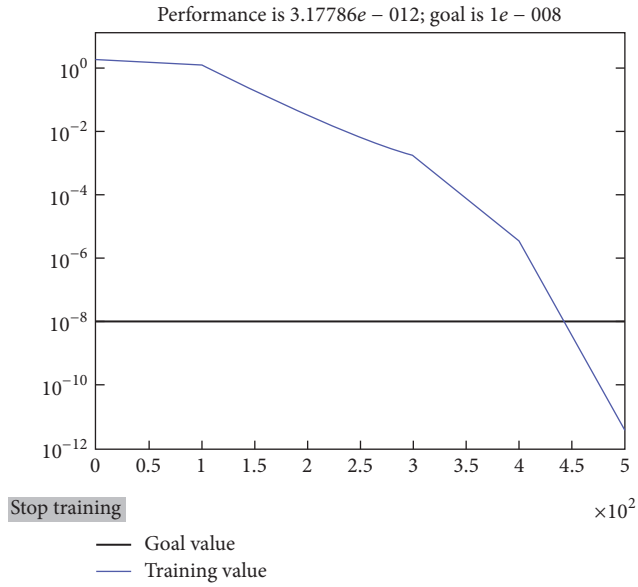


Figure 5: Procedure of training.

(6) *Maximum Number of Network's Training Processes.* It is one of the end conditions of the network training. Stop it when it does not converge until maximum number, which manifests that the property of convergence is bad or the maximum number of training processes is too small.

3.5. The Procedure of BP Neural Network Fault Diagnosis Based on Rule-Learning

Step 1. Classify samples into training samples and test (new) samples.

Step 2. Construct initial fuzzy neural network training system based on Section 3.4 and form improved fuzzy neural network diagnosis system after several training processes.

Step 3. Diagnosis is done by the test samples of trained BP neural network diagnosis system, and judge whether the fault is a new fault or not. A new sample is formed if it is a new fault. We train the improved BP neural network and realize the function of self-learning.

4. Simulation and Test

4.1. Description of Problem. We put rolling bearing as the fault diagnosis target and draw its vibration signal. The inputs are the 8 frequencies of time and frequency territory's power and sharpness, which manifest by S_1–S_{10}. The 10 characteristics are condition attributes $C = \{S_1, S_2, \ldots, S_{10}\}$. The rolling bearing has five states: formal, imbalance, misalignment, rub-impact, and looseness, which are manifested by $(0, 0, 0, 0)$, $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, and $(0, 0, 0, 1)$.

4.2. Sole Fault Simulation. It is done through the experiment bench to sample and denoise the signal as shown in Figure 4.

The training samples are shown in Table 1 after analysing and drawing.

The BP neural network has 10 input nodes, 4 output nodes, and 21 hidden nodes. It uses the learning algorithm of Section 3.4. The maximum training processes number is 1000. The precision of training is $1e - 8$ and learning efficiency is 0.01. The 16 groups of Table 1 are used to train. The procedure is shown in Figure 5.

We can see from Figure 5 that the error request after 450 steps is good and the convergence effect is very good.

In order to test the effect of BP neural network fault diagnosis system after 16 samples training, we acquire 4 test samples in Table 2. The test results are shown in Table 3.

We can conclude from Table 3 that the outputs of 4 test samples coincide with expected value. The accuracy of model is tested.

4.3. Mixed Fault Simulation. The mixed fault is common in the real fault. We set two manual mixed faults as in Table 4.

The samples of Table 4 cannot be distinguished by the BP neural network fault diagnosis system trained by Table 1. But it can acquire the result of Table 5 by the belief rule base of Figure 3.

We train the BP neural network again by using the data of Table 5 as a new training sample. Comparing the expected output and the actual output in Table 5, it can be found that the neural network trained by belief rule base has the diagnosis ability to these two kinds of mixed fault.

5. Conclusion

Because of the distribution of training sample, the veracity of training result of neural network is hard to ensure by only using neural network to diagnose fault. A neural network fault diagnosis based on rule base is promoted, which transfers expert's experience to rule by fuzzy process and then applies to neural network. This method ensures that

the training sample can respond and the veracity of neural network fault diagnosis is improved.

Competing Interests

There are no competing interests related to this paper.

References

[1] D.-H. Zhou and Y.-Y. Hu, "Fault diagnosis techniques for dynamic systems," *Acta Automatica Sinica*, vol. 35, no. 6, pp. 748–758, 2009.

[2] S. X. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms and Tools*, Springer, London, UK, 2nd edition, 2013.

[3] A. Haghani Abandan Sari, *Data-driven design of fault diagnosis systems*, Springer, Wiesbaden, Germany, 2014.

[4] L. Qin, X. He, and D. Zhou, "A survey of fault diagnosis for swarm systems," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 13–23, 2014.

[5] C. Chen, B. Bai, and H. Yu, "Improved algorithm for three-layer percetrons and its application to fault diagnosis," *Journal of Vibration, Measurement & Diagnosis*, vol. 20, no. 1, pp. 19–24, 2000.

[6] S. Ergezinger and E. Thomsen, "An accelerated learning algorithm for multilayer perceptrons: optimization layer by layer," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 31–42, 1995.

[7] K. Yuan and M. Liu, "The application of neural network in fault diagnosis inference," *Journal of Shandong Institute of Commerce and Technology*, no. 6, pp. 97–100, 2006.

[8] B. Liu, C. Sun, and X. Wang, "The neural network methodology for solid rocket motor fault diagnosis," *Computer Simulation*, no. 25, pp. 47–50, 2008.

[9] Y. Zhang and J. Wei, "Software fault predicting system and its knowledge model library," *Radar & ECM*, no. 3, pp. 60–63, 2007.

[10] J. Wu and P. Deng, "A fault diagnosis expert system based on regular complex artificial neural network," *Journal of Hubei Engineering University*, no. 26, pp. 32–35, 2006.

[11] J. Wu and J. Xiao, *Intellectual Fault Diagnosis and Expert System*, Science Press, Beijing, China, 1997.

[12] S. V. Kamarthi and S. Pittner, "Accelerating neural network training using weight extrapolations," *Neural Networks*, vol. 12, no. 9, pp. 1285–1299, 1999.

[13] Fly Science and Technology Product Research and Development Center, *Neural Network Theory and Realization of Malab7*, Publishing House of Electronics Industry, Beijing, China, 2005.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2016, Article ID 1286318, 12 pages
http://dx.doi.org/10.1155/2016/1286318

Research Article

Neural Network Based Fault Detection and Diagnosis System for Three-Phase Inverter in Variable Speed Drive with Induction Motor

Furqan Asghar,[1] Muhammad Talha,[2] and Sung Ho Kim[3]

[1]*Kunsan National University, Saemangeum Campus, Room No. 202/203, Osikdo-Dong, Gunsan-Si, Jeollabuk-Do 573-540, Republic of Korea*
[2]*School of Electronics and Information Engineering, Kunsan National University, Kunsan, Republic of Korea*
[3]*Department of Control and Robotics Engineering, Kunsan National University, Kunsan, Republic of Korea*

Correspondence should be addressed to Furqan Asghar; furqan.asghar@hotmail.com

Received 3 August 2016; Revised 5 October 2016; Accepted 12 October 2016

Academic Editor: Gang Li

Recently, electrical drives generally associate inverter and induction machine. Therefore, inverter must be taken into consideration along with induction motor in order to provide a relevant and efficient diagnosis of these systems. Various faults in inverter may influence the system operation by unexpected maintenance, which increases the cost factor and reduces overall efficiency. In this paper, fault detection and diagnosis based on features extraction and neural network technique for three-phase inverter is presented. Basic purpose of this fault detection and diagnosis system is to detect single or multiple faults efficiently. Several features are extracted from the Clarke transformed output current and used in neural network as input for fault detection and diagnosis. Hence, some simulation study as well as hardware implementation and experimentation is carried out to verify the feasibility of the proposed scheme. Results show that the designed system not only detects faults easily, but also can effectively differentiate between multiple faults. These results prove the credibility and show the satisfactory performance of designed system. Results prove the supremacy of designed system over previous feature extraction fault systems as it can detect and diagnose faults in a single cycle as compared to previous multicycles detection with high accuracy.

1. Introduction

In recent years, induction motors are predominantly fed from pulse width modulation voltage source inverter (PWM-VSI) for variable speed operation in various industrial applications. Indeed, the most common drive in industry is that with a VSI and induction motor. Recently, industry has begun to demand high power ratings. Inverter drive systems have become a solution for high power applications as these systems are more reliable than those supplied directly online. There are several types of fault such as controller faults, current sensor faults, switching device faults, motor faults, and dc bus faults [1, 2]. However, the use of inverters has some drawbacks as the introduction of power electronic converters came with an increased possibility of component failures mainly switching devices faults such as IGBT, MOSFET, and BJT.

These switching devices faults can be classified into open switch fault and short switch fault. A short switch fault not only generates an abnormal overcurrent in the power conversion system and generator but also causes some secondary problems like the demagnetization of synchronous generator. In this case, entire system should be shut down immediately for safety purpose, whereas an open switch fault does not require halting operation, but noise and vibrations can be induced in the system. Furthermore, the overcurrent flow in healthy switches can cause additional faults in these switches. Hence, open switch fault needs to be handled immediately. High costs due to standstill and repair, as well as general need to improve reliability, have led to research in fault detection systems [3–5].

Regarding diagnostics of these open switch faults, some of the previously researched approaches are mentioned here.

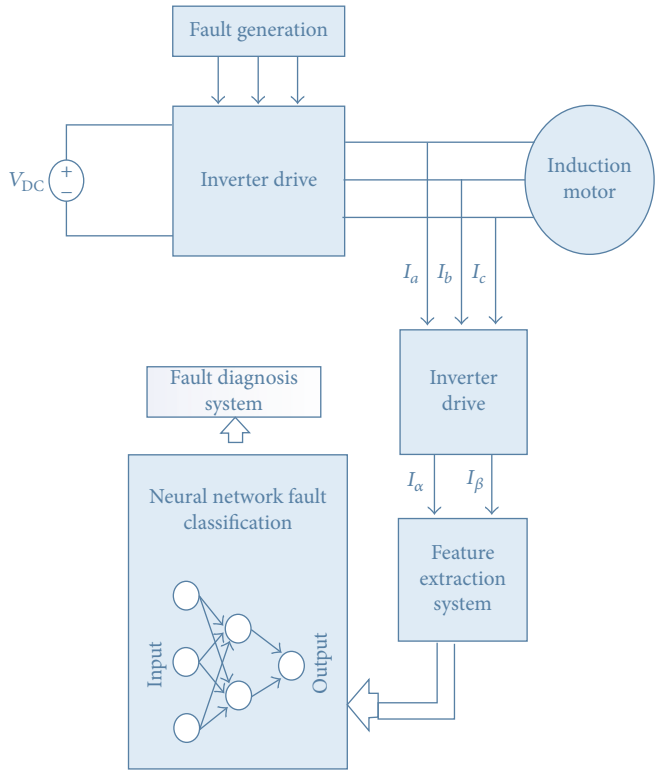


Figure 1: Fault detection and diagnosis system.

Peuget et al. suggested two methods for fault detection based on analysis of current vector trajectory and instantaneous frequency but technique using this frequency cannot detect faulty switches [6]. Khomfoi and Tolbert proposed a fault diagnostic technique to detect and identify fault location in multilevel inverter using neural network. However, this technique require computational effort [7]. Zidani et al. proposed a fuzzy based technique which detects faulty switches in PWM inverter for induction motor using Concordia pattern but this method was applied to single phase inverter [8]. Also Ko and Lee tried to propose fuzzy logic based fault diagnosis system but this technique is applicable only for single fault because fuzzy logic cannot differentiate between single and multiple faults angle as they overlap each other [9].

In this paper, neural network based fault detection and diagnosis method [10, 11] for three-phase inverter feeding an induction motor is designed to detect and localize failures in a set inverter-induction motor without the need of additional sensors or computational effort as shown in Figure 1. This technique can detect single or multiple switching device faults in three-phase inverter system by analyzing the stator current patterns and features extraction from that output current and then using these features in neural network method. Hardware implementation of simulation model is also carried out to confirm the feasibility of proposed scheme. Results proved that the designed fault detection and diagnosis system is more robust, accurate, systematic, effectual, and dynamic in detecting both single and multiple faults. This proposed technique is much better in comparison to previous techniques [7–10] as it can detect even multiple faults with 100% accuracy because of efficient feature extraction system as compared to 95% or lower accuracy of those techniques, and also it can detect single and multiple faults faster even in single current cycle. These simulated and hardware based system results prove the credibility and show the satisfactory performance of system.

2. Structure of Fault Detection and Diagnosis System

Performance diagnosis and status monitoring for variable speed AC drives are a need, more or less depending on its applications. Fault detection and diagnosis can avoid unplanned maintenance and standstill, to make it possible to run an emergency operation in case of faults. We discussed two fault situations in this paper:

(i) Single fault

(ii) Multiple faults

2.1. Feature Extraction System. Feature extraction system must be determined as a system that can provide neural network adequate significant details in pattern set so that highest accuracy in neural network performance can be achieved. Feature extraction system should be universal for different speed references by normalized functions. Also localization of each pattern class should be in limits defined by threshold. In previous research studies, different researchers also try to use feature extraction system for fault detection

and diagnosis for three-phase inverters and induction motors.

Ko and Lee tried to use feature extractor for his fault diagnosis system for inverter in wind turbine system [9]. He considered current angle and diameter as features with fuzzy logic technique for single fault detection only. This research work is not applicable for multiple faults as fuzzy logic will be unable to differentiate between single and multiple fault angles as they are overlapping each other. Therefore, further improvement is required to make this technique applicable for multiple faults detection.

Likewise, Zidani et al. [8] use the same angle and diameter difference in their research work along with fuzzy logic for fault detection. This approach is also applicable for only single fault detection.

Kadri et al. [10] also tried to use feature extraction for fault detection and diagnosis. But in their case, feature extractor is extracting only one feature as shown in

$$S_{\alpha,\beta} = \frac{\sum_{i=1}^{N} I_{s_{\alpha,\beta}}(i)}{\text{length}\left(I_{s_{\alpha,\beta}}\right) * \max\left(I_{s_{\alpha,\beta}}\right)}. \quad (1)$$

This technique is acceptable only for constant speed environment; also they themselves mentioned that feature extraction system needs to be improved for better classification performance. System accuracy is not much high as single feature value can create false detections in case of multiple faults, that is, two or three faults at a same time.

In our proposed system, we used four different features along with neural network system for fault detection and diagnosis system which makes our system more accurate and efficient and separates it from previous techniques. High number of features play a vital role in differentiating single and multiple faults.

We used the below mentioned mathematical equations to calculate our four features for both Simulink and hardware environment separately. To get the features in faulty conditions, we generate faults in inverter system manually for every possible scenario. This process is repeated several times to consider every possible change in features values because of noise and other uncertainties in real time environment. Data range of each feature in every fault condition is determined based on repetitive process results in next step to utilize best possible data for neural network training. Then neural network is trained by this data for further process.

As shown in flow chart in Figure 1, designing ANN based on fault detection and diagnosis system consists of four major steps.

Initially the measured output current data is transformed from three-phase to two-phase using Clarke transformation. This transformation is performed to evaluate the stator current pattern evolution when open circuit power switches fault occur in the inverter as shown in Figure 2. Also, in Figure 3, fault diagnosis space for each switch fault is shown. In healthy and normal conditions, stator current pattern in α-β reference frame is a circle whereas, in case of fault, current pattern is biased towards the direction of faulty switch.

In feature extraction system, the significant assignment which is an analysis of the current is performed for detecting

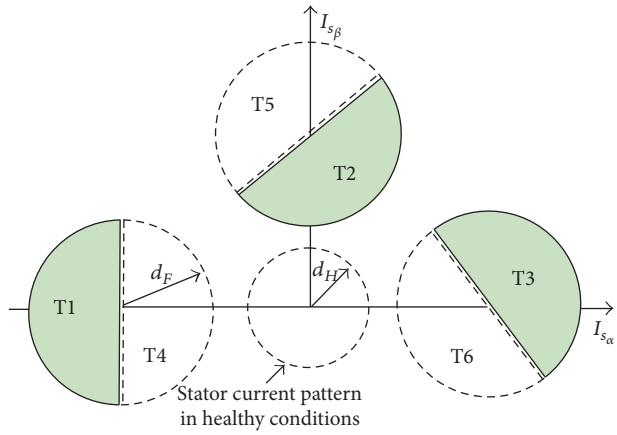


Figure 2: Current patterns in healthy and faulty modes.

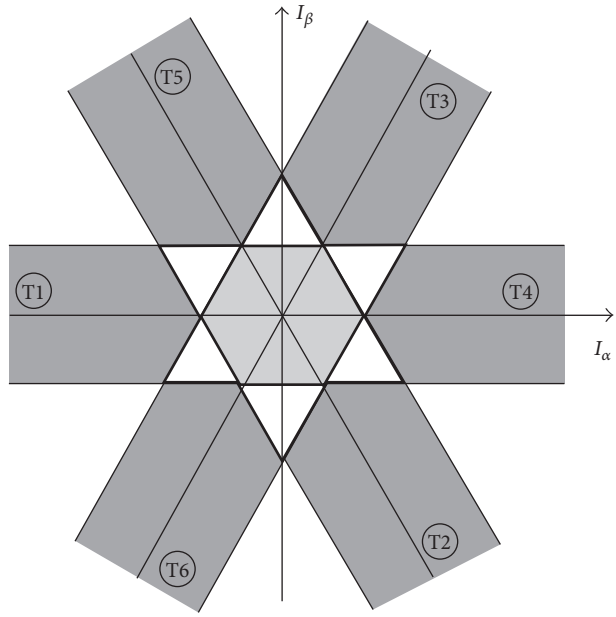


Figure 3: Fault detection and diagnosis space.

faulty switches. As mentioned before, transformation to stationary frame is done and various features from current pattern such as mean, surface, and angle are extracted.

During the fault detection step, faulty switches can be detected and identified using attributes obtained from the previous step. Output of this step will be 1 or 0 in which 0 represents off and 1 represent on to the respective state of switch.

Current means across both axes (α and β) can be calculated using the following equations:

$$I_\alpha = \sum_{i=1}^{N} \frac{I_\alpha(i)}{\text{length}\left(I_\alpha\right)},$$
$$I_\beta = \sum_{i=1}^{N} \frac{I_\beta(i)}{\text{length}\left(I_\beta\right)}, \quad (2)$$

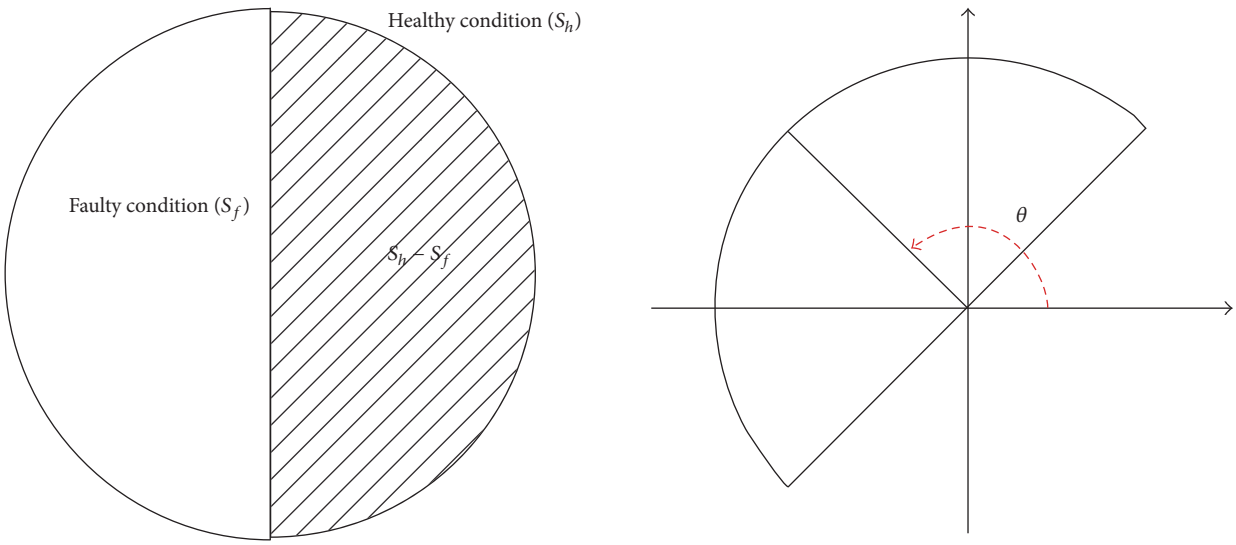


Figure 4: Definition of E_s and I_θ.

where N defines the number of samples. Third feature angle to the current pattern (I_θ) can be calculated as

$$I_\theta = \tan^{-1}\left(\frac{I_\beta}{I_\alpha}\right), \tag{3}$$

where I_β is center of the β-frame and I_α is the center of α = frame. I_θ can be calculated by an angle between the origin and center of mass as shown in Figure 4.

Fourth feature surface difference of the current patterns between healthy and faulty condition shown in Figure 4 can be calculated as

$$E_s = S_h - S_f, \tag{4}$$

where S_h is current vector surface in healthy mode and S_f is current vector surface in faulty mode.

Extracted features data shows that this block output gives an appropriate decorrelation between different kinds of single and multiple faults.

2.2. Artificial Neural Network System. Artificial neural networks (ANNs) are a family of models inspired by biological neural networks which are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown as shown in Figure 5.

Architecture of the designed fault detection and diagnosis neural network is a feed forward network as the input data contain continuous features. Our neural network is based on one input layer with four neurons each for one of four extracted features (I_α (mean), I_β (mean), angle, and surface difference), one hidden layer with 15 neurons, and one output layer with 13 neurons referring to the number of faults we want to detect. Sigmoid activation function is used for hidden and output layers. Target output of system is binary (1 or 0).

Initially, neural network training is required with normal and faulty data. Then this trained neural network is used for fault detection system. Target output for normal case will be as follows:

$$\begin{aligned}&\text{Target Output}\\ &= [1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0],\end{aligned} \tag{5}$$

where 1 represents the normal condition and 0's represent that currently there is no fault. Output will be 1 in case of respective fault such as for fault T1; output will be like as shown in

$$\begin{aligned}&\text{Target Output}\\ &= [0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0].\end{aligned} \tag{6}$$

3. Simulation Studies

Neural network based fault detection and diagnosis system for three-phase inverter with induction motor is briefly described in Section 2. Now the Matlab/Simulink simulation study is examined in this section to confirm its reliability.

3.1. Fault Generation. In our system, we generate fault in three-phase inverter system externally to check the performance of our proposed system in faulty conditions. We generate single and multiple faults by opening the IGBTs of inverter so that system can receive the input signal without respective phases. In case of double faults, usually there is high possibility of fault in two gates used in same phase like T1&T2, T5&T6, and so forth. Complete phase missing can be noticed in case of faults like short circuit or line to line fault. But being on the safe side, we trained our neural network for every possible scenario in two gate switch faults. Some of the generated faults are T1, T2, T3, T1&T2, T2&T3, T4&T6, and so on.

3.2. Feature Data for Training Neural Network. In the start, we need to train neural network to work in efficient way

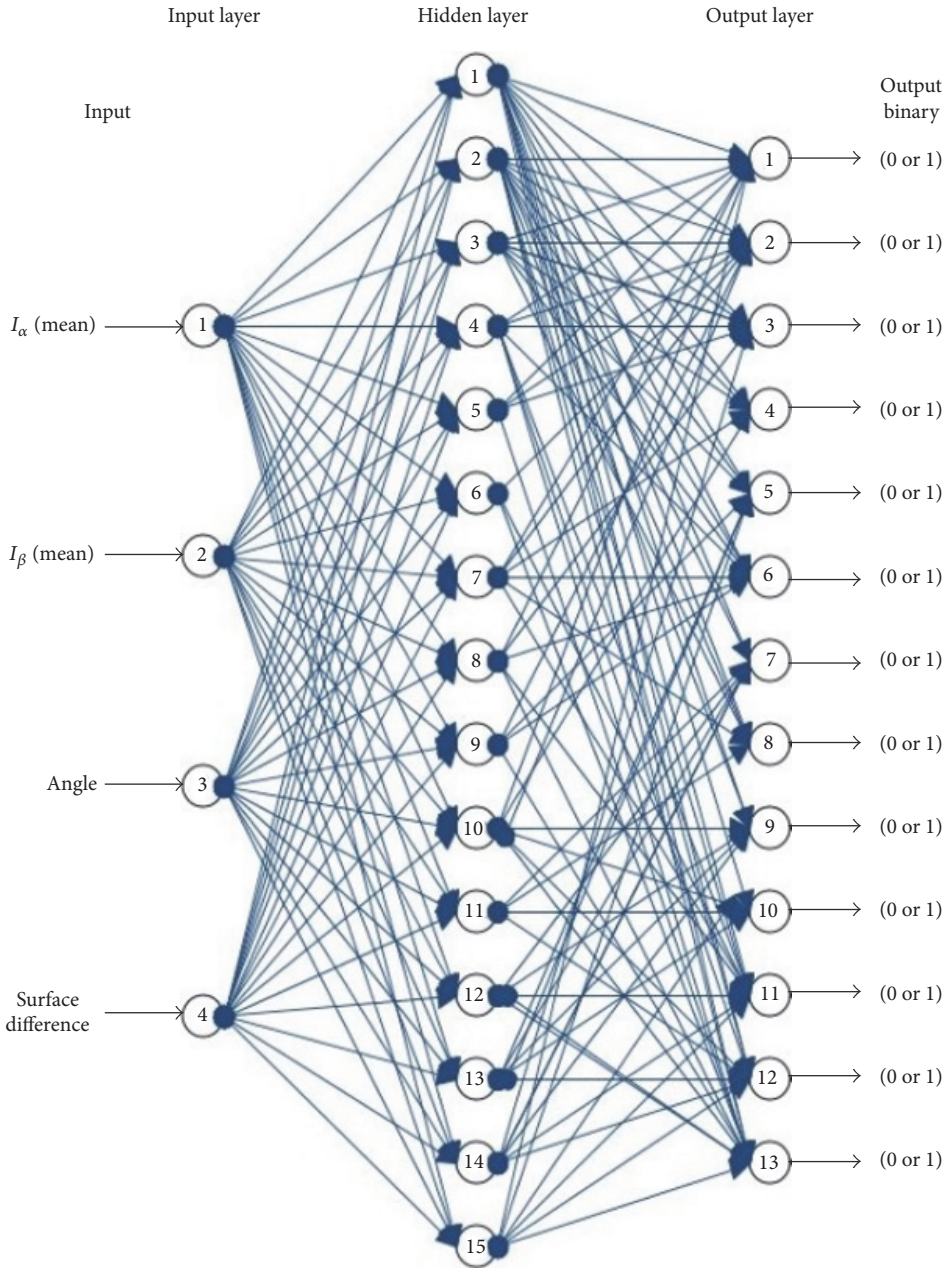


Figure 5: Basic architecture of artificial neural network.

according to desired environment. System requires normal and faulty feature data for training purposes as shown in Table 1.

After training neural network with the above-mentioned data, we can use this neural network system to detect faults in three-phase inverter feeding an induction motor. This system even works in case that extracted features in real time environment are not exactly the same as for training the network.

3.3. Simulation Results. Designed neural network based fault detection and diagnosis system for three-phase inverter in a variable speed drive is tested in case of single and multiple faults at a time. Simulink based system diagram can be seen in Figure 6.

In the simulation test sets, system shows satisfactory classification performance in both single and multiple faults cases.

Internal configuration of feature extraction block in normal mode can be seen in Figures 7 and 8.

In Figure 7, we can see that artificial neural network block output is indicating that system is currently running in normal condition. Likewise, in Figure 8, α-β transformed current pattern graph shows circle which indicates system normal condition.

Designed system response can be seen in Figure 9 as it shows that system is working efficiently in both single and multiple switch fault modes.

Figures 10 and 11 show α-β transformed current pattern graph when system is operating in single fault whereas

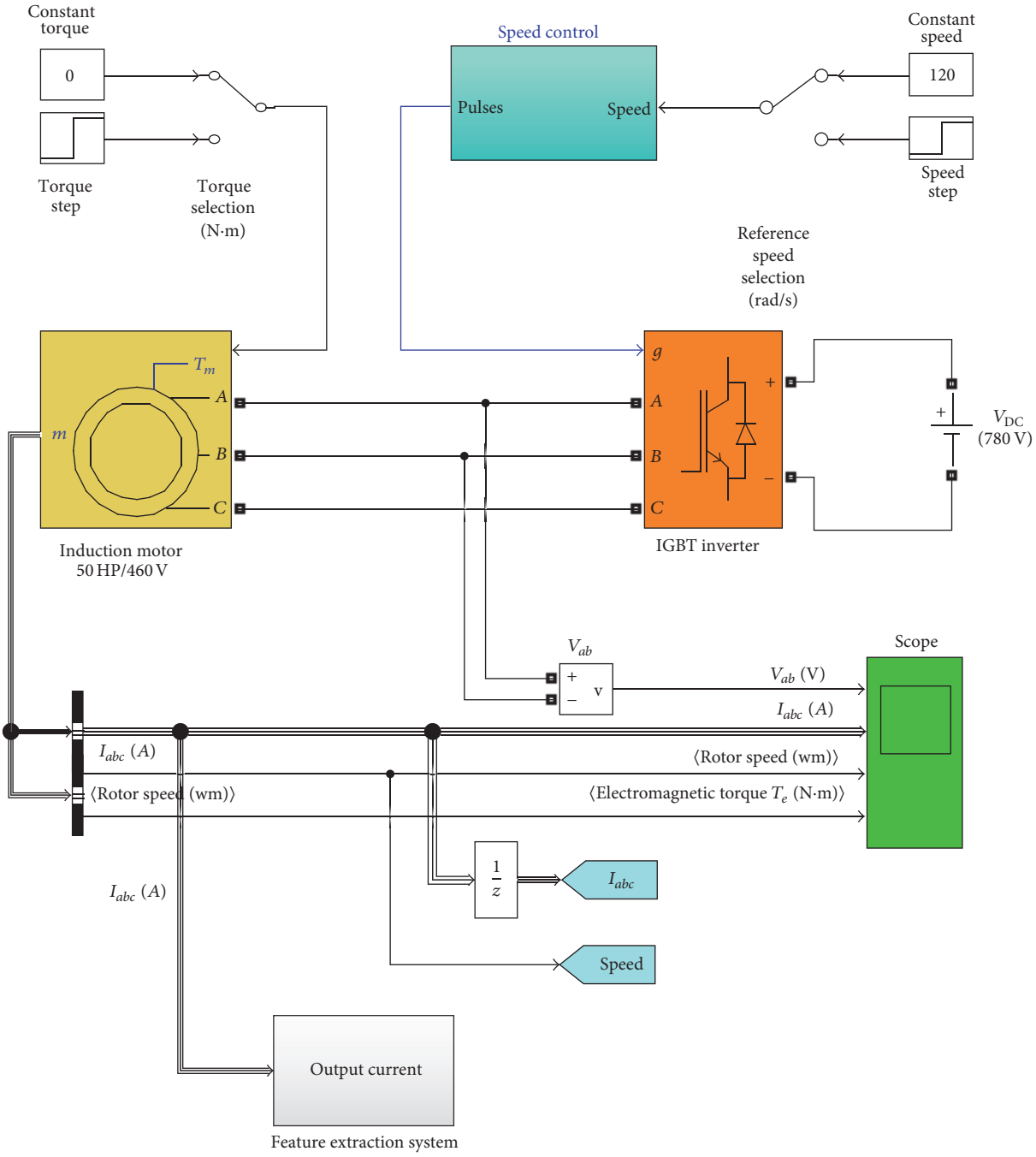


Figure 6: Neural network based fault detection and diagnosis system.

Figure 12 shows α-β transformed current pattern graph during system operating in multiple faults, respectively.

3.4. Comparative Studies. As mentioned in Table 2 according to reference number of fault detection techniques, different researchers [6–10, 12] proposed fault detection and diagnosis techniques for inverters in the past. Some researchers designed their techniques for single phase inverters only.

Comparison of our technique with some of those methods shows that proposed fault detection is more robust and efficient than previously researched methods in both factors: accuracy and response time. As highest accuracy for above-mentioned previous fault detection methods is 95% at its best even most of these fault detection methods are for single fault, whereas, in our case, we are getting 100% accuracy even in multiple faults. Also proposed fault detection method can detect fault even in single current/voltage cycle whereas previous techniques need at least two cycles for fault detection.

4. Hardware Implementation

Proposed neural network based fault detection and diagnosis technique require hardware experimentation to prove their

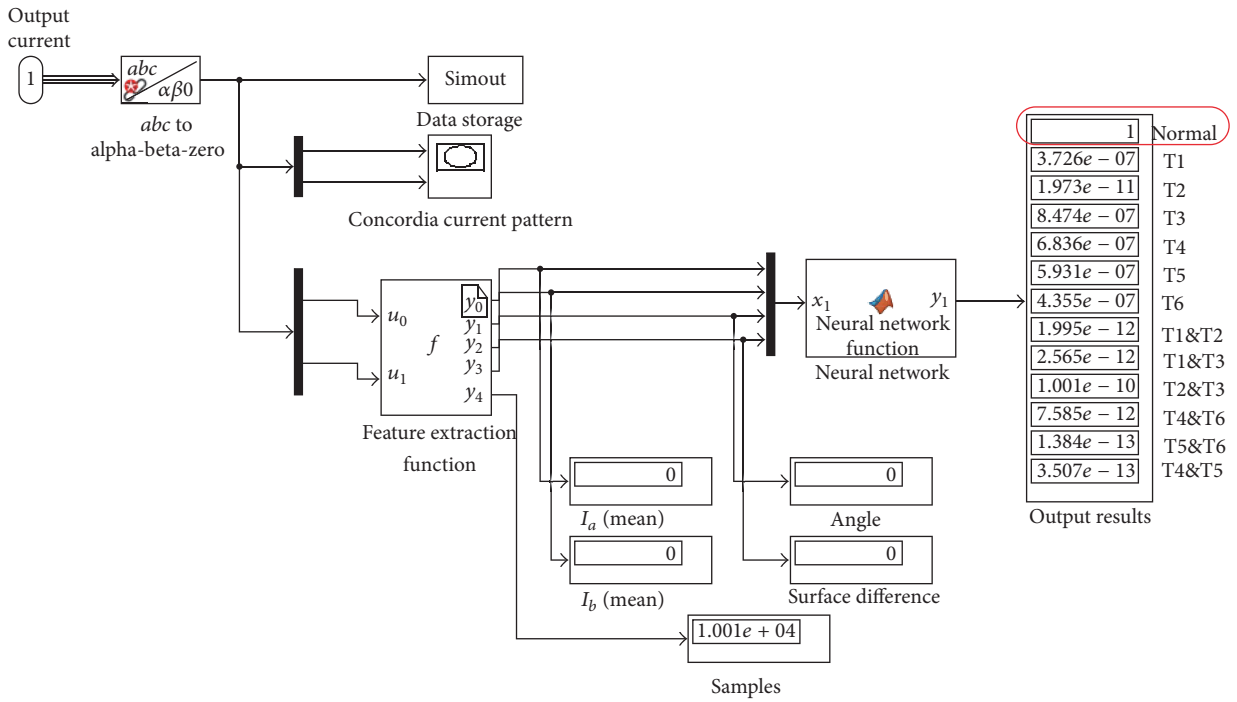


Figure 7: Internal configuration of feature extraction block with system normal mode.

Table 1: Feature data for training neural network.

System condition	Feature data for training			
States	I_α (mean)	I_β (mean)	Angle	Surface
Normal	0.25	0	0	0
T1	−7.32	−3.89	207	12.33
T2	7.09	−4.19	300	12.73
T3	0.08	8.36	89	11.43
T4	7.3	3.95	28	11.26
T5	−6.91	4.37	122	11.6
T6	−0.02	−8.15	269	13.09
T1&T2	−1.27	−9.03	262	37.66
T1&T3	3	5.15	126	36.66
T2&T3	8.45	3.4	22	36.55
T4&T6	7.31	−5.68	308	37.19
T5&T6	−8.15	−3.26	202	38.77
T4&T5	1.76	9.12	79	38.80

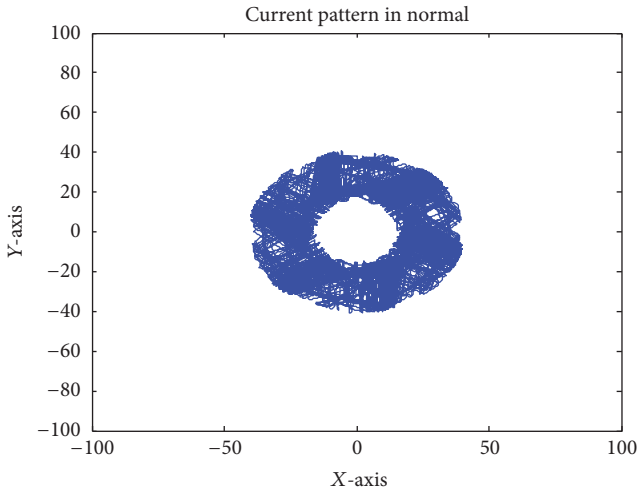


Figure 8: α-β transformed current pattern graph for system in normal mode.

accuracy and authenticity. Therefore, three-phase inverter is used to implement our proposed technique to compare the results with Simulink designed system output as shown in Figure 14. Three-phase inverter output voltages are used for feature extraction and further process.

SPWM inverter [13] is designed by using dsPIC30F4011 digital signal processing chip to generate switching pulses along with Lab-Volt (8134-20). .NET (C#) based programing environment and monitoring system is used to design features extractor, artificial neural network (ANN), fault detection, and monitoring system. NI DAQ X-Series USB-6343 is used to acquire three-phase voltage signals from Lab-Volt output. System block diagram is shown in Figure 13.

4.1. SPWM Inverter Designing and Data Acquisition. dsPIC30F4011 is a high speed digital signal processing chip capable of generating high frequency PWM signals. Therefore, we use this chip to generate the gate pulses for three-phase inverter. Generated gate switching signals to three-phase inverter IGBTs are shown in Figure 15.

Lab-Volt is a training system with multipurpose modules that can be attached with various systems for experimentation

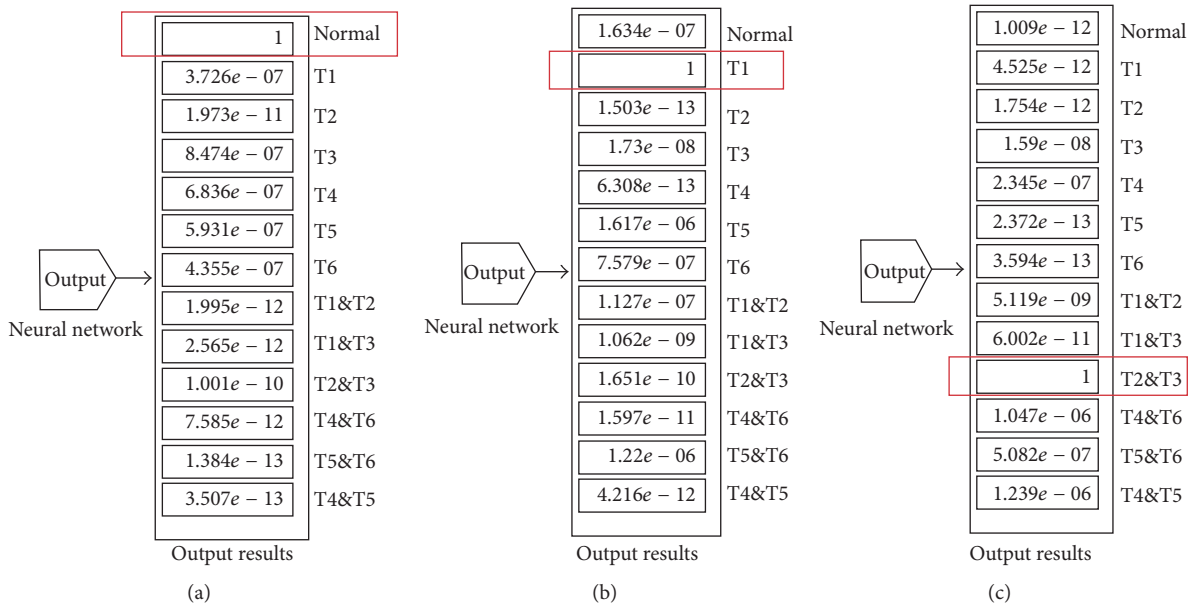


Figure 9: Simulation output results in various modes: (a) normal, (b) single fault, and (c) double fault.

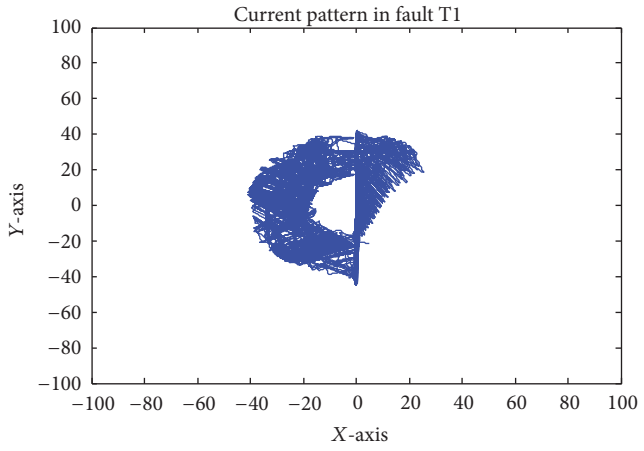


Figure 10: α-β transformed current pattern graph for system in single fault mode T1.

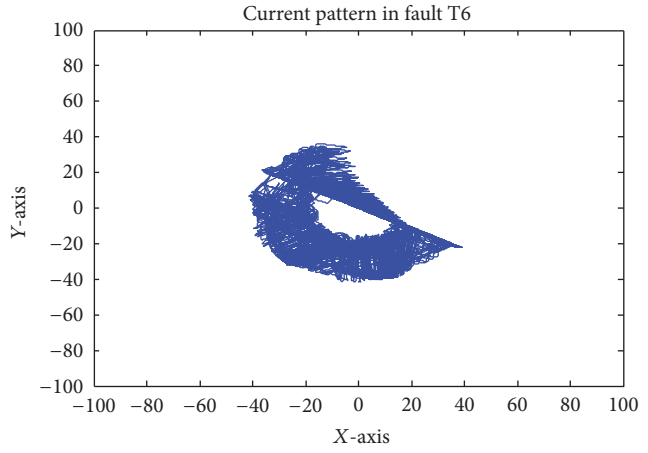


Figure 11: α-β transformed current pattern graph for system in single fault mode T6.

such as battery charging/discharging, data acquisition, Power MOSFETs and IGBTs, AC power interface, wind turbine emulator, and different types of loads such as resistive, capacitive, and inductive. Hence, Lab-Volt along with different modules is used to make three-phase inverter connected to RL load.

NI DAQ X-Series USB-6343 for USB, PCI express, and PXI express is one of the advanced data acquisition devices ever designed by National Instruments. Main features of NI DAQ X-Series USB are onboard timing, triggering, and optimization for use with multicore PCs. This device integrates high performance analog, digital, and counter/timer functionality onto a single device, making them well suited for a broad range of applications, from basic data logging to control and test automation. We are using this device to acquire three-phase output from Lab-Volt.

4.2. Designing of .NET Based Fault Detection and Diagnosis System. Microsoft visual studio is an integrated development environment used to develop computer programs as well as web applications and services. This IDE program supports different programming languages and allows the code editor to support almost every programming language. Built-in languages include C, C++, and C++/CLI, .NET, C#, and F#.

Three-phase voltage signal is being read in C# program and then converted into two-phase using Clark Transformation (α, β-axis). Various features are extracted from this two-phase signal such as I_α (mean), I_β (mean), pattern angle, and surface difference.

These four features are used to train artificial neural network in Matlab environment. Then trained artificial neural network is used in C# program to detect single or multiple

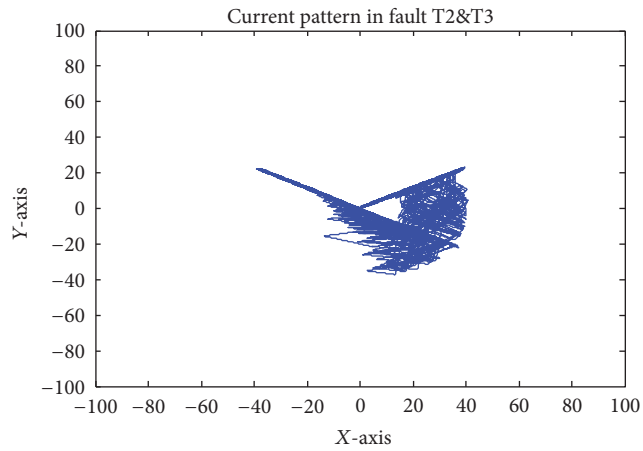


Figure 12: α-β transformed current pattern graph for system in multiple fault mode T2&T3.

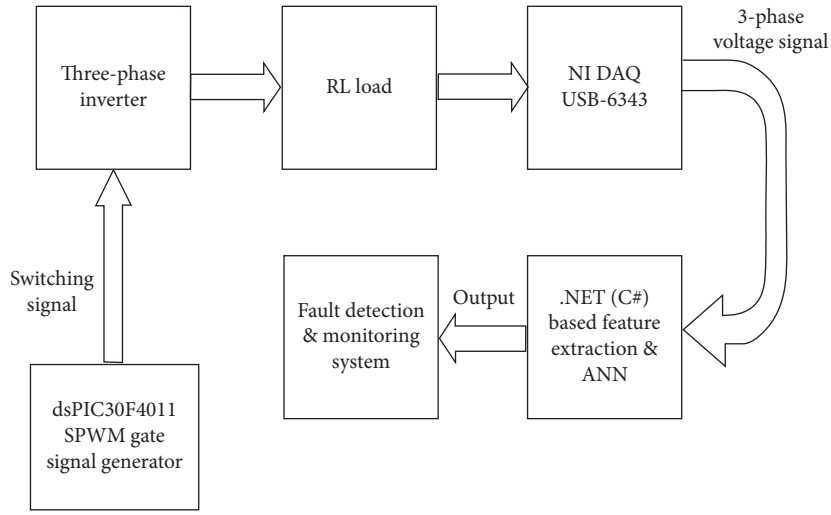


Figure 13: Hardware experimentation block diagram.

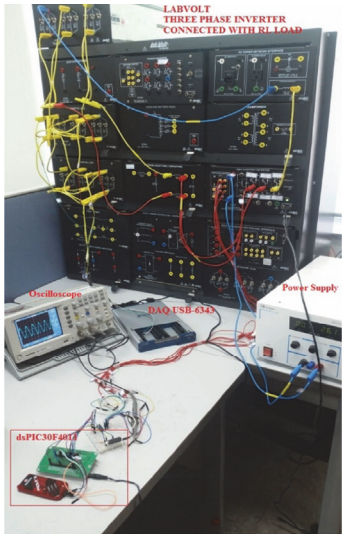


Figure 14: Experimental setup.

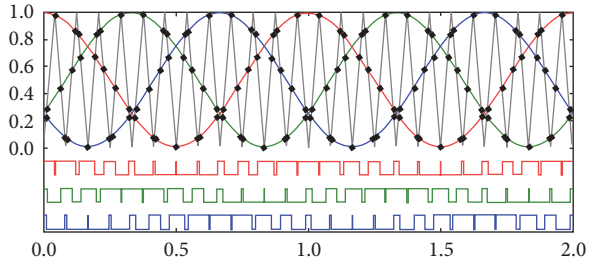


Figure 15: SPWM signal.

faults. Six DIP switches are connected to input of three-phase inverter gate signals to add faults externally. Designed fault detection and diagnosis monitoring system are shown in Figure 16.

This figure shows that, in fault detection monitoring system, there are several options for better monitoring purposes such as loading three-phase output signal along with Clarke

Table 2: Comparison table between different methods.

Fault detection Technique	Accuracy		Response time	Inverter type
	Single	Multiple		
Reference [6]	100%	<80%	Slow	3-phase
Reference [7]	100%	90%	Fast	3-phase
Reference [8]	100%	Not applicable	Medium	3-phase
Reference [9]	100%	Not applicable	Fast	3-phase
Reference [10]	95%	95%	Medium	3-phase
Reference [12]	100%	95%	Slow	3-phase
Proposed method	100%	100%	Fast	3-phase

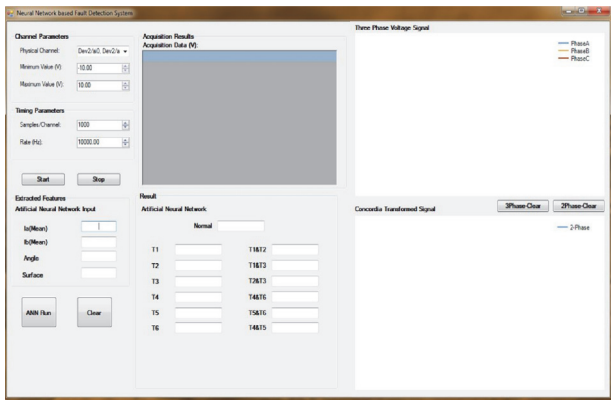


Figure 16: Artificial neural network based fault detection and diagnosis system.

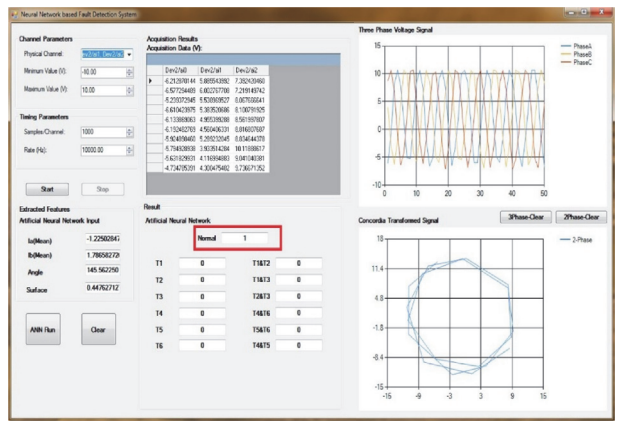


Figure 17: System operating in normal condition.

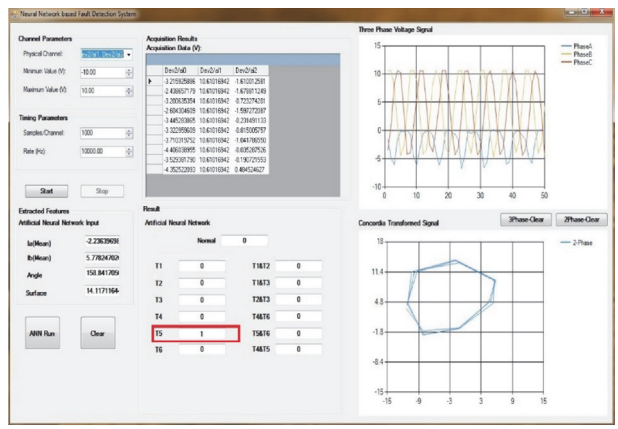


Figure 18: System operating in single fault (T5) condition.

transformed two-phase signal. Receive signal parameter and timing configuration setting are also available. Extracted features values can be monitored continuously.

4.3. Experimentation Results. Experimental setup as shown in Figure 14 is used to perform experiments with three-phase inverter system running in various conditions such as normal, single fault, and multiple faults. Lab-Volt based three-phase inverter can be seen connected with dsPIC30F4011 for gate switching signals, NI DAQ USB for output voltage acquisition, power supply, oscilloscope, and PC system for further processing. Inverter output voltages and frequency are constant, that is, 20 $V_{p\text{-}p}$ and 50 Hz. Output voltage across load is acquired using NI DAQ in .NET based fault detection and diagnosis system. System response is shown in Figures 17, 18, and 19.

Figure 17 shows that three-phase inverter system is working in normal condition as we can see complete three-phase voltage and circle in two-phase graph and normal condition is on in ANN output.

On the other hand, in Figure 18, system is indicating T5 fault as we generate T5 gate signal fault externally and also we can see the faulty condition in three-phase and two-phase voltage graph. Figure 19 shows that system is running with double fault at switches T4&T6 simultaneously.

Experimentation results verified that proposed system is robust and accurate. Also it can detect and diagnose single and multiple faults efficiently.

Performance table for proposed neural network based fault detection and diagnosis system is also shown.

Simulation and hardware experimentation has been performed with every fault condition a number of times to verify the system accuracy. Data with respect to system condition is shown in Table 3.

5. Conclusion

In this research work, neural network based fault detection and diagnosis system for field oriented induction motor

Table 3: Performance table for ANN based fault detection and diagnosis system.

Target States	Actual output Neural network												Performance% Simulation	Performance% Experiment
Normal	1	0	0	0	0	0	0	0	0	0	0	0	100%	100%
T1	0	1	0	0	0	0	0	0	0	0	0	0	100%	100%
T2	0	0	1	0	0	0	0	0	0	0	0	0	100%	100%
T3	0	0	0	1	0	0	0	0	0	0	0	0	100%	100%
T4	0	0	0	0	1	0	0	0	0	0	0	0	100%	100%
T5	0	0	0	0	0	1	0	0	0	0	0	0	100%	100%
T6	0	0	0	0	0	0	1	0	0	0	0	0	100%	100%
T1&T2	0	0	0	0	0	0	0	1	0	0	0	0	100%	100%
T1&T3	0	0	0	0	0	0	0	0	1	0	0	0	100%	100%
T2&T3	0	0	0	0	0	0	0	0	0	1	0	0	100%	100%
T4&T6	0	0	0	0	0	0	0	0	0	0	1	0	100%	100%
T5&T6	0	0	0	0	0	0	0	0	0	0	0	1	100%	100%

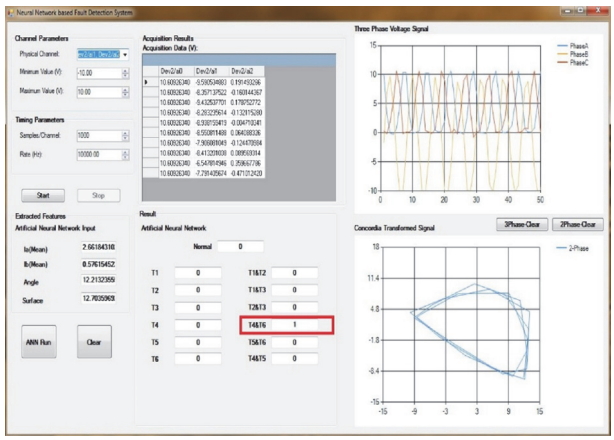


Figure 19: System operating in multiple faults (T4&T6) condition.

drive have been designed and tested in both simulation and hardware environment. Considered fault types for system are switching device open faults. Feature extraction system based on current means, surface difference, and angle to current pattern is also discussed here. Extracted features play a vital role in fault detection and localization. A robust algorithm is needed for getting the correct angle because inaccurate value can lead to incorrect results. No additional sensors and complicated calculations are required for designed system. Additionally, faulty switch detection and identification can be performed even in every single current or voltage cycle with high accuracy which proves that system performance is much better than previous fault detection systems [6–10, 12] as they take two or more cycles for fault detection. Simulated as well as hardware experimentation results as shown above prove the credibility and show the satisfactory performance of designed fault detection and diagnosis system for three-phase inverter feeding an induction motor. Multiple features extraction from three-phase current/voltage output signal plays a key role in proving proposed system supremacy over previous fault detection techniques. As shown in results, proposed system is fast, efficient, and 100% accurate for single or multiple faults.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

[1] D. Kastha and B. K. Bose, "Investigation of fault modes of voltage-fed inverter system for induction motor drive," *IEEE Transactions on Industry Applications*, vol. 30, no. 4, pp. 1028–1038, 1994.

[2] D. E. Kim and D. C. Lee, "Fault diagnosis of three phase PWM inverters using wavelet and SVM," *Journal of Power Electronics*, vol. 9, pp. 377–385, 2009.

[3] A. Muetze and A. Binder, "Practical rules for assessment of inverter-induced bearing currents in inverter-fed AC motors up to 500 kW," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1614–1622, 2007.

[4] M. R. Ubale, R. B. Dhmale, and S. D. Lokhande, "Open switch fault diagnosis in three phase inverter using diagnostic variable method," *International Journal of Research in Engineering and Technology*, vol. 2, no. 12, pp. 636–641, 2013.

[5] S. O. Ibrahim, K. N. Faris, and E. Abo Elzahab, "Implementation of fuzzy modeling system for faults detection and diagnosis in three phase induction motor drive system," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 1, pp. 27–46, 2015.

[6] R. Peuget, S. Courtine, and J.-P. Rognon, "Fault detection and isolation on a pwm inverter by knowledge-based model," *IEEE Transactions on Industry Applications*, vol. 34, no. 6, pp. 1318–1326, 1998.

[7] S. Khomfoi and L. M. Tolbert, "Fault diagnostic system for a multilevel inverter using a neural network," *IEEE Transactions on Power Electronics*, vol. 22, no. 3, pp. 1062–1069, 2007.

[8] F. Zidani, D. Diallo, M. El Hachemi Benbouzid, and R. Nait-Said, "A fuzzy-based approach for the diagnosis of fault modes in a voltage-fed PWM inverter induction motor drive," *IEEE*

Transactions on Industrial Electronics, vol. 55, no. 2, pp. 586–593, 2008.

[9] Y.-J. Ko and K.-B. Lee, "Fault diagnosis of a voltage-fed PWM inverter for a three-parallel power conversion system in a wind turbine," *Journal of Power Electronics*, vol. 10, no. 6, pp. 686–693, 2010.

[10] F. Kadri, S. Drid, F. Djeffal, and L. Chrifi-Alaoui, "Neural classification method in fault detection and diagnosis for voltage source inverter in variable speed drive with induction motor," in *Proceedings of the 8th International Conference and Exhibition on Ecological Vehicles and Renewable Energies (EVER '13)*, pp. 1–5, Monte Carlo, Monaco, March 2013.

[11] V. Geometry and S. Selvaperumal, "Fault detection and classification with optimization techniques for three phase single inverter circuit," *Journal of Power Electronics*, vol. 16, pp. 1–14, 2016.

[12] S. Khomfoi and L. M. Tolbert, "Fault diagnosis and reconfiguration for multilevel inverter drive using AI-based techniques," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 6, pp. 2954–2968, 2007.

[13] S. Phogat, "Analysis of single-phase SPWM inverter," *International Journal of Science and Research*, vol. 3, pp. 1793–1798, 2014.

Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2016, Article ID 2670210, 10 pages
http://dx.doi.org/10.1155/2016/2670210

Research Article

Reliability Assessment of Cloud Computing Platform Based on Semiquantitative Information and Evidential Reasoning

Hang Wei and Pei-Li Qiao

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, Heilongjiang, China

Correspondence should be addressed to Pei-Li Qiao; airwie@qq.com

Received 31 July 2016; Accepted 14 September 2016

Academic Editor: Zhijie Zhou

A reliability assessment method based on evidential reasoning (ER) rule and semiquantitative information is proposed in this paper, where a new reliability assessment architecture including four aspects with both quantitative data and qualitative knowledge is established. The assessment architecture is more objective in describing complex dynamic cloud computing environment than that in traditional method. In addition, the ER rule which has good performance for multiple attribute decision making problem is employed to integrate different types of the attributes in assessment architecture, which can obtain more accurate assessment results. The assessment results of the case study in an actual cloud computing platform verify the effectiveness and the advantage of the proposed method.

1. Introduction

Cloud computing, as the Pay-Per-Use computing service model which has the capacity for large-scale resource scheduling, is attracting more and more attention. Currently, scheduling strategy of cloud computing can be divided into 3 types: performance-driven scheduling strategy [1, 2], profit-driven scheduling strategy [3, 4], and reliability-driven scheduling strategy [5, 6]. With the upsurge of commercial operation, the requirements for the cloud computing systems reliability are more stringent than ever. The cloud systems are more dangerous because the data, services, and computing resource are highly clustered, which is more easily to be broken by the outside hackers or internal users with unstable operation. Recently, the failure events of cloud service frequently occurred. For instance, Azure cloud services of Microsoft had two times interrupt service due to the system failure in March 2015, which affected most parts of the central and eastern United States service areas. In Apple Cloud Services, about 11 iCloud-related features had experienced serious technical issues on May 20, 2015, including iCloud Account and iCloud mail, by which 2 million iCloud users were affected. Therefore, the first 2 strategies are not practicable because the reliability is not considered. Thus, as one of the significant factors of the task scheduling, the reliability assessment should be deeply considered in the cloud computing platforms [7].

Current assessment methods for cloud computing reliability mainly focus on the security assessment method [8], the risk assessment method [9], and the service assessment method [10]. But these approaches make the reliability assessment only in a narrow aspect point of view. The cloud computing system with huge computing process and complex structure is hard to establish the comprehensive assessment model because of the high heterogeneity in the infrastructure nodes. In cloud computing platform, a large number of reliability factors need to be considered, and the information between these factors is closely linked and mutually restricted. More intractably, most reliability factors include uncertainty information, which increase the difficulty for reliability assessment [11].

The essence of reliability assessment is to obtain a reasonable security state through the rational integration of multiattributes [12]. But in actual cloud computing operation, the available knowledge is incomplete and uncertain due to the limitation of information acquisition and resolution. Thus,

the corresponding assessment models must have capability to deal with this uncertain information in assessment process. The current methods of reliability assessment include factor analysis [13], utility theory analysis [14], analytic hierarchy process [15, 16], Bayesian inference [17, 18], fuzzy logic reasoning [9, 19], hybrid method [20], and artificial intelligence reasoning [21–23]. The factor analysis can express the relationship between multiple attributes by a few weighted factors, which aims to reflect the comprehensive information of the system. But this method has a strict standard on the testing data. The utility theory analysis establishes the utility maximum function by analysis various factors. But the capability of information analysis is not good, and the solution of the utility function is too complicated to deal with a large amount of attributes. The analytic hierarchy process can effectively use the combination knowledge which includes both qualitative and quantitative information. After the quantification of the qualitative attributes, the weight vectors are determined to make corresponding assessment. But this method could not give the appropriate results when dealing with the incomplete information. Bayesian inference is a classical method of probabilistic reasoning, which can deal with uncertain problems. A statistical conclusion is obtained by using prior knowledge and sample information, but a large number of repeated trials are needed to obtain the probabilities; thus the training difficulty is increased. Fuzzy logic reasoning has a significant effect on the processing of a large number of inaccurate data, and it is suitable for the complex fuzzy systems. However, the precise membership function is hard to establish, which would produce large deviation of the results. Hybrid method is the combination of two or more computational approaches which can provide better advantages than single approaches. But it is hard to solve the conflict of the attributes in cloud computing systems, because of its parameter adjustment mode. D-S theory [24, 25] not only can deal with both qualitative knowledge and quantitative data very well, but also can deal with various types of uncertain information. Compared with Bayesian inference, the D-S theory can deal with the uncertainty due to the limitation of expert abilities or prior knowledge. After that, Wang et al. [26, 27] proposed evidential reasoning (ER) rule in late 1990s. ER rule is developed as a multicriteria decision analysis (MCDA) approach on the basis of belief decision and D-S theory [28, 29]. It has been widely employed in different fields [30, 31]. Compared to the D-S theory of evidence, the calculation process of the ER rule is linear, which reduces the computational complexity [32]. Moreover, the most important thing is that it can deal with the conflict of evidence attributes which cannot be solved by using D-S theory [33]. The conflict of evidence refers to the basic probability assignment of the two empty focal elements when the evidence is synthesized. The reliability assessment of cloud computing has different types of information needed to be integrated, which is the advantage of ER rule.

According to above description, the innovations of this paper are shown: (1) ER rule is first employed to establish the reliability assessment model of the cloud computing platform in this paper; (2) a new reliability assessment architecture including four aspects is proposed. A case study is given in the last section, and it demonstrates the good performance on assessing the reliability of a real cloud computing platform by using the proposed method.

The paper is organized as follows. In Section 2, the principle of ER rule is described. In Section 3, the reliability assessment of cloud computing platform is described, and the new reliability assessment architecture is proposed. In Section 4, a case study for assessing the reliability of an actual cloud computing platform is given, and the simulation results are analyzed. Finally, this paper is concluded in Section 5.

2. The Evidential Reasoning Rule: Basics

As mentioned above, ER rule can integrate different types of information, which include qualitative knowledge and quantitative data. It can also express various types of uncertainties such as fuzzy uncertainty, probabilistic uncertainty, and ignorance uncertainty [34]. The basic principle of the ER rule is introduced in this section.

Assume that there are M basic attributes $\{r_1, r_2, \ldots, r_i, \ldots, r_M\}$ of a general attribute R in a two-level hierarchy, and $\{w_1, w_2, \ldots, w_i, \ldots, w_M\}$ denotes the weights of the basic attributes, where $0 \le w_i \le 1$. There are N assessment grades; then the basic step of the ER rule can be concluded as follows.

(1) The belief degree should be converted into the basic probability mass. The process is shown as follows:

$$P_{i,j} = w_i \beta_{i,j}, \tag{1}$$

$$P_{i,\Theta} = 1 - w_i \sum_{j=1}^{N} \beta_{i,j}, \tag{2}$$

$$\overline{P}_{i,\Theta} = 1 - w_i, \tag{3}$$

$$\widetilde{P}_{i,\Theta} = w_i \left(1 - \sum_{j=1}^{N} \beta_{i,j} \right), \tag{4}$$

where $P_{i,j}$ denotes the basic probability mass which refers to the degree of the ith basic attribute supporting the hypothesis that the attribute is assessed to the jth grade. $P_{i,\Theta}$ denotes the rest of probability mass which has not been assigned to any consequent according to the ith basic attribute. $\overline{P}_{i,\Theta}$ denotes the unassigned basic probability mass which refers to the unimportant degree of the ith basic attribute. $\widetilde{P}_{i,\Theta}$ denotes the unassigned basic probability mass which refers to the incomplete degree of the ith basic attribute.

(2) ER rule is used to combine the first i basic attributes; the detailed process is described as follows:

$$P_{I(i+1),j} = K_{I(i+1)} \left[P_{I(i),j} P_{i+1,j} + P_{I(i),j} P_{i+1,\Theta} + P_{I(i),\Theta} P_{i+1,j} \right],$$

$$P_{I(i),\Theta} = \overline{P}_{I(i),\Theta} + \widetilde{P}_{I(i),\Theta},$$

$$\begin{aligned}
&\widetilde{P}_{I(i+1),\Theta} \\
&\quad = K_{I(i+1)}\left[\widetilde{P}_{I(i),\Theta}\widetilde{P}_{i+1,\Theta} + \widetilde{P}_{I(i),\Theta}\overline{P}_{i+1,\Theta} + \overline{P}_{I(i),\Theta}\widetilde{P}_{i+1,\Theta}\right], \\
&\overline{P}_{I(i+1),\Theta} = K_{I(i+1)}\left[\overline{P}_{I(i),\Theta}\overline{P}_{i+1,\Theta}\right],
\end{aligned} \tag{5}$$

where $P_{I(i),j}$ denotes the probability mass which refers to the degree of the first i basic attributes supporting the hypothesis that the attribute is assessed to the jth grade. $K_{I(i+1)}$ can be calculated as

$$K_{I(i+1)} = \frac{1}{1 - \sum_{k=1}^{N}\sum_{j=1,z\neq k}^{N} P_{I(i),k}P_{i+1,j}}. \tag{6}$$

(3) Thus, the final belief degree to the jth consequent and the remaining belief degree can be calculated as

$$\begin{aligned}
\widehat{\beta}_j &= \frac{P_{I(M),j}}{1 - \overline{P}_{I(M),\Theta}} \quad \left(j = 1, 2, \ldots, N\right), \\
\widehat{\beta}_\Theta &= \frac{\widetilde{P}_{I(M),\Theta}}{1 - \overline{P}_{I(M),\Theta}}.
\end{aligned} \tag{7}$$

3. Reliability Assessment of Cloud Computing Platform with ER Rule

3.1. Problem Formulation. In this section, the assessment attributes can be divided into 2 categories: quantitative data and qualitative knowledge. The quantitative data represents the data that can be expressed as certain quantity, amount, or range, such as Mean Time to Failures (MTTF) of central units, frequency of network attack, fault tolerant rate of cloud computing platform, and Mean Time to Repair (MTTR) of service. These attributes can be collected by the data monitoring systems. However, the qualitative knowledge is associated with the subjective quality of a situation or phenomenon, such as scalability of cloud system, controllability of access terminal, completeness of service, and stability of service. Qualitative knowledge is abstract and it either does not require measurement or cannot be measured because the reality they represent can only be approximated. Thus the decision making may be associated with uncertainty. Because the knowledge of these aspects is gained through observation combined with interpretative understanding, the uncertainty usually occurs because information is not clearly described or only described by partial and imprecise evidence, such as ability of personalized service and capability of disaster recovery [35]. ER rules can make full use of qualitative knowledge and quantitative data and can also express various types of uncertainties. In this section, a new reliability assessment architecture including four aspects is established, and then the ER rule is used to assess the reliability of the cloud computing platform.

3.2. The Basic Attributes of Cloud Computing Platform Reliability. A four-level reliability attributes structure of cloud computing platform is established, including both quantitative data and qualitative knowledge. Reliability of cloud computing platform is categorized in 4 distinct aspects: network reliability, hardware reliability, software reliability, and service reliability. For each attribute, the symbol "r" is numerically labeled according to the hierarchy, while the symbol "w" is valued as the weight which is one of the important parameters in the process of ER rule. In the aspect of the network reliability, all the attributes are measured as the quantitative data. In particular, considering the significance of network attack frequency in assessing the network reliability, the 4th level attributes are added. In the aspect of hardware reliability, most attributes are scalability of the cloud system and fault tolerance rate of the cloud platform. Scalability is the ability of cloud system to continue to run well when it is changed in volume in order to meet a requirement. The controllability of access terminal is the only qualitative attributes which is defined as an ability of access terminal in aspect of software reliability. Service reliability is an assessment for the Quality of Service (QoS); thus most of attributes are defined as qualitative knowledge. In this paper, we introduce 4-grade assessment levels as the frame of discernment of the reliability assessment model. The basic attributes of cloud computing platform reliability are shown in Table 1.

3.3. The Assessment Grades of the Attributes. In this paper, the assessment grades of the attributes of cloud computing platform are established. For the quantitative attributes, the data range is divided into 4 grades, determined as "excellent, good, common, and bad." The setting up of the interval of quantitative attributes is obtained by the expert experience and practical investigation, which ensure the data precision and traceability. For the qualitative attributes, the grade of assessment can be determined by experts according to the experience or investigation. The assessment rules can be established through the assessment grades, as shown in Table 2.

3.4. The Belief Degree of the Attributes. When the value of an attribute of the assessment is determined, the corresponding belief degree should be calculated in order to obtain the basic probability mass in ((2)-(1)), as shown in the following:

$$\begin{aligned}
\beta_{i,j} &= \frac{R_{i,j+1} - U\left(r_i\right)}{R_{i,j+1} - R_{i,j}} \quad \left(R_{i,j} \le U\left(r_i\right) \le R_{i,j+1}\right), \\
\beta_{i,j+1} &= 1 - \beta_{i,j}, \\
\beta_{i,k} &= 0 \quad \left(k = 1, \ldots, N,\ k \neq j, j+1\right),
\end{aligned} \tag{8}$$

where $U(r_i)$ denotes the value of the attribute r_i and $R_{i,j}$ denotes the reference value of the jth grade for attribute r_i.

4. Case Study

In order to illustrate the detailed process of the reliability assessment, a case which uses the proposed reliability assessment architecture and ER rule to assess the reliability of an actual cloud computing platform is studied in this section.

Table 1: The basic attributes of cloud computing platform reliability.

Reliability	1st level	2nd level	3rd level	4th level
Reliability of cloud computing platform (R)	Network reliability (r_1), ($w_1 = 0.25$)	Network equipment reliability (r_{11}), ($w_{11} = 0.5$)	MTTF of central units (r_{111}), ($w_{111} = 0.6$)	
			MTTF of common units (r_{112}), ($w_{112} = 0.4$)	
		Network system reliability (r_{12}), ($w_{12} = 0.5$)	Frequency of network attack (r_{121}), ($w_{121} = 0.6$)	Dos attack (r_{1211}), ($w_{1211} = 0.4$)
				Probe attack (r_{1212}), ($w_{1212} = 0.3$)
				U2R attack (r_{1213}), ($w_{1213} = 0.2$)
				R2L attack (r_{1214}), ($w_{1214} = 0.1$)
			Vulnerability of network system (r_{122}), ($w_{122} = 0.4$)	
Reliability of cloud computing platform (R)	Hardware reliability (r_2), ($w_2 = 0.25$)	Cloud infrastructure reliability (r_{21}), ($w_{21} = 0.5$)	MTTF of cloud server (r_{211}), ($w_{211} = 0.4$)	
			MTTF of virtual machine (r_{212}), ($w_{212} = 0.3$)	
			Scalability of cloud system (r_{213}), ($w_{213} = 0.3$)	
		Cloud platform reliability (r_{22}), ($w_{22} = 0.5$)	MTTF of cloud platform I/O (r_{221}), ($w_{221} = 0.5$)	
			Fault tolerant rate of cloud platform (r_{222}), ($w_{222} = 0.5$)	
Reliability of cloud computing platform (R)	Software reliability (r_3), ($w_3 = 0.25$)	Application reliability (r_{31}), ($w_{31} = 0.4$)	Failure rate of web application (r_{311}), ($w_{311} = 0.5$)	
			Controllability of access terminal (r_{312}), ($w_{312} = 0.5$)	
		Cloud operating system reliability (r_{32}), ($w_{32} = 0.6$)	MTTF of cloud operating system (r_{321}), ($w_{321} = 0.6$)	
			Vulnerability of cloud operating system (r_{322}), ($w_{322} = 0.4$)	
Reliability of cloud computing platform (R)	Service reliability (r_4), ($w_4 = 0.25$)	Performance reliability (r_{41}), ($w_{41} = 0.3$)	Capacity of throughput (r_{411}), ($w_{411} = 0.2$)	
			Average response time of the service (r_{412}), ($w_{412} = 0.3$)	
			Failure rate of service (r_{413}), ($w_{413} = 0.2$)	
			MTTR of service (r_{414}), ($w_{414} = 0.3$)	
		Service availability (r_{42}), ($w_{42} = 0.3$)	Completeness of service (r_{421}), ($w_{421} = 0.3$)	
			Access success rate (r_{422}), ($w_{422} = 0.4$)	
			Ability of personalized service (r_{423}), ($w_{423} = 0.3$)	
		Service stability (r_{43}), ($w_{43} = 0.4$)	Service continuity and stability (r_{431}), ($w_{431} = 0.4$)	
			Disaster recovery capability (r_{432}), ($w_{432} = 0.4$)	
			Service transparency (r_{433}), ($w_{433} = 0.2$)	

Table 2: The reference values of the assessment grades.

	E	G	C	B
r_{111}	90/kh	60/kh	30/kh	0/kh
r_{112}	60/kh	40/kh	20/kh	0/kh
r_{1211}	0 times/h	10 times/h	20 times/h	30 times/h
r_{1212}	0 times/h	10 times/h	20 times/h	30 times/h
r_{1213}	0 times/h	10 times/h	20 times/h	30 times/h
r_{1214}	0 times/h	10 times/h	20 times/h	30 times/h
r_{122}	0 times/mon	50 times/mon	100 times/mon	150 times/mon
r_{211}	30/kh	20/kh	10/kh	0/kh
r_{212}	15/kh	10/kh	5/kh	0/kh
r_{213}			Given by experts	
r_{221}	9/kh	6/kh	3/kh	0/kh
r_{222}			Given by experts	
r_{311}	0/fit	20/fit	40/fit	60/fit
r_{312}			Given by experts	
r_{321}	9/kh	6/kh	3/kh	0/kh
r_{322}	0 times/mon	200 times/mon	400 times/mon	600 times/mon
r_{411}			Given by experts	
r_{412}	0/ms	100/ms	200/ms	300/ms
r_{413}	0/fit	30/fit	60/fit	90/fit
r_{414}	0/h	10/h	20/h	30/h
r_{421}			Given by experts	
r_{422}	100%	90%	80%	70%
r_{423}			Given by experts	
r_{431}			Given by experts	
r_{432}			Given by experts	
r_{433}			Given by experts	

4.1. The Actual Cloud Computing Platform. In this case, a cloud computing platform of Innovation Lab in HUST is investigated. The architecture of the platform can be divided into 4 layers and 2 centers. The layers consist of cloud infrastructure layer, cloud platform layer, cloud bus layer, and cloud application layer. The centers include cloud management center and cloud data center. The cloud infrastructure layer consists of hardware level and virtual level, including computers, servers, storage units, and network units. Employing technology of virtualization, the virtual level is a solution of IaaS service, which forms a virtual cluster from underlying hardware resources. The cloud platform layer includes service level and computing level. The purpose of the service layer is to provide the basic data or information to the upper layer through the processing of the form of service. The computing level includes distributed computing engine, concurrent computing engine, and utility computing engine. The cloud bus design includes service adapter level, bus level, and application adapter level. The service adapter provides an adapter for the service and bus communication in the cloud platform. The cloud bus is a kind of basic structure that interacts with the service through the service adapter. The application adapter provides an adapter for application software and bus communication in the cloud platform. The cloud application layer can be categorized into three types, such as inner application, external application, and platform of secondary development interface.

The cloud management center includes supercomputing center and security center. The supercomputing center can achieve rapid deployment, automation installation, and upgrade. The security center is responsible for user registration, user authentication, access control, and so forth. The cloud data center is in charge of processing and storage of data in the cloud platform. The architecture of the platform is shown in Figure 1.

4.2. The Process of the Reliability Assessment. The corresponding attributes described in Table 1 are collected from the cloud computing platform described above. Taking the frequency of network attack (r_{121}) as an example, assume that Dos attack $r_{1211} = 2$ times/h, Probe attack $r_{1212} = 13$ times/h, U2R attack $r_{1213} = 8$ times/h, and R2L attack $r_{1214} = 21$ times/h. Thus, the belief degrees of these attributes which are assessed to a grade can be calculated by (8):

$$\beta_{1211,1} = 0.8,$$

$$\beta_{1211,2} = 0.2,$$

$$\beta_{1211,3} = 0,$$

$$\beta_{1211,4} = 0,$$

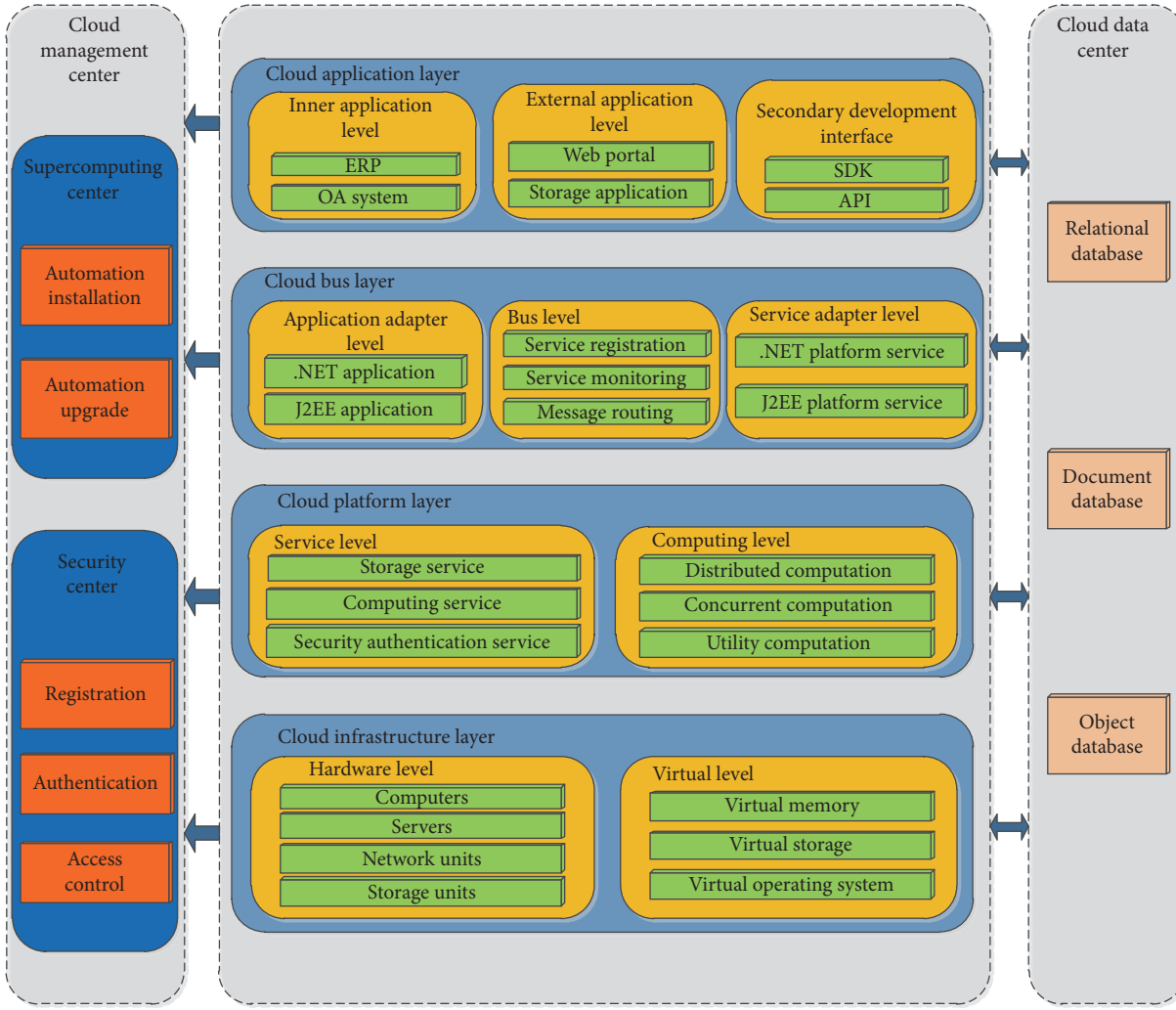


Figure 1: The architecture of the cloud platform.

$$\begin{aligned}
&\beta_{1212,1} = 0.7,\\
&\beta_{1212,2} = 0.3,\\
&\beta_{1212,3} = 0,\\
&\beta_{1212,4} = 0,\\
&\beta_{1213,1} = 0.2,\\
&\beta_{1213,2} = 0.8,\\
&\beta_{1213,3} = 0,\\
&\beta_{1213,4} = 0,\\
&\beta_{1214,1} = 0.9,\\
&\beta_{1214,2} = 0.1,\\
&\beta_{1214,3} = 0,\\
&\beta_{1214,4} = 0.
\end{aligned} \tag{9}$$

The above belief degrees should be converted into the basic probability mass by (1)–(4) and the corresponding weights given in Table 1:

$$\begin{aligned}
&P_{1211,1} = 0.32,\\
&P_{1211,2} = 0.08,\\
&P_{1211,3} = 0,\\
&P_{1211,4} = 0,\\
&P_{1211,\Theta} = 0.6,\\
&\overline{P}_{1211,\Theta} = 0.6,\\
&\widetilde{P}_{1211,\Theta} = 0,\\
&P_{1212,1} = 0.21,\\
&P_{1212,2} = 0.09,\\
&P_{1212,3} = 0,
\end{aligned}$$

$$
\begin{aligned}
P_{1212,4} &= 0,\\
P_{1212,\Theta} &= 0.7,\\
\overline{P}_{1212,\Theta} &= 0.7,\\
\widetilde{P}_{1212,\Theta} &= 0,\\
P_{1213,1} &= 0.04,\\
P_{1213,2} &= 0.16,\\
P_{1213,3} &= 0,\\
P_{1213,4} &= 0,\\
P_{1213,\Theta} &= 0.8,\\
\overline{P}_{1213,\Theta} &= 0.8,\\
\widetilde{P}_{1213,\Theta} &= 0,\\
P_{1214,1} &= 0.09,\\
P_{1214,2} &= 0.01,\\
P_{1214,3} &= 0,\\
P_{1214,4} &= 0,\\
P_{1214,\Theta} &= 0.9,\\
\overline{P}_{1214,\Theta} &= 0.9,\\
\widetilde{P}_{1214,\Theta} &= 0.
\end{aligned}
\tag{10}
$$

Then, in order to integrate the above probability masses and get the upper attribute, $K_{I(i+1)} = K_{I(1212)}$ should be calculated at first:

$$
\begin{aligned}
K_{I(1212)} &= \frac{1}{1 - \sum_{k=1}^{N} \sum_{j=1, j\neq k}^{N} P_{I(1211),k} P_{1212,j}}\\
&= \frac{1}{1 - 0.32 \times 0.09 + 0.08 \times 0.21} = 1.0478.
\end{aligned}
\tag{11}
$$

Thus, the integration masses can be obtained by (5):

$$
\begin{aligned}
P_{I(1212),1} &= K_{I(1212)} \left[P_{I(1211),1} P_{1212,1} + P_{I(1211),1} P_{1212,\Theta} + P_{I(1211),\Theta} P_{1212,1}\right] = 1.0478 \times [0.32 \times 0.21 + 0.32 \times 0.7 + 0.6 \times 0.21] = 0.4371,\\
P_{I(1212),2} &= K_{I(1212)} \left[P_{I(1211),2} P_{1212,2} + P_{I(1211),2} P_{1212,\Theta} + P_{I(1211),\Theta} P_{1212,2}\right] = 1.0478 \times [0.08 \times 0.09 + 0.08 \times 0.7 + 0.6 \times 0.09] = 0.1228,\\
P_{I(1212),3} &= K_{I(1212)} \left[P_{I(1211),3} P_{1212,3} + P_{I(1211),3} P_{1212,\Theta} + P_{I(1211),\Theta} P_{1212,3}\right] = 1.0478 \times [0 \times 0 + 0 \times 0.7 + 0.6 \times 0] = 0,\\
P_{I(1212),4} &= K_{I(1212)} \left[P_{I(1211),4} P_{1212,4} + P_{I(1211),4} P_{1212,\Theta} + P_{I(1211),\Theta} P_{1212,4}\right] = 1.0478 \times [0 \times 0 + 0 \times 0.7 + 0.6 \times 0] = 0,\\
\widetilde{P}_{I(1212),\Theta} &= K_{I(1212)} \left[\widetilde{P}_{I(1211),\Theta} \widetilde{P}_{1212,\Theta} + \widetilde{P}_{I(1211),\Theta} \overline{P}_{1212,\Theta} + \overline{P}_{I(1211),\Theta} \widetilde{P}_{1212,\Theta}\right] = 1.0478 \times [0 \times 0 + 0 \times 0.7 + 0.6 \times 0] = 0,\\
\overline{P}_{I(1212),\Theta} &= K_{I(1212)} \left[\overline{P}_{I(1211),\Theta} \overline{P}_{1212,\Theta}\right] = 1.0478 \times [0.6 \times 0.7] = 0.4401,\\
P_{I(1212),\Theta} &= \overline{P}_{I(1212),\Theta} + \widetilde{P}_{I(1212),\Theta} = 0.4401.
\end{aligned}
\tag{12}
$$

Next, in order to integrate the third probability mass, $K_{I(1213)}$ should be calculated:

$$
\begin{aligned}
K_{I(1213)} &= \frac{1}{1 - \sum_{k=1}^{N} \sum_{j=1, j\neq k}^{N} P_{I(1212),k} P_{1213,j}}\\
&= \frac{1}{1 - 0.4371 \times 0.16 + 0.1228 \times 0.04}\\
&= 1.0809.
\end{aligned}
\tag{13}
$$

Then, the new integration masses can be obtained:

$$
\begin{aligned}
P_{I(1213),1} &= K_{I(1213)} \left[P_{I(1212),1} P_{1213,1} + P_{I(1212),1} P_{1213,\Theta} + P_{I(1212),\Theta} P_{1213,1}\right] = 1.0809 \times [0.4371 \times 0.04 + 0.4371 \times 0.8 + 0.4401 \times 0.04] = 0.4159,\\
P_{I(1213),2} &= K_{I(1213)} \left[P_{I(1212),2} P_{1213,2} + P_{I(1212),2} P_{1213,\Theta} + P_{I(1212),\Theta} P_{1213,2}\right] = 1.0809 \times [0.1228 \times 0.16 + 0.1228 \times 0.8 + 0.4401 \times 0.16] = 0.2035,\\
P_{I(1213),3} &= K_{I(1213)} \left[P_{I(1212),3} P_{1213,3} + P_{I(1212),3} P_{1213,\Theta} + P_{I(1212),\Theta} P_{1213,3}\right] = 1.0809 \times [0 \times 0 + 0 \times 0.8 + 0.4401 \times 0] = 0,\\
P_{I(1213),4} &= K_{I(1213)} \left[P_{I(1212),4} P_{1213,4} + P_{I(1212),4} P_{1213,\Theta} + P_{I(1212),\Theta} P_{1213,4}\right] = 1.0809 \times [0 \times 0 + 0 \times 0.8 + 0.4401 \times 0] = 0,
\end{aligned}
$$

$$\begin{aligned}
\widetilde{P}_{I(1213),\Theta} &= K_{I(1213)}\left[\widetilde{P}_{I(1212),\Theta}\widetilde{P}_{1213,\Theta}\right. \\
&\quad \left. + \widetilde{P}_{I(1212),\Theta}\overline{P}_{1213,\Theta} + \overline{P}_{I(1212),\Theta}\widetilde{P}_{1213,\Theta}\right] = 1.0809 \\
&\quad \times [0 \times 0 + 0 \times 0.8 + 0.4401 \times 0] = 0, \\
\overline{P}_{I(1213),\Theta} &= K_{I(1213)}\left[\overline{P}_{I(1212),\Theta}\overline{P}_{1213,\Theta}\right] = 1.0809 \\
&\quad \times [0.4401 \times 0.8] = 0.3806, \\
P_{I(1213),\Theta} &= \overline{P}_{I(1212),\Theta} + \widetilde{P}_{I(1212),\Theta} = 0.3806.
\end{aligned} \tag{14}$$

Then, $K_{I(1214)}$ should be calculated:

$$\begin{aligned}
K_{I(1214)} &= \frac{1}{1 - \sum_{k=1}^{N}\sum_{j=1,j\neq k}^{N} P_{I(1213),k}P_{1214,j}} \\
&= \frac{1}{1 - 0.4159 \times 0.01 + 0.2035 \times 0.09} \\
&= 1.0230.
\end{aligned} \tag{15}$$

Thus, the new integration masses can be obtained:

$$\begin{aligned}
P_{I(1214),1} &= K_{I(1214)}\left[P_{I(1213),1}P_{1214,1} + P_{I(1213),1}P_{1214,\Theta}\right. \\
&\quad \left. + P_{I(1213),\Theta}P_{1214,1}\right] = 1.0230 \times [0.4159 \times 0.09 \\
&\quad + 0.4159 \times 0.9 + 0.3806 \times 0.09] = 0.4563, \\
P_{I(1214),2} &= K_{I(1214)}\left[P_{I(1213),2}P_{1214,2} + P_{I(1213),2}P_{1214,\Theta}\right. \\
&\quad \left. + P_{I(1213),\Theta}P_{1214,2}\right] = 1.0230 \times [0.2035 \times 0.01 \\
&\quad + 0.2035 \times 0.9 + 0.3806 \times 0.01] = 0.1933, \\
P_{I(1214),3} &= K_{I(1214)}\left[P_{I(1213),3}P_{1214,3} + P_{I(1213),3}P_{1214,\Theta}\right. \\
&\quad \left. + P_{I(1213),\Theta}P_{1214,3}\right] = 1.0230 \times [0 \times 0 + 0 \times 0.9 \\
&\quad + 0.3806 \times 0] = 0, \\
P_{I(1214),4} &= K_{I(1214)}\left[P_{I(1213),4}P_{1214,4} + P_{I(1213),4}P_{1214,\Theta}\right. \\
&\quad \left. + P_{I(1213),\Theta}P_{1214,4}\right] = 1.0230 \times [0 \times 0 + 0 \times 0.9 \\
&\quad + 0.3806 \times 0] = 0, \\
\widetilde{P}_{I(1214),\Theta} &= K_{I(1214)}\left[\widetilde{P}_{I(1213),\Theta}\widetilde{P}_{1214,\Theta}\right. \\
&\quad \left. + \widetilde{P}_{I(1213),\Theta}\overline{P}_{1214,\Theta} + \overline{P}_{I(1213),\Theta}\widetilde{P}_{1214,\Theta}\right] = 1.0230 \\
&\quad \times [0 \times 0 + 0 \times 0.9 + 0.3806 \times 0] = 0, \\
\overline{P}_{I(1214),\Theta} &= K_{I(1214)}\left[\overline{P}_{I(1213),\Theta}\overline{P}_{1214,\Theta}\right] = 1.0230 \\
&\quad \times [0.3806 \times 0.9] = 0.3504, \\
P_{I(1212),\Theta} &= \overline{P}_{I(1212),\Theta} + \widetilde{P}_{I(1212),\Theta} = 0.3504.
\end{aligned} \tag{16}$$

Finally, the belief degrees of frequency of network attack (r_{121}) can be calculated by (7):

$$\begin{aligned}
\widehat{\beta}_1 &= \frac{P_{I(1214),1}}{1 - \overline{P}_{I(1214),\Theta}} = \frac{0.4563}{1 - 0.3504} = \frac{0.4563}{0.6496} = 0.7024, \\
\widehat{\beta}_2 &= \frac{P_{I(1214),2}}{1 - \overline{P}_{I(1214),\Theta}} = \frac{0.1933}{1 - 0.3504} = \frac{0.1933}{0.6496} = 0.2976, \\
\widehat{\beta}_3 &= \frac{P_{I(1214),3}}{1 - \overline{P}_{I(1214),\Theta}} = \frac{0}{1 - 0.3504} = 0, \\
\widehat{\beta}_4 &= \frac{P_{I(1214),4}}{1 - \overline{P}_{I(1214),\Theta}} = \frac{0}{1 - 0.3504} = 0, \\
\widehat{\beta}_\Theta &= \frac{\widetilde{P}_{I(1214),\Theta}}{1 - \overline{P}_{I(1214),\Theta}} = \frac{0}{1 - 0.3504} = 0.
\end{aligned} \tag{17}$$

Through the above conclusions, when $r_{1211} = 2$ times/h, $r_{1212} = 13$ times/h, $r_{1213} = 8$ times/h, and $r_{1214} = 21$ times/h, the final belief degrees of r_{121} can be expressed by $\{E(70.24\%), G(29.76\%), C(0\%), B(0\%)\}$, which means that the frequency of network attack (r_{121}) can be considered as "good."

The other attributes in Table 1 can be integrated by the method described in above process, and the final conclusion of reliability assessment of cloud computing platform can be obtained.

5. Conclusion

Considering the complex attributes, this paper first employed ER rule to make a comprehensive assessment for the reliability of cloud computing platform. ER rule has a good performance in multiple attribute decision making. It can process the evidence with high conflict and complete conflict and then make a more accurate assessment result. Moreover, a new reliability assessment structure of cloud computing for multiple attribute decision making is proposed in this paper. The 4-level reliability attributes mainly include four aspects, where the views of cloud computing servers and users are both taken into account, and the actual situation of reliability in cloud computing platform can be completely expressed. In conclusion, the innovation points of this paper can be concluded: (1) ER rule is first employed to establish the reliability assessment model of the cloud computing platform in this paper; (2) a new reliability assessment architecture including four aspects is proposed. Thus, the above two innovative researches cause a good result for reliability assessment in the complex dynamic cloud computing platform.

Competing Interests

The authors declare that they have no competing interests.

References

[1] W. Tian, Y. Zhao, M. Xu, Y. Zhong, and X. Sun, "A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 153–161, 2015.

[2] M. Abdullahi, M. A. Ngadi, and S. M. Abdulhamid, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment," *Future Generation Computer Systems*, vol. 56, pp. 640–650, 2016.

[3] X. L. Xu, L. Cao, X. Wang, X. Xu, L. Cao, and X. Wang, "Resource pre-allocation algorithms for low-energy task scheduling of cloud computing," *Journal of Systems Engineering & Electronics*, vol. 27, no. 2, pp. 457–469, 2016.

[4] Y. Sun, J. White, S. Eade, and D. C. Schmidt, "ROAR: a QoS-oriented modeling framework for automated cloud resource allocation and optimization," *Journal of Systems and Software*, vol. 116, pp. 146–161, 2016.

[5] A. I. Awad, N. A. El-Hefnawy, and H. M. Abdel_Kader, "Enhanced particle swarm optimization for task scheduling in cloud computing environments," *Procedia Computer Science*, vol. 65, pp. 920–929, 2015.

[6] J. Wang, B. Gong, H. Liu, and S. Li, "Multidisciplinary approaches to artificial swarm intelligence for heterogeneous computing and cloud scheduling," *Applied Intelligence*, vol. 43, no. 3, pp. 662–675, 2015.

[7] P. Qi and L. Li, "A fault recovery–based scheduling algorithm for cloud service reliability," *Security and Communication Networks*, vol. 8, no. 5, pp. 703–714, 2015.

[8] S. H. Albakri, B. Shanmugam, G. N. Samy, N. B. Idris, and A. Ahmed, "Security risk assessment framework for cloud computing environments," *Security and Communication Networks*, vol. 7, no. 11, pp. 2114–2124, 2014.

[9] N. Ahmed and A. Abraham, "Neuro-fuzzy model for assessing risk in cloud computing environment," in *Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015*, vol. 427 of *Advances in Intelligent Systems and Computing*, pp. 137–145, Springer International Publishing, Basel, Switzerland, 2016.

[10] H. Khazaei, J. Mišić, V. B. Mišić, and S. Rashwand, "Analysis of a pool management scheme for cloud computing centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 849–861, 2013.

[11] X. Qiu, Y. Dai, Y. Xiang, and L. Xing, "A hierarchical correlation model for evaluating reliability, performance, and power consumption of a cloud service," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 401–412, 2016.

[12] B. Larsen, P. Ingwersen, and B. Lund, "Data fusion according to the principle of polyrepresentation," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 4, pp. 646–654, 2009.

[13] J. C. de Winter and D. Dodou, "Factor recovery by principal axis factoring and maximum likelihood factor analysis as a function of factor pattern and sample size," *Journal of Applied Statistics*, vol. 39, no. 4, pp. 695–710, 2012.

[14] M. Sundaram, M. J. Smith, D. A. Revicki, L.-A. Miller, S. Madhavan, and G. Hobbs, "Estimation of a valuation function for a diabetes mellitus-specific preference-based measure of health: the diabetes utility index[5]," *PharmacoEconomics*, vol. 28, no. 3, pp. 201–216, 2010.

[15] Z. Ruo-Xin, X.-J. Cui, S.-J. Gong, H.-K. Ren, and K. Chen, "Model for cloud computing security assessment based on AHP and FCE," in *Proceedings of the 9th International Conference on Computer Science and Education (ICCCSE '14)*, pp. 197–204, August 2014.

[16] N. Badie, A. R. C. Hussin, and A. H. Lashkari, "Cloud computing data center adoption factors validity by fuzzy AHP," *International Journal of Computational Intelligence Systems*, vol. 8, no. 5, pp. 854–873, 2015.

[17] Y. Tamura, M. Kawakami, and S. Yamada, "Reliability modeling and analysis for open source cloud computing," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 227, no. 2, pp. 179–186, 2013.

[18] D. Marudhadevi, V. N. Dhatchayani, and V. S. S. Sriram, "A trust evaluation model for cloud computing using service level agreement," *The Computer Journal*, vol. 58, no. 10, pp. 2225–2232, 2014.

[19] Z. Pei, "Intuitionistic fuzzy variables: concepts and applications in decision making," *Expert Systems with Applications*, vol. 42, no. 22, pp. 9033–9045, 2015.

[20] Z. Xuejie, W. Zhijian, and X. Feng, "Reliability evaluation of cloud computing systems using hybrid methods," *Intelligent Automation & Soft Computing*, vol. 19, no. 2, pp. 165–174, 2013.

[21] M. Khaki, I. Yusoff, and N. Islami, "Application of the artificial neural network and neuro-fuzzy system for assessment of groundwater quality," *Clean—Soil, Air, Water*, vol. 43, no. 4, pp. 551–560, 2015.

[22] O. Hussain, H. Dong, and J. Singh, "Semantic similarity model for risk assessment in forming cloud computing SLAs," in *On the Move to Meaningful Internet Systems, OTM 2010: Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25–29, 2010, Proceedings, Part II*, vol. 6427 of *Lecture Notes in Computer Science*, pp. 843–860, Springer, Berlin, Germany, 2010.

[23] J. M. Garibaldi and T. Ozen, "Uncertain fuzzy reasoning: a case study in modelling expert decision making," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 16–30, 2007.

[24] X. Fan and M. J. Zuo, "Fault diagnosis of machines based on D–S evidence theory. Part 1: D–S evidence theory and its improvement," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 366–376, 2006.

[25] R. Sun, H.-Z. Huang, and Q. Miao, "Improved information fusion approach based on D-S evidence theory," *Journal of Mechanical Science and Technology*, vol. 22, no. 12, pp. 2417–2425, 2008.

[26] Y.-M. Wang, J.-B. Yang, D.-L. Xu, and K.-S. Chin, "The evidential reasoning approach for multiple attribute decision analysis using interval belief degrees," *European Journal of Operational Research*, vol. 175, no. 1, pp. 35–66, 2006.

[27] Z.-G. Zhou, F. Liu, L.-C. Jiao et al., "An evidential reasoning based model for diagnosis of lymph node metastasis in gastric cancer," *BMC Medical Informatics and Decision Making*, vol. 13, no. 1, article 123, 2013.

[28] Z.-G. Liu, Q. Pan, and J. Dezert, "A new belief-based K-nearest neighbor classification method," *Pattern Recognition*, vol. 46, no. 3, pp. 834–844, 2013.

[29] Z.-G. Liu, Q. Pan, J. Dezert, and A. Martin, "Adaptive imputation of missing values for incomplete pattern classification," *Pattern Recognition*, vol. 52, pp. 85–95, 2016.

[30] Z. Zhou, M. Folkert, N. Cannon et al., "Predicting distant failure in early stage NSCLC treated with SBRT using clinical parameters," *Radiotherapy & Oncology*, vol. 119, no. 3, pp. 501–504, 2016.

[31] Z.-G. Zhou, F. Liu, L.-L. Li et al., "A cooperative belief rule based decision support system for lymph node metastasis diagnosis in gastric cancer," *Knowledge-Based Systems*, vol. 85, pp. 62–70, 2015.

[32] Z.-G. Liu, Q. Pan, J. Dezert, and G. Mercier, "Credal c-means clustering method based on belief functions," *Knowledge-Based Systems*, vol. 74, pp. 119–132, 2015.

[33] J.-B. Yang and D.-L. Xu, "Evidential reasoning rule for evidence combination," *Artificial Intelligence*, vol. 205, pp. 1–29, 2013.

[34] Z. J. Zhou, C. H. Hu, G. Y. Hu, X. X. Han, B. C. Zhang, and Y. W. Chen, "Hidden behavior prediction of complex systems under testing influence based on semiquantitative information and belief rule base," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2371–2386, 2015.

[35] Z. J. Zhou, L. L. Chang, C. H. Hu, X. X. Han, and Z. G. Zhou, "A new BRB-ER-based model for assessing the lives of products using both failure data and expert knowledge," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 46, no. 11, pp. 1529–1543, 2016.