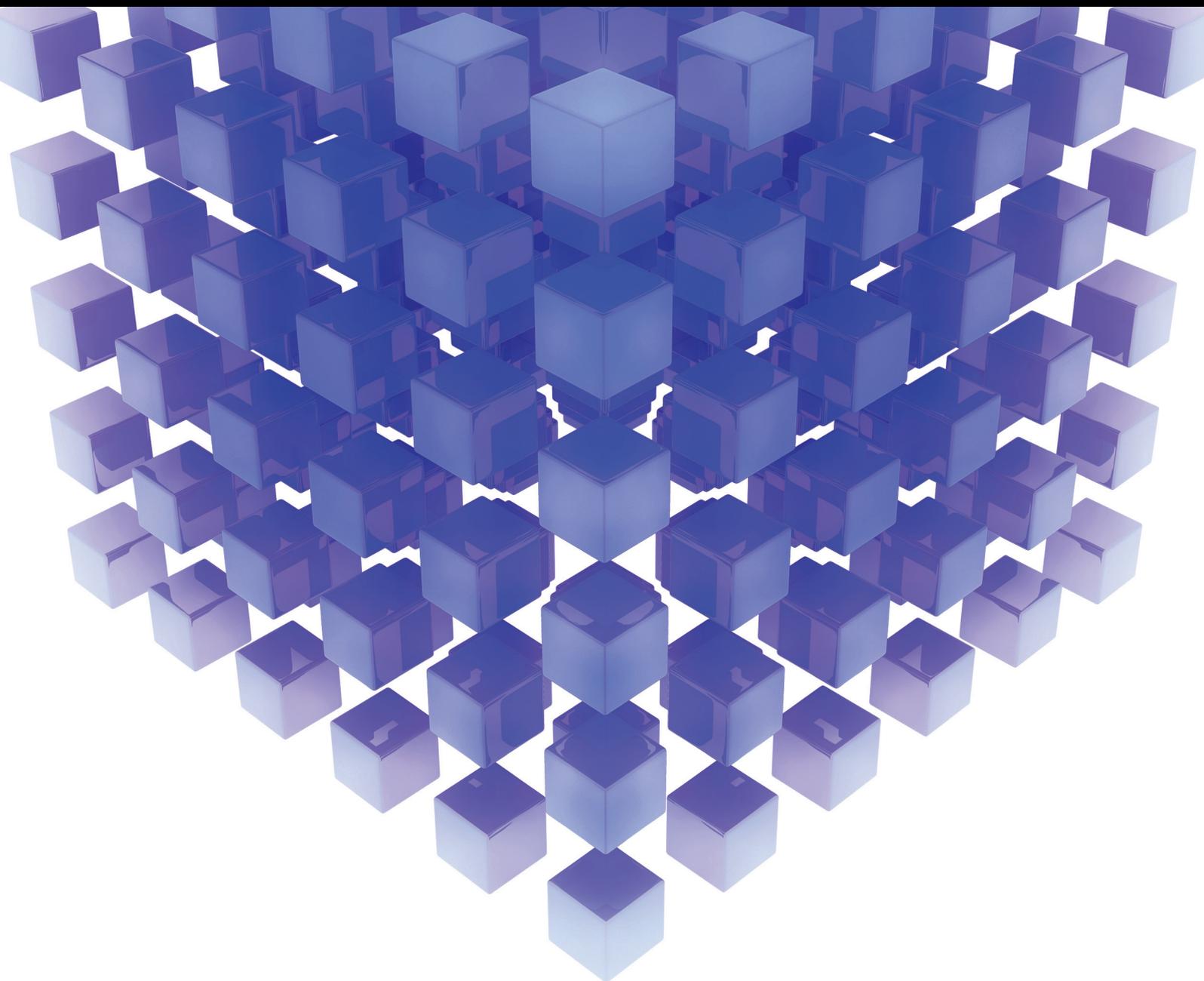


Mathematical Problems in Engineering

# Fault Detection, Estimation/ Reconstruction, and Fault-Tolerant Control: Theory and Practice

Guest Editors: Wen Chen, Afef Fekih, and Zehui Mao





---

**Fault Detection, Estimation/Reconstruction,  
and Fault-Tolerant Control:  
Theory and Practice**

Mathematical Problems in Engineering

---

**Fault Detection, Estimation/Reconstruction,  
and Fault-Tolerant Control:  
Theory and Practice**

Guest Editors: Wen Chen, Afef Fekih, and Zehui Mao



---

Copyright © 2016 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in “Mathematical Problems in Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Editorial Board

- Mohamed Abd El Aziz, Egypt  
Farid Abed-Meraim, France  
José Ángel Acosta, Spain  
Paolo Addresso, Italy  
Claudia Adduce, Italy  
Ramesh Agarwal, USA  
Juan C. Agüero, Australia  
Ricardo Aguilar-López, Mexico  
Tarek Ahmed-Ali, France  
Hamid Akbarzadeh, Canada  
Muhammad N. Akram, Norway  
Mohammad-Reza Alam, USA  
Salvatore Alfonzetti, Italy  
Francisco Alhama, Spain  
Mohammad D. Aliyu, Canada  
Juan A. Almendral, Spain  
Lionel Amodeo, France  
Sebastian Anita, Romania  
Renata Archetti, Italy  
Felice Arena, Italy  
Sabri Arik, Turkey  
Alessandro Arsie, USA  
Eduardo Artioli, Italy  
Fumihiko Ashida, Japan  
Hassan Askari, Canada  
Mohsen Asle Zaeem, USA  
Romain Aubry, USA  
Matteo Aureli, USA  
Francesco Aymerich, Italy  
Seungik Baek, USA  
Khaled Bahlali, France  
Laurent Bako, France  
Stefan Balint, Romania  
Alfonso Banos, Spain  
Roberto Baratti, Italy  
Martino Bardi, Italy  
Azeddine Beghdadi, France  
Denis Benasciutti, Italy  
Abdel-Hakim Bendada, Canada  
Ivano Benedetti, Italy  
Elena Benvenuti, Italy  
Jamal Berakdar, Germany  
Michele Betti, Italy  
Jean-Charles Beugnot, France  
Simone Bianco, Italy
- Gennaro N. Bifulco, Italy  
David Bigaud, France  
Antonio Bilotta, Italy  
Jonathan N. Blakely, USA  
Paul Bogdan, USA  
Alberto Borboni, Italy  
Paolo Boscariol, Italy  
Daniela Boso, Italy  
Guillermo Botella-Juan, Spain  
Abdel-Ouahab Boudraa, France  
Fabio Bovenga, Italy  
Francesco Braghin, Italy  
Michael J. Brennan, UK  
Maurizio Brocchini, Italy  
Julien Bruchon, France  
Michele Brun, Italy  
Javier Buldú, Spain  
Erik Burman, UK  
Tito Busani, USA  
Raquel Caballero-Águila, Spain  
Pierfrancesco Cacciola, UK  
Salvatore Caddemi, Italy  
Jose E. Capilla, Spain  
E. Javier Cara, Spain  
Ana Carpio, Spain  
Carmen Castillo, Spain  
Inmaculada T. Castro, Spain  
Gabriele Cazzulani, Italy  
Luis Cea, Spain  
Miguel E. Cerrolaza, Spain  
M. Chadli, France  
Gregory Chagnon, France  
Ching-Ter Chang, Taiwan  
Michael J. Chappell, UK  
Kacem Chehdi, France  
Peter N. Cheimets, USA  
Chunlin Chen, China  
Xinkai Chen, Japan  
Francisco Chicano, Spain  
Hung-Yuan Chung, Taiwan  
Joaquim Ciurana, Spain  
John D. Clayton, USA  
Giuseppina Colicchio, Italy  
Mario Cools, Belgium  
Sara Coppola, Italy
- Jean-Pierre Corriou, France  
Juan Carlos Cortés López, Spain  
Carlo Cosentino, Italy  
Paolo Crippa, Italy  
Andrea Crivellini, Italy  
Erik Cuevas, Mexico  
Peter Dabnichki, Australia  
Luca D'Acerno, Italy  
Weizhong Dai, USA  
Andrea Dall'Asta, Italy  
Purushothaman Damodaran, USA  
Farhang Daneshmand, Canada  
Fabio De Angelis, Italy  
Stefano de Miranda, Italy  
Filippo de Monte, Italy  
Maria do Rosário de Pinho, Portugal  
Xavier Delorme, France  
Frédéric Demoly, France  
Luca Deseri, USA  
Angelo Di Egidio, Italy  
Ramón I. Diego, Spain  
Yannis Dimakopoulos, Greece  
Zhengtao Ding, UK  
Mohamed Djemai, France  
Manuel Doblaré, Spain  
Alexandre B. Dolgui, France  
Florent Duchaine, France  
George S. Dulikravich, USA  
Bogdan Dumitrescu, Romania  
Horst Ecker, Austria  
Ahmed El Hajjaji, France  
Fouad Erchiqui, Canada  
Anders Eriksson, Sweden  
R. Emre Erkmen, Australia  
Andrea L. Facci, Italy  
Giovanni Falsone, Italy  
Hua Fan, China  
Yann Favennec, France  
Giuseppe Fedele, Italy  
Roberto Fedele, Italy  
Jose R. Fernandez, Spain  
Jesus M. Fernandez Oro, Spain  
Eric Feulvarch, France  
Simme Douwe Flapper, Netherlands  
Thierry Floquet, France

Eric Florentin, France  
Jose M. Framinan, Spain  
Francesco Franco, Italy  
Elisa Francomano, Italy  
Mario L. Fravolini, Italy  
Leonardo Freitas, UK  
Tomonari Furukawa, USA  
Mohamed Gadala, Canada  
Matteo Gaeta, Italy  
Mauro Gaggero, Italy  
Zoran Gajic, Iraq  
Erez Gal, Israel  
Ugo Galvanetto, Italy  
Akemi Gálvez, Spain  
Rita Gamberini, Italy  
Maria L. Gandarias, Spain  
Arman Ganji, Canada  
Xin-Lin Gao, USA  
Zhong-Ke Gao, China  
Giovanni Garcea, Italy  
Fernando García, Spain  
Jose M. Garcia-Aznar, Spain  
Laura Gardini, Italy  
Alessandro Gasparetto, Italy  
Vincenzo Gattulli, Italy  
Oleg V. Gendelman, Israel  
Mergen H. Ghayesh, Australia  
Agathoklis Giaralis, UK  
Anna M. Gil-Lafuente, Spain  
Hector Gómez, Spain  
David González, Spain  
Francisco Gordillo, Spain  
Rama S. R. Gorla, USA  
Oded Gottlieb, Israel  
Nicolas Gourdain, France  
Kannan Govindan, Denmark  
Antoine Grall, France  
Jason Gu, Canada  
Federico Guarracino, Italy  
José L. Guzmán, Spain  
Quang Phuc Ha, Australia  
Masoud Hajarian, Iran  
Frédéric Hamelin, France  
Zhen-Lai Han, China  
Thomas Hanne, Switzerland  
Xiao-Qiao He, China  
Sebastian Heidenreich, Germany  
Luca Heltai, Italy  
Alfredo G. Hernández-Díaz, Spain  
M.I. Herreros, Spain  
Vincent Hilaire, France  
Roland Hildebrand, Germany  
Eckhard Hitzer, Japan  
Jaromir Horacek, Czech Republic  
Muneo Hori, Japan  
András Horváth, Italy  
Gordon Huang, Canada  
Nicolas Hudon, Canada  
Sajid Hussain, Canada  
Asier Ibeas, Spain  
Orest V. Iftime, Netherlands  
Giacomo Innocenti, Italy  
Emilio Insfran, Spain  
Nazrul Islam, USA  
Benoit Iung, France  
Payman Jalali, Finland  
Reza Jazar, Australia  
Khalide Jbilou, France  
Linni Jian, China  
Bin Jiang, China  
Zhongping Jiang, USA  
Ningde Jin, China  
Grand R. Joldes, Australia  
Dylan F. Jones, UK  
Tamas Kalmar-Nagy, Hungary  
Tomasz Kapitaniak, Poland  
Haranath Kar, India  
Konstantinos Karamanos, Belgium  
Jean-Pierre Kenne, Canada  
Chaudry M. Khalique, South Africa  
Do Wan Kim, Republic of Korea  
Nam-Il Kim, Republic of Korea  
Oleg Kirillov, Germany  
Manfred Krafczyk, Germany  
Frederic Kratz, France  
Petr Krysl, USA  
Jurgen Kurths, Germany  
Kyandoghere Kyamakya, Austria  
Davide La Torre, Italy  
Risto Lahdelma, Finland  
Hak-Keung Lam, UK  
Jimmy Lauber, France  
Antonino Laudani, Italy  
Aime' Lay-Ekuakille, Italy  
Nicolas J. Leconte, France  
Marek Lefik, Poland  
Yaguo Lei, China  
Stefano Lenci, Italy  
Roman Lewandowski, Poland  
Panos Liatsis, UAE  
Anatoly Lisnianski, Israel  
Peide Liu, China  
Peter Liu, Taiwan  
Wanquan Liu, Australia  
Yan-Jun Liu, China  
Alessandro Lo Schiavo, Italy  
Jean J. Loiseau, France  
Paolo Lonetti, Italy  
Sandro Longo, Italy  
Sebastian López, Spain  
Luis M. López-Ochoa, Spain  
Vassilios C. Loukopoulos, Greece  
Valentin Lychagin, Norway  
Antonio Madeo, Italy  
José María Maestre, Spain  
Fazal M. Mahomed, South Africa  
Noureddine Manamanni, France  
Didier Maquin, France  
Paolo Maria Mariano, Italy  
Damijan Markovic, France  
Benoit Marx, France  
Franck Massa, France  
Paolo Massioni, France  
Georges A. Maugin, France  
Alessandro Mauro, Italy  
Michael Mazilu, UK  
Driss Mehdi, France  
Roderick Melnik, Canada  
Pasquale Memmolo, Italy  
Xiangyu Meng, Canada  
Jose Merodio, Spain  
Alessio Merola, Italy  
Luciano Mescia, Italy  
Laurent Mevel, France  
Yuri Vladimirovich Mikhlin, Ukraine  
Aki Mikkola, Finland  
Hiroyuki Mino, Japan  
Pablo Mira, Spain  
Vito Mocella, Italy  
Roberto Montanini, Italy  
Gisele Mophou, France  
Rafael Morales, Spain  
Marco Morandini, Italy  
Simone Morganti, Italy

Philip Moser, Germany  
Aziz Moukrim, France  
Emiliano Mucchi, Italy  
Domenico Mundo, Italy  
Jose J. Muñoz, Spain  
Giuseppe Muscolino, Italy  
Marco Mussetta, Italy  
Hakim Naceur, France  
Hassane Naji, France  
Keivan Navaie, UK  
Dong Ngoduy, UK  
Tatsushi Nishi, Japan  
Xesús Nogueira, Spain  
Ben T. Nohara, Japan  
Mohammed Nouari, France  
Mustapha Nourelfath, Canada  
Sotiris K. Ntouyas, Greece  
Roger Ohayon, France  
Mitsuhiro Okayasu, Japan  
Javier Ortega-García, Spain  
Alejandro Ortega-Moñux, Spain  
Naohisa Otsuka, Japan  
Erika Ottaviano, Italy  
Arturo Pagano, Italy  
Alkis S. Paipetis, Greece  
Alessandro Palmeri, UK  
Anna Pandolfi, Italy  
Elena Panteley, France  
Achille Paolone, Italy  
Xosé M. Pardo, Spain  
Manuel Pastor, Spain  
Pubudu N. Pathirana, Australia  
Francesco Pellicano, Italy  
Marcello Pellicciari, Italy  
Haipeng Peng, China  
Mingshu Peng, China  
Zhike Peng, China  
Marzio Pennisi, Italy  
Matjaz Perc, Slovenia  
Francesco Pesavento, Italy  
Dario Piga, Italy  
Antonina Pirrotta, Italy  
Marco Pizzarelli, Italy  
Vicent Pla, Spain  
Javier Plaza, Spain  
Sébastien Poncet, Canada  
Jean-Christophe Ponsart, France  
Mauro Pontani, Italy  
Stanislav Potapenko, Canada  
Sergio Preidikman, USA  
Christopher Pretty, New Zealand  
Carsten Proppe, Germany  
Luca Pugi, Italy  
Giuseppe Quaranta, Italy  
Dane Quinn, USA  
Vitomir Racic, Italy  
Jose Ragot, France  
Kumbakonam Rajagopal, USA  
Gianluca Ranzi, Australia  
Alain Rassineux, France  
S.S. Ravindran, USA  
Alessandro Reali, Italy  
Oscar Reinoso, Spain  
Nidhal Rezg, France  
Ricardo Riaza, Spain  
Gerasimos Rigatos, Greece  
Eugenio Roanes-Lozano, Spain  
Bruno G. M. Robert, France  
José Rodellar, Spain  
Rosana Rodriguez-Lopez, Spain  
Ignacio Rojas, Spain  
Carla Roque, Portugal  
Debasish Roy, India  
Gianluigi Rozza, Italy  
Rubén Ruiz García, Spain  
Antonio Ruiz-Cortes, Spain  
Ivan D. Rukhlenko, Australia  
Mazen Saad, France  
Kishin Sadarangani, Spain  
Mehrdad Saif, Canada  
Miguel A. Salido, Spain  
Roque J. Saltarén, Spain  
Francisco J. Salvador, Spain  
Alessandro Salvini, Italy  
Maura Sandri, Italy  
Miguel A. F. Sanjuan, Spain  
Juan F. San-Juan, Spain  
Roberta Santoro, Italy  
Ilmar Ferreira Santos, Denmark  
José A. Sanz-Herrera, Spain  
Nickolas S. Sapidis, Greece  
Evangelos J. Sapountzakis, Greece  
Andrey V. Savkin, Australia  
Thomas Schuster, Germany  
Mohammed Seaid, UK  
Lotfi Senhadji, France  
Joan Serra-Sagrsta, Spain  
Gerardo Severino, Italy  
Ruben Sevilla, UK  
Leonid Shaikhet, Ukraine  
Hassan M. Shanechi, USA  
Bo Shen, Germany  
Suzanne M. Shontz, USA  
Babak Shotorban, USA  
Zhan Shu, UK  
Dan Simon, Greece  
Luciano Simoni, Italy  
Christos H. Skiadas, Greece  
Michael Small, Australia  
Alba Sofi, Italy  
Francesco Soldovieri, Italy  
Raffaele Solimene, Italy  
Jussi Sopanen, Finland  
Ruben Specogna, Italy  
Sri Sridharan, USA  
Ivanka Stamova, USA  
Salvatore Strano, Italy  
Yakov Strelniker, Israel  
Sergey A. Suslov, Australia  
Thomas Svensson, Sweden  
Andrzej Swierniak, Poland  
Yang Tang, Germany  
Sergio Teggi, Italy  
Alexander Timokha, Norway  
Gisella Tomasini, Italy  
Francesco Tornabene, Italy  
Antonio Tornambe, Italy  
Fernando Torres, Spain  
Sébastien Tremblay, Canada  
Irina N. Trendafilova, UK  
George Tsiatas, Greece  
Antonios Tsourdos, UK  
Vladimir Turetsky, Israel  
Mustafa Tutar, Spain  
Ilhan Tuzcu, USA  
Efstratios Tzirtzilakis, Greece  
Filippo Ubertini, Italy  
Francesco Ubertini, Italy  
Hassan Ugail, UK  
Giuseppe Vairo, Italy  
Kuppalapalle Vajravelu, USA  
Robertt A. Valente, Portugal  
Pandian Vasant, Malaysia  
Miguel E. Vázquez-Méndez, Spain



---

Josep Vehi, Spain  
K. Veluvolu, Republic of Korea  
Alessandro Veneziani, USA  
Fons J. Verbeek, Netherlands  
Franck J. Vernerey, USA  
Georgios Veronis, USA  
Anna Vila, Spain  
Rafael Villanueva, Spain  
Uchechukwu E. Vincent, UK  
Mirko Viroli, Italy  
Michael Vynnycky, Sweden

Shuming Wang, China  
Yan-Wu Wang, China  
Yongqi Wang, Germany  
Roman Wendner, Austria  
Desheng D. Wu, Canada  
Yuqiang Wu, China  
Guangming Xie, China  
Xuejun Xie, China  
Gen Qi Xu, China  
Hang Xu, China  
Joseph-Julien Yamé, France

Xing-Gang Yan, UK  
Luis J. Yebra, Spain  
Peng-Yeng Yin, Taiwan  
Qin Yuming, China  
Vittorio Zampoli, Italy  
Ibrahim Zeid, USA  
Huaguang Zhang, China  
Qingling Zhang, China  
Jian Guo Zhou, UK  
Quanxin Zhu, China  
Mustapha Zidi, France

# Contents

---

**Fault Detection, Estimation/Reconstruction, and Fault-Tolerant Control: Theory and Practice**

Wen Chen, Afef Fekih, and Zehui Mao  
Volume 2016, Article ID 5293171, 2 pages

**Flight Tests of Autopilot Integrated with Fault-Tolerant Control of a Small Fixed-Wing UAV**

Shuo Wang, Ziyang Zhen, Ju Jiang, and Xinhua Wang  
Volume 2016, Article ID 2141482, 7 pages

**A Novel Method of Fault Diagnosis for Rolling Bearing Based on Dual Tree Complex Wavelet Packet Transform and Improved Multiscale Permutation Entropy**

Guiji Tang, Xiaolong Wang, and Yuling He  
Volume 2016, Article ID 5432648, 13 pages

**Calculation and Analysis of Permanent Magnet Eddy Current Loss Fault with Magnet Segmentation**

Bing Li and Ming Li  
Volume 2016, Article ID 7308631, 6 pages

**Minimal-Order Functional Observer-Based Residual Generators for Fault Detection and Isolation of Dynamical Systems**

H. M. Tran and H. Trinh  
Volume 2016, Article ID 2740645, 17 pages

**An Effective Strategy to Build Up a Balanced Test Suite for Spectrum-Based Fault Localization**

Ning Li, Rui Wang, Yu-li Tian, and Wei Zheng  
Volume 2016, Article ID 5813490, 13 pages

**Fault Localization Analysis Based on Deep Neural Network**

Wei Zheng, Desheng Hu, and Jing Wang  
Volume 2016, Article ID 1820454, 11 pages

**Unitary Approximations in Fault Detection Filter Design**

Dušan Krokavec, Anna Filasová, and Pavol Liščinský  
Volume 2016, Article ID 7249803, 14 pages

**Analytical Redundancy Design for Aeroengine Sensor Fault Diagnostics Based on SROS-ELM**

Jun Zhou, Yuan Liu, and Tianhong Zhang  
Volume 2016, Article ID 8153282, 9 pages

**Safety Assessment for Electrical Motor Drive System Based on SOM Neural Network**

Linghui Meng, Peizhen Wang, Zhigang Liu, Ruichang Qiu, Lei Wang, and Chunmei Xu  
Volume 2016, Article ID 2358142, 8 pages

**Monitoring of Distillation Column Based on Indiscernibility Dynamic Kernel PCA**

Qiang Gao, Yong Chang, Zhen Xiao, and Xiao Yu  
Volume 2016, Article ID 9567967, 11 pages

## Editorial

# Fault Detection, Estimation/Reconstruction, and Fault-Tolerant Control: Theory and Practice

Wen Chen,<sup>1</sup> Afef Fekih,<sup>2</sup> and Zehui Mao<sup>3</sup>

<sup>1</sup>*Division of Engineering Technology, Wayne State University, Detroit, MI 48201, USA*

<sup>2</sup>*Department of Electrical and Computer Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504, USA*

<sup>3</sup>*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*

Correspondence should be addressed to Wen Chen; [wchenc@wayne.edu](mailto:wchenc@wayne.edu)

Received 11 July 2016; Accepted 11 July 2016

Copyright © 2016 Wen Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

The increasing complexity of modern engineering systems will correspondingly increase the possibility of system faults and/or failures. The occurrence of sensor, actuator, and/or component failures may dramatically degrade system performance and even result in catastrophic system collapse. As a response to high requirement for system safety, reliability, and survivability, fault diagnosis and fault-tolerant control (FTC) for dynamic systems have been attractive subjects of many investigations in control community and have received considerable attention during the past few decades, and a great deal of research progress has been made.

The articles in this special issue speak on recent development of fault detection, estimation/reconstruction, and fault-tolerant control as well as their applications. The following sections briefly explain the outcome of published articles in this special issue. They are divided into five key research areas based on publications.

## 2. Fault Localization

W. Zheng et al. proposed a fault localization method based on a deep neural network (DNN). It is capable of achieving the complex function approximation and attaining distributed representation for input data by learning a deep nonlinear network structure. Another advantage is its strong ability to learn from a small-sized training dataset. The DNN-based model is trained utilizing the coverage data and the results

of test cases as inputs and the faults can be located by testing the trained model using the virtual test suite. Another article on fault localization is proposed by N. Li et al., which is about building up a balanced test suite for spectrum-based fault localization. They proposed a PNF strategy to reduce test suite and build up a more balanced one for SBFL. They have evaluated the strategy by experiments using the Siemens and Space program. Experiments indicated that the proposed PNF strategy can be used to construct a new test suite effectively. Compared with the original test suite, the new one has a smaller size and more balanced ratio of failed test cases to previous test cases, while it has the same statement coverage and fault localization accuracy.

## 3. Fault-Tolerant Control

S. Wang et al. suggested a fault-tolerant control scheme for autopilot of a small fixed-wing UAV. The flight control system is designed based on an open-source autopilot (Pixhawk). Real-time kinematic (RTK) GPS is introduced due to its high accuracy. Some modifications on the longitudinal and lateral guidance laws are achieved to improve the flight control performance. Moreover, a data fusion-based fault-tolerant control scheme is integrated into altitude control and speed control for altitude and airspeed sensor failures. Actual flight test results show that the fixed-wing UAV can track the desired altitude and speed commands during the whole flight process by the fault-tolerant control-based autopilot.

#### 4. Fault Diagnosis Methods

New fault-diagnostic methods have been presented using different techniques such as observers, filters, SROS-ELM, and the indiscernibility dynamic kernel PCA. D. Krokavec et al. suggested a new fault detection filter design that relaxes the existing conditions by adapting the unitary system principle in approximation of fault detection filter transfer function matrix for continuous-time linear MIMO systems. H. M. Tran and H. Trinh examined the design of minimal-order residual generators for detecting and isolating actuator and/or component faults in dynamical systems. First, existence conditions and design residual generators are derived using only first-order observers to detect and identify the faults. When the first-order functional observers do not exist, then based on a parametric approach to the solution of a generalized Sylvester matrix equation a systematic procedure for designing residual generators utilizing minimal-order functional observers is developed. J. Zhou et al. proposed an analytical redundancy design for aeroengine sensor fault diagnostics based on SROS-ELM. The modified online sequential extreme learning machine is employed to train the model online and estimate sensor measurements. It selectively updates the output weights of neural networks, tackles the problems of singularity and ill-posedness by regularization, and adopts a dual activation function in the hidden nodes. Another article related to a new method for diagnosing faults of rolling bearing based on a dual-tree complex wavelet packet transform and improved multiscale permutation entropy is presented by G. Tang et al. The experimental data analysis results have verified the effectiveness of the presented diagnosis method and demonstrated that this method can be applied to distinguish the different fault types and fault degrees of rolling bearings. Aimed at complicated faults detection of distillation column industrial processes, Q. Gao et al. suggested a new indiscernibility dynamic kernel principal component analysis (I-DKPCA) method. Compared with traditional statistical techniques, the I-DKPCA can not only capture nonlinear property and dynamic characteristic of processes, but also extract relevant items from all of the variables.

#### 5. Safety Assessment

L. Meng et al. pursued a safety assessment for electrical motor based on an SOM neural network. In this work, the system's health clustering and safety identification are carried out. With the switch devices' faults data obtained from the dSPACE simulation platform, the health assessment algorithm is proposed and verified. The results show that the algorithm can achieve the system's fault diagnosis and health assessment.

#### 6. Fault Analysis

B. Li and M. Li calculated and analyzed the effect of the permanent magnet eddy current loss fault due to magnet segmentation. A three-dimensional finite-element model was

firstly established; then, model meshes and boundary conditions were handled specially, and permanent magnet eddy current loss fault was calculated and analyzed. Finally, calculated results were compared and explained.

*Wen Chen  
Afef Fekih  
Zehui Mao*

## Research Article

# Flight Tests of Autopilot Integrated with Fault-Tolerant Control of a Small Fixed-Wing UAV

Shuo Wang,<sup>1,2</sup> Ziyang Zhen,<sup>1,2</sup> Ju Jiang,<sup>1,2</sup> and Xinhua Wang<sup>1,2</sup>

<sup>1</sup>College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>2</sup>Key Laboratory of Internet of Things and Control Techniques of Jiangsu Province, Nanjing 210016, China

Correspondence should be addressed to Ziyang Zhen; [zhenziyang@nuaa.edu.cn](mailto:zhenziyang@nuaa.edu.cn)

Received 25 December 2015; Accepted 15 June 2016

Academic Editor: Wen Chen

Copyright © 2016 Shuo Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A fault-tolerant control scheme for the autopilot of the small fixed-wing UAV is designed and tested by the actual flight experiments. The small fixed-wing UAV called Xiang Fei is developed independently by Nanjing University of Aeronautics and Astronautics. The flight control system is designed based on an open-source autopilot (Pixhawk). Real-time kinematic (RTK) GPS is introduced due to its high accuracy. Some modifications on the longitudinal and lateral guidance laws are achieved to improve the flight control performance. Moreover, a data fusion based fault-tolerant control scheme is integrated in altitude control and speed control for altitude sensor failure and airspeed sensor failure, which are the common problems for small fixed-wing UAV. Finally, the real flight experiments are implemented to test the fault-tolerant control based autopilot of UAV. Real flight test results are given and analyzed in detail, which show that the fixed-wing UAV can track the desired altitude and speed commands during the whole flight process including takeoff, climbing, cruising, gliding, landing, and wave-off by the fault-tolerant control based autopilot.

## 1. Introduction

Unmanned aerial vehicle (UAV) is one kind of aerospace system which has been widely applied in military and civil fields, such as surveillance, attack, fire alarm, and even express delivery. UAV is composed of several parts, in which the flight control system including autopilot and guidance devices is the key part.

Considerable research has been done on autopilot hardware design, attitude control, altitude control methods, and so forth. Pu et al. briefly surveyed the development of UAV and its subsystems including autopilot, ground control station, and navigation system. UAV flight control methods were also summarized [1]. He et al. designed an autopilot based on FPGA with embedded Linux system [2]. Tu and Du designed an autopilot hardware system based on DSP [3]. Most of UAVs use low-performance MEMS sensors as parts of their autopilots because of the limitation of production cost, which increases the probability of fault on these kinds of devices. Zhen et al. designed longitudinal and lateral controllers based on information fusion optimal control [4, 5]. Simulation was implemented on a nonlinear alterable thrust direction UAV,

which showed that information fusion-based optimal control improves the anti-wind disturbance performance, better than traditional single-loop method; however, this method was not verified by real flight tests [4, 5]. Matveev et al. applied sliding mode control theory on the guidance and control of unicycle-like vehicle; considering obstacle avoidance, simulation results indicated the efficiency of the proposed algorithm in border patrolling and obstacle avoidance [6]. This method is also suitable for applications of other UAVs; however, fault-tolerant problem was not considered in the controller design. Yamasaki et al. implemented high-order sliding mode differentiator on the estimation of uncertain sliding surfaces and did some simulations on YF-16 model, and results showed that the second-order sliding mode controller provided excellent path-following performance under the wind turbulence circumstance [7]. Cayero et al. built a thorough model of TX-1570 and designed a nonlinear controller based on adaptive backstepping control theory, and the simulation showed that the flight-path error of longitudinal path traditional is small enough [8]. But fault of sensors was not considered in this paper, and flight test was not carried out.

Health monitoring is responsible of detecting, isolating, and identifying the faults, failures, and upsets in related systems. As an extension, fault-tolerant control is a health management technique. The main role of the fault-tolerant control is to accommodate different faults and failures and aid the overall system in achieving high level of safety and reliability over the lifetime of the system. Fault-tolerant control is very important to the autopilots of UAVs. Some researches on UAV fault-tolerant control theory have been done in recent years. Jourdan et al. demonstrated in-flight MRAC-based inner-loop attitude control in the presence of severe structural faults on UAV in Rockwell Collins Control Techniques [9]. Bateman et al. developed the multiple model adaptive estimation for sensor and actuator faults and simulated a nonlinear model of aircraft with aileron and rudder failures [10]. Reference [11] used West Virginia University UAV simulation environment to verify the nonlinear dynamic inversion approach augmented with an artificial immune system in autonomous UAV flight in the presence of disturbances and actuator failures. Reference [12] gave the flight-test results of fault-tolerant guidance and control algorithm on the GT Twinstar of Georgia Institute of Technology, under the actuator failures and severe structural damage conditions. Reference [13] designed fault-tolerant controller for a tilt-rotor UAV for actuator and sensor faults. The fault-tolerant federated Kalman filter presented in this paper guaranteed the accuracy and robustness of state estimation. At last, numerical simulation was implemented to prove that. Neural control was used in [14] for auto-landing when aircraft is subjected to actuator faults and severe winds, and simulation showed that the neural controller based Single Hidden Layer Feedforward Networks (SLFNs) can achieve safe landing and track reference signals well under actuator failure. But neural controller needs much calculation, which is difficult to implement on MCU. We noticed that most of the fault-tolerant control designs are focused on numerical simulation rather than actual flight tests. Besides, some advanced control theories need a great deal of calculation, which is difficult to implement on autopilot of UAV.

In this paper, we design a fault-tolerant flight control system for an open-source autopilot and give the real flight tests on a small fixed-wing UAV called Xiang Fei, which is developed by Nanjing University of Aeronautics and Astronautics. This fault-tolerant flight control system consists of autopilot, real-time kinematic (RTK) GPS navigation system, and ground control station. For the proposed fault-tolerant control scheme, a sensor redundancy strategy guarantees the robustness of controller, and a failure detection algorithm and a fault-tolerant control algorithm are integrated in altitude control and speed control. The open-source autopilot is Pixhawk [15]. Compared with the original Pixhawk autopilot, we improve the position and altitude measurement using RTK GPS. Finally, the fault-tolerant flight control system is tested on a small fixed-wing UAV to prove the reliability and stability of the initially developed autopilot, which is rare in current researches.

TABLE 1: Parameters and performance of Xiang Fei UAV.

Weight	18 kg
Length	2 m
Max payload	5 kg
Material	Fiberglass
Gear type	Tricycle landing gear
Max range	40 km
Propeller	Wooden 2212
Wingspan	3.6 m
Max cruise time	1 h
Power type	80 cc petrol engine
Cruise speed	30 m/s
Max speed	45 m/s
Area of horizon tail	32 dm <sup>2</sup>
Area of wing	127 dm <sup>2</sup>



FIGURE 1: Picture of Xiang Fei UAV.

## 2. Independently Designed Small Fixed-Wing UAV

Small fixed-wing UAV is widely used in military and civilian applications, including search, monitoring, mapping, and earthquake rescue, because of its flexibility and low cost. Many universities and companies have been doing a mass of researches on fixed-wing UAV, including attitude control, trajectory tracking, and fault-tolerant control. Figure 1 shows a small fixed-wing UAV named Xiang Fei. The details about this aircraft are shown in Table 1. The purpose of designing this UAV was to do research on autonomous flight, including auto-takeoff, path tracking, and auto-landing on moving platform or ship with high accuracy. The small fixed-wing UAV is low-cost and has high performance.

## 3. Autopilot Design of Xiang Fei UAV

Pixhawk autopilot is one of the most famous open-source autopilots, which is designed by the open hardware development team of 3D Robotics Company. Kinds of sensors are integrated in the circuit board, and open-source flight software (ArduPlane) is combined with the real-time operating system. However, there are some shortages in hardware and software design of Pixhawk. For example, barometer

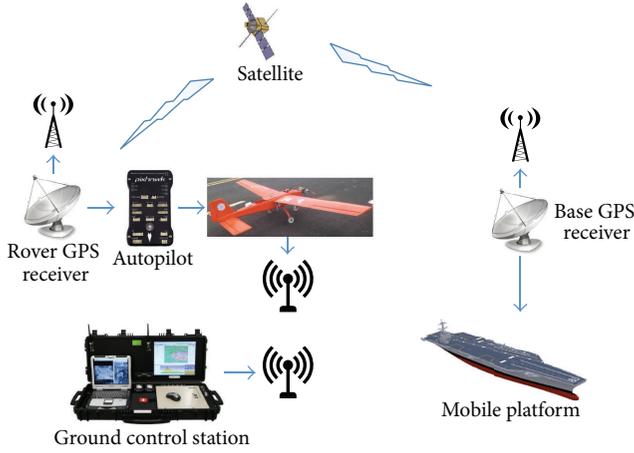


FIGURE 2: Principle diagram of UAV hardware system.

is the only sensor to measure altitude of UAV in Pixhawk, which leads to poor accuracy of altitude. Sensor failures are not considered in the software of Pixhawk. Hence, several improvements on hardware and software will be given in the following.

**3.1. Hardware Design of Autopilot.** Pixhawk contains kinds of on-board sensors, such as low-cost 9-DOF AHRS, barometer, and electronic compass. However, the accuracy and stability of on-board sensors are usually not high enough for UAV flight. Besides, sensor failures have not been considered. To solve these problems, a real-time kinematic (RTK) GPS is integrated in the proposed fault-tolerant flight control system. RTK operation provides centimeter-level accuracy by eliminating errors that are presented in the GPS system. The RTK GPS system contains the base station and the rover station. The base station is placed at ground or moving platform. At the same time, both receivers track the same satellites, and then the message of position corrections of the base station is sent through a radio link to the rover receiver, where these messages are used to calculate the real-time positions of the rover. Figure 2 shows the principle diagram of the whole hardware system. RTK GPS base station is placed on a moving platform where the UAV is expected to land. A pair of radios is used to transmit position corrections for RTK GPS. Pixhawk and RTK GPS rover station are placed on the fixed-wing UAV. Another pair of radios is used to transmit flight data from aircraft to the ground control station.

**3.2. Software Design of Autopilot.** Pixhawk has flexible flight control software, supporting various aircraft vehicles, including UAVs, quadrotors, planes, and helicopters. It contains a variety of interfaces, such as I2C, SPI, and UART, which are all adaptable and open to users. The original flight control software of Pixhawk is Arduplane [16], which is an open-source UAV platform and is able to control autonomous aircrafts, created by DIY Drones community. The flow chart of Arduplane is shown in Figure 3 [16]. In Figure 3, the setup function is the initialization of autopilot, including sensors

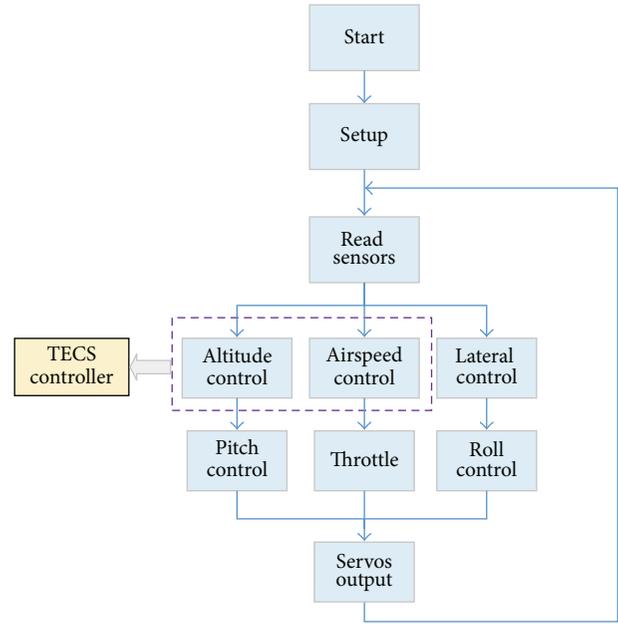


FIGURE 3: Flow chart of Arduplane.

initialization, variables initialization, and flight mission reading. Then an endless loop will be carried out after the setup function. The endless loop contains sensors read function, flight controller logic calculation, and servos output function. TECS controller is used for altitude control and speed control, which will be detailed in the next section.

**3.3. Improved Guidance and Control System of UAV.** In this section, the longitudinal guidance and lateral guidance are redesigned to improve the flight control performance of the UAV.

Total energy control system (TECS) [17] is used in Pixhawk to track altitude and speed commands. Altitude control and speed control are coupled in TECS controller. There are about 20 parameters in TECS controller of Pixhawk. Some parameters have influence on both the altitude and the speed of the UAV, so it is hard to choose proper parameter values due to coupling, especially for the small fixed-wing UAV. To solve this problem, The TECS is replaced by the separate PID control system for altitude and speed tracking, and the parameter of each PID controller can be tuned separately.

The lateral guidance law of Pixhawk uses  $L_1$  control algorithm [18]. The  $L_1$  control method is good at trajectory tracking when following curved paths.  $L_1$  controller selects a reference point on the desired trajectory and generates a lateral acceleration command using the reference point; then the desired roll angle can be obtained using lateral acceleration command through specific transmit formulation. For straight line path tracking, the reference point is on the desired path at a distance ( $L_1$ ) forward of the UAV, as shown in Figure 4.

To decrease the cross track error  $d$ , shown in Figure 4, the UAV should have lateral acceleration. The desired lateral acceleration can be determined by

$$a_y = 2 \frac{V^2}{L_1} \sin \eta, \quad (1)$$

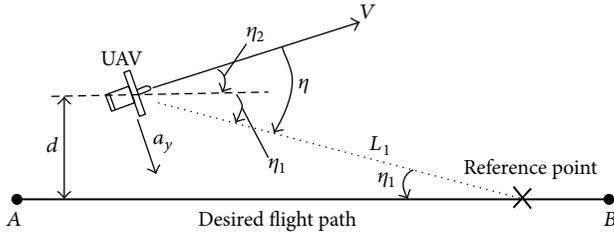


FIGURE 4: Straight line path tracking.

where  $V$  is the airspeed,  $\eta$  is the angle between vector  $V$  and direction from position of UAV to reference point, and  $L_1$  is the distance between the position of UAV and reference point, which can be determined by

$$L_1 = \frac{1}{\pi} \cdot \zeta \cdot T \cdot V, \quad (2)$$

where  $\zeta$  and  $T$  are damping and period, respectively, which can be adjusted by the users.

The angle  $\eta$  can be calculated by

$$\begin{aligned} \eta &= \eta_1 + \eta_2, \\ \eta_1 &= \arcsin \frac{d}{L_1}, \\ \eta_2 &= \arctan \frac{\vec{V} \times \vec{AB}}{\vec{V} \cdot \vec{AB}}, \end{aligned} \quad (3)$$

where  $d$  is the cross track error and  $A$  and  $B$  are the previous position and next position of desired flight path, shown in Figure 5. Vector  $\vec{AB}$  can be obtained by GPS location of positions  $A$ ,  $B$ .

Then, to obtain desired lateral acceleration, the UAV should roll. The desired roll angle can be obtained by

$$\phi = \tan^{-1} \left( \frac{a_y}{g} \right), \quad (4)$$

where  $g$  is the acceleration of gravity and  $a_y$  can be obtained by combination of (1)–(3).

The desired path is generated by connecting each independent position directly for Pixhawk, without any smooth. In the case of sharp turn, for example, in Figure 5(a), the switch of path from  $a$  to  $b$  will lead to sudden large change of desired roll angle calculated by  $L_1$  controller. To fix this problem, we redesigned the desired flight path through adding a coordinated turn stage (line  $c$ ) between the adjacent line paths, shown in Figure 5(b).

The desired roll angle in coordinated turn can be determined by

$$\phi = \frac{V^2}{Rg}, \quad (5)$$

where  $R$  is the radius of turning. The desired roll angle is easy to calculate when the desired radius of turning is known.

## 4. Fault-Tolerant Control Scheme for Xiang Fei UAV

**4.1. Fault-Tolerant Design for Altitude Controller.** Barometer is the unique altitude sensor of Pixhawk. However, the barometer sensor is very sensitive to temperature and atmospheric turbulence, which makes it difficult for altitude tracking and auto-landing with high accuracy. In this paper, we use RTK GPS as the source of altitude data. The accuracy of RTK GPS system can reach centimeter-level; however, RTK GPS system sometimes is affected by communication problem between base station and rover station. When failure occurs in any of the transmitters of RTK GPS receivers, the rover GPS station cannot receive correction messages, which will cause the UAV crash. To fix this problem, we introduce the data fusion scheme for the information of RTK GPS height and barometer height. Thus, the new structure of height controller is shown in Figure 6.

The RTK GPS device used here is Novatel OEM 617. The rover station can output 10 Hz of relative altitude at the frequency of 10 Hz when it works normally; when failure occurs in the communication between base station and rover station, the rover station will not output any data of relative altitude, so we will know whether the RTK GPS is working normally, so we can deal with communication fault when failure occurs.

Although the barometer sensor will generate drift in long time, the instantaneous rate of change of barometer altitude is accurate. So we can use this data as a compensation of RTK GPS altitude when failure occurs on RTK GPS. The fault-tolerant design is the following: let autopilot store barometer altitude and RTK GPS altitude in the same time; when RTK GPS failure occurs, we can calculate an estimation of real altitude based on normal RTK GPS altitude measured before communication fault and barometer altitude rate. The calculation formula is

$$H_{\text{est}} = H_{\text{GPS}} + H_{\text{rate}} * t, \quad (6)$$

where  $H_{\text{est}}$  is the estimation of real altitude,  $H_{\text{GPS}}$  is the RTK GPS altitude measured before failure,  $H_{\text{rate}}$  is the barometer altitude rate, and  $t$  is the duration of failure time.

When RTK GPS works normally again, we change to use normal RTK GPS data in altitude controller. A first-order filter is used to reduce the influence of switch between estimation altitude and RTK GPS altitude.

**4.2. Fault-Tolerant Design for Speed Controller.** The speed control of UAV is very important; because the lift force of UAV is related to the airspeed, sufficient airspeed provides enough lift force for flight safety. Airspeed sensor is usually influenced by electromagnetic interference or block of pitot tube of airspeed sensor. Therefore, it is important to design fault-tolerant control for airspeed sensor failure.

To detect airspeed sensor error, we use GPS speed and airspeed to calculate the wind speed. When wind speed is much larger than the real wind speed, which can be measured by special instrument, that means airspeed sensor is not normal. We use GPS speed as the compensation for airspeed.

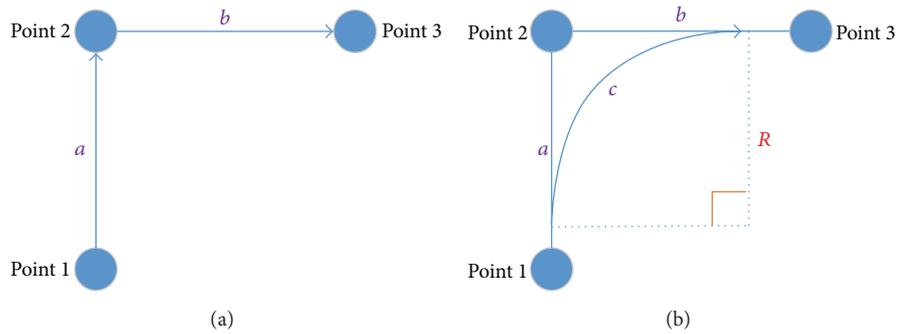


FIGURE 5: Switch of line path.

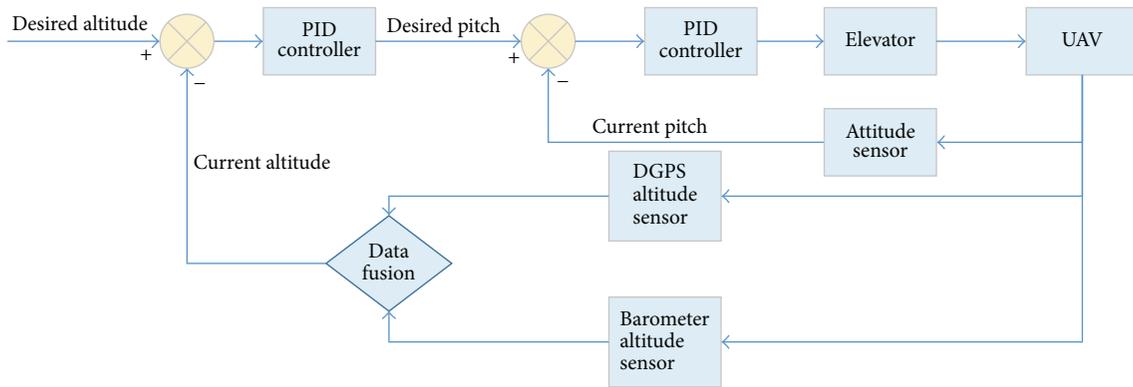


FIGURE 6: Structure of altitude controller.

Figure 7 shows the structure of the fault-tolerant airspeed controller. The difference value between airspeed and GPS speed is input into data fusion. When the speed error is larger than current wind speed which is set by UAV pilot, we change to use GPS speed as the current speed of UAV; otherwise, airspeed sensor data will be used directly. A one-order filter is used to reduce the fluctuation of speed signal switch.

### 5. Flight Test Results and Analysis

The fault-tolerant flight control system is tested on the small fixed-wing UAV called Xiang Fei, shown in Figure 1. The tests contain the whole flight process from the takeoff, climbing, cruising, gliding, and wave-off to the auto-landing. Particularly, the UAV is required to land on a moving platform at the speed of about 2 m/s, which increases the difficulty for flight control. Figure 8 shows the UAV in flight tests.

Figure 9 shows the altitude tracking of entire flight process, including takeoff stage, climbing stage, and landing stage. We can see that the UAV can track the altitude commands very well in the whole flight stages. The error is a little large only on the takeoff stage (at 100 s) and wave-off stage (at 360 s), when insufficient power causes the UAV to climb too slow to track the climbing command. The altitude error is less than 1 m at the auto-landing stage, which is small enough to ensure the UAV landing on specified point of the moving platform.

Figure 10 shows the altitude error in an auto-landing flight test. The altitude error average is about 0.3 m, which is accurate enough for auto-landing. The fluctuation of altitude error may be due to noise of GPS signal.

Figure 11 shows the airspeed and GPS speed of UAV. The desired cruise airspeed is 22 m/s. We can know that the speed keeps stable in the whole flight process. There is some drastic change on airspeed, due to stochastic disturbance on airspeed sensor, which can be weakened by low-pass filter. The GPS speed only has a small difference from airspeed because of the influence of wind.

Figure 12 shows the roll angle tracking of UAV during turning. The error is small enough to guarantee good performance of lateral path tracking. Besides, the maximum roll angle command generated by autopilot is less than 35°, due to our improvement on lateral controller.

### 6. Conclusion

A fault-tolerant flight control system based on Pixhawk for a small fixed-wing UAV is designed to control the UAV flight with high performance against the sensor failure. The fault-tolerant schemes of the altitude controller and the speed controller can ensure the UAV flight normally in case failures occurred on RTK GPS or airspeed sensor. The separation of altitude control loop and speed control loop is suitable for the small fixed-wing UAV. The improvement of lateral guidance

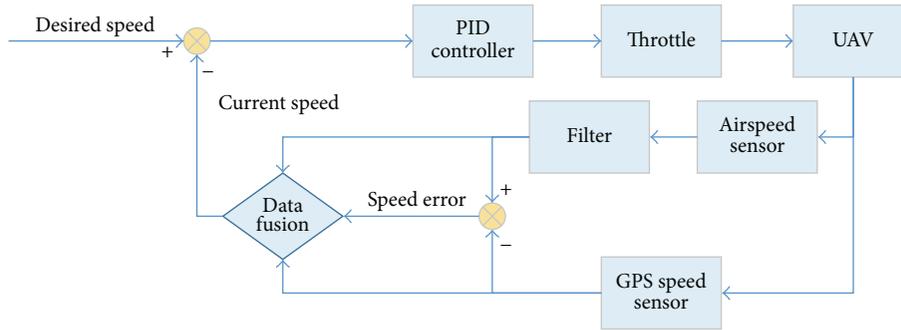


FIGURE 7: Structure of speed controller.



FIGURE 8: UAV in flight experiment.

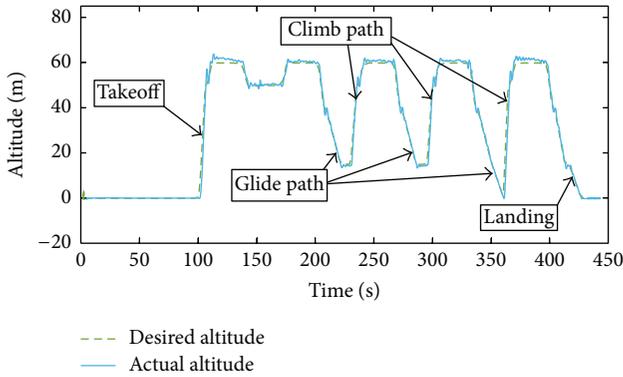


FIGURE 9: Altitude tracking of UAV.

and control system can smooth the desired roll angle, which is much better than original lateral control logic of Pixhawk. Finally, the real flight tests are carried out, in which the whole flight process includes the takeoff, climbing, cruising, gliding, and landing on a moving platform. Flight test results show the effectiveness of the proposed fault-tolerant flight control scheme. The proposed fault-tolerant control scheme is mainly focused on altitude sensor and airspeed sensor failures. Actuator failures can be considered in the future work to improve the robustness of UAV autopilots.

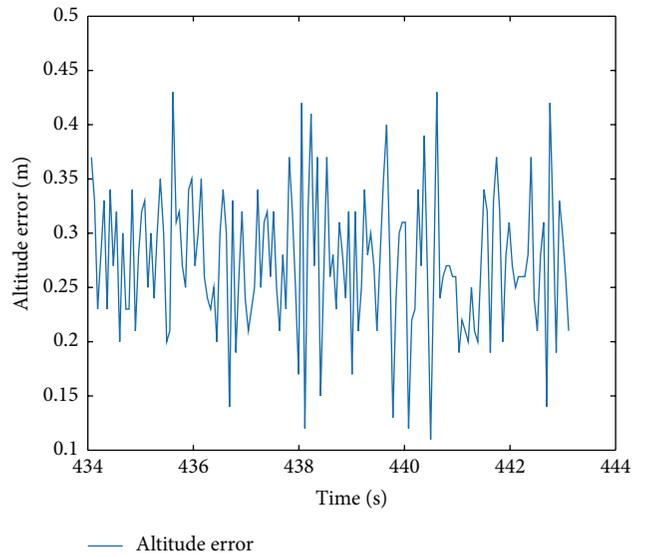


FIGURE 10: Altitude error in auto-landing.

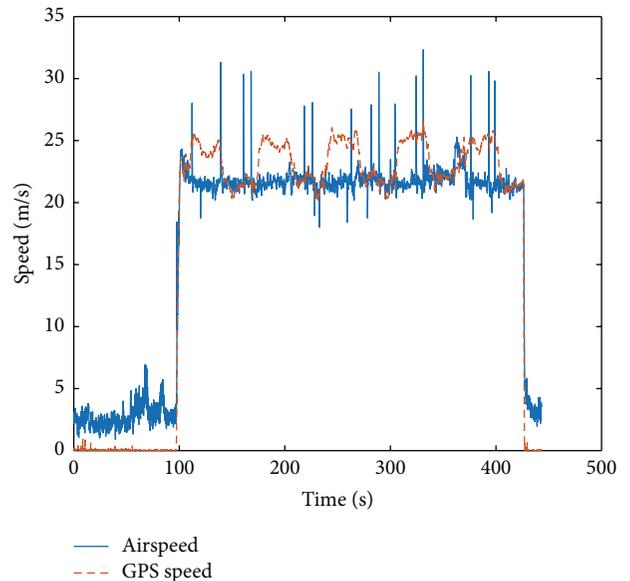


FIGURE 11: Airspeed and GPS speed of UAV.

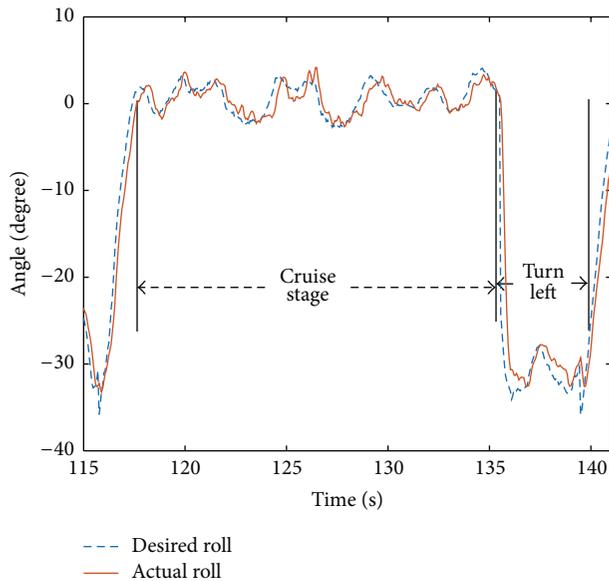


FIGURE 12: Roll angle tracking of UAV.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61304223), the Specialized Research Fund for the Doctoral Program of Higher Education (no. 20123218120015), and the Fundamental Research Funds for the Central Universities (no. NJ20160026).

## References

- [1] H. Z. Pu, Z. Y. Zhen, and M. Xia, "Flight control system of unmanned aerial vehicle," *Transactions of Nanjing University of Aeronautics and Astronautics*, vol. 32, no. 1, pp. 1–8, 2015.
- [2] G. L. He, R. J. Guo, and Y. Shi, "Application of FPGA in small UAV autopilot based on embedded LINUX system," in *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON '07)*, pp. 731–734, Taipei, Taiwan, November 2007.
- [3] H. Tu and X. Du, "The design of small UAV autopilot hardware system based on DSP," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '10)*, pp. 780–783, IEEE, Changsha, China, May 2010.
- [4] Z. Zhen, D. Wang, and Q. Kang, "UAV flight trajectory control based on information fusion control method," in *Proceedings of the International Conference on Networking, Sensing and Control (ICNSC '10)*, pp. 337–341, IEEE, April 2010.
- [5] Z. Zhen, J. Jiang, X. Wang, and D. Wang, "Information fusion-based optimal attitude control for an alterable thrust direction unmanned aerial vehicle," *International Journal of Advanced Robotic Systems*, vol. 10, article 45, 2013.
- [6] A. S. Matveev, H. Teimoori, and A. V. Savkin, "A method for guidance and control of an autonomous vehicle in problems of border patrolling and obstacle avoidance," *Automatica*, vol. 47, no. 3, pp. 515–524, 2011.
- [7] T. Yamasaki, S. N. Balakrishnan, and H. Takano, "Integrated guidance and autopilot for a path-following UAV via high-order sliding modes," in *Proceedings of the American Control Conference (ACC '12)*, pp. 143–148, IEEE, Montreal, Canada, June 2012.
- [8] J. Cayero, B. Morcego, and F. Nejjari, "Modelling and adaptive backstepping control for TX-1570 UAV path tracking," *Aerospace Science and Technology*, vol. 39, pp. 342–351, 2014.
- [9] D. Jourdan, M. Piedmonte, V. Gavrillets, D. Vos, and J. McCormick, "Enhancing UAV survivability through damage tolerant control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1–26, Toronto, Canada, August 2010.
- [10] F. Bateman, H. Noura, and M. Ouladsine, "Fault diagnosis and fault-tolerant control strategy for the aerosonde UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2119–2137, 2011.
- [11] H. Moncayo, M. G. Perhinschi, B. Wilburn, J. Wilburn, and O. Karas, "UAV adaptive control laws using non-linear dynamic inversion augmented with an immunity-based mechanism," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2012.
- [12] G. Chowdhary, E. N. Johnson, R. Chandramohan, M. S. Kimbrell, and A. Calise, "Guidance and control of airplanes under actuator failures and severe structural damage," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1093–1104, 2013.
- [13] S. Park, J. Bae, Y. Kim, and S. Kim, "Fault tolerant flight control system for the tilt-rotor UAV," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2535–2559, 2013.
- [14] J.-M. Bai and H.-J. Rong, "Fault-tolerant autoland controller design using neural network," in *Proceedings of the 31st Chinese Control Conference (CCC '12)*, pp. 3017–3022, July 2012.
- [15] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: a micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.
- [16] H. Bin and A. Justice, "The design of an unmanned aerial vehicle based on the ArduPilot," *Indian Journal of Science and Technology*, vol. 2, no. 4, pp. 12–15, 2009.
- [17] A. A. Lambregts, "Functional integration of vertical flight path and speed control using energy principles," in *Proceedings of the 1st Annual NASA Aircraft Controls Workshop*, pp. 389–409, Hampton, Va, USA, 1993.
- [18] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 1–16, Providence, RI, USA, August 2004.

## Research Article

# A Novel Method of Fault Diagnosis for Rolling Bearing Based on Dual Tree Complex Wavelet Packet Transform and Improved Multiscale Permutation Entropy

Guiji Tang, Xiaolong Wang, and Yuling He

*School of Energy, Power and Mechanical Engineering, North China Electric Power University, Baoding 071000, China*

Correspondence should be addressed to Yuling He; [heyuling1@163.com](mailto:heyuling1@163.com)

Received 8 February 2016; Accepted 7 April 2016

Academic Editor: Wen Chen

Copyright © 2016 Guiji Tang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel method of fault diagnosis for rolling bearing, which combines the dual tree complex wavelet packet transform (DTCWPT), the improved multiscale permutation entropy (IMPE), and the linear local tangent space alignment (LLTSA) with the extreme learning machine (ELM), is put forward in this paper. In this method, in order to effectively discover the underlying feature information, DTCWPT, which has the attractive properties as nearly shift invariance and reduced aliasing, is firstly utilized to decompose the original signal into a set of subband signals. Then, IMPE, which is designed to reduce the variability of entropy measures, is applied to characterize the properties of each obtained subband signal at different scales. Furthermore, the feature vectors are constructed by combining IMPE of each subband signal. After the feature vectors construction, LLTSA is employed to compress the high dimensional vectors of the training and the testing samples into the low dimensional vectors with better distinguishability. Finally, the ELM classifier is used to automatically accomplish the condition identification with the low dimensional feature vectors. The experimental data analysis results validate the effectiveness of the presented diagnosis method and demonstrate that this method can be applied to distinguish the different fault types and fault degrees of rolling bearings.

## 1. Introduction

Rolling bearings are one of the most widely used parts in rotating machineries because they affect the operation reliability, the performance precision, and the service life of the entire equipment. The failures of rolling bearings may cause catastrophic accidents and result in great loss. Therefore, condition monitoring and fault diagnosis for rolling bearings are of great significance in engineering application [1, 2].

Due to the factors such as friction, strike, and structure transmutation, the vibration signals of bearings are often characterized by nonlinearity and nonstationarity. And the major challenge for bearing condition monitoring and fault diagnosis is to acquire the reliable and sensitive features from the vibration signals [3]. In recent years, with the development of nonlinear dynamic theories, a series of nonlinear parameter estimation techniques have been investigated and introduced to the field of bearing condition monitoring and fault diagnosis. For example, the correlation dimension

was chosen as a tool for discovering the fault features of bearings by Kang et al. [4]. Unfortunately, the estimation of correlation dimension usually requires sufficient data, which prevents this technology from being widely used. Yan and Gao [5] applied approximate entropy (AE) to monitor the bearing condition. However, AE depends heavily on the signal length and the calculated value is uniformly smaller than the expected one when processing short term signals [6]. Later, the sample entropy (SE) was proposed by Richman and Moorman [7] to overcome the drawback of AE. The collected signals from bearing systems usually consist of multiple temporal scale structures. But AE and SE both evaluate the complexity of signal at a single scale. Hence, these two approaches have limited performance in analyzing the bearing signal. Considering the disadvantage of the single scale analysis, the multiscale entropy (MSE) was developed by Costa et al. [8] to estimate the complexity of time series over a range of scales, and this technology was used by Zhang et al. [9] to extract the features of bearing signal. However, the

estimation of MSE is easily affected by the outliers of signal, and the computational efficiency of MSE is very low for long term signals.

In literature [10], a new kind of entropy named permutation entropy (PE) was proposed to measure the complexity and detect the dynamic changes of signal. Compared with AE and SE, the calculation of PE is simple and immune to noises. But similar to AE and SE, PE also conducts the entropy measure in a single scale. Then the multiscale permutation entropy (MPE) method based on PE was further proposed by Aziz and Arif [11] to depict the multiple temporal scale structures of the signal. And this method was, respectively, applied by Li and Zheng to bearing fault diagnosis [12, 13]. Nevertheless, the analysis results of MPE are usually unstable for short term signals. Recently, a novel method called improved multiscale permutation entropy (IMPE) was proposed by Azami and Escudero [14] to remedy the weakness of MPE and the effectiveness of this method has been verified by the simulated signal and the real biomedical signal. In view of the advantage of IMPE in digging the inherent features of signal, this method is introduced to the field of fault diagnosis and utilized to identify the condition of rolling bearing in this paper.

Usually, the collected bearing signals are more or less contaminated by external environmental noises, and the interference between the components in the complicated signal is inevitable. These factors lead to the difficulty of feature information extraction using IMPE method directly. And it will be of benefit for the subsequent analysis procedure if the original signals are processed in advance.

Up to now, many signal processing techniques such as empirical mode decomposition (EMD), local mean decomposition (LMD), and discrete wavelet transform (DWT) have been developed and applied in different research fields. As a kind of adaptive signal processing method, EMD could decompose a signal into a set of intrinsic mode functions. However, EMD lacks a forceful mathematical framework and has the problems of mode mixture and end effects. Similar to EMD, the LMD algorithm also involves these drawbacks which have not been fundamentally addressed [15]. DWT is a classic signal processing tool and has been widely used to analyze the mechanical fault signals, but its disadvantages of shift variance and frequency aliasing may cause the loss of useful information. Then Kingsbury [16] proposed the dual tree complex wavelet transform (DTCWT) method, which possesses many excellent properties such as nearly shift invariance, good directional selectivity, and reduced aliasing in comparison with DWT. However, DTCWT cannot achieve multiresolution analysis in the high frequency region where the useful feature information usually exists. As a kind of expansion of DTCWT, dual tree complex wavelet packet transform (DTCWPT) is developed to offset this shortcoming [17]. After performing DTCWPT on the collected signal, precise frequency band partitions over the whole analyzed frequency domain could be achieved and the corresponding subband signals could be obtained. And it will be more effective to discover the feature information of the original signal by analyzing the subband signals using IMPE. Based

on the above analysis, DTCWPT is combined with IMPE for the first time in this study for bearing fault diagnosis.

After feature extraction using DTCWPT and IMPE, the acquired feature vectors need to be fed into the classifier to achieve condition recognition. However, the acquired vectors with high dimension inevitably contain redundancy information. It is time-consuming and may lead to a decline in the diagnostic accuracy if the entire vectors are adopted as the inputs of classifier. Then the manifold learning algorithm named linear local tangent space alignment (LLTSA) [18] is employed in this paper to reduce the dimension of vectors. By using LLTSA, the high dimensional feature vectors are automatically compressed and sensitive feature vectors with lower dimension could be obtained, which will not only reduce the computational burden but also improve the diagnostic precision.

Naturally, an intelligent classifier is needed to automatically distinguish the bearing condition based on the obtained sensitive feature vectors. Extreme learning machine (ELM) [19] is a novel powerful intelligent machine learning approach based on single hidden layer feed-forward networks. Compared with some classic machine learning methods such as support vector machines (SVM), artificial neural network (ANN), and  $K$  nearest neighbor classifier (KNNC), the main advantages of ELM lie in better generalization ability on the small samples, faster learning speed, and less human intervention. Thus, in this paper, ELM is utilized to distinguish the bearing condition.

The rest of this paper is organized as follows. Section 2 proposes the feature extraction method based on DTCWPT and IMPE. Section 3 presents the feature dimension reduction method based on LLTSA. Section 4 briefly introduces the ELM classifier. Section 5 illustrates the detailed procedures of the proposed diagnosis method. In Section 6, the proposed method is applied to rolling bearing experimental data and some comparisons are made. Finally, conclusions are drawn in Section 7.

## 2. Feature Extraction Based on DTCWPT and IMPE

Fault diagnosis for rolling bearings is comprised of feature extraction and pattern recognition. Feature extraction is the most important part during the fault diagnosis, because the bearing conditions are identified according to the extracted features. Aiming to take the advantages of DTCWPT in processing the nonstationary and nonlinear signal and meanwhile utilize the capability of IMPE in characterizing the property of signal, these two methods are combined together to extract the feature information from the bearing signal.

*2.1. A Brief View of DTCWPT.* DTCWPT is an enhancement to the traditional discrete wavelet packet transform (WPT). In the decomposition and the reconstruction process of DTCWPT, two parallel WPTs with different low pass and high pass filters in each level are utilized. These can be, respectively, regarded as the real tree and the imaginary tree in DTCWPT algorithm. And information complementation

can be achieved in the process of signal processing [20]. The decomposition process of DTCWPT is implemented through a set of low pass and high pass filters recursively as follows.

Real tree decomposition is as follows:

$$\begin{aligned} c_{l+1,2N}^{\text{Re}}(k) &= \sum_m h_0(m-2k) c_{l,N}^{\text{Re}}(m), \\ c_{l+1,2N+1}^{\text{Re}}(k) &= \sum_m h_1(m-2k) c_{l,N}^{\text{Re}}(m). \end{aligned} \quad (1)$$

Imaginary tree decomposition is as follows:

$$\begin{aligned} c_{l+1,2N}^{\text{Im}}(k) &= \sum_n g_0(n-2k) c_{l,N}^{\text{Im}}(n), \\ c_{l+1,2N+1}^{\text{Im}}(k) &= \sum_n g_1(n-2k) c_{l,N}^{\text{Im}}(n), \end{aligned} \quad (2)$$

where  $h_0$  and  $h_1$ , respectively, represent the low pass and the high pass filters used by WPT of the real tree, while  $g_0$  and  $g_1$  are the low pass and the high pass filters used by WPT of the imaginary tree.  $c_{l,N}^{\text{Re}}$  and  $c_{l,N}^{\text{Im}}$ , respectively, denote the coefficients in the real tree and the imaginary tree at the  $l$ th level,  $N$ th node. When level  $l = 0$ , coefficients  $c_{0,N}^{\text{Re}}$  and  $c_{0,N}^{\text{Im}}$  are the original signal  $x(t)$ ; namely,  $c_{0,0}^{\text{Re}} = c_{0,0}^{\text{Im}} = x(t)$ . The decomposition process of DTCWPT is illustrated in Figure 1.

The corresponding reconstruction operation of DTCWPT is as follows.

Real tree reconstruction is as follows:

$$\begin{aligned} c_{l,N}^{\text{Re}}(k) &= \sum_m \tilde{h}_0(k-2m) c_{l+1,2N}^{\text{Re}}(m) \\ &+ \sum_m \tilde{h}_1(k-2m) c_{l+1,2N+1}^{\text{Re}}(m). \end{aligned} \quad (3)$$

Imaginary tree reconstruction is as follows:

$$\begin{aligned} c_{l,N}^{\text{Im}}(k) &= \sum_n \tilde{g}_0(k-2n) c_{l+1,2N}^{\text{Im}}(n) \\ &+ \sum_n \tilde{g}_1(k-2n) c_{l+1,2N+1}^{\text{Im}}(n), \end{aligned} \quad (4)$$

where  $\tilde{h}_0$  and  $\tilde{h}_1$ , respectively, represent the low pass and the high pass reconstruction filters used by WPT of the real tree, while  $\tilde{g}_0$  and  $\tilde{g}_1$  denote the low pass and the high pass reconstruction filters used by WPT of the imaginary tree.

## 2.2. Background of PE, MPE, and IMPE

**2.2.1. Permutation Entropy.** The permutation entropy (PE) was proposed to detect the dynamic changes of time series based on comparison of neighboring values of time series [13]. The calculation steps of PE are described as follows.

Given a time series  $Y = \{y_1, y_2, \dots, y_N\}$  with the length of  $N$ ,  $h$  dimensional delay embedding vector at the moment  $t$  can be constructed as  $Y_t^{h,q} = \{y_t, y_{t+q}, \dots, y_{t+(h-1)q}\}$  ( $t = 1, 2, \dots, N - (h-1)q$ ), where  $h$  represents the embedding dimension and  $q$  is the time delay. It is thought that  $Y_t^{h,q}$  has a permutation type  $\pi_{m_0 m_1 \dots m_{h-1}}$  if it satisfies

$$y_{t+m_0 q} \leq y_{t+m_1 q} \leq \dots \leq y_{t+m_{h-1} q}, \quad (5)$$

where  $0 \leq m_i \leq h-1$  and  $m_i \neq m_j$ .

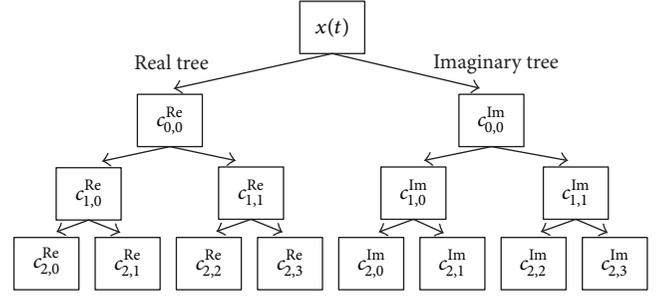


FIGURE 1: Decomposition process of DTCWPT.

There are  $h!$  kinds of different permutation types for  $h$  dimensional vector. For each permutation type  $\pi \in T$  ( $T$  denotes the set of all permutation types),  $p(\pi)$  demonstrates the relative frequency as follows:

$$\begin{aligned} p(\pi) &= \frac{\text{Number} \{t \mid 1 \leq t \leq N - (h-1)q, Y_t^{h,q} \text{ has type } \pi\}}{N - (h-1)q}. \end{aligned} \quad (6)$$

Then PE of time series  $Y$  is calculated as follows:

$$\text{PE}(Y, h, q) = - \sum_{\pi \in T} p(\pi) \ln(p(\pi)). \quad (7)$$

**2.2.2. Multiscale Permutation Entropy.** The multiscale permutation entropy (MPE) is defined as the PE set of time series at different scales. Considering the time series  $X = \{x_1, x_2, \dots, x_N\}$  with the length of  $N$ , the computational procedures of MPE are as follows.

(1) The original time series is firstly divided into several coarse-grained series  $y_j^{(\tau)}$  according to (8) and the schematic of this procedure is shown in Figure 2:

$$y_j^{(\tau)} = \frac{1}{\tau} \sum_{i=(j-1)\tau+1}^{j\tau} x_i \quad \left(1 \leq j \leq \frac{N}{\tau}\right), \quad (8)$$

where  $\tau$  denotes the scale factor.

(2) The PE of each coarse-grained series is calculated based on (6) and (7) and then plotted as a function of the scale factor  $\tau$ , which can be expressed as follows:

$$\text{MPE}(X, h, q, \tau) = \text{PE}(y_j^{(\tau)}, h, q). \quad (9)$$

**2.2.3. Improved Multiscale Permutation Entropy.** From Figure 2, it can be found that the coarse-grained procedure in MPE method can be considered as the procedure of averaging the original time series within a  $\tau$ -length window and then downsampling by a scale factor of  $\tau$ . However, the imprecise and unreliable results may occur in the process of downsampling at a certain scale [14]. To overcome the drawback of MPE, IMPE algorithm is proposed, and the calculation steps are as follows.

(1) For a defined scale factor  $\tau$ , the original time series is divided into  $\tau$  different coarse-grained series  $z_i^{(\tau)} =$

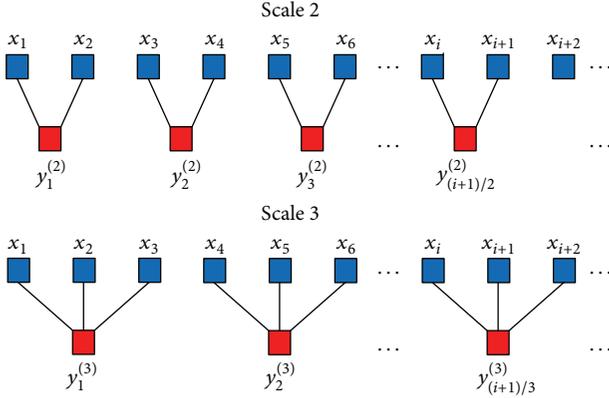


FIGURE 2: The schematic of the coarse-grained procedure for scale factors  $\tau = 2$  and  $\tau = 3$ .

$\{y_{i,1}^{(\tau)}, y_{i,2}^{(\tau)}, \dots\}$  ( $i = 1, 2, \dots, \tau$ ) based on the following equation:

$$y_{i,j}^{(\tau)} = \frac{\sum_{f=0}^{\tau-1} x_{f+i+\tau(j-1)}}{\tau}. \quad (10)$$

Then the  $\tau$  different coarse-grained series  $z_i^{(\tau)}$  ( $i = 1, \dots, \tau$ ) corresponding to the scale factor  $\tau$  are considered in IMPE algorithm, while, in MPE algorithm, only  $z_1^{(\tau)}$  is taken into account.

(2) Calculate the PE of each coarse-grained series  $z_i^{(\tau)}$  ( $i = 1, \dots, \tau$ ) corresponding to the scale factor  $\tau$  separately. Then, IMPE could be obtained based on the average value of PE:

$$\text{IMPE}(X, h, q, \tau) = \frac{1}{\tau} \sum_{i=1}^{\tau} \text{PE}(z_i^{(\tau)}, h, q). \quad (11)$$

**2.3. Feature Extraction.** Usually, the collected signals of the bearings with local defect are complicated and the interference between the components in the signal is inevitable. Besides, the differences among the original signals of the bearings in various operating conditions may be subtle. These factors will result in the difficulties of feature information extraction. Then the signal processing procedure combining DTCWPT with IMPE is presented to address this issue.

As a useful tool for signal processing, DTCWPT is suitable for analyzing the complicated bearing signals. The original signal could be decomposed into several subband signals using DTCWPT, and the subband signal will be simpler than the original signal. Then the interference between the components in each subband signal will be slighter than that in the original signal. And the hidden features in the original signal will be easier to be discovered by analyzing the subband signals. Therefore, DTCWPT is regarded as a preprocessing technology to analyze the original signal. And the IMPE algorithm, which can effectively evaluate the complexity and detect the dynamic changes of the signal, is used in the subsequent analysis process.

After performing DTCWPT on the original signal, each node of the wavelet packet coefficients is reconstructed at

a single level and the corresponding subband signals could be obtained. Then IMPE is further used to calculate the PE values of each subband signal at different scales. If the decomposition level of DTCWPT is  $l$  and the scale factor of IMPE is  $\tau$ , then the number of the obtained subband signal is  $2^l$  and the number of the calculated PE values of each subband signal is  $\tau$ . Therefore,  $2^l \times \tau$  PE values could be obtained for every original signal, and the constructed feature vectors based on these PE values could be used to comprehensively reflect the differences of the signals under different bearing conditions.

### 3. LLTSA for Dimension Reduction

For the classifier, the large amount of features will not only increase the computational complexity but also lead to a decline in the classification accuracy. Therefore, the dimension of the obtained feature vectors needs to be reduced. The objective of the dimension reduction in fault diagnosis mainly contains two aspects: (1) removing the disturbed and redundant information within the high dimensional feature vectors; (2) increasing the separability of the samples, namely, making different-class samples far from each other while making same-class samples close to each other.

Based on the previous analysis, in this paper, the LLTSA algorithm is utilized to compress the original vectors into the new vectors with a lower dimension. The basic idea of LLTSA is to use the tangent space in the neighborhood of a data point to represent the local geometry of the feature. Then the local manifold structures of space are lined up to construct the global coordinates [21].

Given a dataset  $X_{\text{ORG}} = [x_{\text{org}1}, x_{\text{org}2}, \dots, x_{\text{org}N}]$  from Euclidean space  $R^m$ , generally,  $X_{\text{ORG}}$ , an underlying  $d$  dimensional nonlinear manifold  $M^d$  ( $M^d \subset R^d$ ) embedded in  $R^m$  ( $d < m$ ) exists. Then the target problem for LLTSA is to find transformation matrix  $A$  which can map the original set  $X_{\text{ORG}} = [x_{\text{org}1}, x_{\text{org}2}, \dots, x_{\text{org}N}]$  in  $R^m$  to the set  $Y = [y_1, y_2, \dots, y_N]$  in  $R^d$ ; that is,

$$Y = A^T X_{\text{ORG}} H_N, \quad (12)$$

where  $H_N = I - ee^T/N$  represents the centering matrix,  $I$  is the identifying matrix,  $e$  is  $N$  dimensional column vector of all ones, and  $N$  denotes the number of the data.

The LLTSA algorithm procedures are described as follows.

(1) *PCA Projection.* Project the raw dataset  $X_{\text{ORG}}$  into the PCA subspace by throwing away the minor components. In order to make it clear,  $X$  is used to represent the dataset in the PCA subspace in the following steps and  $A_{\text{pca}}$  is applied to denote the transformation matrix of PCA.

(2) *Determining the Neighborhood Size.* The Euclidean distance matrix for all data points is constructed, and the  $k$  nearest neighbors  $x_{i_j}$  ( $j = 1, 2, \dots, k$ ) belonging to the same class of point  $x_i$  ( $i = 1, 2, \dots, N$ ) are obtained by analyzing the distance matrix.

(3) *Extracting Local Information.* Compute the tangent space matrix  $V_i$  composed of  $d$  eigenvectors of  $X_i H_k$  ( $X_i = [x_{i_1}, x_{i_2}, \dots, x_{i_k}]$ ) corresponding to its  $d$  largest eigenvalues, and  $H_k = I - ee^T/k$ .

(4) *Constructing Alignment Matrix.* Form matrix  $B$  by locally summing as follows:

$$B(I_i, I_i) \leftarrow B(I_i, I_i) + W_i W_i^T \quad (i = 1, 2, \dots, N), \quad (13)$$

where the initialization  $B = 0$ , and  $I_i = \{i_1, i_2, \dots, i_k\}$  denotes the set of indices for the  $k$  nearest neighbors of  $x_i$  and  $W_i = H_k(I - V_i V_i^T)$  ( $i = 1, 2, \dots, N$ ).

(5) *Computing the Maps.* Compute the eigenvectors and the eigenvalues for the generalized eigenvalue problem as

$$X H_N B H_N X^T a = \lambda X H_N X^T a. \quad (14)$$

Then the eigenvector  $\alpha_1, \alpha_2, \dots, \alpha_d$  ordered according to the eigenvalues  $\lambda_1 < \lambda_2 < \dots < \lambda_d$  could be obtained, and  $A_{LLTSA} = (\alpha_1, \alpha_2, \dots, \alpha_d)$ . Thus, the ultimate transformation matrix is as follows:  $A = A_{pca} A_{LLTSA}$  and  $X \rightarrow Y = A^T X_{ORG} H_N$ .

Due to the good clustering performance of LLTSA, the  $d$  dimensional eigenvector set  $Y$  outputted by LLTSA can be served as the input vectors of the classifier for the pattern recognition.

#### 4. ELM Classifier

The ELM proposed by Huang et al. [22] is a new and fast machine learning technique based on single layer feed-forward networks. A brief description of ELM is as follows.

Given a training dataset with  $N$  samples  $\{x_i, y_i\}_{i=1}^N$ , where  $x_i \in R^d$  is the input vector and  $y_i \in R^s$  stands for the target vector, the output of ELM with  $L$  hidden neurons can be represented as

$$\sum_{i=1}^L b_i g(\omega_i \cdot x_j + \beta_i) = o_j \quad (j = 1, 2, \dots, N), \quad (15)$$

where  $g(\cdot)$  is the activation function,  $\omega_i$  is the vector of the link weights between the  $i$ th hidden neuron and the input layer,  $b_i$  is the vector of the link weights between the  $i$ th hidden neuron and the output layer,  $\beta_i$  indicates the bias of the  $i$ th hidden neuron, and  $o_j$  is the output vector of the  $j$ th input sample. If ELM can approximate these samples without error, then

$$\sum_{i=1}^L b_i g(\omega_i \cdot x_j + \beta_i) = y_j \quad (j = 1, 2, \dots, N). \quad (16)$$

And (16) can be rewritten as

$$HB = Y, \quad (17)$$

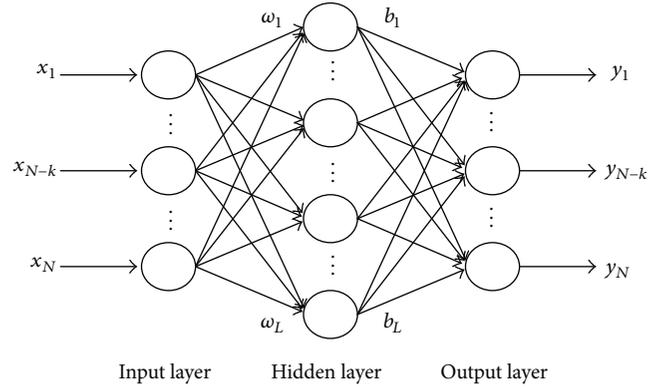


FIGURE 3: The structure of ELM.

where  $H$  denotes the output matrix of the hidden layer and can be expressed as

$$H = \begin{pmatrix} g(\omega_1 \cdot x_1 + \beta_1) & \cdots & g(\omega_L \cdot x_1 + \beta_L) \\ \vdots & \ddots & \vdots \\ g(\omega_1 \cdot x_N + \beta_1) & \cdots & g(\omega_L \cdot x_N + \beta_L) \end{pmatrix}_{N \times L} \quad (18)$$

and  $B = [b_1, b_2, \dots, b_k]^T$  is the matrix of the link weights from the hidden layer to the output layer, while  $Y = [y_1, y_2, \dots, y_N]^T$  is the matrix of the target vectors. Typically,  $B$  can be determined by the Moore-Penrose (MP) generalized inverse of  $H$ :

$$B = H^* T. \quad (19)$$

Then, utilizing the MP inverse method, the ELM generalization performance can be achieved. The structure of ELM is displayed in Figure 3.

#### 5. The Proposed Fault Diagnosis Method

Based on the advantages of DTCWPT, IMPE, LLTSA, and ELM, a novel bearing fault diagnosis method is proposed in this paper, and the flow chart of this method is shown in Figure 4. The detailed procedures are described as follows.

(1) Process the collected samples using DTCWPT and acquire the corresponding subband signals. Considering the tradeoff between the classification accuracy and the computational burden, without loss of generality, the decomposition level of DTCWPT is set to 2 in this study. Then each sample is decomposed into four subband signals after performing DTCWPT.

(2) Apply IMPE algorithm to calculate the PE values of the obtained subband signals at different scales. Before using IMPE, four parameters including the embedding dimension  $h$ , the length of signal  $N$ , the time delay  $q$ , and the scale factor  $\tau$  need to be set. Since  $h$  determines the number of accessible states  $h!$ , the estimation of PE relies heavily on the selected embedding dimension. If the dimension is too small, the scheme will not work because there are too few distinct states. When the dimension is too large, it will lead to being

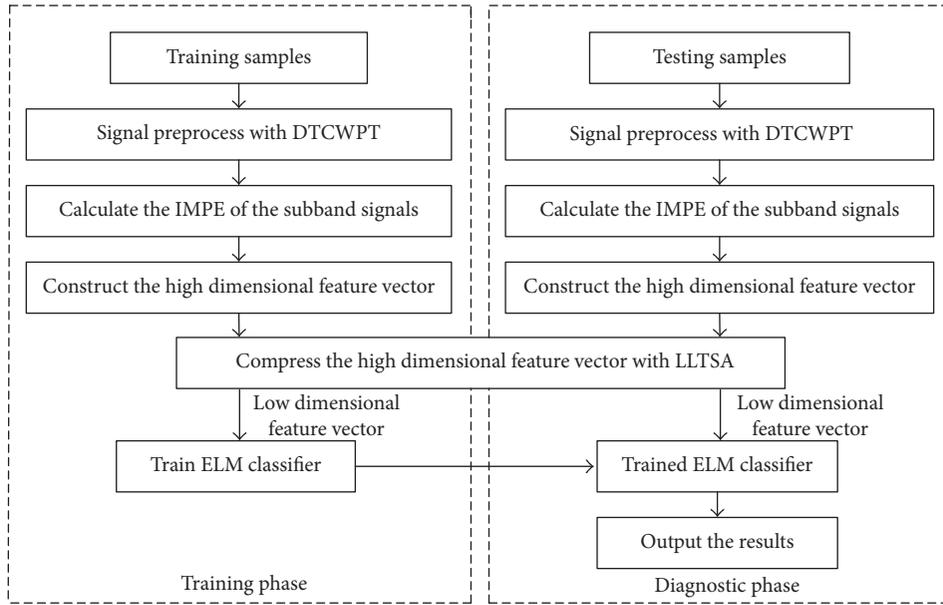


FIGURE 4: Flow chart of the proposed method.

time-consuming. To evaluate the complexity of the signal, the embedding dimension  $h$  is often chosen by tradeoff between the information loss and the computational burden. In this paper,  $h$  is set to 4. The signal length  $N$  also influences the estimation of PE. It is noticeable that  $N$  should satisfy the criterion  $N \geq 5h!$  which is recommended in literature [23] to obtain a reliable statistics. However, a too large value of  $N$  will decrease the computational efficiency. The signal with 1024 points is enough to obtain a reliable result. Therefore, we set  $N = 1024$  in this study. The time delay  $q$  has little effect on the calculated result; here we set  $q = 1$ . As for the selection of the scale factor  $\tau$ , when  $\tau$  is too small, the acquired feature information from the signal will be insufficient. On the other hand, if the scale factor  $\tau$  is too large, the obtained PE values in large scales will be unstable. Taking these constraints into consideration, based on the criterion  $\tau \leq N/(h+1)!$  proposed in literature [14] and the selected  $N$  and  $h$ , the scale factor  $\tau$  is set to 8 in this paper.

(3) Combine the calculated IMPE of each subband signal and construct the feature vector for each sample. Since each sample is decomposed into four subband signals, and the number of calculated PE is 8 for each subband signal, the dimension of the constructed feature vector is 32; namely, 32 features are extracted for each sample.

(4) Utilize LLTSA algorithm to compress the dimension of the constructed feature vectors and acquire the new feature vectors with lower dimension. In LLTSA algorithm, two parameters including the neighborhood size  $k$  and the intrinsic dimension  $d$  need to be adjusted. If parameter  $k$  is too small, LLTSA cannot well discover the intrinsic structure information of the high dimensional feature vectors. Contrarily, LLTSA will lose the ability of nonlinear dimension reduction. As for the intrinsic dimension  $d$ , if this parameter is chosen larger than what it really is, much redundant information will be preserved. When it is selected

smaller, useful information of the feature vectors will be thrown out during the dimension reduction. For LLTSA, there is an approximate linear relation between the optimal neighborhood size  $k$  and the intrinsic dimension  $d$  [24]. Hence, we choose  $k = d$  according to this linear relationship in this paper, using the cross validation method to determine the intrinsic dimension  $d$ .

(5) Feed the acquired new feature vectors into the ELM classifier for training and testing and distinguish the bearing condition automatically. Compared with some classic classifiers, ELM requires less human interventions. Only the number of the hidden neurons needs to be selected. Generally, as long as the number of hidden neurons is larger than 20, the classification accuracy of ELM will remain stable [25]. Therefore, the number of the hidden neurons is set to 20 in this paper.

## 6. Analysis on Experimental Data

**6.1. Experimental Data Description.** The experimental data from Case Western Reserve University are applied to verify the proposed method [26]. Figure 5 displays the experimental system, which consists of an electric motor, a torque transducer/encoder, and a dynamometer. The SKF6205-2RS deep groove ball bearing supporting the shaft at the drive end was used in the test. The rolling bearings were seeded with single point defects whose diameters were 0.1778 mm, 0.3556 mm, and 0.5334 mm, respectively, using the electric discharge machining technology. The defects were set on the inner race, the outer race, and the rolling element, respectively. An accelerometer was mounted on the motor housing to collect the vibration signals of the bearings under three different kinds of fault types as well as normal condition. The rotating speed of the motor was 1797 r/min and the sampling frequency was 12000 Hz. Every fault type contains

TABLE 1: The detailed description of the experimental datasets.

Fault type	Fault diameter (mm)	Fault degree	Number of training samples	Number of testing samples	Class label
Normal	0		10	40	1
Inner race	0.1778	Slight	10	40	2
	0.3556	Medium	10	40	3
	0.5334	Severe	10	40	4
Outer race	0.1778	Slight	10	40	5
	0.3556	Medium	10	40	6
	0.5334	Severe	10	40	7
Rolling element	0.1778	Slight	10	40	8
	0.3556	Medium	10	40	9
	0.5334	Severe	10	40	10

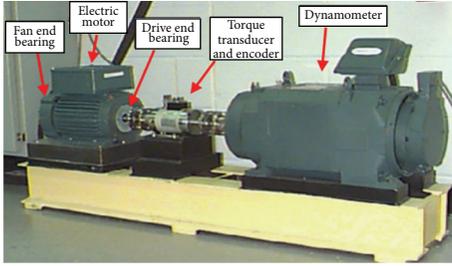


FIGURE 5: The rolling bearing experimental system.

three kinds of fault degrees with respect to different defect diameters. Therefore, a ten-condition classification problem for the bearings is investigated in this paper. The collected vibration signals are divided into several nonoverlapping segments with the length of 1024 points, and each segment is a sample. Each bearing condition includes 50 samples, from which 10 samples are randomly selected to train the classifier, while the residual 40 samples are used for testing. The detailed description of the experimental datasets is displayed in Table 1. The time domain waveforms of the samples of the bearings, respectively, with the slight inner race fault (Slight-IRF), the medium inner race fault (Medium-IRF), the severe inner fault (Severe-IRF), the slight outer race fault (Slight-ORF), the medium outer race fault (Medium-ORF), the severe outer race fault (Severe-ORF), the slight rolling element fault (Slight-REF), the medium rolling element fault (Medium-REF), and the severe rolling element fault (Severe-REF), as well as the normal condition, are shown in Figure 6, respectively.

**6.2. Results and Discussions.** Since the measured vibration signals of the bearings under different conditions represent the nonlinear and nonstationary characteristics, it is difficult to distinguish the different fault types and fault degrees only using the time domain waveforms in Figure 6. Therefore, it is very essential to perform an effective method to identify different operating conditions accurately. Then the proposed diagnosis method is applied.

Firstly, in order to reduce the interference among the components in the original sample and discover the hidden feature information more effectively, each sample is decomposed to 2 levels using DTCWPT. Then four subband signals containing different frequency band information could be obtained. For the sake of space, only the decomposition results of the samples with the slight inner race fault (Slight-IRF) are shown in Figure 7 as a representative.

According to the flow chart of the proposed diagnosis method indicated in Figure 1, after completing the signal decomposition and reconstruction, the IMPE algorithm is then utilized to extract the features at different scales from each subband signal for each sample. Figure 8 illustrates the IMPE values of subband signal 1, subband signal 2, subband signal 3, and subband signal 4 over 8 scales under 10 conditions. As shown in Figure 8, for each subband signal under different conditions, the divisibility among the PE values is high at some scales, while the differences of the PE values are not obvious at some scales. It is still unable to distinguish the different fault types with various fault degrees from the IMPE curves in Figure 8. Then the feature vector is constructed based on the acquired PE values of four subband signals for each sample. And a multifault classifier is applied to recognize the different bearing conditions. In this paper, the ELM classifier is used to achieve this purpose.

If the constructed feature vectors containing 32 PE values are directly taken as the inputs of the classifier, it will be time-consuming. Even worse, ELM cannot effectively distinguish the conditions of samples since feature vectors inevitably contain certain interference and redundancy information. Then LLTSA is further employed to compress the high dimensional feature vectors.

Before using LLTSA, an important problem about selecting the intrinsic dimension  $d$  of the original feature vectors needs to be addressed. In this paper, this parameter is determined using a fivefold cross validation method [27, 28]. That is, the 100 training samples are randomly divided into five equal-sized subsets. Each subset is validated on the ELM classifier that was trained using the other four subsets. The process was repeated 5 times; the accuracy rate of the classifier is then obtained by means of averaging the recorded accuracy

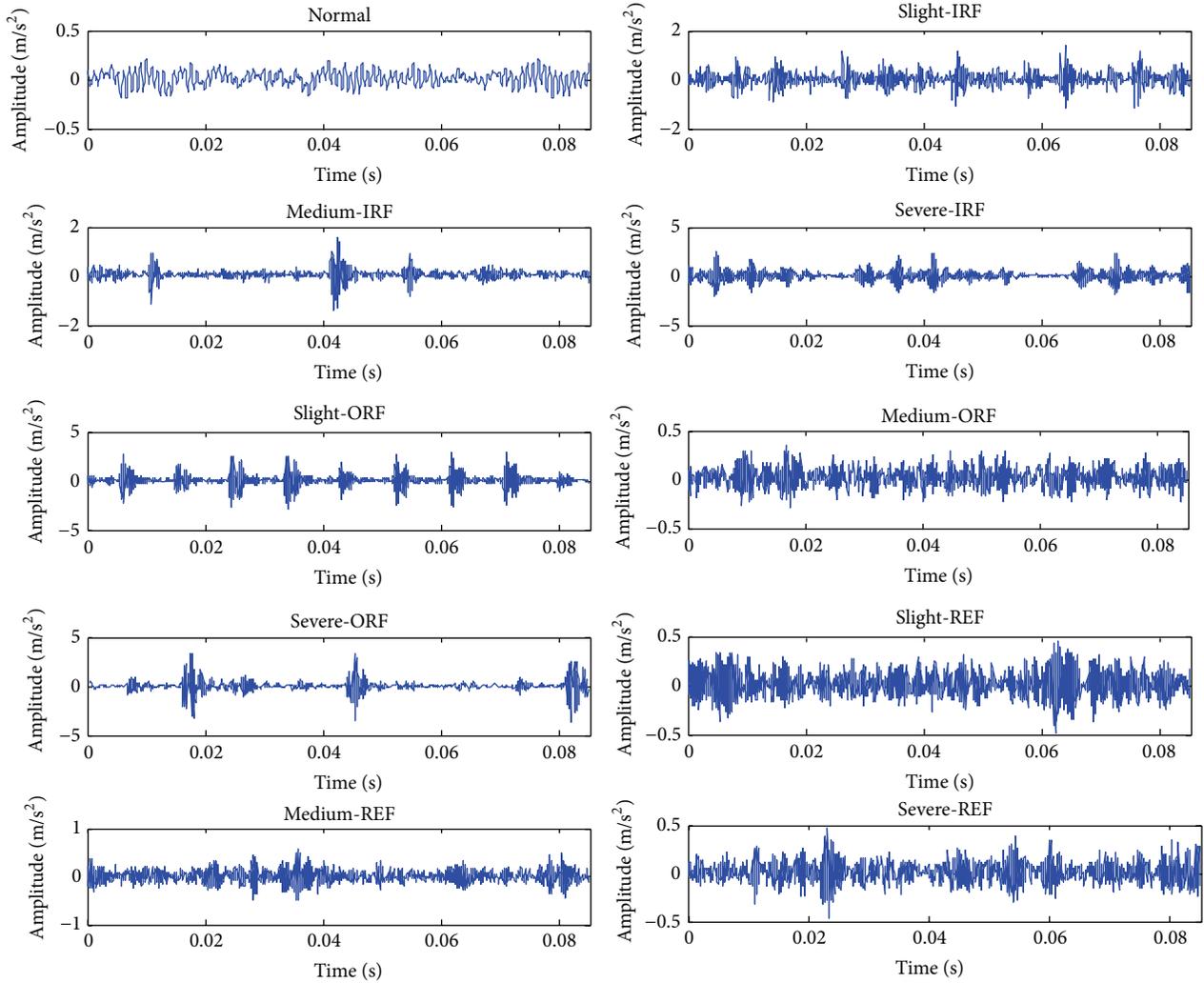


FIGURE 6: The time domain waveforms of the samples of the bearings under ten kinds of different conditions.

rate in each testing fold. Finally, choose parameter  $d$  which provides the best classification accuracy. In this paper, the intrinsic dimension  $d$  in the five-fold validation varies in the interval  $[3, m/2]$  with an incremental step size of 1, where  $m = 32$  denotes the dimension of the original feature vectors. Figure 9 shows the curve of the classification accuracy versus the intrinsic dimension. As indicated in Figure 9, the accuracy reaches 100% when the intrinsic dimension is larger than 10. In order to avoid information redundancy as far as possible, we select the intrinsic dimension  $d = 10$ ; then the neighborhood size  $k$  is set to 10 according to the approximate linear relationship between the intrinsic dimension and the neighborhood size.

After parameter selections, LLTSA is performed on the constructed feature vectors. Then the original high dimensional feature vectors are projected into a low dimensional space, based on which the new 10 dimensional feature vectors could be obtained. Then feed them into ELM for training and testing. After training the classifier with the 100 feature vectors of the training samples, the remaining 400 feature vectors of the testing samples are used to test the ELM

classifier. The classification results of the classifier are shown in Figure 10, where the red asterisks denote the ELM actual output classifications of the samples, while the blue squares represent the desired output classifications. The 100 samples on the left side and the 400 samples on the right side of the dotted line are, respectively, the training samples and the testing samples. It is suggested that, for each sample, the actual ELM output classification is consistent with the desired one. There is no misclassified sample, and the recognition accuracy achieves a perfect level of 100%. The proposed method obtains perfect classification results, which means that this method is exactly suitable and effective in bearing fault diagnosis.

In order to verify the advantage of IMPE, as a representative, a comparison is taken between IMPE and MPE by analyzing the 50 independent subband signals of the bearing with the slight inner race fault. The selected parameters in MPE algorithm are the same as those in IMPE algorithm. Figure 11 represents the mean values and the standard deviations of the PE values, using the IMPE and the MPE algorithms. The following conclusions can be drawn from

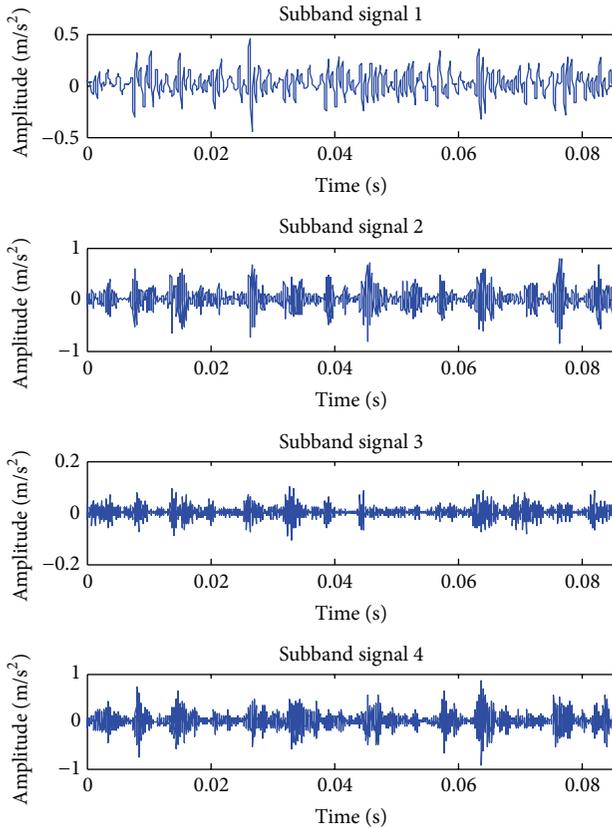


FIGURE 7: DTCWPT results of the sample with the slight inner race fault.

Figure 11. Firstly, the mean curves of the PE values derived from IMPE are really close to those derived from MPE. Secondly, compared with the MPE algorithm, the IMPE algorithm is able to get smaller standard deviations of the PE values. These conclusions can also be drawn through analyzing the subband signals of the bearings under the other conditions. It is indicated that the IMPE algorithm is more stable than the traditional MPE algorithm, which means that IMPE can provide a more accurate PE estimation on the nonlinear and nonstationary signals.

To further illustrate the advantage of IMPE, the feature vectors extracted by the processing method based on DTCWPT, MPE, and LLTSA are also fed into the ELM classifier to distinguish the bearing conditions. The actual ELM output classifications and the desired output classifications of the training and the testing samples are shown in Figure 12. On the right side of the dotted line, the locations of eight red asterisks are inconsistent with those of the blue squares. It is indicated that eight testing samples are misclassified and the classification accuracy is 98%. It can be easily observed from Figure 12 that two testing samples with Slight-IRF are misclassified as Medium-IRF, two testing samples with Slight-ORF are misclassified as Medium-IRF and Medium-ORF, a testing sample with Severe-ORF is misclassified as Medium-IRF, and three testing samples with Medium-REF are misclassified as Slight-REF and Medium-IRF, respectively.

The comparison results displayed in Figures 10 and 12 provide compelling evidence that IMPE can provide more accurate estimation of entropy values with higher distinguishability than MPE. These analysis results can be explained by the fact that when the MPE algorithm is used to analyze the short term series, the calculation points will be decreased exponentially as the scale factor is increased. It can not only give rise to the questionable and uncertain estimations of the entropy values but also increase the standard deviations of the features. However, the IMPE algorithm is able to avoid the drawbacks of MPE effectively and result in better classification accuracy.

To validate the necessity of the dimension reduction using LLTSA, the constructed original feature vectors without dimension reduction are adopted as the inputs of the ELM classifier for a comparison. The classification results are displayed in Figure 13, from which it can be seen that three testing samples with the rolling element fault are misclassified into the wrong fault degrees. The accuracy is 99.25%, which is lower than that of the method with dimension reduction. In the process of the dimension reduction, LLTSA can get the low dimensional sensitive feature vectors from the high dimensional feature vectors with interference and redundancies. Therefore, the recognition precision of ELM could be improved. It is indicated that the dimension reduction using LLTSA is of benefit for the bearing condition classification. Also, the necessity of this procedure is demonstrated at the same time.

In addition, in order to verify the superiority of the proposed feature extraction method based on DTCWPT, IMPE, and LLTSA, the calculated IMPE values of the original samples are directly taken as the input feature vectors of the ELM classifier. The 100 training samples and the 400 testing samples, as well as the selected parameters, remain the same as mentioned previously. The actual ELM output classification and the desired output classification of all the samples are shown in Figure 14, where 28 testing samples on the right side of the dotted line are misclassified. The recognition accuracy is 93%. It is shown that the extracted features of the samples directly using IMPE cannot completely reflect the distinctions of different bearing conditions. Thus, the obtained classification results of the ELM classifier are unsatisfied. This comparison demonstrates the superiority of the proposed feature extraction method which combines IMPE with DTCWPT and LLTSA due to the abilities of DTCWPT and LLTSA in restraining the interference among the components and highlighting the feature information of the samples.

Finally, the recognition accuracies of ELM, SVM, ANN, and KNNC using different feature extraction methods are compared. The training and the testing samples are the same for each comparison. And the feature vectors taken as the inputs of these classifiers are extracted by four different methods, respectively. The first method is the proposed method used in this paper, that is, the combination of DTCWPT, IMPE, and LLTSA (DTCWPT + IMPE + LLTSA). The second method utilizes MPE instead of IMPE and obtains the feature vectors based on DTCWPT, MPE, and LLTSA (DTCWPT + MPE + LLTSA). The third method extracts the feature vectors

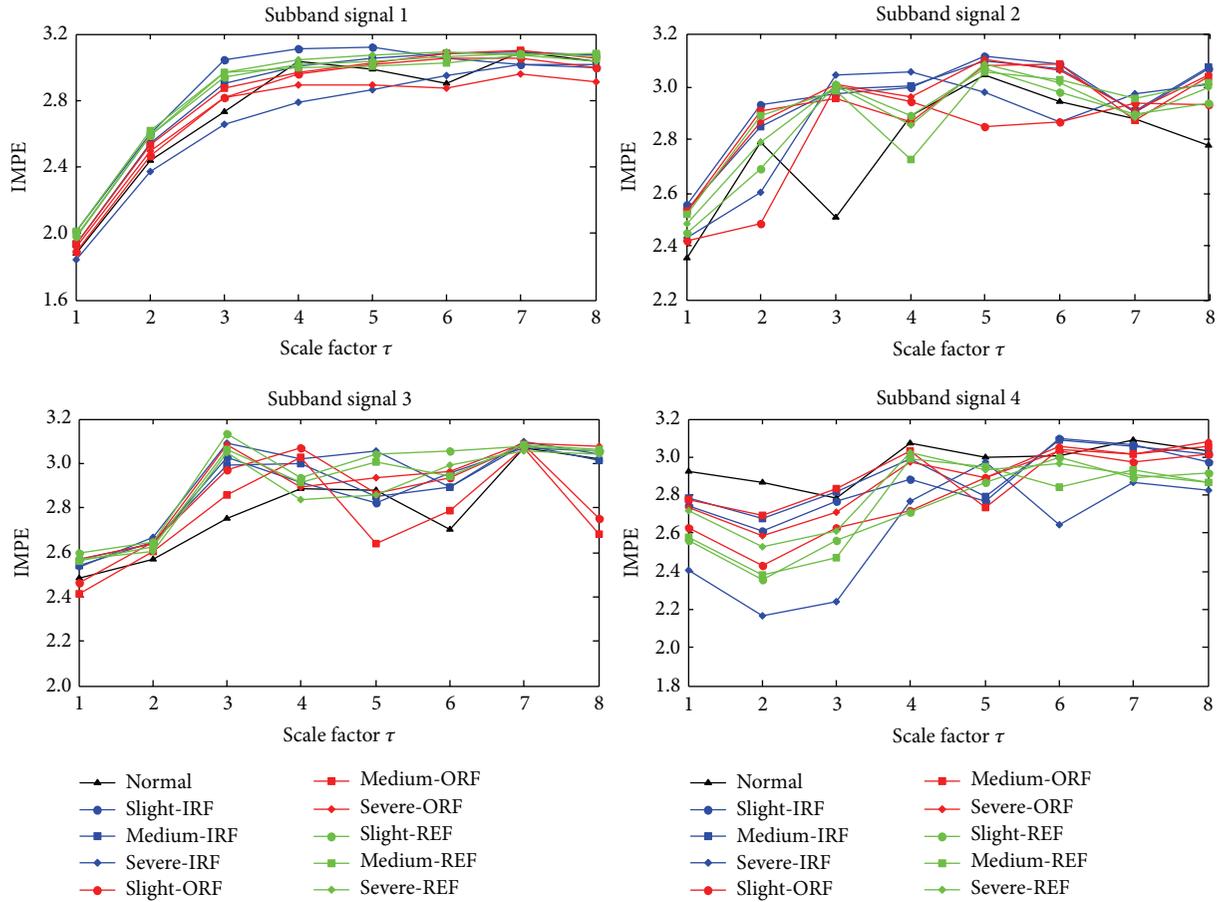


FIGURE 8: IMPE of each subband signal over 8 scales under 10 different conditions.

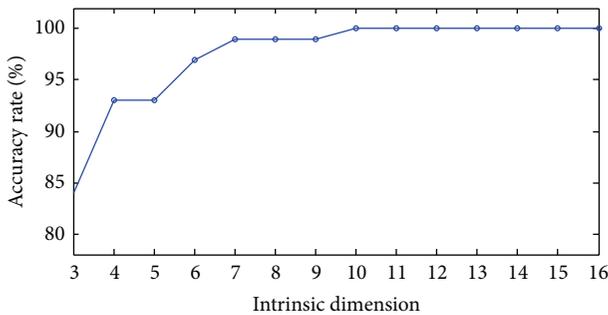


FIGURE 9: Curve of classification accuracy versus intrinsic dimension.

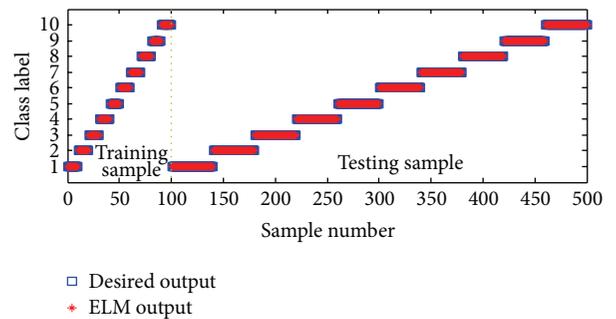


FIGURE 10: Classification results of the proposed method.

using DTCWPT and IMPE without LLTSA (DTCWPT + IMPE). The last method treats the calculated IMPE of the samples as the feature vectors (IMPE). The parameters of SVM are chosen as follows: the penalty factor  $C$  is set to 100 and the RBF kernel parameter  $\gamma$  is set to 0.01 [29]. The parameters of ANN are selected as follows: the number of the hidden neurons  $N = 20$ , the maximum number of the iterations  $I = 500$ , the learning rate  $\alpha = 0.1$ , and the training error  $e = 0.001$  [30]. The neighborhood number  $k$  of KNNC is set to 7 [31]. The classification results of ELM, SVM, ANN,

and KNNC using different feature extraction methods are shown in Table 2 and Figure 15. No matter what kind of method, it can be noted that the classification accuracy of ELM is higher than that of the other three classifiers. This verifies the advantage of ELM in classification performance.

It is suggested from Table 2 and Figure 15 that, using the feature vectors extracted through the first method, the testing accuracies of the classifiers are all higher than those of the classifiers using the feature vectors extracted via the other three kinds of methods. On one hand, for the average testing accuracies of the four classifiers, the first feature extraction

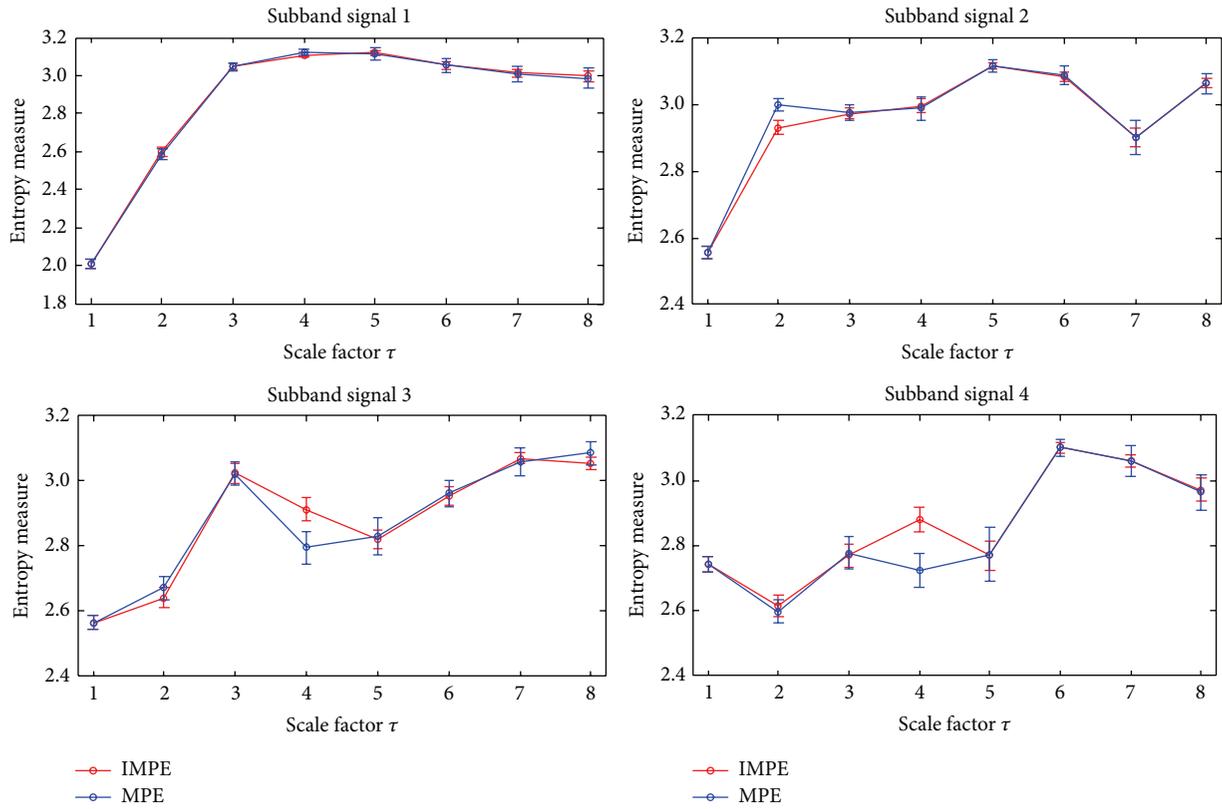


FIGURE 11: MPE and IMPE comparison results of the samples of the bearing with slight inner race fault.

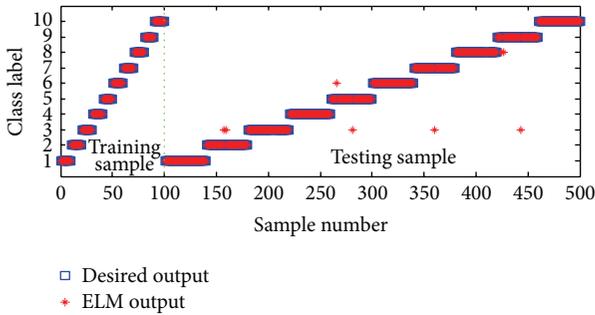


FIGURE 12: Classification results of the method based on DTCWPT, MPE, and LLTSA.

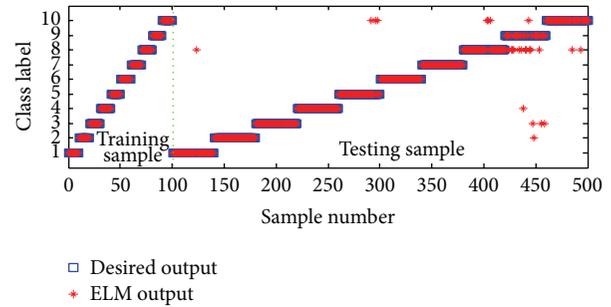


FIGURE 14: Classification results of the method based on IMPE.

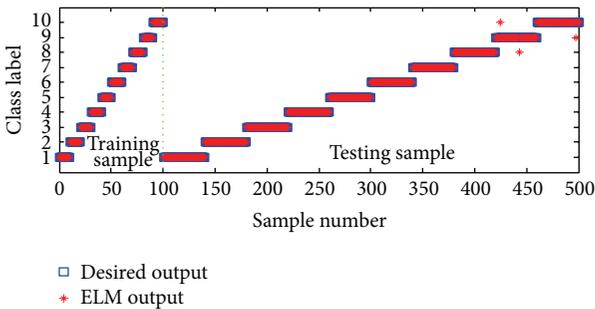


FIGURE 13: Classification results of the method based on DTCWPT and IMPE.

method is 2.19%, 0.62%, and 7.93%, better than the second, the third, and the last method, respectively, which in turn verifies the advantage of the presented feature extraction method based on DTCWPT, IMPE, and LLTSA.

### 7. Conclusions

IMPE is a recently proposed novel technique for evaluating the complexity and detecting the dynamic changes of time series. Its application in bearing fault diagnosis is firstly investigated in this work. And a novel fault diagnosis method for rolling bearings combining IMPE with DTCWPT, LLTSA, and ELM is proposed in this paper. Focusing on the nonlinear and nonstationary characteristics of the bearing vibration

TABLE 2: Classification results of different classifiers with feature vectors extracted by different methods.

Classifier	Testing accuracies with feature vectors extracted by different methods			
	DTCWPT + IMPE + LLTSA	DTCWPT + MPE + LLTSA	DTCWPT + IMPE	IMPE
ELM	100%	98%	99.25%	93%
SVM	99.75%	97.75%	99.25%	91.5%
ANN	99%	96%	98.25%	90.5%
KNNC	99.5%	97.75%	99%	91.5%
Average of four classifiers	99.56%	97.37%	98.94%	91.63%

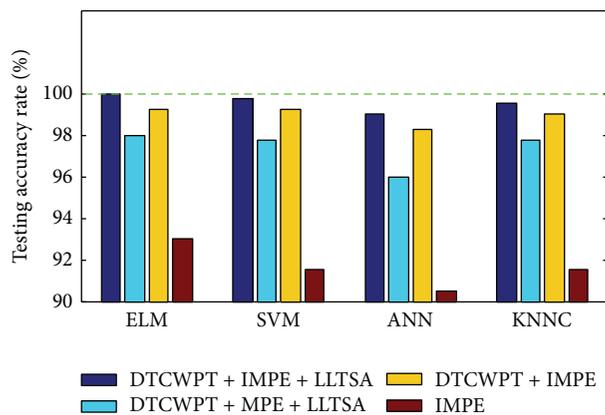


FIGURE 15: Classification results of different classifiers with feature vectors extracted by different methods.

signals, DTCWPT is employed to preprocess the signal and obtain the corresponding subband signals. IMPE is then taken as the feature extractor to calculate the PE values of each subband signal at different scales. To solve the dimension reduction problem of the constructed feature vectors, LLTSA is applied to compress the high dimensional vectors and sift out the principal sensitive features used to construct the new low dimensional vectors. Besides, the ELM classifier is adopted to implement the condition identification. For comparison purpose, the presented feature extraction method is compared with other methods. The comparison results indicate that the presented method is able to obtain the feature vectors with a higher divisibility. Also, the classification performance of the ELM classifier is also compared with other widely used classifiers, and the advantage of ELM is verified by the comparison results. The experimental data analysis results demonstrate that the proposed fault diagnosis method in this paper is suitable and effective in recognizing the different fault types and fault degrees of rolling bearings.

In the highly automated industry, since the proposed diagnosis method is data-driven without operators' experiences, it is much easier to be widely used. It is mentioned that the proposed method is a promising approach, which is not limited to rolling bearing fault diagnosis but also could be applied in fault diagnosis of other mechanical equipment.

To some extent, limited by the consumption of computer resources, the proposed diagnosis method may not be satisfactory enough in real time. In addition, only the constant working load is discussed in this paper. If the working load is dramatically changed, the accuracy and the efficiency of the proposed method may be influenced. Consequently, further studies will be focused on solving this problem.

## Competing Interests

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

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 51307058 and no. 51475164), the Natural Science Foundation of Hebei Province, China (no. 2014502052 and no. 2015502013), and the Fundamental Research Funds for the Central Universities (no. 2015ZD27 and no. 2015XS120).

## References

- [1] G. J. Tang, X. L. Wang, and Y. L. He, "Diagnosis of compound faults of rolling bearings through adaptive maximum correlated kurtosis deconvolution," *Journal of Mechanical Science and Technology*, vol. 30, no. 1, pp. 43–54, 2016.
- [2] X. Ding, Q. He, and N. Luo, "A fusion feature and its improvement based on locality preserving projections for rolling element bearing fault classification," *Journal of Sound and Vibration*, vol. 335, pp. 367–383, 2015.
- [3] J. Zhou, J. Xiao, H. Xiao, W. Zhang, W. Zhu, and C. Li, "Multi-fault diagnosis for rolling element bearings based on intrinsic mode permutation entropy and ensemble optimal extreme learning machine," *Advances in Mechanical Engineering*, vol. 6, Article ID 803919, 10 pages, 2014.
- [4] J. Q. Kang, F. Sun, and Y. B. Liu, "Fault diagnosis of the rolling bearing based on correlation dimension improved algorithm," *Chinese Journal of Scientific Instrument*, vol. 27, no. 6, pp. 404–406, 2006.
- [5] R. Yan and R. X. Gao, "Approximate entropy as a diagnostic tool for machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 824–839, 2007.
- [6] Y.-H. Pan, Y.-H. Wang, S.-F. Liang, and K.-T. Lee, "Fast computation of sample entropy and approximate entropy in

- biomedicine,” *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, pp. 382–396, 2011.
- [7] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate and sample entropy,” *American Journal of Physiology—Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.
- [8] M. Costa, A. L. Goldberger, and C.-K. Peng, “Multiscale entropy analysis of complex physiologic time series,” *Physical Review Letters*, vol. 89, no. 6, 2002.
- [9] L. Zhang, G. L. Xiong, H. S. Liu, H. J. Zou, and W. Z. Guo, “Bearing fault diagnosis using multi-scale entropy and adaptive neuro-fuzzy inference,” *Expert Systems with Applications*, vol. 37, no. 8, pp. 6077–6085, 2010.
- [10] C. Bandt and B. Pompe, “Permutation entropy: a natural complexity measure for time series,” *Physical Review Letters*, vol. 88, no. 17, pp. 1741021–1741024, 2002.
- [11] W. Aziz and M. Arif, “Multiscale permutation entropy of physiological time series,” in *Proceedings of the 9th International Multitopic Conference (INMIC '05)*, pp. 1–6, IEEE, Karachi, Pakistan, December 2005.
- [12] Y. B. Li, M. Q. Xu, Y. Wei, and W. H. Huang, “A new rolling bearing fault diagnosis method based on multiscale permutation entropy and improved support vector machine based binary tree,” *Measurement*, vol. 77, pp. 80–94, 2016.
- [13] J. Zheng, J. Cheng, and Y. Yang, “Multiscale permutation entropy based rolling bearing fault diagnosis,” *Shock and Vibration*, vol. 2014, Article ID 154291, 8 pages, 2014.
- [14] H. Azami and J. Escudero, “Improved multiscale permutation entropy for biomedical signal analysis: interpretation and application to electroencephalogram recordings,” *Biomedical Signal Processing and Control*, vol. 23, pp. 28–41, 2016.
- [15] J. D. Zheng, J. Cheng, and Y. Yang, “A rolling bearing fault diagnosis approach based on LCD and fuzzy entropy,” *Mechanism and Machine Theory*, vol. 70, pp. 441–453, 2013.
- [16] N. G. Kingsbury, “The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters,” in *Proceedings of the 8th IEEE Digital Signal Processing Workshop*, 1998.
- [17] I. Bayram and I. W. Selesnick, “On the dual-tree complex wavelet packet and  $M$ -band transforms,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2298–2310, 2008.
- [18] F. Li, B. P. Tang, and R. S. Yang, “Rotating machine fault diagnosis using dimension reduction with linear local tangent space alignment,” *Measurement*, vol. 46, no. 8, pp. 2525–2539, 2013.
- [19] H. L. Chen, B. Yang, D. Y. Liu et al., “Using blood indexes to predict overweight statuses: an extreme learning machine based approach,” *PLOS ONE*, vol. 10, no. 11, Article ID e0143003, 2015.
- [20] J. X. Qu, Z. S. Zhang, and T. Gong, “A novel intelligent method for mechanical fault diagnosis based on dual-tree complex wavelet packet transform and multiple classifier fusion,” *Neurocomputing*, vol. 171, pp. 837–853, 2016.
- [21] T. H. Zhang, J. Yang, D. L. Zhao, and X. L. Ge, “Linear local tangent space alignment and application to face recognition,” *Neurocomputing*, vol. 70, no. 7–9, pp. 1547–1553, 2007.
- [22] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [23] M. Matilla-García, “A non-parametric test for independence based on symbolic dynamics,” *Journal of Economic Dynamics & Control*, vol. 31, no. 12, pp. 3889–3903, 2007.
- [24] X. Q. Lu and Y. Yuan, “Hybrid structure for robust dimensionality reduction,” *Neurocomputing*, vol. 124, pp. 131–138, 2014.
- [25] X. He, H. L. Wang, J. H. Lu, and W. Jiang, “Analog circuit fault diagnosis method based on preferred wavelet packet and ELM,” *Chinese Journal of Scientific Instrument*, vol. 34, no. 11, pp. 2614–2619, 2013.
- [26] Bearing Data Center, Case Western Reserve University, 2006, <http://cseggroups.case.edu/bearingdatacenter/pages/download-data-file>.
- [27] Y. B. Li, M. Q. Xu, R. X. Wang, and W. H. Huang, “A fault diagnosis scheme for rolling bearing based on local mean decomposition and improved multiscale fuzzy entropy,” *Journal of Sound and Vibration*, vol. 360, Article ID 12638, pp. 277–299, 2016.
- [28] Y. Zhang, B. Li, W. Wang, T. Sun, X. Yang, and L. Wang, “Supervised locally tangent space alignment for machine fault diagnosis,” *Journal of Mechanical Science and Technology*, vol. 28, no. 8, pp. 2971–2977, 2014.
- [29] S. J. Dong, L. L. Chen, B. P. Tang, X. Y. Xu, Z. Y. Gao, and J. Liu, “Rotating machine fault diagnosis based on optimal morphological filter and local tangent space alignment,” *Shock and Vibration*, vol. 2015, Article ID 893504, 9 pages, 2015.
- [30] K. Zhao, C. Y. Wang, J. Hu et al., “Prostate cancer identification: quantitative analysis of T2-weighted MR images based on a back propagation artificial neural network model,” *Science China Life Sciences*, vol. 58, no. 7, pp. 666–673, 2015.
- [31] F. Li, J. X. Wang, B. P. Tang, and D. Q. Tian, “Life grade recognition method based on supervised uncorrelated orthogonal locality preserving projection and K-nearest neighbor classifier,” *Neurocomputing*, vol. 138, pp. 271–282, 2014.

## Research Article

# Calculation and Analysis of Permanent Magnet Eddy Current Loss Fault with Magnet Segmentation

**Bing Li and Ming Li**

*College of Engineering, Bohai University, Jinzhou 121013, China*

Correspondence should be addressed to Bing Li; [libinger521@163.com](mailto:libinger521@163.com)

Received 16 January 2016; Accepted 5 April 2016

Academic Editor: Wen Chen

Copyright © 2016 B. Li and M. Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates the problem of calculating and analyzing the effect of the permanent magnet eddy current loss fault due to magnet segmentation. Taking an interior permanent magnet synchronous motor with inverter supplied as an example, the rated power of motor was 2.2 kW. Three-dimensional finite-element model was firstly established based on finite-element software. Then, the model mesh and boundary conditions were handled specially; permanent magnet eddy current loss fault was calculated and analyzed theoretically with magnet segmentation from space harmonic and time harmonic, respectively. Finally, calculation results were compared and explained. A useful conclusion for permanent magnet synchronous motor design has been obtained.

## 1. Introduction

With the development of power electronic devices and the improvement of the motor control technology, the permanent magnet synchronous motor has attracted more and more attention for its advantages of high efficiency, wide speed regulation, and high power density. But with the increase of the power density, it is worthy of considering how to make the motor temperature rise in the range of allowable limit value. Reducing the temperature rise of the motor should start from two aspects: how to improve the cooling capacity of motor and how to reduce the loss of motor. Because of the high speed and large carrier frequency of the permanent magnet synchronous motor, the eddy current loss of the permanent magnet is large.

In order to reduce the eddy current loss of permanent magnet, researchers have conducted a lot of research [1–5]. Among the studies, the axial section method of permanent magnet is widely accepted. Based on the previous research, the eddy current loss of interior permanent magnet synchronous motor is studied in this paper. Taking an interior permanent magnet synchronous motor with a rated power 2.2 kW as an example, a three-dimensional (3D) finite-element model is established. Permanent magnet eddy current loss fault was calculated and analyzed theoretically

with magnet segmentation from space harmonic and time harmonic, respectively; at the same time, the finite-element method is verified by analytic method.

One of the main problems of NdFeB permanent magnet synchronous motor is thermal demagnetization, which is caused by the permanent magnet eddy current loss. In particular, the permanent magnet AC servomotor is mostly used in the fractional slot concentrated winding whose magnetic motive force (MMF) harmonic content is very rich [6]. Thus, reducing the eddy current loss of permanent magnet has attracted more and more attention, in which the axial section of the pole is used to reduce the eddy current loss of the permanent magnet [7–9]. It is widely adopted by the designers of the motor.

There are two reasons for permanent magnet to generate eddy current losses [10–12]. One is the slotted stator and stator winding distribution caused by uneven distribution of MMF and the space harmonic. The second is the nonsinusoidal time harmonic of the stator current caused by inverter power supply. Permanent magnet eddy current losses can be expressed as

$$P_{\text{mag}} = \sum_k \frac{J_k^2}{\sigma} dv. \quad (1)$$

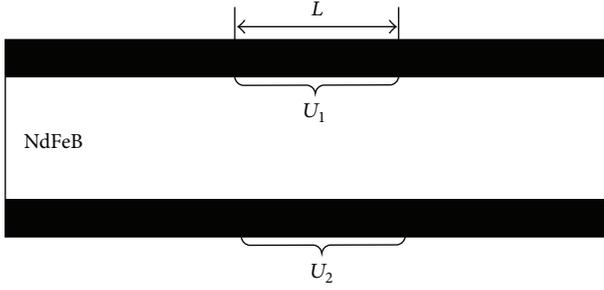


FIGURE 1: Surface potential difference of permanent magnet.

## 2. Analysis and Calculation of Eddy Current Loss of Permanent Magnet of Interior Permanent Magnet Motor

Suppose there is an infinite neodymium iron boron permanent magnet; the external magnetic field is parallel to the surface in the direction of the  $z$ -axis; the vortex at a certain point in the permanent magnet can be decomposed into two mutually perpendicular eddy current densities; it can be determined by

$$J_e = \frac{\sigma U}{L}, \quad (2)$$

where  $J_e$  is eddy current density,  $\sigma$  is electrical conductivity of permanent magnet materials,  $U$  is the voltage between the nodes in the calculation unit, and  $L$  is distance between nodes in a computing unit. As shown in Figure 1, suppose the potential difference between two points on the two sides of the permanent magnet block is  $U_1, U_2$ . For easy calculation, it can be set as  $U_1 = U_2 = U$ . The potential difference of the permanent magnet in the thickness direction is ignored again.

The total voltage of the 4 points on the surface of the permanent magnet which is composed of the two ends of the loop is

$$2U(t) = \frac{d\phi(t)}{dt} = \frac{AdB(t)}{dt}, \quad (3)$$

where  $A$  is loop area and  $B(t)$  is magnetic density in the circuit; it can be expressed as

$$B(t) = B_0 + B_a, \quad (4)$$

where  $B_0$  is static magnetic density and  $B_a$  is dynamic magnetic flux induced by stator armature current.  $B_a$  is regarded as a nonsinusoidal periodic function; the trigonometric series of  $B_a$  is decomposed into Fourier series:

$$B_a = \sum_{n=1}^{\infty} (a_n \cos n\omega t + b_n \sin n\omega t), \quad (5)$$

where  $a_n$  and  $b_n$  are the amplitude of the dynamic magnetic flux generated by the harmonic component of the armature current in the permanent magnet.

Then,

$$\begin{aligned} \frac{dB(t)}{dt} &= \frac{d(B_0 + B_a)}{dt} \\ &= \sum_{n=1}^{\infty} n\omega (b_n \cos n\omega t - a_n \sin n\omega t), \end{aligned} \quad (6)$$

$$2U(t) = \frac{AdB(t)}{dt} = A \sum_{n=1}^{\infty} n\omega (b_n \cos n\omega t - a_n \sin n\omega t). \quad (7)$$

The effective value of  $U(t)$  is

$$\begin{aligned} U &= \frac{\omega A}{2} \cdot \sqrt{\sum_{n=1}^{\infty} \frac{n^2 (a^2 + b^2)}{2}} \\ &= A\pi f \cdot \sqrt{\sum_{n=1}^{\infty} \frac{n^2 (a^2 + b^2)}{2}}. \end{aligned} \quad (8)$$

According to formula (7), the eddy current density in the permanent magnet is obtained:

$$\begin{aligned} J_e &= \frac{\omega f \sigma A}{L} \cdot \sqrt{\sum_{n=1}^{\infty} \frac{n^2 (a^2 + b^2)}{2}} \\ &= \pi f \sigma h_m \cdot \sqrt{\sum_{n=1}^{\infty} \frac{n^2 (a^2 + b^2)}{2}}, \end{aligned} \quad (9)$$

where  $h_m$  is permanent magnet magnetization direction length, so the eddy current loss power density of the permanent magnet can be obtained:

$$P_e = \frac{J_e^2}{\sigma} = \pi^2 f^2 \sigma h_m^2 \cdot \sum_{n=1}^{\infty} \frac{n^2 (a^2 + b^2)}{2}. \quad (10)$$

Eddy current loss density is

$$W_e = \pi^2 f \sigma h_m^2 \cdot \sum_{n=1}^{\infty} \frac{n^2 (a^2 + b^2)}{2}. \quad (11)$$

Eddy current loss density of permanent magnet is integral and then it can get eddy current loss in permanent magnet.

## 3. Effect of Pole Axial Section on the Eddy Current Loss Fault of Permanent Magnet Produced by Space Harmonic Generation

The basic parameters of an interior permanent magnet synchronous motor with a rated power 2.2kW in this paper are shown in Table 1. The 3D model of motor is constructed by using the finite-element analysis software. In order to save the computing resources, coupled with cycle magnetic distribution, a unit of the motor is modeled and calculated. Figure 2 shows 3D finite-element mesh model of a 2.2kW permanent magnet synchronous motor.

TABLE 1: Parameters of 2.2 kW motor and permanent magnet.

Rated power/kW	2.2
Phase number	3
Poles number	8
Stator outer diameter/mm and inter diameter/mm	155, 98
Stator slot number	36
Permanent magnet material and electrical conductivity/s·m <sup>-1</sup>	Nd-Fe-B, 6.9 × 10 <sup>5</sup>
Residual magnetic flux density/T	B <sub>r</sub> = 1.18
Permanent magnet size/mm	60 × 33 × 3

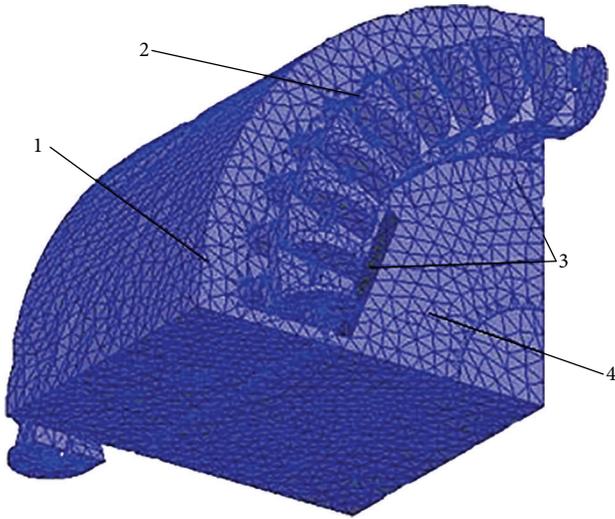


FIGURE 2: 3D sectional drawing of 2.2 kW permanent magnet synchronous motor (1 is the stator, 2 is the windings, 3 is the permanent magnet, and 4 is the rotor).

In order to keep the consistency for carrying on mesh each time, the 3D model is set up by the insulation boundary condition and the zero excitation. The eddy current loss of the permanent magnets which are divided into different segments is calculated under the condition of no load. The results are shown in Figure 3 and Table 2.

From Figure 3, the vortex line is cut off by the axial section of the permanent magnet, and it is formed locally in the section of the magnetic pole. In comparison with Table 2, the mean of eddy current loss increases with the number of magnetic poles.

#### 4. Effect of Pole Axial Section on the Eddy Current Loss of Permanent Magnet Produced by Time Harmonic Generation

Permanent magnet eddy current loss mainly is generated by the time harmonic generation. Literature [13] shows that eddy current loss is the biggest when the axial length of the permanent magnet is equal to 2.3 times the penetration depth. Permanent magnet of the penetration depth can be

TABLE 2: Eddy current losses in different segments of a magnet.

Segment	Maximum eddy density/A·m <sup>-2</sup>	Minimum eddy density/A·m <sup>-2</sup>	Mean value of eddy current loss/W
1	2.01	1.83 × 10 <sup>-9</sup>	0.29
2	0.49	2.0 × 10 <sup>-9</sup>	0.13
3	0.35	0.23 × 10 <sup>-9</sup>	0.067
4	0.68	0.78 × 10 <sup>-9</sup>	0.041
5	0.24	0.46 × 10 <sup>-9</sup>	0.021
6	0.16	0.68 × 10 <sup>-9</sup>	0.013

defined as the depth of magnetic field act on the inside of the permanent magnet. Magnetic field intensity decreases exponentially with the increase in depth. The penetration depth can be calculated as [14–17]

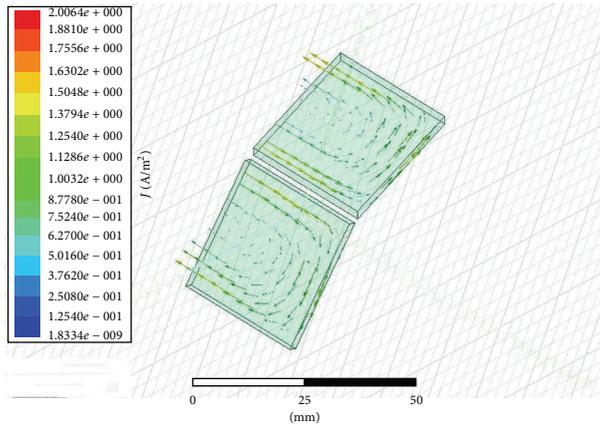
$$\delta = \frac{1}{\sqrt{\pi f \mu \sigma}}, \quad (12)$$

where  $\delta$  is the permanent magnet penetration depth,  $f$  is sinusoidal frequency,  $\mu$  is absolute permeability, and  $\sigma$  is conductivity.

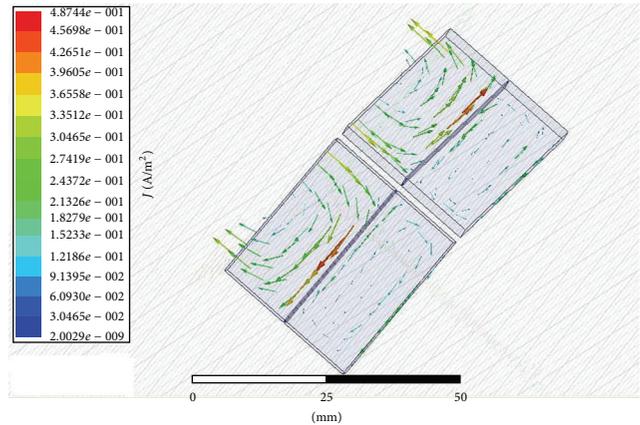
Figure 4 is the current waveform which is measured by inverter power supply test at rated frequency  $f = 100$  HZ. Figure 5 shows the harmonic amplitude after Fourier decomposition. From Figure 4, the harmonic amplitude of the current waveform is smaller at rated speed of the motor.

According to the conclusion of literature [13], this paper selects the 37th harmonic of current waveform whose permanent magnet penetration depth is about 11.2 mm corresponding to the 37th harmonic [17–21]. When permanent magnet eddy current loss was calculated by using the ANSOFT software, in order to ensure the synchronous motor work, the eddy current losses in the permanent magnets under the fundamental current and 37th times harmonic current were calculated firstly, and then permanent magnet eddy current loss generated by the fundamental was calculated. The difference between the two results is the permanent magnet eddy current loss by 37th harmonic.

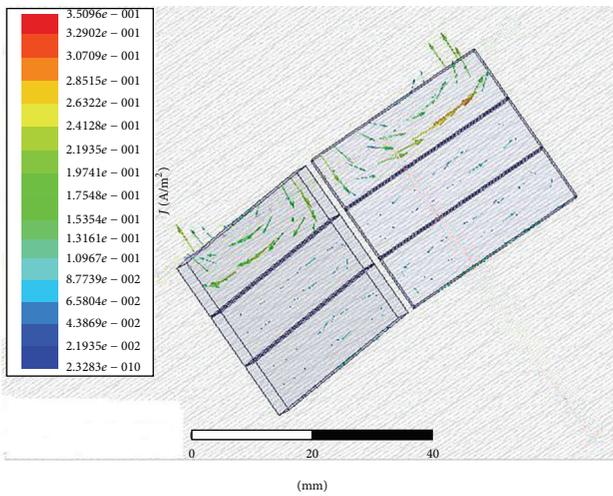
The eddy current losses of permanent magnets which are segmented into two and three are calculated, and the eddy current losses produced by the thirty-seventh time harmonic current source are shown in Table 3. From Figure 6, the eddy current losses in the different segments of the permanent magnet under the interaction of the fundamental wave and the 37th harmonic can be seen, only the fundamental wave and only the 37th harmonic. It can be seen that permanent magnet eddy current loss had not been reduced with segments numbers of the permanent magnet increasing from Table 3. When segments numbers of the permanent magnet are three, at this time the ratio of pole axial length and depth of penetration is 1.8; the eddy current loss generated by 37th times harmonic in permanent magnet not only failed to reduce but also increased.



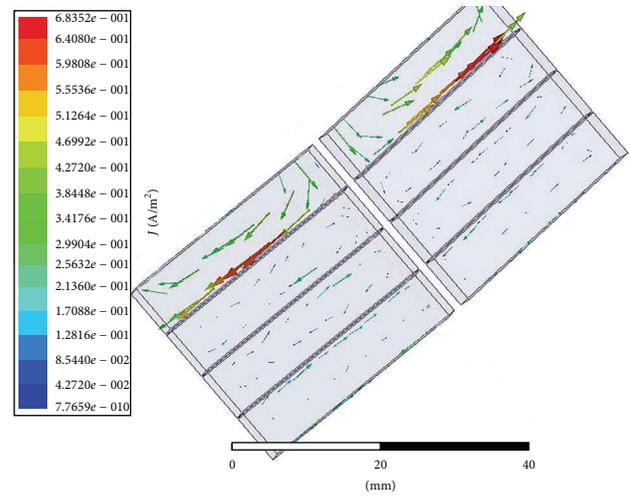
(a)



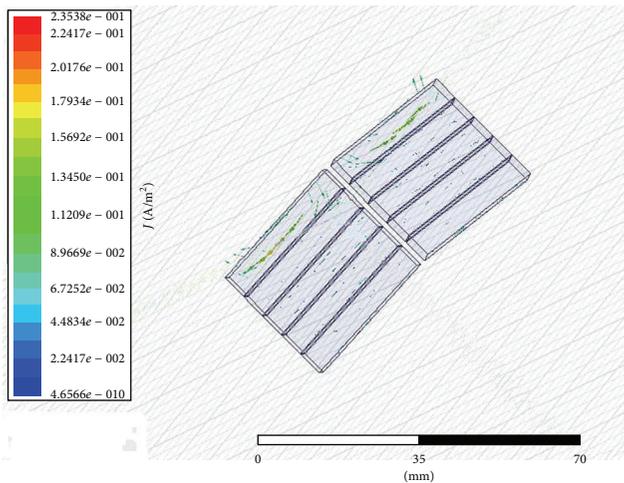
(b)



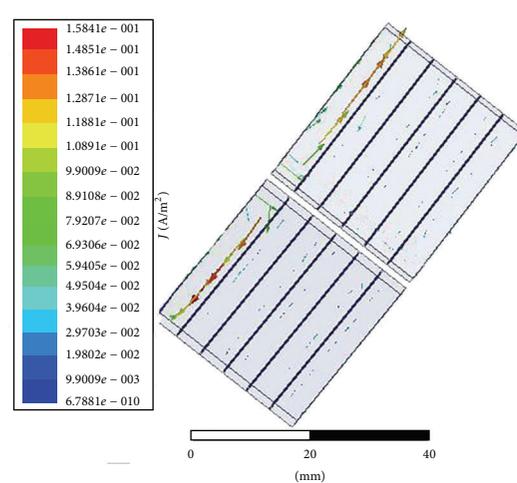
(c)



(d)



(e)



(f)

FIGURE 3: (a) Eddy current density of a permanent magnet. (b) Eddy current density of two-segment permanent magnet. (c) Eddy current density of three-segment permanent magnet. (d) Eddy current density of four-segment permanent magnet. (e) Eddy current density of five-segment permanent magnet. (f) Eddy current density of six-segment permanent magnet.

TABLE 3: Eddy current losses of permanent magnet under 37th harmonic currents at different stages.

Number of permanent magnet segments	The ratio of magnetic length and depth of penetration	Eddy current losses under the first and 37th harmonic excitation/W	Eddy current losses under the first harmonic excitation/W	Eddy current losses under the 37th harmonic excitation/mW
1	5.4	0.8233	0.8024	20.9
2	2.7	0.5187	0.5152	3.5
3	1.8	0.3607	0.3325	28.2

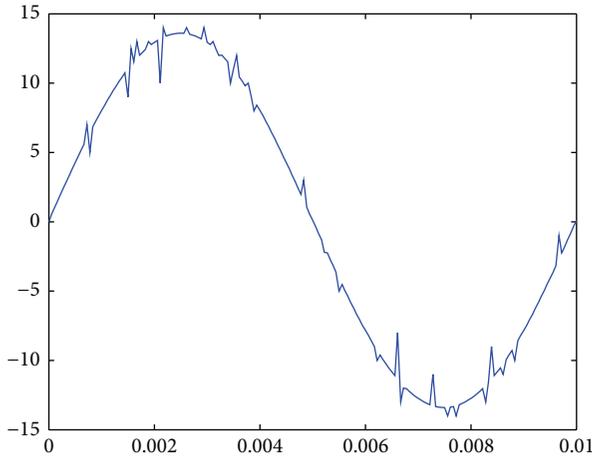


FIGURE 4: Measured current waveform of frequency converter.

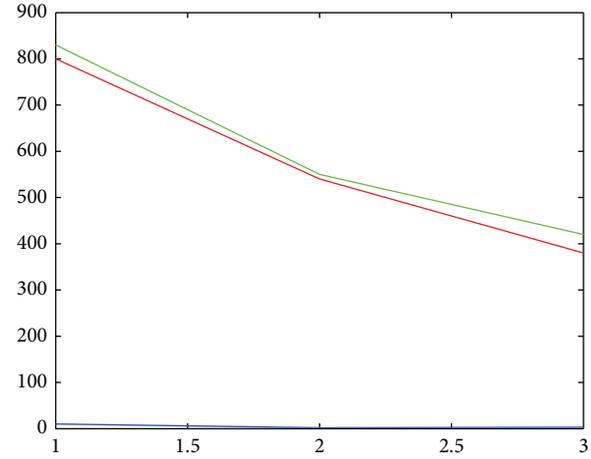


FIGURE 6: Eddy current loss of permanent magnet at different stages.

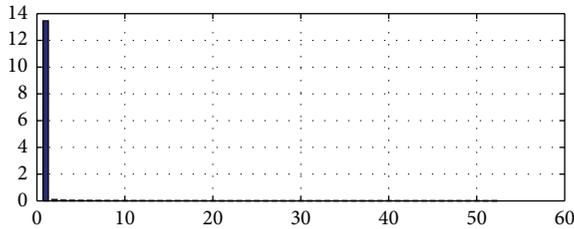


FIGURE 5: Frequency converter output current harmonic analysis.

### 5. Calculation and Result Analysis of the Fault

For the 2.2kW permanent magnet synchronous motor, the permanent magnet eddy current loss caused by the space harmonics increased while the number of segments decreased, and the permanent magnet eddy current loss caused by the time harmonics did not increase with the number of segments decreasing. The reason for why the permanent magnet eddy current loss generated by fundamental and 37th times harmonic increased with the number of segments decreased is that the 37th harmonic current amplitude is smaller and the permanent magnet eddy current loss is mostly generated by fundamental waveform current.

Why did the permanent magnet eddy current loss caused by the space harmonics increase with the number of segments decreasing? The first reason is that the axial split magnetic pole is equivalent to the oblique pole, and the air gap flux density waveform is improved; the other reason is that

the sectional pole blocked the formation of eddy current loop, so the permanent magnetic eddy current loss decreased with the increased of number of segments. The source of excitation provided by the inverter contains a large number of harmonic components and higher harmonic amplitude. The depth of penetration of low order harmonic is larger and more than magnetic direction length, so it is not considered. However, the depth of penetration of high order harmonic is smaller and the skin effect is very strong, so permanent magnet eddy current loss produced by time harmonic will have a maximum value.

### 6. Conclusions

The permanent magnet eddy current loss caused by the space harmonics increased with the number of segments decreasing, and the permanent magnet eddy current loss caused by the time harmonics did not increase with the number of segments decreasing. So, when the motor is designed, especially designing the high speed motor, by using the magnet segmentation to reduce the eddy current loss in the magnets, first of all, consider output current waveform of the inverter and pay attention to the ratio between the pole axial length and the penetration depth of high harmonics waveform of current. Lastly, the compromise between cost of segmental magnetic pole and the magnitude of reducing the eddy current loss also needs to be considered; generally the numbers of segments do not exceed 4.

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] P. B. Reddy, T. M. Jahns, P. J. McCleer, and T. P. Bohn, "Design, analysis and fabrication of a high-performance fractional-slot concentrated winding surface PM machine," in *Proceedings of the 2nd IEEE Energy Conversion Congress and Exposition (ECCE '10)*, pp. 1074–1081, Atlanta, Ga, USA, September 2010.
- [2] Y. Kawase, T. Ota, and H. Fukunaga, "3-D eddy current analysis in permanent magnet of interior permanent magnet motors," *IEEE Transactions on Magnetics*, vol. 36, no. 4, pp. 1863–1866, 2000.
- [3] D. Ishak, Z. Q. Zhu, and D. Howe, "Eddy-current loss in the rotor magnets of permanent-magnet brushless machines having a fractional number of slots per pole," *IEEE Transactions on Magnetics*, vol. 41, no. 9, pp. 2462–2469, 2005.
- [4] W. L. Soong and T. J. E. Miller, "Field-weakening performance of brushless synchronous AC motor drives," *IEE Proceedings: Electric Power Applications*, vol. 141, no. 6, pp. 331–340, 1994.
- [5] M. Nakano, H. Kometani, and M. Kawamura, "A study on Eddy-current losses in rotors of surface permanent-magnet synchronous machines," *IEEE Transactions on Industry Applications*, vol. 42, no. 2, pp. 429–435, 2006.
- [6] W. N. Fu and Z. J. Liu, "Estimation of eddy-current loss in permanent magnets of electric motors using network-field coupled multislice time-stepping finite-element method," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 1225–1228, 2002.
- [7] S. J. Park, H. W. Park, M. H. Lee, and F. Harashima, "A new approach for minimum-torque-ripple maximum-efficiency control of BLDC motor," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 109–114, 2000.
- [8] K. Atallah, J. Wang, and D. Howe, "Torque-ripple minimization in modular permanent-magnet brushless machines," *IEEE Transactions on Industry Applications*, vol. 39, no. 6, pp. 1689–1695, 2003.
- [9] A. G. Jack, B. C. Mecrow, P. G. Dickinson, and D. Stephenson, "Permanent-magnet machines with powdered iron cores and prepressed windings," *IEEE Transactions on Industry Applications*, vol. 36, pp. 1077–1084, 2003.
- [10] B. C. Mecrow, A. G. Jack, J. A. Haylock, and J. Coles, "Fault-tolerant permanent magnet machine drives," *IEE Proceedings—Electric Power Applications*, vol. 143, no. 6, pp. 437–442, 1996.
- [11] Z. Q. Zhu and D. Howe, "Instantaneous magnetic field distribution in brushless permanent magnet dc motors, part III: effect of stator slotting," *IEEE Transactions on Magnetics*, vol. 29, no. 1, pp. 143–151, 1993.
- [12] F. Deng, "Commutation-caused eddy-current losses in permanent-magnet brushless DC motors," *IEEE Transactions on Magnetics*, vol. 33, no. 5, pp. 4310–4318, 1997.
- [13] Z. Q. Zhu, K. Ng, N. Schofield, and D. Howe, "Improved analytical modeling of rotor eddy current loss in brushless machines equipped with surfaced-mounted permanent magnets," *IEE Proceedings—Electric Power Applications*, vol. 151, no. 6, pp. 641–650, 2004.
- [14] V. Hrabovcová, P. Rafajdus, M. Franko, and P. Hudák, *Measurements and Modeling of Electrical Machines*, EDIS, Zilina, Slovakia, 2004 (Slovak).
- [15] A. G. Jack, B. C. Mecrow, and J. A. Haylock, "A comparative study of permanent magnet and switched reluctance motors for high-performance fault-tolerant applications," *IEEE Transactions on Industry Applications*, vol. 32, no. 4, pp. 889–895, 1996.
- [16] D. Ishak, Z. Q. Zhu, and D. Howe, "Permanent-magnet brushless machines with unequal tooth widths and similar slot and pole numbers," *IEEE Transactions on Industry Applications*, vol. 41, no. 2, pp. 584–590, 2005.
- [17] G. S. Liew, N. Ertugrul, W. L. Soong, and J. Gayler, "Investigation of axial field permanent magnet motor utilising amorphous magnetic material," *Australian Journal of Electrical and Electronics Engineering*, vol. 3, no. 2, pp. 111–120, 2007.
- [18] A. Vagati, P. Guglielmi, M. Pastorelli, and G. Pellegrino, "Position-sensorless control of permanent-magnet-assisted synchronous reluctance motor," *IEEE Transactions on Industry Applications*, vol. 40, no. 2, pp. 615–622, 2004.
- [19] B. J. Chalmers, L. Musaba, and D. F. Gosden, "Variable-frequency synchronous motor drives for electric vehicles," *IEEE Transactions on Industry Applications*, vol. 32, no. 4, pp. 896–903, 1996.
- [20] T. Kojima, T. Tanno, K. Maeyama, and H. Okutsu, "Method for manufacturing an axial air-gap electronic motor," U.S. Patent 7 120 986, Oct. 17, 2006.
- [21] K. Yamazaki and A. Abe, "Loss investigation of interior permanent-magnet motors considering carrier harmonics and magnet eddy currents," *IEEE Transactions on Industry Applications*, vol. 45, no. 2, pp. 659–665, 2009.

## Research Article

# Minimal-Order Functional Observer-Based Residual Generators for Fault Detection and Isolation of Dynamical Systems

**H. M. Tran and H. Trinh**

*School of Engineering, Deakin University, Geelong, VIC 3217, Australia*

Correspondence should be addressed to H. Trinh; [hieu.trinh@deakin.edu.au](mailto:hieu.trinh@deakin.edu.au)

Received 10 January 2016; Revised 22 March 2016; Accepted 5 April 2016

Academic Editor: Zehui Mao

Copyright © 2016 H. M. Tran and H. Trinh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper examines the design of minimal-order residual generators for the purpose of detecting and isolating actuator and/or component faults in dynamical systems. We first derive existence conditions and design residual generators using only first-order observers to detect and identify the faults. When the first-order functional observers do not exist, then based on a parametric approach to the solution of a generalized Sylvester matrix equation, we develop systematic procedures for designing residual generators utilizing minimal-order functional observers. Our design approach gives lower-order residual generators than existing results in the literature. The advantages for having such lower-order residual generators are obvious from the economical and practical points of view as cost saving and simplicity in implementation can be achieved, particularly when dealing with high-order complex systems. Numerical examples are given to illustrate the proposed fault detection and isolation schemes. In all of the numerical examples, we design minimum-order residual generators to effectively detect and isolate actuator and/or component faults in the system.

## 1. Introduction

Industrial systems have become more and more complicated and expensive, and thus the requirement of safety and reliability in system operations is paramount [1]. In fact, high complex modern engineering systems are vulnerable to unavoidable faults. Faults can happen due to an internal event, change in environmental conditions, error in the design of the system, sensor failure, actuator malfunctioning, or even human mistakes during operation. The unexpected faults disrupt the system operation, break down part of or the whole system, and can lead to fatal consequences. Fault detection and isolation (FDI) providing failure signals, therefore, have been considered as a vital aspect of system control research. This research issue has attracted extensive studies to safeguard equipment as well as amend the safety and reliability of modern system performance. Since the early 1970s, there have been a large number of fruitful results in FDI which can be found in, for example, [1–30] and the references therein.

In fault diagnosis area, there are two well-known model based fault diagnosis approaches. The first is the fault estimation based approach (see, e.g., [3–8]). By this approach, the systems are decoupled into fault-free and fault dependent parts. The faults can then be estimated via the observers of the fault-free part. The other is the residual-based approach which makes use of state observers to generate diagnostic signals or, in other words, residual generators. In [15], the authors provided a useful comparison between the two approaches. This study is interesting as it provides insightful knowledge and some shortcomings of each approach. The philosophy behind the residual-based approach is to estimate the system state vector based on the control inputs and the measured outputs. The residual generators are then constructed by a properly weighted output estimation error. The residuals are able to detect and identify the fault happening in the systems [9–14]. Obviously, the residual signals are expected to be close to zero in a fault-free condition and deviate from zero in the presence of a fault. For FDI purpose, a decision rule which is normally based on a threshold set is

engaged to test for the likelihood of faults occurring in the systems.

Regarding the residual-based approach, the residual signals in most studies are generated via primarily employing full-order state observers or filter schemes which can be found in, for instance, [16–25]. However, the design and implementation of the residual generators using full-order state observers normally require high degrees of complexity and computational work, especially when dealing with high-order complex systems. As stated in [30], in certain applications such as fault diagnostic or control system design, an estimation of the entire state vector is not necessary. Meanwhile, there have been some FDI schemes based on reduced-order state observers (see, e.g., [26–29]). In these schemes, the residual generators are constructed based on the observers of the partial state vector. By that a significantly lower order of the residual generators, in comparison with the ones employing full-order state observers, can be achieved. With an attempt to simplify the design of the residual generators, this trend is identified as very important from the practical point of view (see [31–35]).

In this paper, by using minimal-order functional observers, we present a new FDI scheme to detect and identify  $l$  unpredictable actuator and/or component fault signals entering from the inputs of dynamical systems. The systems considered in this paper have  $n$  state variables,  $p$  measured outputs, and  $m$  control inputs. Our approach to FDI utilizes the advantages of functional observer schemes which can be found in [36–41]. Due to the fact that functional observers are employed instead of full-order state observers (or, for that matter, reduced-order Luenberger state observers) to generate the residual signals, substantial reduction in the complexity of the overall designed FDI schemes can now be consequently achieved. Particularly, we first present a simple solution approach where residual generators are designed using only first-order functional observers to detect and isolate faults in the systems. When first-order functional observers are not possible to design, then the residual generators are designed based on minimal-order functional observers. To minimize the observer order, we employ a parametric technique to the solutions of a generalized Sylvester matrix equation appearing in the existence conditions. Regarding the timely fault detection (FD), designing the residuals only involves the functional observers with the order as low as  $[(n - p + 1)/p]$ , where  $[x]$  denotes the smallest integer larger than  $x$ . This is a new finding and has not yet been reported in the literature. The result is significant since it is clear that, for  $p > 1$ , the order of the residual generator is much lower than any of the existing results in the literature. Furthermore, for timely isolation of  $l$  independent actuator and/or component faults in the system, we propose to construct a bank of  $l$  residual generators, each with the order as low as  $[(n - p + 1)/(p - 1)]$ . Thus, it is clear that our proposed FD and FI schemes, taking advantage of functional observers, are identified as most beneficial for complex large-scale systems where, by default,  $n$  and  $p$  are large values.

The organization of this paper is as follows. In the next section (Section 2), we present the system description and

preliminaries where we consider linear dynamical systems with  $l$  unpredictable actuator and/or component fault signals entering from the system inputs. This section also introduces a novel residual generator that is constructed from reduced-order functional observers. Section 3 presents a detailed analysis on the design of minimal-order residual generators to trigger, in a timely manner, any fault that enters into the system. This follows by Section 4 where a bank of  $l$  residual generators is designed to isolate  $l$  likely faults. To achieve this, each residual generator is designed, using first-order or minimal-order functional observers, to be insensitive to one specific fault but sensitive to the remaining  $(l - 1)$  faults. Subsequently, a logic table can be drawn up from the  $l$  residual generator outputs to detect and isolate the faults. Section 5 presents extensive numerical examples to highlight the attractive features of our proposed FDI schemes. Finally, Section 6 concludes the paper.

## 2. System Description and Preliminaries

In this paper, we consider dynamical systems with  $p$  outputs and  $m$  inputs and with unpredictable actuator fault signals  $f(t) \in \mathbb{R}^l$  entering from the system inputs. In state space models, the systems are governed by the following equations:

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

where  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$ , and  $y(t) \in \mathbb{R}^p$  are the state, input, and output vectors, respectively.  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ , and  $C \in \mathbb{R}^{p \times n}$  are known constant system matrices.  $D \in \mathbb{R}^{n \times l}$  is the fault identity matrix. We assume that  $\text{rank}(C) = p$ , without loss of generality, the fault identity matrix  $D$  is also assumed to be a known full-column rank matrix, and the faults  $f_i(t)$ ,  $i = 1, 2, \dots, l$ , are linearly independent. This assumption avoids vagueness which may appear when some faults occur simultaneously; as a result, the residual generators may not detect these faults due to the zero overall effect of these faults [26].

Let us consider the following reduced-order functional observer:

$$\dot{\omega}(t) = N\omega(t) + Gy(t) + Hu(t),\tag{2}$$

where  $\omega(t) \in \mathbb{R}^q$ ,  $1 \leq q < n$ ,  $N \in \mathbb{R}^{q \times q}$ ,  $G \in \mathbb{R}^{q \times p}$ , and  $H \in \mathbb{R}^{q \times m}$  are observer parameters to be determined such that  $\omega(t)$  is an asymptotic estimate of a linear function  $Lx(t)$  when no fault appears in the system, that is,  $f(t) = 0$ , and  $L \in \mathbb{R}^{q \times n}$  is a matrix to be determined for the purpose of FDI.

We now define a residual generator,  $r(t)$ , which is used to trigger the faults in the system:

$$r(t) = T\omega(t) + Fy(t),\tag{3}$$

where  $T \in \mathbb{R}^{1 \times q}$  and  $F \in \mathbb{R}^{1 \times p}$  are residual parameters.

Figure 1 shows block-diagram implementation of the residual generator as defined in (2)-(3). By that, only known information of the inputs,  $u(t)$ , and the outputs,  $y(t)$ , of the system is utilized to generate the residual generator  $r(t)$ .

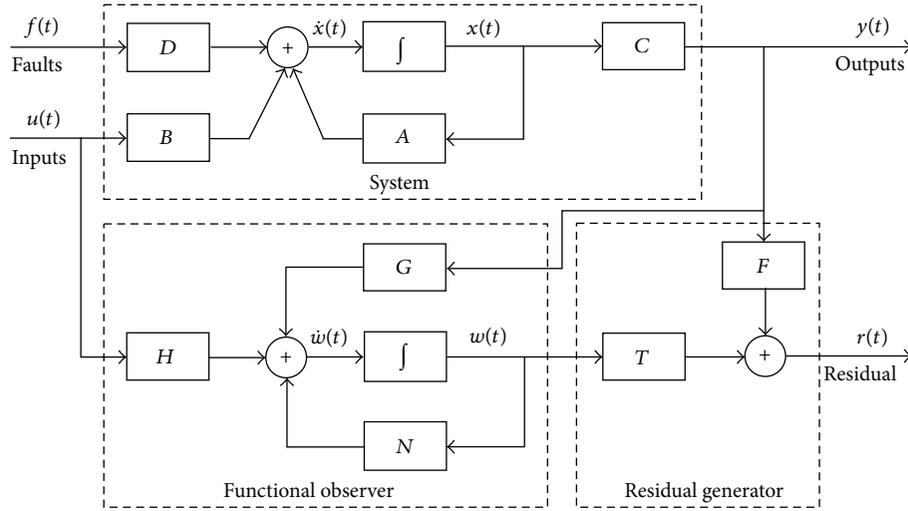


FIGURE 1: Block-diagram implementation of a residual generator.

Obviously, regarding the FD purpose, parameters  $T$ ,  $F$ ,  $L$ ,  $N$ ,  $G$ , and  $H$  should be determined to meet the required functions of the residual generator such that

$$\lim_{t \rightarrow \infty} r(t) = \begin{cases} 0 & \text{if } f(t) = 0 \\ c \text{ or undefined} & \text{if } f(t) \neq 0, \end{cases} \quad (4)$$

where  $c \neq 0$ ,  $f(t) = 0$  implies a faultless condition, and  $f(t) \neq 0$  implies a faulty condition.

Error dynamics  $e(t)$ , which is defined to be the difference between the estimation  $\omega(t)$  and the linear function  $Lx(t)$ , is expressed as follows:

$$e(t) = \omega(t) - Lx(t). \quad (5)$$

By taking the derivative of the error vector (5) and then substituting  $\dot{\omega}(t)$  from (2) and  $\dot{x}(t)$  from (1), the error dynamics can be written as follows:

$$\begin{aligned} \dot{e}(t) = & Ne(t) + (NL + GC - LA)x(t) \\ & + (H - LB)u(t) - LDf(t). \end{aligned} \quad (6)$$

It is clear from (6) that, under the faultless condition, the error asymptotically converges to zero if the following conditions hold.

**Proposition 1.** Under faultless conditions, that is,  $f(t) = 0$ ,  $\omega(t)$  is an asymptotic estimate of  $Lx(t)$  (i.e.,  $e(t) \rightarrow 0$  asymptotically) for any initial conditions  $x(0)$  and  $\omega(0)$  if

$$N \text{ is Hurwitz}, \quad (7)$$

$$NL + GC - LA = 0, \quad (8)$$

$$H - LB = 0. \quad (9)$$

*Proof.* When no fault occurs in the system, if conditions (8) and (9) are satisfied then (6) is reduced to  $\dot{e}(t) = Ne(t)$ . Thus,

if  $N$  is Hurwitz then  $e(t) \rightarrow 0$  asymptotically. Note also that if all the eigenvalues of  $N$  can be arbitrarily assigned, then  $e(t) \rightarrow 0$  with any prescribed rate. This completes the proof of Proposition 1.  $\square$

**Definition 2.** The functional observer (2) is an Asymptotic Estimator if  $\omega(t)$  is an asymptotic estimate of  $Lx(t)$  when  $f(t) = 0$ .

Since Proposition 1 is satisfied and based on Definition 2, the error dynamics of an asymptotic observer takes the following form:

$$\dot{e}(t) = Ne(t) - LDf(t). \quad (10)$$

Substituting (5) into (3), the residual generator  $r(t)$  can be obtained as follows:

$$r(t) = Te(t) + (TL + FC)x(t). \quad (11)$$

According to (4) and (11), the residual generator can satisfy its proposed functions if the following conditions hold.

**Proposition 3.** An Asymptotic Estimator of form (2) can satisfy  $r(t)$  according to (4) for any  $x(0)$  and  $\omega(0)$  if

$$TL + FC = 0, \quad (12)$$

$$LD \neq 0. \quad (13)$$

*Proof.* If  $f(t) = 0$  and also (12) and (13) are satisfied then according to (10) we have  $e(t) \rightarrow 0$  asymptotically and according to (11) we have  $r(t) \rightarrow 0$  asymptotically. When a fault appeared in the system, that is,  $f(t) \neq 0$ , and condition (13) holds, based on (10) we have  $e(t) \neq 0$ . Accordingly, the residual signals (11) deviate from zero, which indicates the fault happened in the system. Thus, the required functions of the residual generator as stated in (4) are satisfied. This completes the proof of Proposition 3.  $\square$

Now, the design of the functional observer (2) and the residual generator (3) reduces to determining the unknown matrices  $L$ ,  $N$ ,  $G$ ,  $H$ ,  $T$ , and  $F$  such that conditions (7)–(9), (12), and (13) are satisfied.

### 3. Fault Detection Scheme

In Section 2, we showed that the functional observer (2) and the residual generator (3) can detect faults in the system if all the unknown parameters satisfying the conditions stated in Propositions 1 and 3 are found. Observe that if  $L$  is known, then from (9),  $H = LB$ , and condition (13), that is,  $LD \neq 0$ , can be easily verified. In this section, we propose a systematic procedure for solving coupled matrix equations (8) and (12) with the requirement that matrix  $N$  is Hurwitz. Note that, here, not only do we require that matrix  $N$  be Hurwitz, but also we desire its eigenvalues to be placed at some prespecified locations in the  $s$ -plane in order to achieve timely detection of faults.

Let us first simplify matrix equations (8) and (12) by a partition technique defined in [38]. Regarding this, an invertible matrix  $P \in \mathbb{R}^{n \times n}$  is introduced:

$$P = [C^+ \ C^\perp], \quad (14)$$

where  $C^+ \in \mathbb{R}^{n \times p}$  denotes the Moore-Penrose inverse of  $C$ , that is,  $CC^+ = I_p$ , and  $C^\perp \in \mathbb{R}^{n \times (n-p)}$  denotes an orthogonal basis for the null-space of  $C$ , that is,  $CC^\perp = 0$ . Now the following partitions are defined:

$$CP = [I_p \ 0], \quad (15)$$

$$LP = [L_1 \ L_2], \quad (16)$$

$$P^{-1}AP = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad (17)$$

where submatrices  $L_1 \in \mathbb{R}^{q \times p}$ ,  $L_2 \in \mathbb{R}^{q \times (n-p)}$ ,  $A_{11} \in \mathbb{R}^{p \times p}$ ,  $A_{12} \in \mathbb{R}^{p \times (n-p)}$ ,  $A_{21} \in \mathbb{R}^{(n-p) \times p}$ , and  $A_{22} \in \mathbb{R}^{(n-p) \times (n-p)}$ .

Postmultiplying both sides of (8) and (12) by the matrix  $P$ , we obtain

$$NLP + GCP - LPP^{-1}AP = 0, \quad (18)$$

$$TLP + FCP = 0.$$

By substituting (15)–(17) into (18), we achieve the following equations:

$$G = L_1A_{11} + L_2A_{21} - NL_1, \quad (19)$$

$$F = -TL_1, \quad (20)$$

$$NL_2 - L_2A_{22} - L_1A_{12} = 0, \quad (21)$$

$$TL_2 = 0. \quad (22)$$

It is clear from (19) and (20) that matrices  $G$  and  $F$  can be computed once  $N$ ,  $L_1$ ,  $L_2$ , and  $T$  are found to satisfy (21) and (22).

In the following, we discuss the solutions to coupled matrix equations (21) and (22). Observe that the solution to (21) depends on the characteristic of matrix  $A_{12} \in \mathbb{R}^{p \times (n-p)}$ . Hence, we classify  $A_{12}$  into two cases, namely, Case 1 where  $A_{12}$  is a column matrix and Case 2 where  $A_{12}$  is a row matrix.

*Case 1* (first-order residual generators). In this case, we consider that  $p > n/2$ ; accordingly  $A_{12}$  is a column matrix. For this, we only need to design a first-order residual generator to detect faults in the system. Regarding this, the design procedure is detailed as follows.

**Theorem 4.** *A first-order functional observer (2) always exists for system (1) when  $A_{12}$  is a column matrix; that is,  $p > n/2$ . Furthermore, a residual generator (3) can be constructed to detect faults in the system if condition (13) is satisfied.*

*Proof.* To design a first-order functional observer, that is,  $q = 1$ ,  $N \in \mathbb{R}^{1 \times 1}$  can be chosen to be any negative scalar. Thus, let

$$N = s, \quad s < 0. \quad (23)$$

By letting  $L_2 = 0$ , (21) is reduced to the following:

$$L_1A_{12} = 0. \quad (24)$$

Since  $A_{12}$  is a column matrix, a solution to (24) where  $L_1 \neq 0$  always exists. Let  $\mathcal{N}(X)$  be a matrix of row basis vectors for the row-null-space of  $X$ ; that is,  $\mathcal{N}(X)X = 0$ . Therefore, the solution for  $L_1$  in (24) can be achieved by first computing  $\widehat{L}_1$  according to

$$\widehat{L}_1 = \mathcal{N}(A_{12}) \quad (25)$$

and then  $L_1$  can be selected as any row of  $\widehat{L}_1$ .

Matrix  $T \neq 0$  in (22) can be arbitrarily chosen to be any nonzero scalar, say,  $\alpha$ , since  $L_2 = 0$ . Finally, if condition (13) is satisfied, a residual generator using a first-order functional observer exists to detect faults in the system. Matrices  $H$ ,  $G$ , and  $F$  can then be easily obtained from (9), (19), and (20), respectively, to complete the design of the residual generator. This completes the proof of Theorem 4.  $\square$

It is worth pointing out that, in the derivation of a first-order residual generator, we do not impose the requirement that the matrix pair  $(A, C)$  is observable.

*Case 2* (minimal-order residual generators). In this case, we consider that  $1 < p \leq n/2$ ; accordingly  $A_{12}$  is a row matrix. Matrix  $A_{12}$  is considered to be full rank; that is,  $\text{rank}(A_{12}) = p$ . For this, we design a minimal-order residual generator for FD by presenting a solution to coupled matrix equations (21) and (22) via a parametric technique. Here,  $N$  has preassigned distinct eigenvalues satisfying condition (7). For completeness, let us first present a parametric solution [42] to generalized Sylvester matrix equation (21). For the solution of (21), we require that the pair  $(A_{12}, A_{22})$  be observable. This implies that the pair  $(C, A)$  is observable [38].

Let  $N \in \mathbb{R}^{q \times q}$  with  $q$  distinct eigenvalues be defined as follows:

$$N = Q^{-1} \Lambda Q, \quad (26)$$

where  $Q \in \mathbb{R}^{q \times q}$  is any freely chosen invertible matrix and  $\Lambda = \text{diag}(s_1, s_2, \dots, s_q)$ ,  $s_i \neq s_j$  for  $i \neq j$  and  $\text{Re}\{s_i\} < 0$  for all  $i = 1, 2, \dots, q$ . With  $N$  as defined in (26),  $L_1$  and  $L_2$  satisfying (21) are given in the following parametric forms [42]:

$$L_1 = Q \left[ U(s_1) b_1 \quad U(s_2) b_2 \quad \cdots \quad U(s_q) b_q \right]^\top, \quad (27)$$

$$L_2 = Q \left[ Z(s_1) b_1 \quad Z(s_2) b_2 \quad \cdots \quad Z(s_q) b_q \right]^\top, \quad (28)$$

where  $b_i \in \mathbb{C}^p$  ( $i = 1, 2, \dots, q$ ) are free vectors satisfying  $b_i = \bar{b}_j$  if  $s_i = \bar{s}_j$ .  $\bar{s}_j$  denotes the complex conjugate of  $s_i$ .  $U(s) \in \mathbb{R}^{p \times p}$  and  $Z(s) \in \mathbb{R}^{(n-p) \times p}$  are coprime polynomial matrices satisfying the following coprime factorization:

$$(sI_{n-p} - A_{22}^\top)^{-1} A_{12}^\top = Z(s) U^{-1}(s). \quad (29)$$

The reader can refer to [42] for a numerically reliable algorithm to compute  $Z(s)$  and  $U(s)$ . Also, as suggested in [38],  $U(s)$  and  $Z(s)$  can be conveniently computed according to the following equations:

$$U(s) = \det(sI_{n-p} - A_{22}) I_p, \quad (30)$$

$$Z(s) = \text{adj}(sI_{n-p} - A_{22}^\top) A_{12}^\top, \quad (31)$$

where  $\det(\cdot)$  and  $\text{adj}(\cdot)$  denote the determinant and the adjugate matrix of matrix  $(\cdot)$ , respectively. For any given  $A_{22}$ , the characteristic polynomial can be obtained:

$$\begin{aligned} a(s) &\triangleq \det(sI_{n-p} - A_{22}) \\ &= s^{n-p} + a_1 s^{n-p-1} + a_2 s^{n-p-2} + \cdots + a_{n-p}, \end{aligned} \quad (32)$$

where the coefficients  $a_i$ ,  $i = 1, 2, \dots, (n-p)$ , are real constants. The adjugate matrix  $\text{adj}(\cdot)$  is then obtained as follows:

$$\begin{aligned} \text{adj}(sI_{n-p} - A_{22}^\top) &= Y_1 s^{n-p-1} + Y_2 s^{n-p-2} + Y_3 s^{n-p-3} \\ &\quad + \cdots + Y_{n-p}, \end{aligned} \quad (33)$$

where  $Y_i$ ,  $i = 1, 2, \dots, n-p$ , are computed by using the coefficients of  $a(s)$  and matrix  $A_{22}$ , where

$$\begin{aligned} Y_1 &= I_{n-p}, \\ Y_2 &= Y_1 A_{22}^\top + a_1 I_{n-p}, \\ Y_3 &= Y_2 A_{22}^\top + a_2 I_{n-p}, \\ &\vdots \\ Y_{n-p} &= Y_{n-p-1} A_{22}^\top + a_{n-p-1} I_{n-p}. \end{aligned} \quad (34)$$

Note that if the preassigned eigenvalues of  $N$  are complex then the resulting matrices  $\Lambda$ ,  $L_1$ , and  $L_2$  are also complex. As reported in [14], we can obtain real matrices  $\Lambda$ ,  $L_1$ , and  $L_2$  by applying the following simple computation. Here, without loss of generality, we assume that  $s_1$  and  $s_2, s_2 = \bar{s}_1 = \alpha + \beta j$ , are a pair of the eigenvalues containing complex values, and all other eigenvalues of  $N$  are real and distinct. The real matrices  $\Lambda$ ,  $L_1$ , and  $L_2$  can be obtained as follows:

$$\Lambda = \text{block-diag}(\Lambda_1, s_3, \dots, s_q),$$

$$\Lambda_1 = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix},$$

$$L_1 \quad (35)$$

$$= Q \left[ \text{Re}\{U(s_1) b_1\} \quad \text{Im}\{U(s_1) b_1\} \quad \cdots \quad U(s_q) b_q \right]^\top,$$

$$L_2$$

$$= Q \left[ \text{Re}\{Z(s_1) b_1\} \quad \text{Im}\{Z(s_1) b_1\} \quad \cdots \quad Z(s_q) b_q \right]^\top,$$

where  $\text{Re}\{\cdot\}$  and  $\text{Im}\{\cdot\}$  denote the real and the imaginary parts of  $\{\cdot\}$ , respectively.

Now, by incorporating  $L_2$  as defined in (28) into (22), we can solve for  $T$ . Let us now present the result for Case 2 by the following theorem.

**Theorem 5.** *With  $1 < p \leq n/2$  and  $\text{rank}(A_{12}) = p$ , a residual generator (3) with an order as low as  $q = [(n-p+1)/p]$  can be constructed to detect faults in the system if condition (13) is satisfied.*

*Proof.* First, let us express  $TQ$  as follows:

$$TQ = [t_1 \quad t_2 \quad \cdots \quad t_q], \quad (36)$$

where  $t_i \neq 0$ ,  $i = 1, 2, \dots, q$ , are arbitrarily real numbers.

Once  $t_i$  are scalars, by substituting (28) and (36) into the transpose of (22) and after some rearrangement the following equation is obtained:

$$\begin{bmatrix} Z(s_1) t_1 & Z(s_2) t_2 & \cdots & Z(s_q) t_q \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix} = 0. \quad (37)$$

Now, (37) can be expressed as follows:

$$\mathcal{Z} b = 0, \quad (38)$$

where  $\mathcal{Z} \in \mathbb{R}^{(n-p) \times qp}$  and  $b \in \mathbb{R}^{qp}$  and

$$\begin{aligned} \mathcal{Z} &= [Z(s_1) t_1 \quad Z(s_2) t_2 \quad \cdots \quad Z(s_q) t_q], \\ b &= [b_1^\top \quad b_2^\top \quad \cdots \quad b_q^\top]^\top. \end{aligned} \quad (39)$$

Similar to the solution to (24), let  $\mathcal{N}(\mathcal{Z})$  be defined such that  $\mathcal{Z}\mathcal{N}(\mathcal{Z}) = 0$ ; then a solution to (38) always exists for  $b \neq 0$  if  $\mathcal{N}(\mathcal{Z}) \neq 0$ . Accordingly, if  $\mathcal{Z}$  is a row matrix, that is, the number of columns is greater than the number of rows, the solution to  $b \neq 0$  always exists. That implies the following result:

$$q \geq \left\lceil \frac{n-p+1}{p} \right\rceil, \quad (40)$$

and then  $b \neq 0$  exists and can be taken to be any column of  $\mathcal{N}(\mathcal{Z})$ .

As a result, with  $q$  prescribed eigenvalues for  $N$  (i.e.,  $s_i, i = 1, 2, \dots, q$ , are given) and for any arbitrary invertible matrix  $Q$ , matrices  $N, L_1$ , and  $L_2$  are computed based on (26), (27), and (28), respectively. Matrix  $L$  is then calculated by (16) and condition (13) can be verified. If (13) is satisfied, the residual generator  $r(t)$  can detect faults in the system. Finally,  $H, G, F$ , and  $T$  are obtained from (9), (19), (20), and (36), respectively, to complete the design of the observer  $\omega(t)$  and the residual  $r(t)$ . This completes the proof of Theorem 5.  $\square$

*Remark 6.* The observer order  $q$  can be assigned to satisfy condition (40), where a solution to (38) always exists for  $b \neq 0$ . Thus, for the purpose of designing minimal-order observers, we only need to choose  $q = \lceil (n-p+1)/p \rceil$ .

*Remark 7.* External disturbances or the uncertainties are unavoidable during the operation of the systems. If we take disturbances in the system where the disturbance matrix is defined as  $E \in \mathbb{R}^{n \times r}$  and the disturbance vector is  $d(t) \in \mathbb{R}^r$ , system (1) now can be governed by the following equations:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Df(t) + Ed(t), \\ y(t) &= Cx(t). \end{aligned} \quad (41)$$

Based on the disturbance decoupling approach [7], the influence of the unknown disturbance can be eliminated. Using this approach, the conditions in Proposition 1 of this paper are as follows:

$$\begin{aligned} N &\text{ is Hurwitz,} \\ NL + GC - LA &= 0, \\ H - LB &= 0, \\ LE &= 0. \end{aligned} \quad (42)$$

Since the condition  $LE = 0$  is added, the solutions to the observer parameters will be more strict. Nevertheless, based on the solution approach in this paper, it is possible to design minimal-order observer-based residual generators to detect faults in the systems with unknown disturbances. In this case, the residual functions are also defined as (4) which means the threshold can be set as zero. In addition, it is also possible to extend the result to include nonlinearities in the systems. This topic deserves further research and will be a subject for future research. Another issue that deserves further research is related to the robust threshold selection. How to solve the robust threshold selection problem under uncertainties is an

important research question and the reader is referred to the most recent and interesting research [43].

*Remark 8.* We conclude this section by considering the case where  $1 < p \leq n/2$  and  $\text{rank}(A_{12}) < p$ . Note that, for this case, a first-order residual generator can be designed since a solution to (24) where  $L_1 \neq 0$  exists due to the fact that  $\text{rank}(A_{12}) < p$ . Indeed, there exists  $\mathcal{N}(A_{12}) \neq 0$  such that  $\mathcal{N}(A_{12})A_{12} = 0$ . Accordingly,  $L_1 \neq 0$  is obtained and the rest of the observer and residual generator parameters can be derived in the same way as reported in Case 1. A numerical example can be easily constructed to demonstrate this case. However, due to space limiting reason, a numerical example is not given in this paper.

#### 4. Fault Isolation Scheme

In Section 3, we discussed the design of minimal-order residual generators for FD in the systems. However, identifying the faults is also considered as an important task in system control since it provides the fault positions [15]. The information about the fault positions is then sent to the decision-maker for further actions dealing with the faults. In this regard, we present a systematic procedure to construct a bank of minimal-order residual generators which is used to isolate the faults. Hence, our proposed FI scheme significantly reduces the complexity in the implementation. The scheme is thus more suited to deal with complex high-order systems. Before showing the procedure, let us mention a logic for the fault isolation from the following [44].

We consider, in this paper, that there may be  $l$  likely faults occurring in the system. Accordingly, for the purpose of FI, we propose to design a bank of  $l$  residual generators, in which each residual generator is expected to be insensitive to one specific fault but sensitive to the rest of the remaining faults. Thus, a  $k$ th residual generator,  $r_k(t)$ , is designed to be insensitive to the  $k$ th fault, that is,  $f_k(t)$ , and sensitive to the rest of the remaining faults. By that, when there is no fault in the system, the outputs of the residual generators remain as zero. However, when a fault, say,  $f_k(t)$ , has occurred, the output of the  $k$ th residual generator remains as zero; meanwhile, the outputs of the other  $(l-1)$  residual generators are not zero but turn to  $c_k \neq 0$ , where  $c_k$  is as defined in (52). Consequently, by measuring the output values of the  $l$  residual generators or using a set of residual thresholds, when a fault appears, we can identify the position of the fault.

For simpler explaining of the logic, let us take an example where there are four possible faults, that is,  $f_k(t), k = 1, 2, 3, 4$ , in the system. Accordingly, a bank of four minimal-order residual generators,  $r_k(t), k = 1, 2, 3, 4$ , is designed to isolate the faults. Let us use a logic “1” to indicate that the output of a residual generator is bigger than a threshold value, that is,  $c_k \neq 0$ , and a logic “0” to indicate that the output of a residual generator remains as zero (or less than a threshold value). Table 1 shows the logic for the fault isolation. In the table, for instance, when a fault  $f_1(t)$  has occurred, residual  $r_1(t)$  is insensitive to  $f_1(t)$ ; hence it appears as “0.” Meanwhile, other residuals can trigger  $f_1(t)$  so they are expressed as “1.” Therefore, we can identify that fault  $f_1(t)$  has occurred.

TABLE 1: Fault isolation logic table.

	$r_1(t)$	$r_2(t)$	$r_3(t)$	$r_4(t)$
No fault	0	0	0	0
$f_1(t)$	0	1	1	1
$f_2(t)$	1	0	1	1
$f_3(t)$	1	1	0	1
$f_4(t)$	1	1	1	0

However, it is noted that this logic is only correct when the faults happen independently and also not simultaneously. In Section 2, we have made an assumption that avoids such a situation and hence the logic expressed in Table 1 can thus be used to isolate the faults in the system.

Now, by utilizing the above FI logic and for the rest of this section, we present systematic procedures to design each residual generator in the residual bank,  $r_k(t)$ , with minimal order. For this, let us rearrange and partition the fault vector,  $f(t)$ , and the fault identity matrix,  $D$ , as follows:

$$f(t) = \begin{bmatrix} f_k(t) \\ f_R(t) \end{bmatrix}, \quad (43)$$

$$D = [D_k \ D_R],$$

where  $f_k(t)$  is the  $k$ th fault,  $f_R(t)$  contains the remaining  $(l-1)$  faults,  $D_k \in \mathbb{R}^{n \times 1}$ , and  $D_R \in \mathbb{R}^{n \times (l-1)}$ .

Thus, system (1) can now be rewritten as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + D_k f_k(t) + D_R f_R(t), \\ y(t) &= Cx(t). \end{aligned} \quad (44)$$

Similar to Section 2, let us now consider a reduced-order functional observer for the purpose of fault isolation:

$$\dot{\omega}_k(t) = N_k \omega_k(t) + G_k y(t) + H_k u(t), \quad (45)$$

where  $\omega_k(t) \in \mathbb{R}^q$ ,  $1 \leq q < n$ ,  $N_k \in \mathbb{R}^{q \times q}$ ,  $G_k \in \mathbb{R}^{q \times p}$ , and  $H_k \in \mathbb{R}^{q \times m}$  are observer parameters to be determined such that  $\omega_k(t)$  is an asymptotic estimate of the linear function  $L_k x(t)$  when  $f(t) = 0$  and  $L_k \in \mathbb{R}^{q \times n}$  is a matrix to be determined for the purpose of fault isolation.

Let an error vector  $e_k(t)$  be defined as follows:

$$e_k(t) = \omega_k(t) - L_k x(t). \quad (46)$$

By taking the derivative of (46) and using (44) and (45), the following error dynamics is obtained:

$$\begin{aligned} \dot{e}_k(t) &= N_k e_k(t) + (N_k L_k + G_k C - L_k A) x(t) \\ &\quad + (H_k - L_k B) u(t) - L_k D_k f_k(t) \\ &\quad - L_k D_R f_R(t). \end{aligned} \quad (47)$$

**Proposition 9.** Under no-fault conditions,  $\omega_k(t)$  is an asymptotic estimate of the function  $L_k x(t)$  (i.e.,  $e_k(t) \rightarrow 0$  asymptotically) for any  $x(0)$  and  $\omega_k(0)$  if

$$N_k \text{ is Hurwitz}, \quad (48)$$

$$N_k L_k + G_k C - L_k A = 0, \quad (49)$$

$$H_k - L_k B = 0. \quad (50)$$

*Proof.* The proof of Proposition 9 is the same as the proof of Proposition 1; thus it is omitted here.  $\square$

By using the estimate  $\omega_k(t)$  and the system outputs,  $y(t)$ , the residual generator  $r_k(t)$ , which is insensitive to fault  $f_k(t)$  and sensitive to the other faults,  $f_R(t)$ , is now proposed:

$$r_k(t) = T_k \omega_k(t) + F_k y(t), \quad (51)$$

where  $r_k(t) \in \mathbb{R}$ ,  $T_k \in \mathbb{R}^{1 \times q}$ , and  $F_k \in \mathbb{R}^{1 \times p}$ .

Since faults  $f(t)$  occur independently, matrices  $T_k$  and  $F_k$  need to be found to satisfy the following functions of  $r_k(t)$ :

$$\lim_{t \rightarrow \infty} r_k(t) = \begin{cases} 0 & \text{if } f_R(t) = 0 \\ c_k \text{ or undefined} & \text{if } f_R(t) \neq 0, \end{cases} \quad (52)$$

where  $c_k \neq 0$ ,  $f_R(t) = 0$  means that, except fault  $f_k(t)$  which may or may not happen, other faults do not happen, and  $f_R(t) \neq 0$  means that any of the  $(l-1)$  faults can happen in the system.

If all the conditions in Proposition 9 are satisfied, from (45), (46), and (51), the residual  $r_k(t)$  can be governed in the following equations:

$$\begin{aligned} \dot{e}_k(t) &= N_k e_k(t) - L_k D_k f_k(t) - L_k D_R f_R(t), \\ r_k(t) &= T_k e_k(t) + (T_k L_k + F_k C) x(t). \end{aligned} \quad (53)$$

**Proposition 10.** The residual generator  $r_k(t)$  is insensitive to fault  $f_k(t)$  and sensitive to faults  $f_R(t)$  for any  $x(0)$  and  $\omega_k(0)$  if

$$T_k L_k + F_k C = 0, \quad (54)$$

$$L_k D_k = 0, \quad (55)$$

$$L_k D_R \neq 0. \quad (56)$$

*Proof.* The proof can be constructed by following similar lines as in the proof of Proposition 3 and thus it is omitted here.  $\square$

From Propositions 9 and 10, we can now determine the unknown parameters  $L_k$ ,  $N_k$ ,  $G_k$ ,  $H_k$ ,  $T_k$ , and  $F_k$  to complete the design of the residual generator  $r_k(t)$ . Clearly, matrix  $H_k$  can be calculated from (50) and condition (56) can be easily verified if  $L_k$  is found. Therefore, the residual generator design now reduces to find the solutions to unknown matrices  $L_k$ ,  $N_k$ ,  $G_k$ ,  $T_k$ , and  $F_k$  which satisfy three conditions (49), (54), and (55). Here, we again assign stable eigenvalues to  $N_k$

and then the fault isolation scheme can be achieved in the following sections.

As in Section 3, let us use matrix  $P$  defined in (14) and the partitions in (15) and (17) and the following partitions:

$$L_k P = [L_{k1} \ L_{k2}], \quad (57)$$

$$P^{-1} D_k = \begin{bmatrix} D_{k1} \\ D_{k2} \end{bmatrix}, \quad (58)$$

where  $L_{k1} \in \mathbb{R}^{q \times p}$ ,  $L_{k2} \in \mathbb{R}^{q \times (n-p)}$ ,  $D_{k1} \in \mathbb{R}^{p \times 1}$ , and  $D_{k2} \in \mathbb{R}^{(n-p) \times 1}$ .

Accordingly, matrix equations (49), (54), and (55) can be partitioned to the following:

$$G_k = L_{k1} A_{11} + L_{k2} A_{21} - N_k L_{k1}, \quad (59)$$

$$F_k = -T_k L_{k1}, \quad (60)$$

$$N_k L_{k2} - L_{k2} A_{22} - L_{k1} A_{12} = 0, \quad (61)$$

$$T_k L_{k2} = 0, \quad (62)$$

$$L_{k1} D_{k1} + L_{k2} D_{k2} = 0. \quad (63)$$

Clearly, with prescribed eigenvalues for  $N_k$  in (48), from three matrix equations (61)–(63) matrices  $L_{k1}$ ,  $L_{k2}$ , and  $T_k$  can be found. Moreover, based on (59) and (60),  $G_k$  and  $F_k$  are calculated, respectively, to complete the design of the residual generator. For this, we will consider two cases, namely, Cases 3 and 4. In Case 3, we present a solution to matrix equations (61)–(63) with only a first-order residual generator designed. Indeed, it is possible that a first-order residual generator exists when  $\mathcal{N}([A_{12} \ D_{k1}]) \neq 0$ . This is discussed in Case 3. When a first-order residual generator does not exist, then in Case 4, based on a parametric solution approach, a residual generator can be constructed with the order as low as  $[(n-p+1)/(p-1)]$ .

*Case 3 (first-order residual generators).* In this case, we design a bank of first-order residual generators  $r_k(t)$ ,  $k = 1, 2, \dots, l$ , to isolate faults in the system. The existence conditions of each residual generator are presented in the following theorem.

**Theorem 11.** *A first-order residual generator  $r_k(t)$  exists if  $\mathcal{N}([A_{12} \ D_{k1}]) \neq 0$  and condition (56) holds.*

*Proof.* The proof of Theorem 11 is similar to that of Theorem 4. For completeness, it is given here. To design a first-order residual generator  $r_k(t)$ , that is,  $q = 1$ ,  $N_k \in \mathbb{R}^{1 \times 1}$  can be assigned as any negative scalar, say,  $N_k = s_k < 0$ . By letting  $L_{k2} = 0$ , we can arbitrarily choose  $T_k \neq 0$  as any real number  $\alpha_k$ ,  $\alpha_k \neq 0$ . Accordingly, (61)–(63) are now reduced to

$$L_{k1} \Theta_k = 0, \quad \text{where } \Theta_k = [A_{12} \ D_{k1}]. \quad (64)$$

As in Case 1 (see also Remark 8), a solution to (64) where  $L_{k1} \neq 0$  always exists if  $\mathcal{N}(\Theta_k) \neq 0$  such that  $\mathcal{N}(\Theta_k) \Theta_k = 0$ .

Therefore, solutions to  $L_{k1}$  according to (64) can be computed by first finding  $\hat{L}_{k1}$ , where

$$\hat{L}_{k1} = \mathcal{N}(\Theta_k) \quad (65)$$

and then  $L_{k1}$  is selected as any row of  $\hat{L}_{k1}$ .

Finally, if condition (56) is satisfied, that is,  $L_k D_R \neq 0$ , the rest of the parameters of the residual generator can be determined. This completes the proof of Theorem 11.  $\square$

*Remark 12.* Note that whenever  $p > (n+1)/2$ ,  $\mathcal{N}(\Theta_k) \neq 0$  always exists since  $\Theta_k$  is a column matrix. As a result, a solution to  $L_{k1} \neq 0$  always exists and a first-order residual generator can exist to isolate faults in the system.

*Case 4 (minimal-order residual generators).* In the case that a bank of first-order residual generators is not possible to design by applying the method presented in Case 3, we can employ a parametric approach to design a bank of minimal-order residual generators to isolate faults in the system. For this, as in Case 2, we assume that the pair  $(C, A)$  is observable. In the remainder of this section, we present a systematic procedure which helps to design each residual generator,  $r_k(t)$ , of the residual bank. For this, we consider the case where  $\text{rank}([A_{12} \ D_{k1}]) = p$  and let us use similar notations as in Case 2 (Section 3) to state the solution of generalized Sylvester equation (61), where

$$N_k = Q_k^{-1} \Lambda_k Q_k, \quad (66)$$

$$L_{k1} = Q_k [U(s_{k1}) b_{k1} \ \cdots \ U(s_{kq}) b_{kq}]^T, \quad (67)$$

$$L_{k2} = Q_k [Z(s_{k1}) b_{k1} \ \cdots \ Z(s_{kq}) b_{kq}]^T. \quad (68)$$

The following theorem presents the existence condition and the minimal order of the residual generator  $r_k(t)$ .

**Theorem 13.** *A residual generator  $r_k(t)$ , with an order as low as  $[(n-p+1)/(p-1)]$ , exists if condition (56) holds.*

*Proof.* The design of residual generator  $r_k(t)$  is reduced to solving three matrix equations (61)–(63) and with the satisfaction of condition (56). Since  $Q_k$  is any invertible matrix, by substituting  $L_{k1}$  and  $L_{k2}$  from (67) and (68) into the transpose of (63), the following equation is obtained:

$$\begin{bmatrix} D_{k1} \\ D_{k2} \end{bmatrix}^T \begin{bmatrix} U(s_{k1}) b_{k1} & U(s_{k2}) b_{k2} & \cdots & U(s_{kq}) b_{kq} \\ Z(s_{k1}) b_{k1} & Z(s_{k2}) b_{k2} & \cdots & Z(s_{kq}) b_{kq} \end{bmatrix} = 0. \quad (69)$$

From (69), the following  $q$  equations are obtained:

$$[D_{k1}^T \ D_{k2}^T] \begin{bmatrix} U(s_{ki}) \\ Z(s_{ki}) \end{bmatrix} b_{ki} = 0, \quad i = 1, 2, \dots, q. \quad (70)$$

It is a fact that there always exist nonsingular matrices  $\Omega_{ki} \in \mathbb{R}^{p \times p}$  ( $i = 1, 2, \dots, q$ ),  $\Omega_{ki}\Omega_{ki}^{-1} = I_p$ , such that the following partitions are satisfied:

$$\begin{aligned} [D_{k1}^\top \ D_{k2}^\top] \begin{bmatrix} U(s_{ki}) \\ Z(s_{ki}) \end{bmatrix} \Omega_{ki} &= [X_{ki1} \ X_{ki2}], \\ \Omega_{ki}^{-1} b_{ki} &= \begin{bmatrix} b_{ki1} \\ b_{ki2} \end{bmatrix}, \end{aligned} \quad (71)$$

where  $X_{ki1} \in \mathbb{R}^{1 \times 1}$ ,  $X_{ki1} \neq 0$ ,  $X_{ki2} \in \mathbb{R}^{1 \times (p-1)}$ ,  $b_{ki1} \in \mathbb{R}^{1 \times 1}$ , and  $b_{ki2} \in \mathbb{R}^{(p-1) \times 1}$ .

From (70) and (71), the following equations are obtained:

$$b_{ki} = \Omega_{ki} \begin{bmatrix} -X_{ki1}^{-1} X_{ki2} \\ I_{(p-1)} \end{bmatrix} b_{ki2}, \quad i = 1, 2, \dots, q. \quad (72)$$

Let us denote that

$$T_k Q_k = [t_{k1} \ t_{k2} \ \dots \ t_{kq}], \quad (73)$$

where  $t_{ki} \neq 0$ ,  $i = 1, 2, \dots, q$ , are arbitrarily chosen scalars.

By substituting  $L_{k2}$  from (68) and  $T_k Q_k$  from (73) into the transpose of (62), we obtain

$$\begin{bmatrix} Z(s_{k1}) t_{k1} & Z(s_{k2}) t_{k2} & \dots & Z(s_{kq}) t_{kq} \end{bmatrix} \begin{bmatrix} b_{k1} \\ b_{k2} \\ \vdots \\ b_{kq} \end{bmatrix} = 0. \quad (74)$$

Substituting (72) into (74), the following equation is obtained:

$$\Phi_k \tilde{b}_k = 0, \quad (75)$$

where  $\Phi_k \in \mathbb{R}^{(n-p) \times q(p-1)}$  and  $\tilde{b}_k \in \mathbb{R}^{q(p-1) \times 1}$  and

$$\begin{aligned} \tilde{b}_k &= [b_{k12}^\top \ b_{k22}^\top \ \dots \ b_{kq2}^\top]^\top, \\ \Phi_{ki} &= t_{ki} Z(s_{ki}) \Omega_{ki} \begin{bmatrix} -X_{ki1}^{-1} X_{ki2} \\ I_{(p-1)} \end{bmatrix}, \\ \Phi_k &= [\Phi_{k1} \ \Phi_{k2} \ \dots \ \Phi_{kq}]. \end{aligned} \quad (76)$$

As in (38), if we define  $\mathcal{N}(\Phi_k)$  such that  $\Phi_k \mathcal{N}(\Phi_k) = 0$ , then a solution to  $\tilde{b}_k \neq 0$  exists if  $\mathcal{N}(\Phi_k) \neq 0$ . Specifically,  $\mathcal{N}(\Phi_k) \neq 0$  always exists if the order  $q$  satisfies the following condition:

$$q \geq \left\lceil \frac{n-p+1}{p-1} \right\rceil, \quad (77)$$

and then  $\tilde{b}_k$  can be taken to be any column of matrix  $\hat{b}_k$ , where  $\hat{b}_k$  is computed as

$$\hat{b}_k = \mathcal{N}(\Phi_k). \quad (78)$$

It is noted that if we assign an order  $q$  such that condition (77) holds, a solution to  $\tilde{b}_k \neq 0$  always exists. Therefore, with an attempt to minimize the residual generator order, we only need to prescribe the order as low as  $q = \lceil (n-p+1)/(p-1) \rceil$  which is identified to be the lowest order satisfying (77).

Consequently, with prescribed eigenvalues of  $N_k$  and arbitrarily chosen  $t_{ki} \neq 0$ , matrices  $L_{k1}$ ,  $L_{k2}$ , and  $L_k$  are then easily calculated from (67), (68), and (57), respectively. If condition (56) is satisfied, a minimal-order residual generator exists to isolate the faults. The remaining parameters  $G_k$ ,  $F_k$ , and  $T_k$  can be obtained based on (59), (60), and (73), respectively, to complete the design of residual generator  $r_k(t)$ . This completes the proof of Theorem 13.  $\square$

## 5. Numerical Examples

*5.1. Example 1.* In this example, we consider timely detection of faults in a dynamical system which has  $n = 8$ ,  $p = 5$ ,  $m = 2$ , and  $l = 2$ . Since we have the case where  $p > (n+1)/2$  and as discussed in Case 1 (Section 3) and Case 3 (Section 4), we can indeed design only first-order residual generators to effectively detect and isolate the faults in the system. For this example, the system matrices  $C = [I_5 \ 0]$ ,  $A$ ,  $B$ , and  $D$  are as given below:

$$\begin{aligned} A &= \begin{bmatrix} -1 & 0 & 1 & 1 & -2 & 1 & -3 & 0 \\ 0 & -5 & 3 & -4 & 0 & 1 & 0 & -1 \\ 1 & 1 & -8 & 3 & 0 & 2 & 1 & -2 \\ -4 & 0 & 2 & -6 & 1 & -5 & -2 & 1 \\ 1 & 0 & 0 & 1 & -1 & 2 & 0 & 2 \\ -2 & 0 & 1 & 2 & 0 & -3 & -1 & 1 \\ 2 & 1 & -1 & 0 & -1 & 0 & -8 & 3 \\ -2 & 0 & -1 & -4 & 0 & -6 & -3 & -2 \end{bmatrix}, \\ B &= \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 1 & -1 \\ 0 & 1 \\ 3 & 1 \\ 1 & -3 \\ -1 & 0 \\ 1 & 1 \end{bmatrix}, \\ D &= \begin{bmatrix} 4 & 2 \\ -2.5 & -1.5 \\ 2 & 3 \\ -1 & -1 \\ 1 & 0 \\ -1 & 2 \\ -1 & 0 \\ 4 & 2.5 \end{bmatrix}. \end{aligned} \quad (79)$$

Since  $C$  is already in the desired form, that is,  $C = [I_5 \ 0]$ ,  $P$  is an identity matrix; that is,  $P = I_8$ . Accordingly, by (17), submatrices  $A_{11}$ ,  $A_{12}$ ,  $A_{21}$ , and  $A_{22}$  are obtained, where

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \left[ \begin{array}{ccccc|ccc} -1 & 0 & 1 & 1 & -2 & 1 & -3 & 0 \\ 0 & -5 & 3 & -4 & 0 & 1 & 0 & -1 \\ 1 & 1 & -8 & 3 & 0 & 2 & 1 & -2 \\ -4 & 0 & 2 & -6 & 1 & -5 & -2 & 1 \\ 1 & 0 & 0 & 1 & -1 & 2 & 0 & 2 \\ \hline -2 & 0 & 1 & 2 & 0 & -3 & -1 & 1 \\ 2 & 1 & -1 & 0 & -1 & 0 & -8 & 3 \\ -2 & 0 & -1 & -4 & 0 & -6 & -3 & -2 \end{array} \right]. \quad (80)$$

Now, the design of first-order residual generators can be readily carried out to detect and isolate the faults in the system.

**5.1.1. First-Order Residual Generator Detects Faults in the System.** It is clear that  $A_{12}$  is a column matrix and thus its null-space exists; that is,  $\mathcal{N}(A_{12}) \neq 0$ . As discussed in Case 1 (Section 3), a first-order functional observer exists for the FD purpose. Let us assign  $N = -8$ ;  $L_1$  is then computed according to (25), where

$$L_1 = [4 \ -27 \ 14 \ 1 \ 0]. \quad (81)$$

Since letting  $L_2 = [0 \ 0 \ 0]$  and by (16), matrix  $L$  is obtained as

$$L = [4 \ -27 \ 14 \ 1 \ 0 \ 0 \ 0 \ 0]. \quad (82)$$

By that, condition (13) is found to be satisfied since  $LD = [110.5 \ 89.5] \neq 0$ . Hence, a first-order residual generator now can be designed to detect faults in the system. We complete the design by first choosing  $T = 0.2$  and matrices  $H$ ,  $G$ , and  $F$  are then obtained based on (9), (19), and (20), respectively, where

$$\begin{aligned} H &= [-36 \ -40], \\ G &= [38 \ -67 \ -75 \ 156 \ -7], \\ F &= [-0.8 \ 5.4 \ -2.8 \ -0.2 \ 0]. \end{aligned} \quad (83)$$

Figure 2 shows that the first-order residual generator can detect faults  $f_1(t)$  and  $f_2(t)$  in the system. Fault  $f_1(t)$  appears at the time  $t = 10$  s and clears from the time  $t = 20$  s. Fault  $f_2(t)$  happens from the time  $t = 30$  s to the time  $t = 40$  s. It is clear from the figure that while the faults are happening, the residual generator triggers them, whereas when the faults clear, the residual generator converges to zero as expected. Note also that the residual generator is insensitive to the inputs  $u(t)$ . It is clear from this example that the residual generator is designed by using a significantly lower-order (only first-order) functional observer. In contrast, existing

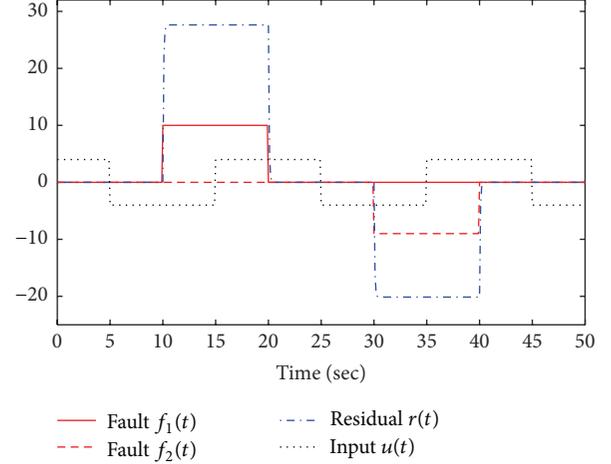


FIGURE 2: First-order observer-based residual generator detects faults in the system.

FD schemes using full-order or reduced-order state observers would give higher-order schemes. This example thus serves to highlight the attractiveness of our FD scheme proposed in this paper.

**5.1.2. First-Order Residual Bank Isolates Faults in the System.** In this section, we demonstrate our solution approach to isolate the faults in the system as presented in Case 3 (Section 4). For this system, there are two possible faults; thus we design a bank of two first-order residual generators,  $r_1(t)$  and  $r_2(t)$ , to isolate the faults. Let us assign the eigenvalue for both residuals as  $N_k = -6$ ,  $k = 1, 2$ . Now the conditions, which ensure the existence of each residual generator, will be verified and the design of the residual bank will be carried out.

**Residual Generator  $r_1(t)$ .** Here,  $r_1(t)$  is designed to be insensitive to fault  $f_1(t)$  but sensitive to  $f_2(t)$ . From partitions (43) and (58),  $D_1$ ,  $D_R$ , and  $D_{11}$  are as given below:

$$\begin{aligned} D_1 &= [4 \ -2.5 \ 2 \ -1 \ 1 \ -1 \ -1 \ 4]^T, \\ D_R &= [2 \ -1.5 \ 3 \ -1 \ 0 \ 2 \ 0 \ 2.5]^T, \\ D_{11} &= [4 \ -2.5 \ 2 \ -1 \ 1]^T. \end{aligned} \quad (84)$$

According to (65),  $L_{11}$  is found by computing  $\mathcal{N}(\Theta_1)$ . This gives

$$L_{11} = [-0.2353 \ 0.5882 \ 1.1765 \ 0.9412 \ 1]. \quad (85)$$

Since  $L_{12} = [0 \ 0 \ 0]$ , from (57),  $L_1 = [L_{11} \ 0 \ 0 \ 0]$  and condition (56) is satisfied since  $L_1 D_R = 1.2353 \neq 0$ . Hence, based on Theorem 11, first-order residual generator  $r_1(t)$  exists for FI. Now, by assigning  $T_1 = 8$ , matrices  $H_1$ ,  $G_1$ , and  $F_1$  are calculated based on (50), (59), and (60),

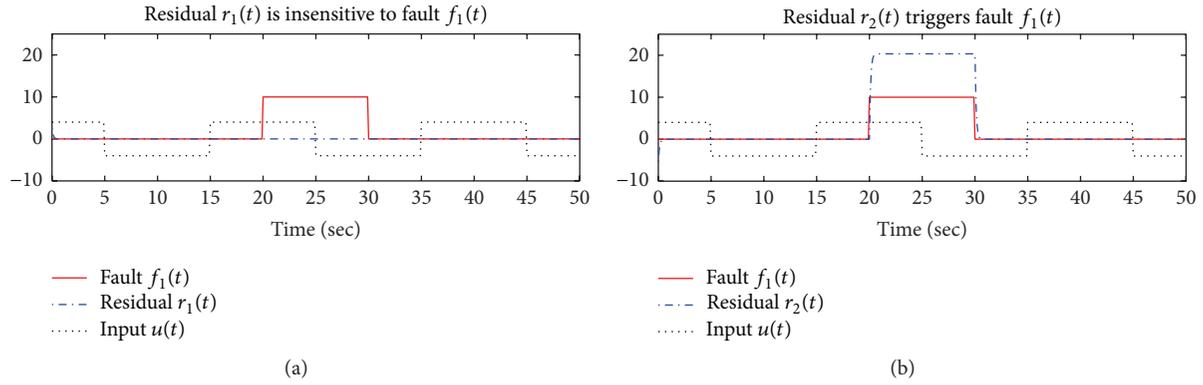


FIGURE 3: First-order observer-based residual bank isolates faults in the system.

respectively, to complete the design of residual generator  $r_1(t)$ , where

$$\begin{aligned} H_1 &= [5.1176 \quad 1.3529], \\ F_1 &= [1.8824 \quad -4.7059 \quad -9.4118 \quad -7.5294 \quad -8], \\ G_1 &= [-2.7647 \quad 1.7647 \quad 1.0588 \quad 1.9412 \quad 6.4118]. \end{aligned} \quad (86)$$

*Residual Generator  $r_2(t)$ .* By applying the same process as for the residual generator  $r_1(t)$ , we have the following results for the residual generator  $r_2(t)$ , which is insensitive to fault  $f_2(t)$  and sensitive to fault  $f_1(t)$ :

$$\begin{aligned} D_2 &= [2 \quad -1.5 \quad 3 \quad -1 \quad 0 \quad 2 \quad 0 \quad 2.5]^T, \\ D_R &= [4 \quad -2.5 \quad 2 \quad -1 \quad 1 \quad -1 \quad -1 \quad 4]^T, \\ D_{21} &= [2 \quad -1.5 \quad 3 \quad -1 \quad 0]^T, \\ L_{21} &= [-0.2905 \quad 0.9609 \quad 0.9832 \quad 0.9274 \quad 1], \\ L_2 &= [L_{21} \quad 0 \quad 0 \quad 0]. \end{aligned} \quad (87)$$

Since  $L_2 D_R = -1.5251 \neq 0$ , condition (56) holds. Consequently, the residual generator  $r_2(t)$  exists with a first order. Now, we can complete the design of  $r_2(t)$  by calculating matrices  $H_2$ ,  $G_2$ , and  $F_2$ , where

$$\begin{aligned} H_2 &= [5.6145 \quad 1.9050], \\ F_2 &= [2.3240 \quad -7.6872 \quad -7.8659 \quad -7.4190 \quad -8] \\ G_2 &= [-3.1788 \quad 1.9441 \quad 2.4804 \quad -0.1844 \quad 6.5084]. \end{aligned} \quad (88)$$

Figure 3 shows that a bank of two first-order residual generators can effectively and timely isolate the faults in the system. Here, fault  $f_1(t)$  happens in the system from the time 20s and clears from the time 30s. By that, the output of residual generator  $r_1(t)$  remains as zero. Meanwhile, that of the residual generator  $r_2(t)$  derives from zero. Consequently, by observing the residual outputs, we can identify that fault  $f_1(t)$  happened in the system. Clearly, the residual bank can effectively and timely isolate the faults in the system. This example again serves to illustrate the simplicity of our proposed FI scheme using minimal-order functional observers.

*5.2. Example 2.* This example is given to demonstrate Case 2 (Section 3) and Case 4 (Section 4) by considering a system with  $n = 8$ ,  $p = 3$ ,  $m = 2$ , and  $l = 3$ , where  $C = [I_3 \quad 0]$ , and  $A$ ,  $B$ , and  $D$  are as given below:

$$\begin{aligned} A &= \left[ \begin{array}{cc|ccc} A_{11} & A_{12} & & & \\ A_{21} & A_{22} & & & \end{array} \right] \\ &= \left[ \begin{array}{ccc|ccc} -5 & 2 & 0 & 1 & 0 & 1 & 9 & -1 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 1 \\ 1 & 0 & -5 & 3 & 4 & 0 & -3 & 2 \\ \hline -3 & 1 & 1 & -8 & 3 & 1 & 4 & 2 \\ -2 & -4 & 0 & 0 & -6 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & -2 & -3 & 3 & -1 \\ 2 & 0 & 2 & 3 & 7 & 1 & -7 & 0 \\ 1 & 1 & 0 & -2 & 1 & 0 & 0 & -1 \end{array} \right], \\ B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \\ 0 & 0 \\ 0 & 1 \\ 1 & -2 \\ 1 & 1 \\ -1 & 0 \end{bmatrix}, \\ D &= \begin{bmatrix} 1 & -2 & 1 \\ -2 & -1 & -2 \\ -2 & 1 & 3 \\ 0 & -3 & 1 \\ 2 & 0 & -1 \\ 2 & 1 & 0 \\ -2 & -1 & 1 \\ 1 & -1 & 2 \end{bmatrix}. \end{aligned} \quad (89)$$

From (30)–(34), we obtain the following pair of coprime polynomial matrices  $U(s)$  and  $Z(s)$ :

$$U(s) = (s^5 + 25s^4 + 214s^3 + 762s^2 + 1118s + 582)I_3, \quad (90)$$

$$Z(s) = (\Upsilon_1 s^4 + \Upsilon_2 s^3 + \Upsilon_3 s^2 + \Upsilon_4 s + \Upsilon_5) A_{12}^\top,$$

where

$$\begin{aligned} \Upsilon_1 &= I_5, \\ \Upsilon_2 &= \begin{bmatrix} 17 & 0 & 1 & 3 & -2 \\ 3 & 19 & -2 & 7 & 1 \\ 1 & 0 & 22 & 1 & 0 \\ 4 & 1 & 3 & 18 & 0 \\ 2 & 0 & -1 & 0 & 24 \end{bmatrix}, \\ \Upsilon_3 &= \begin{bmatrix} 87 & 3 & 25 & 31 & -32 \\ 61 & 107 & -9 & 91 & 12 \\ 18 & 1 & 152 & 18 & -2 \\ 46 & 12 & 47 & 110 & -7 \\ 31 & 0 & -19 & 5 & 186 \end{bmatrix}, \\ \Upsilon_4 &= \begin{bmatrix} 160 & 13 & 131 & 90 & -139 \\ 212 & 211 & 135 & 286 & -27 \\ 79 & 12 & 378 & 87 & -33 \\ 141 & 38 & 218 & 261 & -73 \\ 125 & 5 & -83 & 39 & 514 \end{bmatrix}, \\ \Upsilon_5 &= \begin{bmatrix} 90 & 12 & 150 & 72 & -168 \\ 162 & 138 & 270 & 246 & -186 \\ 64 & 15 & 333 & 90 & -113 \\ 102 & 33 & 267 & 198 & -171 \\ 116 & 9 & -33 & 54 & 359 \end{bmatrix}. \end{aligned} \quad (91)$$

For this example, by applying the scheme discussed in Case 2, we only need to design a residual generator based on a second-order functional observer to detect the faults in the system. Furthermore, for isolating the faults, we use only a bank of three residual generators which are designed based on third-order functional observers. The design is carried out in the following sections.

**5.2.1. Second-Order Residual Generator Detects Faults in the System.** As  $A_{12}$  has full-row rank, that is,  $\text{rank}(A_{12}) = 3$  and  $1 < p < n/2$ , this falls into Case 2 (Section 3). By that, we can design a residual generator using a functional observer which has an order of  $q = [(8 - 3 + 1)/3] = 2$ , that is, second order, to detect the faults in the system.

Let us assign the eigenvalues of  $N$  to be  $s_1 = -4$  and  $s_2 = -5$  and choose  $Q = I_2$ . Also, scalars  $t_i$  are chosen as  $TQ = [-0.5 \quad -0.5]$ . From (38),  $\mathcal{X}$  is obtained, where

$$\mathcal{X}^\top = \begin{bmatrix} -56.0 & -376 & 110.0 & -63.0 & 77.0 \\ -25.0 & -191 & 65.5 & -31.5 & 35.5 \\ -11.0 & -124 & 38.0 & -27.0 & 14.0 \\ -90.5 & -846 & 35.0 & -83.5 & 80.0 \\ -2.0 & -256 & 68.0 & -30.0 & -8.0 \\ 64.5 & -194 & 17.0 & -98.5 & -80.0 \end{bmatrix}. \quad (92)$$

Clearly,  $\mathcal{X}$  is a row matrix; as a result,  $b \neq 0$  exists and is obtained by taking any column of  $\mathcal{N}(\mathcal{X})$  in (38). Accordingly,  $b_1$  and  $b_2$  are obtained, where

$$\begin{aligned} b_1 &= [0.499 \quad -0.788 \quad -0.323]^\top, \\ b_2 &= [-0.046 \quad 0.152 \quad 0.014]^\top. \end{aligned} \quad (93)$$

From (27) and (28),  $L_1$  and  $L_2$  are obtained:

$$\begin{aligned} L_1 &= \begin{bmatrix} -8.991 & 14.184 & 5.813 \\ 9.521 & -31.531 & -2.842 \end{bmatrix}, \\ L_2 &= \begin{bmatrix} 9.441 & -9.441 \\ -5.465 & 5.465 \\ 17.877 & -17.877 \\ -4.143 & 4.143 \\ -11.935 & 11.935 \end{bmatrix}^\top. \end{aligned} \quad (94)$$

Since  $P = I_8$ ,  $L = [L_1 \quad L_2]$  and it is easy to verify condition (13) that

$$LD = \begin{bmatrix} -27.811 & 15.241 & -33.030 \\ 57.093 & 4.016 & 77.164 \end{bmatrix} \neq 0. \quad (95)$$

Thus, condition (13) is satisfied and hence a residual generator using a second-order functional observer can be constructed to detect the faults. The parameters of the observer and the residual generator are obtained by (26), (19), (9), (20), and (36), where

$$\begin{aligned} N &= \begin{bmatrix} -4 & 0 \\ 0 & -5 \end{bmatrix}, \\ G &= \begin{bmatrix} -8.628 & 43.934 & -22.533 \\ 3.243 & -126.445 & 16.720 \end{bmatrix}, \\ H &= \begin{bmatrix} 22.491 & -36.989 \\ -18.990 & 16.672 \end{bmatrix}, \\ F &= [0.265 \quad -8.673 \quad 1.485], \\ T &= [-0.5 \quad -0.5]. \end{aligned} \quad (96)$$

Figure 4 indicates that the residual generator can detect the faults  $f_1(t)$ ,  $f_2(t)$ , and  $f_3(t)$  in the system. It is clear in this example that the design of the residual generator is very simple and systematic.

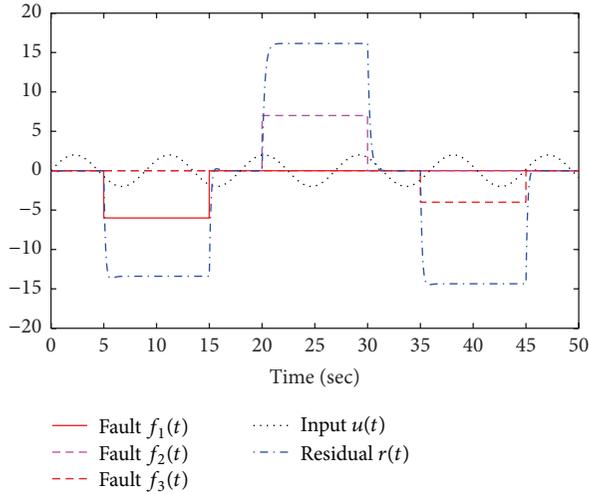


FIGURE 4: Second-order observer-based residual generator detects the faults in the system.

**5.2.2. Third-Order Residual Bank Isolates the Faults in the System.** By applying the scheme in Case 4, for this system, we now proceed to design a bank of three residual generators which are based on third-order functional observers,  $q = [(8 - 3 + 1)/(3 - 1)] = 3$ , to isolate the faults happening in the system.

To design the residual bank, we, for the sake of simplicity, firstly assign the poles of  $N_k$  to be at  $s_{k1} = -4$ ,  $s_{k2} = -5$ , and  $s_{k3} = -7$ ,  $k = 1, 2, 3$ .  $Q_k$  is chosen as  $Q_k = I_3$ , and  $T_k Q_k$  is chosen as  $T_k Q_k = [1 \ 1 \ 1]$  for all the three residual generators. In the remainder of this example, we will verify the conditions, which guarantee the existence of each residual generator, and complete the design by determining all the necessary parameters.

**Residual Generator  $r_1(t)$ .** This residual generator is designed such that it is insensitive to fault  $f_1(t)$  but sensitive to faults  $f_2(t)$  and  $f_3(t)$ . From partitions (43) and (58),  $D_1$ ,  $D_R$ ,  $D_{11}$ , and  $D_{12}$  are obtained as

$$\begin{aligned} D_1 &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ -2 \ -1]^T, \\ D_R &= \begin{bmatrix} -2 & -1 & 1 & -3 & 0 & 1 & -1 & -1 \\ 1 & -2 & 3 & 1 & -1 & 0 & 1 & 2 \end{bmatrix}^T, \end{aligned} \quad (97)$$

$$D_{11} = [1 \ 1 \ 0]^T,$$

$$D_{12} = [0 \ 1 \ 0 \ -2 \ -1]^T.$$

From (75),  $\Phi_1$  is obtained, where

$$\Phi_1^T = \begin{bmatrix} -4.42 & 16.64 & -24.11 & 1.78 & 3.82 \\ -7.59 & 49.36 & -17.89 & 20.72 & 12.68 \\ -19.21 & 295.01 & -127.02 & 38.58 & 36.52 \\ -106.06 & 602.41 & -42.87 & 218.16 & 139.72 \\ -44.15 & 119.43 & -15.88 & 6.03 & 29.01 \\ -174.63 & 246.42 & 19.41 & 83.22 & 93.74 \end{bmatrix}. \quad (98)$$

Since  $\Phi_1$  is a row matrix,  $\tilde{b}_1 \neq 0$  in (75) exists and is taken from  $\mathcal{N}(\Phi_1)$ . Accordingly, from (72),  $b_{11}$ ,  $b_{12}$ , and  $b_{13}$  are obtained, where

$$\begin{aligned} b_{11} &= [0.142 \ -0.684 \ 0.721]^T, \\ b_{12} &= [-0.018 \ 0.057 \ -0.083]^T, \\ b_{13} &= [-0.007 \ 0.026 \ 0.023]^T. \end{aligned} \quad (99)$$

From (67) and (68),  $L_{11}$  and  $L_{12}$  are obtained:

$$\begin{aligned} L_{11} &= \begin{bmatrix} -2.555 & 12.318 & -12.985 \\ 3.685 & -11.767 & 17.173 \\ 0.620 & -2.302 & -2.108 \end{bmatrix}, \\ L_{12} &= \begin{bmatrix} -2.450 & 7.670 & -5.219 \\ 24.219 & -33.046 & 8.827 \\ 3.598 & -3.646 & 0.048 \\ 13.725 & -15.829 & 2.103 \\ 6.532 & -9.470 & 2.938 \end{bmatrix}^T. \end{aligned} \quad (100)$$

By (57) and  $P = I_8$  that  $L_1 = [L_{11} \ L_{12}]$  and condition (56) is satisfied since

$$L_1 D_R = \begin{bmatrix} -29.503 & 20.212 & 9.619 \\ -66.028 & 84.683 & -7.166 \end{bmatrix}^T \neq 0. \quad (101)$$

Thus, residual generator  $r_1(t)$  based on a third-order functional observer exists, and from (50), (59), (60), and (73), the rest of the parameters of the residual generator are obtained, where

$$\begin{aligned} H_1 &= \begin{bmatrix} -4.749 & 56.053 \\ 10.852 & -70.521 \\ -2.273 & 10.639 \end{bmatrix}, \\ G_1 &= \begin{bmatrix} -5.217 & -60.950 & 34.388 \\ 7.361 & 90.688 & -20.341 \\ 1.980 & -50.161 & -5.276 \end{bmatrix}, \\ F_1 &= [-1.751 \ 1.751 \ -2.079], \\ T_1 &= [1 \ 1 \ 1]. \end{aligned} \quad (102)$$

By applying the same procedure as for designing  $r_1(t)$ , the following results are obtained for residual generators  $r_2(t)$  and  $r_3(t)$ .

*Residual Generator  $r_2(t)$ .* This residual generator is designed such that it is insensitive to fault  $f_2(t)$  but sensitive to faults  $f_1(t)$  and  $f_3(t)$ . The following results are obtained:

$$\begin{aligned}
 D_2 &= [-2 \ -1 \ 1 \ -3 \ 0 \ 1 \ -1 \ -1]^\top, \\
 D_R &= \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & -2 & -1 \\ 1 & -2 & 3 & 1 & -1 & 0 & 1 & 2 \end{bmatrix}^\top, \\
 D_{21} &= [-2 \ -1 \ 1]^\top, \\
 D_{22} &= [-3 \ 0 \ 1 \ -1 \ -1]^\top, \\
 b_{21} &= [0.311 \ -0.870 \ 0.451]^\top, \\
 b_{22} &= [-0.038 \ 0.143 \ -0.066]^\top, \\
 b_{23} &= [0.012 \ -0.114 \ 0.037]^\top, \\
 L_{21} &= \begin{bmatrix} -5.595 & 15.664 & -8.121 \\ 7.945 & -29.678 & 13.729 \\ -1.099 & 10.261 & -3.347 \end{bmatrix}, \\
 L_{22} &= \begin{bmatrix} 1.229 & 2.172 & -3.401 \\ 13.219 & -17.184 & 3.965 \\ 11.325 & -14.487 & 3.161 \\ 8.705 & -10.821 & 2.115 \\ 1.283 & -2.166 & 0.883 \end{bmatrix}^\top, \\
 L_2 D_R &= \begin{bmatrix} 12.279 & -4.025 & -4.020 \\ -62.006 & 112.689 & -35.150 \end{bmatrix}^\top \neq 0.
 \end{aligned} \tag{103}$$

Since  $L_2 D_R \neq 0$ , condition (56) is satisfied. Hence, residual generator  $r_2(t)$  using a third-order functional observer exists and its design is completed by obtaining the following parameters:

$$\begin{aligned}
 H_2 &= \begin{bmatrix} 5.031 & 23.0595 \\ -1.468 & -42.438 \\ -0.053 & 13.366 \end{bmatrix}, \\
 G_2 &= \begin{bmatrix} 1.708 & -14.560 & 15.436 \\ -11.903 & -34.079 & -4.983 \\ 12.100 & 40.992 & -9.027 \end{bmatrix}, \\
 F_2 &= [-1.251 \ 3.752 \ -2.260], \\
 T_2 &= [1 \ 1 \ 1].
 \end{aligned} \tag{104}$$

*Residual Generator  $r_3(t)$ .* This residual generator is designed such that it is insensitive to fault  $f_3(t)$  but sensitive to faults  $f_1(t)$  and  $f_2(t)$ . The following results are obtained:

$$\begin{aligned}
 D_3 &= [1 \ -2 \ 3 \ 1 \ -1 \ 0 \ 1 \ 2]^\top, \\
 D_R &= \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & -2 & -1 \\ -2 & -1 & 1 & -3 & 0 & 1 & -1 & -1 \end{bmatrix}^\top, \\
 D_{31} &= [1 \ -2 \ 3]^\top, \\
 D_{32} &= [1 \ -1 \ 0 \ 1 \ 2]^\top, \\
 b_{31} &= [0.005 \ -0.515 \ 0.846]^\top, \\
 b_{32} &= [-0.003 \ 0.051 \ -0.107]^\top, \\
 b_{33} &= [-0.007 \ -0.031 \ 0.056]^\top, \\
 L_{31} &= \begin{bmatrix} -0.096 & 9.269 & -15.237 \\ 0.673 & -10.609 & 22.336 \\ 0.617 & 2.822 & -5.040 \end{bmatrix}, \\
 L_{32} &= \begin{bmatrix} -6.524 & 13.4706 & -6.9468 \\ 17.257 & -21.027 & 3.770 \\ 1.943 & -3.059 & 1.116 \\ 13.946 & -18.635 & 4.689 \\ 12.033 & -15.847 & 3.814 \end{bmatrix}^\top, \\
 L_3 D_R &= \begin{bmatrix} 34.384 & -49.531 & 9.259 \\ -28.778 & 22.610 & 4.358 \end{bmatrix}^\top \neq 0.
 \end{aligned} \tag{105}$$

Since  $L_3 D_R \neq 0$ , condition (56) is satisfied. Hence, the residual  $r_3(t)$  exists with a third order. The rest of the parameters of the residual are obtained, where

$$\begin{aligned}
 H_3 &= \begin{bmatrix} -11.478 & 51.822 \\ 17.162 & -66.490 \\ -2.432 & 14.090 \end{bmatrix}, \\
 G_3 &= \begin{bmatrix} 19.109 & -35.905 & 34.661 \\ -39.747 & 40.643 & -20.740 \\ 25.508 & -0.046 & -8.764 \end{bmatrix}, \\
 F_3 &= [-1.194 \ -1.482 \ -2.059], \\
 T_3 &= [1 \ 1 \ 1].
 \end{aligned} \tag{106}$$

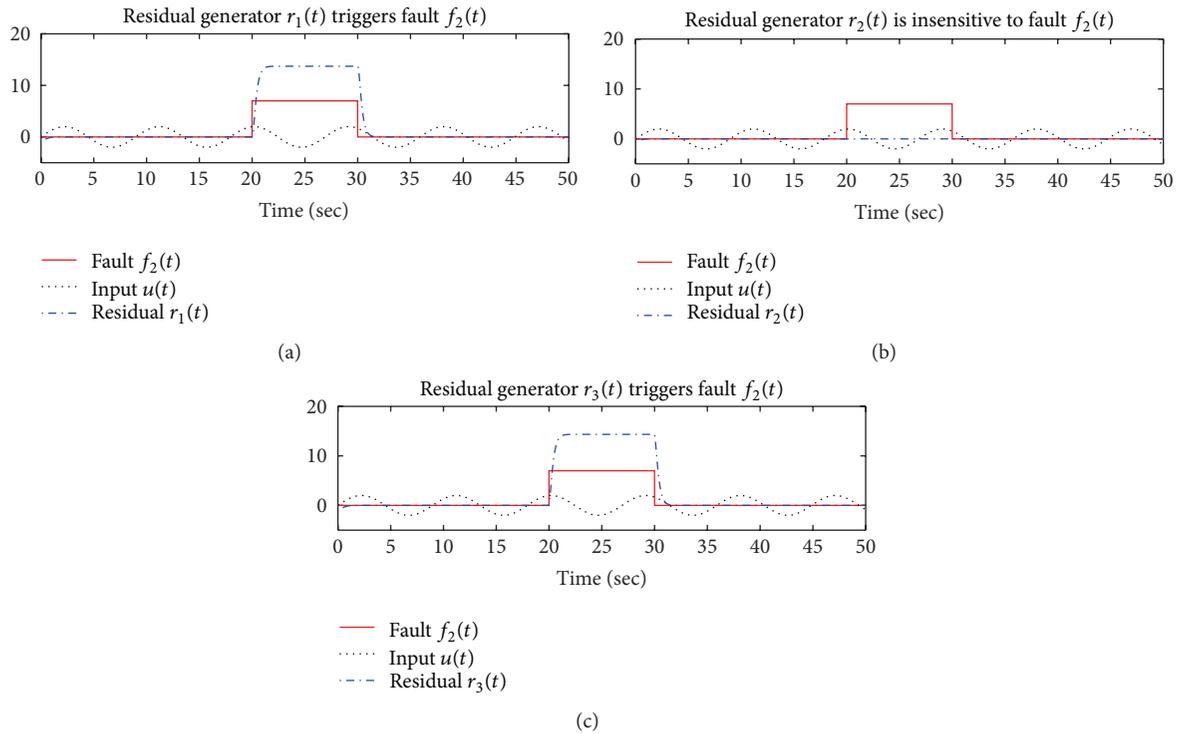


FIGURE 5: Third-order observer-based residual bank isolates the faults in the system.

Figure 5 indicates that a bank of three third-order residual generators can isolate the faults  $f_1(t)$ ,  $f_2(t)$ , and  $f_3(t)$  in the system. In the figure, when fault  $f_2(t)$  occurs in the system, the residual  $r_2(t)$  is insensitive to the fault; thus its output remains as zero. Meanwhile, the residual generators  $r_1(t)$  and  $r_3(t)$  trigger the fault. By that we can isolate fault  $f_2(t)$  in the system. This example again further highlights the efficiency of our proposed FI scheme in this paper.

## 6. Conclusion

This work has presented novel FD and FI schemes using minimal-order functional observers to construct residual generators to timely detect and isolate actuator faults in dynamical systems. The proposed method is based on solving a generalized Sylvester matrix equation via a parametric approach. Existence conditions and systematic procedures for designing the proposed FD and FI schemes have been presented. The reduced order and the simplicity are the hallmark of the proposed novel FD and FI schemes. It is envisaged that these proposed schemes will have widespread applications in control systems to detect and isolate actuator and/or component faults.

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] 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.
- [2] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.
- [3] M. Hou and P. C. Muller, "Design of observers for linear systems with unknown inputs," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 871–875, 1992.
- [4] M. Aldeen and R. Sharma, "Estimation of states, faults and unknown disturbances in non-linear systems," *International Journal of Control*, vol. 81, no. 8, pp. 1195–1201, 2008.
- [5] Q. P. Ha and H. Trinh, "State and input simultaneous estimation for a class of nonlinear systems," *Automatica*, vol. 40, no. 10, pp. 1779–1785, 2004.
- [6] Z. Gao and H. Wang, "Descriptor observer approaches for multivariable systems with measurement noises and application in fault detection and diagnosis," *Systems & Control Letters*, vol. 55, no. 4, pp. 304–313, 2006.
- [7] H. Trinh and Q. P. Ha, "State and input simultaneous estimation for a class of time-delay systems with uncertainties," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 6, pp. 527–531, 2007.
- [8] S. S. Delshad, A. Johansson, M. Darouach, and T. Gustafsson, "Robust state estimation and unknown inputs reconstruction

- for a class of nonlinear systems: multiobjective approach," *Automatica*, vol. 64, pp. 1–7, 2016.
- [9] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic, Norwell, Mass, USA, 1999.
- [10] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.
- [11] J. Yang and F. Zhu, "FDI design for uncertain nonlinear systems with both actuator and sensor faults," *Asian Journal of Control*, vol. 17, no. 1, pp. 213–224, 2015.
- [12] P. M. Frank and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of Process Control*, vol. 7, no. 6, pp. 403–424, 1997.
- [13] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: robustness and applications," *Control Engineering Practice*, vol. 5, no. 5, pp. 671–682, 1997.
- [14] G. R. Duan and R. J. Patton, "Robust fault detection using Luenbergertype unknown input observers—a parametric approach," *International Journal of Systems Science*, vol. 32, no. 4, pp. 533–540, 2001.
- [15] J. Su, W.-H. Chen, and B. Li, "Disturbance observer based fault diagnosis," in *Proceedings of the 33rd Chinese Control Conference (CCC '14)*, pp. 3024–3029, Nanjing, China, July 2014.
- [16] G. R. Duan and R. J. Patton, "Robust fault detection in linear systems using Luenberger observers," in *Proceedings of the UKACC International Conference on Control*, vol. 2, pp. 1468–1473, Swansea, UK, September 1998.
- [17] W. Ge and C. Z. Fang, "Detection of faulty components via robust observation," *International Journal of Control*, vol. 47, no. 2, pp. 581–599, 1988.
- [18] H. Liu, D. Liu, C. Lu, and X. Wang, "Fault diagnosis of hydraulic servo system using the unscented Kalman Filter," *Asian Journal of Control*, vol. 16, no. 6, pp. 1713–1725, 2014.
- [19] F. Zhu and F. Cen, "Full-order observer-based actuator fault detection and reduced-order observer-based fault reconstruction for a class of uncertain nonlinear systems," *Journal of Process Control*, vol. 20, no. 10, pp. 1141–1149, 2010.
- [20] C. Edwards, S. K. Spurgeon, and R. J. Patton, "Sliding mode observers for fault detection and isolation," *Automatica*, vol. 36, no. 4, pp. 541–553, 2000.
- [21] O. A. Z. Sotomayor and D. Odloak, "Observer-based fault diagnosis in chemical plants," *Chemical Engineering Journal*, vol. 112, no. 1–3, pp. 93–108, 2005.
- [22] J. Chen, R. J. Patton, and H.-Y. Zhang, "Design of unknown input observers and robust fault detection filters," *International Journal of Control*, vol. 63, no. 1, pp. 85–105, 1996.
- [23] P. Rosa and C. Silvestre, "Fault detection and isolation of LPV systems using set-valued observers: an application to a fixed-wing aircraft," *Control Engineering Practice*, vol. 21, no. 3, pp. 242–252, 2013.
- [24] T.-G. Park, "Designing fault detection observers for linear systems with mismatched unknown inputs," *Journal of Process Control*, vol. 23, no. 8, pp. 1185–1196, 2013.
- [25] F. Xu, V. Puig, C. Ocampo-Martinez, F. Stoican, and S. Oлару, "Actuator-fault detection and isolation based on set-theoretic approaches," *Journal of Process Control*, vol. 24, no. 6, pp. 947–956, 2014.
- [26] M. Hou and P. C. Muller, "Fault detection and isolation observers," *International Journal of Control*, vol. 60, no. 5, pp. 827–846, 1994.
- [27] J. Gertler, "Fault detection and isolation using parity relations," *Control Engineering Practice*, vol. 5, no. 5, pp. 653–661, 1997.
- [28] D. Koenig and S. Mammar, "Design of a class of reduced order unknown inputs nonlinear observer for fault diagnosis," in *Proceedings of the American Control Conference*, vol. 3, pp. 2143–2147, Arlington, Va, USA, June 2001.
- [29] A. Varga, "On designing least order residual generators for fault detection and isolation," in *Proceedings of the 16th International Conference on Control Systems and Computer Science*, pp. 323–330, Bucharest, Romania, May 2007.
- [30] Y. Xiong and M. Saif, "Unknown disturbance inputs estimation based on a state function observer design," *Automatica*, vol. 39, no. 8, pp. 1389–1398, 2003.
- [31] Z. Yuan, G. C. Vansteenkiste, and C. Wen, "Improving the observer-based FDI design for efficient fault isolation," *International Journal of Control*, vol. 68, no. 1, pp. 197–218, 1997.
- [32] P. S. Teh and H. Trinh, "Design of unknown input functional observers for nonlinear systems with application to fault diagnosis," *Journal of Process Control*, vol. 23, no. 8, pp. 1169–1184, 2013.
- [33] H. Trinh, T. Fernando, K. Emami, and D. C. Huong, "Fault detection of dynamical systems using first-order functional observers," in *Proceedings of the 8th IEEE International Conference on Industrial and Information Systems (ICIIS '13)*, pp. 197–200, IEEE, Peradeniya, Sri Lanka, December 2013.
- [34] D. C. Huong, H. Trinh, H. M. Tran, and T. Fernando, "Approach to fault detection of time-delay systems using functional observers," *Electronics Letters*, vol. 50, no. 16, pp. 1132–1134, 2014.
- [35] K. Emami, T. Fernando, B. Nener, H. Trinh, and Y. Zhang, "A functional observer based fault detection technique for dynamical systems," *Journal of the Franklin Institute*, vol. 352, no. 5, pp. 2113–2128, 2015.
- [36] M. Darouach, "Existence and design of functional observers for linear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 940–943, 2000.
- [37] H. Trinh and T. Fernando, "On the existence and design of functional observers for linear systems," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '07)*, pp. 1974–1979, Harbin, China, August 2007.
- [38] H. Trinh and T. Fernando, *Functional Observers for Dynamical Systems*, Springer, Berlin, Germany, 2012.
- [39] F. Rotella and I. Zambettakis, "On functional observers for linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 5, pp. 1354–1360, 2013.
- [40] T. Fernando and H. Trinh, "A system decomposition approach to the design of functional observers," *International Journal of Control*, vol. 87, no. 9, pp. 1846–1860, 2014.
- [41] T. L. Fernando, H. M. Trinh, and L. Jennings, "Functional observability and the design of minimum order linear functional observers," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1268–1273, 2010.
- [42] G.-R. Duan, "Solutions of the equation  $AV + BW = VF$  and their application to eigenstructure assignment in linear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 276–280, 1993.

- [43] J. Su and W. H. Chen, "Fault diagnosis for vehicle lateral dynamics with robust threshold," in *Proceedings of the IEEE International Conference on Industrial Technology*, Taipei, Taiwan, March 2016.
- [44] A. Termehchy, A. Afshar, and M. Javidsharifi, "A novel design of unknown input observer for fault diagnosis in the Tennessee-Eastman process system to solve non-minimum phase problem," in *Proceedings of the IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA '13)*, pp. 1–6, Kuala Lumpur, Malaysia, November 2013.

## Research Article

# An Effective Strategy to Build Up a Balanced Test Suite for Spectrum-Based Fault Localization

Ning Li,<sup>1</sup> Rui Wang,<sup>2</sup> Yu-li Tian,<sup>1</sup> and Wei Zheng<sup>3</sup>

<sup>1</sup>*School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China*

<sup>2</sup>*Beijing Institute of Information and Control, Beijing 100048, China*

<sup>3</sup>*School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Ning Li; [lining@nwpu.edu.cn](mailto:lining@nwpu.edu.cn)

Received 25 December 2015; Revised 7 March 2016; Accepted 23 March 2016

Academic Editor: Wen Chen

Copyright © 2016 Ning Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

During past decades, many automated software faults diagnosis techniques including Spectrum-Based Fault Localization (SBFL) have been proposed to improve the efficiency of software debugging activity. In the field of SBFL, suspiciousness calculation is closely related to the number of failed and passed test cases. Studies have shown that the ratio of the number of failed and passed test case has more significant impact on the accuracy of SBFL than the total number of test cases, and a balanced test suite is more beneficial to improving the accuracy of SBFL. Based on theoretical analysis, we proposed an PNF (Passed test cases, Not execute Faulty statement) strategy to reduce test suite and build up a more balanced one for SBFL, which can be used in regression testing. We evaluated the strategy making experiments using the Siemens program and Space program. Experiments indicated that our PNF strategy can be used to construct a new test suite effectively. Compared with the original test suite, the new one has smaller size (average 90% test case was reduced in experiments) and more balanced ratio of failed test cases to passed test cases, while it has the same statement coverage and fault localization accuracy.

## 1. Introduction

Software fault localization is an activity of identifying the exact locations of program faults. It is one of the most tedious and time-consuming tasks in program debugging [1]. To decrease the cost of software debugging, many automatic software fault localization techniques have been proposed in recent years [2–8]. Among those, Spectrum-Based Fault Localization (SBFL) techniques have attracted a lot of attention, such as Tarantula [9] and Ochiai [10]. SBFL techniques are simple and highly efficient, and they usually calculate the suspiciousness of program entities according to different dynamic behaviours of failed and passed test executions.

In the field of SBFL, more than 30 kinds of formulas have been proposed to calculate suspiciousness of program entity [11]. And it is essential to count the number of failed and passed test cases in each SBFL technique. Due to that, we can speculate that the ratio of failed test cases to passed test cases might affect the accuracy of software fault location. In this paper, we call the ratio of the number of failed test cases to

the number of passed test cases as class ratio. Test cases reduction is an important topic in respect of the test case number. Some studies [12–15] focused on test cases reduction technologies for SBFL, but they primarily reduced the total number of test cases, without considering class ratio.

Only a few studies focused on the impact of class ratio on the accuracy of software fault localization [16–18]. The experiments in Gong et al.'s work [16] showed that the class imbalance phenomenon of test suites would negatively affect the efficiency of SBFL. They used two methods to change the class ratio of failed to passed test cases. (1) The first method fixed the total size of test suites and changed the ratio of failed to passed test cases. (2) The second method fixed the number of failed cases and changed the number of passed test cases. They selected some test cases from an original test suite randomly when generating a new test suite. Base on the work, Gao et al. [17] conducted a theoretical study to generate balanced test suite. They cloned the failed test cases for suitable number of times to catch up with the number of

passed test cases. Their theoretical analysis result suggested that the efficiency of SBFL can be improved under certain conditions and impaired at no time by using their strategy.

In this paper, we proposed a nonrandom PNF (Passed test cases, Not execute Faulty statement) test case selection strategy to build up a reduced and balanced test suite. This strategy is applicable for some regression testing. For example, part of test cases should be selected from the original test suite when software platform is updated. Different from cloning failed test cases in literature [17], we prefer to select passed test cases from the original test suite to construct a new balanced test suite. This strategy will not only make the test suite more balanced than before, but also significantly reduce the size of the original test suite without decreasing statement coverage.

This paper is structured as follows. In Section 2, we analyse the PNF strategy from a theoretical perspective. Then the experiments on Siemens and Space are presented in Section 3. Finally Section 4 concludes the paper and outlines our further work.

## 2. Theoretical Analysis

In this section, we conduct a theoretical analysis about the impact of increasing passed test cases on the accuracy of SBFL. Here, we take two typical SBFL techniques Tarantula and Nashi2 as representative. Based on the analysis result, we propose PNF strategy to build up a balanced test suite, which can not only reduce the size of test suite but also hold the accuracy of SBFL like the original test suite.

*2.1. PNF Strategy.* In order to describe our PNF strategy, we define the following symbols at first:

- (i)  $T_{\text{orig}}$ : the original test suite.
- (ii)  $T_{\text{init}}$ : the initial test suite which consists of some test cases selected from  $T_{\text{orig}}$ .
- (iii)  $T_{\text{new}}$ : the increased test suite in which some test cases are added based on  $T_{\text{init}}$ .
- (iv)  $\text{Susp}_s$ : suspiciousness of the statement  $s$ .
- (v)  $P_{\text{orig}}$ : the number of all passed test cases in  $T_{\text{orig}}$ .
- (vi)  $F_{\text{orig}}$ : the number of all failed test cases in  $T_{\text{orig}}$ .
- (vii)  $P$ : the number of all passed test cases in  $T_{\text{init}}$ .
- (viii)  $F$ : the number of all failed test cases in  $T_{\text{init}}$ .
- (ix)  $N_p^s$ : the number of passed test cases which execute the statement  $s$  in  $T_{\text{init}}$ .
- (x)  $N_f^s$ : the number of failed test cases which execute the statement  $s$  in  $T_{\text{init}}$ .
- (xi)  $N_p^{s'}$ : the number of passed test cases which execute the statement  $s$  in  $T_{\text{new}}$ .
- (xii)  $N_f^{s'}$ : the number of failed test cases which execute the statement  $s$  in  $T_{\text{new}}$ .

In the fault localization report of SBFL, all statements are often ranked by their suspiciousness in descending order. A smaller Rank indicates a higher likelihood of being faulty

statement. Let statement  $i$  represent the faulty statement and let  $j$  represent any of the nonfaulty statements. Suppose  $\text{Rank}_i > \text{Rank}_j$  in the original test suite; if the inequality of  $\text{Rank}_i > \text{Rank}_j$  is not changed after modifying the class ratio of failed to passed test cases, we regard it as a positive change strategy for modifying the class ratio.

In this paper, we use the PNF strategy to modify class ratio and construct the new test suite. In PNF strategy, we hold the failed test cases unchanged and then change the number of passed test cases. The detailed steps are listed as follows:

- (1) Build up an initial test suite  $T_{\text{init}}$ . Copy all failed test cases from  $T_{\text{orig}}$  and select part of passed test cases from  $T_{\text{orig}}$ . Here, the number of selected passed test cases is equal to  $F_{\text{orig}}$ . It means that the class ratio is 1:1 (failed:passed).
- (2) Build up a new test suite  $T_{\text{new}}$  in which the class ratio is 1:K (failed:passed). Here, the number of passed test cases is  $K$  times of  $P$ . Moreover, the increased test cases do not execute the faulty statement  $i$ .
- (3) Take the coverage information into consideration. We also calculate the statement coverage because of the importance of the coverage criterion in software testing. We give the priority to the passed test cases which contribute to the statement coverage when selecting a new test case. If the statement coverage of  $T_{\text{new}}$  is lower than that of  $T_{\text{orig}}$ , we add some additional passed test cases to ensure this goal (the statement coverage of  $T_{\text{new}}$  is equal to that of  $T_{\text{orig}}$ ).

Algorithm 1 of PNF is used to select passed test cases and build up a balanced test suite.

To do a detailed theoretical analysis, we describe some common points when applying this strategy to increase passed test cases:

- (1) Since we do not increase any failed test cases, we get  $N_f^i = N_f^i = F$ ,  $N_f^j = N_f^j$ .
- (2)  $N_p^j$  denotes the number of passed test cases that executed the statement  $j$ , and its value range is  $[N_p^j, N_p^j + (K - 1) \times P]$ , where  $K > 1$ .
- (3) Since the increased passed test cases do not execute the faulty statement  $i$ , we get  $N_p^i = N_p^i$ .
- (4) Because all failed test cases definitely execute the faulty statement  $i$ ,  $N_f^i = F = F_{\text{orig}}$ .

In the next sections, we do the detailed theoretical analysis by taking typical Tarantula and Nashi2 as examples.

*2.2. Theoretical Analysis in Tarantula.* The suspiciousness formula of Tarantula is

$$\text{Susp}_s = \frac{N_f^s/F}{N_f^s/F + N_p^s/P}. \quad (1)$$

We try to calculate the object equation:

$$\text{Susp}_{i'} - \text{Susp}_{j'}, \quad (2)$$

```

Input: initial passed testcases: oldPassedList,
         initial failed testcases: oldFailedList,
         class ratio: ratio
Output: new passed testcases: newPassedList
// (1) find passed test cases which do not
      execute faulty statements
for each tID (passed test cases ID) in oldPassedList do
  stmtSet: get executed statements set of tID;
  if stmtSet not contain faulty lines then
    add (tID, sizeofstmtSet) into nonFaultMap;
  end
if nonFaultMap is empty then
  add all tID  $\in$  oldPassedList into nonFaultMap;
// (2) find passed test cases which execute as
      many non-faulty statements as possible
sort nonFaultMap by its value (size of stmtSet);
validPassList = nonFaultMap.Key;
// (3) build up a new test cases
newPasscnt = ratio * oldFailedList.size();
newPassedList = new ArrayList(newPasscnt);
// statement coverage
smSet = get the set of all executed statements;;
while newPassedList.size < newPasscnt do
  ti = a new test case of validPassList;
  smSetTi = get executed statements by ti;
  if one element in smSetTi  $\notin$  smSet then
    add ti to newPassedList;
    add smSetTi to smSet;
  end
if statement coverage of  $T_{\text{new}}$  < that of  $T_{\text{orig}}$  then
  // improve statement coverage
  find out the valuable passed test cases;
  add valuable passed test cases to newPassedList;
// make sure of the required class ratio
while newPassedList.size < newPasscnt do
  search unselected test case from validPassList;
  add it to newPassedList;
end
return newPassedList;

```

ALGORITHM 1: Selected best passed test cases (PNF).

where  $Susp_{i_i}$  and  $Susp_{i_j}$  denote the suspiciousness of statement  $i$  and statement  $j$  after increasing passed test cases. We discuss three cases based on the relation of the suspiciousness of the faulty statement and nonfaulty statement in the initial test suite.

According to (1) and  $N_f^i = F$ , (2) can be expressed as follows:

$$\begin{aligned} &\Rightarrow \frac{1}{1 + N_p^i/KP} - \frac{N_f^j/F}{N_f^j/F + N_p^j/KP} \\ &\Rightarrow \frac{N_f^j}{F} + \frac{N_p^j}{KP} - \frac{N_f^j}{F} - \frac{N_p^j}{F} \times \frac{N_p^i}{KP} \end{aligned}$$

$$\begin{aligned} &\Rightarrow \frac{N_p^j}{KP} - \frac{N_p^j}{F} \times \frac{N_p^i}{KP} \\ &\Rightarrow \frac{1}{KP} \times \left( N_p^j - \frac{N_p^j \times N_p^i}{F} \right). \end{aligned} \quad (3)$$

Here, we proof that PNF strategy is positive from three cases, respectively:  $Susp_i > Susp_j$ ,  $Susp_i = Susp_j$ , and  $Susp_i < Susp_j$ .

*Case 1* ( $Susp_i > Susp_j$ ). Because the suspiciousness of statement  $i$  is greater than the suspiciousness of statement  $j$ , then we can express it as follows:

$$T = \text{Tarantula}(i) - \text{Tarantula}(j) > 0. \quad (4)$$

That is,

$$\frac{N_f^i/F}{N_f^i/F + N_p^i/P} - \frac{N_f^j/F}{N_f^j/F + N_p^j/P} > 0. \quad (5)$$

Since  $N_f^i = F$ , (5) can be simplified as follows:

$$\begin{aligned} &\Rightarrow \frac{1}{1 + N_p^i/P} - \frac{N_f^j/F}{N_f^j/F + N_p^j/P} > 0 \\ &\Rightarrow \frac{N_p^j}{P} - \frac{N_f^j}{F} \times \frac{N_p^i}{P} > 0 \\ &\Rightarrow F \times N_p^j - N_f^j \times N_p^i > 0 \\ &\Rightarrow N_p^i < \frac{N_p^j \times F}{N_f^j}. \end{aligned} \quad (6)$$

To ensure that our strategy is positive, we need to have  $Susp_{i_i} - Susp_{i_j} > 0$ .

According to previous calculation, we know

$$\begin{aligned} &Susp_{i_i} - Susp_{i_j} > 0 \Rightarrow \\ &\frac{1}{KP} \times \left( N_p^j - \frac{N_p^j \times N_p^i}{F} \right) > 0 \\ &\quad \because \frac{1}{KP} > 0, \\ &\therefore N_p^j - \frac{N_p^j \times N_p^i}{F} > 0 \Rightarrow \\ &N_p^i - \frac{N_p^j \times F}{N_f^j} < 0. \end{aligned} \quad (7)$$

According to the above analysis, this problem can be simplified to the following proof:

The following is given:  $N_p^i < (N_p^j \times F)/N_f^j$ .

The following is proved:  $N_p^{i,j} < (N_f^j \times N_p^i)/F$ .

*Proof.* For the faulty statement  $i$ , because the increased passed test cases do not execute this statement,  $N_p^{i,i} = N_p^i$ . For any statement  $j$  ( $j \neq i$ ), because we increased  $(K - 1) \times P$  passed test cases, the value range of  $N_p^{i,j}$  is

$$[N_p^j, N_p^j + (K - 1) \times P], \quad (8)$$

where  $N_p^{i,j}$  denotes the number of passed test cases which executed the statement  $j$ . Consider

$$\begin{aligned} \because N_p^i &< \frac{N_p^j \times F}{N_f^j} \\ \therefore N_p^i &< \frac{[N_p^j, N_p^j + (k - 1) \times P] \times F}{N_f^j} \\ &= \frac{N_p^{i,j} \times F}{N_f^j} \implies \end{aligned} \quad (9)$$

$$N_p^i - \frac{N_p^{i,j} \times F}{N_f^j} < 0,$$

$$\therefore N_p^i = N_p^i \implies$$

$$N_p^i - \frac{N_p^{i,j} \times F}{N_f^j} < 0 \implies$$

$$\text{Susp}_{i,i} > \text{Susp}_{i,j}.$$

□

Based on the proof, if  $\text{Susp}_i > \text{Susp}_j$ , we can get  $\text{Susp}_{i,i} > \text{Susp}_{i,j}$  when we use the strategy to select passed test cases. That is to say, for the faulty statement  $i$  whose suspiciousness is higher than the suspiciousness of the statement  $j$  before increasing passed test cases, its suspiciousness is still higher than the suspiciousness of statement  $j$  after increasing passed test cases. It shows that the rank of the faulty statement  $i$  will not decrease. Therefore, the strategy of increasing passed test cases is a positive approach to select passed test cases in this condition.

*Case 2* ( $\text{Susp}_i = \text{Susp}_j$ ). Similarly with the first case, when we increase passed test cases to  $K$  ( $K > 1$ ) times than before, the value range  $N_p^{i,j}$  of is  $[N_p^j, N_p^j + (K - 1) \times P]$ :

$$(i) \text{ If } N_p^{i,j} = N_p^j,$$

$$\therefore \text{Susp}_i = \text{Susp}_j \implies$$

$$N_p^i = \frac{N_p^j \times F}{N_f^j} \quad (\because N_p^i = N_p^i) \implies$$

$$N_p^i = \frac{N_p^{i,j} \times F}{N_f^j} \implies$$

$$\text{Susp}_{i,i} = \text{Susp}_{i,j}.$$

(10)

$$(ii) \text{ If } N_p^{i,j} > N_p^j,$$

$$\therefore \text{Susp}_i > \text{Susp}_j \implies$$

$$N_p^i < \frac{N_p^j \times F}{N_f^j} \quad (\because N_p^i = N_p^i) \implies$$

(11)

$$N_p^i < \frac{N_p^{i,j} \times F}{N_f^j} \implies$$

$$\text{Susp}_{i,i} > \text{Susp}_{i,j}.$$

Based on the above analysis, if  $\text{Susp}_i = \text{Susp}_j$ , we can get  $\text{Susp}_{i,i} \geq \text{Susp}_{i,j}$  when we use the strategy to select passed test cases. It implies that the rank of the faulty statement  $i$  will not decrease while it maybe increases. When we increase passed test cases, if we select passed test cases which execute as many nonfaulty statements ( $N_p^{i,j} > N_p^j$ ) as possible, it will enhance the rank of the faulty statement. The strategy of increasing passed test cases in this way will be a positive approach to enlarge test suite.

*Case 3* ( $\text{Susp}_i < \text{Susp}_j$ ). The object equation is the same as previous analysis:

$$\text{Susp}_{i,i} - \text{Susp}_{i,j} = \frac{1}{KP} \times \left( N_p^{i,j} - \frac{N_p^{i,j} \times N_p^i}{F} \right). \quad (12)$$

Since  $1/KP > 0$ ,  $F > 0$ , we only focus on the numerator:

$$\implies N_p^{i,j} \times F - N_p^i \times N_f^j$$

$$\because N_p^i = N_p^i \quad (13)$$

$$\implies N_p^{i,j} \times F - N_p^i \times N_f^j.$$

When  $N_p^i > (N_p^j \times F)/N_f^j$  ( $\text{Susp}_i < \text{Susp}_j$ ), because  $N_p^{i,j} \geq N_p^j$ , the result of  $(N_p^i \times N_f^j - N_p^{i,j} \times F)$  could be one of the three cases:  $> 0$ ,  $= 0$ , and  $< 0$ . But as we can know, the following condition can effectively reduce the negative effects:

$$N_p^i < \frac{N_p^{i,j} \times F}{N_f^j} \quad (\text{Susp}_{i,i} > \text{Susp}_{i,j}). \quad (14)$$

Therefore, we should make the value of  $N_p^{i,j}$  as large as possible; namely, we should select those passed test cases which execute as many nonfaulty statements as possible.

2.3. *Theoretical Analysis in Nashi2.* The suspiciousness formula of Nashi2 is

$$\text{Susp}_s = N_f^s - \frac{N_p^s}{P+1}. \quad (15)$$

The symbols in (15) are the same with the formula of Tarantula. We use the similar analysis process with Tarantula. Therefore, we still try to calculate the difference between  $\text{Susp}_i$  of the faulty statement and  $\text{Susp}_j$  of the statement and expect that the rank of faulty statement in the report of SBFL is not impaired. The difference between  $\text{Susp}_{i_i}$  and  $\text{Susp}_{i_j}$  can be expressed as follows:

$$N_f^i - \frac{N_p^i}{KP+1} - \left( N_f^j - \frac{N_p^j}{KP+1} \right). \quad (16)$$

We discuss three cases based on the relation of  $\text{Susp}_i$  (suspiciousness of the faulty statement) and  $\text{Susp}_j$  (suspiciousness of the statement) in the original test suite as Section 2.2.

*Case 1* ( $\text{Susp}_i > \text{Susp}_j$ ). According to  $\text{Susp}_i > \text{Susp}_j$ , we have

$$\begin{aligned} N_f^i - \frac{N_p^i}{P+1} &> \left( N_f^j - \frac{N_p^j}{P+1} \right) \implies \\ F - \frac{N_p^i}{P+1} &> \left( N_f^j - \frac{N_p^j}{P+1} \right) \implies \\ N_p^i &< (F - N_f^j)(P+1) + N_p^j. \end{aligned} \quad (17)$$

In order to ensure  $\text{Susp}_{i_i} > \text{Susp}_{i_j}$ , the object equation of (16) can be transformed into the following:

$$\begin{aligned} F - \frac{N_p^i}{KP+1} - \left( N_f^j - \frac{N_p^j}{KP+1} \right) &> 0 \implies \\ N_p^i &< (F - N_f^j)(KP+1) + N_p^j \implies \\ N_p^i &< (F - N_f^j)(KP+1) + N_p^j. \end{aligned} \quad (18)$$

According to the above analysis, this problem can be simplified to the following proof:

The following is given:  $N_p^i < (F - N_f^j)(P+1) + N_p^j$ .

The following is proved:  $N_p^i < (F - N_f^j)(KP+1) + N_p^j$ .

*Proof.* Consider the following:

$$\because K > 1$$

$$\therefore KP + 1 > P + 1$$

$$\therefore N_p^i < (F - N_f^j)(P+1) + N_p^j \implies$$

$$N_p^i < (F - N_f^j)(P+1) + N_p^j \implies$$

$$N_p^i < (F - N_f^j)(KP+1) + N_p^j$$

$$\because N_p^j > N_p^j \implies$$

$$N_p^i < (F - N_f^j)(KP+1) + N_p^j.$$

(19)

□

This proof shows that the suspiciousness of the faulty statement  $i$  is still higher than the suspiciousness of statement  $j$  after passed test cases are increased. It means that the rank of the faulty statement does not reduce.

*Case 2* ( $\text{Susp}_i = \text{Susp}_j$ ). According to this condition, we have

$$N_p^i = (F - N_f^j)(P+1) + N_p^j. \quad (20)$$

In order to ensure  $\text{Susp}_{i_i} \geq \text{Susp}_{i_j}$ , the object equation of (16) can be transformed into the following:

$$F - \frac{N_p^i}{KP+1} - \left( N_f^j - \frac{N_p^j}{KP+1} \right) \geq 0. \quad (21)$$

*Proof.* Because  $KP + 1 > 0$ , we can only focus on the numerator and simplify it into

$$\begin{aligned} &\implies (F - N_f^j)(KP+1) + N_p^j - N_p^i \\ &\implies (F - N_f^j)(KP+1) + N_p^j - N_p^i \\ \because N_p^i &= (F - N_f^j)(P+1) + N_p^j \\ &\implies (F - N_f^j)(KP+1) + N_p^j - [(F - N_f^j)(P+1) + N_p^j] \\ &\implies P(K-1)(F - N_f^j) + (N_p^j - N_p^i) \\ \because K > 1, P > 0, F - N_f^j &\geq 0, N_p^j - N_p^i \geq 0 \\ \therefore P(K-1)(F - N_f^j) &+ (N_p^j - N_p^i) \geq 0 \\ &\implies F - \frac{N_p^i}{KP+1} - \left( N_f^j - \frac{N_p^j}{KP+1} \right) \geq 0. \end{aligned} \quad (22)$$

□

The above inequality implies that the suspiciousness of the faulty statement  $i$  must be equal to or higher than the suspiciousness of statement  $j$  after passed test cases are increased. Moreover, we should make the value of  $N_p^j$  as large as possible, and it means that we should select the passed test cases which execute as many nonfaulty statements as possible.

*Case 3* ( $\text{Susp}_i < \text{Susp}_j$ ). According to this condition, we have

$$N_p^i > (F - N_f^j)(P+1) + N_p^j. \quad (23)$$

We need to calculate the relation between  $N_p^{hi}$  and  $(F - N_f^j)(KP + 1) + N_p^{lj}$ .

Because  $(F - N_f^j)(KP + 1) > (F - N_f^j)(P + 1)$  and  $N_p^{lj} > N_p^j$ , the relation between  $N_p^{hi}$  and  $(F - N_f^j)(KP + 1) + N_p^{lj}$  can be one of the three cases: higher, equal, and lower.

When  $N_p^{hi} < (F - N_f^j)(KP + 1) + N_p^{lj}$ , it shows  $Susp_{ij} > Susp_{ij}$ . This case can effectively reduce the negative effects on the rank of the faulty statement by increasing passed test cases. Therefore, we should make the value of  $N_p^{lj}$  as large as possible; namely, we should select those passed test cases which execute as many nonfaulty statements as possible.

**2.4. Summary of Strategy.** According to the analysis and proof about Tarantula and Nashi2 in Sections 2.2 and 2.3, we draw a conclusion that the accuracy of SBFL would not be affected by the class ratio of failed to passed test cases when we change the class ratio using a nonrandom PNF strategy. Here, PNF strategy means holding the number of failed test cases unchanged and modify the class ratio by increasing passed test cases. When increasing passed test cases, the selected passed test cases are expected not to execute the faulty statement and to execute as many nonfaulty statements as possible. According to the analysis, we can build up a balanced test suite with the PNF strategy. The new test suite has the following advantages:

- (1) A smaller size than the original test suite.
- (2) A more balanced ratio of failed to passed test cases than the original one.
- (3) Keeping the same statement coverage as the original test suite.
- (4) At least keeping the same fault localization accuracy of SBFL as the original test suite.

Despite the knowledge about the location of a fault in PNF strategy, it can be replaced by other information, such as the top  $N$  suspicious statements according to suspiciousness calculation. However, in order to get more exact result, the knowledge about the location of a fault is required in this paper. This PNF strategy is not applicable to the regression testing in which target program has to be modified. However, it can be used to construct a new test suite for the regression testing in which target program is unchanged. For example, regression testing is often required for all developed products in a company when OS is updated, package is applied, or platform is changed.

### 3. Experiment

Because the effectiveness of our PNF strategy has been analysed and proven from a theoretical perspective in Section 2, we took two small programs (tcas and totinfo) from Siemens program suite and one large program (Space) as samples to verify our strategy from an experimental perspective in this section.

TABLE 1: Information of three target programs.

Program	LOC	Faulty versions	Total test cases
tcas	174	41	1608
totinfo	407	23	1052
Space	9127	38	13585

**3.1. Experiment Setup.** Siemens and Space (<http://sir.unl.edu/portal/index.php>) dataset have been widely used in the research work of fault localization, and the original information of the three target programs is listed in Table 1. All the programs are written in C. The average fault localization accuracy of all available versions for a target program is the final accuracy of the program.

There are several faults in some faulty versions of tcas and Space, and we call these versions multiple faults versions. In experiment, we simplified the interactions and interferences between multiple faults presented in [19] and took one multiple faults version as several single fault versions. For these multiple versions, we supposed that only the most suspicious fault could be localized in every iteration, then fixed it, and entered it into the next iteration to find out another fault. Although this way is not very efficient, it is similar to the fault localization process of real software testing in a certain degree. In addition, we used 20 of 38 versions of Space program in our experiment and excluded other versions in which there were compile errors or no failed test cases.

Symbol  $R$  refers to different test case class ratio and symbol  $V$  denotes different program versions. In experiment, we generated 7 different class ratios for each program suite, as  $R_1, R_2, R_3, R_4, R_5, R_6, R_7 = 1, 2, 4, 8, 16, 32, 64$ , where  $R_7 = 64$  means that the number of passed cases is 64 times the number of failed cases in test suite. Symbol  $E = \langle R_i, V_j \rangle$  ( $i = 1, \dots, m; j = 1, \dots, k$ ) refers to one test, and it means experiment on version  $V_j$  with test case class ratio  $R_i$ .

To evaluate the influence of class ratio in fault localization activity, we performed 4 basic fault localization techniques, Tarantula, Nashi2, Jaccard, and Ochiai, on each faulty program version with different class ratio. Because of the time and space constraints, our case studies did not use other advanced fault localization techniques, such as RBF, DStar, and others proposed in [4, 6, 20–22].

Since failed test cases are more contributive to fault localization, we generated test case suites by remaining all the failed cases and increasing passed cases in accordance with the different class ratio. When we constructed test suites, we used random strategy and nonrandom PNF strategy, respectively. The random strategy selects passed cases randomly from the original passed test cases to generate a new test suite, while the nonrandom PNF strategy selects passed cases according to Algorithm 1.

**3.2. Results Measured by Score.** In this section, we used suspiciousness Score to measure the accuracy of fault localization,

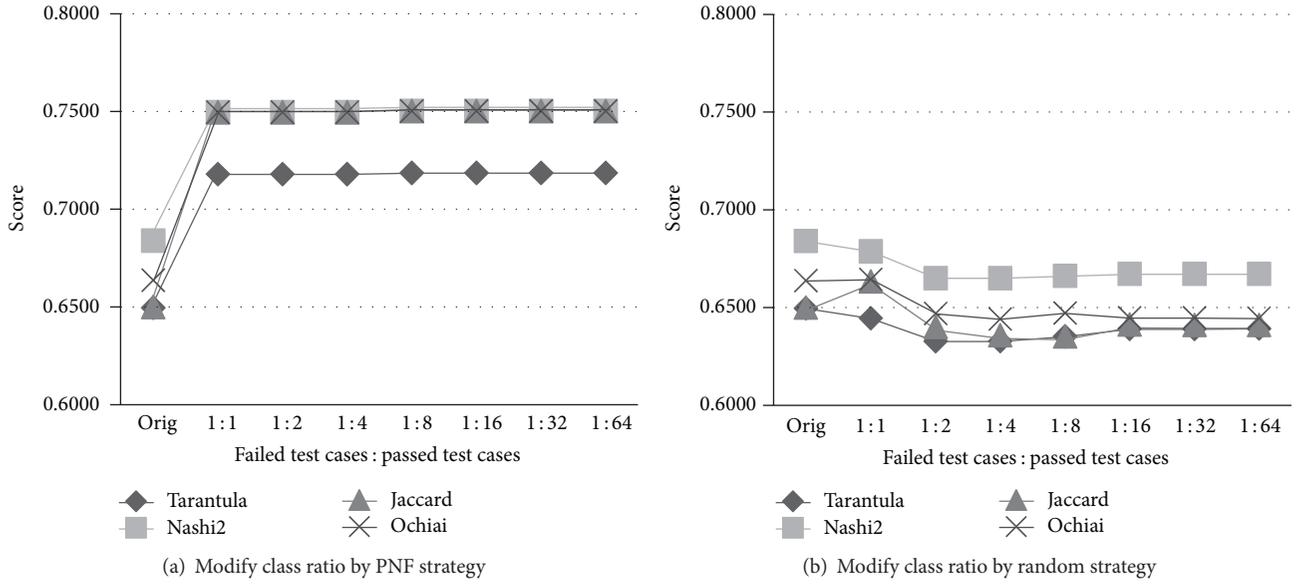


FIGURE 1: Score with different class ratio (tcas).

which has been widely used in software fault localization [1, 2, 23]:

$$\text{Score} = 1 - \frac{\text{Rank}}{\text{Total}} \quad (24)$$

In (24), Total denotes the total number of statements, and Rank denotes the rank of the faulty statement. Rank/Total presents the percentage of code that needs to be examined before the faults are identified. A higher score means higher efficiency of fault localization. When calculating score, we used First-Line strategy to deal with the same suspiciousness situations in which we assigned all statements sharing the same suspiciousness with the first ranking number of them. And we took the average of the score of each faulty version of the same program as the final score of the program. For multiple faults version, we assigned the highest score among several iterations as the score of the version.

Figures 1, 2, and 3 present our experiment results on tcas, totinfo, and Space respectively. In the horizontal axis of these figures, “orig” denotes the original test suite and “1: N” means that the class ratio of failed test cases to passed test cases is 1: N.

The three figures show that class ratio has effect on fault localization performance when using random strategy to generate test suite for all four SBFL techniques. For tcas and totinfo, we can get better accuracy of fault localization with lower class ratio, which implies that a balanced test suite is more efficient for SBFL by selecting passed test cases in a random strategy, which is also consistent with the result of literature [16]. But when we use the nonrandom PNF strategy to enlarge passed test cases, the class ratio can do barely nothing to fault localization performance, which conformed with our theoretical analysis. Consequently, whether the class ratio has the effect on the accuracy of SBFL is closely related

to the strategy for generating the new test cases. From the results, it may be observed that the scores of the new test suites constructed by PNF strategy are higher than the scores of original test suite for the three target programs, while the strong point does not always hold when using random strategy to build up a new test suite.

As shown in Figure 3, there is a small difference between Space and tcas/totinfo. Figure 3(b) about Space program indicates that the four classical SBFL techniques could get better fault localization accuracy with a more balanced test suite using random strategy. It is the same with tcas and totinfo, while the fault localization accuracy becomes better when the distribution of test cases becomes more unbalanced using PNF strategy. It is a different trend compared with tcas and totinfo. But no matter using PNF strategy or random strategy, we can still achieve higher score than the original test suite. The experiment evidence of our PNF strategy is effective for building up a relative balanced test suite. In this figure, 1: 2 is the best class ratio considering both the accuracy of fault localization and the size of test suite.

**3.3. Results Measured by NScore.** This section evaluates our theoretical analysis with NScore measurement conforming with literature [16]. NScore could be calculated by the following equation:

$$\text{NScore} = \frac{N_{\text{detected}}}{N_{\text{total}}} \times 100. \quad (25)$$

In (25),  $N_{\text{detected}}$  is the number of faulty program versions in which the accuracy of fault localization is higher than a threshold  $\theta$  and  $N_{\text{total}}$  is the total number of program versions.

The experimental results of tcas and totinfo using different threshold  $\theta$  (0.8/0.9/0.95) are presented in Figures 4 and 5.

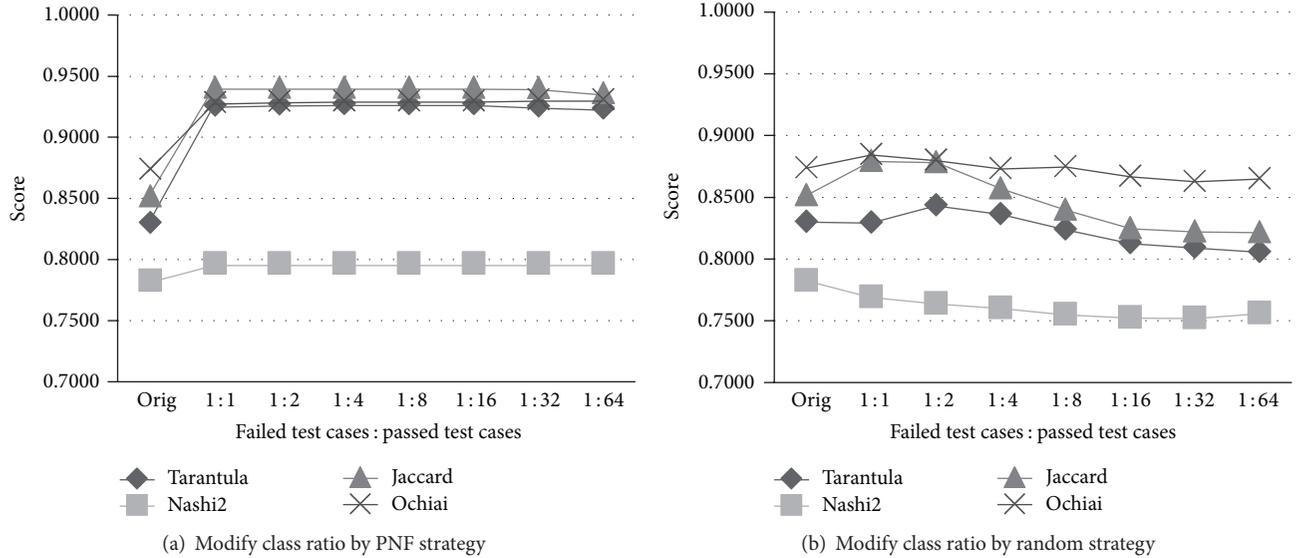


FIGURE 2: Score with different class ratio (totinfo).

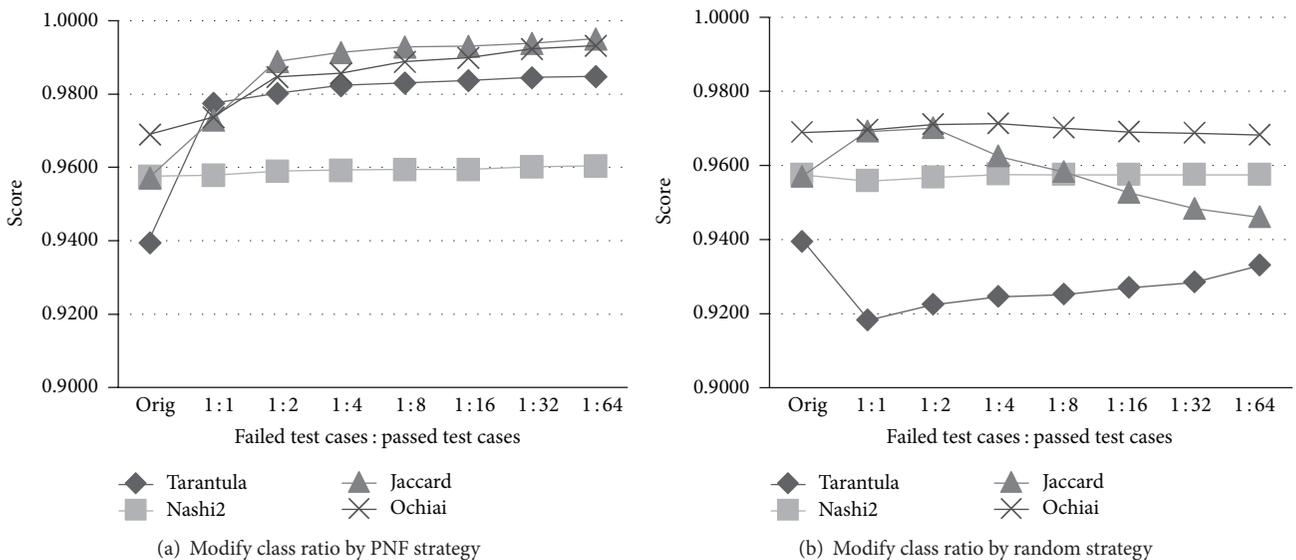


FIGURE 3: Score with different class ratio (Space).

For tcas program, the most left top point in Figure 4(a) means that the scores in 50% of versions are higher than 0.8. Namely, if the threshold  $\theta$  of accuracy is 0.8 and the class ratio is 1:1, then 50% of faults can be localized correctly with Tarantula. The following can be observed: (1) The performance of fault localization is more stable using PNF strategy than random strategy to generate test case; (2) higher fault localization accuracy can be achieved using PNF strategy instead of random strategy. The totinfo program has the same tendency with the tcas.

Figure 6 illustrates the experiment result of Space program. Similar to the results in Section 3.2, there is a small difference between the Siemens and Space program. Figures 6(c) and 6(d) indicate the following: No matter using PNF or random strategy for Nashi2, NScore of Space does not change with the variation of class ratio, while NScore has an increased tendency along with the growth of unbalanced class ratio for Tarantula presented in Figures 6(a) and 6(b). Although a thorough analysis about the reason causing the difference has not been conducted yet, the PNF strategy can reduce

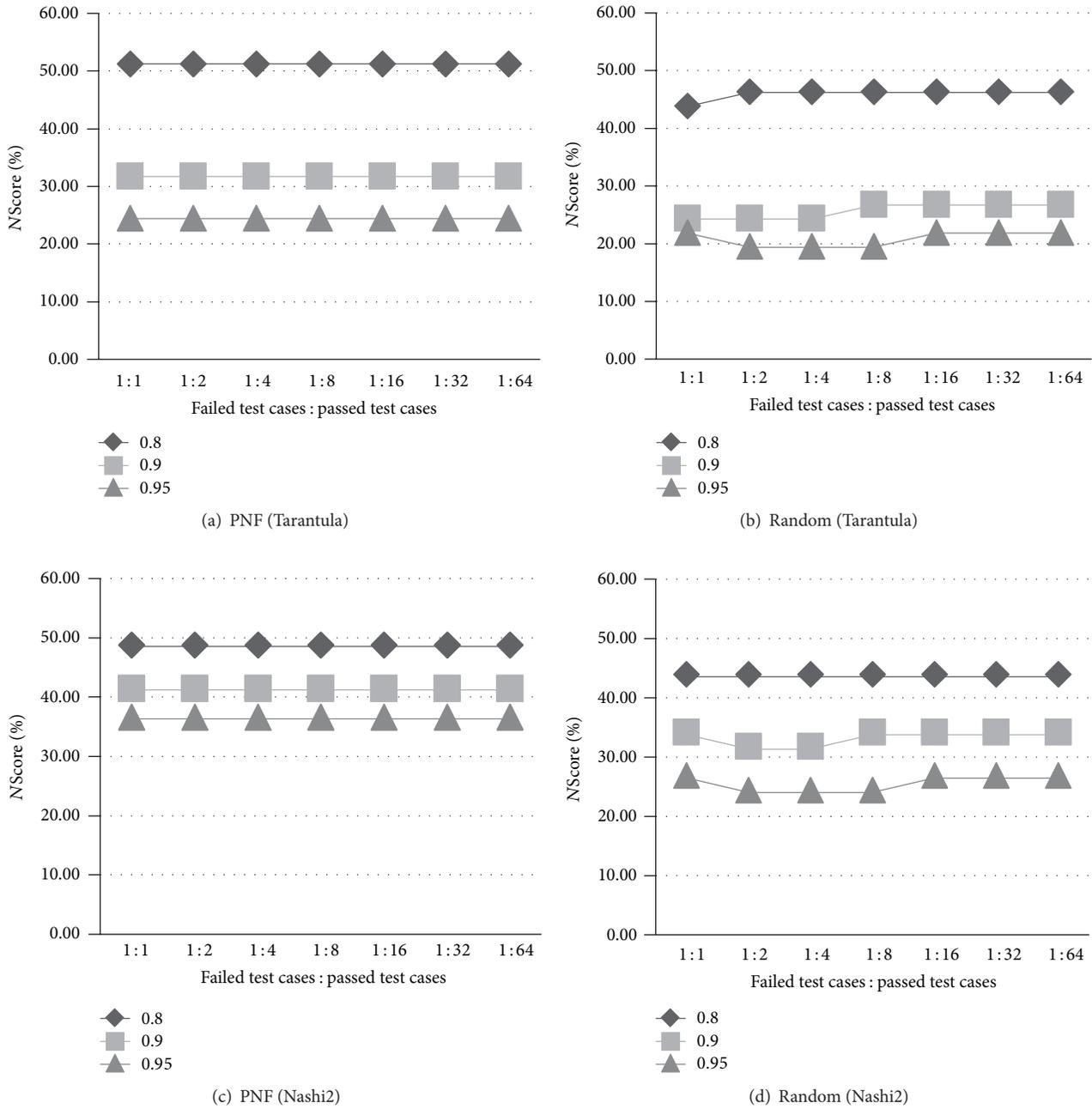


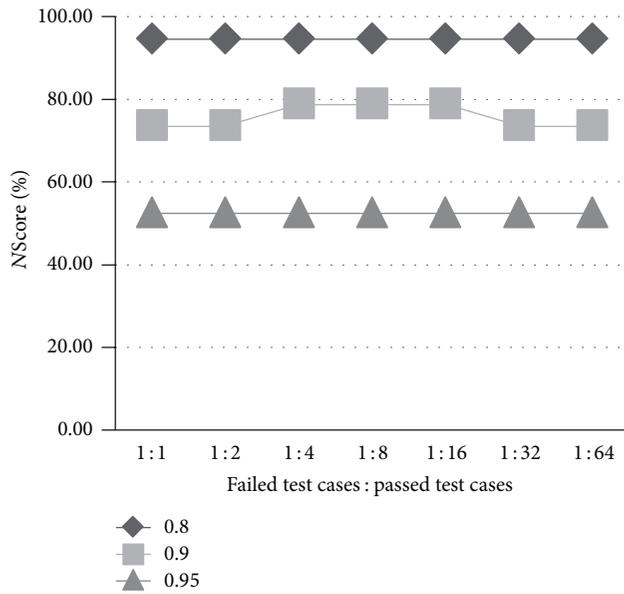
FIGURE 4: NScore with different class ratio (tcas).

the original test suite to a more balanced test suite and get better fault localization accuracy.

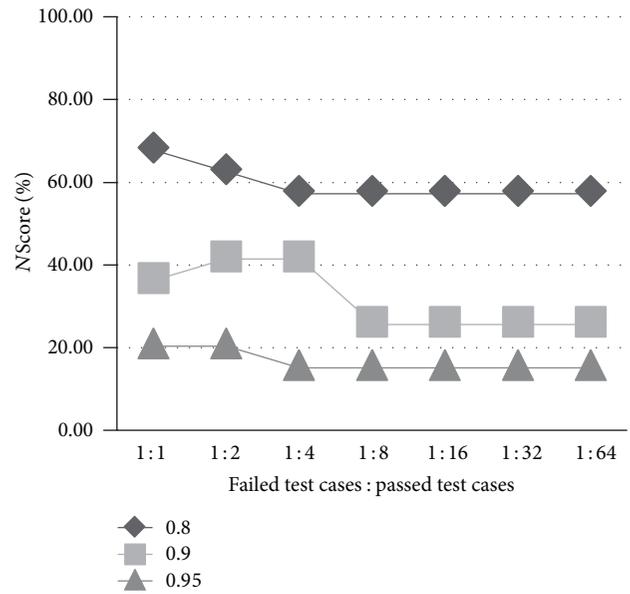
**3.4. Experiment Summary and Discussion.** The experiments using Score and NScore imply that class ratio really has influence on the accuracy of faulty localization. From the experiment results, we concluded that we can build up a balanced test suite with PNF strategy and ensure the accuracy of fault localization with the new balanced test suite. Also, the size of new test suite is smaller than original one. In

experiments, the test case reduction rate was over 95% in tcas program, 90% in totinfo program, and 90% in Space program.

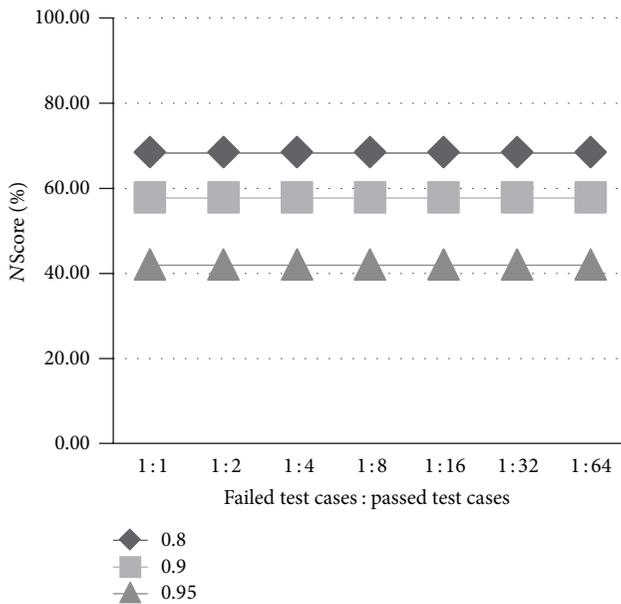
In experiment, we also found that the score of different faulty versions has a significant difference with the same SBFL method, while the results of some version have same tendency. The average score will make us ignore the difference. Table 2 is the score of different tcas versions with several SBFL methods, and it is sorted by Tarantula's score in descending order. From this table, we observed that the maximum score of tcas is 0.9859, but the minimum score is only 0.1268. Why are the scores of some versions so different? Nashi2, Jaccard,



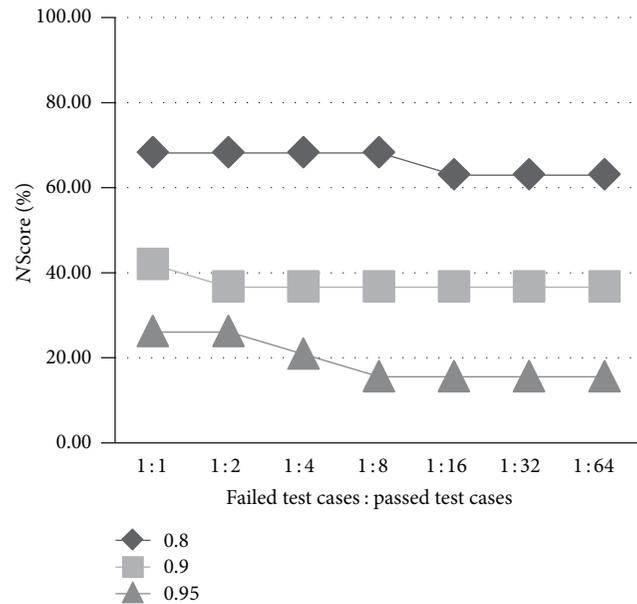
(a) PNF (Tarantula)



(b) Random (Tarantula)



(c) PNF (Nashi2)



(d) Random (Nashi2)

FIGURE 5: NScore with different class ratio (totinfo).

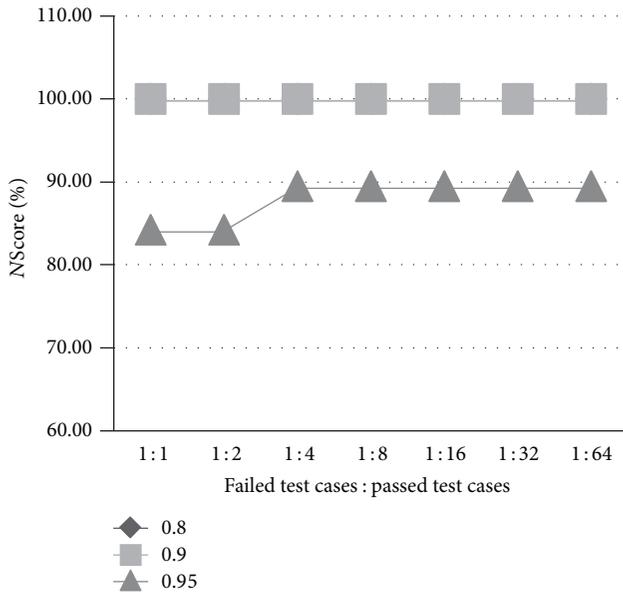
and Ochiai have the same problem. It is worth doing more studies from this perspective.

#### 4. Conclusions

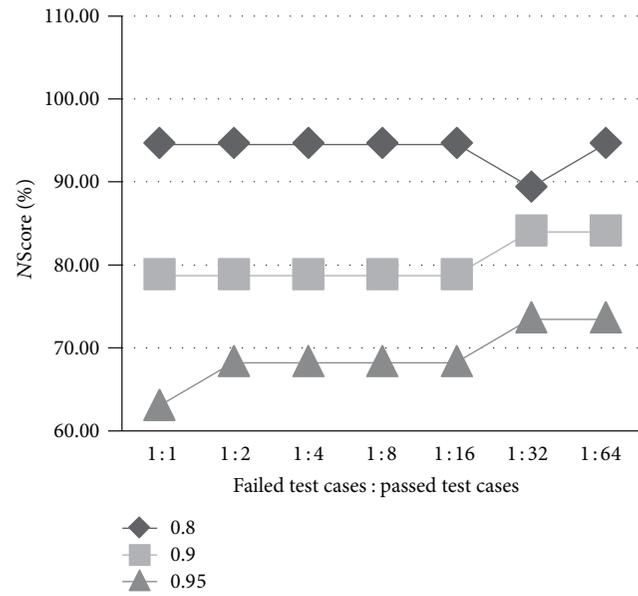
Every suspiciousness calculation of SBFL method is closely related to the number of passed and failed test cases. Previous studies have shown a balanced test suite, which means that the class ratio of the number of failed to passed test cases is similar

and is more beneficial to improve the accuracy of SBFL. In this paper, we proposed a PNF strategy to building up a balanced test suite according to the theoretical analysis and evaluated it by experiments using different SBFL methods.

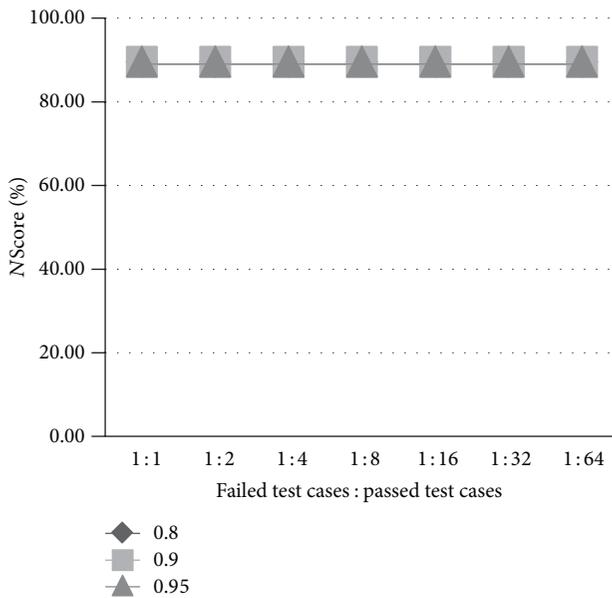
In the PNF strategy, in order to construct a new balanced test suite, we kept the failed test cases unchanged and selected passed test cases from the original test suite according to certain rules: the selected passed test cases should not execute the faulty statement and should execute



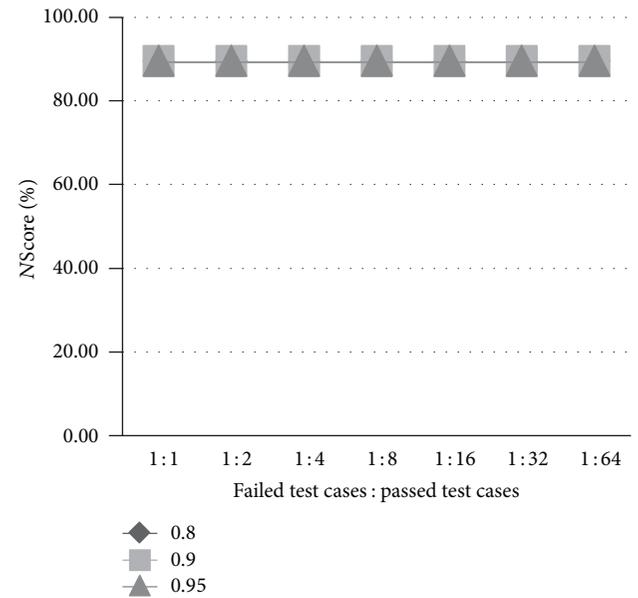
(a) PNF (Tarantula)



(b) Random (Tarantula)



(c) PNF (Nashi2)



(d) Random (Nashi2)

FIGURE 6: NScore with different class ratio (space).

as many nonfaulty statements as possible with the consideration of statement coverage. The experiments indicated the PNF strategy is effective for SBFL. Based on the original test suite, it can generate a new more balanced test suite, which has smaller test suite size (average 90% test cases are reduced), the same accuracy of SBFL, and the same statement coverage.

However, PNF strategy still has some limitations. For example, when there are only few failed test cases in the original test suite, the selected passed test cases by PNF

strategy must be not enough for normal testing. And the process about the multiple faulty program is not enough. The work can be improved in following directions: (1) To reveal the other factors which have the effect on the accuracy of SBFL besides the class ratio; (2) To combine the PNF strategy with test suite reduction techniques. These studies will improve the efficiency of testing and the accuracy of fault localization. In addition, how to build up a balanced test suite without the knowledge about the location of a fault is one of our future work.

TABLE 2: Score of different versions (tcas).

Version	Tarantula	Nashi2	Jaccard	Ochiai
v36	0.9859	0.9859	0.6761	0.9859
v25	0.9718	0.9718	0.5775	0.9718
v39	0.9718	0.9718	0.9577	0.9718
v1	0.9577	0.9577	0.8521	0.9577
v4	0.9577	0.9577	0.2113	0.9577
v37	0.9437	0.9577	0.9859	0.9437
v6	0.8732	0.8732	0.1408	0.8732
v35	0.6761	0.7606	0.2113	0.6901
v28	0.6761	0.7606	0.1408	0.6901
v2	0.6761	0.6901	0.9577	0.6901
v29	0.6620	0.6761	0.6761	0.662
v20	0.5915	0.5915	0.2254	0.5915
v21	0.5915	0.5915	0.5915	0.5915
v22	0.5775	0.5775	0.5915	0.5775
v24	0.5775	0.5775	0.5775	0.5775
v9	0.5775	0.5775	0.1690	0.5775
v14	0.5915	0.5915	0.2113	0.5915
v16	0.2254	0.2254	0.5915	0.2254
v7	0.2254	0.3099	0.8732	0.2254
v12	0.2113	0.3099	0.5775	0.2254
v13	0.2113	0.2676	0.2113	0.2254
v26	0.1831	0.2676	0.9718	0.2113
v8	0.1690	0.169	0.2254	0.169
v5	0.1408	0.3099	0.9577	0.2817
v18	0.1268	0.3099	0.2254	0.2676

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

The work is supported by the National Natural Science Foundation of China (61402370, 61502392), Fundamental Research Funds for the Central Universities (3102015JSJ0004, 3102014JSJ0013), and Aerospace Science and Technology Innovation Foundation of China (2014H03FK011).

## References

- [1] J. A. Jones and M. J. Harrold, "Empirical evaluation of the tarantula automatic fault-localization technique," in *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering (ASE '05)*, pp. 273–282, ACM, November 2005.
- [2] T. Y. Chen, X. Xie, F.-C. Kuo, and B. Xu, "A revisit of a theoretical analysis on spectrum-based fault localization," in *Proceedings of the IEEE 39th Annual Computer Software and Applications Conference (COMPSAC '15)*, vol. 1, pp. 17–22, IEEE, Taichung, Taiwan, July 2015.
- [3] W. E. Wong and V. Debroy, "A survey of software fault localization," Tech. Rep., The University of Texas at Dallas, Richardson, Tex, USA, 2009.
- [4] W. E. Wong, V. Debroy, R. Golden, X. Xu, and B. Thuraisingham, "Effective software fault localization using an RBF neural network," *IEEE Transactions on Reliability*, vol. 61, no. 1, pp. 149–169, 2012.
- [5] W. Wen, "Software fault localization based on program slicing spectrum," in *Proceedings of the 34th International Conference on Software Engineering (ICSE '12)*, pp. 1511–1514, Zurich, Switzerland, June 2012.
- [6] X. Ju, S. Jiang, X. Chen, X. Wang, Y. Zhang, and H. Cao, "HSFal: effective fault localization using hybrid spectrum of full slices and execution slices," *Journal of Systems and Software*, vol. 90, no. 1, pp. 3–17, 2014.
- [7] H. Cheng, D. Lo, Y. Zhou, X. Wang, and X. Yan, "Identifying bug signatures using discriminative graph mining," in *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA '09)*, pp. 141–151, Association for Computing Machinery, Chicago, Ill, USA, July 2009.
- [8] D. Gopinath, R. N. Zaeem, and S. Khurshid, "Improving the effectiveness of spectra-based fault localization using specifications," in *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE '12)*, pp. 40–49, IEEE, September 2012.
- [9] J. A. Jones, M. J. Harrold, and J. Stasko, "Visualization of test information to assist fault localization," in *Proceedings of the 24th International Conference on Software Engineering (ICSE '02)*, pp. 467–477, May 2002.
- [10] R. Abreu, P. Zoetewij, and A. J. C. Van Gemund, "An evaluation of similarity coefficients for software fault localization," in *Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing (PRDC '06)*, pp. 39–46, Riverside, Calif, USA, December 2006.
- [11] T. Y. Chen, X. Xie, F.-C. Kuo, and B. Xu, "A theoretical analysis of the risk evaluation formulas for spectrum-based fault localization," *ACM Transactions on Software Engineering and Methodology*, vol. 22, no. 4, article 40, 2013.
- [12] D. Hao, L. Zhang, Y. Pan, H. Mei, and J. Sun, "On similarity-awareness in testing-based fault localization," *Automated Software Engineering*, vol. 15, no. 2, pp. 207–249, 2008.
- [13] D. Hao, T. Xie, L. Zhang, X. Wang, J. Sun, and H. Mei, "Test input reduction for result inspection to facilitate fault localization," *Automated Software Engineering*, vol. 17, no. 1, pp. 5–31, 2010.
- [14] G. Dandan, W. Tiantian, S. Xiaohong, and M. Peijun, "A test-suite reduction approach to improving fault-localization effectiveness," *Computer Languages, Systems and Structures*, vol. 39, no. 3, pp. 95–108, 2013.
- [15] J. Xuan and M. Monperrus, "Test case purification for improving fault localization," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 52–63, ACM, November 2014.
- [16] C. Gong, Z. Zheng, W. Li, and P. Hao, "Effects of class imbalance in test suites: an empirical study of spectrum-based fault localization," in *Proceedings of the 36th Annual IEEE International Computer Software and Applications Conference Workshops (COMPSACW '12)*, pp. 470–475, IEEE, Izmir, Turkey, July 2012.
- [17] Y. Gao, Z. Zhang, L. Zhang, C. Gong, and Z. Zheng, "A theoretical study: the impact of cloning failed test cases on the effectiveness of fault localization," in *Proceedings of the 13th International Conference on Quality Software (QSIC '13)*, pp. 288–291, Naging, China, July 2013.
- [18] P. Rao, Z. Zheng, T. Y. Chen, N. Wang, and K. Cai, "Impacts of test suite's class imbalance on spectrum-based fault localization

- techniques,” in *Proceedings of the 13th International Conference on Quality Software (QSIC '13)*, pp. 260–267, Najing, China, July 2013.
- [19] V. Debroy and W. E. Wong, “Insights on fault interference for programs with multiple bugs,” in *Proceedings of the 20th International Symposium on Software Reliability Engineering (ISSRE '09)*, pp. 165–174, IEEE, Karnataka, India, November 2009.
- [20] W. Eric Wong, V. Debroy, and B. Choi, “A family of code coverage-based heuristics for effective fault localization,” *Journal of Systems and Software*, vol. 83, no. 2, pp. 188–208, 2010.
- [21] W. E. Wong, V. Debroy, and D. Xu, “Towards better fault localization: a crosstab-based statistical approach,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 3, pp. 378–396, 2012.
- [22] W. Eric Wong, V. Debroy, R. Gao, and Y. Li, “The dstar method for effective software fault localization,” *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 290–308, 2014.
- [23] W. E. Wong and Y. Qi, “BP neural network-based effective fault localization,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 19, no. 4, pp. 573–597, 2009.

## Research Article

# Fault Localization Analysis Based on Deep Neural Network

**Wei Zheng, Desheng Hu, and Jing Wang**

*College of Software & Microelectronics, Northwestern Polytechnical University, Xi'an, China*

Correspondence should be addressed to Desheng Hu; [hudeshengnpu@gmail.com](mailto:hudeshengnpu@gmail.com)

Received 24 December 2015; Accepted 31 March 2016

Academic Editor: Wen Chen

Copyright © 2016 Wei Zheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With software's increasing scale and complexity, software failure is inevitable. To date, although many kinds of software fault localization methods have been proposed and have had respective achievements, they also have limitations. In particular, for fault localization techniques based on machine learning, the models available in literatures are all shallow architecture algorithms. Having shortcomings like the restricted ability to express complex functions under limited amount of sample data and restricted generalization ability for intricate problems, the faults cannot be analyzed accurately via those methods. To that end, we propose a fault localization method based on deep neural network (DNN). This approach is capable of achieving the complex function approximation and attaining distributed representation for input data by learning a deep nonlinear network structure. It also shows a strong capability of learning representation from a small sized training dataset. Our DNN-based model is trained utilizing the coverage data and the results of test cases as input and we further locate the faults by testing the trained model using the virtual test suite. This paper conducts experiments on the Siemens suite and Space program. The results demonstrate that our DNN-based fault localization technique outperforms other fault localization methods like BPNN, Tarantula, and so forth.

## 1. Introduction

Many efforts have been made for debugging a generic program, especially in the stage of identifying where the bugs are, which is known as fault localization. It has been proved that fault localization is one of the most expensive and time-consuming debugging activities. With software's increasing scale and complexity, the debugging activities are more difficult to perform; thus, there is a high demand for automatic fault localization techniques that can guide programmers to the locations of faults [1]. In this way, it will facilitate the software development process and reduce maintenance cost [2]. At present, many kinds of software fault localization methods have been proposed. Probabilistic program dependence graph (PPDG) is presented by Baah et al. in [3] and it gives out the conditional probability of each node to locate fault. Tarantula, which is proposed by Jones and Harrold in [4], indicates that a program entity executed by failed test cases should be suspected and they use different colors to represent the degree of suspiciousness. SOBER, which is proposed by Liu et al. in [5], uses the differences of predicated truth value between successful and failed execution to guide

the activities of finding faults. CBT, namely, crosstab-based technique, is proposed by Wong et al. [6] to calculate the suspiciousness of each executable statement as the detected priority. Meanwhile, other fault localization techniques are presented in [7, 8], such as delta debugging and predicate switching. By modifying the variables or their values at a particular point during the execution to change program state, these techniques are able to identify the factor that triggers the program failure. Specifically, Zeller and Hildebrandt [7] propose the delta debugging to reduce the factors triggering the failures to a small set of variables by comparing the program state difference between the failed test case and successful test case. Predicate switching, which is presented by Zhang et al. [8], alters the program state of failed execution by changing the predicate. The predicate is labeled as critical predicate if its switch can make program execute successfully.

Being robust and widely applied, many machine learning and data mining techniques have been adopted to facilitate the fault localization in recent years [9]. Wong and Qi propose a backpropagation (BP) neural network in [10], which utilizes the coverage data of test cases (e.g., the coverage data with respect to which statements are executed by which test case)

and their corresponding execution results to train the network and, then, input the coverage of a set of virtual test cases (e.g., each test case covers only one statement) to the trained network, and the outputs are regarded as the suspiciousness (i.e., likelihood of containing the bug) of each executable statement. But the BP neural network leads to problems like local minima. Aiming to solve this problem, Wong et al. propose another fault localization method based on radial basis function (RBF) network in [11], which is less sensitive to those problems. Although machine learning techniques exhibit good performance in the field of fault localization, the models they used for fault location are all shallow architecture. With shortcomings like limited ability to express complex function under limited amount of sample data as well as restricted generalization ability for intricate problem, the faults cannot be analyzed exactly via those methods.

This paper proposes a fault localization method based on deep neural network (DNN). With the capability of estimating complicated functions by learning a deep nonlinear network structure and further attaining distributed representation of input data, this method exhibits strong ability to learn representation from minority sample data. Moreover, DNN is one of the deep learning models that has been successfully applied in many other areas of software engineering [12, 13]. For example, the researchers in Microsoft Research adopt DNN to decrease the error rate of speech recognition in [12], which is one of the greatest breakthroughs in that field in the recent ten years. We use the Siemens suite and Space program as platforms to evaluate and demonstrate the effectiveness of DNN-based fault localization technique. The remainder of this paper is organized as follows. Section 2 provides an introduction of some related studies. Then, in Section 3, we elaborate our DNN-based fault localization method and provide a toy example to help readers understand this method. We conduct empirical studies on two suites (i.e., Siemens suite and Space program) in Section 4. Then, Section 5 follows where we report the result of the performance (effectiveness) comparison between our method and others and make a discussion. Section 6 lists possible threats to the validity of our approach. We present our conclusion and future work in Section 7.

## 2. Related Work

Recent years have witnessed the successful application of machine learning techniques in the field of fault localization. However, many models with shallow architectures encounter drawbacks like their restricted ability to express complex function under limited amount of sample data, such as BP neural network and support vector machine. And the generalization ability for intricate problem is also restrained. With the rapid development of the deep learning, many researchers begin to adopt deep neural networks to tackle the limitations of shallow architectures gradually. Before presenting our DNN-based fault localization technique, we introduce some related work that contributes to our novel approach.

*2.1. Deep Neural Network Model.* In 2006, deep neural network is firstly presented by Hinton et al. in the journal Science

[14]. The rationale of DNN is that the neural network model is firstly divided into a number of two-layer models before we learn the whole model, and then we train the two-layer neural network model layer by layer and finally get the initial weights of multilayer neural networks by composing the trained two-layer neural networks, the whole process of which is called layerwise pretraining [15]. The hidden layer of neural network can extract features from the input layer due to its abstraction. Thus, the neural networks with multiple hidden layers are better at network processing and network generalization and achieve faster convergence rate. We elaborate on the theory of deep neural networks which is cited as the basis of our technique here.

DNN is a sort of feed-forward artificial neural network with multiple hidden layers, and each node at the same hidden layer can use the same nonlinear function to map the feature input from the layer below to the current node. DNN structure is very flexible due to the multiple hidden layers and multiple hidden nodes, so DNN demonstrates excellent capacity to fit the highly complex nonlinear relationship between inputs and outputs.

Generally, DNN model can be utilized for regression or classification. In this paper, the model is considered to be a classification model, but in order to output continuous suspiciousness values, we do not normalize the incentive value of DNN's last layer into integer. The relationship between inputs and outputs in DNN model can be interpreted as follows:

$$\begin{aligned} v^0 &= \text{input}, \\ v^{l+1} &= \rho(z^l(v^l)), \\ z^l(v^l) &= w^l(v^l) + b^l, \quad 0 \leq l < L, \\ \text{output} &= v^L. \end{aligned} \tag{1}$$

According to the above formulas, we can obtain the final output by transforming the features vector of the first layer  $v^0$  into a processed feature vector  $v^l$  through  $L$  layers of nonlinear transformation. During the training process of DNN model, we need to determine the weight matrix  $w^l$  and offset vector  $b^l$  of  $l$ th layer. By utilizing the difference between the target outputs and the actual outputs to construct a cost function, we can then train the DNN by backpropagation (BP) algorithm.

Mainly, the design of DNN model includes steps like designing the number of network layers, the number of nodes in each layer, the transfer function between the layers, and so forth.

*2.1.1. The Design of the Network Layer.* Generally, deep neural network consists of three parts: the input layer, hidden layer, and output layer. The study of the neural network layers mainly aims to identify the number of hidden layers to determine the number of layers of a network. In neural networks, hidden layer has an effect of abstraction and can extract features from the input. At the same time, the number

of hidden layers directly determines the processing ability of network to extract feature. If the number of hidden layers is too small, it cannot represent and fit the intricate problems with limited amount of sample data, while, with increased number of hidden layers, the processing ability of the entire network will improve. However, too many hidden layers also produce some adverse effects, such as the increased complexity of calculation and local minima. Thus, we draw a conclusion that too many or too few hidden layers are all unfavorable for the network training. We must choose the appropriate number of network layers to adapt to the different practical problems.

**2.1.2. The Design of Number of Nodes.** Usually, the nodes of input layer and output layer can be determined directly when the training samples of deep neural network are confirmed; thus, determining the number of nodes in the hidden layer is most critical. If we set up too many hidden nodes, it will increase the training time and train the neural network excessively (i.e., remember some unnecessary information) and lead to problems like overfitting, whereas the network is not able to handle complex problem as its ability to obtain information gets poorer with too few hidden nodes. The number of hidden nodes has a direct relationship with inputs and outputs, and it needs to be determined according to several experiments.

**2.1.3. The Design of Transfer Function.** Transfer function reflects the complex relationship between the input and its output. Different problems are suited to different transfer functions. In our work, we use the common sigmoid function as a nonlinear transfer function:

$$\rho(s) = \frac{1}{1 + e^{-s}}. \quad (2)$$

Here,  $s$  is the input and  $\rho(s)$  is the output.

**2.1.4. The Design of Learning Rate and Impulse Factor.** The learning rate and impulse factor have an important impact on DNN as well. A smaller learning rate will increase training time and slow down the speed of convergence; on the other hand, it generates network shocks further. For the choice of impulse factor, we can use its “inertia” to cushion the network shocks. We choose the appropriate learning rate and impulse factor according to the size of the sample.

**2.2. Learning Algorithm of DNN.** There exist various pre-training methods for DNN and mainly they are divided into two categories, namely, the unsupervised pretraining and supervised pretraining.

**2.2.1. Unsupervised Pretraining.** For the unsupervised pretraining, it usually adopts Restricted Boltzmann Machine (RBM) to initialize a DNN. RBM is proposed by Hinton in 2010 [16]. The structure of a RBM is depicted in Figure 1. The RBM is divided into two layers: the first one is visible layer with visible units and the second one is hidden layer with hidden units. Once the unsupervised pretraining has been

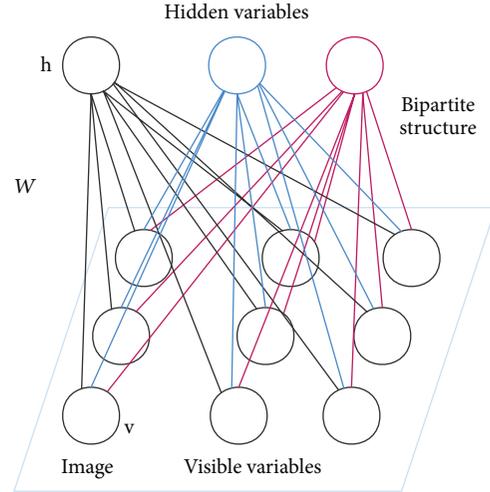


FIGURE 1: Architecture of a Restricted Boltzmann Machine.

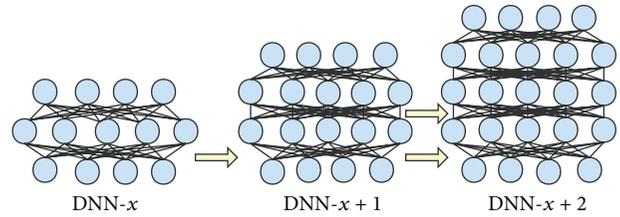


FIGURE 2: Architecture of supervised pretraining.

completed, the obtained parameters will be used as the initial weights of the DNN.

**2.2.2. Supervised Pretraining.** In order to reduce the inaccuracy of unsupervised training, we adopt the supervised pretraining proposed in [17]. The general architecture is shown in Figure 2. It works as follows: the BP algorithm is utilized to train a neural network, and the topmost layer is replaced by a randomly initialized hidden layer and a new random topmost layer after every pretraining. The network is unceasingly trained again until convergence or reaching the expected number of hidden layers.

In our work, the number of nodes in the input layer of DNN model is equal to the dimension of the input feature vector. The output layer has only one output node, that is, suspiciousness value. After a distinctive pretraining, the BP algorithm is used to fine-tune the parameters of the model. Here,  $y_{1:T}$  is the training sample, and the goal is to minimize the squared error sum between the training samples  $y_{1:T}$  and labeling  $x_{1:T}$ . The objective function can be interpreted as follows:

$$E(W, b) = \frac{1}{2} \sum_t \|f(y_t, W, b) - x_t\|^2. \quad (3)$$

Suppose that layer index is  $0, 1, \dots, L$ , node layer is represented by  $v_0, v_1, \dots, w_0, w_1, \dots, w_{L-1}$  denotes the weight layer, and  $k_L$  is the index of nodes in the  $L$ th layer and then

take the derivative of the weight matrix  $W^{L-1}$  and offset vector  $b^{L-1}$ :

$$\begin{aligned} E(W, b) &= \frac{1}{2} \sum_t \|\rho(z^{L-1} - x_t)\| \\ &= \frac{1}{2} \sum_t \sum_{k_L} (\rho(z_{k_L}^{L-1}) - x_{t,k_L})^2 \\ &= \frac{1}{2} \sum_t \sum_{k_L} \left( \rho \left( \sum_{k_{L-1}} w_{k_L k_{L-1}}^{L-1} v_{k_{L-1}}^{L-1} + b_{k_L}^{L-1} \right) - x_{t,k_L} \right)^2. \end{aligned} \quad (4)$$

Take the derivative for each component of  $W^{L-1}$  and  $b^{L-1}$ :

$$\begin{aligned} \frac{\partial E(W, b)}{\partial W_{k_L k_{L-1}}^{L-1}} &= \sum_t (\rho(z_{k_L}^{L-1}) - x_{t,k_L}) \rho'(z_{k_L}^{L-1}) (v_{k_{L-1}}^{L-1}), \\ \frac{\partial E(W, b)}{\partial b_{k_L}^{L-1}} &= \sum_t (\rho(z_{k_L}^{L-1}) - x_{t,k_L}) \rho'(z_{k_L}^{L-1}). \end{aligned} \quad (5)$$

Then, take the derivative of the weight matrix  $W^{L-2}$  and offset vector  $b^{L-2}$ :

$$\begin{aligned} E(W, b) &= \frac{1}{2} \sum_t \|\rho(z^{L-1} - x_t)\| = \frac{1}{2} \sum_t \sum_{k_L} (\rho(z_{k_L}^{L-1}) - x_{t,k_L})^2 = \frac{1}{2} \sum_t \sum_{k_L} \left( \rho \left( \sum_{k_{L-1}} w_{k_L k_{L-1}}^{L-1} v_{k_{L-1}}^{L-1} + b_{k_L}^{L-1} \right) - x_{t,k_L} \right)^2 \\ &= \frac{1}{2} \sum_t \sum_{k_L} \left( \rho \left( \sum_{k_{L-1}} w_{k_L k_{L-1}}^{L-1} \rho(z_{k_L}^{L-1}) + b_{k_L}^{L-1} - x_{t,k_L} \right)^2 \right) \\ &= \frac{1}{2} \sum_t \sum_{k_L} \left( \rho \left( \sum_{k_{L-1}} w_{k_L k_{L-1}}^{L-1} \rho \left( \sum_{k_{L-2}} W_{k_{L-1} k_{L-2}}^{L-2} v_{k_{L-2}}^{L-2} + b_{k_{L-1}}^{L-2} \right) + b_{k_L}^{L-1} - x_{t,k_L} \right)^2 \right). \end{aligned} \quad (6)$$

Take the derivative for each component of  $W^{L-2}$  and  $b^{L-2}$ :

$$\begin{aligned} \frac{\partial E(W, b)}{\partial W_{k_L k_{L-2}}^{L-2}} &= \sum_t \sum_{k_L} [(\rho(z_{k_L}^{L-1}) - x_{t,k_L}) \rho'(z_{k_L}^{L-1}) W_{k_L k_{L-1}}^{L-1}] \\ &\quad \cdot \rho'(z_{k_{L-1}}^{L-2}) v_{k_{L-2}}^{L-2}, \end{aligned} \quad (7)$$

$$\begin{aligned} \frac{\partial E(W, b)}{\partial b_{k_L}^{L-2}} &= \sum_t \sum_{k_L} [(\rho(z_{k_L}^{L-1}) - x_{t,k_L}) \rho'(z_{k_L}^{L-1}) W_{k_L k_{L-1}}^{L-1}] \\ &\quad \cdot \rho'(z_{k_{L-1}}^{L-2}). \end{aligned}$$

The result is as follows:

$$\begin{aligned} \frac{\partial E(W, b)}{\partial W^l} &= \sum_t e^{l+1}(t) (v^l(t))^T, \\ \frac{\partial E(W, b)}{\partial b^l} &= \sum_t e^{l+1}(t). \end{aligned} \quad (8)$$

Here,

$$\begin{aligned} e_{k_L}^L(t) &= (\rho(z_{k_L}^{L-1}) - x_{t,k_L}) \rho'(z_{k_L}^{L-1}), \\ e_{k_{L-1}}^{L-1}(t) &= \sum_{k_L} e_{k_L}^L(t) W_{k_L k_{L-1}}^{L-1} \rho'(z_{k_{L-1}}^{L-2}). \end{aligned} \quad (9)$$

After vectorization,

$$\begin{aligned} e^L(t) &= \text{diag}(\rho'(z_{k_L}^{L-1})) (\rho(z^{L-1}) - x_t), \\ e_k^L(t) &= \text{diag}(\rho'(z_{k_{L-1}}^{L-2})) (W^{L-1})^T e^L(t). \end{aligned} \quad (10)$$

The recursive process is as follows:

$$\begin{aligned} e^L(t) &= Q^L(t) (f(y_t, (W, b)^0) - x_t), \\ e^l(t) &= Q^L(t) (W^l)^T e^{l+1}(t), \\ Q^l(t) &= \text{diag}(\rho'(z^{l-1}(v^{l-1}(t))))), \\ \rho'(z) &= \rho(z) \cdot (1 - \rho(z)). \end{aligned} \quad (11)$$

In order to calculate the error between the actual output and the expected output, we output samples  $y_{1:T}$  to DNN and then execute the forward process of DNN. Meanwhile, we calculate the outputs of all the hidden layer nodes and output nodes, and now we can compute to get the error  $e^L(t)$ . Next, backpropagation procedure is executed and error of the nodes on each hidden layer is calculated iteratively after obtaining  $e^L(t)$ . The parameters of DNN can be updated layer by layer according to the following formula:

$$\begin{aligned} (W^l, b^l)^{m+1} &= (W^l, b^l)^m + \Delta(W^l, b^l)^m, \quad 0 \leq l \leq L, \\ \Delta(W^l, b^l)^m &= (1 - \alpha) \varepsilon \left( \frac{\partial E}{\partial (W^l, b^l)} \right) \\ &\quad + \alpha \Delta(W^l, b^l)^{m-1}, \quad 0 \leq l \leq L. \end{aligned} \quad (12)$$

TABLE 1: The coverage data and execution result of test case.

	$s_1$	$s_2$	$s_3$	$\cdots$	$s_{m-1}$	$s_m$	$R$
$t_1$	1	0	1	$\cdots$	0	1	0
$t_2$	1	1	1	$\cdots$	0	1	0
$t_3$	1	0	1	$\cdots$	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$t_{n-1}$	1	0	0	$\cdots$	1	0	1
$t_n$	1	1	1	$\cdots$	0	0	0

Here,  $\varepsilon$  is the learning rate,  $\alpha$  is the impulse factor, and  $m$  denotes the  $m$ th iteration. In the process of fault localization, we input the virtual test matrix  $Y$  into the DNN model and then execute the forward process of DNN, and finally the output is the suspiciousness value of each statement.

### 3. DNN-Based Fault Localization Technique

While the previous section provides an overview of the DNN model, this part will present the methodology of DNN-based fault localization technique in detail. With a focus on how to build a DNN model for fault localization problem, meanwhile, we will summarize the procedures of localizing faults using DNN and provide a concrete example for demonstration at the end of this part.

**3.1. Fault Localization Algorithm Based on Deep Neural Network.** Suppose that there is a program  $P$  with  $m$  executable statements and  $n$  test cases to be executed,  $t_i$  denotes the  $i$ th test case, while vectors  $c_{t_i}$  and  $r_{t_i}$  represent the corresponding coverage data and execution result, respectively, after executing the test case  $t_i$ , and  $s_j$  is  $j$ th executable statement of program  $P$ . Here,  $c_{t_i} = [(c_{t_i})_1, (c_{t_i})_2, \dots, (c_{t_i})_m]$ . If the executable statement  $s_j$  is covered by  $t_j$ , we assign a real value 1 to  $(c_{t_i})_j$ ; otherwise, we assign 0 to it. If the test case  $t_i$  is executed successfully, we assign a real value 0 to  $r_{t_i}$ ; otherwise, it is assigned with 1. We depict the coverage data and execution result of test case through Table 1. For instance, from Table 1, we can draw a conclusion that  $s_2$  is not covered with failed test case  $t_3$ , while  $s_3$  is covered with failed test case  $t_2$  according to Table 1.

The network can reflect the complex nonlinear relationship between the coverage data and execution result of test case when the training of deep neural network is completed. We can identify the suspicious code of faulty version by trained network. At the same time, constructing a set of virtual test cases is necessary, as shown in the following equation.

*The Virtual Test Set.* Consider

$$\begin{bmatrix} c_{v_1} \\ c_{v_2} \\ \vdots \\ c_{v_m} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}; \quad (13)$$

the vector  $c_{v_1}, c_{v_2}, \dots, c_{v_m}$  denotes the coverage data of test cases  $v_1, v_2, \dots, v_m$ .

In the set of virtual test cases, the test case  $v_j$  only covers executable statement  $s_j$ . Since each test case only covers one statement, thus the statement is highly suspicious if the corresponding test case is failed. For example,  $s_j$  is very likely to contain bugs with a failed test case  $v_j$ , and that means we should preferentially check the statements which are covered by failed test cases. But such set of test cases does not exist in reality; thus, the execution result of test case in the virtual test set is difficult to get in practice. So we input  $c_{v_j}$  to the trained DNN, and the output  $r_{v_j}$  represents the probability of test case execution result. The value of  $r_{v_j}$  is proportional to the suspicious degree of containing bugs of  $s_j$ .

Specific fault location algorithm is as follows:

- (1) Construct the DNN model with one input layer and one output layer, and identify the appropriate number of hidden layers according to the experiment scale. Suppose the number of input layer nodes is  $m$ , the number of hidden layer nodes is  $n$ , the number of output layer nodes is 1, and the adopted transfer function is sigmoid function  $\rho(s) = 1/(1 + e^{-s})$ .
- (2) Utilize the coverage data and execution result of test case as training sample set which is inputted into the DNN model, and then train the DNN model to obtain the complex nonlinear mapping relationship between the coverage data  $c_{t_i}$  and execution result  $r_{t_i}$ .
- (3) Input the coverage data vector of virtual test set  $c_{v_j}$  ( $1 \leq j \leq m$ ) into the DNN model and get the output  $r_{v_j}$  ( $1 \leq j \leq m$ ).
- (4) The output  $r_{v_j}$  reflects the probability that executable statement  $s_j$  contains the bugs, that is, suspiciousness value, and then conduct descending ranking for  $r_{v_j}$ .
- (5) Rank  $s_j$  ( $1 \leq j \leq m$ ) according to their corresponding suspiciousness value and check the statement one by one from the most suspicious to the least until the faults are finally located.

**3.2. The Example of Fault Localization Algorithm Based on Deep Neural Network.** Here, we illustrate the application of fault localization analysis based on deep neural network by a concrete example. It is depicted in Table 2.

Table 2 shows that the function of program  $\text{Mid}(\cdot)$  is to get the middle number by comparing three integers. The program has twelve statements and ten test cases. There is one fault in the program, which is contained by statement (6). Here, “•” denotes that the statement is covered by the corresponding test case. The blank denotes that the statement is not covered by test case. P represents that the test case is executed successfully while F represents that the execution results of test case are failed.

The coverage data and execution results of test cases are shown in Table 2. Table 3 correlates with Table 2. Here, number 1 replaces “•” and indicates that the statement is covered by the corresponding test case, and number 0

TABLE 2: Coverage and execution result of Mid function.

Function	Test cases									
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$
Mid( $x, y, z$ ) {int $m$ ; (1) $m = z$ (2) if ( $y < z$ ) (3) if ( $x < y$ ) (4) $m = y$ (5) else if ( $x < z$ ) (6) $m = y$ //bug (7) else (8) if ( $x > y$ ) (9) $m = y$ (10) else if ( $x > z$ ) (11) $m = x$ (12) printf ( $m$ );} Pass/fail	3, 3, 5	1, 2, 3	3, 2, 2	5, 5, 5	1, 1, 4	5, 3, 4	3, 2, 1	5, 4, 2	2, 1, 3	5, 2, 6
	•	•	•	•	•	•	•	•	•	•
	•	•	•	•	•	•	•	•	•	•
	•	•			•	•			•	•
	•	•			•				•	•
		•								
			•	•			•	•		
			•	•			•	•		
			•				•			•
	•	•	•	•	•	•	•	•	•	•
	P	P	P	P	P	P	P	P	F	F

TABLE 3: Switched Table 2.

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_{12}$	$r$
$t_1$	1	1	1	0	1	1	0	0	0	0	0	1	0
$t_2$	1	1	1	1	0	0	0	0	0	0	0	1	0
$t_3$	1	1	0	0	0	0	1	1	1	0	0	1	0
$t_4$	1	1	0	0	0	0	1	1	0	1	0	1	0
$t_5$	1	1	1	0	1	1	0	0	0	0	0	1	0
$t_6$	1	1	1	0	1	0	0	0	0	0	0	1	0
$t_7$	1	1	0	0	0	0	1	1	1	0	0	1	0
$t_8$	1	1	0	0	0	0	1	1	1	0	0	1	0
$t_9$	1	1	1	0	1	1	0	0	0	0	0	1	1
$t_{10}$	1	1	1	0	1	1	0	0	0	0	0	1	1

replaces the blank and represents that the statement is not covered by the corresponding test case. In the last column, number 1 replaces F and indicates that the execution results of test case are failed, while number 0 replaces P and represents that the test case is executed successfully.

The concrete fault localization process is as follows:

- (1) Construct the DNN model with one input layer, one output layer, and three hidden layers. The number of input layer nodes is 12, and the number of hidden layer nodes is simply set as 4. The number of output layer nodes is 1, and the transfer function adopted is the sigmoid function  $\rho(s) = 1/(1 + e^{-s})$ .
- (2) Use the coverage data and execution result of test case as training sample set to input into the DNN model. First, we input the vector (1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1) and its execution result 0 and, next, input the second vector (1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1) and its execution result 0 until the coverage data and execution results of the ten test cases are all inputted into the network. We then train the DNN model utilizing the inputted

training data, and further we input the next training dataset (i.e., another ten test cases' coverage data and execution results) iteratively to optimize the DNN model until we reach the condition of convergence. The final DNN model we obtained after several times' iteration reveals the complex nonlinear mapping relationship between the coverage data and execution result.

- (3) Construct the virtual test set with twelve test cases and ensure that each test case only covers one executable statement. The set of virtual test cases is depicted in Table 4.
- (4) Input the virtual test set into the trained DNN model and get the suspiciousness value of each corresponding executable statement, and then rank the statements according to their suspiciousness values.
- (5) Table 5 shows the descending ranking of statements according to their suspiciousness value. According to the table of ranking list, we find that the suspiciousness value of the sixth statement which contains the bug in program Mid( $\cdot$ ) is the greatest (i.e., rank first); thus, we can locate the fault by checking only one statement.

## 4. Empirical Studies

So far, the modeling procedures of our DNN-based fault localization technique have been discussed in the previous section. In this section, we empirically compare the performance of this technique with that of Tarantula localization technique [4], PPDG localization technique [3], and BPNN localization technique [10] on the Siemens suite and Space program to demonstrate the effectiveness of our DNN-based fault localization method. The Siemens suite and Space

TABLE 4: Virtual test suite.

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_{12}$
$t_1$	1	0	0	0	0	0	0	0	0	0	0	0
$t_2$	0	1	0	0	0	0	0	0	0	0	0	0
$t_3$	0	0	1	0	0	0	0	0	0	0	0	0
$t_4$	0	0	0	1	0	0	0	0	0	0	0	0
$t_5$	0	0	0	0	1	0	0	0	0	0	0	0
$t_6$	0	0	0	0	0	1	0	0	0	0	0	0
$t_7$	0	0	0	0	0	0	1	0	0	0	0	0
$t_8$	0	0	0	0	0	0	0	1	0	0	0	0
$t_9$	0	0	0	0	0	0	0	0	1	0	0	0
$t_{10}$	0	0	0	0	0	0	0	0	0	1	0	0
$t_{11}$	0	0	0	0	0	0	0	0	0	0	1	0
$t_{12}$	0	0	0	0	0	0	0	0	0	0	0	1

TABLE 5

Statement	Suspiciousness	Rank
1	0.0448	11
2	0.1233	2
3	0.0879	5
4	0.0937	4
5	0.0747	7
6 (fault)	0.1593	1
7	0.1191	3
8	0.0758	6
9	0.0661	8
10	0.0436	12
11	0.0650	9
12	0.0649	10

program can be downloaded from [18]. The evaluation standard based on statements ranking is proposed by Jones and Harrold in [4] to compare the effectiveness of the Tarantula localization technique with other localization techniques. Tarantula localization technique can produce the suspiciousness ranking of statements. The programmers examine the statements one by one from the first to the last until the fault is located and the percentage of statements without being examined is defined as the score of fault localization technique, that is, EXAM score. Since the DNN-based fault localization technique, Tarantula localization technique, and PPDG localization technique all produce a suspiciousness value ranking list for executable statements, we adopt the EXAM score as the evaluation criterion to measure the effectiveness.

**4.1. Data Collection.** The physical environment on which our experiments were carried out included 3.40 GHz Intel Core i7-3770 CPU and 16 GB physical memory. The operating systems were Windows 7 and Ubuntu 12.10. Our compiler was gcc 4.7.2. We conducted experiments on the MATLAB R2013a. To collect the coverage data of each statement, toolbox software named Gcov was used here. The test case

execution results of faulty program versions can be obtained by the following method:

- (1) We run the test cases on fault-free program versions to get the execution results.
- (2) We run the test cases on faulty program versions to get the execution results.
- (3) We compare the results of fault-free program versions and results of faulty program versions. If they are equal, the test case is successful; otherwise, it is failed.

**4.2. Programs of the Test Suite.** In our experiment, we conducted two studies on the Siemens suite and Space program to demonstrate the effectiveness of the DNN-based fault localization technique.

**4.2.1. The Siemens Suite.** The Siemens suite has been considered as a classical test sample which is widely employed in studies of fault localization techniques. And, in this paper, our experiment begins with it. The Siemens suite contains seven C programs with the corresponding faulty versions and test cases. Each faulty version has only one fault, but it may cross the multiline statements or even multiple program functions in the program. Table 6 provides the detailed introduction of the seven C programs in Siemens suite, including the program name, number of faulty versions of each program, number of statement lines, number of executable statements lines, and the number of test cases.

The function of the seven C programs is different. The Print\_tokens and Print\_tokens 2 programs are used for lexical analysis, and the function of Replace program is pattern replacement. The Schedule and Schedule 2 programs are utilized for priority scheduler while the function of Tcas is altitude separation, and the Tot\_info program is used for information measure.

The Siemens suite contains 132 faulty versions, while we choose 122 faulty versions to perform experiments. We omitted the following versions: versions 4 and 6 of Print\_tokens, version 9 of Schedule 2, version 12 of Replace, versions 13, 14, and 36 of Tcas, and versions 6, 9, and 21 of Tot\_info. We eliminated these versions because of the following:

- (a) There is no syntactic difference between the correct version and the faulty versions (e.g., only difference in the header file).
- (b) The test cases never fail when executing the programs of faulty versions.
- (c) The faulty versions have segmentation faults when executing the test cases.
- (d) The difference between the correct versions and the faulty versions is not included in executable statements of program and cannot be replaced. In addition, there exists absence of statements in some faulty versions and we cannot locate the missing statements directly. In this case, we can only select the associated statements to locate the faults.

TABLE 6: Summary of the Siemens suite.

Program	Number of faulty versions	LOC (lines of code)	Number of executable statements	Number of test cases
Print_tokens	7	565	172	4130
Print_tokens 2	10	510	146	4115
Replace	32	563	175	5542
Schedule	9	412	140	2650
Schedule 2	10	307	115	2710
Tcas	41	173	59	1608
Tot_info	23	406	100	1052

4.2.2. *The Space Program.* As the executable statements of Siemens suite programs are only about hundreds of lines, the scale and complexity of the software have increased gradually and the program may be of as many as thousands or ten thousands of lines. So we verify the effectiveness of our DNN-based fault localization technique on large-scale program set, that is, Space program.

The Space program contains 38 faulty versions; each version has more than 9000 statements and 13585 test cases. In our experiment, due to similar reasons which have been depicted in Siemens suite, we also omit 5 faulty versions and select another 33 faulty versions.

4.3. *Experimental Process.* In this paper, we conducted experiments on the 122 faulty versions of Siemens suite and 33 faulty versions of Space program. For the DNN modeling process, we adopted three hidden layers; that is, the structure of the network is composed of one input layer, three hidden layers, and one output layer, which is based on previous experience and the preliminary experiment. According to the experience and sample size, we estimate the number of hidden layer nodes (i.e., num) by the following formula:

$$\text{num} = \text{round}\left(\frac{n}{30}\right) * 10. \quad (14)$$

Here,  $n$  represents the number of input layer nodes. The impulse factor is set as 0.9, and the range of learning rate is  $\{0.01, 0.001, 0.0001, 0.00001\}$ ; we select the most appropriate learning rate as the final parameter according to the sample scale.

Now, we take Print\_tokens program as an example and introduce the experimental process in detail. The other program versions can refer to the Print\_tokens. The Print\_tokens program includes seven faulty versions and we pick out five versions to conduct the experiment. The experiment process is as follows:

- (1) Firstly, we compile the source program of Print\_tokens and run the test case set to get the expected execution results of test cases.
- (2) We compile the five faulty versions of Print\_tokens program using GCC and similarly get the actual execution results of test cases and acquire the coverage data of test cases by using Gcov technique.
- (3) We construct the set of virtual test cases for the five faulty versions of Print\_tokens program, respectively.
- (4) Then, we compare the execution results of test cases between the source program and the five faulty versions of Print\_tokens to get the final execution results of the test cases of five faulty versions.
- (5) We integrate the coverage data and the final results of test cases of the five faulty versions, respectively, as input vectors to train the deep neural network.
- (6) Finally, we utilize the set of virtual test cases to test the deep neural network to acquire the suspiciousness of the corresponding statements and then rank the suspiciousness value list.

## 5. Results and Analysis

According to the experimental process described in the previous section, we perform the fault localization experiments on deep neural network and BP neural network and statistically analyze the experimental results. We then compare the experimental results with Tarantula localization technique and PPDG localization technique. The experimental results of Tarantula and PPDG have been introduced in [3, 4], respectively.

5.1. *The Experimental Result of Siemens Suite.* Figure 3 shows the effectiveness of DNN, BP neural network, Tarantula, and PPDG localization techniques on the Siemens suite.

In Figure 3, the  $x$ -axis represents the percentage of statements without being examined until the fault is located, that is, EXAM score, while the  $y$ -axis represents the percentage of faulty versions whose faults have been located. For instance, there is a point labeled on the curve whose value is (70, 80). This means the technique is able to find out eighty percent of the faults in the Siemens suite with seventy percent of statements not being examined; that is, we can identify eighty percent of the faults in the Siemens suite by only examining thirty percent of the statements.

Table 7 offers further explanation for Figure 3.

According to the approach of experiment statistics from other fault localization techniques, the EXAM score can be divided into 11 segments. Each 10% is treated as one segment, but the caveat is that the programmers are not able to identify

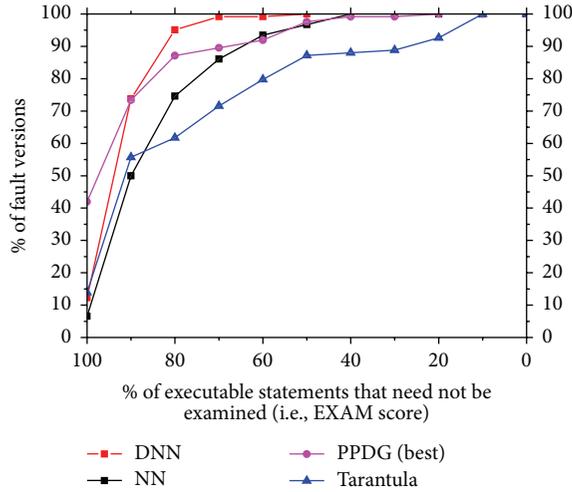


FIGURE 3: Effectiveness comparison on the Siemens suite.

TABLE 7: Effectiveness comparison of four techniques.

EXAM score	% of the faulty versions			
	DNN	BPNN	PPDG	Tarantula
100%-99%	12.29%	6.56%	41.94%	13.93%
99%-90%	61.48%	43.44%	31.45%	41.80%
90%-80%	21.31%	24.59%	13.71%	5.74%
80%-70%	4.10%	11.48%	2.42%	9.84%
70%-60%	0.00%	7.38%	2.42%	8.20%
60%-50%	0.82%	3.28%	5.65%	7.38%
50%-40%	0.00%	3.28%	1.61%	0.82%
40%-30%	0.00%	0.00%	0.00%	0.82%
30%-20%	0.00%	0.00%	0.80%	4.10%
20%-10%	0.00%	0.00%	0.00%	7.38%
10%-0%	0.00%	0.00%	0.00%	0.00%

the faults without examining any statement; thus, the abscissa can only be infinitely close to 100%. Due to that factor, we divide the 100%–90% into two segments, that is, 100%–99% and 99%–90%. The data of every segment can be used to evaluate the effectiveness of fault localization technique. For example, for the 99%–90% segment of DNN, the percentage of faults identified in the Siemens suite is 61.48% while the percentage of faults located is 43.44% for the BPNN technique. From Table 7, we can find that, for the segments 50%–40%, 40%–30%, 30%–20%, 20%–10%, and 10%–0% of DNN, the percentage of faults identified in each segment is 0.00%. That indicates that the DNN-based method has found out all the faulty versions after examining 50% of statements.

We analyzed the above data and summarized the following points:

- (1) Figure 3 reveals that the overall effectiveness of our DNN-based fault localization technique is better than that of the BP neural network fault localization technique as the curve representing DNN-based approach is always above the curve denoting BPNN-based

method. Also, the DNN-based technique is able to identify the faults by examining fewer statements compared with the method based on BP neural network. For instance, by examining 10% of statements, the DNN-based method is able to identify 73.77% of the faulty versions while the BP neural network fault localization technique only finds out 50% of the faulty versions.

- (2) Compared with the Tarantula fault localization technique, the DNN-based approach dramatically improves the effectiveness of fault localization on the whole. For example, the DNN-based method is capable of finding out 95.08% of the faulty versions after examining 20% of statements while the Tarantula fault localization technique only identifies 61.47% of the faulty versions after examining the same amount of statements. For the score segment of 100%–99%, the effectiveness of the DNN-based fault localization technique is relatively similar to that of Tarantula; for instance, DNN-based method can find out 12.29% of the faulty versions with 99% of statements not examined (i.e., the abscissa values are 99), and its performance is close to that of Tarantula fault localization technique.
- (3) Compared with the PPDG fault localization technique, the DNN-based fault localization technique reflects improved effectiveness as well for the score range of 90%–0%. Moreover, the DNN-based fault localization technique is capable of finding out all faulty versions by only examining 50% of statements while PPDG fault localization technique needs to examine 80% of statements. For the score range of 100%–99%, the performance of DNN-based fault localization technique is inferior to PPDG fault localization technique.

In conclusion, overall, the DNN-based fault localization technique improves the effectiveness of localization considerably. In particular, for the score range of 90%–0%, the improved effectiveness of localization is more obvious compared with other fault localization techniques as it needs less statements to be examined to further identify the faults. The DNN-based fault localization technique is able to find out all faulty versions by only examining 50% of statements, which is superior to BPNN, Tarantula, and PPDG fault localization technique, while, for the score segment of 100%–99%, the performance of our DNN-based approach is similar to that of Tarantula fault localization method while it is a bit inferior to that of PPDG fault localization technique.

*5.2. The Experimental Result of Space Program.* As the executable statements of Siemens suite programs are only about hundreds of lines, we further verify the effectiveness of DNN-based fault localization technique in large-scale datasets.

Figure 4 demonstrates the effectiveness of DNN and BP neural network localization techniques on the Space program.

From Figure 4, we can clearly find that the effectiveness of our DNN-based method is obviously higher than that

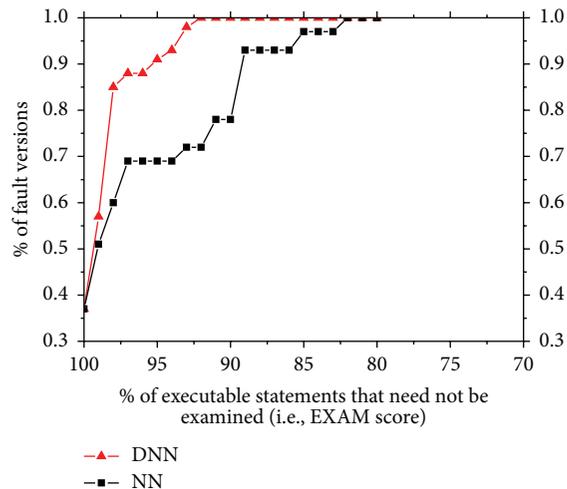


FIGURE 4: Effectiveness comparison on the Space program.

of the BP neural network fault localization technique. The DNN-based fault localization technique is able to find out all faulty versions without examining 93% of statements while BP neural network identifies all faulty versions with 83% of the statements not examined. That comparison indicates that the performance of our DNN-based approach is superior as it reduces the number of statements to be examined. The experimental results show that DNN-based fault localization technique is also highly effective in large-scale datasets.

## 6. Threats to the Validity

There may exist several threats to the validity of the technique presented in this paper. We discuss some of them in this section.

We empirically determine the structure of the network, including the number of hidden layers and the number of hidden nodes. In addition, we determine some important model parameters by grid search method. So probably there exist some better structures for fault localization model. As the DNN model we trained for the first time usually is not the most optimal one, thus empirically we need to modify the parameters of the network several times to try to obtain a more optimal model.

In the field of fault localization, we may encounter multiple-bug programs and we need to construct some new models that can locate multiple bugs. In addition, because the number of input nodes of the model is equal to the lines of executable statements, thus when the number of executable statements is very large, the model will become very big as well. That results in a limitation of fault localization for some big-scale software and we may need to control the scale of the model or to improve the performance of the computer we use. Moreover, virtual test sets built to calculate the suspiciousness of each test case which only covers one statement may not be suitably designed. When we use the virtual test case to describe certain statement, we assume that if a test case is predicated as failed, then the statement covered by it will have a bug. However, whether this assumption is reasonable has not been confirmed. As this assumption is important for

our fault localization approach, thus we may need to test this hypothesis or to propose an alternative strategy.

## 7. Conclusion and Future Work

In this paper, we propose a DNN-based fault localization technique as DNN is able to simulate the complex nonlinear relationship between the input and the output. We conduct an empirical study on Siemens suite and Space program. Further, we compare the effectiveness of our method with other techniques like BP neural network, Tarantula, and PPDG. The results show that our DNN-based fault localization technique performs best. For example, for Space program, DNN-based fault localization technique only needs to examine 10% of statements to fully identify all faulty versions, while BP neural network fault localization technique needs to examine 20% of statements to find out all faulty versions.

As software fault localization is one of the most expensive, tedious, and time-consuming activities during the software testing process, thus it is of great significance for researchers to automate the localization process [19]. By leveraging the strong feature-learning ability of DNN, our approach helps to predict the likelihood of containing fault of each statement in a certain program. And that can guide programmers to the location of statements' faults, with minimal human intervention. As deep learning is widely applied and demonstrates good performance in research field of image processing, speech recognition, and natural language processing and in related industries as well [13], with the ever-increasing scale and complexity of software, we believe that our DNN-based method is of great potential to be applied in industry. It could help to improve the efficiency and effectiveness of the debugging process, boost the software development process, reduce the software maintenance cost, and so forth.

In our future work, we will apply deep neural network in multifaults localization to evaluate its effectiveness. Meanwhile, we will conduct further studies about the feasibility of applying other deep learning models (e.g., convolutional neural network, recurrent neural network) in the field of software fault localization.

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] M. A. Alipour, "Automated fault localization techniques: a survey," Tech. Rep., Oregon State University, Corvallis, Ore, USA, 2012.
- [2] W. Eric Wong and V. Debroy, "A survey of software fault localization," Tech. Rep. UTDCS-45-09, Department of Computer Science, The University of Texas at Dallas, 2009.
- [3] G. K. Baah, A. Podgurski, and M. J. Harrold, "The probabilistic program dependence graph and its application to fault diagnosis," *IEEE Transactions on Software Engineering*, vol. 36, no. 4, pp. 528–545, 2010.
- [4] J. A. Jones and M. J. Harrold, "Empirical evaluation of the tarantula automatic fault-localization technique," in *Proceedings of the 20th IEEE/ACM International Conference on Automated*

- Software Engineering (ASE '05)*, pp. 273–282, Long Beach, Calif, USA, November 2005.
- [5] C. Liu, L. Fei, X. Yan, J. Han, and S. P. Midkiff, “Statistical debugging: a hypothesis testing-based approach,” *IEEE Transactions on Software Engineering*, vol. 32, no. 10, pp. 831–847, 2006.
  - [6] W. E. Wong, T. Wei, Y. Qi, and L. Zhao, “A crosstab-based statistical method for effective fault localization,” in *Proceedings of the 1st International Conference on Software Testing, Verification and Validation (ICST '08)*, pp. 42–51, Lillehammer, Norway, April 2008.
  - [7] A. Zeller and R. Hildebrandt, “Simplifying and isolating failure-inducing input,” *IEEE Transactions on Software Engineering*, vol. 28, no. 2, pp. 183–200, 2002.
  - [8] X. Zhang, N. Gupta, and R. Gupta, “Locating faults through automated predicate switching,” in *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*, pp. 272–281, Shanghai, China, May 2006.
  - [9] S. Ali, J. H. Andrews, T. Dhandapani et al., “Evaluating the accuracy of fault localization techniques,” in *Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE '09)*, pp. 76–87, IEEE Computer Society, Auckland, New Zealand, November 2009.
  - [10] W. E. Wong and Y. Qi, “BP neural network-based effective fault localization,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 19, no. 4, pp. 573–597, 2009.
  - [11] W. E. Wong, V. Debroy, R. Golden, X. Xu, and B. Thuraisingham, “Effective software fault localization using an RBF neural network,” *IEEE Transactions on Reliability*, vol. 61, no. 1, pp. 149–169, 2012.
  - [12] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH '11)*, pp. 437–440, Florence, Italy, August 2011.
  - [13] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
  - [14] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
  - [15] A. Mohamed, G. Dahl, and G. Hinton, “Deep belief networks for phone recognition,” in *Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, Whistler, Canada, December 2009.
  - [16] G. E. Hinton, “A practical guide to training restricted boltzmann machines,” Tech. Rep. UTML TR 2010-003, Department of Computer Science, University of Toronto, 2010.
  - [17] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
  - [18] <http://sir.unl.edu>.
  - [19] D. Binldey, “Source code analysis: a road map,” in *Proceedings of the Future of Software Engineering (FOSE '07)*, pp. 104–119, IEEE Computer Society, Minneapolis, Minn, USA, May 2007.

## Research Article

# Unitary Approximations in Fault Detection Filter Design

**Dušan Krokavec, Anna Filasová, and Pavol Liščinský**

*Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia*

Correspondence should be addressed to Dušan Krokavec; [dusan.krokavec@tuke.sk](mailto:dusan.krokavec@tuke.sk)

Received 19 January 2016; Revised 17 March 2016; Accepted 28 March 2016

Academic Editor: Zehui Mao

Copyright © 2016 Dušan Krokavec et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper is concerned with the fault detection filter design requirements that relax the existing conditions reported in the previous literature by adapting the unitary system principle in approximation of fault detection filter transfer function matrix for continuous-time linear MIMO systems. Conditions for the existence of a unitary construction are presented under which the fault detection filter with a unitary transfer function can be designed to provide high residual signals sensitivity with respect to faults. Otherwise, reflecting the emplacement of singular values in unitary construction principle, an associated structure of linear matrix inequalities with built-in constraints is outlined to design the fault detection filter only with a Hurwitz transfer function. All proposed design conditions are verified by the numerical illustrative examples.

## 1. Introduction

A conventional control for complex systems may result in unsatisfactory performances in the event of system component malfunctions. In order to remedy these weaknesses, different approaches to control system design are developed to tolerate component malfunctions and to maintain acceptable performances of the system with faults. The proposed control structures are known as fault-tolerant control (FTC) systems and they force the ability to accommodate component failures. In that sense, research in FTC is subject of a wide range of publications reflecting faults effect on control structure reconfiguration [1, 2] and fault estimation [3, 4], as well as fault residuals generation, analysis, and evaluation [5, 6]. The ideal approach would be to construct disturbance-decoupled residuals, with responsiveness and sensibility to the faults, as shown in [7].

To scale up accuracy of fault detection, it is eligible to craft residuals with high sensitivity to faults under robustness to disturbances. One of the options is the use of  $H_\infty/H_-$  optimization principle [8–11]. The restriction of this method is mainly the necessity of existence of a full rank direct-feed external gain matrix from faults to residuals [12], which limits them to be used only to residuals revealing actuator faults. One of the other methods, based on unitary system

properties, is proposed in [13, 14], where optimization is realized inherently in the sense that if the singular values of a unitary system are assigned as the magnitude frequency response of a first-order transfer function then strictly  $H_\infty$  norm is the maximum and  $H_-$  index is the minimum of the generalized gain value of the transfer function. It should be noted that such approximation does not always exist, the construction is not unique, and the proposed design algorithm, exploiting the Riccati equation based formulation, can be often infeasible. The remarks, which admitted more properties of  $H_\infty$  norm and  $H_-$  index, as well as other comments found to be very valuable in designing the unitary systems, are presented in Section 2.

Reflecting the basic results in unitary approximation of a square system transfer function matrix [13, 15], the principle was applied in [16] to design a unitary construction of the fault detection filter transfer function matrix. This technique, like that based on the classical procedure [14], is applicable to multivariable systems if the matching conditions are satisfied, that is, the unobservable modes of the used unitary approximation of the fault transform function matrix are stable. If these conditions are not satisfied, a feedback configuration of the fault detection filter, which optimizes sensitivity between the fault input and the output residuals, does not exist.

To the best of our knowledge, so far, no more results than those given above on application of unitary approximations in the fault detection filter design are available in the literature, probably because the standard progression may not have a solution. This motivates this study, reflecting the matrix parameters implying an unstable unitary construction of the fault detection filter, to reformulate the design conditions in such a way that the fault detection filter with a Hurwitz transfer function matrix can be designed. The result is an associated structure of linear matrix inequalities, reflecting the principle of singular values emplacement in unitary construction, the associated state, and output variable constraints as well as an additive stabilizing feedback in fault detection filter. In this sense the proposed method attempts to combine the unitary approximation principle with the singular value placement to obtain a simple but practical algorithm for designing suboptimal performances of the fault detection filter transfer function matrix, that is, for a particular combination of three matrix parameters designed by the proposed method to find a fault detection filter with acceptable sensitivity, which cannot be reduced below some limiting value merely by manipulating stability of the filter.

Searching for gain matrices of the fault detection filter state-space description and ensuring the unitary model gain matrix values for a stable fault detection filter, the proposed design task is with unitary conditions reaching approximately the theoretical limits for the prescribed set of singular values. To analyze stability of the observer-based fault detection filter, the proposed conditions use standard arguments and require to solve only LMIs with the built-in prescribed constraints depending on the system output and fault input matrix structures. Within unitary solutions, the results are similar with those obtained by the method proposed in [17], but with reducing the ranges of all problem variables as much as possible.

The paper is organized as follows. Placed immediately after Introduction, Section 2 presents the problem statement and Section 3 summarizes in the basic preliminaries the auxiliary lemmas on the issue of the design task. The enhanced structure of unitary fault detection filter transfer function matrix, as well as the sets of LMIs, reflecting the quadratic Lyapunov function to describe the filter stability, is theoretically explained in Section 4 and the structures of the fault detections filters are given in Section 5. Two examples are provided to demonstrate the proposed approach in Section 6 and, finally, Section 7 draws some concluding remarks.

Used notations are conventional so that  $\mathbf{x}^T$ ,  $\mathbf{X}^T$  denote transpose of the vector  $\mathbf{x}$  and matrix  $\mathbf{X}$ , respectively,  $\mathbf{X} = \mathbf{X}^T > 0$  means that  $\mathbf{X}$  is a symmetric positive definite matrix, the symbol  $\mathbf{I}_n$  marks the  $n$ th order unit matrix,  $\rho(\mathbf{X})$  and  $\text{rank}(\mathbf{X})$  indicate the eigenvalue spectrum and rank of a square matrix  $\mathbf{X}$ ,  $\mathbf{Y}^\perp$  designates the orthogonal complement to a rank-deficient matrix  $\mathbf{Y}$ ,  $\sigma_i(\mathbf{Z})$  labels the  $i$ th singular value of matrix  $\mathbf{Z}$ ,  $\mathbb{R}$  denotes the set of real numbers, and  $\mathbb{R}^n$ ,  $\mathbb{R}^{n \times r}$  refer to the set of all  $n$  dimensional real vectors and  $n \times r$  real matrices, respectively.

## 2. The Problem Statement

The systems under consideration are linear MIMO continuous-time dynamic systems represented as follows:

$$\begin{aligned}\dot{\mathbf{q}}(t) &= \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{f}(t) + \mathbf{E}\mathbf{d}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{q}(t),\end{aligned}\tag{1}$$

where  $\mathbf{q}(t) \in \mathbb{R}^n$ ,  $\mathbf{u}(t) \in \mathbb{R}^r$ , and  $\mathbf{y}(t) \in \mathbb{R}^m$  are vectors of the state, input, and output variables, respectively,  $\mathbf{f}(t) \in \mathbb{R}^p$  is fault vector, and  $\mathbf{d}(t) \in \mathbb{R}^{r_d}$  is vector of disturbance. The real matrices  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times r}$ ,  $\mathbf{C} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{F} \in \mathbb{R}^{n \times p}$ , and  $\mathbf{E} \in \mathbb{R}^{n \times r_d}$  are finite values, satisfying the rank conditions  $\text{rank}(\mathbf{F}) = p$ ,  $\text{rank}(\mathbf{C}) = m$ ,  $p = m$ , and  $p < n$ . Moreover, it is supposed that the matrix  $\mathbf{V} = \mathbf{C}\mathbf{F}$  is regular matrix such that  $\mathbf{V} \in \mathbb{R}^{m \times m}$ .

Problem of the interest is a unitary representation of the fault detection filter for the system with the square transfer function matrix of unknown fault input and the residuals. Note that such construction of unitary systems to given linear system, with respect to the singular values of the system transfer function matrix, is not a unique task also for square linear systems [13, 15].

## 3. Basic Preliminaries

*3.1. Linear Systems.* If  $\mathcal{H}$  and  $\mathcal{E}$  are Krein spaces [18], a continuous linear transformation

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} : \begin{array}{c} \mathcal{H} \\ \mathcal{E} \end{array} \oplus \longrightarrow \begin{array}{c} \mathcal{H} \\ \mathcal{E} \end{array} \oplus\tag{2}$$

is called the linear system. The underlying Krein space  $\mathcal{H}$  is called the state space and the auxiliary Krein space  $\mathcal{E}$  is called the coefficient space or the external space [19]. The transformation  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  is the main input and output transformation, respectively, and the operator  $\mathbf{D}$  is called the external operator.

The transfer function  $\mathbf{G}(s)$  of the linear system is defined by

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D},\tag{3}$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times r}$ ,  $\mathbf{C} \in \mathbb{R}^{m \times n}$ , and  $\mathbf{D} \in \mathbb{R}^{m \times r}$  are real matrices,  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$  is the identity matrix, and a complex number  $s$  is the transform variable (Laplace variable) of the Laplace transform [20].

The eigenvalues of  $\mathbf{A}$  are typically either real or complex-conjugate pairs. If  $\mathbf{A}$  has no imaginary eigenvalues then  $\mathbf{G}(j\omega)$  is defined for all  $\omega \in \mathbb{R}$ , where  $\omega$  is the frequency variable and  $j := \sqrt{-1}$ . The singular values of the transfer function matrix  $\mathbf{G}(s)$ , evaluated on the imaginary axis, are  $\sigma_i(\mathbf{G}(j\omega))$ , where the  $i$ th singular value of the complex matrix  $\mathbf{G}(j\omega)$  is the nonnegative square-root of the  $i$ th largest eigenvalue of  $\mathbf{G}(j\omega)\mathbf{G}^*(j\omega)$ , where  $\mathbf{G}^*(j\omega)$  is the adjoint of  $\mathbf{G}(j\omega)$ . It is usually assumed that the singular values are ordered such that  $\sigma_i \geq \sigma_{i+1}$ ,  $i = 1, 2, \dots, n-1$ .

Expressing the generalized gain of the system transfer function matrix  $\mathbf{G}(s)$  as the 2-norm ratio of the input and output vectors, then the maximum and minimum singular values of  $\mathbf{G}(j\omega)$  will constitute upper and lower bounds on this gain. The ratio between the maximum and minimum singular value is denoted by the condition number  $\kappa$ ; that is,

$$\kappa = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{\sigma_1}{\sigma_n}. \quad (4)$$

The condition number plotted versus frequency variable  $\omega$  outlines the system sensitivity to the direction of the input vector. If  $\kappa \gg 1$ , the generalized gain of the transfer function matrix will vary considerably with the input vector direction and  $\mathbf{G}(s)$  is said to be ill-conditioned. Conversely, if  $\kappa \approx 1$ , the generalized gain of the transfer function matrix will be insensitive to the input direction and the system is said to be well-conditioned [21].

The  $H_\infty$  norm of the transfer function matrix  $\mathbf{G}(s)$  is [22]

$$\|\mathbf{G}\|_\infty = \sup_{\omega \in \mathbb{R}} \sigma_1(\mathbf{G}(j\omega)). \quad (5)$$

The  $H_\infty$  norm expresses the maximum of generalized gain of the system transfer function matrix for a class of input signals characterized by their 2-norm [23].

The  $H_-$  index of the transfer function matrix  $\mathbf{G}(s)$  is defined as [8]

$$\|\mathbf{G}\|_- = \inf_{\omega \in \mathbb{R}} \sigma_n(\mathbf{G}(j\omega)). \quad (6)$$

The  $H_-$  index expresses the minimum of generalized gain of the system transfer function matrix for a class of input signals characterized by their 2-norm. Note that  $H_-$  index of a nonsquare system transfer function matrix is associated with rank of this matrix, that is, that  $H_-$  index is not completely dual to  $H_\infty$  norm [24, 25].

It is evident that using the singular values, a system is assessed in more detail. Some other reflections can be found, for example, in [26–29].

*Definition 1.* A stable linear time-invariant system of  $m$ -inputs and  $m$ -outputs (square system) is defined as a unitary system if the singular values of its transfer function (transfer function matrix)  $\mathbf{G}(s)$  satisfy [13]

$$\sigma_1 = \sigma_2 = \dots = \sigma_m, \quad (7)$$

where  $\sigma_i$  is the  $i$ th singular value of  $\mathbf{G}(s)$ .

*Definition 2* (see [30]). Let  $\mathbf{F} \in \mathbb{R}^{h \times h}$ ,  $\text{rank}(\mathbf{F}) = k < h$  be a rank-deficient matrix. Then the null space  $\mathbf{U}_F^T$  of  $\mathbf{F}$  is the orthogonal complement of the row space of  $\mathbf{F}$ . An orthogonal complement  $\mathbf{F}^\perp$  of  $\mathbf{F}$  is

$$\mathbf{F}^\perp = \mathbf{F}^\circ \mathbf{U}_E^T, \quad (8)$$

where  $\mathbf{F}^\circ$  is an arbitrary matrix of appropriate dimension.

Considering the regular matrix  $\mathbf{V} = \mathbf{CF}$ , the following state coordinate transformation of system (1) can be done.

**Lemma 3.** *If there are requirements for a regular matrix  $\mathbf{V}$ , defined as product of the matrix parameters  $\mathbf{C}$  and  $\mathbf{F}$  of the square system (1), to apply*

$$\begin{aligned} \mathbf{C} &= [\mathbf{V} \ \mathbf{0}] \mathbf{T}, \\ \mathbf{TF} &= \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix}, \\ \mathbf{V} &= \mathbf{CF}, \end{aligned} \quad (9)$$

then the transform matrix  $\mathbf{T} \in \mathbb{R}^{n \times n}$  takes the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{V}^{-1} \mathbf{C} \\ \mathbf{F}^\perp \end{bmatrix}, \quad (10)$$

where  $m = p$ ,  $\mathbf{V}^{-1} \mathbf{C} \in \mathbb{R}^{m \times n}$ , and  $\mathbf{F}^\perp \in \mathbb{R}^{(n-m) \times n}$ , respectively, and  $\mathbf{F}^\perp$  is the left orthogonal complement to  $\mathbf{F}$ .

*Proof.* Rewriting the first term of (9) as

$$\mathbf{C} = [\mathbf{V} \ \mathbf{0}] \mathbf{T} = [\mathbf{V} \ \mathbf{0}] \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix} = \mathbf{C}, \quad (11)$$

it is evident that

$$\mathbf{V} \mathbf{T}_1 = \mathbf{C}, \quad \mathbf{T}_1 = \mathbf{V}^{-1} \mathbf{C}. \quad (12)$$

Analyzing the second term of (9), that is,

$$\mathbf{TF} = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix} \mathbf{F} = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix}, \quad (13)$$

the following condition results:

$$\mathbf{T}_2 \mathbf{F} = \mathbf{0}, \quad \mathbf{T}_2 = \mathbf{F}^\perp. \quad (14)$$

Thus, (12) and (14) imply (10).

It is easily verified using (11) and (14) that

$$\mathbf{T}_1 \mathbf{F} = \mathbf{V}^{-1} \mathbf{CF} = \mathbf{V}^{-1} \mathbf{V} = \mathbf{I}_p, \quad (15)$$

$$\mathbf{CT}^{-1} = [\mathbf{V} \ \mathbf{0}], \quad (16)$$

respectively. This concludes the proof.  $\square$

**3.2. Structures of Unitary Fault Transfer Function Matrices.** The basic structure of the unitary fault transfer function matrix is introduced by the following lemma.

**Lemma 4.** *For system (1) with  $m = p$ , the transform matrix  $\mathbf{T}$  of the structure (10), and a prescribed positive scalar  $s_o \in \mathbb{R}$  there exists the matrix  $\mathbf{L}^o \in \mathbb{R}^{n \times m}$  such that the fault transfer function matrix can be approximated as*

$$\mathbf{G}_f(s) \approx \mathbf{G}_\Delta(s) \mathbf{G}_{f\Delta}(s), \quad (17)$$

where

$$\mathbf{G}_{f\Delta}(s) = \mathbf{C} (s\mathbf{I}_n - (\mathbf{A} - \mathbf{L}^o\mathbf{C}))^{-1} = \frac{\mathbf{V}}{s + s_o}, \quad (18)$$

$$\mathbf{G}_\Delta(s) = \mathbf{I}_m + \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{M}, \quad (19)$$

$$\mathbf{A}_o = \mathbf{TAT}^{-1} = \begin{bmatrix} \mathbf{A}_{o11} & \mathbf{A}_{o12} \\ \mathbf{A}_{o21} & \mathbf{A}_{o22} \end{bmatrix}, \quad (20)$$

$$\mathbf{L}^o = \begin{bmatrix} s_o\mathbf{I}_m + \mathbf{A}_{o11} \\ \mathbf{A}_{o21} \end{bmatrix}, \quad (21)$$

$$\mathbf{M} = \mathbf{T}^{-1}\mathbf{L}^o.$$

*Proof (compare [13, 16]).* Since

$$\mathbf{G}_f(s) = \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{F} \quad (22)$$

is the fault transfer function matrix of dimension  $m \times m$ , then (22) can be rewritten by using (9), (16), and (20) as

$$\begin{aligned} \mathbf{G}_f(s) &= \mathbf{CT}^{-1}\mathbf{T}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{T}^{-1}\mathbf{TF} \\ &= \mathbf{CT}^{-1}(\mathbf{T}(s\mathbf{I}_n - \mathbf{A})\mathbf{T}^{-1})^{-1}\mathbf{TF} \\ &= [\mathbf{V} \ \mathbf{0}] (s\mathbf{I}_n - \mathbf{A}_o)^{-1} \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0} \end{bmatrix}. \end{aligned} \quad (23)$$

Specifying the matrix product  $\mathbf{A}^o = \mathbf{TMCT}^{-1}$ , where  $\mathbf{M} \in \mathbb{R}^{n \times m}$  is a real matrix then, by exploiting (10) and (16), it yields

$$\begin{aligned} \mathbf{A}^o &= \mathbf{TMCT}^{-1} = \begin{bmatrix} \mathbf{V}^{-1}\mathbf{C} \\ \mathbf{F}^\perp \end{bmatrix} \mathbf{M} [\mathbf{V} \ \mathbf{0}] \\ &= \begin{bmatrix} \mathbf{V}^{-1}\mathbf{CMV} & \mathbf{0} \\ \mathbf{F}^\perp\mathbf{MV} & \mathbf{0} \end{bmatrix}, \end{aligned} \quad (24)$$

and accepting the block matrix structure of (21) and (24), it can define

$$\Delta\mathbf{A}_o = \mathbf{A}_o - \mathbf{A}^o = \begin{bmatrix} \mathbf{A}_{o11} - \mathbf{V}^{-1}\mathbf{CMV} & \mathbf{A}_{o12} \\ \mathbf{A}_{o21} - \mathbf{F}^\perp\mathbf{MV} & \mathbf{A}_{o22} \end{bmatrix}. \quad (25)$$

Setting

$$\begin{aligned} \mathbf{A}_{o11} - \mathbf{V}^{-1}\mathbf{CMV} &= -s_o\mathbf{I}_m, \\ \mathbf{A}_{o21} - \mathbf{F}^\perp\mathbf{MV} &= \mathbf{0}, \end{aligned} \quad (26)$$

where  $s_o \in \mathbb{R}$  is a prescribed positive real value, and rewriting (26) as

$$\begin{bmatrix} s_o\mathbf{I}_m + \mathbf{A}_{o11} \\ \mathbf{A}_{o21} \end{bmatrix} = \begin{bmatrix} \mathbf{V}^{-1}\mathbf{C} \\ \mathbf{F}^\perp \end{bmatrix} \mathbf{MV} = \mathbf{TMV}, \quad (27)$$

then, with

$$\mathbf{M} = \mathbf{T}^{-1}\mathbf{L}^o, \quad (28)$$

it is

$$\begin{bmatrix} s_o\mathbf{I}_m + \mathbf{A}_{o11} \\ \mathbf{A}_{o21} \end{bmatrix} = \mathbf{TT}^{-1}\mathbf{L}^o\mathbf{V} = \mathbf{L}^o\mathbf{V}. \quad (29)$$

Moreover, for (24) and (25), the following yields

$$\mathbf{A}^o = \mathbf{TMCT}^{-1} = \mathbf{L}^o\mathbf{CT}^{-1}, \quad (30)$$

$$\Delta\mathbf{A}_o = \mathbf{A}_o - \mathbf{A}^o = \mathbf{T}(\mathbf{A} - \mathbf{MC})\mathbf{T}^{-1} = \mathbf{T}\Delta\mathbf{AT}^{-1},$$

where

$$\Delta\mathbf{A} = \mathbf{A} - \mathbf{MC} = \mathbf{A} - \mathbf{T}^{-1}\mathbf{L}^o\mathbf{C}, \quad (31)$$

and (25) takes the form

$$\Delta\mathbf{A}_o = \begin{bmatrix} -s_o\mathbf{I}_m & \mathbf{A}_{o12} \\ \mathbf{0} & \mathbf{A}_{o22} \end{bmatrix}. \quad (32)$$

Defining the transfer function matrix  $\mathbf{G}_{f\Delta}(s)$  as

$$\mathbf{G}_{f\Delta}(s) = \mathbf{C} (s\mathbf{I}_n - \Delta\mathbf{A})^{-1} \mathbf{F}, \quad (33)$$

then with (32) it is

$$\begin{aligned} \mathbf{G}_{f\Delta}(s) &= \mathbf{CT}^{-1} (s\mathbf{I}_n - \mathbf{T}\Delta\mathbf{AT}^{-1})^{-1} \mathbf{TF} \\ &= [\mathbf{V} \ \mathbf{0}] (s\mathbf{I}_n - \Delta\mathbf{A}_o)^{-1} \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0} \end{bmatrix}. \end{aligned} \quad (34)$$

Since

$$s\mathbf{I}_n - \Delta\mathbf{A}_o = \begin{bmatrix} (s + s_o)\mathbf{I}_m & -\mathbf{A}_{o12} \\ \mathbf{0} & s\mathbf{I}_{n-m} - \mathbf{A}_{o22} \end{bmatrix}, \quad (35)$$

$$\begin{aligned} (s\mathbf{I}_n - \Delta\mathbf{A}_o)^{-1} &= \begin{bmatrix} (s + s_o)^{-1}\mathbf{I}_m & (s + s_o)^{-1}\mathbf{A}_{o12}(s\mathbf{I}_{n-m} - \mathbf{A}_{o22})^{-1} \\ \mathbf{0} & (s\mathbf{I}_{n-m} - \mathbf{A}_{o22})^{-1} \end{bmatrix}, \end{aligned} \quad (36)$$

substituting (36) into (34) it can obtain

$$\mathbf{G}_{f\Delta}(s) = [\mathbf{V} \ \mathbf{0}] (s\mathbf{I}_n - \Delta\mathbf{A}_o)^{-1} \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0} \end{bmatrix} = \frac{\mathbf{V}}{s + s_o}, \quad (37)$$

which implies (18).

The transfer function (33) together with (31) and (22) can be written the way that

$$\begin{aligned} \mathbf{G}_{f\Delta}(s) &= \mathbf{C} (s\mathbf{I}_n - \Delta\mathbf{A})^{-1} \mathbf{F} \\ &= \mathbf{C} (s\mathbf{I}_n - (\mathbf{A} - \mathbf{MC}))^{-1} \mathbf{F} \\ &= \mathbf{C} ((s\mathbf{I}_n - \mathbf{A})(\mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{MC}))^{-1} \mathbf{F} \\ &= \mathbf{C} (\mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{MC})^{-1} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{F}, \end{aligned} \quad (38)$$

which gives, by using the equality

$$(\mathbf{FC})^{-1} \mathbf{FC} = \mathbf{I}_n, \quad (39)$$

the expression for  $\mathbf{G}_{f\Delta}(s)$  as follows:

$$\begin{aligned} \mathbf{G}_{f\Delta}(s) &= \mathbf{C} \left( \mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{MC} \right)^{-1} (\mathbf{FC})^{-1} \\ &\cdot \mathbf{FC} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{F} = \mathbf{C} \left( \mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{MC} \right)^{-1} \\ &\cdot (\mathbf{FC})^{-1} \mathbf{F} \mathbf{G}_f(s), \end{aligned} \quad (40)$$

where  $\mathbf{G}_f(s)$ , introduced in (22), is the fault transfer function matrix.

After some manipulations,

$$\begin{aligned} \mathbf{G}_{f\Delta}(s) &= \mathbf{C} \left( (\mathbf{FC}) \left( \mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{MC} \right) \right)^{-1} \mathbf{F} \mathbf{G}_f(s) \\ &= \mathbf{C} \left( \mathbf{F} \left( \mathbf{I}_m + \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{M} \right) \mathbf{C} \right)^{-1} \mathbf{F} \mathbf{G}_f(s), \end{aligned} \quad (41)$$

and  $\mathbf{G}_{f\Delta}(s)$  can be approximated as

$$\mathbf{G}_{f\Delta}(s) \approx \left( \mathbf{I}_m + \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{M} \right)^{-1} \mathbf{G}_f(s). \quad (42)$$

Thus, using (19) and then (41) implies (17). This concludes the proof.  $\square$

With existence of such transformation, the structure of (24) really means that there exist the subset of transformed state variables whose dynamics is explicitly affected by the fault  $\mathbf{f}(t)$  and a second one, whose dynamics is not affected explicitly by the fault  $\mathbf{f}(t)$ .

*Remark 5.* It is important to note the fact that the eigenvalues of  $\mathbf{A}$  and of  $\mathbf{A}_o$  are the same whenever  $\mathbf{A}_o$  is related to  $\mathbf{A}$  as  $\mathbf{A}_o = \mathbf{TAT}^{-1}$  for any invertible  $\mathbf{T}$ , as it is defined in (20) [31]. But this does not mean that if eigenvalues of the matrix  $\mathbf{A}_o$  are stable then eigenvalues of the matrix  $\mathbf{A}_{o22}$  are also stable. This is a limitation of the methodology based on (32) and for a stable system it can lead to an unstable structure (35). It requires an additional stabilization, but this stabilization generally violates the desired unitary form of the fault transfer function matrix.

Defined by (7), a linear time-invariant system is considered as unitary if all singular values of its transfer function matrix are equal. Because the construction given in Lemma 4 is not unique, some equivalent structures can be used. One is introduced by the following lemma.

**Lemma 6.** *An equivalent structure of the fault transfer function matrix of system (1) takes the form*

$$\mathbf{G}_f(s) \approx \mathbf{G}_\Delta^\circ(s) \left( \mathbf{G}_{f\Delta}^\circ(s) - \mathbf{V} \right), \quad (43)$$

where, for a matrix  $\mathbf{N} \in \mathbb{R}^{n \times m}$ ,

$$\mathbf{G}_{f\Delta}^\circ(s) = \mathbf{C} (s\mathbf{I}_n - (\mathbf{A} - \mathbf{NC}))^{-1} (\mathbf{F} + \mathbf{NV}) + \mathbf{V}, \quad (44)$$

$$\mathbf{G}_\Delta^\circ(s) = \mathbf{I}_m + \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{N}. \quad (45)$$

*Proof (compare [15]).* Considering the associated system (44), it can be written for the resolvent matrix of a matrix  $(\mathbf{A} - \mathbf{NC})$  that

$$\begin{aligned} &(s\mathbf{I}_n - (\mathbf{A} - \mathbf{NC}))^{-1} \\ &= \left( (s\mathbf{I}_n - \mathbf{A}) \left( \mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{NC} \right) \right)^{-1} \\ &= \mathbf{G}_\Delta^\diamond(s) (s\mathbf{I}_n - \mathbf{A})^{-1}, \end{aligned} \quad (46)$$

where

$$\mathbf{G}_\Delta^\diamond(s) = \left( \mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{NC} \right)^{-1}. \quad (47)$$

Therefore, the substitution of (9) and (46) in (44) leads to

$$\begin{aligned} \mathbf{G}_{f\Delta}^\circ(s) &= \mathbf{C} \mathbf{G}_\Delta^\diamond(s) (s\mathbf{I}_n - \mathbf{A})^{-1} (\mathbf{F} + \mathbf{NV}) + \mathbf{V} \\ &= \mathbf{C} \left( \mathbf{G}_\Delta^\diamond(s) (s\mathbf{I}_n - \mathbf{A})^{-1} (\mathbf{I}_n + \mathbf{NC}) + \mathbf{I}_n \right) \mathbf{F}, \end{aligned} \quad (48)$$

and it yields with (47)

$$\begin{aligned} \mathbf{G}_{f\Delta}^\circ(s) &= \mathbf{C} \left( \mathbf{G}_\Delta^\diamond(s) \left( (s\mathbf{I}_n - \mathbf{A})^{-1} + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{NC} \right) + \mathbf{I}_n \right) \mathbf{F} \\ &= \mathbf{C} \left( \mathbf{G}_\Delta^\diamond(s) \left( (s\mathbf{I}_n - \mathbf{A})^{-1} + \mathbf{G}_\Delta^{\diamond-1}(s) \right) \right) \mathbf{F} \\ &= \mathbf{C} \mathbf{G}_\Delta^\diamond(s) (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{F} + \mathbf{CF}. \end{aligned} \quad (49)$$

Since using equality (39), it can obtain

$$\begin{aligned} \mathbf{C} \mathbf{G}_\Delta^\diamond(s) &= \mathbf{C} \mathbf{G}_\Delta^\diamond(s) (\mathbf{FC})^{-1} \mathbf{FC} \\ &= \mathbf{C} \left( \mathbf{FC} \left( \mathbf{I}_n + (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{NC} \right) \right)^{-1} \mathbf{FC} \\ &= \mathbf{C} \left( \mathbf{F} \left( \mathbf{I}_m + \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{N} \right) \mathbf{C} \right)^{-1} \mathbf{FC}, \end{aligned} \quad (50)$$

and  $\mathbf{C} \mathbf{G}_\Delta^\diamond(s)$  can be approximated as

$$\mathbf{C} \mathbf{G}_\Delta^\diamond(s) \approx \left( \mathbf{I}_m + \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{N} \right)^{-1} \mathbf{C} = \mathbf{G}_\Delta^{\circ-1}(s) \mathbf{C}, \quad (51)$$

and then the substitution of (51) into (49) gives

$$\begin{aligned} \mathbf{G}_{f\Delta}^\circ(s) &\approx \mathbf{G}_\Delta^{\circ-1}(s) \mathbf{C} (s\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{F} + \mathbf{CF} \\ &= \mathbf{G}_\Delta^{\circ-1}(s) \mathbf{G}_f(s) + \mathbf{V}. \end{aligned} \quad (52)$$

Thus, (52) implies (43). This concludes the proof.  $\square$

**Corollary 7.** *Considering that  $\mathbf{M} = \mathbf{N}$ , (19) and (45) imply  $\mathbf{G}_\Delta^\circ(s) = \mathbf{G}_\Delta(s)$ . Since (17) defines  $\mathbf{G}_f(s) = \mathbf{G}_\Delta(s) \mathbf{G}_{f\Delta}(s)$ , it has to be satisfied with respect to (43) that*

$$\mathbf{G}_f(s) = \mathbf{G}_\Delta(s) \left( \mathbf{G}_{f\Delta}^\circ(s) - \mathbf{V} \right) = \mathbf{G}_\Delta(s) \mathbf{G}_{f\Delta}(s), \quad (53)$$

which gives, with (18),

$$\begin{aligned} \mathbf{G}_{f\Delta}^\circ(s) &= \mathbf{G}_{f\Delta}(s) + \mathbf{V} = \mathbf{V} \left( \frac{1}{s + s_o} + 1 \right) \\ &= \frac{\mathbf{V}}{s + s_o} (s + s_o + 1); \end{aligned} \quad (54)$$

that is

$$\frac{\mathbf{G}_{f\Delta}^{\circ}(s)}{s + s_o + 1} = \frac{\mathbf{V}}{s + s_o} = \mathbf{G}_{f\Delta}(s). \quad (55)$$

This corollary gives the possibility to combine the results of Lemmas 4 and 6 in the design of unitary fault transfer function matrix by the way specified in the following section.

#### 4. State-Space Description of Enhanced Structures

To exploit the properties of the structures presented above, the enhanced form of unitary fault transfer matrix is proposed in the form

$$\mathbf{G}_f^{\circ}(s) = \mathbf{C}(s\mathbf{I}_n - \mathbf{A} + (\mathbf{M} - \mathbf{N})\mathbf{C})^{-1}\mathbf{V}, \quad (56)$$

where  $\mathbf{M}$  is introduced in (20) and  $\mathbf{N}$  is designed in such a way that  $\mathbf{F} + \mathbf{N}\mathbf{V} = \mathbf{0}$ .

To formulate the stability condition of the unitary system, approximated by the equivalent transform function matrix (56), the following theorems are given.

**Theorem 8.** *The state-space representation of the enhanced structure of transfer function matrix (56) in the form of a closed-loop system is*

$$\dot{\mathbf{q}}^{\circ}(t) = (\mathbf{A} - \mathbf{J}\mathbf{C})\mathbf{q}^{\circ}(t) + \mathbf{V}\mathbf{f}(t), \quad (57)$$

$$\mathbf{y}^{\circ}(t) = \mathbf{C}\mathbf{q}^{\circ}(t), \quad (58)$$

$$\mathbf{w}^{\circ}(t) = \mathbf{F}^T\mathbf{q}^{\circ}(t), \quad (59)$$

where  $\mathbf{w}^{\circ}(t)$  is the performance evaluation signal,

$$\mathbf{J} = \mathbf{M} - \mathbf{N}, \quad \mathbf{N} = -\mathbf{F}\mathbf{V}^{-1}, \quad \mathbf{M} = \mathbf{T}^{-1}\mathbf{L}^o, \quad (60)$$

and the system constraint is

$$\mathbf{R}^{\circ} = \begin{bmatrix} \mathbf{0} & \mathbf{S} \\ \mathbf{S}^T & -\mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{F}\mathbf{V}^T \\ \mathbf{V}\mathbf{F}^T & -\mathbf{V}\mathbf{V}^T \end{bmatrix} \geq \mathbf{0}. \quad (61)$$

*Proof.* Use the Laplace transform property [32]; then (57) with the zero state vector initial condition implies

$$(s\mathbf{I}_n - \mathbf{A} + \mathbf{J}\mathbf{C})\tilde{\mathbf{q}}^{\circ}(s) = \mathbf{V}\tilde{\mathbf{f}}^{\circ}(s), \quad (62)$$

$$\tilde{\mathbf{q}}^{\circ}(s) = (s\mathbf{I}_n - \mathbf{A} + (\mathbf{M} - \mathbf{N})\mathbf{C})^{-1}\mathbf{V}\tilde{\mathbf{f}}^{\circ}(s), \quad (63)$$

respectively, and

$$\tilde{\mathbf{y}}^{\circ}(s) = \mathbf{C}(s\mathbf{I}_n - \mathbf{A} + (\mathbf{M} - \mathbf{N})\mathbf{C})^{-1}\mathbf{V}\tilde{\mathbf{f}}^{\circ}(s), \quad (64)$$

while (64) implies (56).

Considering that

$$\mathbf{F} + \mathbf{N}\mathbf{V} = \mathbf{0}, \quad (65)$$

premultiplying the right side of (65) by  $\mathbf{V}^T$  leads to

$$\mathbf{F}\mathbf{V}^T + \mathbf{N}\mathbf{V}\mathbf{V}^T = \mathbf{0}, \quad (66)$$

which implies that

$$\mathbf{N} = -\mathbf{F}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T)^{-1}. \quad (67)$$

Since  $\mathbf{V}$  is a regular matrix, postmultiplying the right side of (67) by  $\mathbf{V}\mathbf{F}^T$  gives

$$\mathbf{F}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{F}^T = \mathbf{F}\mathbf{F}^T = -\mathbf{N}\mathbf{V}\mathbf{F}^T \geq \mathbf{0}. \quad (68)$$

Thus, (59) implies

$$\begin{aligned} \mathbf{w}^{\circ T}(t)\mathbf{w}^{\circ}(t) &= \mathbf{q}^{\circ T}(t)\mathbf{F}^T\mathbf{F}\mathbf{q}^{\circ}(t) \\ &= \mathbf{q}^{\circ T}(t)\mathbf{F}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{F}^T\mathbf{q}^{\circ}(t); \end{aligned} \quad (69)$$

it is evident that (69) is the constraint given on  $\mathbf{q}^{\circ}(t)$  and using the Schur complement property (68) implies quadratic constraint (61).

Finally, since  $\mathbf{V}$  is regular, (65) implies (60). This concludes the proof.  $\square$

**Theorem 9.** *The equivalent system (57) and (58) is stable if there exists a symmetric positive definite matrix  $\mathbf{P} \in \mathbb{R}^{n \times n}$  such that*

$$\mathbf{P} = \mathbf{P}^T > \mathbf{0}, \quad (70)$$

$$\begin{bmatrix} \mathbf{P}(\mathbf{A} - \mathbf{J}\mathbf{C}) + (\mathbf{A} - \mathbf{J}\mathbf{C})^T\mathbf{P} & \mathbf{F}\mathbf{V}^T \\ \mathbf{V}\mathbf{F}^T & -\mathbf{V}\mathbf{V}^T \end{bmatrix} < \mathbf{0}, \quad (71)$$

and the common gain matrix is given in (60).

*Proof.* Since the Lyapunov function candidate can be considered in the form

$$v(\mathbf{q}^{\circ}(t)) = \mathbf{q}^{\circ T}(t)\mathbf{P}\mathbf{q}^{\circ}(t) + \int_0^t \mathbf{w}^{\circ T}(v)\mathbf{w}^{\circ}(v)dv > 0, \quad (72)$$

where  $\mathbf{P} \in \mathbb{R}^{n \times n}$  is a symmetric, positive, and definite matrix, then the time derivative of (72) can be written as

$$\begin{aligned} \dot{v}(\mathbf{q}^{\circ}(t)) &= \dot{\mathbf{q}}^{\circ T}(t)\mathbf{P}\mathbf{q}^{\circ}(t) + \mathbf{q}^{\circ T}(t)\mathbf{P}\dot{\mathbf{q}}^{\circ}(t) \\ &\quad + \mathbf{w}^{\circ T}(t)\mathbf{w}^{\circ}(t) < 0. \end{aligned} \quad (73)$$

Substituting (69), as well as (57) for the fault-free regime, into (73), then inequality (73) can be rewritten as

$$\begin{aligned} \dot{v}(\mathbf{q}^{\circ}(t)) &= \mathbf{q}^{\circ T}(t)(\mathbf{A} - \mathbf{J}\mathbf{C})^T\mathbf{P}\mathbf{q}^{\circ}(t) \\ &\quad + \mathbf{q}^{\circ T}(t)\mathbf{P}(\mathbf{A} - \mathbf{J}\mathbf{C})\mathbf{q}^{\circ}(t) \\ &\quad + \mathbf{q}^{\circ T}(t)\mathbf{F}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{F}^T\mathbf{q}^{\circ}(t) < 0, \end{aligned} \quad (74)$$

which implies

$$(\mathbf{A} - \mathbf{J}\mathbf{C})^T\mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{J}\mathbf{C}) + \mathbf{F}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{F}^T < \mathbf{0}. \quad (75)$$

Thus, using the Schur complement, (75) implies (71). This concludes the proof.  $\square$

If the set of eigenvalues of  $\mathbf{A}_{o22}$  contains an unstable eigenvalue, the conditions have to be extended to design a stable fault detection filter (a fault detection filter with the Hurwitz transfer function [33]). Because the matrix block  $\mathbf{A}_{o22}$  is unstable and unobservable in the form of the state-space description (57), the synthesis of an additional observer gain is required to use a dual form, considering that the couple  $(\mathbf{A}, \mathbf{C})$  is observable. Moreover, to obtain sufficient dynamic range of residual signals, in the solution could be included system output constraints.

**Theorem 10.** *The equivalent system (57) and (58) with unstable matrix block  $\mathbf{A}_{o22}$  is stabilizable if there exists a symmetric positive definite matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$ , a regular matrix  $\mathbf{S} \in \mathbb{R}^{m \times m}$ , and a matrix  $\mathbf{U} \in \mathbb{R}^{n \times m}$  such that*

$$\mathbf{Q} = \mathbf{Q}^T > 0, \quad (76)$$

$$\begin{bmatrix} \mathbf{Q}(\mathbf{A} - \mathbf{J}\mathbf{C})^T + (\mathbf{A} - \mathbf{J}\mathbf{C})\mathbf{Q} - \mathbf{U}\mathbf{C} - \mathbf{C}^T\mathbf{U}^T & \mathbf{Q}\mathbf{C}^T + \mathbf{F}\mathbf{V}^T \\ \mathbf{C}\mathbf{Q} + \mathbf{V}\mathbf{F}^T & -\mathbf{V}\mathbf{V}^T \end{bmatrix} \quad (77)$$

< 0,

$$\mathbf{C}\mathbf{Q} = \mathbf{S}\mathbf{C}, \quad (78)$$

where the matrix  $\mathbf{J}$  is given in (60).

If (76)–(78) are admissible, then

$$\begin{aligned} \mathbf{H} &= \mathbf{U}\mathbf{S}^{-1}, \\ \mathbf{J}^\circ &= \mathbf{J} + \mathbf{H}, \end{aligned} \quad (79)$$

$$\mathbf{G}_f^\circ(s) = \mathbf{C}(s\mathbf{I}_n - \mathbf{A} + \mathbf{J}^\circ\mathbf{C})^{-1}\mathbf{V}.$$

*Proof.* Writing autonomous, fault-free free model (1) in the dual state-space form [34] as

$$\dot{\mathbf{p}}(t) = \mathbf{A}^T\mathbf{p}(t) + \mathbf{C}^T\mathbf{d}(t), \quad (80)$$

$$\mathbf{y}(t) = \mathbf{E}^T\mathbf{p}(t), \quad (81)$$

and then considering the Lyapunov function candidate of the form

$$v(\mathbf{p}(t)) = \mathbf{p}^T(t)\mathbf{Q}\mathbf{p}(t) > 0, \quad (82)$$

where  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  is a positive definite matrix, it has to be

$$\dot{v}(\mathbf{p}(t)) = \dot{\mathbf{p}}^T(t)\mathbf{Q}\mathbf{p}(t) + \mathbf{p}^T(t)\mathbf{Q}\dot{\mathbf{p}}(t) < 0, \quad (83)$$

and by substituting (80) into (83) it can obtain

$$\begin{aligned} \dot{v}(\mathbf{p}(t)) &= (\mathbf{A}^T\mathbf{p}(t) + \mathbf{C}^T\mathbf{d}(t))^T\mathbf{Q}\mathbf{p}(t) \\ &+ \mathbf{p}^T(t)\mathbf{Q}(\mathbf{A}^T\mathbf{p}(t) + \mathbf{C}^T\mathbf{d}(t)) < 0. \end{aligned} \quad (84)$$

Introducing the notation

$$\mathbf{p}^{\circ T}(t) = [\mathbf{p}^T(t) \ \mathbf{d}^T(t)], \quad (85)$$

then (84) can be redefined using the Krasovskii theorem (see, e.g., [35]), as

$$\dot{v}(\mathbf{p}^\circ(t)) = \mathbf{p}^{\circ T}(t)\mathbf{Q}^\circ\mathbf{p}^\circ(t) \leq -\mathbf{p}^{\circ T}(t)\mathbf{R}^\circ\mathbf{p}^\circ(t) < 0, \quad (86)$$

where  $\mathbf{R}^\circ$  is given in (61). Then (86) implies

$$\dot{v}(\mathbf{p}^\circ(t)) = \mathbf{p}^{\circ T}(t)(\mathbf{Q}^\circ + \mathbf{R}^\circ)\mathbf{p}^\circ(t) < 0, \quad (87)$$

where

$$\mathbf{Q}^\circ + \mathbf{R}^\circ = \begin{bmatrix} \mathbf{Q}\mathbf{A}^T + \mathbf{A}\mathbf{Q} & \mathbf{Q}\mathbf{C}^T + \mathbf{F}\mathbf{V}^T \\ \mathbf{C}\mathbf{Q} + \mathbf{V}\mathbf{F}^T & -\mathbf{V}\mathbf{V}^T \end{bmatrix} < 0. \quad (88)$$

Inserting  $\mathbf{A} \leftarrow (\mathbf{A} - \mathbf{J}\mathbf{C} - \mathbf{H}\mathbf{C})$ , where  $\mathbf{H} \in \mathbb{R}^{n \times n}$  is an additive observer gain then

$$\begin{bmatrix} \mathbf{Q}(\mathbf{A} - \mathbf{J}\mathbf{C} - \mathbf{H}\mathbf{C})^T + (\mathbf{A} - \mathbf{J}\mathbf{C} - \mathbf{H}\mathbf{C})\mathbf{Q} & \mathbf{Q}\mathbf{C}^T + \mathbf{F}\mathbf{V}^T \\ \mathbf{C}\mathbf{Q} + \mathbf{V}\mathbf{F}^T & -\mathbf{V}\mathbf{V}^T \end{bmatrix} < 0. \quad (89)$$

Setting

$$\mathbf{H}\mathbf{C}\mathbf{Q} = \mathbf{H}\mathbf{S}\mathbf{S}^{-1}\mathbf{C}\mathbf{Q} = \mathbf{U}\mathbf{C}\mathbf{Q}^{-1}\mathbf{Q} = \mathbf{U}\mathbf{C}, \quad (90)$$

where  $\mathbf{S} \in \mathbb{R}^{m \times m}$  is a regular matrix and

$$\begin{aligned} \mathbf{U} &= \mathbf{H}\mathbf{S}, \\ \mathbf{S}^{-1}\mathbf{C} &= \mathbf{C}\mathbf{Q}^{-1}, \end{aligned} \quad (91)$$

then (89) implies (77) and (91) gives (78). This concludes the proof.  $\square$

## 5. Fault Detection Filters

Denoting

$$\begin{aligned} \mathbf{e}(t) &= \mathbf{q}^\circ(t), \\ \mathbf{J} &= \mathbf{M} - \mathbf{N}, \end{aligned} \quad (92)$$

$$(\mathbf{J} = \mathbf{M} - \mathbf{N} + \mathbf{H}),$$

it yields in the fault-free regime

$$\dot{\mathbf{e}}(t) = (\mathbf{A} - \mathbf{J}\mathbf{C})\mathbf{e}(t), \quad (93)$$

which is the autonomous equation of the estimation error of Luenberger observer to the nominal noise and fault-free system (1), defined in the form [4]

$$\dot{\mathbf{q}}_e(t) = \mathbf{A}\mathbf{q}_e(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{J}\mathbf{C}(\mathbf{q}(t) - \mathbf{q}_e(t)). \quad (94)$$

Introducing  $\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}_e(t)$ , then with (1) and (94) it yields

$$\begin{aligned} \dot{\mathbf{e}}(t) &= \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{f}(t) + \mathbf{E}\mathbf{d}(t) - \mathbf{A}\mathbf{q}_e(t) \\ &- \mathbf{B}\mathbf{u}(t) - \mathbf{J}\mathbf{C}(\mathbf{q}(t) - \mathbf{q}_e(t)) \end{aligned} \quad (95)$$

$$= (\mathbf{A} - \mathbf{J}\mathbf{C})\mathbf{e}(t) + \mathbf{F}\mathbf{f}(t) + \mathbf{E}\mathbf{d}(t),$$

$$\tilde{\mathbf{e}}(s) = (s\mathbf{I}_n - (\mathbf{A} - \mathbf{J}\mathbf{C}))^{-1}(\mathbf{F}\tilde{\mathbf{f}}(s) + \mathbf{E}\tilde{\mathbf{d}}(s)), \quad (96)$$

respectively, where  $\tilde{\mathbf{e}}(s)$ ,  $\tilde{\mathbf{f}}(s)$ , and  $\tilde{\mathbf{d}}(s)$  stand for the Laplace transform of the  $n$  dimensional observer error vector, the  $m$  dimensional fault vector, and the  $p$  dimensional disturbance vector.

Designing the fault residuals as

$$\begin{aligned}\mathbf{r}(t) &= \mathbf{V}^{-1}\mathbf{C}\mathbf{e}(t), \\ \tilde{\mathbf{r}}(s) &= \mathbf{V}^{-1}\mathbf{C}\tilde{\mathbf{e}}(s),\end{aligned}\quad (97)$$

then the fault detection filter transfer function matrices of the fault and the disturbance are

$$\begin{aligned}\mathbf{G}_f(s) &= \mathbf{V}^{-1}\mathbf{C}(s\mathbf{I}_n - (\mathbf{A} - \mathbf{J}\mathbf{C}))^{-1}\mathbf{F}, \\ \mathbf{G}_d(s) &= \mathbf{V}^{-1}\mathbf{C}(s\mathbf{I}_n - (\mathbf{A} - \mathbf{J}\mathbf{C}))^{-1}\mathbf{E}.\end{aligned}\quad (98)$$

It is evident that, with  $\mathbf{J}$  of the structure (60),  $\mathbf{G}_f(s)$  is a unitary transfer function matrix with optimized singular values related properties.

## 6. Illustrative Examples

To illustrate the design principles, two examples are presented. First, Theorems 9 and 10 are applied to study the unitary fault detection transfer function matrix design problems in Example 1. Then, considering a real world model of the chemical reactor [36], the directional residuals properties are demonstrated in Example 2.

*Example 1.* The linear system, represented by model (1), is with the matrix parameters

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} -1.5780 & -3.1660 & -2.1392 & -1.1056 \\ 1.5478 & -1.5604 & -1.5612 & -5.9870 \\ 3.6630 & 2.5695 & -1.5962 & -5.6508 \\ 0.0947 & 4.9043 & 6.7035 & -1.4961 \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} 0.0000 & -0.7674 \\ -1.5953 & -0.6395 \\ -1.5404 & -0.6380 \\ 1.0513 & 0.8971 \end{bmatrix}, \\ \mathbf{F} &= \begin{bmatrix} 0.0000 & -0.7674 \\ -1.5953 & -0.6395 \\ \pm 1.5404 & -0.6380 \\ 1.0513 & 0.8971 \end{bmatrix}, \\ \mathbf{E} &= \begin{bmatrix} -1.5780 \\ 1.5478 \\ 3.6630 \\ 0.0947 \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} 0.0000 & -1.8600 & 0.0000 & 1.9003 \\ 0.4970 & 0.0000 & -1.8805 & 0.0000 \end{bmatrix}.\end{aligned}\quad (99)$$

Considering the signum (+) in the (3, 1) matrix element of  $\mathbf{F}$  then

$$\mathbf{V} = \mathbf{C}\mathbf{F} = \begin{bmatrix} 4.9650 & 2.8942 \\ -2.8967 & 0.8184 \end{bmatrix}, \quad \det(\mathbf{V}) \neq 0, \quad (100)$$

and the parameters of the matrix  $\mathbf{T}$  are computed as follows:

$$\begin{aligned}\mathbf{V}^{-1}\mathbf{C} &= \begin{bmatrix} -0.1156 & -0.1223 & 0.4373 & 0.1249 \\ 0.1983 & -0.4329 & -0.7501 & 0.4422 \end{bmatrix}, \\ \mathbf{F}^\perp &= \begin{bmatrix} -0.8366 & 0.2595 & 0.4248 & -0.2286 \\ 0.1130 & 0.6631 & 0.2007 & 0.7122 \end{bmatrix}, \\ \mathbf{T} &= \begin{bmatrix} -0.1156 & -0.1223 & 0.4373 & 0.1249 \\ 0.1983 & -0.4329 & -0.7501 & 0.4422 \\ -0.8366 & 0.2595 & 0.4248 & -0.2286 \\ 0.1130 & 0.6631 & 0.2007 & 0.7122 \end{bmatrix}.\end{aligned}\quad (101)$$

Computing (20) and separating the blocks of the matrix  $\mathbf{A}^\circ$  give the results

$$\begin{aligned}\mathbf{A}_{o11} &= \begin{bmatrix} -4.6582 & -4.6807 \\ 12.5921 & 5.1671 \end{bmatrix}, \\ \mathbf{A}_{o12} &= \begin{bmatrix} -2.3248 & 0.4532 \\ 4.4851 & 4.3760 \end{bmatrix}, \\ \mathbf{A}_{o21} &= \begin{bmatrix} -7.6293 & -5.7984 \\ -5.8979 & -10.6135 \end{bmatrix}, \\ \mathbf{A}_{o22} &= \begin{bmatrix} -4.2384 & -0.1943 \\ -4.0113 & -2.5012 \end{bmatrix},\end{aligned}\quad (102)$$

where the stable eigenvalue spectrum

$$\rho(\mathbf{A}_{o22}) = \{-4.6082, -2.1314\} \quad (103)$$

gives the possibility to obtain a unitary fault detection filter. Thus, choosing  $s_o = 5$ , it is obtained using (20) and (60) that

$$\begin{aligned}\mathbf{L}^\circ &= \begin{bmatrix} 0.3418 & -4.6807 \\ 12.5921 & 10.1671 \\ -7.6293 & -5.7984 \\ -5.8979 & -10.6135 \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} -0.0142 & -0.2290 \\ -2.7392 & 0.2023 \\ -3.0371 & -3.5498 \\ -0.6039 & -3.1706 \end{bmatrix},\end{aligned}$$

$$\mathbf{N} = \begin{bmatrix} 0.1786 & 0.3061 \\ 0.2537 & -0.1159 \\ 0.0472 & 0.6127 \\ -0.2779 & -0.1134 \end{bmatrix},$$

$$\mathbf{J} = \begin{bmatrix} -0.1927 & -0.5352 \\ -2.9929 & 0.3181 \\ -3.0843 & -4.1625 \\ -0.3260 & -3.0572 \end{bmatrix},$$
(104)

respectively. Thus, constructing the system matrix

$$\mathbf{A}_e = \mathbf{A} - \mathbf{J}\mathbf{C}$$

$$= \begin{bmatrix} -1.3120 & -3.5245 & -3.1456 & -0.7393 \\ 1.3897 & -7.1272 & -0.9629 & -0.2996 \\ 5.7318 & -3.1673 & -9.4238 & 0.2103 \\ 1.6141 & 4.2980 & 0.9545 & -0.8766 \end{bmatrix},$$
(105)

the stable eigenvalues spectrum of  $\mathbf{A}_e$  is

$$\rho(\mathbf{A}_e) = \{-2.1314 \quad -4.6082 \quad -6.0000 \quad -6.0000\},$$
(106)

and the steady-state value of the fault detection filter transfer function matrix is

$$\mathbf{G}_f(0) = -\mathbf{V}^{-1}\mathbf{C}\mathbf{A}_e^{-1}\mathbf{F} = \begin{bmatrix} 0.1667 & 0.0000 \\ 0.0000 & 0.1667 \end{bmatrix}.$$
(107)

For completeness it can be verified that in sense of Lyapunov stability there exists the positive definite matrix  $\mathbf{P}$  such that (70) and (71) are affirmative.

Considering the sign (-) in  $\mathbf{F}$ , that is,  $\mathbf{F} = \mathbf{B}$ , it changes signum of the (2, 1) element of  $\mathbf{V}$  that is

$$\mathbf{V} = \mathbf{C}\mathbf{F} = \begin{bmatrix} 4.9650 & 2.8942 \\ 2.8967 & 0.8184 \end{bmatrix}, \quad \det(\mathbf{V}) \neq 0,$$
(108)

and the parameters of the matrix  $\mathbf{T}$  are now computed as follows:

$$\mathbf{V}^{-1}\mathbf{C} = \begin{bmatrix} 0.3329 & 0.3523 & -1.2597 & -0.3599 \\ -0.5711 & -1.2470 & 2.1610 & 1.2741 \end{bmatrix},$$

$$\mathbf{F}^\perp = \begin{bmatrix} 0.1753 & -0.5316 & 0.7661 & 0.3158 \\ 0.4507 & 0.5187 & -0.0422 & 0.7253 \end{bmatrix},$$

$$\mathbf{T} = \begin{bmatrix} 0.3329 & 0.3523 & -1.2597 & -0.3599 \\ -0.5711 & -1.2470 & 2.1610 & 1.2741 \\ 0.1753 & -0.5316 & 0.7661 & 0.3158 \\ 0.4507 & 0.5187 & -0.0422 & 0.7253 \end{bmatrix}.$$
(109)

Computing (20) and separating the blocks of the matrix  $\mathbf{A}^\circ$  give now the results

$$\mathbf{A}_{o11} = \begin{bmatrix} 18.5471 & 13.4846 \\ -43.8674 & -25.9954 \end{bmatrix},$$

$$\mathbf{A}_{o12} = \begin{bmatrix} -9.8798 & -3.0397 \\ 12.5359 & 10.1353 \end{bmatrix},$$

$$\mathbf{A}_{o21} = \begin{bmatrix} -10.0330 & -6.2445 \\ -11.4737 & -6.7856 \end{bmatrix},$$

$$\mathbf{A}_{o22} = \begin{bmatrix} 2.9984 & 2.1854 \\ 4.2953 & -1.7808 \end{bmatrix},$$
(110)

where the unstable eigenvalue spectrum

$$\rho(\mathbf{A}_{o22}) = \{4.4943, -3.2767\}$$
(111)

does not give the possibility to obtain a strictly unitary fault detection filter.

As above, choosing  $s_o = 5$ , it is obtained using (20) and (60) that

$$\mathbf{L}^\circ = \begin{bmatrix} 5.8844 & 4.1078 & -0.9991 \\ -11.1342 & 31.0383 & -8.0335 \\ 8.6811 & -3.7656 & 10.7293 \\ -4.0432 & 19.2186 & -2.6660 \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} -1.5138 & 5.0746 \\ -3.4282 & 2.6391 \\ -4.9462 & 3.2020 \\ -0.1713 & -4.7003 \end{bmatrix},$$

$$\mathbf{N} = \begin{bmatrix} 0.5145 & -0.8819 \\ 0.1266 & 0.3338 \\ 0.1360 & 0.2987 \\ -0.4023 & 0.3267 \end{bmatrix},$$
(112)

$$\mathbf{J} = \mathbf{M} - \mathbf{N} = \begin{bmatrix} -2.0283 & 5.9564 \\ -3.5548 & 2.3054 \\ -5.0822 & 2.9033 \\ 0.2310 & -5.0270 \end{bmatrix},$$

which results in the following unstable eigenvalues spectrum

$$\rho(\mathbf{A}_e) = \rho(\mathbf{A} - \mathbf{J}\mathbf{C})$$

$$= \{4.4943 \quad -3.2767 \quad -6.0000 \quad -6.0000\}.$$
(113)

It is evident that using the coordinate transformation defined by the transform matrix (10) and the block matrix  $\mathbf{A}_{o22}$  in the matrix structure (20) is unobservable in the structure (57)-(58), while the eigenvalues of  $\mathbf{A}_{o22}$  determine the unprecribed subset of eigenvalues of  $\rho(\mathbf{A}_e)$  (compare  $\rho(\mathbf{A}_e)$  and  $\rho\mathbf{A}_{o22}$  in this part of example).

To stabilize the fault detection filter, an additive gain  $\mathbf{H} \in \mathbb{R}^{n \times m}$  is computed solving the set of inequalities (76) and (77) and equality (78), where, with

$$\mathbf{U} = \begin{bmatrix} 14.0196 & -42.1540 \\ 26.1053 & 12.0055 \\ 15.8767 & -10.3000 \\ 2.2226 & -0.2937 \end{bmatrix}, \quad (114)$$

$$\mathbf{S} = \begin{bmatrix} 9.5763 & 5.0779 \\ 2.7170 & 7.2266 \end{bmatrix},$$

it is obtained that

$$\mathbf{H} = \begin{bmatrix} 3.8956 & -8.5706 \\ 2.8161 & -0.3175 \\ 2.5758 & -3.2353 \\ 0.3043 & -0.2545 \end{bmatrix}, \quad (115)$$

$$\mathbf{J}^\circ = \mathbf{J} + \mathbf{H} = \begin{bmatrix} 1.8673 & -2.6141 \\ -0.7387 & 1.9879 \\ -2.5064 & -0.3320 \\ 0.5353 & -5.2815 \end{bmatrix}.$$

This ensures the stable eigenvalues spectrum of  $\mathbf{A}_e$  as follows:

$$\rho(\mathbf{A} - \mathbf{J}^\circ \mathbf{C}) = \{-2.3882 + 6.4561i \quad -1.5853 + 4.9788i\}, \quad (116)$$

but the fault detection filter transfer function matrix is not unitary and works with the steady-state value

$$\mathbf{G}_f(0) = -\mathbf{V}^{-1} \mathbf{C} \mathbf{A}_e^{-1} \mathbf{F} = \begin{bmatrix} -0.0484 & 0.0474 \\ 0.2132 & 0.0262 \end{bmatrix}. \quad (117)$$

It is obvious that for a stable  $\mathbf{A}_{o22}$  the fault detection filter with a unitary transfer function can be designed, while for an unstable  $\mathbf{A}_{o22}$  the fault detection filter only with a Hurwitz transfer function can be nominated.

Because the matrix  $\mathbf{A}$  is not Hurwitz, for the simulation purpose, the system is stabilized using the state feedback control law

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{q}(t), \quad (118)$$

where  $\mathbf{K} \in \mathbb{R}^{3 \times 4}$  is the gain matrix. Since, according to the separation principle, the control gain matrix can be designed independently of the fault detection filter parameters design, the MATLAB function  $K = \text{place}(\cdot)$  is used. Designing, for simplicity, with the prescribed desired set of closed-loop system matrix eigenvalues  $\rho(\mathbf{A} - \mathbf{BK}) = \{-1, -2, -3, -4\}$ , the gain matrix is

$$\mathbf{K} = \begin{bmatrix} -0.8205 & -1.7778 & -0.1700 & 2.6615 \\ -1.1104 & 3.5495 & 3.0193 & 1.3570 \end{bmatrix}. \quad (119)$$

Note the control law design could be created, for example, using the bounded real lemma LMI to reflect the  $H_\infty$  norm of the disturbance transfer function matrix (see, e.g., [37]), though this still does not solve completely the problem of integrated design of fault detection and FTC. But such a task is significantly beyond the scope of this paper.

In the simulation, the initial conditions are  $\mathbf{q}(0) = [1 \ 0 \ 0 \ 0]$  and  $\mathbf{q}_e(0) = \mathbf{0}$  and the variance of the disturbance noise  $d(t)$  is  $\sigma_d^2 = 0.01$ . The single fault  $f_2(t)$  is considered in both cases, while this fault is modeled as the step function with amplitude equal to one and continuing from the starting time instant  $t = 50$  s.

In Figures 1 and 2 are shown the singular values plot (a) and the fault detection filter response (b), both for the systems under state control in autonomous regime. The value  $s_o = 5$  was chosen in order to not decelerate the observer dynamics conditioned by the stable eigenvalues of  $\mathbf{A}_{o22}$ .

*Example 2.* The unstable system is represented by the chemical reactor model [36] in the form (1). The system model matrices are given as follows:

$$\mathbf{A} = \begin{bmatrix} 1.380 & -2.080 & 6.715 & -5.676 \\ -0.581 & -4.290 & 0.000 & 0.675 \\ 10.672 & 4.273 & -6.654 & 5.893 \\ 0.482 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0.000 & 0.000 & 0.000 \\ 5.679 & -1.000 & 0.000 \\ 1.136 & -3.146 & 1.324 \\ 1.136 & 0.000 & 3.496 \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} 0.000 & 0.000 & 0.000 \\ 5.679 & -1.000 & 0.000 \\ 1.136 & -3.146 & 1.324 \\ 1.136 & 0.000 & 3.496 \end{bmatrix}, \quad (120)$$

$$\mathbf{E} = \begin{bmatrix} 1.400 \\ 1.504 \\ 2.233 \\ 0.610 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 4 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

with the corresponding parameters, as defined in (9) and (10),

$$\mathbf{V} = \mathbf{C}\mathbf{F} = \begin{bmatrix} 1.136 & -3.146 & 1.324 \\ 1.136 & 0.000 & 3.496 \\ 5.679 & -1.000 & 0.000 \end{bmatrix},$$

$$\det(\mathbf{V}) \neq 0,$$

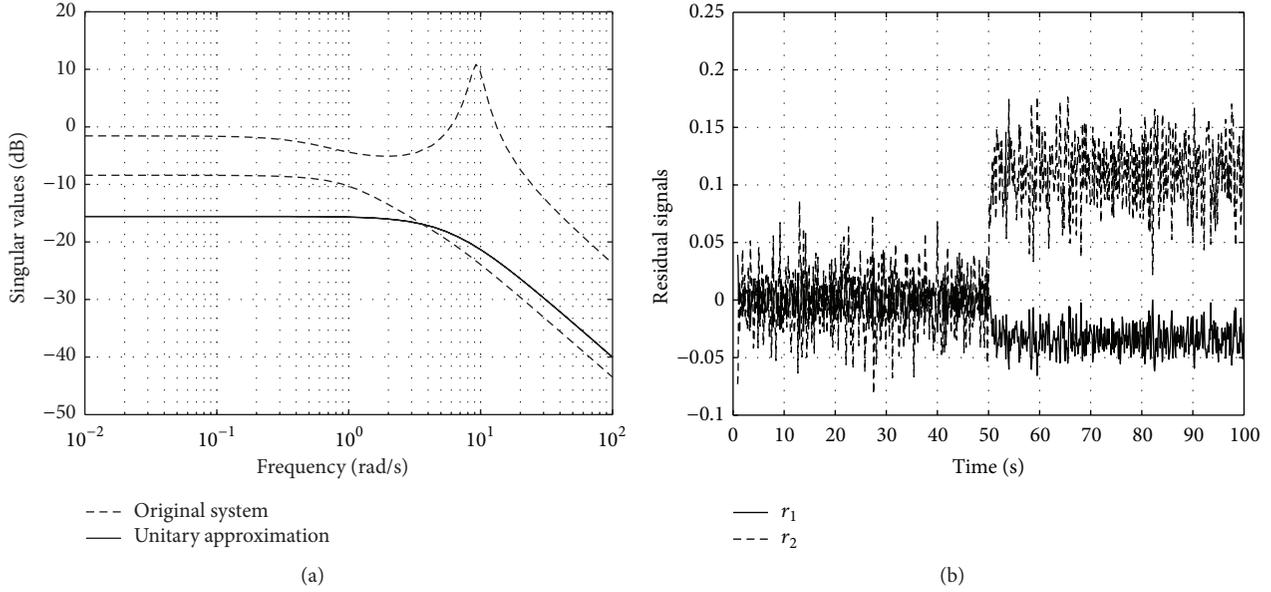


FIGURE 1: (a) Singular values plot. (b) Residuals.

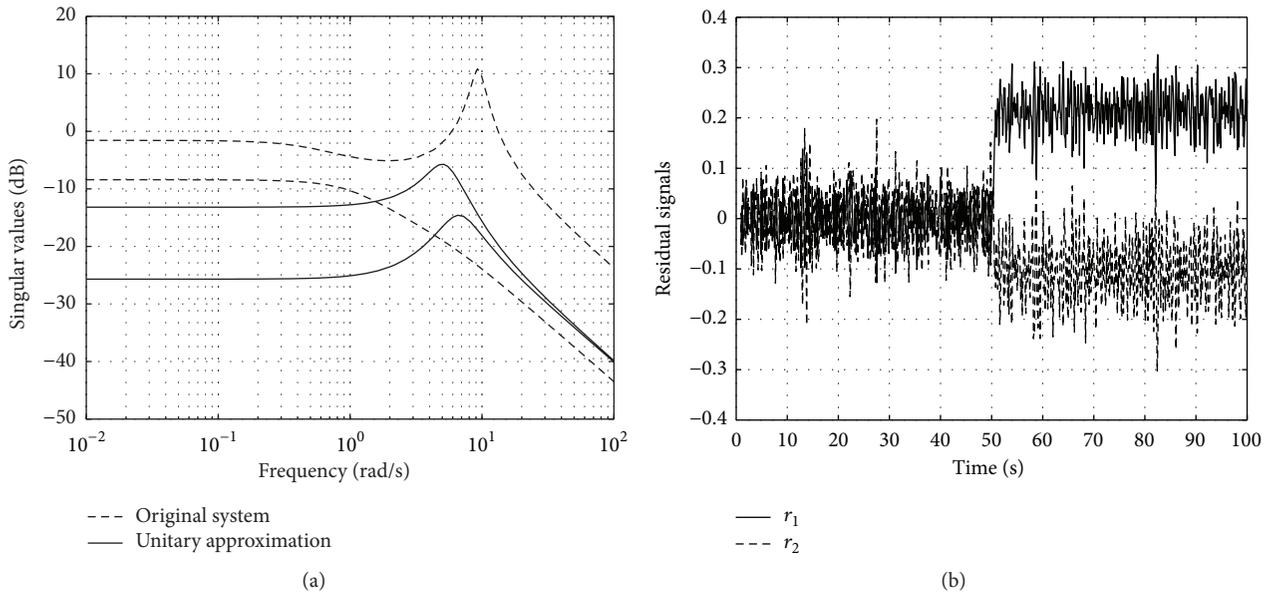


FIGURE 2: (a) Singular values plot. (b) Residuals.

$$\mathbf{V}^{-1}\mathbf{C} = \begin{bmatrix} -0.2331 & 0.1833 & -0.0583 & 0.0221 \\ -1.3237 & 0.0411 & -0.3309 & 0.1253 \\ 0.0757 & -0.0596 & 0.0189 & 0.2789 \end{bmatrix},$$

$$\mathbf{F}^\perp = [-1.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000],$$

$$\mathbf{T} = \begin{bmatrix} -0.2331 & 0.1833 & -0.0583 & 0.0221 \\ -1.3237 & 0.0411 & -0.3309 & 0.1253 \\ 0.0757 & -0.0596 & 0.0189 & 0.2789 \\ -1.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}.$$

(121)

Referring to (20), for the blocks of  $\mathbf{A}^\circ$ , the following terms are computed:

$$\mathbf{A}_{o11} = \begin{bmatrix} -6.1156 & 4.1078 & -0.9991 \\ -11.1342 & 19.0383 & -8.0335 \\ 8.6811 & -3.7656 & -1.2707 \end{bmatrix},$$

$$\mathbf{A}_{o12} = \begin{bmatrix} -3.8116 \\ -21.0651 \\ 2.7614 \end{bmatrix},$$

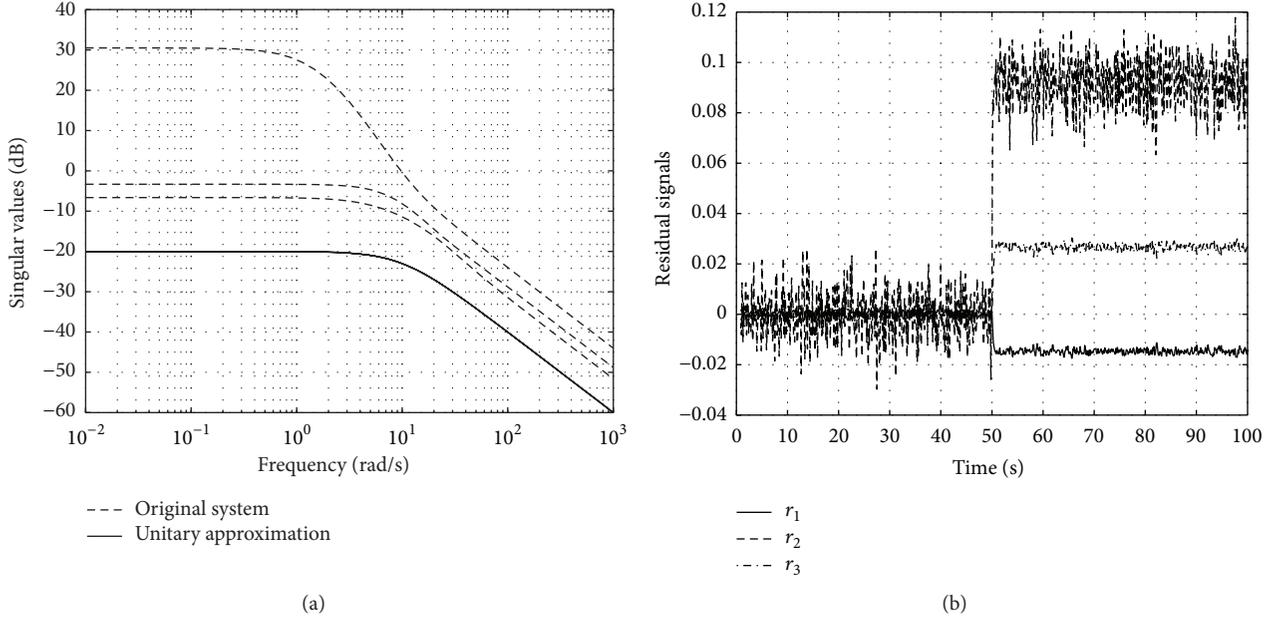


FIGURE 3: (a) Singular values plot. (b) Residuals.

$$\begin{aligned} \mathbf{A}_{o21} &= [-4.0432 \quad 19.2186 \quad -2.6660], \\ \mathbf{A}_{o22} &= [-23.3200], \end{aligned} \quad (122)$$

implying that the fault detection filter with a unitary transfer function can be designed. Choosing  $s_o = 9$ , it is obtained using (21) and (66) that

$$\begin{aligned} \mathbf{L}^o &= \begin{bmatrix} 2.8844 & 4.1078 & -0.9991 \\ -11.1342 & 28.0383 & -8.0335 \\ 8.6811 & -3.7656 & 7.7293 \\ -4.0432 & 19.2186 & -2.6660 \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} 6.1750 & -1.5760 & -0.2080 \\ 0.0000 & 0.6750 & 4.7100 \\ 2.3460 & 5.8930 & 4.2730 \\ 1.3430 & 6.8960 & 4.2730 \end{bmatrix}, \\ \mathbf{N} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \\ \mathbf{J} &= \begin{bmatrix} 6.1750 & -1.5760 & -0.2080 \\ 0.0000 & 0.6750 & 5.7100 \\ 3.3460 & 5.8930 & 4.2730 \\ 1.3430 & 7.8960 & 4.2730 \end{bmatrix}, \end{aligned} \quad (123)$$

while the eigenvalues spectrum of  $\mathbf{A}_e$  and the steady-state value of the fault detection filter transfer function matrix  $\mathbf{G}_f(0)$  are

$$\begin{aligned} \rho(\mathbf{A}_e) &= \{-23.32 \quad -10.00 \quad -10.00 \quad -10.00\}, \\ \mathbf{G}_f(0) &= -\mathbf{V}^{-1}\mathbf{C}\mathbf{A}_e^{-1}\mathbf{F} = \begin{bmatrix} 0.1 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.1 \end{bmatrix}, \end{aligned} \quad (124)$$

respectively. It is evident that for all diagonal elements of  $\mathbf{G}_f(0)$  it yields the relation  $(s_o + 1)^{-1} = 0.1$ .

Also in this example the matrix  $\mathbf{A}$  is not Hurwitz and the system is stabilized using the state feedback control law. Designing, for simplicity, with the prescribed set of closed-loop system matrix eigenvalues  $\rho(\mathbf{A} - \mathbf{B}\mathbf{K}) = \{-2, -3, -3, -4\}$ , the gain matrix is computed as follows:

$$\mathbf{K} = \begin{bmatrix} -0.9686 & -0.2676 & -0.1543 & 0.1358 \\ -4.1485 & -0.6006 & 0.3189 & -0.9145 \\ 0.0893 & 1.4926 & -0.1525 & 0.7086 \end{bmatrix}. \quad (125)$$

In the simulation, the initial conditions are  $\mathbf{q}(0) = [0 \quad 1 \quad 0 \quad 0]$  and  $\mathbf{q}_e(0) = \mathbf{0}$  and the variance of the disturbance noise  $d(t)$  is  $\sigma_d^2 = 0.01$ , while a single fault on the second actuator is considered. The fault is initialized at the time instant  $t = 50$  s as an additive step function with the amplitude equal to one.

In Figure 3 are shown the singular values plot (a) and the fault detection filter response (b), both for the system under state control in autonomous regime. Since the stable

eigenvalue of  $A_{o22}$  determines the sufficiently fast estimator dynamics, the value  $s_o = 9$  is chosen only from the numerical point of view. Moreover, it is possible to see in Figure 3 the directional properties of the output signals of the fault detection filter.

## 7. Concluding Remarks

The approach of solving a unitary approximation of a square fault detection filter transfer function matrix is presented in the context of multiple singular values design, where the conditions for existence of a unitary construction are presented. If the design conditions are satisfied, by choosing one related singular value, the explicit relations for the filter gain matrix design are obtained, which gives a stable fault detection filter with a unitary transfer function to provide high residual signals sensitivity with respect to faults.

Otherwise, reflecting the emplacement of singular values in unitary construction principle and combining the resulting filter gain matrix but for a structure with unstable set of observer system matrix eigenvalues, an associated structure of linear matrix inequalities, as well as one matrix equality together with built-in state and output variable constraints, is outlined to compute an additive stabilizable gain matrix and, in consequence, to design the fault detection filter but only with a Hurwitz transfer function. Formulated in sense of the second Lyapunov method, stability conditions guaranteeing the asymptotic convergence of fault detection filter state are derived for continuous-time linear systems. The numerical simulation results show very good approximation performances.

Although the results represent an improvement on solutions, some conservatism may exist since a common matrix variable is required to satisfy the LMI with a quadratic constraint, but only for systems which do not satisfy the matching condition, that is, when the fault detection filter with a Hurwitz transfer function matrix has to be designed. Although the design conditions are not formulated in terms of robust stability, under nominal occasions, the robustness is flattened to comparable design methods [38–40]. The robustness still remains an open and challenging problem.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

The work presented in this paper was supported by VEGA, the Grant Agency of the Ministry of Education, and the Academy of Science of Slovak Republic, under Grant no. 1/0348/14. This support is gratefully acknowledged.

## References

- [1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, Springer, Berlin, Germany, 2006.
- [2] M. S. Mahmoud and Y. Xia, *Analysis and Synthesis of Fault-Tolerant Control Systems*, John Wiley & Sons, Chichester, UK, 2014.
- [3] H. Noura, D. Theilliol, J. C. Ponsart, and A. Chamseddine, *Fault-tolerant Control Systems: Design and Practical Applications*, Springer, London, UK, 2009.
- [4] K. Zhang, B. Jiang, and P. Shi, *Observer-Based Fault Estimation and Accommodation for Dynamic Systems*, Springer, Berlin, Germany, 2013.
- [5] S. Ding, *Model-Based Fault Diagnosis Techniques. Design Schemes, Algorithms, and Tools*, Springer, Berlin, Germany, 2008.
- [6] A. Zolghadri, D. Henry, J. Cieslak, D. Efimov, and P. Goupil, *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles. From Theory to Application*, Springer, London, UK, 2014.
- [7] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic System*, Kluwer Academic, Norwell, Mass, USA, 1999.
- [8] M. Hou and R. J. Patton, “An LMI approach to  $H_2/H_\infty$  fault detection observers,” in *Proceedings of the UKACC International Conference on Control (CONTROL '96)*, pp. 305–310, Exeter, UK, September 1996.
- [9] S. X. Ding, T. Jeansch, P. M. Frank, and E. L. Ding, “A unified approach to the optimization of fault detection systems,” *International Journal of Adaptive Control and Signal Processing*, vol. 14, no. 7, pp. 725–745, 2000.
- [10] N. Liu and K. Zhou, “Optimal robust fault detection for linear discrete time systems,” *Journal of Control Science and Engineering*, vol. 2008, Article ID 829459, 16 pages, 2008.
- [11] W. Ding, Z. Mao, B. Jiang, and W. Chen, “Fault detection for a class of nonlinear networked control systems with Markov transfer delays and stochastic packet drops,” *Circuits, Systems, and Signal Processing*, vol. 34, no. 4, pp. 1211–1231, 2015.
- [12] Y. Zhang, L. Wu, J. Li, and X. Chen, “LMI approach to mixed  $H_2/H_\infty$  fault detection observer design,” *Transactions of Tianjin University*, vol. 18, no. 5, pp. 343–349, 2012.
- [13] Z. Zhao, W.-F. Xie, H. Hong, and Y. Zhang, “Unitary system I. Constructing a unitary fault detection observer,” in *Proceedings of the 18th IFAC World Congress*, pp. 7725–7730, Milano, Italy, 2011.
- [14] Z. Zhao, W.-F. Xie, H. Hong, and Y. Zhang, “Unitary system II. Application to  $H1/H$  optimization of strictly-proper systems,” in *Proceedings of the 18th IFAC World Congress*, pp. 7731–7736, Milano, Italy, 2011.
- [15] D. Krokavec and A. Filasová, “Construction process of unitary systems using LMI based approach,” *IFAC-PapersOnLine*, vol. 48, no. 14, pp. 174–179, 2015.
- [16] D. Krokavec, A. Filasová, and P. Liščinský, “On fault detection filters design with unitary transfer function matrices,” *Journal of Physics: Conference Series*, vol. 659, no. 1, Article ID 012036, 2015.
- [17] F. N. Chowdhury and W. Chen, “A modified approach to observer-based fault detection,” in *Proceedings of the IEEE Multi-Conference on Systems and Control, 16th IEEE International Conference on Control Applications*, pp. 946–951, Singapore, 2007.
- [18] T. Y. Azizov and I. S. Iokhvidov, *Linear Operators in Spaces with an Indefinite Metric*, John Wiley & Sons, Chichester, UK, 1989.
- [19] M. H. Yang, “The state space of a canonical linear system,” *Journal of the Korean Mathematical Society*, vol. 32, no. 3, pp. 447–459, 1995.

- [20] L. Debnath and D. Bhatta, *Integral Transforms and Their Applications*, Chapman & Hall, Boca Raton, Fla, USA, 2007.
- [21] S. Toffner-Clausen, *System Identification and Robust Control. A Case Study Approach*, Springer, London, UK, 1996.
- [22] B. A. Francis, Ed., *A Course in  $H_\infty$  Control Theory*, vol. 88 of *Lecture Notes in Control and Information Sciences*, Springer, Berlin, Germany, 1987.
- [23] D. Boyd and V. Balakrishnan, "A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L1-norm," *Systems & Control Letters*, vol. 15, no. 1, pp. 1–7, 1990.
- [24] X. Li and H. H. T. Liu, "Minimum system sensitivity study of linear discrete time systems for fault detection," *Mathematical Problems in Engineering*, vol. 2013, Article ID 276987, 13 pages, 2013.
- [25] Y. Li, T. Zhang, X. Liu, and X. Jiang, "Study on  $H_2$  index of stochastic linear continuous-time systems," *Mathematical Problems in Engineering*, vol. 2015, Article ID 837053, 10 pages, 2015.
- [26] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, Maxwell Macmillan International, Singapore, 1992.
- [27] M. Green and D. J. N. Limebeer, *Linear Robust Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.
- [28] O. N. Gasparyan, *Linear and Nonlinear Multivariable Feedback Control. A Classical Approach*, John Wiley & Sons, Chichester, UK, 2008.
- [29] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas*, Princeton University Press, Princeton, NJ, USA, 2009.
- [30] W. A. Adkins and S. H. Weintraub, *Algebra—An Approach via Module Theory*, Springer, Berlin, Germany, 1992.
- [31] F. W. Fairman, *Linear Control Theory: The State Space Approach*, John Wiley & Sons, Chichester, UK, 1998.
- [32] J. W. Miles, *Integral Transforms in Applied Mathematics*, Cambridge University Press, New York, NY, USA, 1971.
- [33] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 2002.
- [34] D. G. Schultz and J. L. Melsa, *State Functions and Linear Control Systems*, McGraw-Hill, New York, NY, USA, 1967.
- [35] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control. A Lyapunov-Based Approach*, Princeton University Press, Princeton, NJ, USA, 2008.
- [36] J. Kautsky, N. K. Nichols, and P. Van Dooren, "Robust pole assignment in linear state feedback," *International Journal of Control*, vol. 41, no. 5, pp. 1129–1155, 1985.
- [37] D. Krokavec and A. Filasová, "LMI constraints on system eigenvalues placement in dynamic output control design," in *Proceedings of the IEEE Conference on Control Applications (CCA '15)*, pp. 1749–1754, Sydney, Australia, September 2015.
- [38] I. M. Jaimoukha, Z. Li, and V. Papakos, "A matrix factorization solution to the  $H_2/H_\infty$  fault detection problem," *Automatica*, vol. 42, no. 11, pp. 1907–1912, 2006.
- [39] A. Saberi, A. A. Stoorvogel, and P. Sannuti, *Filtering Theory. With Applications to Fault Detection, Isolation, and Estimation*, Birkhäuser, Boston, Mass, USA, 2007.
- [40] X. Li, H. H. T. Liu, and B. Jiang, "Fault detection filter design with optimization and partial decoupling," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1951–1956, 2015.

## Research Article

# Analytical Redundancy Design for Aeroengine Sensor Fault Diagnostics Based on SROS-ELM

**Jun Zhou, Yuan Liu, and Tianhong Zhang**

*Jiangsu Province Key Laboratory of Aerospace Power System, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China*

Correspondence should be addressed to Jun Zhou; zhoujunnj@yeah.net

Received 27 December 2015; Accepted 3 April 2016

Academic Editor: Wen Chen

Copyright © 2016 Jun Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Analytical redundancy technique is of great importance to guarantee the reliability and safety of aircraft engine system. In this paper, a machine learning based aeroengine sensor analytical redundancy technique is developed and verified through hardware-in-the-loop (HIL) simulation. The modified online sequential extreme learning machine, selective updating regularized online sequential extreme learning machine (SROS-ELM), is employed to train the model online and estimate sensor measurements. It selectively updates the output weights of neural networks according to the prediction accuracy and the norm of output weight vector, tackles the problems of singularity and ill-posedness by regularization, and adopts a dual activation function in the hidden nodes combining neural and wavelet theory to enhance prediction capability. The experimental results verify the good generalization performance of SROS-ELM and show that the developed analytical redundancy technique for aeroengine sensor fault diagnosis based on SROS-ELM is effective and feasible.

## 1. Introduction

As the control and health management of aircraft engine highly relies on the precise and reliable sensor measurements, sensor is one of the most important components of the aeroengine system. With the increase of control and monitoring variables, the types and number of sensors used in aeroengine are growing [1]. However, most sensors work in severe environment of high temperature, high pressure, and strong vibration which is also changing rapidly [2]; thus they are very vulnerable to breakdown. Therefore, some measures should be adopted to ensure the correctness of sensor readings. For this challenge, sensor redundancy technique is a good solution. Generally there are two kinds of sensor redundancy: one is hardware redundancy and the other is analytical redundancy [3]. Hardware redundancy adopts more than one sensor to measure the same engine variable, but it is at the price of extra cost, maintenance, additional weight, and more space to furnish sensors. Analytical redundancy constructs redundant estimates of the sensor readings using numerical algorithms to reduce the weight and cost of aeroengine. With the remarkable advantages, the

analytical redundancy technique has been widely researched and applied in aeroengine system since it originated. Wallhagen and Arpasi demonstrated the application of sensor analytical redundancy technique for enhancing the reliability of engine control system [4]. Corley et al. developed a fault indication and correction (FICA) system and employed it to engine control system for T700, JTDE, and F404, which laid the theoretical basis and set an excellent example for the application of analytical redundancy in engine control system [5, 6].

The analytical redundancy design methods fall into three categories: model-based, data-driven, and the hybrid approach. Model-based techniques can be utilized to diagnose new sensor faults with no prior knowledge and experience, but it depends on accurate on-board adaptive engine model whose reliability may decline with the increase of modeling uncertainties and nonlinear complexities [7]. On the other side, the data-driven method needs no knowledge about the intricate engine working principle and complicated modeling skills and thus attracts lots of interest and concern. Botros et al. presented an application of optimized radial basis function neural networks based data mining to sensor faults

detection on gas-turbine-driven compressor stations [8]. Huang employed autoassociative neural networks to detect sensor failures at the absence of models and reconstruct engine control system [9]. Joly et al. developed a gas-turbine diagnostics structure using several artificial neural networks for a high bypass ratio military turbofan engine [10]. Ogaji et al. also conceived artificial neural networks based sensor faults diagnosis for gas-turbine. The system was trained to detect, isolate, and assess faults [11]. However, most of the existed methods are trained offline, which cannot capture and adapt to the dynamic changes of system characteristics. Besides, conventional machine learning algorithms that the analytical redundancy technique is based on have weakness such as learning samples slowly and requiring massive computing resource.

Extreme learning machine (ELM) is a novel and efficient learning algorithm for training single-hidden layer feed-forward neural networks (SLFNs) proposed by Huang et al. [12]. It has been proven that ELM method has not only classification capability but also universal approximation capability [13, 14]. In addition, as verified in [14], ELM can learn much faster than traditional SVM, LS-SVM, and neural networks while achieving similar or much better generalization performance. Nevertheless, ELM is an offline learning algorithm. In order to learn data one-by-one or chunk-by-chunk online, Liang et al. proposed a fast and accurate online sequential extreme learning machine (OS-ELM) based on the idea of ELM [15]. Different from adjusting learning parameters iteratively like gradient-based neural networks does, it randomly generates input weights and hidden biases and determines the output weights analytical according to the sequentially arriving data. However, there are still limitations such as singularity and ill-posedness problems, potentially inconsistent and unstable performance. To alleviate such weaknesses, this paper proposed a modified online sequential extreme learning machine, selective updating regularized online sequential extreme learning machine (SROS-ELM). And then an aeroengine analytical redundancy technique based on that is developed and verified through HIL simulation.

The rest of this paper is organized as follows. Section 2 gives a brief review of OS-ELM. Section 3 presents SROS-ELM algorithm and evaluates its performance. Section 4 describes the analytical redundancy technique based on SROS-ELM in detail. The verification of the developed technique through hardware-in-the-loop (HIL) simulation is shown in Section 5. Conclusions are drawn in Section 6.

## 2. Brief Review of OS-ELM

For  $N$  arbitrary distinct samples  $(\mathbf{x}_i, \mathbf{t}_i)$ , where  $\mathbf{x}_i = [x_{i1} \ x_{i2} \ \cdots \ x_{in}]^T \in R^n$  is the input vector,  $\mathbf{t}_i = [t_{i1} \ t_{i2} \ \cdots \ t_{im}]^T \in R^m$  is the target vector. The output function of SLFN with  $L$  hidden neurons and activation function  $g(x)$  can be represented as

$$\sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \quad j = 1, \dots, N, \quad (1)$$

where  $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  is the weight vector connecting the input layer and the  $i$ th hidden neuron,  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  is the weight vector connecting the  $i$ th hidden neuron and the output layer, and  $b_i$  is the bias of the  $i$ th hidden neuron.

The  $N$  equations can be written in form of matrix

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \quad (2)$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}, \quad (3)$$

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L}.$$

$\mathbf{H}$  is named hidden layer output matrix [12]. The hidden node parameters  $\mathbf{w}_i$  and  $b_i$  are simply assigned with random values and need not to be tuned. Thus, the output weight vector  $\boldsymbol{\beta}$  is the only parameter that needs to be calculated. The determination of the output weights can be simplified as seeking the least-square solution to the given linear system, which may be expressed as

$$\boldsymbol{\beta} = \mathbf{H}^\dagger \mathbf{T}, \quad (4)$$

where  $\mathbf{H}^\dagger$  is Moore-Penrose generalized inverse of matrix  $\mathbf{H}$ , which can be calculated through orthogonal projection as  $\mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$  when  $\mathbf{H}^T \mathbf{H}$  is nonsingular. Substituting into (4),  $\boldsymbol{\beta}$  becomes

$$\boldsymbol{\beta} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{T}. \quad (5)$$

For (5), it can be solved recursively by [15]

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P}_k \mathbf{h}_{k+1}^T \mathbf{h}_{k+1} \mathbf{P}_k}{1 + \mathbf{h}_{k+1}^T \mathbf{P}_k \mathbf{h}_{k+1}}, \quad (6)$$

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{P}_{k+1} \mathbf{h}_{k+1}^T (\mathbf{t}_{k+1} - \mathbf{h}_{k+1} \boldsymbol{\beta}_k),$$

where

$$\mathbf{h}_{k+1} = [g(\mathbf{w}_1 \cdot \mathbf{x}_{k+1} + b_1) \ g(\mathbf{w}_2 \cdot \mathbf{x}_{k+1} + b_2) \ \cdots \ g(\mathbf{w}_L \cdot \mathbf{x}_{k+1} + b_L)], \quad (7)$$

and the initialization procedure can be completed by  $\mathbf{P}_0 = (\mathbf{H}_0^T \mathbf{H}_0)^{-1}$  and  $\boldsymbol{\beta}_0 = \mathbf{P}_0 \mathbf{H}_0^T \mathbf{T}_0$ .

The OS-ELM algorithm is suitable for SLFNs with additive or RBF hidden nodes. It can deal with sequential learning tasks much faster than other sequential algorithms with good prediction accuracy. However, the analytical determination of the output weights according to formula (5) is on the hypothesis that  $\mathbf{H}^T\mathbf{H}$  is nonsingular which is not always satisfied. Besides, the ill-posedness or singularity problems have not been paid attention to, which may do great harm to the generalization performance. Furthermore, the output weights are always updated and the debasement in generalization performance because of some new arriving samples has not been considered. The solution for these problems will be investigated in the next section.

### 3. SROS-ELM

In this section, the SROS-ELM is proposed on the basis of OS-ELM and its performance is evaluated on some benchmark data sets.

*3.1. Formula Derivation.* On the basis of ridge regression theory [18], the stability will be improved and better generalization performance can be achieved by introducing a positive value  $\lambda$ , which is also called the regularization factor,

to the diagonal elements of  $\mathbf{H}^T\mathbf{H}$  when determining the output weight vector  $\boldsymbol{\beta}$  [19]. Thus, formula (5) becomes as follows:

$$\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1} \mathbf{H}^T\mathbf{T}. \quad (8)$$

In order to train the SLFNs online, now we need to derivate the recursive formula for updating the output weights.

For an initial training subset,  $\Omega = \{\mathbf{x}_i, \mathbf{t}_i\}_1^{N_0}$ , the initial output weight vector  $\boldsymbol{\beta}_0$  can be estimated by

$$\boldsymbol{\beta}_0 = \mathbf{L}_0^{-1} \mathbf{H}_0^T \mathbf{T}_0, \quad (9)$$

where  $\mathbf{L}_0 = (\mathbf{H}_0^T \mathbf{H}_0 + \lambda \mathbf{I})$ ,  $\mathbf{H}_0 = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_{N_0}]^T$ , and  $\mathbf{T}_0 = [\mathbf{t}_1 \ \mathbf{t}_2 \ \cdots \ \mathbf{t}_{N_0}]^T$ .

Supposing that there is a new sample  $\{\mathbf{x}_{N_0+1}, \mathbf{t}_{N_0+1}\}$ , the output weights can be determined by

$$\begin{aligned} \boldsymbol{\beta}_1 &= \left( \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{h}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{h}_1 \end{bmatrix} + \lambda \mathbf{I} \right)^{-1} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{h}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{t}_1 \end{bmatrix} \\ &= (\mathbf{H}_0^T \mathbf{H}_0 + \mathbf{h}_1^T \mathbf{h}_1 + \lambda \mathbf{I})^{-1} (\mathbf{H}_0^T \mathbf{T}_0 + \mathbf{h}_1^T \mathbf{t}_1), \end{aligned} \quad (10)$$

where

$$\mathbf{h}_1 = [g(\mathbf{w}_1 \cdot \mathbf{x}_{N_0+1} + b_1) \ g(\mathbf{w}_2 \cdot \mathbf{x}_{N_0+1} + b_1) \ \cdots \ g(\mathbf{w}_L \cdot \mathbf{x}_{N_0+1} + b_L)], \quad (11)$$

$\mathbf{t}_1 = \mathbf{t}_{N_0+1}$ .

Let  $\mathbf{L}_1 = \mathbf{H}_0^T \mathbf{H}_0 + \mathbf{h}_1^T \mathbf{h}_1 + \lambda \mathbf{I} = \mathbf{L}_0 + \mathbf{h}_1^T \mathbf{h}_1$ ; substituting this into (10), then the output weights become

$$\begin{aligned} \boldsymbol{\beta}_1 &= \mathbf{L}_1^{-1} (\mathbf{H}_0^T \mathbf{T}_0 + \mathbf{h}_1^T \mathbf{t}_1) = \mathbf{L}_1^{-1} (\mathbf{L}_0 \boldsymbol{\beta}_0 + \mathbf{h}_1^T \mathbf{t}_1) \\ &= \mathbf{L}_1^{-1} (\mathbf{L}_1 - \mathbf{h}_1^T \mathbf{h}_1) \boldsymbol{\beta}_0 + \mathbf{L}_1^{-1} \mathbf{h}_1^T \mathbf{t}_1 \\ &= \boldsymbol{\beta}_0 + \mathbf{L}_1^{-1} \mathbf{h}_1^T (\mathbf{t}_1 - \mathbf{h}_1 \boldsymbol{\beta}_0). \end{aligned} \quad (12)$$

In generalization, for the  $(k+1)$ th observation, the output weights can be updated by

$$\begin{aligned} \mathbf{L}_{k+1} &= \mathbf{L}_k + \mathbf{h}_{k+1}^T \mathbf{h}_{k+1}, \\ \boldsymbol{\beta}_{k+1} &= \boldsymbol{\beta}_k + \mathbf{L}_{k+1}^{-1} \mathbf{h}_{k+1}^T (\mathbf{t}_{k+1} - \mathbf{h}_{k+1} \boldsymbol{\beta}_k). \end{aligned} \quad (13)$$

$\mathbf{L}_{k+1}$  is a matrix with the size of  $L \times L$ , where  $L$  denotes the number of hidden nodes in neural networks. As the number of hidden nodes is usually quite large and the computation of the inverse matrix is resource consuming, the update formula for  $\mathbf{L}_{k+1}^{-1}$  can be expressed using Woodbury formula [20] so as to save computing cost:

$$\begin{aligned} \mathbf{L}_{k+1}^{-1} &= (\mathbf{L}_k + \mathbf{h}_{k+1}^T \mathbf{h}_{k+1})^{-1} \\ &= \mathbf{L}_k^{-1} - \mathbf{L}_k^{-1} \mathbf{h}_{k+1}^T (1 + \mathbf{h}_{k+1} \mathbf{L}_k^{-1} \mathbf{h}_{k+1}^T)^{-1} \mathbf{h}_{k+1} \mathbf{L}_k^{-1}. \end{aligned} \quad (14)$$

When the new samples arrive one-by-one rather than block-by-block, formula (14) can be written in a simple format on the basis of Sherman-Morrison formula [20]:

$$\mathbf{L}_{k+1}^{-1} = \mathbf{L}_k^{-1} - \frac{\mathbf{L}_k^{-1} \mathbf{h}_{k+1}^T \mathbf{h}_{k+1} \mathbf{L}_k^{-1}}{1 + \mathbf{h}_{k+1} \mathbf{L}_k^{-1} \mathbf{h}_{k+1}^T}. \quad (15)$$

Let  $\mathbf{P}_k = \mathbf{L}_k^{-1}$ ; then the equations for updating become

$$\begin{aligned} \mathbf{P}_{k+1} &= \mathbf{P}_k - \frac{\mathbf{P}_k \mathbf{h}_{k+1}^T \mathbf{h}_{k+1} \mathbf{P}_k}{1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T}, \\ \boldsymbol{\beta}_{k+1} &= \boldsymbol{\beta}_k + \mathbf{P}_{k+1} \mathbf{h}_{k+1}^T (\mathbf{t}_{k+1} - \mathbf{h}_{k+1} \boldsymbol{\beta}_k). \end{aligned} \quad (16)$$

Further,  $\mathbf{P}_{k+1} \mathbf{h}_{k+1}^T$  can be written as

$$\begin{aligned} \mathbf{P}_{k+1} \mathbf{h}_{k+1}^T &= \mathbf{P}_k \mathbf{h}_{k+1}^T - \mathbf{P}_k \mathbf{h}_{k+1}^T (1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T)^{-1} \\ &\quad \cdot \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T = \mathbf{P}_k \mathbf{h}_{k+1}^T (1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T)^{-1} \\ &\quad \cdot (1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T) - \mathbf{P}_k \mathbf{h}_{k+1}^T (1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T)^{-1} \end{aligned}$$

$$\begin{aligned}
& \cdot \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T = \mathbf{P}_k \mathbf{h}_{k+1}^T \left(1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T\right)^{-1} \\
& \cdot \left(1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T - \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T\right) \\
& = \frac{\mathbf{P}_k \mathbf{h}_{k+1}^T}{1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T}.
\end{aligned} \tag{17}$$

Let  $\mathbf{Q}_{k+1} = \mathbf{P}_{k+1} \mathbf{h}_{k+1}^T = \mathbf{P}_k \mathbf{h}_{k+1}^T / (1 + \mathbf{h}_{k+1} \mathbf{P}_k \mathbf{h}_{k+1}^T)$ ; a simple recursive formula for the output weights can be achieved:

$$\begin{aligned}
\boldsymbol{\beta}_{k+1} &= \boldsymbol{\beta}_k + \mathbf{Q}_{k+1} (\mathbf{t}_{k+1} - \mathbf{h}_{k+1} \boldsymbol{\beta}_k), \\
\mathbf{P}_{k+1} &= \mathbf{P}_k - \mathbf{Q}_{k+1} \mathbf{h}_{k+1} \mathbf{P}_k.
\end{aligned} \tag{18}$$

As investigated by Zhu et al. [21] and Bartlett [22], the prediction accuracy of neural networks is determined by not only the training error but also the norm of output weights. The networks with weights of smaller norm tend to perform superior generalization. Taking account of these two factors, the output weights are updated selectively according to both the prediction error and the norm of output weights:

$$\begin{aligned}
\boldsymbol{\beta}_{k+1} &= \begin{cases} \boldsymbol{\beta}_{k+1} & R_e > \varepsilon \text{ or } (R_e \leq \varepsilon, \boldsymbol{\beta}_{k+1} < \boldsymbol{\beta}_k) \\ \boldsymbol{\beta}_k & R_e \leq \varepsilon, \boldsymbol{\beta}_{k+1} \geq \boldsymbol{\beta}_k, \end{cases} \\
\mathbf{P}_{k+1} &= \begin{cases} \mathbf{P}_k - \mathbf{Q}_{k+1} \mathbf{h}_{k+1} \mathbf{P}_k & R_e > \varepsilon \text{ or } (R_e \leq \varepsilon, \boldsymbol{\beta}_{k+1} < \boldsymbol{\beta}_k) \\ \mathbf{P}_k & R_e \leq \varepsilon, \boldsymbol{\beta}_{k+1} \geq \boldsymbol{\beta}_k, \end{cases}
\end{aligned} \tag{19}$$

where  $R_e = \|\mathbf{t}_{k+1} - \hat{\mathbf{t}}_{k+1}\| = \|\mathbf{t}_{k+1} - \mathbf{h}_{k+1} \boldsymbol{\beta}_k\|$  denotes the prediction error of  $\mathbf{t}_{k+1}$  and  $\varepsilon$  is the threshold for decision.

Based on the theory proposed by [23], the activation functions of the hidden nodes have great impact on the performance of neural networks. As verified in [24], the neural networks taking inverse hyperbolic sine and Morlet wavelet as dual activation function can enhance dealing with nonlinearity and dynamic systems and achieve promising performance. Following that, this paper adopts a dual activation function in the hidden units combining neural and wavelet theory to improve prediction capability:

$$g(x) = \frac{1}{2} \left( \operatorname{arcsinh}(x) + \cos(5x) e^{(-0.5x^2)} \right). \tag{20}$$

In summary, the procedure of the SROS-ELM goes as follows:

- (1) Initialization: for the initial training subset,  $\Omega = \{\mathbf{x}_i, \mathbf{t}_i\}_1^{N_0}$ ,  $N_0 \geq L$ ,
  - (a) randomly set the value of input weights  $\mathbf{w}_i$  and bias  $b_i$ ,  $i = 1, 2, \dots, L$  and choose a proper  $\lambda$ ;
  - (b) calculate hidden layer output matrix  $\mathbf{H}_0 = [\mathbf{h}_1 \ \mathbf{h}_2 \ \dots \ \mathbf{h}_{N_0}]^T$ ;
  - (c) estimate the initial output weight vector  $\boldsymbol{\beta}_0$ ,  $\boldsymbol{\beta}_0 = \mathbf{P}_0 \mathbf{H}_0^T \mathbf{T}_0$ , where  $\mathbf{P}_0 = (\mathbf{H}_0^T \mathbf{H}_0 + \lambda \mathbf{I})^{-1}$  and  $\mathbf{T}_0 = [\mathbf{t}_1 \ \mathbf{t}_2 \ \dots \ \mathbf{t}_0]^T$ ;
  - (d) set  $k = 0$ .

TABLE 1: Performance comparison on regression application.

Data sets	Algorithms	RMSE		# nodes
		Training	Testing	
California housing	SROS-ELM	0.1301	0.1319	50
	OS-ELM [15]	0.1303	0.1332	50
	Stochastic BP [15]	0.1688	0.1704	9
Abalone	MRAN [16]	0.1598	0.1586	64
	SROS-ELM	0.0752	0.0771	25
	OS-ELM [15]	0.0754	0.0777	25
Auto-MPG	Stochastic BP [15]	0.0996	0.0972	11
	MRAN [17]	0.0836	0.0837	87574
	SROS-ELM	0.0675	0.0728	25
Auto-MPG	OS-ELM [15]	0.0680	0.0745	25
	Stochastic BP [15]	0.1112	0.1028	13
	MRAN [17]	0.1086	0.1376	4.46

- (2) Updating output weights: for the new arriving training sample,  $\{\mathbf{x}_{k+1}, \mathbf{t}_{k+1}\}$ ,

(a) calculate

$$\begin{aligned}
\mathbf{h}_{k+1} &= [g(\mathbf{w}_1 \cdot \mathbf{x}_{k+1} + b_1) \ g(\mathbf{w}_2 \cdot \mathbf{x}_{k+1} + b_1) \ \dots \ g(\mathbf{w}_L \cdot \mathbf{x}_{k+1} + b_L)];
\end{aligned} \tag{21}$$

- (b) calculate  $\mathbf{Q}_{k+1}$  and  $\boldsymbol{\beta}_{k+1}$  according to (18) and selectively update  $\boldsymbol{\beta}_{k+1}$  and  $\mathbf{P}_{k+1}$ ;

(c) set  $k = k + 1$ . Go to step (a).

**3.2. Evaluation Test.** As aeroengine analytical redundancy design using machine learning is a regression problem, the proposed SROS-ELM is evaluated on some real-world regression applications which are also used in [15] and the performance is compared with some popular algorithms. A sigmoid function is taken as hidden unit activation function for OS-ELM, while in SROS-ELM dual activations combining inverse hyperbolic sine function and Morlet wavelet are adopted. Since SROS-ELM has the same computational complexity as OS-ELM, there is no need to take the comparison of training and testing time into consideration. The root mean square error (RMSE) which is defined as the deviation measurement between the predicted and the target values is taken as the performance criterion. Table 1 has listed the average results of fifty trials on different benchmark data sets. From Table 1, it is obvious that the proposed SROS-ELM outperforms OS-ELM in generalization performance while the numbers of hidden nodes are the same. Moreover, the SROS-ELM always achieves smaller RMSE than some other popular sequential learning algorithms; thus it is more suitable for aeroengine analytical redundancy design which will be illustrated specifically in the next section.

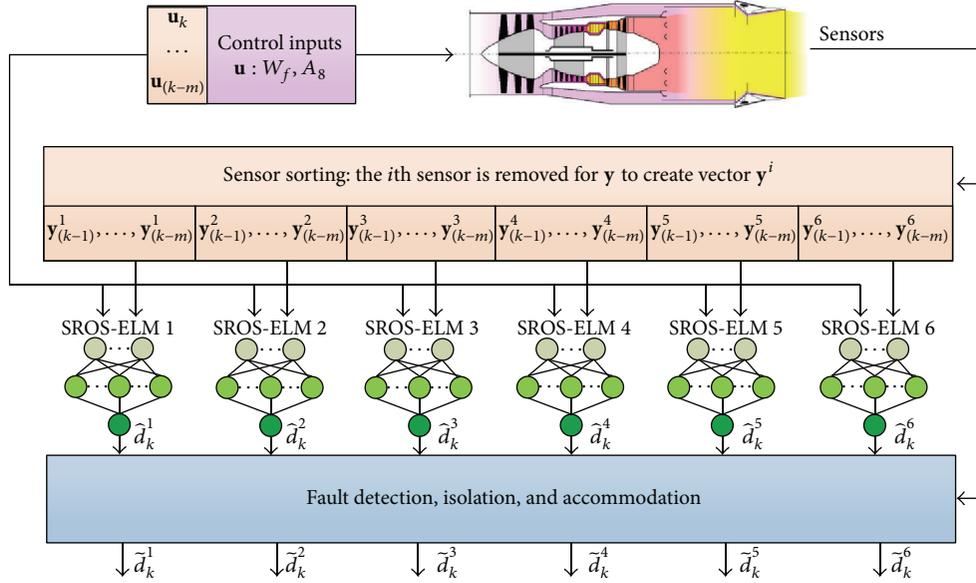


FIGURE 1: Aeroengine analytical redundancy based on SROS-ELM.

#### 4. Aeroengine Analytical Redundancy Design

As the reliability and safety of aircraft engine system can never be emphasized too much and sensor fault diagnostics plays an important role in guaranteeing them, in this section an analytical redundancy technique based on SROS-ELM is developed for sensor fault detection, isolation, and accommodation. Figure 1 illustrates the schematic, which works as follows: firstly the engine control inputs and sensor measurements are pretreated and reorganized for the SROS-ELM based sensor measurement estimators, following that the estimators perform the estimations for the corresponding sensor measurements according to the inputs of SROS-ELM, finally the sensor fault detection, isolation, and accommodation are achieved with the fault diagnosis and signal reconstruction strategy.

From Figure 1 it can be seen that six estimators are built and each is separately responsible for the estimation of the fan speed ( $N_f$ ), the core speed ( $N_h$ ), the pressure after compressor ( $P_3$ ), the temperature after compressor ( $T_3$ ), the pressure after low pressure turbine ( $P_{45}$ ), and the temperature after low pressure turbine ( $T_{45}$ ). For every estimator, the inputs are composed of the engine control inputs such as the main fuel flow and the nozzle throat area and the other five sensors' measurements. With the aim to enhance the estimation capability and obtain good accuracy especially during dynamic process, not only the current control inputs and sensor measurements but also the values of some previous steps are adopted as the neural networks' inputs.

The estimator based on RSOS-ELM for each sensor can be expressed as

$$\hat{d}_k^i = f_i(y_{(k-1)}^i, \dots, y_{(k-m)}^i, \mathbf{u}_k, \dots, \mathbf{u}_{(k-m)}), \quad (22)$$

$$i = 1, \dots, 6,$$

where  $\hat{d}_k^i$  is the estimation of the  $i$ th sensor,  $\mathbf{y}^i$  is a vector obtained by removing the  $i$ th sensor value from all the six sensors' measurements  $\mathbf{y}$ ,  $\mathbf{u} = [W_f \ A_8]$  denotes the engine control inputs including the fuel flow  $W_f$  and the nozzle throat area  $A_8$ , and  $m$  is the embedded dimension.

The fault detection and isolation are realized by comparing the residuals to the fault threshold  $FC$ . The residuals are defined as the deviations between the estimated and measured sensor values:

$$r_k^i = |\hat{d}_k^i - d_k^i|, \quad i = 1, \dots, 6, \quad (23)$$

where  $\hat{d}_k^i$  represents the estimated value for the  $i$ th sensor and  $d_k^i$  denotes the measured value of the  $i$ th sensor. If  $r_k^i < FC$ , it can be judged that the  $i$ th sensor is normal; otherwise the sensor is faulty. Once a sensor is found to break down, the estimated value by the estimator will take place of the incorrect sensor measurement to achieve fault accommodation. Therefore, reliable and accurate sensor signal  $\hat{d}_k^i$  can be provided for engine control and health management.

#### 5. Verification through HIL Simulation

Hardware-in-the-loop simulation is an efficient way for the verification and validation of safety-critical control system such as aeroengine control system [25]. It enables the real-time simulation wherein the actual controller or important components under test couple with the simulated plant in a manner similar to the real-world, which can effectively save the development cost and reduce the risks of accident occurring at test [26]. Hence, HIL simulation is widely applied in industry and recommended by Electrical and Electronics Engineers (IEEE), International Atomic Energy Agency (IAEA), and the United States Nuclear Regulatory Commission (US-NRC) [27]. Aiming at verifying the

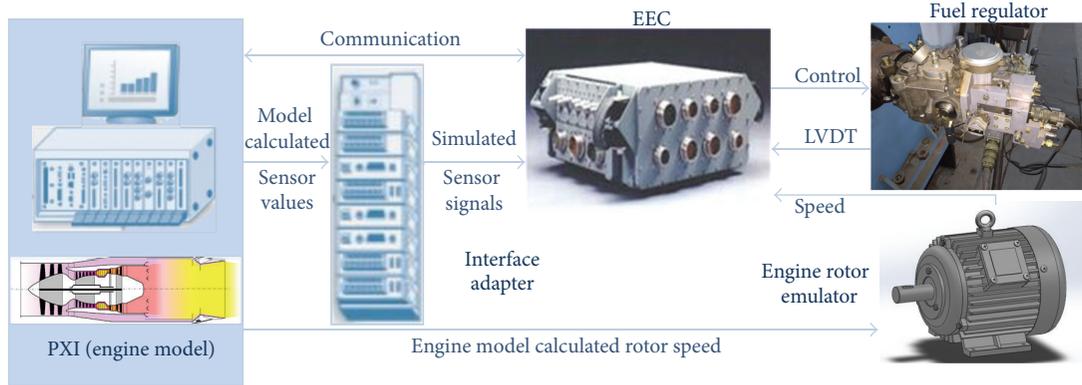


FIGURE 2: Framework of the HIL simulation system.

developed aeroengine analytical redundancy technique, the HIL simulation platform for engine control system in the laboratory is introduced and then some HIL simulation experiments are performed and analyzed in this section.

**5.1. HIL Simulation System.** The HIL simulation system for the design and testing of aeroengine control system is mainly composed of model computer, simulator, electronic engine controller (EEC), fuel subsystem, and aeroengine shaft rotating speed simulating subsystem. Figure 2 gives the framework of the HIL simulation system. The actual aeroengine is emulated by a component level model (CLM) with high accuracy running on model computer, which outputs the calculated parameters based on the control inputs from EEC and flight environment such as flight altitude and Mach number. The simulator plays the role of bridging the engine model and EEC by converting the EEC control signals to digital value for engine model calculating and simulating the sensor signals for EEC sampling according to engine model output. What is more, it can be used to inject sensor faults to test the sensor fault diagnosis method and the fault-tolerance control ability of EEC. The fuel subsystem is employed to provide a working condition similar to the real-world for the actuators in fuel control loop; thus this HIL simulation is also called wet rig test. As shown in Figure 3, the fuel subsystem mainly consists of fuel tank, booster pump, flowmeter, fuel regulator, pips, and many sensors. Fuel regulator is one of the most important actuators in engine control system. It is a key component in fuel subsystem which needs to be evaluated, so the actual object is utilized in the test system. The rotating speed of rotor in aeroengine is simulated by a motor. In general, the whole HIL simulation system provides a flexible simulated testing environment for the verification and validation of engine control and health management units.

**5.2. HIL Simulation Experiments.** When developing the sensor measurement estimators based on SROS-ELM, the input-target pairs are firstly normalized into the interval [0 1]. Considering that the primary dynamic characteristic

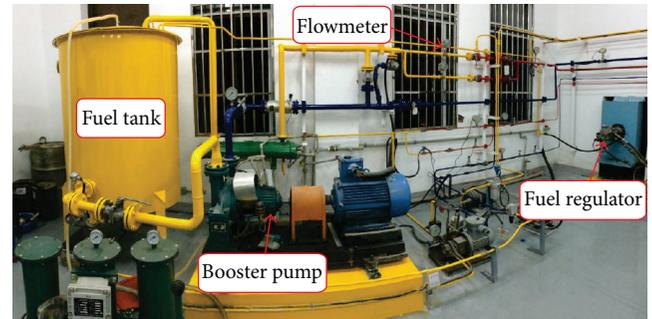


FIGURE 3: Fuel subsystem in the HIL simulation system.

of aircraft engine is consistent with that of a second-order inertial element, the value of embedded dimension is chosen to be  $m = 2$ . The threshold is selected as  $FC = 0.015$  after trial and error. In order to evaluate the estimation accuracy of SROS-ELM, an assessment criterion which represents the sensor estimation relative deviation is defined as follows:

$$RD = \left| \frac{\text{Estimated value} - \text{Actual value}}{\text{Actual value}} \right|. \quad (24)$$

Firstly, sensor faults under steady state are simulated for the six sensors, respectively. With the throttle lever randomly pushed and pulled at the height of  $H = 0$  and the Mach number of  $Ma = 0$ , a bias fault with the magnitude of 3% is injected into  $T_{45}$  at time of 53.52 s. Figure 4 shows the experiment results of fault diagnosis and sensor signal reconstruction. The faults of other sensors are simulated in the same way and the results are listed in Table 2 due to the limited space. It can be learned that when a sensor breaks down, there is an obvious difference between the sensor measurement and the estimated value of the SROS-ELM based estimator. Apparently, the estimated value follows the actual value more tightly than the measured value of the sensor with fault. Owing to the fault diagnosis and accommodation strategy, the erroneous sensor signal is abandoned and replaced by the estimated value. Besides, the final output

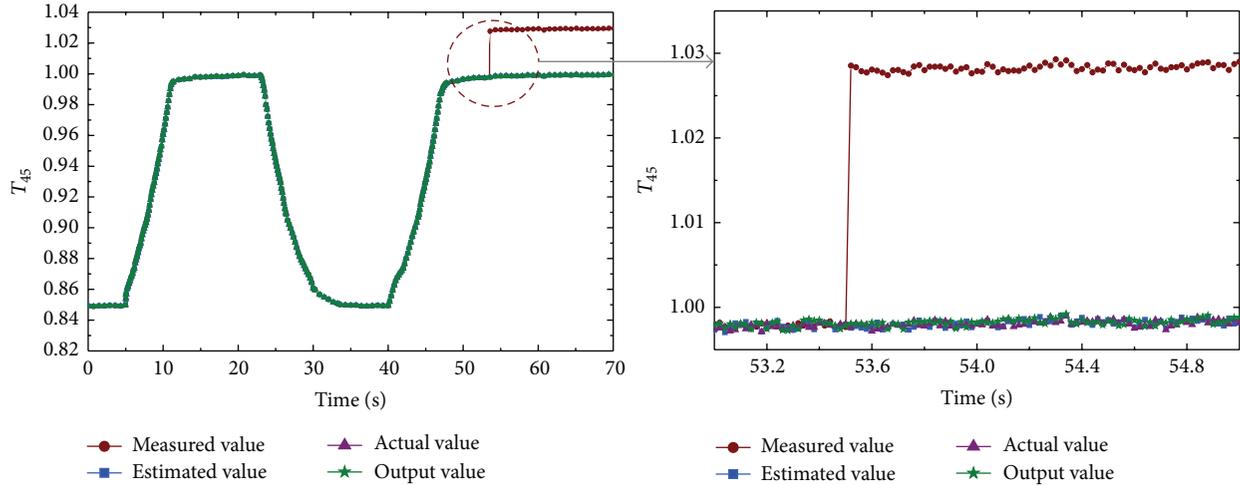
FIGURE 4: Fault diagnostic and signal reconstruction for  $T_{45}$  sensor under steady state.

TABLE 2: Experiment results of sensor measurements estimation under steady state.

Sensor fault under steady state	$RD_{\max}$						Estimation time/ms
	$N_h$	$N_l$	$T_3$	$T_{45}$	$P_3$	$P_{45}$	
$N_h$ sensor	0.0020	0.0021	0.0012	0.0018	0.0034	0.0023	0.6657
$N_l$ sensor	0.0017	0.0023	0.0015	0.0020	0.0036	0.0025	0.6654
$T_3$ sensor	0.0022	0.0021	0.0013	0.0016	0.0033	0.0021	0.6669
$T_{45}$ sensor	0.0016	0.0025	0.0016	0.0021	0.0038	0.0019	0.6660
$P_3$ sensor	0.0018	0.0028	0.0011	0.0019	0.0032	0.0020	0.6706
$P_{45}$ sensor	0.0016	0.0024	0.0014	0.0016	0.0035	0.0022	0.6693

value deviates from the actual value very slightly, which ensures the approximate correctness of the sensor signals provided to the engine controller. Therefore, the engine controller can output right commands according to the control law, thereby enabling the engine to work properly even in the case of a sensor fault. Moreover, from the estimation times in Table 2, it can be seen that all the estimation times are within 1 millisecond and far less than the engine control time step of 20 milliseconds, which completely meets the requirements of the engine control system for real-time performance.

In the second experiment, fault detection and signal reconstruction for sensors during a dynamic process using analytical redundancy techniques are validated. The results for  $N_l$  sensor fault are presented in Figure 5, and Table 3 tabulates the results for other sensor faults during a dynamic process. It is clear that when a sensor fault occurs, the outputs of the analytical redundancy unit follow the actual values rather than the wrong sensor measurements, which shows that the sensor fault is detected and indicated once the malfunction arises. From Table 3, it can be seen that the sensor measurement estimations have good accuracy, with the maximum relative deviations being about 0.53%. All in all, it shows that the designed analytical redundancy based on SROS-ELM is feasible and effective, thus the operational safety and engine reliability can be enhanced with this technique.

## 6. Conclusion

OS-ELM can learn data one-by-one or block-by-block online with fast training speed, but it is quite likely to suffer from singularity or ill-posedness problems. On the basis of OS-ELM, a modified algorithm, SROS-ELM, is proposed in this paper. It selectively updates output weights of neural networks according to the prediction accuracy and the norm of the output weight vector, tackling the problems of singularity and ill-posedness by regularization, and adopts a dual activation function in the hidden neurons combining neural networks and wavelet theory to enhance prediction capability. Experiments on the benchmark data sets show that SROS-ELM outperforms some other popular sequential learning algorithms in generalization performance. On the basis of that, a SROS-ELM-based aeroengine analytical redundancy technique is developed for sensor fault detection, isolation, and signal reconstruction. Furthermore, the developed technique is verified through HIL simulation on an experimental platform. The experiment results show the effectiveness and feasibility of the SROS-ELM-based analytical redundancy technique, which suggests that the technique is a promising approach to improving sensor reliability in an aeroengine system.

## Competing Interests

The authors declare that they have no competing interests.

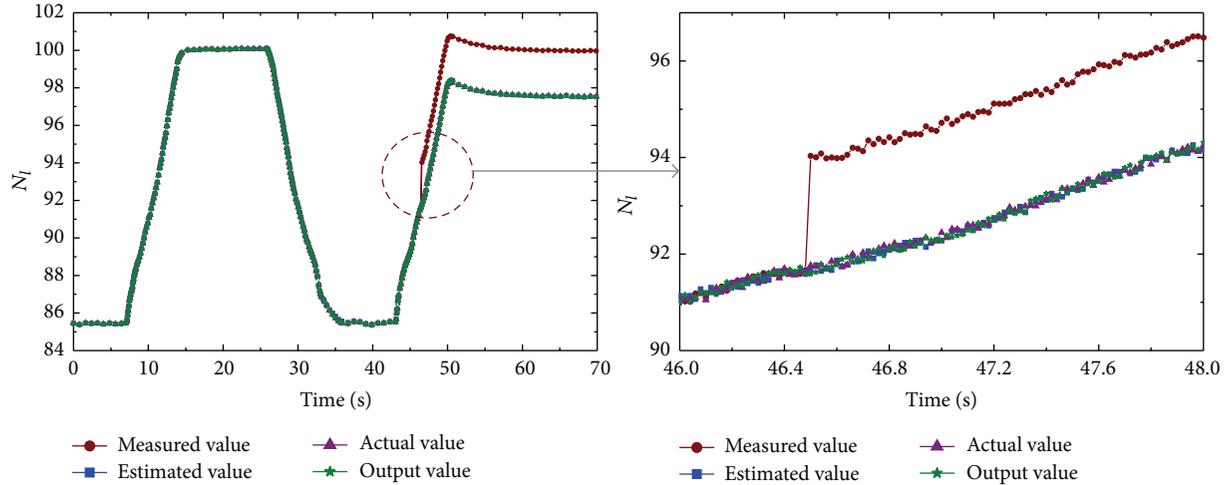


FIGURE 5: Fault diagnostic and signal reconstruction for  $N_I$  sensor during dynamic process.

TABLE 3: Experiment results of sensor measurements estimation during dynamic process.

Sensor fault during dynamic process	$RD_{\max}$						Estimation time/ms
	$N_h$	$N_I$	$T_3$	$T_{45}$	$P_3$	$P_{45}$	
$N_h$ sensor	0.0030	0.0022	0.0014	0.0021	0.0051	0.028	0.6122
$N_I$ sensor	0.0032	0.0028	0.0018	0.0026	0.0053	0.0026	0.6391
$T_3$ sensor	0.0035	0.0020	0.0016	0.0020	0.0045	0.0023	0.6384
$T_{45}$ sensor	0.0034	0.0026	0.0010	0.0019	0.0053	0.0022	0.6402
$P_3$ sensor	0.0038	0.0024	0.0018	0.0018	0.0051	0.0027	0.6431
$P_{45}$ sensor	0.0036	0.0030	0.0017	0.0021	0.0046	0.0029	0.6128

## Acknowledgments

This work was supported by the Funding of the Jiangsu Innovation Program for Graduate Education (no. KYLX\_0305) and the National Natural Science Foundation of China (no. 51176075).

## References

- [1] S. Garg, K. Schadow, W. Horn, H. Pfoertner, and I. Stiharu, *Sensor and Actuator Needs for More Intelligent Gas Turbine Engines*, National Aeronautics and Space Administration, Glenn Research Center, Cleveland, Ohio, USA, 2010.
- [2] S. Garg, "Propulsion controls and health management research at NASA glenn research center," Tech. Rep., NASA Glenn Research Center, 2002.
- [3] W. C. Merrill, "Sensor failure detection for jet engines using analytical redundancy," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 6, pp. 673–682, 1985.
- [4] R. E. Wallhagen and D. J. Arpasi, "Self-teaching digital-computer program for fail-operational control of a turbojet engine in a sea-level test stand," Contract RTOP 501-24, NASA, 1974.
- [5] R. Corley and H. Spang, "Failure detection and correction for turbofan engines," in *Proceedings of the Joint Automatic Control Conference*, ASME, IEEE, ISA, and SME, San Francisco, Calif, USA, 1977.
- [6] A. D. Pisano, "Failure indication and corrective action for turboshaft engines," *Journal of the American Helicopter Society*, vol. 25, pp. 36–42, 1980.
- [7] Z. N. Sadough Vanini, K. Khorasani, and N. Meskin, "Fault detection and isolation of a dual spool gas turbine engine using dynamic neural networks and multiple model approach," *Information Sciences*, vol. 259, pp. 234–251, 2014.
- [8] K. K. Botros, G. Kibrya, and A. Glover, "A demonstration artificial neural-networks-based data mining for gas-turbine-driven compressor stations," *Journal of Engineering for Gas Turbines and Power*, vol. 124, no. 2, pp. 284–297, 2002.
- [9] X.-H. Huang, "Sensor fault diagnosis and reconstruction of engine control system based on autoassociative neural network," *Chinese Journal of Aeronautics*, vol. 17, no. 1, pp. 23–27, 2004.
- [10] R. B. Joly, S. O. T. Ogaji, R. Singh, and S. D. Probert, "Gas-turbine diagnostics using artificial neural-networks for a high bypass ratio military turbofan engine," *Applied Energy*, vol. 78, no. 4, pp. 397–418, 2004.
- [11] S. O. T. Ogaji, R. Singh, and S. D. Probert, "Multiple-sensor fault-diagnoses for a 2-shaft stationary gas-turbine," *Applied Energy*, vol. 71, no. 4, pp. 321–339, 2002.
- [12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [13] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with

- random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [14] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.
- [15] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [16] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57–67, 2005.
- [17] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 6, pp. 2284–2292, 2004.
- [18] A. E. Hoerl and R. W. Kennard, "Ridge regression: applications to nonorthogonal problems," *Technometrics*, vol. 12, pp. 69–82, 1970.
- [19] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
- [20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3, JHU Press, 2012.
- [21] Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.
- [22] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.
- [23] G. Daqi and Y. Genxing, "Influences of variable scales and activation functions on the performances of multilayer feedforward neural networks," *Pattern Recognition*, vol. 36, no. 4, pp. 869–878, 2002.
- [24] K. Javed, R. Gouriveau, and N. Zerhouni, "SW-ELM: a summation wavelet extreme learning machine algorithm with a *priori* parameter initialization," *Neurocomputing*, vol. 123, pp. 299–307, 2014.
- [25] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Engineering Practice*, vol. 7, no. 5, pp. 643–653, 1999.
- [26] T. H. Bradley, B. A. Moffitt, D. N. Mavris, T. F. Fuller, and D. E. Parekh, "Hardware-in-the-loop testing of a fuel cell aircraft powerplant," *Journal of Propulsion and Power*, vol. 25, no. 6, pp. 1336–1344, 2009.
- [27] D. J. Rankin and J. Jiang, "A hardware-in-the-loop simulation platform for the verification and validation of safety control systems," *IEEE Transactions on Nuclear Science*, vol. 58, no. 2, pp. 468–478, 2011.

## Research Article

# Safety Assessment for Electrical Motor Drive System Based on SOM Neural Network

Linghui Meng, Peizhen Wang, Zhigang Liu, Ruichang Qiu, Lei Wang, and Chunmei Xu

*School of Electrical Engineering, Beijing Jiaotong University, Beijing Engineering Research Center of Electric Rail Transportation, Beijing 100044, China*

Correspondence should be addressed to Linghui Meng; 13810476488@163.com

Received 19 December 2015; Accepted 16 February 2016

Academic Editor: Wen Chen

Copyright © 2016 Linghui Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of the urban rail train, safety and reliability have become more and more important. In this paper, the fault degree and health degree of the system are put forward based on the analysis of electric motor drive system's control principle. With the self-organizing neural network's advantage of competitive learning and unsupervised clustering, the system's health clustering and safety identification are worked out. With the switch devices' faults data obtained from the dSPACE simulation platform, the health assessment algorithm is verified. And the results show that the algorithm can achieve the system's fault diagnosis and health assessment, which has a point in the health assessment and maintenance for the train.

## 1. Introduction

Urban rail train traction drive system is the key subsystem of the train, which is the guarantee for the train's safe and smooth running. However, urban rail train's motor drive system is a multivariable, nonlinear, strong coupling complex system. Its failure frequency and failure mode are intricate, mutual coupling interference, which seriously affected the safety and reliability of the train. However, the present process of fault identification and intervention "backwardness" determines its inevitable failure, which is limited for improving the safety. So the traditional urban rail train motor drive system needs online, real-time, fast health assessment and safety assessment. Safety control measures also should be taken in time to ensure the train's safe running.

Reference [1] introduced the idea of the machine learning and artificial intelligence to fault diagnosis for the motor drive system, which is effective for the intelligent algorithm to solve the complex system's fault diagnosis; [2] used the entropy weight multi-information algorithm to complete a comprehensive health assessment for the high-speed catenary, which plays a positive role in health monitoring and safety early warning for the catenary; [3, 4] applied the SOM neural network to the fault diagnosis and health evaluation for the steam turbine and the forest system, respectively, which

embodied the SOM network algorithm's unique advantage of fault diagnosis and health evaluation.

References [5–8] used the traditional methods of the physical aging damage mechanism of the system level for the damage assessment and reliability modeling of the train, which achieved the reliability assessment and safety prognosis for the train. References [9–16] used model-based and different intelligent methods such as the neural network and fuzzy logic to research on the diagnosis of power converters. In this paper, based on the above literatures, the control principle of train's motor drive system is analyzed to get the health characteristics factor. Then the data is obtained from the dSPACE fault simulation platform. At last, the self-organizing feature map network intelligent algorithm is realized by MATLAB2011b to complete the health assessment, and the results showed that the health degree and the safety can be calculated and assessed accurately, which can provide a positive guiding role in safety warning and maintenance for the train.

## 2. The Analysis of System Health Parameters

As can be seen from Figure 1, urban rail train's motor drive system mainly includes the traction inverter and traction motor which is controlled by the space vector modulation

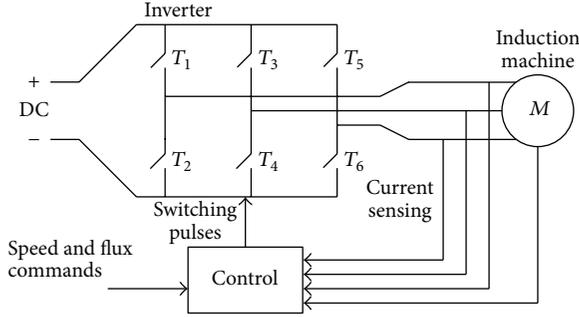


FIGURE 1: AC motor drive system.

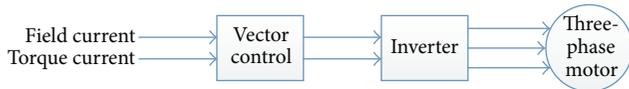


FIGURE 2: Vector control system.

algorithm. The basic principle of vector control is to measure and control the stator current vector of the induction motor and control field current and torque current of the induction motor, respectively, according to the principle of magnetic field orientation which is described in Figure 2, so as to realize the induction motor torque control.

Traction inverter converts the DC voltage required by the traction system to variable voltage and variable frequency three-phase AC power supply for three-phase induction motor. So the inverter's output voltage and current waveform quality directly affect the performance of motor drive system and also reflect the health status of the motor drive system. In addition, three-phase motor's output torque and speed also directly reflect the traction motor's traction ability, which also indirectly reflects the health status of the motor drive system. So the three-phase output voltage, three-phase output current, output torque, and speed are identified as the health variables to analyze the health characteristics of motor drive system.

### 3. Health Characteristic Parameters Extraction

**3.1. Data Preprocessing.** Lu Murphey et al. [1] presented a model-based fault diagnostics system which used the three-phase voltages and currents as feature signal for detecting and locating multiple classes of faults in an electric drive and achieved good results. Bowen [17] and Diallo et al. [18] researched on the fault diagnosis of inverter's open circuit with the three-phase currents and voltages. Based on the literature above, when the system is in a stable state, the health status of motor drive system can be reflected by three-phase voltage, three-phase current, and torque and speed, which are, respectively,  $V_{an}, V_{bn}, V_{cn}, I_a, I_b, I_c, T_e$ , and  $S$ , of which  $V_{an}, V_{bn}$ , and  $V_{cn}$  represent the root mean square value of three-phase voltage, respectively,  $I_a, I_b$ , and  $I_c$  represent the root mean square value of three-phase current, respectively, and  $T$  and  $S$  represent the average torque and speed, respectively.

Suppose the number of sampling voltages  $V_{in}$  ( $i = a, b, c$ ) in time  $t$  is  $N$ , and the samples are  $V_{in}^1, V_{in}^2 \dots V_{in}^N$ , respectively; then

$$V_{in} = \sqrt{\frac{(V_{in}^1)^2 + (V_{in}^2)^2 + \dots + (V_{in}^N)^2}{N}}. \quad (1)$$

The number of sampled currents  $I_i$  ( $i = a, b, c$ ) in time  $t$  is  $M$ , and the samples are  $I_i^1, I_i^2 \dots I_i^M$ , respectively; then

$$I_i = \sqrt{\frac{(I_i^1)^2 + (I_i^2)^2 + \dots + (I_i^M)^2}{M}}. \quad (2)$$

The number of types of sampled torque  $T_e$  in time  $t$  is  $P$ , and the samples are  $T_e^1, T_e^2 \dots T_e^P$ , respectively; then

$$T_e = \frac{T_e^1 + T_e^2 + \dots + T_e^P}{P}. \quad (3)$$

The number of sampled speeds  $S$  in time  $t$  is  $Q$ , and the samples are  $S^1, S^2 \dots S^Q$ , respectively; then

$$S = \frac{S^1 + S^2 + \dots + S^Q}{Q}. \quad (4)$$

**3.2. Health Characteristics Factor Calculation.** Suppose that the variables' rating standard values of the system in steady state are  $V_{an}, V_{bn}, V_{cn}, I_a, I_b, I_c, T_e$ , and  $S$ , respectively, and the values different between the actual state and standard state represent the health status of the system. The greater the deviation is, the worse the system health is. To the contrary, the smaller the deviation is, and the better the system health is. Therefore, the health characteristics factor can be described as

$$\begin{aligned} x_1 &= \left| \frac{V_{an} - V_{an}^*}{V_{an}^*} \right|, \\ x_2 &= \left| \frac{V_{bn} - V_{bn}^*}{V_{bn}^*} \right|, \\ x_3 &= \left| \frac{V_{cn} - V_{cn}^*}{V_{cn}^*} \right|, \\ x_4 &= \left| \frac{I_a - I_a^*}{I_a^*} \right|, \\ x_5 &= \left| \frac{I_b - I_b^*}{I_b^*} \right|, \\ x_6 &= \left| \frac{I_c - I_c^*}{I_c^*} \right|, \\ x_7 &= \left| \frac{T_e - T_e^*}{T_e^*} \right|, \\ x_8 &= \left| \frac{S - S^*}{S^*} \right|. \end{aligned} \quad (5)$$

#### 4. Health Assessment Based on Self-Organizing Feature Map Network

4.1. *The Principle of Self-Organizing Feature Map Network.* Self-organizing feature map (SOM) model [19] is a kind of competitive neural network, which introduces the self-organizing characteristics and is the same as competitive neural network by using unsupervised learning style. The difference is that the self-organizing map network can not only learn the distribution of input samples but also identify the topology of the input vector. Classifications are performed by multiple neurons interop with each other. Figure 3 is the network's structure diagram. The self-organizing map network contains two layers which are the input layer and the output layer. the input and output neurons are linked together by weight; at the same time, neighboring neurons are also linked by weight vector. The transfer function of the output neurons is mainly the linear function so the output value is the sum of the linear weighted input value. Suppose that the number of input neurons is  $m$ , the output neuron is  $n$ , weight is  $w_{ij}$ , and the output value of the output neurons  $Y_j$  will be

$$Y_j = f \left( \sum_i x_i w_{ij} \right). \quad (6)$$

Self-organizing feature map algorithm is a kind of clustering method without teachers, which can map any input mode into one-, two-, or multidimensional discrete graphics in the output layer and still keep its structure unchanged. The learning process can be described as follows: for each of the networks input health characteristics, it just adjusts parts of the weights, which make the weight vectors more close or far from the input vector. And the adjustment process is called the competitive learning. As the continuous learning, the weight vectors in the input space are separated and form a mode which represents the input space, respectively, which realizes the clustering of the health characteristics and health level.

4.2. *The Network Learning and Health Assessment.* Figure 4 is the flow chart of for the network learning and system health assessment. The healthier the system is, the smaller the deviation value is. Therefore the fault degree of the clustering neurons' output is lower, and the health degree is higher. The following is to explain how the clustering parameters are selected and the training steps of the network, respectively.

4.2.1. *The Selection of Clustering Parameters.* Parameters here are mainly the number of categories. When we use the self-organizing feature map network to cluster, the structure of the competition layer is set as  $3 \times 3$  and the clustering category of the number is 9. As for meticulous category the system health will be divided into many health levels that do not have much point. While it is not enough detailed, if only it is divided into two categories. As the input vector is 8 dimensions, so the network's input layer contains eight neuron nodes and the competitive layer contains nine neuron nodes. After the training, each input vector belongs to a competitive layer

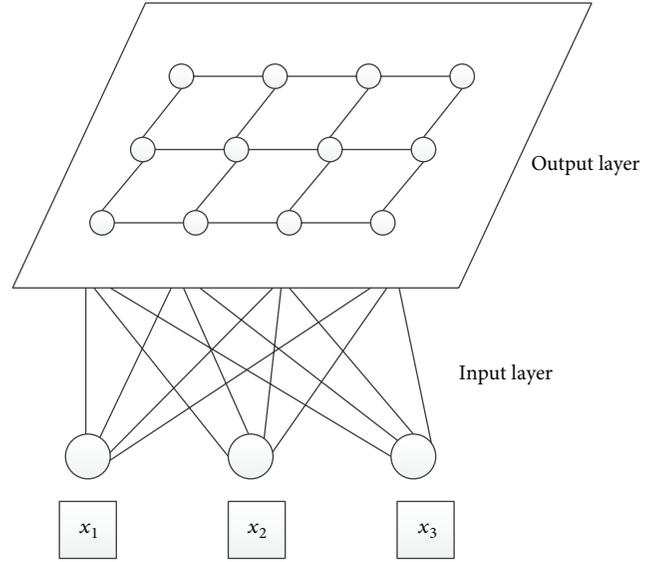


FIGURE 3: SOM neural network.

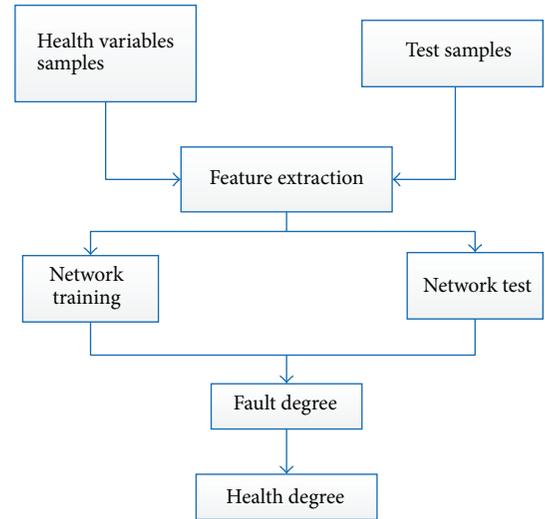


FIGURE 4: The flow chart of network learning and health assessment.

node. And the fault degree  $D$  of network's output ranges from 0.1 to 0.9. 0.1 means the system is in normal or safe state, while 0.9 means that the system has a major failure, and the number between them means that the system is in a state of degradation failure. The higher the fault degree is, the more serious the system failure is. Thus the system health degree  $H$  is defined as follows:

$$H = (1 + \varepsilon - D) \times 100\%. \quad (7)$$

Among them,  $D$  is the system's damage degree which is also the fault degree.  $H$  is the system's health degree, and  $\varepsilon$  is the correction coefficient of the system health and generally ranges from 0 to 0.1.

4.2.2. *Network Training.* SOM network's training steps [19] are as follows.

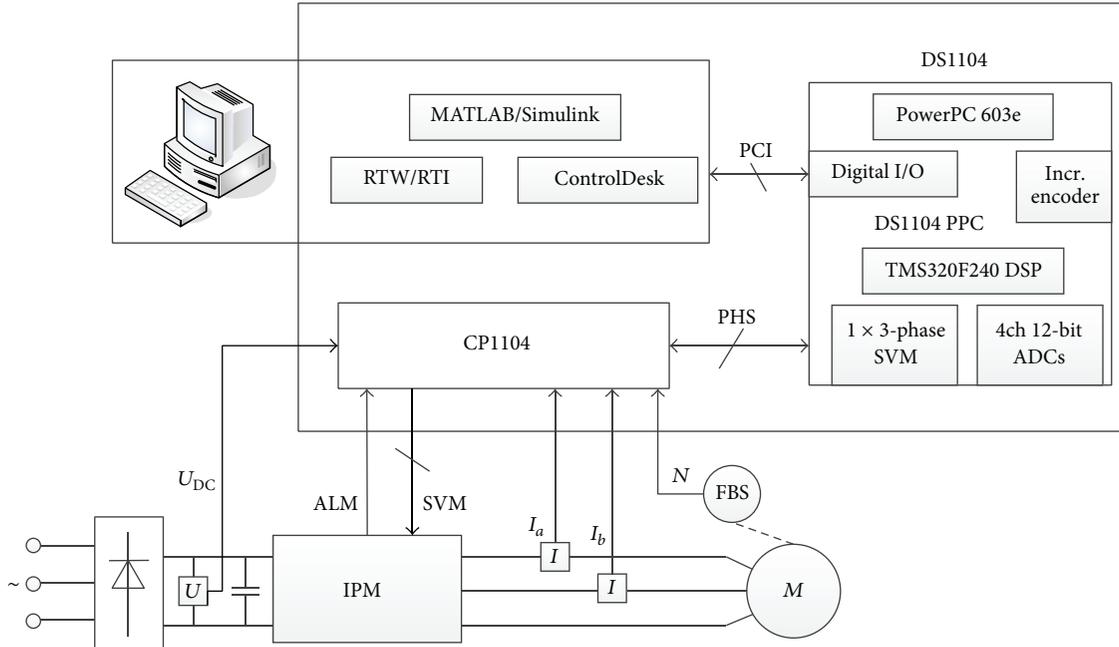


FIGURE 5: System health degradation simulation platform based on dSPACE.

*Set the Variable.* As the input sample vector is  $x = [x_1, x_2, \dots, x_8]$ , each sample is eight-dimensional vector.  $\omega_i(k) = [\omega_{i1}(k), \omega_{i2}(k), \dots, \omega_{im}(k)]$  is the weight vector between each of input nodes and output neurons.

*Initialization.* Small random values are used as initialized weights; then the input vector and weights are normalized as follows:

$$\begin{aligned} x' &= \frac{x}{\|x\|} \\ \omega'_i(k) &= \frac{\omega_i(k)}{\|\omega_i(k)\|}. \end{aligned} \quad (8)$$

*Network Input.* Samples do dot product with the weight vector, and the maximum of the output neurons will win the competition. As the sample and weight vectors have been normalized, so the minimum Euclidean distance can be worked out by calculating the maximum dot product:

$$D = \|x - \omega\|. \quad (9)$$

The neurons who get the minimum Euclidean distance will win as the winning neuron.

*Update the Weights.* For the neurons on the winning neuron topological neighborhood, Kohonen rule is applied to update:

$$\omega(k+1) = \omega(k) + \eta(x - \omega(k)). \quad (10)$$

Different distance functions can be used to determine the neighborhood; the commonly used Euclidean distance (dist) is the Manhattan distance (mandist), and so forth.

Update the learning rate  $\eta$  and the topological neighborhood and normalize the learned weights. Learning rates and the neighborhood sizes are adjusted according to the stage of sorting and adjustment.

*Determine If Convergence.* Determine whether the number of iterations reaches the maximum; if it did not reach the maximum number of iterations, then go to the third step or end the algorithm.

## 5. Simulation Experiments and Results Analysis

*5.1. Simulation Model and Test.* The motor drive fault simulation experiment is conducted by the dSPACE real-time simulation platform [17]. The platform mainly consists of the computer, dSPACE software system, dSPACE hardware control board DS1104 PPC, dSPACE hardware control panel CP1104, voltage and current sensors, signal detection unit, and an intelligent power module (IPM). The vector control system structure of the induction motor set up by dSPACE real-time simulation platform is shown in Figure 5.

Figure 5 shows the architecture of the real-time dSPACE simulation system. With the dSPACE software control model, the motor vector control model can be quickly converted into code and downloaded to the DS1104 hardware control panel with RTW/RTI. The control panel CP1104 converted the control and protection signal from TMS320F240DSP to a standard IPM signal for controlling the power switch. At the same time, the voltage, current, speed, and other signals are input into the CP1104 hardware control panel via the signal conditioning board and then fed to the input of the model to form a closed loop control. In this paper, a different number of

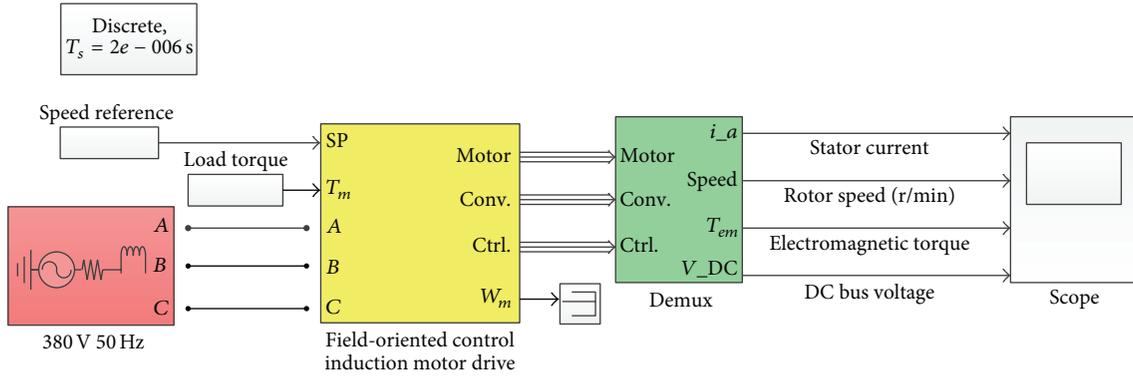


FIGURE 6: Top-level block diagram of simulation model.

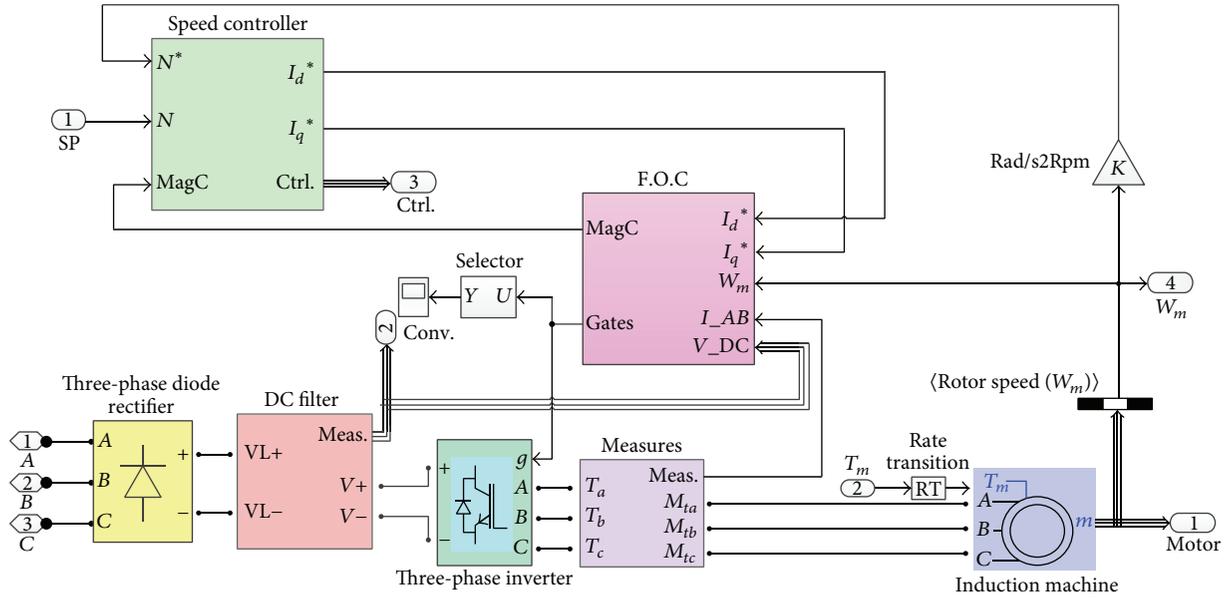


FIGURE 7: Block diagram of RFOC model.

TABLE 1: System health degradation state table.

Number	Condition
1	Healthy
2	$T_1$ OC
3	$T_2$ OC
4	$T_1, T_2$ OC
5	$T_1, T_3$ OC
6	$T_1, T_2, T_3$ OC
7	$T_1, T_2, T_4$ OC
8	$T_1, T_2, T_3, T_4$ OC
9	$T_1, T_2, T_3, T_4, T_5$ OC
10	$T_1, T_2, T_3, T_4, T_5, T_6$ OC

fault switching devices are triggered to simulate the system's different fault degree. And the switch's open circuit faults are simulated through the blockade of the pulses by dSPACE software. Table 1 is the system health degradation state table.

Figure 6 is induction motor simulation model which is controlled by the rotor field-oriented vector. The step length of simulation model is fixed in discrete simulation algorithm. The simulation step size is usually chosen as one percent of the switching cycle. In this simulation the switching frequency is 5 kHz and the simulation step  $T_s$  is  $2e - 6$  s.

As shown in Figure 7, the simulation model includes seven modules, of which the main circuit is a typical LCI structure, including a three-phase diode rectifier, DC filter unit, three-phase inverter, measuring unit, and the motor model. Three-phase 380 V/50 Hz AC power runs through a rectifier and a DC link filter and then from the three-phase AC voltage inverter to the induction motor. The speed control system includes a speed control module and a field-oriented control module.

5.2. Health Degradation Simulation and Calculation. Figure 8 is the motor drive system's fault simulation test platform. DC voltage is about 513~537 V which is rectified from 380 V

TABLE 2: Degradation data tables of system health state.

Number	Condition	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
1	Healthy	0.05	0.03	0.02	0.03	0.04	0.02	0.01	0.02
2	$T_1$ OC	0.11	0.1	0.13	0.09	0.16	0.08	0.15	0.1
3	$T_1, T_2$ OC	0.23	0.2	0.19	0.16	0.18	0.24	0.22	0.17
4	$T_1, T_2, T_3$ OC	0.26	0.33	0.28	0.32	0.31	0.34	0.33	0.35
5	$T_1, T_2, T_3, T_4$ OC	0.41	0.45	0.38	0.42	0.46	0.47	0.42	0.4
6	$T_1, T_2, T_3, T_4, T_5$ OC	0.71	0.75	0.68	0.72	0.66	0.57	0.72	0.74
7	$T_1, T_2, T_3, T_4, T_5, T_6$ OC	0.81	0.85	0.78	0.82	0.86	0.77	0.82	0.84

TABLE 3: System health assessment results.

Number	Condition	Fault degree	Health degree
1	Healthy	0.1	90%
2	$T_1$ OC	0.2	80%
4	$T_1, T_2$ OC	0.4	60%
6	$T_1, T_2, T_3$ OC	0.5	50%
8	$T_1, T_2, T_3, T_4$ OC	0.6	40%
9	$T_1, T_2, T_3, T_4, T_5$ OC	0.7	30%
10	$T_1, T_2, T_3, T_4, T_5, T_6$ OC	0.9	10%



FIGURE 8: Motor drive system fault simulation test platform.

three-phase AC voltage by the uncontrollable rectifier. Then the DC voltage is transformed to three-phase AC by the IPM module, and the inverter output power is 2.2 kW of the induction motor load. From the dSPACE simulation platform we can get eight state variables. And they can be acquired and preprocessed as the health degrees as in Table 2.

Different switching devices faults are triggered to simulate different fault degrees of motor drive system. The health degraded data in Table 2 is input into the self-organizing feature map network with MATLAB2011b. The network's connection weights, weight distance, and position are shown in Figures 9~11, and the health assessment results are in Table 3. From the table, we can see that, with the increase of the number of fault switching devices, the system fault degree increased, and the health degree reduced. What is more, when the number of faults is one, the health degree is still 80 percent, which means the system can still operate in a degraded state. When the number of faults is two, the health degree is 60 percent which drops 20 percent compared with

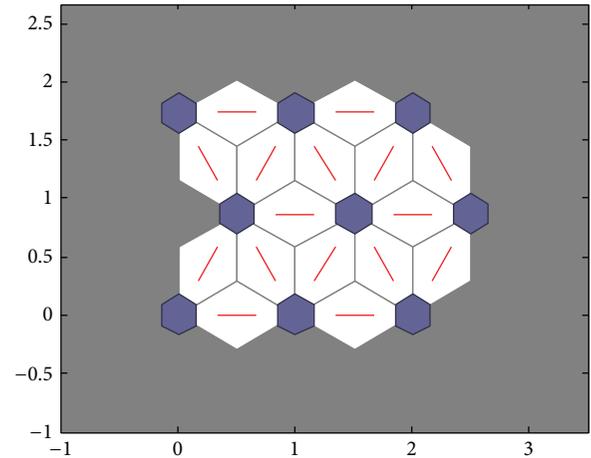


FIGURE 9: SOM neighbor connections.

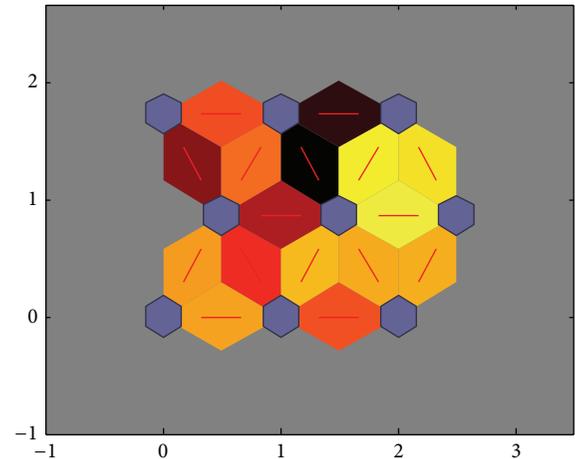


FIGURE 10: SOM neighbor weight distances.

one fault device. But it is still above 50 percent which is higher than three or more fault devices. So we must take tolerance control measures or maintenance action to keep the system safe before two fault devices as soon as possible in order to avoid property loss or casualties.

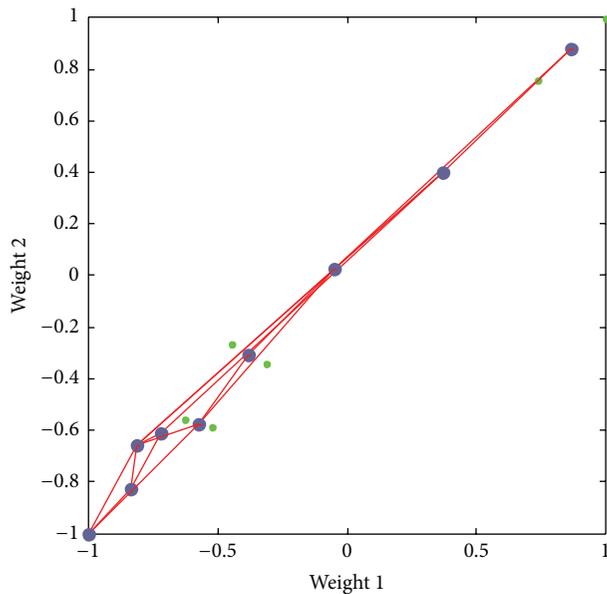


FIGURE 11: SOM weight positions.

## 6. Conclusions

This paper extracted the health variables of the motor drive system by analyzing the control principles and fault mechanism firstly. Then they are preprocessed to get the health degree. With the self-organizing feature map network's unsupervised and autonomous learning characteristics, the system fault is clustered and recognized quickly through the competition clustering. The fault of the switching device is taken as example to validate the algorithm by the simulation experiment and demonstration. Finally the health degree is put forward to complete the system's health assessment, which has an important guiding significance for railway motor drive system's safety assessment and maintenance.

Of course, due to the limited time and ability, this paper just put forward a preliminary health assessment scheme and algorithm. Later there is a need for research of the capacitance's aging damage, electrical insulation failure, sensor failure, and also the analysis of different failure mode effect on the system in order to realize the online health assessment and safety early warning for the train's safety, reliability, and stability.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

The work was supported by the National Natural Science Foundation of High Speed Rail Joint Funds (U1134204).

## References

- [1] Y. Lu Murphey, M. Abul Masrur, Z.-H. Chen, and B. Zhang, "Model-based fault diagnosis in electric drives using machine learning," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 3, pp. 290–303, 2006.
- [2] H.-B. Cheng, Z.-Y. He, H.-T. Hu, X.-Q. Mu, B. Wang, and Y.-X. Sun, "Comprehensive evaluation of health status of high-speed railway catenaries based on entropy weight," *Journal of the China Railway Society*, vol. 36, no. 3, pp. 19–24, 2014.
- [3] P. Zhi-Song, W. Qiong, N. Gui-Qiang, and H. Gu-Yu, "A SOM-based of fault diagnosis for WAN," in *Proceedings of the International Conference on Industrial and Information Systems (IIS '09)*, pp. 207–210, Haikou, China, April 2009.
- [4] M. Shi, C. Zhao, and Z. Guo, "Forest health assessment based on self-organizing map neural network," *Chinese Journal of Ecology*, vol. 30, no. 6, pp. 1295–1303, 2011.
- [5] L.-H. Meng, Z.-G. Liu, L.-J. Diao, C.-M. Xu, and L. Wang, "Evaluation of reliability of urban rail train traction inverter system," *Journal of the China Railway Society*, vol. 36, no. 9, pp. 34–38, 2014.
- [6] H. Wang, Y. Wang, and C. Xie, "Reliability modeling and assigning for CRH2 electric multiple unit," *Journal of the China Railway Society*, vol. 31, no. 5, pp. 108–112, 2009.
- [7] X. Lu, Z. Liu, and M. Shen, "Research on the damage model of electrical locomotives traction subsystem based on the stress damage," *Journal of Beijing Jiaotong University*, vol. 33, no. 6, pp. 13–16, 2009.
- [8] M. Molaei, H. Oraee, and M. Fotuhi-Firuzabad, "Markov model of drive-motor systems for reliability calculation," in *Proceedings of the International Symposium on Industrial Electronics (ISIE '06)*, pp. 2286–2291, Québec, Canada, July 2006.
- [9] J.-S. Wang, S.-X. Li, and J. Gao, "SOM neural network fault diagnosis method of polymerization kettle equipment optimized by improved PSO algorithm," *The Scientific World Journal*, vol. 2014, Article ID 937680, 12 pages, 2014.
- [10] B. Akin, S. Choi, U. Orguner, and H. A. Toliyat, "A simple real-time fault signature monitoring tool for motor-drive-embedded fault diagnosis systems," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1990–2001, 2011.
- [11] J. M. Bossio, C. H. De Angelo, G. R. Bossio, and G. O. García, "Fault diagnosis on induction motors using Self-Organizing Maps," in *Proceedings of the 9th IEEE/IAS International Conference on Industry Applications (INDUSCON '10)*, pp. 1–6, Sao Paulo, Brazil, November 2010.
- [12] R. L. De Araujo Ribeiro, C. B. Jacobina, E. R. C. Da Silva, and A. M. N. Lima, "Fault detection of open-switch damage in voltage-fed PWM motor drive systems," *IEEE Transactions on Power Electronics*, vol. 18, no. 2, pp. 587–593, 2003.
- [13] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques," *IEEE Transactions on Power Electronics*, vol. 47, no. 5, pp. 994–1004, 2002.
- [14] 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.
- [15] Y. L. Murphey, M. A. Masrur, Z. Chen, and B. Zhang, "Model-based fault diagnosis in electric drives using machine learning," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 3, pp. 290–303, 2006.

- [16] J. O. Estima and A. J. M. Cardoso, "A new approach for real-time multiple open-circuit fault diagnosis in voltage-source inverters," *IEEE Transactions on Industry Applications*, vol. 47, no. 6, pp. 2487–2494, 2011.
- [17] C. U. Bowen, "Simulation study for inverter-fed motor drive system under fault conditions," *Electric Machines and Control*, vol. 11, no. 6, pp. 578–583, 2007.
- [18] 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.
- [19] C. Delpha, D. Diallo, E. H. B. Mohamed, and C. Marchand, "Pattern recognition for diagnosis of inverter FED induction machine drive: a step toward reliability," in *Proceedings of the IET Colloquium on Reliability of Electromagnetic Systems*, pp. 1–5, Paris, France, May 2007.

## Research Article

# Monitoring of Distillation Column Based on Indiscernibility Dynamic Kernel PCA

**Qiang Gao, Yong Chang, Zhen Xiao, and Xiao Yu**

*Tianjin Key Laboratory for Control Theory & Applications in Complicated Systems, Tianjin University of Technology, Tianjin 300384, China*

Correspondence should be addressed to Yong Chang; [chang133122332@sina.com](mailto:chang133122332@sina.com)

Received 31 August 2015; Accepted 10 January 2016

Academic Editor: Wen Chen

Copyright © 2016 Qiang Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at complicated faults detection of distillation column industrial process, it has faced a grave challenge. In this paper, a new indiscernibility dynamic kernel principal component analysis (I-DKPCA) method is presented and applied to distillation column. Compared with traditional statistical techniques, I-DKPCA not only can capture nonlinear property and dynamic characteristic of processes but also can extract relevant variables from all the variables. Applying this new method to distillation column process (a hardware-in-the-loop simulation system), the results prove the proposed method has great advantages, that is, lower missing rate and higher detection rate for the faults, compared with KPCA and DPCA.

## 1. Introduction

With the emergence and development of industrial 4.0, modern industrial processes are more complicated in both structure and automatic degree. The safety and reliable issues about the industrial processes have become the most critical problems for system design [1–14]. To avoid abnormal accidents and losses, the process monitoring problem has become a severe topic in the area of process control. Among different process models, multivariate statistical process monitoring provides a data-driven framework for monitoring the industrial process. With the wide use of smart sensors, a large amount of data is collected in industrial processes; process information can be extracted directly from the huge amounts of the process data without considering complicated system models by data-driven methods, which lead to data-driven methods that have attracted much attention in the recent research works. Principal component analysis (PCA) is one of most widely used models in statistical process monitoring [15–20].

PCA is a basic multivariate statistical method which can extract useful information from large amount of process data by reducing dimensions. And the process data is divided into

systematic part that reflects normal data change and noisy part that reflects the variation of noise. Hotelling's  $T^2$  statistic and SPE statistic are used for chemical process monitoring to detect the changes of process variation in the principal component subspace and residual, respectively. And it is applied to petroleum and chemical industry widely. The conventional PCA has been well performed in only steady-state linear processes. However, dynamic and nonlinear characteristics are widespread in many industrial processes.

To handle nonlinearity, a lot of methods have been proposed (Kramer, 1991; Dong and McAvoy, 1996; and Chiang et al., 2001), such as neural network PCA and kernel PCA (KPCA) [23–29]. And neural network PCA needs training to determine the number of principal components; KPCA was developed to overcome this problem. The basic idea of KPCA is that the mapped data are analyzed by conventional PCA method in feature space. The traditional PCA is a static method. It is difficult to acquire the serial correlation of data [30]. But industrial processes are dynamic, which will lead to fault missing. To handle this problem, the dynamic characteristics should be taken into consideration when developing a monitoring model [31]. Ku et al. developed dynamic PCA which takes into account serial correlation in

the data by augmenting each observation vector with the previous  $s$  observations. After years of research, DPCA has been applied to many fields [30, 32, 33].

In this paper, to improve the PCA, we propose a new nonlinear dynamic process monitoring method based on indiscernibility dynamic kernel PCA (I-DKPCA). The proposed method can not only capture nonlinear property and dynamic characteristic of industrial processes but also simplify the process data by extracting valid data. We compare the results of DPCA, KPCA, and I-DKPCA for detecting various faults in distillation column industrial process.

The remaining sections of this paper are organized as follows. Section 2 explains the new I-DKPCA algorithm in detail. In Section 3, we applied the four methods to distillation column. At last conclusions are drawn in Section 4.

## 2. Algorithm of New I-DKPCA

There are strong dynamic and nonlinear characteristics in the industrial processes; the I-DKPCA was a nonlinear dynamic method and proposed aiming at these two characteristics. For the I-DKPCA algorithm, some faults may not affect all the operating and process variables. To a given fault, some variables are not influenced. The proposed indiscernibility dynamic kernel PCA finds the variables which are affected by the faults severely, and these variables are extracted to form new sample matrix and test matrix. Therefore the proposed method has higher sensitivity and accuracy for process monitoring. The new method consists of three parts.

**2.1. Indiscernibility and the Cross-Degree ( $\mu$  and  $\eta$ ).** In process industry, for a complex system, there are a lot of process and operation variables; these variables are collected and stored; in the traditional multivariate statistical process monitoring, the selection of control variable often considered all the process variables, which caused a lot of inconvenience for process monitoring; for an actual fault, only a few variables are affected in the process. So a new dynamic kernel principal component analysis method is put forward in this paper. Two parameters are proposed, the indiscernibility degree and the degree of cross. The new method can get rid of irrelevant variables, reduce the data dimension, simplify the calculation algorithm, and improve the efficiency and accuracy of fault diagnosis.

The train samples  $Z = [z_1, z_2, \dots, z_{m'}]$ ,  $z_i \in R^{n'}$  are gotten from normal process; the average for each variable is as follows:

$$\hat{z}_i = \frac{1}{n'} \sum_j^{n'} z_{ji}, \quad i = 1, 2, \dots, m'. \quad (1)$$

The test samples  $Z' = [z'_1, z'_2, \dots, z'_{m'}]$ ,  $z'_i \in R^n$  are gotten from abnormal process; the average for each variable is as follows:

$$\hat{z}'_i = \frac{1}{n} \sum_j^n z'_{ji}, \quad i = 1, 2, \dots, m', \quad (2)$$

where  $n'$  is the number of samples and  $m'$  is the number of variables.

To determine the threshold,

$$\delta_i = \frac{\hat{z}_i + \hat{z}'_i}{2}, \quad (3)$$

where the threshold value is to distinguish the abnormal data of the train data and the abnormal data of the fault data; if there are data which are beyond (below) the threshold in the train data, those data would be considered the abnormal data. If there are data which are below (beyond) the threshold in the fault data, these data would also be considered the abnormal data. Take all the abnormal data of train data and fault data in a set of fault samples; the wrong points are called samples of fault point, shown as follows:

$$U_i = \{u_{i1}, u_{i2}, \dots, u_{ik}\}, \quad 0 \leq k \leq 2n, \quad i = 1, 2, \dots, m', \quad (4)$$

where  $m'$  is the number of variables.  $k$  is the number of wrong points; for different variable,  $k$  is different.

The parameter of the indiscernibility degree which is proposed in this paper is represented by  $\mu_i$  as follows:

$$\mu_i = \sum_j^k \left| u_{ij} - \frac{\hat{z}_i + \hat{z}'_i}{2} \right|, \quad u_{ij} \in U_i. \quad (5)$$

Parameter of the degree of cross is the ratio of the number of wrong points and the number of all samples in each variable and is represented by  $\eta_i$  as follows:

$$\eta_i = \frac{k}{2n} \times 100\%, \quad i = 1, 2, \dots, m'. \quad (6)$$

For each known fault, we need to set a limit of  $\mu$  and  $\eta$  to choose the variables; in this paper, the author obtained the best value of  $\mu$  and  $\eta$  by many simulation results; in general, as the value of  $\mu$  and  $\eta$  is smaller, the effort is better, and get rid of irrelevant variables, and make the monitor data more concise. Because of reducing the irrelevant variables and simplifying the computation, according to the selected variables to monitor the production processes, the effect of diagnosis is better.

**2.2. Dynamic Characteristic Analysis.** To consider the dynamic of the new data  $X = [x_1, x_2, \dots, x_m]$ ,  $x_i \in R^n$ , the PCA methods can be extended to take the serial correlations into account by augmenting each observation vector with the previous  $s$  observations and stacking the data matrix in the following manner [3, 34]:

$$X(s) = \begin{bmatrix} x_t^T & x_{t-1}^T & \cdots & x_{t-s}^T \\ x_{t-1}^T & x_{t-2}^T & \cdots & x_{t-s-1}^T \\ \cdots & \cdots & \cdots & \cdots \\ x_{t+s-n}^T & x_{t+s-n-1}^T & \cdots & x_{t-n}^T \end{bmatrix}, \quad (7)$$

where  $x_t$  is the  $m$ -dimensional observation vector in the training set at the time instance  $t$ . As shown in Figure 1. The number of lags  $s$  is selected by [32, 35, 36]. The DPCA can get rid of the correlation of the data in some degree and improve the accuracy of diagnosis.

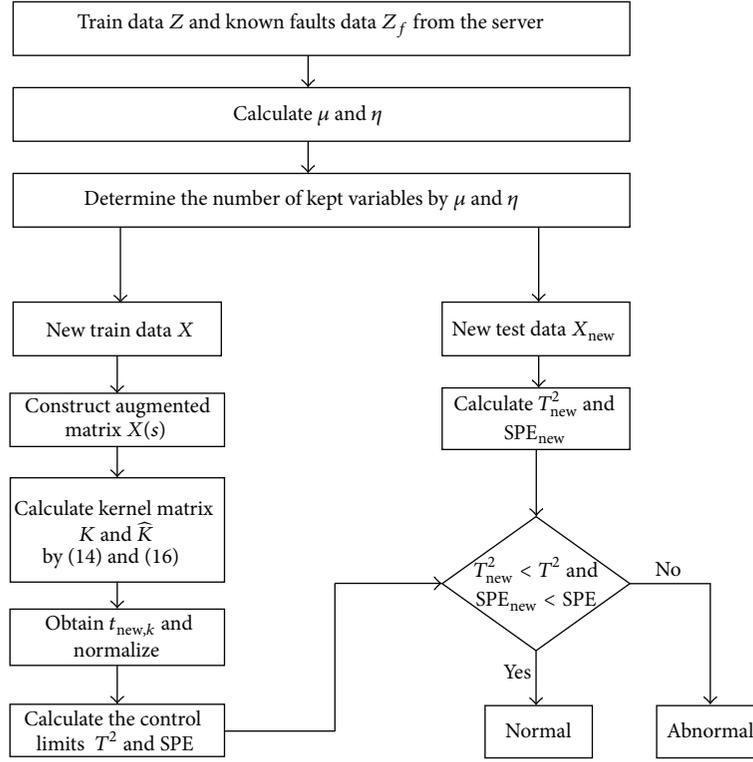


FIGURE 1: Online monitoring of I-DKPCA.

2.3. *Kernel Principal Component Analysis.* Assuming the new augmented matrix  $X(s)$  is mapped nonlinearly into a high dimensional feature space  $\Phi : R^n \rightarrow F$ . The original data  $x_j$  become  $\Phi(x_j)$  after mapping. Suppose the data mapped into the feature space is centered; that is,

$$\sum_{k=1}^N \Phi(x_k) = 0; \quad (8)$$

the covariance matrix  $C^F$  in the feature space is defined as

$$C^F = \frac{1}{N} \sum_{j=1}^N \Phi(x_j) \Phi(x_j)^T. \quad (9)$$

To perform PCA in the feature space we make eigenvalues ( $\lambda$ ) and eigenvectors ( $\nu$ ) that satisfy

$$\lambda \nu = C^F \nu, \quad (10)$$

where  $\lambda > 0$  and  $\nu \in F \setminus \{0\}$  is the eigenvector of correspondent  $\lambda$ . Therefore, for  $\lambda \neq 0$ , solution  $\nu$  can be regarded as a linear combination of  $\Phi(x_1), \dots, \Phi(x_n)$ ; that is,

$$\nu = \sum_{i=1}^N \alpha_i \Phi(x_i). \quad (11)$$

Then, by multiplying  $\Phi(x_j)$  with the left of both sides of (10), we obtain

$$\lambda \langle \Phi(x_k), \nu \rangle = \langle \Phi(x_k), C^F \nu \rangle. \quad (12)$$

Substituting (6) and (9) into (10), we obtain

$$\begin{aligned} & \lambda \sum_{i=1}^N \alpha_i \langle \Phi(x_k), \Phi(x_i) \rangle \\ &= \frac{1}{N} \sum_{i=1}^N \alpha_i \left\langle \Phi(x_k), \sum_{j=1}^N \Phi(x_j) \right\rangle \langle \Phi(x_j), \Phi(x_i) \rangle. \end{aligned} \quad (13)$$

Defining a kernel matrix  $K \in R^{N \times N}$ ,

$$[K]_{ij} = k_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle. \quad (14)$$

Inserting (12) into (11), (7) can be represented by the following simple form:

$$\lambda \alpha = \left( \frac{1}{n} \right) K \alpha. \quad (15)$$

For  $\forall \lambda > 0$ ,  $\alpha = [\alpha_1, \dots, \alpha_N]^T$ . The kernel matrix,  $K$ , is centered in the feature space using the formula

$$\widehat{K} = K - I_N K - K I_N + I_N K I_N, \quad (16)$$

where  $I_N$  is an  $(n \times n)$  matrix in which every element is equal to  $1/n$ . To normalize  $\alpha_1, \dots, \alpha_n$ ,

$$\langle \nu_k, \nu_k \rangle = 1, \quad k = 1, \dots, n. \quad (17)$$

Taking  $v_k = \sum_{i=1}^N \alpha_i^k \Phi(x_i)$  into (15),

$$\begin{aligned} 1 &= \left\langle \sum_{i=1}^N \alpha_i^k \Phi(x_i), \sum_{j=1}^N \alpha_j^k \Phi(x_j) \right\rangle \\ &= \sum_{i=1}^N \sum_{j=1}^N \alpha_i^k \alpha_j^k \langle \Phi(x_i) \Phi(x_j) \rangle = \sum_{i=1}^N \sum_{j=1}^N \alpha_i^k \alpha_j^k K_{ij} \\ &= \langle \alpha_k, K \alpha_k \rangle = \lambda_k \langle \alpha_k, \alpha_k \rangle. \end{aligned} \quad (18)$$

By projecting  $\Phi(x)$  onto eigenvectors  $v_k$  in the feature space, principal components are extracted:

$$t_k = \langle v_k, \Phi(x) \rangle = \sum_{i=1}^N \alpha_i^k \langle \Phi(x_i), \Phi(x) \rangle. \quad (19)$$

For  $k = 1, \dots, p$ ,  $p$  is the number of principal components in KPCA. By (17), we obtain a score vector  $t_{\text{new},k} = [t_{\text{new},1}, t_{\text{new},2}, \dots, t_{\text{new},p}]^T$  for  $x_{\text{new}}$ .

At present, a number of kernel functions are used in KPCA. In our study, only a Gaussian function is selected for it is widely used. Consider

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{\sigma^2}\right). \quad (20)$$

After obtaining  $t_k$ , two complementary multivariate control charts can be applied to process fault diagnosis. The first is Hotelling's  $T^2$  chart which can monitor the variation in the space of the principal components, and it is defined as

$$T^2 = [t_1, \dots, t_p] \Lambda^{-1} [t_1, \dots, t_p]^T, \quad (21)$$

where  $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  is the diagonal matrix consisting of the eigenvalues of  $K$ . The upper control limit based on  $T^2$  is obtained using  $F$  distribution and is given by

$$T_\alpha^2 = \frac{p(n^2 - 1)}{n(n - 1)} F_{p, n-p; \alpha}, \quad (22)$$

where  $F_{p, n-p; \alpha}$  is the upper  $\alpha$  critical point of  $F$  distribution with  $(p, n - p)$  degrees of freedom.

The SPE chart represents Euclidean distance from the model space and it is defined as

$$\text{SPE} = \|\phi(x) - \phi_p(x)\|^2 = \sum_{i=1}^n t_i^2 - \sum_{i=1}^p t_i^2, \quad (23)$$

where  $\phi_p(x)$  is the reconstructed feature vector with  $p$  principal components in the feature space;  $\phi(x)$  is identical to  $\phi_p(x)$  if  $p$  equals  $n$ . The details of formula (21) can be found in [27].

Assuming that the prediction errors are normally distributed, the upper control limit on the SPE at significance level  $\alpha$  is obtained using  $\chi^2$  distribution and is given by

$$\text{SPE}_\alpha = g \chi_h^2, \quad (24)$$

where  $g = b/2a$ ,  $h = 2a^2/b$ , and  $a$  and  $b$  are the mean and variance of the SPE at each time interval.

**2.4. I-DKPCA Based Monitoring.** Figure 1 shows the flow-chart used for the necessary calculation of monitoring based on I-DKPCA.

The first step is setting up model by off-line training:

- (1) Acquire train data  $Z = [z_1, z_2, \dots, z_{m'}]$ ,  $z_i \in R^n$  and known faults data  $Z_f$ , where  $n$  is the number of samples and  $m'$  is the number of variables.
- (2) Calculate  $\mu$  and  $\eta$  by (5) and (6).
- (3) Determine the number of kept variables.
- (4) Construct new train data  $X$ .
- (5) Construct augmented matrix  $X(s)$  by (7).
- (6) Apply KPCA to augmented matrix  $X(s)$ .
- (7) Calculate  $T_\alpha^2$  and  $\text{SPE}_\alpha$ .

The second step is on-line monitoring:

- (1) Acquire the new test data  $X_{\text{new}}$  by first step.
- (2) Calculate  $T_{\text{new}}^2$  and  $\text{SPE}_{\text{new}}$ .
- (3) If  $T_{\text{new}}^2 < T_\alpha^2$  and  $\text{SPE}_{\text{new}} < \text{SPE}_\alpha$  the process is normal; else the process is abnormal.

### 3. Fault Diagnosis of Distillation Column

Distillation column is very important in the field of chemical industry, which has been widely applied to chemical and oil refining enterprises. Mixed liquid is separated into various components by distillation column. The principle of distillation process is separating mixture by the different features such as temperature and volatile liquid separation (boiling point) so as to achieve the purpose of purification.

Distillation column is an indispensable device of chemical process and oil refining enterprise; once faults appear, they will bring great losses to enterprise, so the fault detection and diagnosis of distillation column are important to chemical production. There are many factors which impact the distillation column production operation and product quality; distillation column is a complex system of more than one parameter in the process of distillation. Physical phenomena occur, such as flow and heat transfer; there are many practical difficulties to truly understand the actual process, and, in the development of early fault diagnosis, many scholars put forward the mathematical model to describe the process of actually solving the problem of fault diagnosis of the distillation column, mainly divided into static kind and dynamic kind of mathematical simulation model. These models have a lot of defects; however, they do not meet the needs of diagnosis.

Mathematical model of distillation column cannot be instead of the actual process really; before the modeling, the researchers have to come up with some assumptions to simplify model, because it is difficult to solve complicated mathematical model, which leads to vast differences between model and actual process. So, considering these problems, multivariate statistical process monitoring provides a data-driven framework for monitoring the industrial process

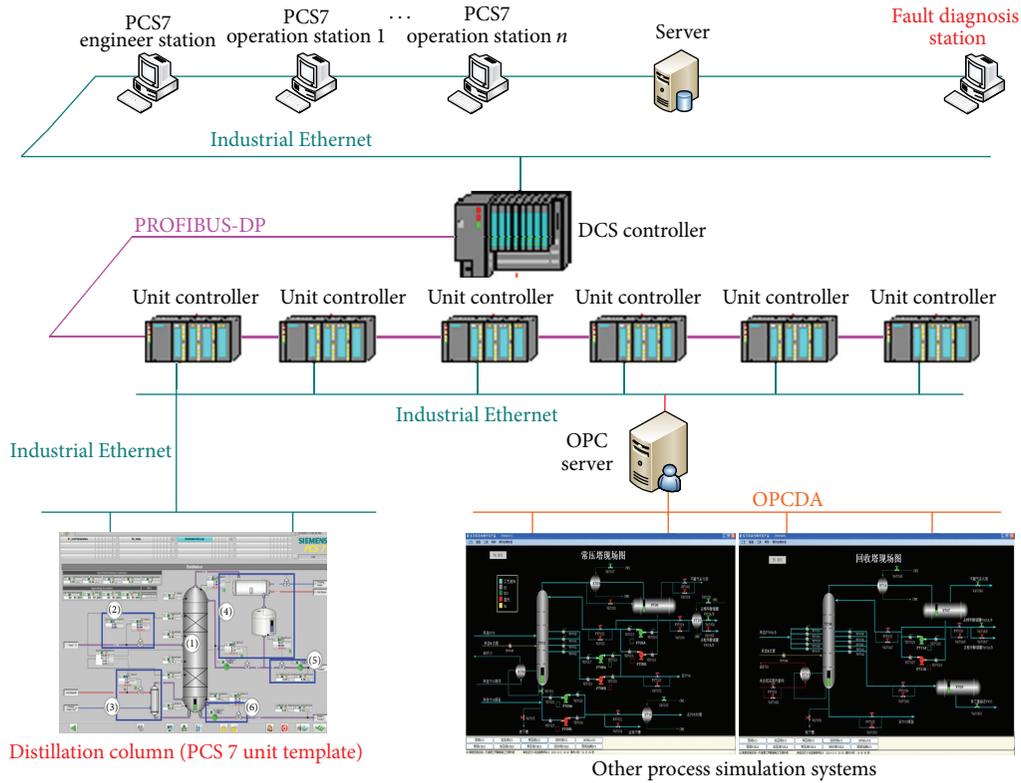


FIGURE 2: The process industry integrated automation hardware-in-the-loop simulation system.

without accurate physical models, which is convenient for implementation.

In industrial production, due to the huge scale of the object, complex of production raw material, equipment maintenance difficult, and the risk of high temperature and high pressure of the production processes, it is difficult to carry out experimental study of industrial fault monitoring for theory researchers. So the hardware-in-the-loop simulation system is developed in this paper, using real hardware controller and industrial control network to build distributed control system (DCS). Develop computer simulation model as controlled object. In this paper, distillation column system (PCS 7 Unit Template) of Siemens as one of controlled objects has been studied, as shown in Figure 2.

Based on hardware-in-the-loop simulation system the valve faults, concentration faults, flow faults, and so on can be set. For the same fault, different fault parameters can be set; fault and disturbance in the distillation column also can be set according to the actual situation; for example, feed concentration can be reduced or increased 0.01; these faults are unable to simulate in the Tennessee Eastman process and other simulation processes. Satisfy our research of fault diagnosis, and there is no need to worry about the limitations of the factory.

Traditional fault diagnosis methods can not accurately and quickly detect the faults. With the development of the Internet and wide use of smart and wireless sensors and wireless communications and mobile devices, a large

amount of process data has been produced; how to use the process data to determine whether the industrial process is normal is a hot topic. In this paper data-driven methods (especially dynamic kernel PCA) are presented for applying to monitoring of distillation column. According to the data of the distillation column system, a framework about process data hierarchy can be seen in Figure 3.

The distillation column consists of six components, as shown in Figure 4: 1 represents distillation column, 2 represents feed, 3 represents vaporizer (“Vapor”), 4 represents condenser with reflux (“Reflux”), 5 represents extraction of the head product (“Light end”), and 6 represents extraction of the bottom product (“Heavy end”).

A flowchart of distillation column process is given in Figure 4. Distillation column is an indispensable device of chemical industrials and oil refining enterprises; once the system shows valve faults, concentration faults, and flow faults, the enterprises would suffer a great loss. Thus the fault detection and diagnosis of distillation tower become an important link in chemical production. This section studies the application of data-driven algorithms in the distillation column; the author selected the 14 monitoring variables, as shown in Table 1, and collected 960 groups of sample data under normal condition for off-line training model. Set faults as shown in Table 2; the faults mainly included three categories, respectively: the valve faults, concentration faults, and flow faults; the author collected 960 pieces of observation data from distillation column process for on-line testing. The

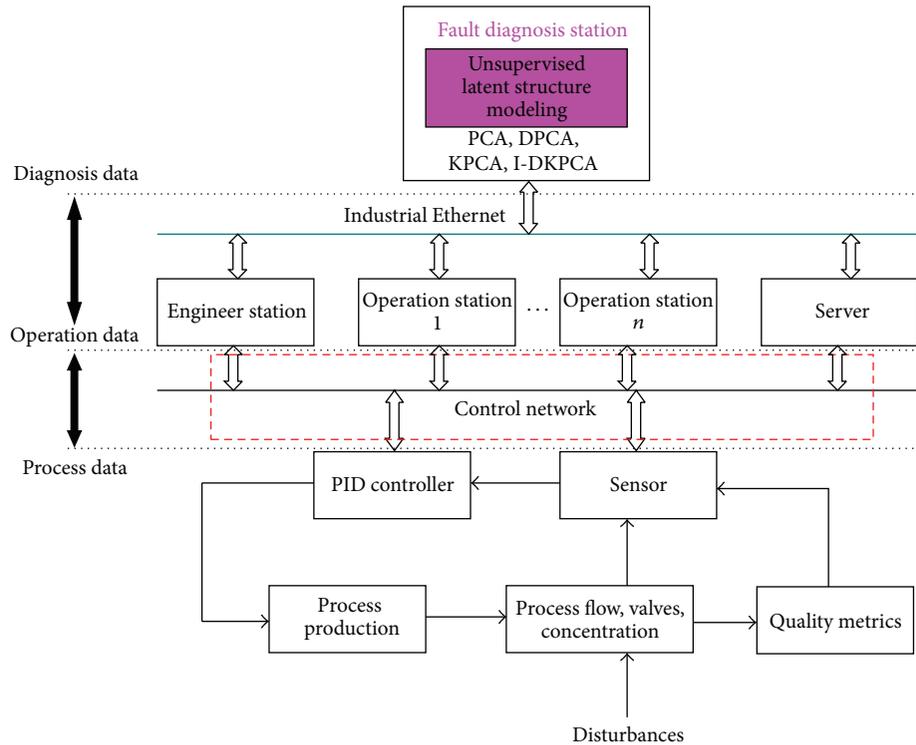


FIGURE 3: Distillation column process data hierarchy.

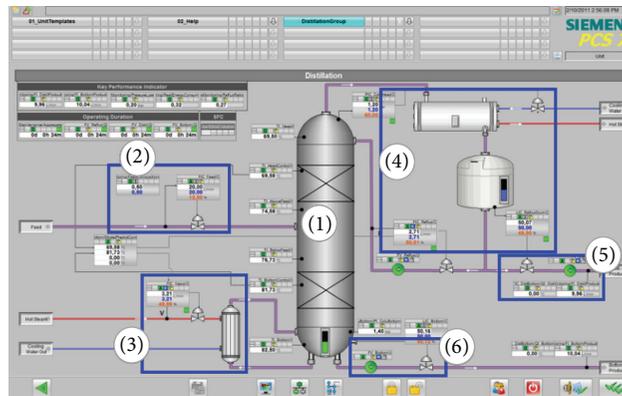


FIGURE 4: Distillation column system.

faults were introduced to the process at  $t = 5$  h and the faults' effect persisted until 16 hours.

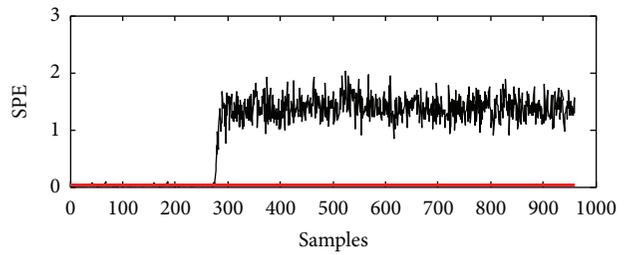
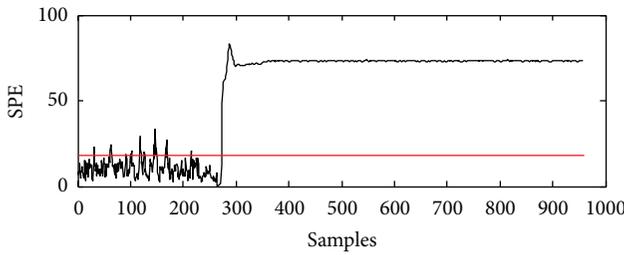
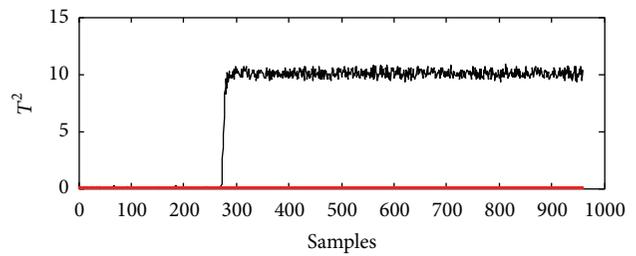
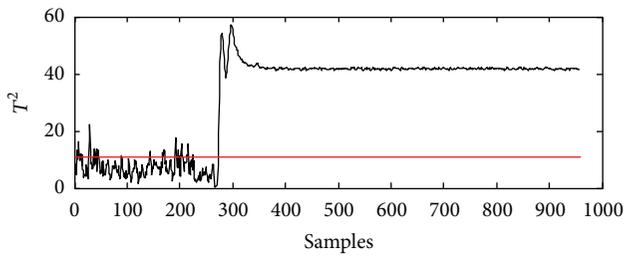
Faults 4, 5, and 6 are valve faults which are large change, so these faults could be detected by the data-driven methods and the alarm system of DCS. Because the valve faults were similar, the author has only given the results of fault 6. As shown in Figure 5,  $T^2$  index and SPE index of DPCA had poor results that the false rates were significantly high.  $T^2$  index and SPE index of KPCA perform better than DPCA. But, in this connection,  $T^2$  index and SPE index of I-DKPCA were better than KPCA and DPCA obviously.

Fault 1 is a big deviation of feed concentration. As shown in Figure 6, we can see that KPCA and I-DKPCA could detect fault more quickly than DPCA. The early detection can help the industries avoid greater loss. Besides, the false alarm rates of I-DKPCA were lower than DPCA and KPCA. In a word, compared with DPCA and KPCA, I-DKPCA shows the best performance.

Fault 2 was also a fault of feed concentration and fault 3 was a flow fault. Compared with fault 1, fault 2 was a small deviation; concentration value only changes 0.01, and fault 3 was also a small deviation; flow value changes 0.1. These

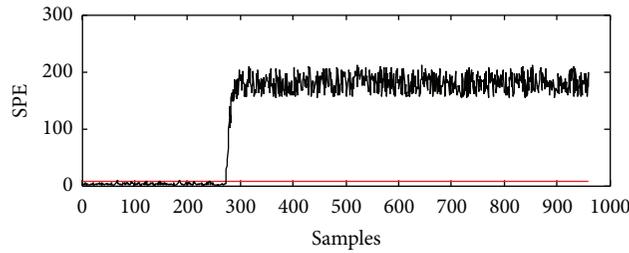
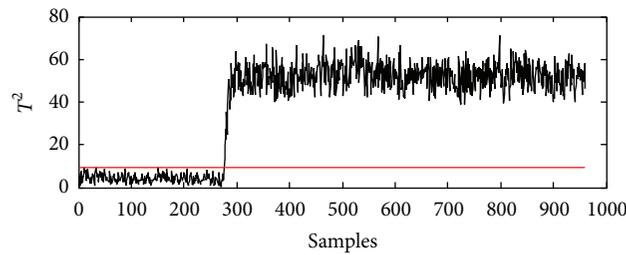
TABLE 1: Fourteen variables used for monitoring.

Number	Symbol	Variable
1	TI_Head	Temperature at the head of the distillation column
2	PI_Head	Pressure at the head of the distillation column
3	TI_Above	Temperature above the feed
4	TI_Below	Temperature below the feed
5	TI_Bottom	Temperature at the bottom of the distillation column
6	FI_Vapor	The steam flow
7	FI_Feed	Feed flow
8	FV_ColuHead	Cooling water valve
9	LI_Reflux	Level of reflux
10	FV_Vapor	Vapor valve
11	FV_Reflux	Reflux valve
12	FI_Reflux	Head product flow
13	FV_Bottom	The bottom of the distillation column valve
14	LI_Bottom	Level at the bottom of the distillation column



(a) DPCA

(b) KPCA



(c) I-DKPCA

FIGURE 5: Monitoring charts for fault 6. (a) DPCA, (b) KPCA, and (c) I-DKPCA.

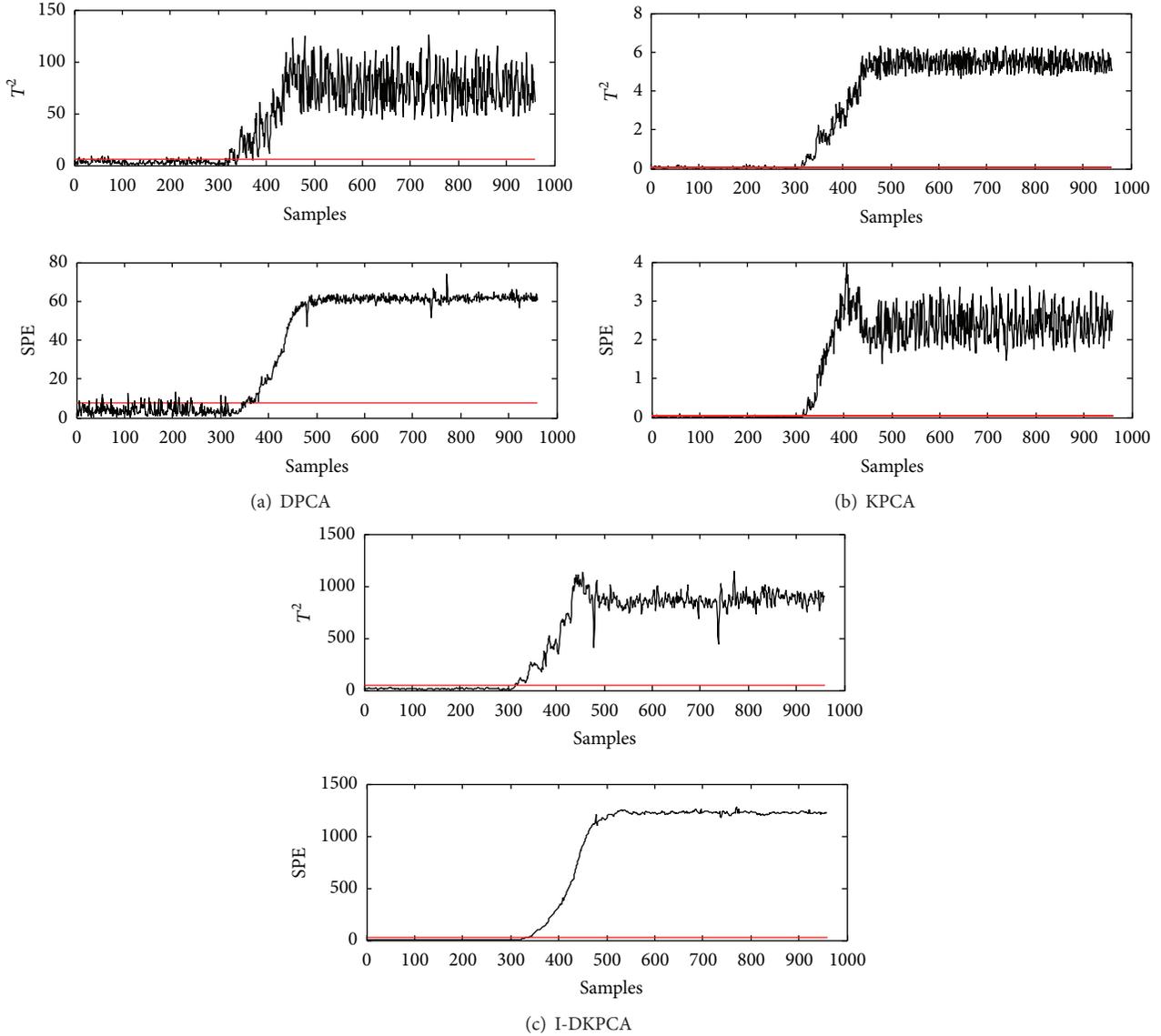


FIGURE 6: Monitoring charts for fault 1. (a) DPCA, (b) KPCA, and (c) I-DKPCA.

TABLE 2: Process faults for distillation column.

Number	Disturbances	Location
1	Increasing concentration Parameter: 0.5–0.6 at $t = 0.5$ h	Feed
2	Increasing concentration Parameter: 0.5–0.51 at $t = 0.5$ h	Feed
3	Increasing flow rate Parameter: 20–20.1 at $t = 0.5$ h	Feed
4	Increase valve opening Parameter: 12.5%–18% at $t = 0.5$ h	Feed
5	Decrease valve opening Parameter: 50%–45% at $t = 0.5$ h	Reflux
6	Decrease valve opening Parameter: 50%–45% at $t = 0.5$ h	Vapor

small deviations could not be detected by the alarm system of DCS. Some monitoring charts for fault 2 and 3 are shown

in Figures 7 and 8. SPE chart of DPCA could detect the faults, but  $T^2$  index of DPCA had big missed detection rates. Both  $T^2$  index and SPE index of KPCA give very high detection rates, but the false alarm rates were also high. Only the SPE index and  $T^2$  index of I-DKPCA perform the best results which had the lowest false alarm rates and highest detection rates.

In conclusion, the data-driven methods could apply to distillation column well, and the I-DKPCA based monitoring performs highest detection rate and smallest missed rates for all faults, in particular, when the small deviation happens.

#### 4. Conclusion

In this paper, process industry integrated automation hardware-in-the-loop simulation system has been developed,

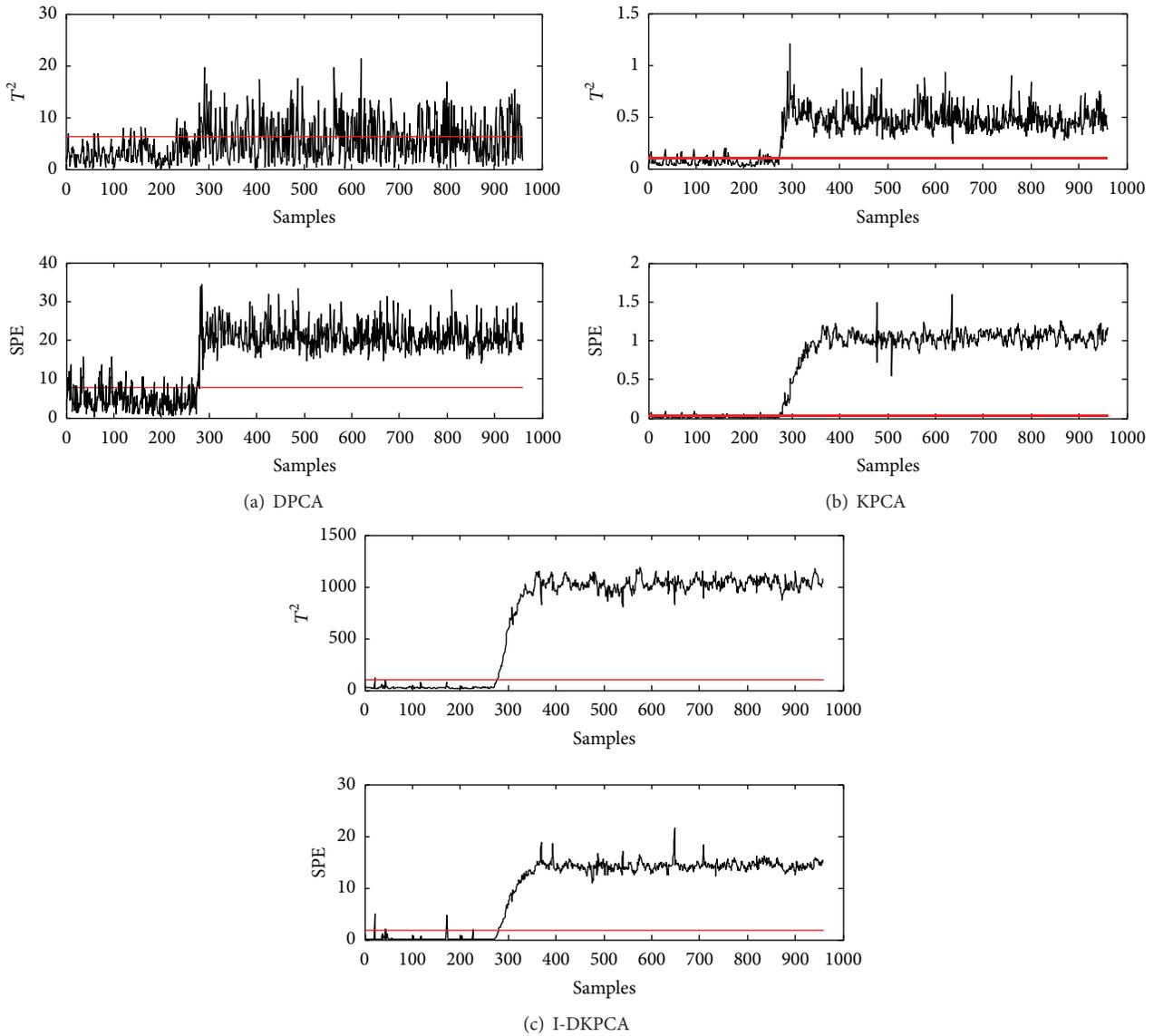


FIGURE 7: Monitoring charts for fault 2. (a) DPCA, (b) KPCA, and (c) I-DKPCA.

using real hardware controller and industrial control network to build distributed control system and distillation column system (PCS 7 Unit Template) of Siemens as one of controlled objects for fault diagnosis research, which made us get wanted data easily regardless of the loss. Realize the simulation of fault diagnosis from purely simulation object to the hardware-in-the-loop simulation system and promote the application of the data-driven technique in the actual industrial system.

A new indiscernibility dynamic kernel principal component analysis (I-DKPCA) method was proposed; the new method not only considered dynamic and nonlinear characteristics of industrial processes but also reduced the data dimension by indiscernibility and the cross-degree, got rid of irrelevant variables, simplified the calculation, and

improved the efficiency and accuracy of fault diagnosis. Through the applications of DPCA, KPCA, and I-DKPCA in the distillation column, the results showed that the proposed I-DKPCA performed better than DPCA and KPCA for all faults, especially for small faults.

### Conflict of Interests

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

### Acknowledgments

This work was supported by Young and Middle-Aged Innovation Talents Cultivation Plan of Higher Institutions in

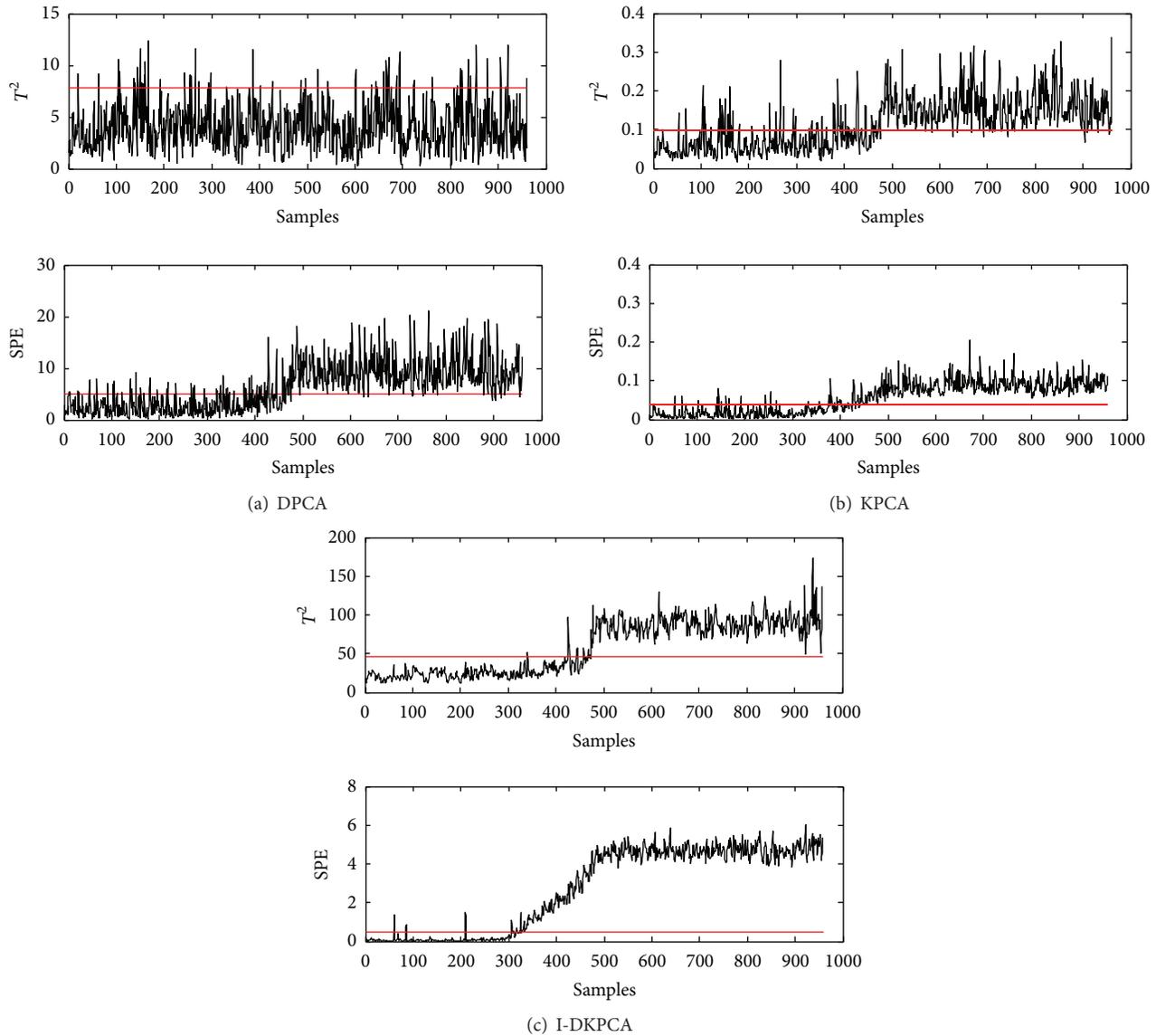


FIGURE 8: Monitoring charts for fault 3. (a) DPCA, (b) KPCA, and (c) I-DKPCA.

Tianjin (Grant no. 20130830) and National Natural Science Foundation of China (Grant no. 61403279).

## References

- [1] Q. P. He, S. J. Qin, and J. Wang, "A new fault diagnosis method using fault directions in Fisher discriminant analysis," *AIChE Journal*, vol. 51, no. 2, pp. 555–571, 2005.
- [2] J. Sun, C. Wang, J. Sun, and L. Wang, "Analog circuit soft fault diagnosis based on PCA and PSO-SVM," *Journal of Networks*, vol. 8, no. 12, pp. 2791–2796, 2013.
- [3] S. Xie and S. Krishnan, "Dynamic principal component analysis with nonoverlapping moving window and its applications to epileptic EEG classification," *The Scientific World Journal*, vol. 2014, Article ID 419308, 10 pages, 2014.
- [4] G. Stefatos and A. Ben Hamza, "Dynamic independent component analysis approach for fault detection and diagnosis," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8606–8617, 2010.
- [5] E. L. Russell, L. H. Chiang, and R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 81–93, 2000.
- [6] J. Fan and Y. Wang, "Fault detection and diagnosis of nonlinear non-Gaussian dynamic processes using kernel dynamic independent component analysis," *Information Sciences*, vol. 259, pp. 369–379, 2014.
- [7] Y. Yang, X. Li, X. Liu, and X. Chen, "Wavelet kernel entropy component analysis with application to industrial process monitoring," *Neurocomputing*, vol. 147, pp. 395–402, 2015.
- [8] K. P. Detroja, R. D. Gudi, and S. C. Patwardhan, "Plant-wide detection and diagnosis using correspondence analysis," *Control Engineering Practice*, vol. 15, no. 12, pp. 1468–1483, 2007.

- [9] N. Zhang, X. Tian, L. Cai, and X. Deng, "Process fault detection based on dynamic kernel slow feature analysis," *Computers & Electrical Engineering*, vol. 41, pp. 9–17, 2015.
- [10] L. Wang and H. Shi, "Multivariate statistical process monitoring using an improved independent component analysis," *Chemical Engineering Research and Design*, vol. 88, no. 4, pp. 403–414, 2010.
- [11] Z. Ge and Z. Song, "Process monitoring based on independent component analysis–principal component analysis (ICA–PCA) and similarity factors," *Industrial and Engineering Chemistry Research*, vol. 46, no. 7, pp. 2054–2063, 2007.
- [12] Y. Zhang and H. Zhang, "Fault detection for time-varying processes," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1527–1535, 2014.
- [13] Y. Zhang and S. Li, "Modeling and monitoring between-mode transition of multimodes processes," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2248–2255, 2013.
- [14] Y. Zhang, J. An, and C. Ma, "Fault detection of non-Gaussian processes based on model migration," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1517–1526, 2013.
- [15] L. H. Chiang, E. L. Russell, and R. D. Braatz, "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 2, pp. 243–252, 2000.
- [16] Y. Zhang, "Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM," *Chemical Engineering Science*, vol. 64, no. 5, pp. 801–811, 2009.
- [17] L. H. Chiang, R. J. Pell, and M. B. Seasholtz, "Exploring process data with the use of robust outlier detection algorithms," *Journal of Process Control*, vol. 13, no. 5, pp. 437–449, 2003.
- [18] J.-M. Lee, S. Joe Qin, and I.-B. Lee, "Fault detection of nonlinear processes using kernel independent component analysis," *The Canadian Journal of Chemical Engineering*, vol. 85, no. 4, pp. 526–536, 2007.
- [19] M. N. Nashalji, M. A. Shoorehdeli, and M. Teshnehlab, "Fault detection of the Tennessee Eastman process using improved PCA and neural classifier," *Advances in Intelligent and Soft Computing*, vol. 75, pp. 41–50, 2010.
- [20] K. A. Kosanovich and M. J. Piovoso, "PCA of wavelet transformed process data for monitoring," *Intelligent Data Analysis*, vol. 1, no. 2, pp. 85–99, 1997.
- [21] D. Dong and T. J. McAvoy, "Nonlinear principal component analysis—based on principal curves and neural networks," *Computers & Chemical Engineering*, vol. 20, no. 1, pp. 65–78, 1996.
- [22] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.
- [23] Q. Jiang and X. Yan, "Improved fault detection in nonlinear chemical processes using WKPCA-SVDD," *Korean Journal of Chemical Engineering*, vol. 31, no. 11, pp. 1935–1942, 2014.
- [24] S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 1, pp. 55–67, 2005.
- [25] Y. Zhang, C. Zhang, and W. Zhang, "Statistical analysis of nonlinear processes based on penalty factor," *Mathematical Problems in Engineering*, vol. 2014, Article ID 945948, 9 pages, 2014.
- [26] Y. Zhang and S. J. Qin, "Improved nonlinear fault detection technique and statistical analysis," *AIChE Journal*, vol. 54, no. 12, pp. 3207–3220, 2008.
- [27] C. F. Alcalá and S. J. Qin, "Reconstruction-based contribution for process monitoring with kernel principal component analysis," *Industrial & Engineering Chemistry Research*, vol. 49, no. 17, pp. 7849–7857, 2010.
- [28] Y. Zhang, J. An, and H. Zhang, "Monitoring of time-varying processes using kernel independent component analysis," *Chemical Engineering Science*, vol. 88, pp. 23–32, 2013.
- [29] Y. Zhang, N. Yang, and S. Li, "Fault isolation of nonlinear processes based on fault directions and features," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1567–1572, 2014.
- [30] P. Jiang, S. Qin, J. Wu, and B. Sun, "Time series analysis and forecasting for wind speeds using support vector regression coupled with artificial intelligent algorithms," *Mathematical Problems in Engineering*, vol. 2015, Article ID 939305, 14 pages, 2015.
- [31] S. W. Choi, J. Morris, and I.-B. Lee, "Dynamic model-based batch process monitoring," *Chemical Engineering Science*, vol. 63, no. 3, pp. 622–636, 2008.
- [32] S. W. Choi and I.-B. Lee, "Nonlinear dynamic process monitoring based on dynamic kernel PCA," *Chemical Engineering Science*, vol. 59, no. 24, pp. 5897–5908, 2004.
- [33] C. K. Yoo, S. W. Choi, and I.-B. Lee, "Dynamic monitoring method for multiscale fault detection and diagnosis in MSPC," *Industrial and Engineering Chemistry Research*, vol. 41, no. 17, pp. 4303–4317, 2002.
- [34] Y. Zhang and T. F. Edgar, "On-line batch process monitoring using modified dynamic batch PCA," in *Proceedings of the American Control Conference (ACC '07)*, pp. 2551–2556, New York, NY, USA, July 2007.
- [35] W. E. Larimore, *Statistical Methods in Control and Signal Processing*, Marcel Dekker, 1997.
- [36] J. Chen and C.-M. Liao, "Dynamic process fault monitoring based on neural network and PCA," *Journal of Process Control*, vol. 12, no. 2, pp. 277–289, 2002.