

Complexity

Advances in Modelling, Monitoring, and Control for Complex Industrial Systems

Lead Guest Editor: Zhiwei Gao

Guest Editors: Sing Kiong Nguang and De-xing Kong





**Advances in Modelling, Monitoring,
and Control for Complex Industrial Systems**

Complexity

**Advances in Modelling, Monitoring,
and Control for Complex Industrial Systems**

Lead Guest Editor: Zhiwei Gao

Guest Editors: Sing Kiong Nguang and De-xing Kong



Copyright © 2019 Hindawi. All rights reserved.




This is a special issue published in “Complexity.” 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



- José A. Acosta, Spain
Carlos F. Aguilar-Ibáñez, Mexico
Mojtaba Ahmadiéh Khanesar, UK
Tarek Ahmed-Ali, France
Alex Alexandridis, Greece
Basil M. Al-Hadithi, Spain
Juan A. Almendral, Spain
Diego R. Amancio, Brazil
David Arroyo, Spain
Mohamed Boutayeb, France
Átila Bueno, Brazil
Arturo Buscarino, Italy
Guido Caldarelli, Italy
Eric Campos-Canton, Mexico
Mohammed Chadli, France
Émile J. L. Chappin, Netherlands
Diyi Chen, China
Yu-Wang Chen, UK
Giulio Cimini, Italy
Danilo Comminiello, Italy
Sara Dadras, USA
Sergey Dashkovskiy, Germany
Manlio De Domenico, Italy
Pietro De Lellis, Italy
Albert Diaz-Guilera, Spain
Thach Ngoc Dinh, France
Jordi Duch, Spain
Marcio Eisenkraft, Brazil
Joshua Epstein, USA
Mondher Farza, France
Thierry Floquet, France
Mattia Frasca, Italy
José Manuel Galán, Spain
Lucia Valentina Gambuzza, Italy
Bernhard C. Geiger, Austria
Carlos Gershenson, Mexico
Peter Giesl, UK
Sergio Gómez, Spain
Lingzhong Guo, UK
Xianggui Guo, China
Sigurdur F. Hafstein, Iceland
Chittaranjan Hens, India
Giacomo Innocenti, Italy
Sarangapani Jagannathan, USA
Mahdi Jalili, Australia
Jeffrey H. Johnson, UK
M. Hassan Khooban, Denmark
Abbas Khosravi, Australia
Toshikazu Kuniya, Japan
Vincent Labatut, France
Lucas Lacasa, UK
Guang Li, UK
Qingdu Li, Germany
Chongyang Liu, China
Xiaoping Liu, Canada
Xinzhi Liu, Canada
Rosa M. Lopez Gutierrez, Mexico
Vittorio Loreto, Italy
Noureddine Manamanni, France
Didier Maquin, France
Eulalia Martínez, Spain
Marcelo Messias, Brazil
Ana Meštrović, Croatia
Ludovico Minati, Japan
Ch. P. Monterola, Philippines
Marcin Mrugalski, Poland
Roberto Natella, Italy
Sing Kiong Nguang, New Zealand
Nam-Phong Nguyen, USA
B. M. Ombuki-Berman, Canada
Irene Otero-Muras, Spain
Yongping Pan, Singapore
Daniela Paolotti, Italy
Cornelio Posadas-Castillo, Mexico
Mahardhika Pratama, Singapore
Luis M. Rocha, USA
Miguel Romance, Spain
Avimanyu Sahoo, USA
Matilde Santos, Spain
Josep Sardanyés Cayuela, Spain
Ramaswamy Savitha, Singapore
Hiroki Sayama, USA
Michele Scarpiniti, Italy
Enzo Pasquale Scilingo, Italy
Dan Selișteanu, Romania
Dehua Shen, China
Dimitrios Stamovlasis, Greece
Samuel Stanton, USA
Roberto Tonelli, Italy
Shahadat Uddin, Australia
Gaetano Valenza, Italy
Dimitri Volchenkov, USA
Christos Volos, Greece
Zidong Wang, UK
Yan-Ling Wei, Singapore
Honglei Xu, Australia
Yong Xu, China
Xinggang Yan, UK
Baris Yuçe, UK
Massimiliano Zanin, Spain
Hassan Zargarzadeh, USA
Rongqing Zhang, USA
Xianming Zhang, Australia
Xiaopeng Zhao, USA
Quanmin Zhu, UK

Contents

Advances in Modelling, Monitoring, and Control for Complex Industrial Systems

Zhiwei Gao , Sing Kiong Nguang , and De-Xing Kong 
Editorial (3 pages), Article ID 2975083, Volume 2019 (2019)


A New Mathematical Modeling Method for Four-Stage Helicopter Main Gearbox and Dynamic Response Optimization

Yuan Chen , Rupeng Zhu , Guanghu Jin, Yeping Xiong, Jie Gao, and Meijun Liao
Research Article (13 pages), Article ID 5274712, Volume 2019 (2019)

Design of a Logistics Nonlinear System for a Complex, Multiechelon, Supply Chain Network with Uncertain Demands

Aaron Guerrero Campanur, Elias Olivares-Benitez , Pablo A. Miranda , Rodolfo Eleazar Perez-Loaiza, and Jose Humberto Ablanedo-Rosas 
Research Article (16 pages), Article ID 4139601, Volume 2018 (2019)

Fractional-Derivative Approximation of Relaxation in Complex Systems

Kin M. Li, Mihir Sen, and Arturo Pacheco-Vega 
Research Article (12 pages), Article ID 8318519, Volume 2018 (2019)

Maximizing the Coverage of Roadmap Graph for Optimal Motion Planning

Jae-Han Park  and Tae-Woong Yoon 
Research Article (23 pages), Article ID 9104720, Volume 2018 (2019)

Adaptive Fuzzy Control for Stochastic Pure-Feedback Nonlinear Systems with Unknown Hysteresis and External Disturbance

Xikui Liu, Yingying Ge, and Yan Li 
Research Article (11 pages), Article ID 1487134, Volume 2018 (2019)

Interface Data Modeling to Detect and Diagnose Intersystem Faults for Designing and Integrating System of Systems

Kyung-Min Seo  and Kwang-Phil Park
Research Article (21 pages), Article ID 7081501, Volume 2018 (2019)




Networked Control System Design for Turbofan Aeroengines with Aging and Deterioration

Ruichao Li , Sing Kiong Nguang , Yingqing Guo , and Yifeng Chen 
Research Article (13 pages), Article ID 6010216, Volume 2018 (2019)

Delay-Dependent Stability in Uncalibrated Image-Based Dynamic Visual Servoing Robotic System

Tao Li, Hui Zhao , and Yu Chang
Research Article (14 pages), Article ID 1360874, Volume 2018 (2019)

Distributed Control of a Manufacturing System with One-Dimensional Cellular Automata

Irving Barragan-Vite , Juan C. Seck-Tuoh-Mora, Norberto Hernandez-Romero , Joselito Medina-Marin , and Eva S. Hernandez-Gress
Research Article (15 pages), Article ID 7235105, Volume 2018 (2019)

Couple-Group Consensus: A Class of Delayed Heterogeneous Multiagent Systems in Competitive Networks

Lianghao Ji , Yue Zhang, and Yiliu Jiang


Research Article (11 pages), Article ID 7386729, Volume 2018 (2019)

Optimal Allocation Method of Discrete Manufacturing Resources for Demand Coordination between Suppliers and Customers in a Fuzzy Environment

Wei Xu  and Yinyun Yu 

Research Article (18 pages), Article ID 1410957, Volume 2018 (2019)

Towards Reduced-Order Models of Solid Oxide Fuel Cell

Maciej Ławryńczuk 

Research Article (18 pages), Article ID 6021249, Volume 2018 (2019)

Estimation and Synthesis of Reachable Set for Singular Markovian Jump Systems

Yucai Ding  and Hui Liu

Research Article (10 pages), Article ID 5139840, Volume 2018 (2019)

Editorial

Advances in Modelling, Monitoring, and Control for Complex Industrial Systems

Zhiwei Gao ¹, Sing Kiong Nguang ², and De-Xing Kong ³

¹Faculty of Engineering and Environment, University of Northumbria, Newcastle upon Tyne, NE1 8ST, UK

²Department of Electrical and Computer Engineering, University of Auckland, Auckland 1010, New Zealand

³Department of Mathematics, Zhejiang University, Hangzhou, China

Correspondence should be addressed to Zhiwei Gao; zhiwei.gao@northumbria.ac.uk

Received 23 December 2018; Accepted 24 December 2018; Published 27 February 2019

Copyright © 2019 Zhiwei 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.

1. Introduction

Nowadays, industrial systems, such as chemical process, manufacturing process, power network, sustainable energy systems, transportation systems, wireless communication network, robotic systems, and biomedical systems, are becoming more complex, which are generally composed of a couple of interconnected systems, possess high-nonlinear and stochastic dynamics, are equipped with multiple control loops, and operate under noisy environments and varying loads. On the other hand, industrial systems are becoming more expensive, which have higher requirement for operation performance, productiveness, availability, reliability, and safety. Moreover, industry 4.0 has become the current trend of automation industries, which has great impacts on improving the reliability and operation performance of complex industrial systems. Therefore, it is paramount but challenging to develop effective techniques in modelling, monitoring, and control for complex industrial systems. As rapid developments of industrial automation, cloud and cognitive computation, internet of things, artificial intelligence, cyber-physical systems, and sensor technologies, modelling, monitoring, and control for complex industrial systems have been greatly stimulated and new techniques and applications have emerged in recent years.

This special issue aims to provide a platform for researchers and engineers to report their recent results, exchange research ideas, and overlook emerging research and application directions in modelling, monitoring, and advanced control for complex industrial systems. After a

strict peer-review process, thirteen papers are selected out of forty submissions. The selected papers can be categorized into modelling and optimization for complex industrial systems, monitoring and diagnosis for complex industrial systems, and advanced control for complex industrial systems, respectively. The papers selected in this special issue will be reviewed following the categories mentioned above.

2. Modelling and Optimization for Complex Industrial Systems

Solid oxide fuel cells (SOFCs) are electrochemical devices converting chemical energy stored in hydrocarbon fuels into electrical energy, which are alternatives to conventional power generation schemes. It is interesting to describe dynamic behavior of a solid oxide fuel cell, which is multi-input and multioutput process in essence. In the paper entitled “Towards Reduced-Order Models of Solid Oxide Fuel Cell” authored by M. Lawrynczuk, modelling issue of a solid oxide fuel cell is investigated. By using the first-principle theory, full-order model is established firstly, and reduced-models are then derived by using state-truncation method and residualisation model-reduction method, respectively. The effectiveness of the established models is demonstrated by using simulation studies.

Recently, fractional-order differentiation equations have received attention in modelling of complex systems. The paper entitled “Fractional-Derivative Approximation of Relaxation in Complex Systems,” authored by K. M. Li *et al.*, presents a system identification procedure to

build a fractional-order differential equation model to accurately describe time-dependent behavior of a class of complex systems. Caputo definition of the fractional derivative, Mittag-Leffler function, and simulated annealing optimization algorithm are used to identify the model parameters. The proposed system identification modelling algorithm is validated by using experimental data of a shell-and-tube heat exchanger and a multiroom building test bed.

Four-stage helicopter main gearbox has a complex structure, and its dynamic response has a significant influence on the fatigue life of the system. As a result, it is significant to carry out modelling and dynamic analysis of the four-stage helicopter gearbox. In the paper entitled “A New Mathematical Modeling Method for Four-Stage Helicopter Main Gearbox and Dynamic Response Optimisation” authored by Y. Chen *et al.*, finite element method and lumped mass method are synthesized to model the overall multinode dynamic model of a four-stage helicopter main gearbox.

Logistics and supply chain networks consist in a large amount of interconnected actors with significant nonlinear and stochastic features; therefore, it is challenging to model and analyse this kind of complex industrial systems. In the paper entitled “Design of a Logistics Nonlinear System for a Complex, Multiechelon, Supply Chain Network with Uncertain Demands” authored by A. G. Campanur *et al.*, the design problem of a four-echelon supply chain is investigated, which is represented as an inventory-location model with uncertain demand and continuous review inventory policy. A mixed integer nonlinear programming model is used to seek the optimal design of the distribution network, and a linearized integer programming model is also used to save computational cost.

In order to respond to unpredictable market changes, manufacturing industries are required to reconstruct rapidly. Discrete manufacturing has thus become the mainstream model in manufacturing industries owing to its capabilities of handling noncontinuity and configurability. In the paper entitled “Optimal Allocation Method of Discrete Manufacturing Resources for Demand Coordination between Suppliers and Customers in a Fuzzy Environment” authored by W. Xu and Y. Yu, a resource optimization configuration approach is presented on the basis of the customers’ expectation bias and the suppliers’ profit maximization under fuzzy environment. A bi-level planning model for an optimal configuration of manufacturing resources is established, and a fast nondominated sorting genetic algorithm is used to solve the model. The validity and feasibility of the model are illustrated by an example and simulations.

Automatic motion planning for industrial robots plays a crucial role in the next-generation manufacturing systems such as industrial 4.0. The paper entitled “Maximizing the Coverage of Roadmap Graph for Optimal Motion Planning” authored by J.-H. Park and T.-W. Yoon addresses an optimal algorithm to optimize a roadmap graph which can cover arbitrary morphologies of the free configuration space with maximal coverage. The proposed approach is applied to an industrial robot model, and the obtained roadmap graph is demonstrated to be effective in path-finding tests via simulation studies.

3. Monitoring and Diagnosis for Complex Industrial Systems

Nowadays, industrial systems are becoming more complex and expensive; thus they have higher demand and requirement on system reliability and safety. As a result, monitoring and diagnosis for complex industrial systems have been a popular research topic during the last decades. Industry 4.0 has further stimulated this research and application field.

In system of systems engineering, system integrators play key roles in forming compatible and reliable interfaces between subsystems. In the paper entitled “Interface Data Modeling to Detect and Diagnose Intersystem Faults for Designing and Integrating System of Systems” authored by K.-M. Seo and K.-P. Park, a systematic solution is proposed to identify and diagnose interface faults during the phase of the system design and integration. The presented approach is applied to an underwater shipbuilding system with 10 interfaces, and 14 fault scenarios are identified and diagnosed, leading to a reliable submarine construction.

4. Advanced Control for Complex Industrial Systems

Singular systems are also called DAE (differential-algebraic equation) systems, descriptor systems, generalized dynamic systems, and implicit systems, which arise in convenient and natural modelling process. Many engineering systems such as electrical engineering systems, mechanical systems, chemical systems, economic systems, and environmental systems can be modeled as singular systems. Stochastic behavior in some engineering systems such as Markovian jumping would further add the complexity of system dynamics. In the paper entitled “Estimation and Synthesis of Reachable Set for Singular Markovian Jump Systems” authored by Y. Ding and H. Lui, reachable set estimation and state-feedback control are investigated. Sufficient conditions on state/output reachable sets are developed, and state-feedback controllers are designed correspondingly.

Hysteresis, widely existing in mechanical systems, would degrade system performance and even cause system instability. Moreover, stochastic uncertainties usually occur in engineering systems, which have brought more challenges in control synthesis. In the paper entitled “Adaptive Fuzzy Control for Stochastic Pure-Feedback Nonlinear Systems with Unknown Hysteresis and External Disturbance” authored by X. Liu *et al.*, the tracking control problem for a class of stochastic nonlinear systems subjected to unknown hysteresis is investigated. The mean-value theorem, Nussbaum function, Lyapunov function, and back-stepping techniques are used in the analysis and design. An adaptive fuzzy controller is designed to ensure that the closed-loop system is semi-globally uniformly bounded and the tracking errors converge to a small neighborhood of the origin. The proposed control algorithms are demonstrated by simulation studies.

Vision-based control has been a mainstreaming technique in industrial robotic systems; however, visual feedback delays and uncalibrated visual parameters would limit control system performance, or even damage the system stability. The

paper entitled “Delay-Dependent Stability in Uncalibrated Image-Based Dynamic Visual Servoing Robotic System” authored by T. Li *et al.* addresses an adaptive control technique to stabilize a robotic system subjected to visual delays and uncalibrated visual parameters. The delay-dependent stability conditions are also provided to reveal the relationship between the system stability and the delayed time.

Consensus is a fundamental issue of the coordinated control for multiagent systems, which has a wide application in mobile robot systems, distributed target tracking, group decision making, and so forth. Group consensus is an extension of the conventional consensus issue, which has been paid attention recently. In the paper entitled “Couple-Group Consensus: A Class of Delayed Heterogeneous Multiagent Systems in Competitive Networks” authored by L. Ji *et al.*, couple-group consensus is dealt with for a class of heterogeneous multiagent systems subjected to both input and communication delays. A novel distributed coordination control protocol is addressed to ensure the heterogeneous system to achieve couple-group consensus asymptotically.

Distributed engine control would be the next revolution in turbine engine control systems. Networked engine control is an intermediate distributed engine control approach, which has been paid many attentions. It is evident that transmission delays and packet dropout are inevitable during data transmission. Moreover, system performance may degrade due to component aging. As a result, how to make the system robust against transmission delays, packet dropout, and component aging is an important issue in network control systems. In the paper entitled “Networked Control System Design for Turbofan Aeroengines with Aging and Deterioration” authored by R. Li *et al.*, a network control system design approach is proposed for a mixed two-spool turbofan aeroengine with aging and deterioration so that the system can achieve robustness against dropouts and delays as well as satisfactory dynamic performance against aging and deterioration.

With an increasing demand of top quality and variety of products, modern manufacturing industries are stimulated to automate their production processes. However, modelling and control for automated manufacturing systems are challenging due to rapid changes in global markets, unexpected events, and discrete characteristics of both cellular automata and manufacturing systems. In the paper entitled “Distributed Control of a Manufacturing System with One-Dimensional Cellular Automata” authored by I. Barragan-Vite *et al.*, a distributed control modelling method is proposed for an automated manufacturing process. A procedure is provided to achieve local evolution rule of the automation on the basis of the relationships among the resources and material flow in manufacturing process. The effectiveness of the proposed distributed control strategy is illustrated and demonstrated by case studies.

5. Conclusion

An overview of the 13 selected papers for this special issue has been presented, reflecting the most recent progress in the research field. We hope this special issue can further stimulate interest in advanced modelling, monitoring, control,

and optimization for complex industrial systems from both academic societies and automation industries. More research results and practical applications on complex industrial systems are expected in the future.

Conflicts of Interest

The guest editor team does not have any conflicts of interest or private agreements with companies.

Acknowledgments

The guest editors would like to express great appreciation to all authors who supported this special issue by contributing papers. The guest editor team is deeply indebted to all reviewers for their insightful and constructive comments that helped shape this special issue. Finally, the guest editors would like to thank the research supports from the Alexander von Humboldt Foundation, the NNSFC grant (61673074), and their institutes.

Zhiwei Gao
Sing Kiong Nguang
De-Xing Kong

Research Article

A New Mathematical Modeling Method for Four-Stage Helicopter Main Gearbox and Dynamic Response Optimization

Yuan Chen ^{1,2}, Rupeng Zhu ¹, Guanghu Jin,¹ Yeping Xiong,² Jie Gao,³ and Meijun Liao³

¹College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

²Engineering and the Environment, University of Southampton, Boldrewood Innovation Campus, SO16 7QF Southampton, UK

³AECCE Hunan Aviation Powerplant Research Institute, Zhuzhou 412002, China

Correspondence should be addressed to Rupeng Zhu; rpzhu@nuaa.edu.cn

Received 28 June 2018; Accepted 4 September 2018; Published 26 February 2019

Academic Editor: Sing Kiong Nguang

Copyright © 2019 Yuan 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.

A new mathematical modeling method, namely, the finite element method and the lumped mass method (LMM-FEM) mixed modeling, is applied to establish the overall multinode dynamic model of a four-stage helicopter main gearbox. The design of structural parameters of the shaft is the critical link in the four-stage gearbox; it affects the response of multiple input and output branches; however, only the meshing pairs were frequently shown in the dynamic model in previous research. Therefore, each shaft is also treated as a single node and the shaft parameters are coupled into the dynamic equations in this method, which is more accurate for the transmission chain. The differential equations of the system are solved by the Fourier series method, and the dynamic response of each meshing element is calculated. The sensitivity analysis method and parameter optimization method are applied to obtain the key shaft parameters corresponding to each meshing element. The results show that the magnitude of dynamic response in converging meshing pair and tail output pair is higher than that of other meshing pairs, and the wall thickness has great sensitivity to a rotor shaft. In addition, the sensitivity analysis method can be used to select the corresponding shaft node efficiently and choose parameters appropriately for reducing the system response.

1. Introduction

The four-stage helicopter main gearbox has three input branches; they converge on the central gear and go one way to the rotor and the other way to the tail chain. The transmission chain includes many parts, which make its structure very complicated. The meshing pairs include internal excitations such as time-varying meshing stiffness and transmission error; in addition, external excitations like torques of the engine, rotor, and tail load also have an important effect on the system. The dynamic response has a profound influence on the fatigue life of the system; therefore, it is necessary to carry out the dynamic analysis for the four-stage helicopter gearbox.

Regarding gearbox modeling, Kubur et al. [1] established the dynamic model of a multishaft helical gear reduction system formed by flexible shafts and studied the influence of some of the key system parameters under forced vibrations. Raclot and Vexé [2] simulated the contributions of shape deviations and mounting errors to the dynamic behaviour

of multistage geared systems. Choy et al. [3] presented a multistage multimesh gear transmission system; the individual modal component responses and the overall system dynamics of the gearbox were predicted. Dzitkowski and Dymarek [4] obtained the multistage gear mechanical characteristics by properly selecting the dynamical properties of the system based on using the active synthesis method. Chen et al. [5] established the vibration model of a four-stage main transmission system in a helicopter through the lumped mass method, and the influence of the torsional stiffness of shafts on the first five orders of the system's natural frequency was studied. However, in this dynamic model, the shaft was not regarded as a node and all the differential equations are related to the meshing pairs.

In the area of gear response analysis, Parker et al. [6] analyzed the dynamic response of a helicopter planetary gear system under different ranges of operating speeds and torques based on a finite element method and focused on the gear contact conditions. Vexé and Flamand [7] calculated

dynamic tooth loads and response on a planetary gear set by the Ritz method and compared the results with those given by direct integrations for highly reduced computation times. Chaari et al. [8] compared the dynamic response of healthy planetary gears with cracked planetary gears in both the time and frequency domains applied with the Wigner-Ville distribution method. Walha et al. [9] investigated dynamics of a two-stage gear system involving backlash and time-dependent mesh stiffness, and the decomposition of a non-linear system into some linear systems was solved by the Newmark iterative algorithm. Zhou et al. [10] developed a coupled lateral-torsional nonlinear dynamic model with 16 degrees of freedom (16-DOF) of gear-rotor-bearing transmission system considering the nonlinear features, and the mean load excitation had a complicated influence on the coupled system; they concluded that the torsional vibration was the dominant response in the geared system. Chen et al. [11] detected the key shafts of the four-stage helicopter gearbox and analyzed their sensitivity to each branch of the system through the lumped mass method.

Gearbox parameter optimization plays an important role in the helicopter design. Chen and Shao [12] demonstrated the effectiveness of the proposed mesh stiffness model under the influences of the tooth profile modification, not only applied in low contact ratio, but also in high contact ratio. Yang et al. [13] presented the gearbox parameter optimization method by the artificial bee colony algorithm when diagnosing the gear faults and verified the theory through a two-stage parallel shaft gearbox. Bozca [14–16] studied gearbox geometric parameter optimization to reduce rattle noise in an automotive transmission based on a torsional vibration model, and the module, number of teeth, axial clearance, and backlash could be improved through this method. However, system dynamic optimizations were mostly through gear parameters or profile modification; the shaft parameters were not regarded as variables in these researches.

About the sensitivity analysis to system parameters, Lin and Parker [17] and Guo and Parker [18] investigated the natural frequency and vibration mode sensitivities to system parameters. Chen et al. [19] studied the response sensitivity to system parameters like gear mesh stiffness, damping, diameter ratio, and gear mass unbalance in a coupled gear system. However, sensitivity analysis based on multinode dynamic modeling has not been captured by these researches.

In summary, the research of most scholars focuses on the planetary gear chain of the helicopter main gearbox. The dynamic equations also seldom reflect the characteristics of the shaft parameters as independent nodes; thus, it is meaningful to improve the dynamic modeling and propose the optimization method.

2. LMM-FEM Mixed Modeling

The dynamic modeling process of a typical four-stage helicopter main gearbox is shown in Figure 1 [20]. Figure 1(a) is the system dynamic model, there are seven meshing elements (A, B, C, D, E, and F) in the system, and the elements F and G are internal and external meshing pairs in the planetary system, which contains one sun gear, six planet gears, and one carrier. θ is the rotational degree of freedom (DOF) of each node. Figure 1(b) is a comparison of the LMM modeling method and the LMM-FEM mixed modeling method. In this new method, the gearbox is regarded as finite critical nodes, which include a shaft and a meshing pair; each DOF corresponds to each node of the system. The system's generalized DOF coordinate vector X is

$$X = \left\{ \theta_1^{(j)}, \theta_2^{(j)}, \theta_3^{(j)}, \theta_4^{(j)}, \theta_5^{(j)}, \theta_6^{(j)}, \theta_7^{(j)}, \theta_8^{(j)}, \theta_9, \theta_{10}, \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}, \theta_{15}, \theta_{16}, \theta_{17}, \theta_{18}, \theta_{19}, \theta_{20}, \theta_{21}, \theta_{22}, \theta_{23}, \theta_{24} \right\}^T. \quad (1)$$

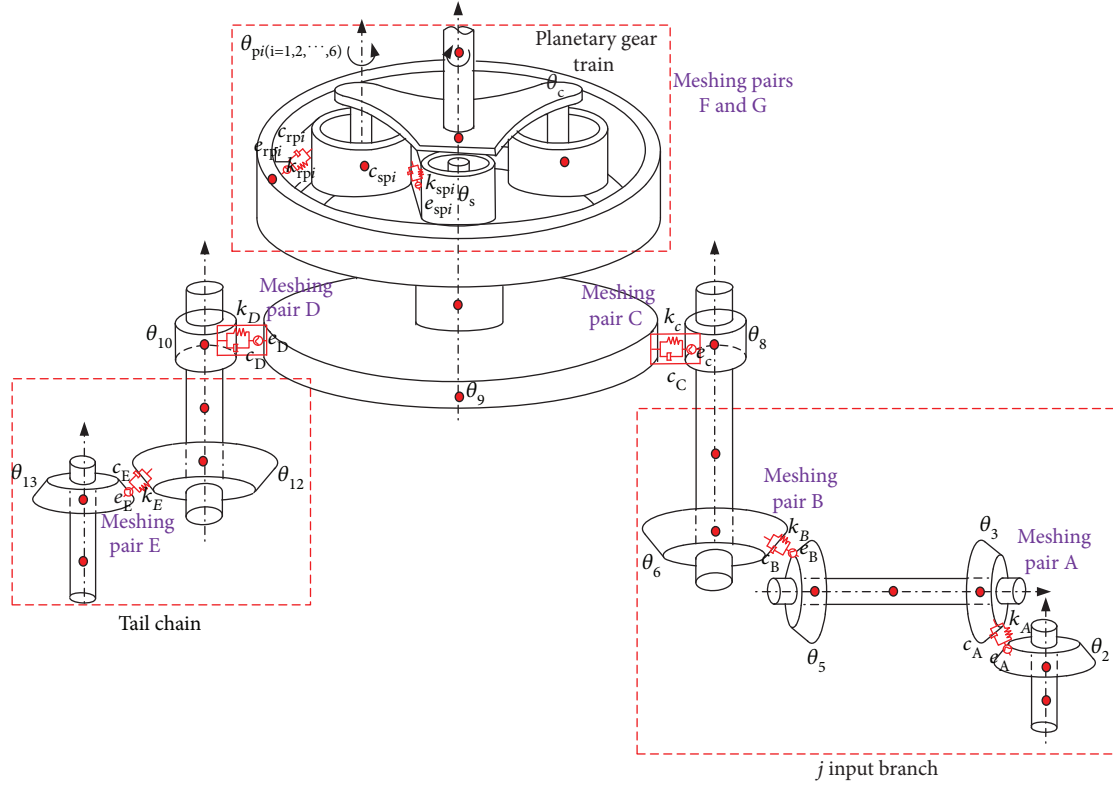
2.1. Dynamic Modeling of the Meshing Pairs. For the seven meshing pairs (A, B, C, D, E, F, and G) highlighted in Figure 1, the time-varying meshing stiffness can be expressed in the Fourier series with meshing frequency ω [21]:

$$\begin{cases} k(t) = k_m + k_a \sin(\omega t + \beta), \\ k_{spi}(t) = k_{m,spi} + k_{a,spi} \sin(\omega t + \beta_{spi}), \\ k_{rpi}(t) = k_{m,rpi} + k_{a,rpi} \sin(\omega t + \beta_{rpi}), \end{cases} \quad (2)$$

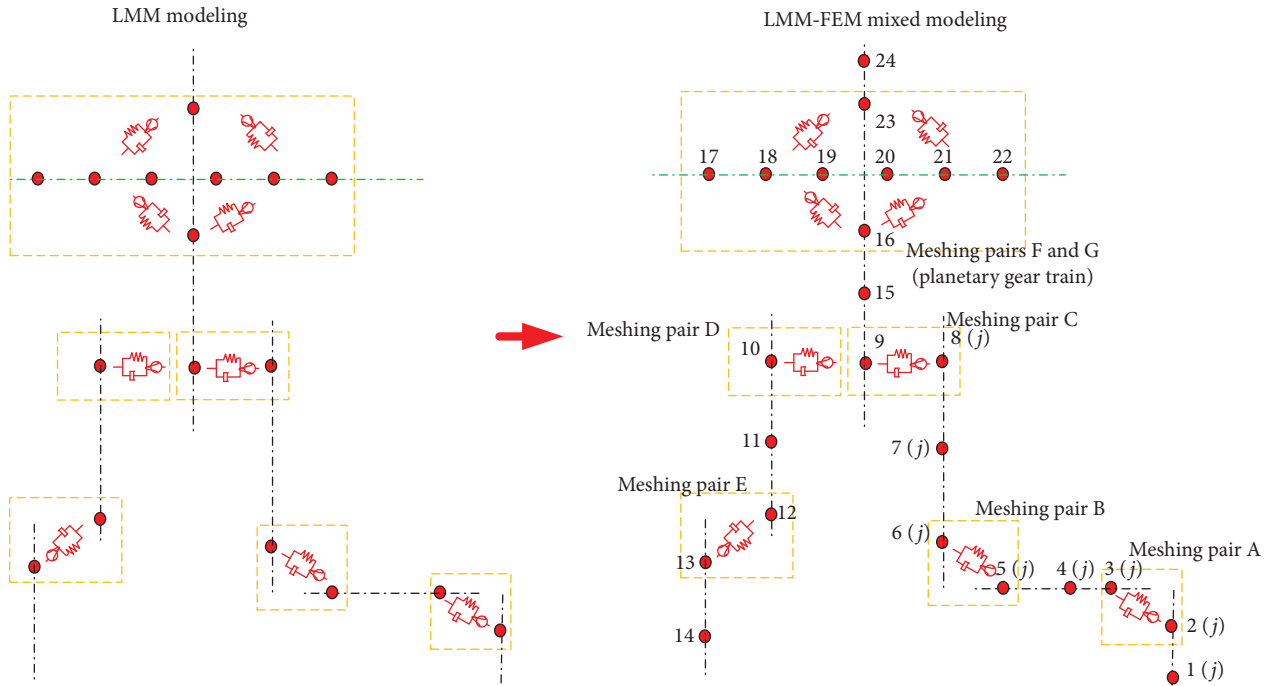
where $k(t)$, $k_{spi}(t)$, and $k_{rpi}(t)$ are time-varying meshing stiffness of each gear pair; k_m and k_a are the average and maximum variable meshing stiffness; β is the initial phase of meshing stiffness; and ω is the fundamental meshing frequency.

The transmission errors are shown in the same way:

$$\begin{cases} e(t) = e_m + e_a \sin(\omega t + \varphi), \\ e_{spi}(t) = A_{spi} \sin(\omega t + \varphi_{spi}) + E_{pi} \sin(\omega_{pH} t + \varphi_{pi} + \alpha) + E_s \sin\left[\omega_{sH} t + \varphi_s - \frac{2\pi(i-1)}{N} + \alpha\right], \\ e_{rpi}(t) = A_{rpi} \sin(\omega t + \varphi_{rpi}) + E_{pi} \sin(\omega_{pH} t + \varphi_{pi} - \alpha) + E_r \sin\left[\omega_{rH} t + \varphi_r - \frac{2\pi(i-1)}{N} - \alpha\right], \end{cases} \quad (3)$$



(a) Dynamic model



(b) Modeling comparison

FIGURE 1: Dynamic modeling for the multinode system.

where $e(t)$, $e_{spi}(t)$, and $e_{rpi}(t)$ are time-varying transmission errors of each gear pair; e_m and e_a are static and dynamic transmission error amplitude; φ is the initial phase of

transmission error; ω_{pH} , ω_{sH} , and ω_{rH} are rotational frequency of planet gear, sun gear, and carrier; α is the pressure angle; and N is the number of planet gear.

Tooth deflection along the meshing line is defined as the dynamic response and presented as follows:

$$X(t) = \theta_d r_d - \theta_p r_p - e(t), \quad (4)$$

where θ_d and θ_p are rotational DOF of drive gear and driven gear and r_d and r_p are the radii of the base circle in the drive gear and driven gear.

The dynamic forces of each gear pair $F(t)$ are defined as follows:

$$\begin{cases} F(t) = F^p(t) + F^d(t), \\ F^p(t) = k(t)X(t), \\ F^d(t) = c(t)\dot{X}(t), \end{cases} \quad (5)$$

where $c(t)$ is the meshing damping and $\dot{X}(t)$ is the relative velocity along the meshing line.

2.2. Differential Equation of Multinode System. According to the dynamic modeling above, the differential equation of the multinode dynamic model can be deduced through Newton's law:

Node 1: shaft node

$$J_1 \ddot{\theta}_1^{(j)} + G_1 \frac{\pi D_1^4}{32 l_1} \left(1 - \frac{d^4}{D^4}\right) \theta_1^{(j)} = T_{Ej} \quad (6)$$

Nodes 2–3: meshing pair A

$$\begin{cases} J_2 \ddot{\theta}_2^{(j)} + [F_{2,3}^{p(j)}(t) + F_{2,3}^{d(j)}(t)] r_1 + G_1 \frac{\pi D_1^4}{32 l_1} \left(1 - \frac{d^4}{D^4}\right) (\theta_2^{(j)} - \theta_1^{(j)}) = 0, \\ J_3 \ddot{\theta}_3^{(j)} - [F_{2,3}^{p(j)}(t) + F_{2,3}^{d(j)}(t)] r_1 + G_4 \frac{\pi D_4^4}{32 l_4} \left(1 - \frac{d_4^4}{D_4^4}\right) (\theta_3^{(j)} - \theta_4^{(j)}) = 0 \end{cases} \quad (7)$$

Node 4: shaft node

$$\left\{ J_4 \ddot{\theta}_4^{(j)} - G_4 \frac{\pi D_4^4}{32 l_4} \left(1 - \frac{d_4^4}{D_4^4}\right) (\theta_3^{(j)} - \theta_4^{(j)}) + G_4 \frac{\pi D_4^4}{32 l_4} \left(1 - \frac{d_4^4}{D_4^4}\right) (\theta_4^{(j)} - \theta_5^{(j)}) = 0 \right. \quad (8)$$

Nodes 5–6: meshing pair B

$$\begin{cases} J_5 \ddot{\theta}_5^{(j)} + [F_{5,6}^{p(j)}(t) + F_{5,6}^{d(j)}(t)] r_5 + G_4 \frac{\pi D_4^4}{32 l_4} \left(1 - \frac{d_4^4}{D_4^4}\right) (\theta_5^{(j)} - \theta_4^{(j)}) = 0, \\ J_6 \ddot{\theta}_6^{(j)} - [F_{5,6}^{p(j)}(t) + F_{5,6}^{d(j)}(t)] r_6 + G_7 \frac{\pi D_7^4}{32 l_7} \left(1 - \frac{d_7^4}{D_7^4}\right) (\theta_6^{(j)} - \theta_7^{(j)}) = 0 \end{cases} \quad (9)$$

Node 7: shaft node

$$\left\{ J_7 \ddot{\theta}_7^{(j)} - G_7 \frac{\pi D_7^4}{32 l_7} \left(1 - \frac{d_7^4}{D_7^4}\right) (\theta_6^{(j)} - \theta_7^{(j)}) + G_7 \frac{\pi D_7^4}{32 l_7} \left(1 - \frac{d_7^4}{D_7^4}\right) (\theta_7^{(j)} - \theta_8^{(j)}) = 0 \right. \quad (10)$$

Nodes 8–10: meshing pairs C and D

$$\begin{cases} J_8 \ddot{\theta}_8^{(j)} + [F_{8-9}^p(t) + F_{8-9}^d(t)] r_8 + G_7 \frac{\pi D_7^4}{32 l_7} \left(1 - \frac{d_7^4}{D_7^4}\right) (\theta_8^{(j)} - \theta_7^{(j)}) = 0, \\ J_9 \ddot{\theta}_9 - \sum_{j=1}^3 [F_{8-9}^p(t) + F_{8-9}^d(t)] r_9 + G_{15} \frac{\pi D_{15}^4}{32 l_{15}} \left(1 - \frac{d_{15}^4}{D_{15}^4}\right) (\theta_9 - \theta_{15}) + [F_{9-10}^p(t) + F_{9-10}^d(t)] r_9 = 0, \\ J_{10} \ddot{\theta}_{10} - [F_{9-10}^p(t) + F_{9-10}^d(t)] r_{10} + G_{11} \frac{\pi D_{11}^4}{32 l_{11}} \left(1 - \frac{d_{11}^4}{D_{11}^4}\right) (\theta_{10} - \theta_{11}) = 0 \end{cases} \quad (11)$$

Node 11: shaft node

$$J_{11} \ddot{\theta}_{11} + G_{11} \frac{\pi D_{11}^4}{32 l_{11}} \left(1 - \frac{d_{11}^4}{D_{11}^4}\right) (\theta_{11} - \theta_{12}) - G_{11} \frac{\pi D_{11}^4}{32 l_{11}} \left(1 - \frac{d_{11}^4}{D_{11}^4}\right) (\theta_{10} - \theta_{11}) = 0 \quad (12)$$

Node 12: meshing pair E

$$\begin{cases} J_{12} \ddot{\theta}_{12} + [F_{12-13}^p(t) + F_{12-13}^d(t)] r_{12} + G_{11} \frac{\pi D_{11}^4}{32 l_{11}} \left(1 - \frac{d_{11}^4}{D_{11}^4}\right) (\theta_{12} - \theta_{11}) = 0, \\ J_{13} \ddot{\theta}_{13} - [F_{12-13}^p(t) + F_{12-13}^d(t)] r_{13} + G_{14} \frac{\pi D_{14}^4}{32 l_{14}} \left(1 - \frac{d_{14}^4}{D_{14}^4}\right) (\theta_{13} - \theta_{14}) = 0 \end{cases} \quad (13)$$

Node 14: shaft node

$$J_{14} \ddot{\theta}_{14} + G_{14} \frac{\pi D_{14}^4}{32 l_{14}} \left(1 - \frac{d_{14}^4}{D_{14}^4}\right) \theta_{14} = T_t \quad (14)$$

Node 15: shaft node

$$J_{15} \ddot{\theta}_{15} + G_{15} \frac{\pi D_{15}^4}{32 l_{15}} \left(1 - \frac{d_{15}^4}{D_{15}^4}\right) (\theta_{15} - \theta_{16}) - G_{15} \frac{\pi D_{15}^4}{32 l_{15}} \left(1 - \frac{d_{15}^4}{D_{15}^4}\right) (\theta_9 - \theta_{15}) = 0 \quad (15)$$

Nodes 16–23: meshing pairs F and G

$$\begin{cases} J_{16}\ddot{\theta}_{16} + \sum_{i=1}^6 [F_{spi}^p(t) + F_{spi}^d(t)] r_{16} + G_{15} \frac{\pi D_{15}^4}{32L_{15}} \left(1 - \frac{d_{15}^4}{D_{15}^4}\right) (\theta_{16} - \theta_{15}) = 0, \\ J_{pi}\ddot{\theta}_{pi} - [F_{spi}^p(t) + F_{spi}^d(t)] r_{pi} + [F_{cpi}^p(t) + F_{cpi}^d(t)] r_{pi} = 0, \\ J_{23}\ddot{\theta}_{23} - \sum_{i=1}^6 [F_{cpi}^p(t) + F_{cpi}^d(t)] r_{pi} + G_{24} \frac{\pi D_{24}^4}{32L_{24}} \left(1 - \frac{d_{24}^4}{D_{24}^4}\right) (\theta_{23} - \theta_{24}) = 0 \end{cases} \quad (16)$$

Node 24: shaft node

$$\left\{ J_{24}\ddot{\theta}_{24} + \sum_{i=1}^6 [F_{cpi}^p(t) + F_{cpi}^d(t)] r_{pi} + G_{24} \frac{\pi D_{24}^4}{32L_{24}} \left(1 - \frac{d_{24}^4}{D_{24}^4}\right) (\theta_{24} - \theta_{23}) = -T_r \right. \quad (17)$$

where T_{Ej} is denoted as the torque of engine j ($j=1,2,3$); D and d are the outer and inner diameters of the hollow shaft; G is the shear elastic modulus of the shaft; T_t is the output torque of the tail chain; and T_r is the output torque of the rotor shaft.

The equations for each DOF could be written as the following matrix-vector form:

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = \{F\}. \quad (18)$$

The excitation $\{F\}$ could be expanded to the Fourier series with the fundamental frequency as well; k^{th} order excitation is

$$F_k = \{A_1\}_k \sin \omega_k t + \{A_2\}_k \cos \omega_k t. \quad (19)$$

The excitation causes the system to generate a response [22]:

$$\{\Delta x\}_k = \{B_1\}_k \sin \omega_k t + \{B_2\}_k \cos \omega_k t, \quad (20)$$

where $\{B_1\}_k$ and $\{B_2\}_k$ could be solved by the following equation:

$$\begin{bmatrix} -\omega_k^2[M] + [\bar{K}] & -\omega_k[C] \\ \omega_k[C] & -\omega_k^2[M] + [\bar{K}] \end{bmatrix} \begin{Bmatrix} \{B_1\}_k \\ \{B_2\}_k \end{Bmatrix} = \begin{Bmatrix} \{A_1\}_k \\ \{A_2\}_k \end{Bmatrix}. \quad (21)$$

The dynamic response is the linear superposition of the results corresponded by each order:

$$\{\Delta x(t)\} = \sum_{k=1}^5 \{ \{B_1\}_k \sin \omega_k t + \{B_2\}_k \cos \omega_k t \}. \quad (22)$$

3. Numerical Calculation

3.1. System Parameter and Response Calculation. The parameters of nodes are shown in Table 1. In addition, the system is powered by three engines, the maximum output power of each engine is 1500 kW, and engine speed is 10,000 rpm. The output power of the rotor is 4000 kW, and the rotor speed is 300 rpm. The transmission mechanical efficiency is 95%.

The calculation results can be seen from Figure 2, and the dynamic response exhibits periodic vibrations under multi-frequency excitation caused by time-varying meshing stiffness and transmission errors; thus, the response amplitudes are different. The gear rotational frequency reduces with the transmission chain following the output branch; in other words, the deceleration effect is obvious.

By comparison, it indicates that the dynamic response of meshing pair E has the maximum amplitude due to the heavy tail output load. In addition, the magnitude of the response in meshing pair C, especially pair C3, has a relatively higher response as well, as a result of the three-input branches converging at the central gear. Regarding the planetary gear train, the response of planetary gear 1 is greater than that of

TABLE 1: Four-stage helicopter main gearbox parameters.

Element	Node	Tooth number	Module	Face width (m)	Transmission error (μm)	Initial phase of transmission error	Initial phase of meshing stiffness	Shaft D/d (m)	Shaft length (m)
Shaft element	1(<i>i</i>)	N/A	N/A	N/A	N/A	N/A	N/A	0.04/0.03	0.8
Meshing element A	2(<i>i</i>)	28	5	0.04	12; 18;	$\pi/3$; $\pi/9$;	0; 0; 0	N/A	N/A
	3(<i>i</i>)	86	5	0.04	14 12;	$4\pi/3$; $\pi/3$;	0; 0; 0	N/A	N/A
	4(<i>i</i>)	N/A	N/A	N/A	18; 14	$\pi/9$; $4\pi/3$			
	5(<i>i</i>)	38	4.5	0.35	22; 17;	$\pi/3$; $-4\pi/3$;	$\pi/3$; $\pi/9$;	N/A	N/A
Meshing element B	6(<i>i</i>)	88	4.5	0.35	15 22;	0 $-\pi/9$;	$\pi/3$; $\pi/3$;	N/A	N/A
	7(<i>i</i>)	N/A	N/A	N/A	17; 15	$-4\pi/3$; 0	$\pi/9$; $-\pi/3$	N/A	N/A
	8(<i>i</i>)	65	3	0.42	N/A	N/A	N/A	0.06/0.05	1.6
	9	144	3	0.42	8; 10	$-\pi$; $-\pi/6$	0; 0; 0	N/A	N/A
Shaft element	10	70	3	0.42	10	0; $-\pi/6$	0; 0; 0	N/A	N/A
	11	N/A	N/A	N/A	10	$-\pi/6$	0	N/A	N/A
	12	40	3.5	0.47	N/A	N/A	N/A	0.04/0.03	1.4
	13	70	3.5	0.47	12 12	$\pi/9$ $\pi/9$	0 0	N/A N/A	N/A N/A
Meshing elements C and D	14	N/A	N/A	N/A	N/A	N/A	N/A	0.04/0.03	1.1
	15	N/A	N/A	N/A	N/A	N/A	N/A	0.14/0.11	0.8
	16	68	5	0.4	13; 8; 6; 8;	$-4\pi/3$; $\pi/9$; $-\pi/7$; $-\pi/9$;	π ; 0; $4\pi/7$;	N/A	N/A
	17	37	5	0.4	12; 5 13; 8	$\pi/3$; $-\pi/9$ $-4\pi/3$; $-\pi$	0; $-\pi/3$; 0 π ; $-\pi/3$	N/A	N/A
Shaft element	18	37	5	0.4	8; 9	$\pi/9$; 0	0; $4\pi/3$	N/A	N/A
	19	37	5	0.4	6; 7	$-\pi/7$; $4\pi/3$	$4\pi/7$; $-\pi$	N/A	N/A
	20	37	5	0.4	8; 6	$-\pi/9$; $-\pi/9$	0; $-4\pi/3$	N/A	N/A
	20	37	5	0.4	8; 6	$-\pi/9$; $-\pi/9$	0; $-4\pi/3$	N/A	N/A

TABLE 1: Continued.

Element	Node	Tooth number	Module	Face width (m)	Transmission error (μm)	Initial phase of transmission error	Initial phase of meshing stiffness	Shaft D/d (m)	Shaft length (m)
	21	37	5	0.4	12; 6	$\pi/3; \pi/9$	$-\pi/3; 0$	N/A	N/A
	22	37	5	0.4	5; 8	$-\pi/9; -5\pi/3$ $-\pi; 0$	$0; \pi$ $-\pi/3$	N/A	N/A
	23	142	5	0.4	8; 9; 7 6; 6; 8	$4\pi/3; -\pi/9$ $\pi/9; -5\pi/3$	$4\pi/3$ $-\pi; 4\pi/3$ $0; \pi$	N/A	N/A
Shaft element	24	N/A	N/A	N/A	N/A	N/A	N/A	0.12/0.09	2.0

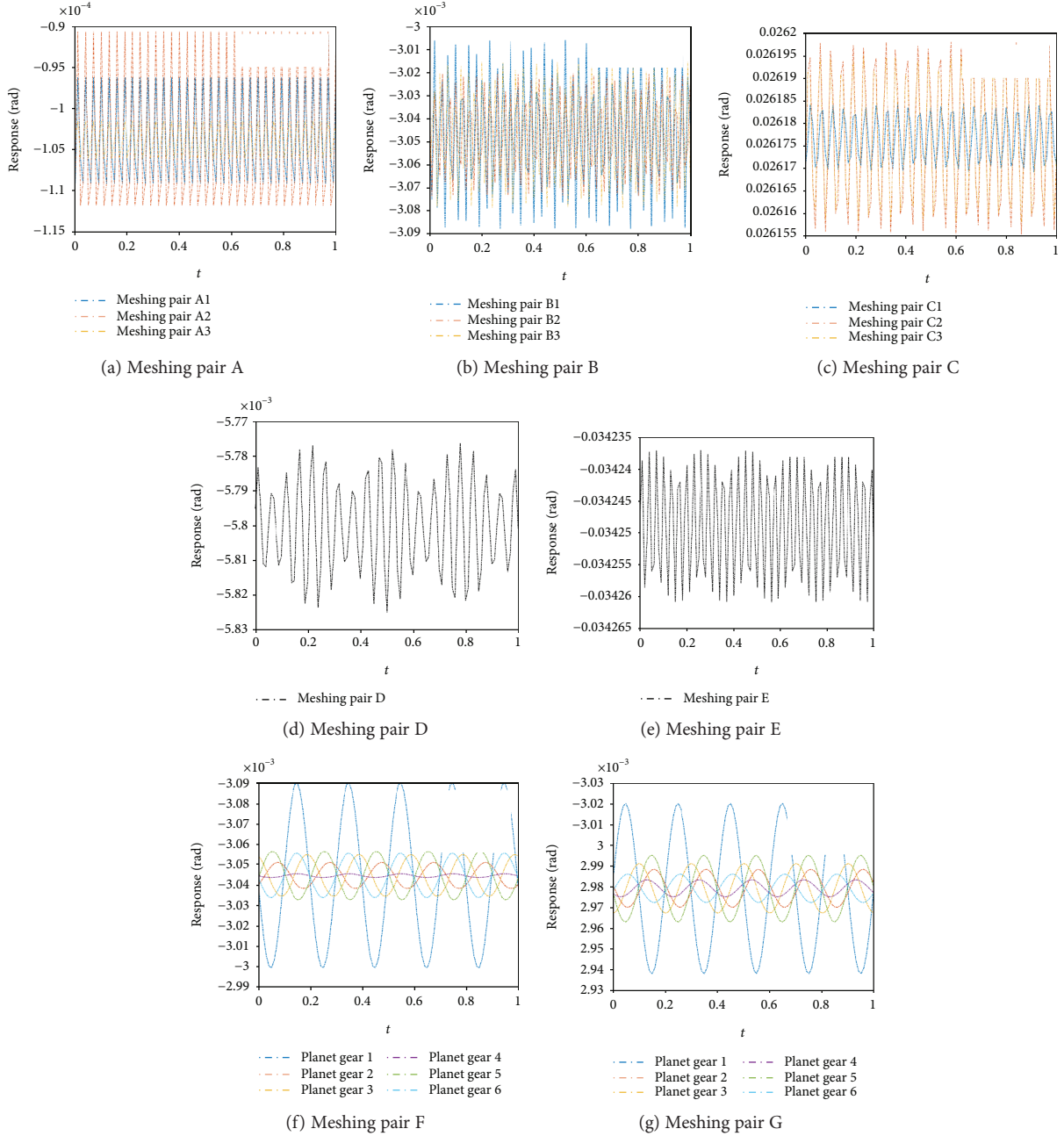


FIGURE 2: Dynamic response regarding tooth deflection in each gear pair.

the other planetary gears due to its largest transmission and manufacturing error.

3.2. Sensitivity Analysis of Responses Influenced by Shaft Wall Thickness. The sensitivity analysis method is widely used to study the key variables under many uncertainties, which can provide a parametric reference for the design of the four-stage helicopter gearbox. In this paper, the ratio of the inner and outer diameters of the shaft is regarded as a variable, the sensitivity coefficient of each pair is calculated, and the key shafts affecting the response characteristics are investigated.

The ratio of the inner and outer diameters of the shaft is defined as α ; the sensitivity coefficient of the response $x(t)$ to α is defined as follows:

$$S_{\alpha}^x = \frac{\left| \frac{(x'_{\alpha} - x_{\alpha}) / x_{\alpha}}{(\alpha' - \alpha) / \alpha} \right|}{\left| \frac{(\alpha' - \alpha) / \alpha}{(\alpha' - \alpha) / \alpha} \right|} \times 100\%, \quad (23)$$

where x and x' are the response amplitudes corresponding to the ratio of α and α' .

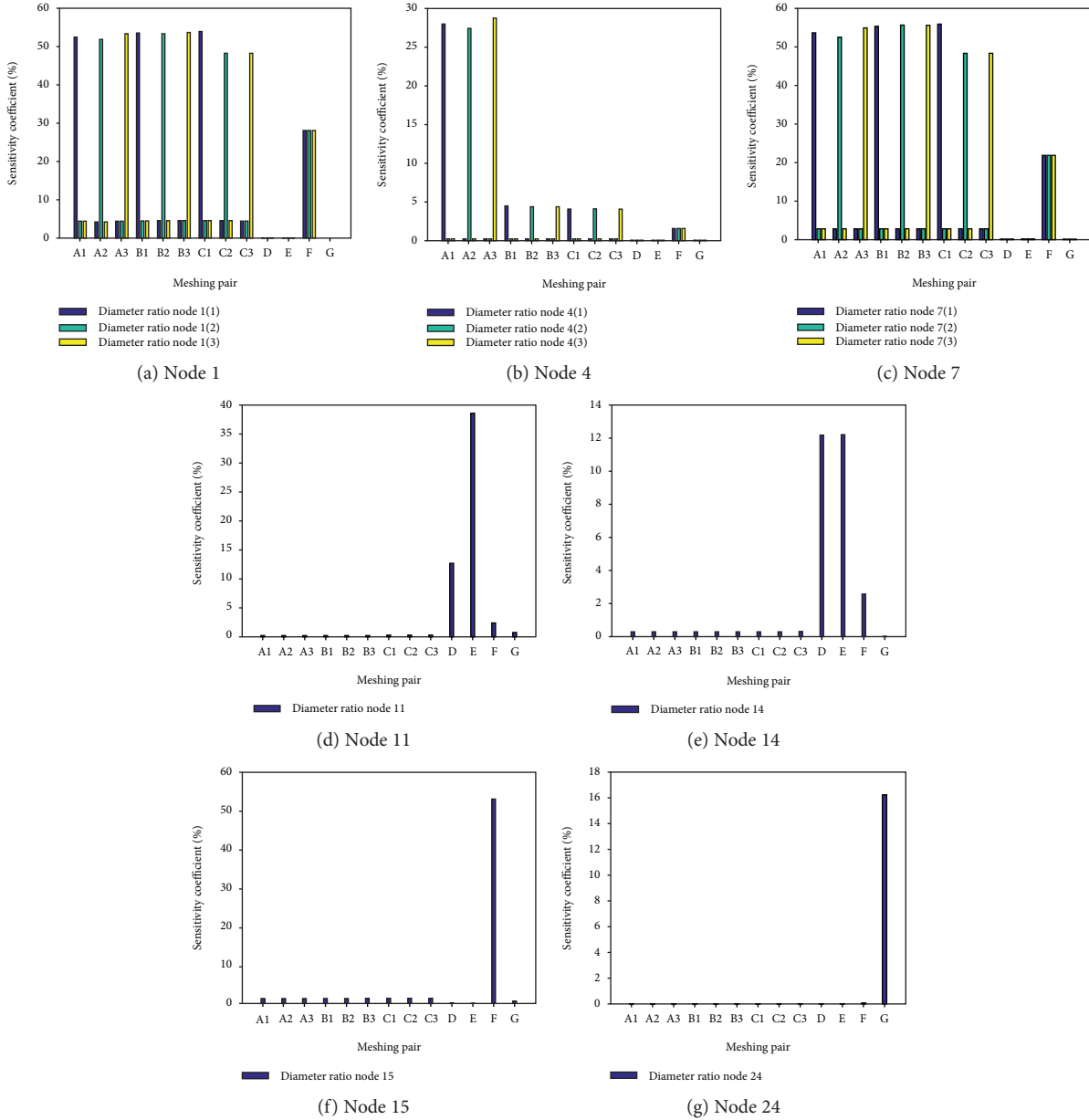


FIGURE 3: Sensitivity coefficient of each meshing pair influenced by α .

By changing the diameter ratio α , the sensitivity coefficient of each meshing pair is shown in Figure 3. According to Figure 3(a), the sensitivity coefficients of node 1 (i) to meshing pairs (A, B, and C) are greater than 50%, and the coefficient is gradually increased with the transmission chain. Therefore, the diameter ratio of node 1 is the most sensitive to meshing pair C. In addition, node 1 is also about 28% sensitive to the response of the planetary gear system, and the other meshing pairs are insensitive to this node.

According to Figure 3(b), the sensitivity coefficient of node 4 and meshing pair A is the highest, which directly affects the dynamic response of the first-stage deceleration, and the sensitivity of the subsequent stages is gradually reduced. The sensitivity of node 7 in Figure 3(c) has the similar influence law as that of node 1, and the sensitivity coefficients are quite close as well. In Figure 3(d), node 11 has the

TABLE 2: Sensitivity correspondence of wall thickness.

Meshing pair	Most sensitive node
A, B, C	Node 1 and node 7
D	Node 14
E	Node 11
F	Node 15
G	Node 24

highest sensitivity (14.5% and 38%) with its adjacent meshing pair D and meshing pair E. In Figure 3(e), node 14 is highly sensitive to the meshing pairs of the tail chain, which includes meshing pairs D and E. In Figure 3(f), the wall

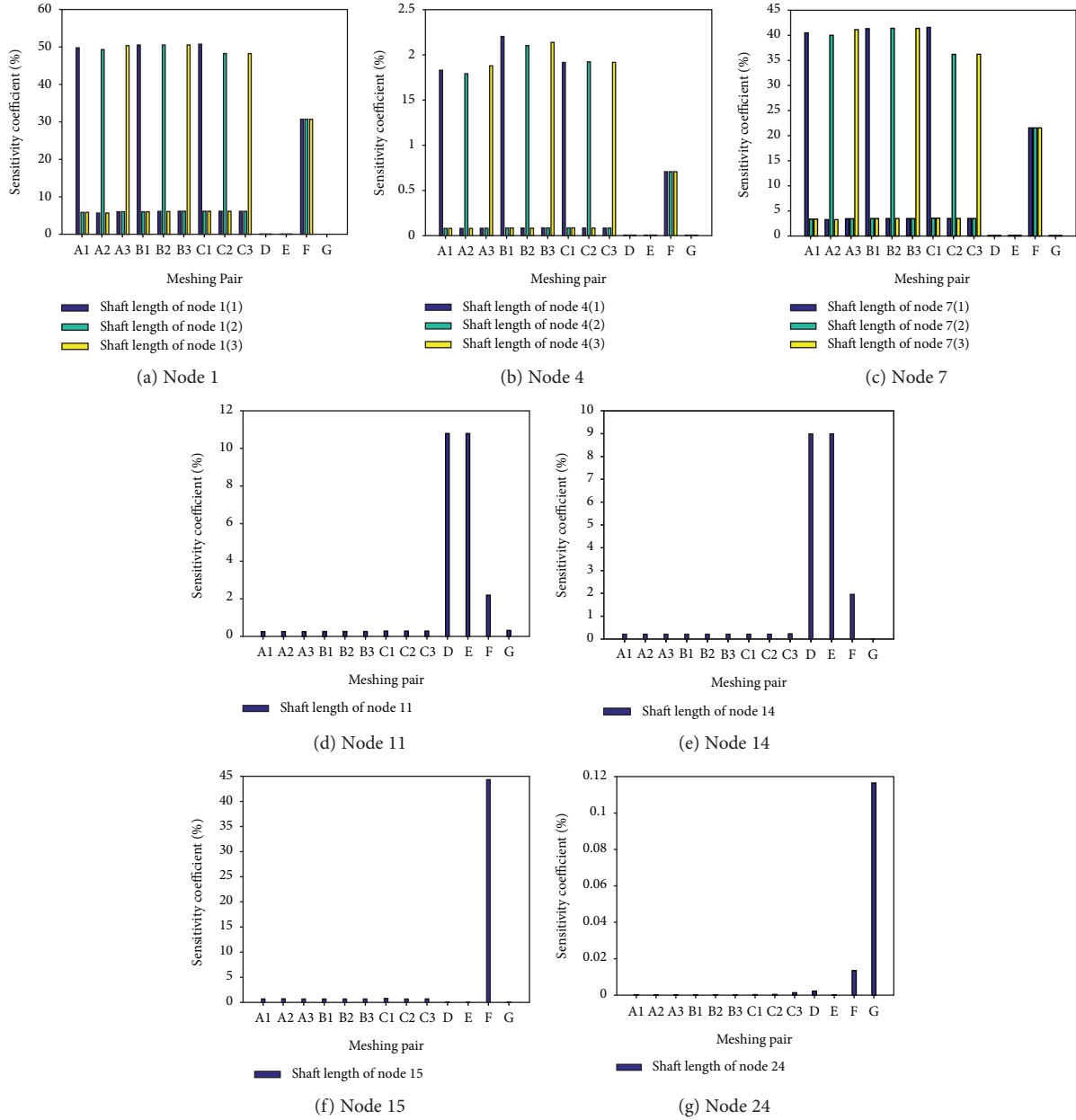


FIGURE 4: Sensitivity coefficient of each meshing pair influenced by shaft length.

thickness of node 15, the input shaft of the sun gear, directly affects the response of the planetary gear train; thus, it is the most critical node of meshing pair F. Node 24 in Figure 3(g), the rotor shaft, has the sensitivity of 17% to meshing pair G, and the coefficient is much larger than that of other nodes.

From these figures, the sensitivity correspondence and shaft selection method regarding wall thickness could be generalized in Table 2, which could be a reference for reducing the vibration shock.

3.3. Sensitivity Analysis of Responses Influenced by Shaft Length. The sensitivity coefficient of the response $x(t)$ to the shaft length is defined as follows [23]:

$$S_l^x = \frac{\left| (x_l' - x_l) / x_l \right|}{\left| (l' - l) / l \right|} \times 100\%, \quad (24)$$

where x and x' are the response amplitudes corresponding to lengths l and l' .

By changing the only variable parameter shaft length l , the sensitivity coefficient of each meshing pair is calculated in the same way, as is shown in Figure 4. From Figures 4(a)–4(c), it can be seen that the three meshing pairs of node 1, node 4, and node 7 have the highest sensitivity to three meshing pairs in the input branch, but the sensitivity coefficient of node 4 is relatively low. In Figures 4(d) and 4(e), node 11 and node 14

TABLE 3: Sensitivity correspondence of shaft length.

Meshing pair	Most sensitive node
A, B, C	Node 1
D, E	Node 11
F	Node 15
G	None

TABLE 4: Parameter optimization of the hollow shaft.

Parameter	Original group	Improved group	Optimization effect
α_1	0.75	0.6	N/A
l_1	0.8	0.6	N/A
α_{11}	0.75	0.9	N/A
l_{11}	1.4	1.8	N/A
X_{C3}	0.0262	0.0171	-34.7%
X_E	0.0343	0.0219	-36.1%

are similar in the influence law; they both have about 10% sensitivity coefficient to meshing pairs D and E. In Figure 4(f), the coefficient of node 15 to meshing pair F in the planetary gear system is 44%; with the coefficient depicted in Figure 4(f), it could be concluded that the wall thickness and shaft length of node 15 are critical parameters to internal meshing pair in the planetary gear train. In Figure 4(g), although node 24 is the most sensitive to the external meshing pair of the planetary gear system, the sensitivity coefficient is only 0.12%. In addition, the highest response sensitivity correspondence regarding shaft length is generalized in Table 3.

3.4. Structural Parameter Optimization Based on Response Sensitivity Analysis. By the sensitivity analysis of the meshing pairs based on the wall thickness and the shaft length, the most sensitive parameter has been investigated according to Tables 2 and 3. From Figure 2, it could be noted that the maximum response amplitude of the four-stage system is located at meshing pair E, according to the sensitivity analysis; the corresponding node is node 11. The second largest response amplitude is situated at meshing pair C3, and the corresponding node is node 1.

Therefore, a parameter optimization method is provided by changing the structural parameters of the shaft node without considering other errors and phase changes. It is initially set that the design range of the shaft diameter ratio α of node 1 is 0.6 to 0.9, and the range of the shaft length l is 0.6 to 1.4 m. After going through the parameters in the range, the selected diameter ratio and shaft length are shown in Table 4 as an improved group. Similarly, α_{11} is limited to 0.75~0.9, and l_{11} is 1.2~1.8 m; the original group and improved group are selected in the same way. Calculate the dynamic response of the original group and the improved group separately, as shown in Figure 5. From the figure and table, it can be concluded that the response of meshing pair C3 is reduced by 34.7% and that of meshing pair E

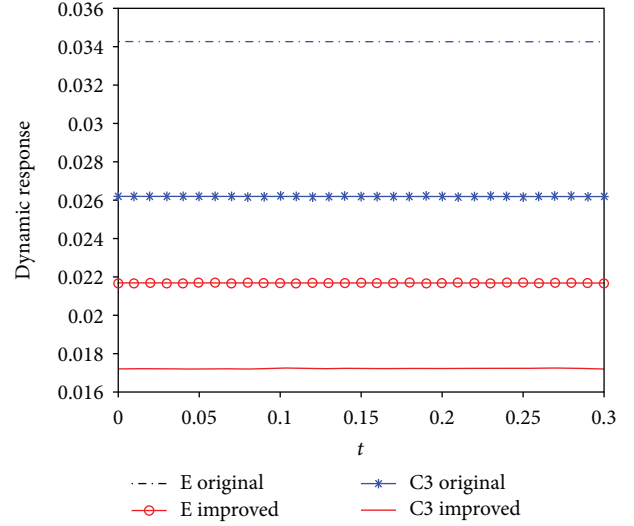


FIGURE 5: Dynamic response optimization comparison regarding tooth deflection.

is decreased by 36.1%, and the improvement effect is quite obvious.

4. Conclusion

Based on the new dynamic modeling method and the response characteristics analysis, the overall multinode dynamic model of a four-stage helicopter main gearbox is established. By solving coupling differential equations, the sensitive laws of wall thickness and length of the hollow shaft are obtained. Therefore, the structural parameter optimization method is proposed to improve the original parameter group. The results enable us to draw the following conclusions:

- (1) The magnitude of dynamic response in meshing pairs E and C3 is higher than that of other meshing pairs due to the load torque and converging influence
- (2) Each meshing pair has the most sensitive node according to the sensitivity analysis; parameter sensitivity correspondence tables are proposed for the main gearbox design
- (3) Regarding the rotor shaft (node 24), the wall thickness has great sensitivity to response; nevertheless, the influence of shaft length is negligible
- (4) In the design of the four-stage helicopter main gearbox, the shaft corresponding to node 1 should be relatively short and thick, and the shaft corresponding to node 11 should be comparatively long and thin, which can effectively reduce the response of the system
- (5) The sensitivity analysis method can be applied to quickly select the corresponding shaft node and its parameters for reducing the response, which has a good auxiliary effect on system modification. The improvement effect is obvious according to the group comparison

Data Availability

The data statement is provided in the supplementary section.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work described in this paper is fully supported by the National Natural Science Foundation of PRC (Grant no. 51775265), China Scholarship Council, NUAA Short Visiting Program (Grant no. 180403DF05), and Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant no. KYCX17_0242).

Supplementary Materials

The uploaded supplementary material is related to the manuscript “A New Mathematical Modeling Method for Four-Stage Helicopter Main Gearbox and Dynamic Response Optimization.” It could validate the authenticity and feasibility of the proposed algorithms. (*Supplementary Materials*)

References

- [1] M. Kubur, A. Kahraman, D. M. Zini, and K. Kienzle, “Dynamic analysis of a multi-shaft helical gear transmission by finite elements: model and experiment,” *Journal of Vibration and Acoustics*, vol. 126, no. 3, pp. 398–406, 2004.
- [2] J. P. Raclot and P. Velex, “Simulation of the dynamic behaviour of single and multi-stage geared systems with shape deviations and mounting errors by using a spectral method,” *Journal of Sound and Vibration*, vol. 220, no. 5, pp. 861–903, 1999.
- [3] F. K. Choy, Y. K. Tu, M. Savage, and D. P. Townsend, “Vibration signature and modal analysis of multi-stage gear transmission,” *Journal of the Franklin Institute*, vol. 328, no. 2-3, pp. 281–298, 1991.
- [4] T. Dzitkowski and A. Dymarek, “Active reduction of identified machine drive system vibrations in the form of multi-stage gear units,” *Mechanics*, vol. 20, no. 2, 2014.
- [5] Y. Chen, R.-p. Zhu, Y.-p. Xiong, and G.-h. Jin, “Analysis on natural characteristics of four-stage main transmission system in three-engine helicopter,” *Vibroengineering PROCEDIA*, vol. 12, pp. 19–23, 2017.
- [6] R. G. Parker, V. Agashe, and S. M. Vijayakar, “Dynamic response of a planetary gear system using a finite element/contact mechanics model,” *Journal of Mechanical Design*, vol. 122, no. 3, pp. 304–310, 2000.
- [7] P. Velex and L. Flamand, “Dynamic response of planetary trains to mesh parametric excitations,” *Journal of Mechanical Design*, vol. 118, no. 1, pp. 7–14, 1996.
- [8] F. Chaari, T. Fakhfakh, and M. Haddar, “Dynamic analysis of a planetary gear failure caused by tooth pitting and cracking,” *Journal of Failure Analysis and Prevention*, vol. 6, no. 2, pp. 73–78, 2006.
- [9] L. Walha, T. Fakhfakh, and M. Haddar, “Nonlinear dynamics of a two-stage gear system with mesh stiffness fluctuation, bearing flexibility and backlash,” *Mechanism and Machine Theory*, vol. 44, no. 5, pp. 1058–1069, 2009.
- [10] S. Zhou, G. Song, M. Sun, and Z. Ren, “Nonlinear dynamic response analysis on gear-rotor-bearing transmission system,” *Journal of Vibration and Control*, vol. 24, no. 9, pp. 1632–1651, 2018.
- [11] Y. Chen, R. Zhu, G. Jin, and Y. Xiong, “Influence of shaft torsional stiffness on dynamic response of four-stage main transmission system,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 6141035, 12 pages, 2018.
- [12] Z. Chen and Y. Shao, “Mesh stiffness calculation of a spur gear pair with tooth profile modification and tooth root crack,” *Mechanism and Machine Theory*, vol. 62, pp. 63–74, 2013.
- [13] D. Yang, Y. Liu, S. Li, X. Li, and L. Ma, “Gear fault diagnosis based on support vector machine optimized by artificial bee colony algorithm,” *Mechanism and Machine Theory*, vol. 90, pp. 219–229, 2015.
- [14] M. Bozca and P. Fietkau, “Empirical model based optimization of gearbox geometric design parameters to reduce rattle noise in an automotive transmission,” *Mechanism and Machine Theory*, vol. 45, no. 11, pp. 1599–1612, 2010.
- [15] M. Bozca, “Torsional vibration model based optimization of gearbox geometric design parameters to reduce rattle noise in an automotive transmission,” *Mechanism and Machine Theory*, vol. 45, no. 11, pp. 1583–1598, 2010.
- [16] M. Bozca, “Transmission error model-based optimisation of the geometric design parameters of an automotive transmission gearbox to reduce gear-rattle noise,” *Applied Acoustics*, vol. 130, pp. 247–259, 2018.
- [17] J. Lin and R. G. Parker, “Sensitivity of planetary gear natural frequencies and vibration modes to model parameters,” *Journal of Sound and Vibration*, vol. 228, no. 1, pp. 109–128, 1999.
- [18] Y. Guo and R. G. Parker, “Sensitivity of general compound planetary gear natural frequencies and vibration modes to model parameters,” *Journal of Vibration and Acoustics*, vol. 132, no. 1, pp. 11006–11013, 2010.
- [19] C.-S. Chen, S. Natsiavas, and H. D. Nelson, “Coupled lateral-torsional vibration of a gear-pair system supported by a squeeze film damper,” *Journal of Vibration and Acoustics*, vol. 120, no. 4, pp. 860–867, 1998.
- [20] Y. Chen, R. P. Zhu, Z. C. Feng, G. H. Jin, and W. Zhang, “Influence of oil film stiffness on natural characteristics of single-rotor three-input helicopter main gearbox,” *Vibroengineering PROCEDIA*, vol. 17, pp. 7–12, 2018.
- [21] D.-p. Sheng, R.-p. Zhu, G.-h. Jin, F.-x. Lu, and H.-y. Bao, “Dynamic load sharing characteristics and sun gear radial orbits of double-row planetary gear train,” *Journal of Central South University*, vol. 22, no. 10, pp. 3806–3816, 2015.
- [22] M. Li and L. Yu, “Analysis of the coupled lateral torsional vibration of a rotor-bearing system with a misaligned gear coupling,” *Journal of Sound and Vibration*, vol. 243, no. 2, pp. 283–300, 2001.
- [23] J. V. Milanović Smiee, C. P. N. Fu, R. Radosavljević, and Z. Lazarević, “Sensitivity of torsional modes and torques to uncertainty in shaft mechanical parameters,” *Electric Power Components and Systems*, vol. 29, no. 10, pp. 867–881, 2001.

Research Article

Design of a Logistics Nonlinear System for a Complex, Multiechelon, Supply Chain Network with Uncertain Demands

Aaron Guerrero Campanur,¹ Elias Olivares-Benitez ,² Pablo A. Miranda ,³
Rodolfo Eleazar Perez-Loaiza,⁴ and Jose Humberto Ablanedo-Rosas ⁵

¹ITSUruapan, Tecnológico Nacional de Mexico, Carr. Uruapan-Carapan 5555, Uruapan, Michoacan 60015, Mexico

²Facultad de Ingenieria, Universidad Panamericana, Prolongacion Calzada Circunvalacion Poniente 49, Zapopan, Jalisco 45010, Mexico

³Department of Engineering Sciences, Universidad Andres Bello, Quillota 980, Viña del Mar, 2531015, Chile

⁴ITApizaco, Tecnológico Nacional de Mexico, Av. Instituto Tecnológico S/N, Apizaco, Tlaxcala 90300, Mexico

⁵College of Business Administration, University of Texas at El Paso, 500 W. University Avenue, El Paso, TX 79968, USA

Correspondence should be addressed to Elias Olivares-Benitez; eolivaresb@up.edu.mx

Received 29 June 2018; Revised 3 October 2018; Accepted 28 October 2018; Published 12 November 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Aaron Guerrero Campanur 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.

Industrial systems, such as logistics and supply chain networks, are complex systems because they comprise a big number of interconnected actors and significant nonlinear and stochastic features. This paper analyzes a distribution network design problem for a four-echelon supply chain. The problem is represented as an inventory-location model with uncertain demand and a continuous review inventory policy. The decision variables include location at the intermediate levels and product flows between echelons. The related safety and cyclic inventory levels can be computed from these decision variables. The problem is formulated as a mixed integer nonlinear programming model to find the optimal design of the distribution network. A linearization of the nonlinear model based on a piecewise linear approximation is proposed. The objective function and nonlinear constraints are reformulated as linear formulations, transforming the original nonlinear problem into a mixed integer linear programming model. The proposed approach was tested in 50 instances to compare the nonlinear and linear formulations. The results prove that the proposed linearization outperforms the nonlinear formulation achieving convergence to a better local optimum with shorter computational time. This method provides flexibility to the decision-maker allowing the analysis of scenarios in a shorter time.

1. Introduction and Context of the Problem

Supply chain design is a critical strategy for achieving competitiveness in current global business environments. The supply chain consists of all functions involved, directly and indirectly, in meeting customer needs. Decision-making in supply chain management is classified into three hierarchical levels: strategic (long-term), tactical (medium-term), and operational (short-term). Frequently, these decisions are analyzed and solved independently at each planning stage, which generates an overall suboptimal solution when compared to solutions from comprehensive models. Therefore, supply chain management demands the use of new strategies and technologies to meet the current challenges of economic

globalization [1], especially the highly dynamic behavior of customer demand. This characteristic imposes a greater challenge in developing and designing responsive and efficient supply chain networks. The mathematical formulation in the operational planning stage is highly critical and must model the relevant characteristics of the supply chain.

Facility location problems, which are typically used to design distribution networks, involve determining the sites where to install resources, as well as the assignment of potential customers to those resources. This family of problems typically assumes a linear cost function and a set of deterministic customer demand. These assumptions avoid modelling interactions between facility location and inventory management decisions. According to these facts, the

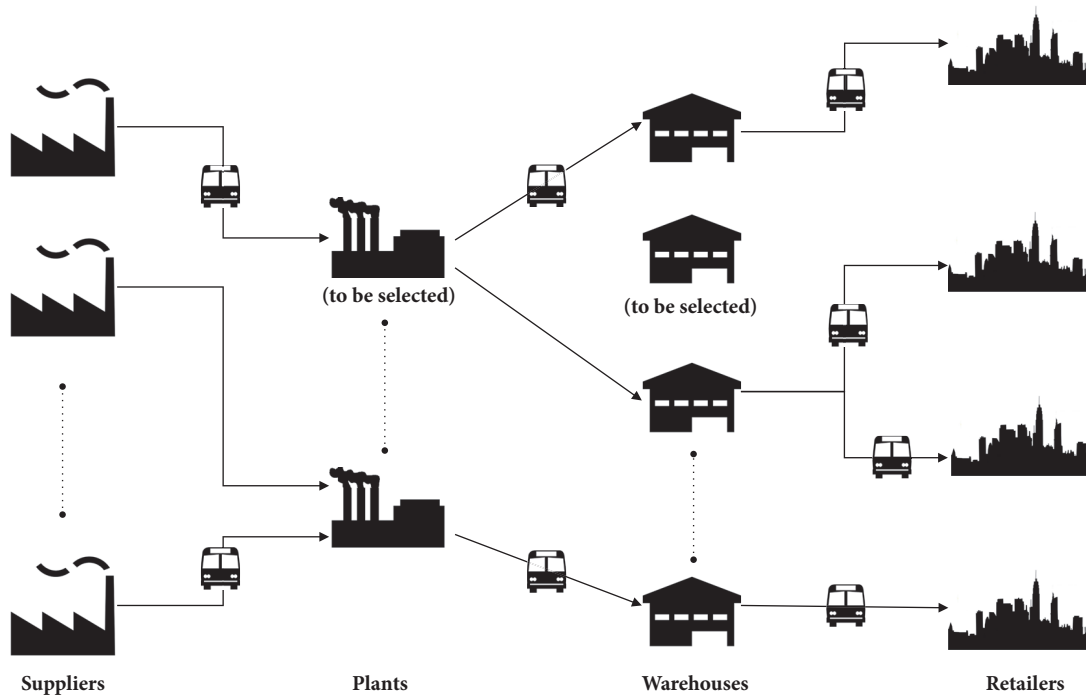


FIGURE 1: Four-echelon supply chain.

inventory-location research literature is aimed to integrate the demand uncertainty and the risk-pooling effect into supply chain network design [2].

This paper presents a mixed integer nonlinear programming formulation aimed to optimize the distribution network design. The model includes decisions on plants and warehouses location, transportation and assignment between facilities (plants-warehouses, warehouses-retailers), supplier selection, and implicit inventory decisions. The model is named a Four-Echelon Inventory-Location Problem with Supplier Selection (4EILP-SS). The mathematical formulation models a four-echelon supply chain, with a set of potential sites for allocation of plants and warehouses, as shown in Figure 1. Each warehouse and plant has a limited capacity and manages a continuous review inventory policy to satisfy the demand of retailers and warehouses, respectively.

The objective is to determine the flow of material in each echelon and the location of plants and warehouses. Inventory management decisions are simultaneously modeled with the design of the distribution network. The problem of locating facilities is commonly used as a base for the design of supply chains. However, the effect of risk integration by inventory management decisions is not directly modeled by the classic facility location problem (FLP). Additionally, supplier selection is a relevant issue because it affects many important aspects of the supply chain performance. In some real cases, for example in the context of the automotive industry, it is common to consider the involvement of suppliers in the supply chain planning stage. Supplier selection is a complex process that involves identifying potential suppliers, requesting information, establishing contract terms,

negotiation, and evaluation. Several criteria are considered for supplier selection: quality, delivery time, price, service, and flexibility, among others. In the problem studied, we are assuming that the available suppliers were preselected considering several of the criteria described before, and the last decision was made based on the outcome of the optimization model that considers cost, lead time, and capacity. This is a common practice in industry, especially automotive industry, which is the motivation of this work, where an exclusive group of suppliers has the appropriate certifications by Original Equipment Manufacturers to participate in bids and requests.

Additionally, in the recent years, there has been an offshore moving of automotive manufacturing plants from industrialized countries to emerging economies. This happened in China, India, Mexico, Brazil, Thailand, and Turkey, among others [3]. This required the design of new supply chains to meet the demand, including new plants and warehouses.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature in supply chain design, with special emphasis on risk integration that results from the integration of an inventory management policy. Section 3 describes the features of the problem and presents two mathematical formulations: a nonlinear mixed integer programming model and a linearization of the aforementioned model using a piecewise linear function to obtain a linear mixed integer programming model. Section 4 presents the experimentation and results for some instances. Lastly, Section 5 presents the conclusions and final remarks.

2. Literature Review

Inventory-location problems (ILPs) have been extensively studied over the past two decades. This section presents a review of the main contributions reported in the literature, showing trends in formulations and solution approaches. Although ILPs are extensions of Facility Location Problems (FLPs), this paper does not discuss the related FLP literature. Interested readers may consult Hamacher and Drezner [4], Daskin [5], Nickel and Puerto [6], Melo et al. [7], and Eiselt and Marianov [8, 9].

Jayaraman [10] studied a basic ILP, based on a mixed integer programming model for locating warehouses and satisfying customer's demand for different products. The inventory modeling included the EOQ (Economic Order Quantity) model with fixed lot size and deterministic demand. Nozick and Turnquist [11] investigated an ILP, which included fixed warehouse location costs, and transportation and inventory costs. The problem was solved using a combination of a greedy and improvement algorithm. Erlebacher and Meller [12] presented an ILP with stochastic demands for a two-stage supply chain network, along with some heuristic procedures to solve a variety of instances. Another two-echelon ILP was studied by Nozick and Turnquist [13]. The model minimizes the inventory and unfulfilled demand costs. They used a combination of iterative greedy and improvement algorithms as a solution strategy. Nozick and Turnquist [14] analyzed a distribution system and the trade-off between facility costs, inventory costs, transportation costs, and customer responsiveness. They built an efficient frontier for assessing the different solution configurations.

Daskin et al. [15], Shen et al. [16], and Miranda and Garrido [17] presented similar versions of ILPs, proposing mixed nonlinear integer-programming formulations. The model considers a continuous control inventory policy with stochastic demand, the safety stock cost is determined by a chance-constrained approach, and the ordering decisions are based on the EOQ model with order quantities as decision variables. Daskin et al. [15] and Miranda and Garrido [17] solved the model using Lagrangian relaxation, while Shen et al. [16] reformulated the model as a set-covering problem, and developed a column-generation algorithm. Shu et al. [18] studied the column generation approach proposed by Shen et al. [16], showing that the pricing problem gives rise to a new class of the submodular function minimization problem. They discussed how the general pricing problem could be solved by exploiting certain special structures.

Snyder et al. [19] presented a stochastic programming-based ILP including some random parameters defined by discrete scenarios. The location model explicitly handles the economies of scale and risk-pooling effects that result from consolidating inventory sites. They used a Lagrangian relaxation-based algorithm. Shu et al. [20] proposed a two-stage stochastic model to address an integrated location, and two-echelon, inventory network design under uncertainty. They defined the problem as a two-stage nonlinear discrete optimization problem. The first stage decides which warehouses to open and the second one decides the warehouse-retailer assignments and the

two-echelon inventory replenishment. They used a set covering formulation transformed into a pricing problem, which was solved using column generation.

ILPs with inventory capacity constraints were analyzed by Miranda and Garrido [21, 22] and Ozsen et al. [23]. Ozsen et al. [23] introduced a 100% service level constraint for inventory capacity to a previous ILP presented by Daskin et al. [15], while Miranda and Garrido [21, 22] discussed how the inventory capacity constraint can be controlled by a previously defined service level (usually less than 100%), based on a chance-constrained formulation. Ozsen et al. [23] and Miranda and Garrido [21, 22] developed a Lagrangian relaxation-based solution approach. Miranda and Cabrera [24] and Cabrera et al. [25] introduced an ILP with periodic inventory control review, where inventory capacity constraints are modeled in a stochastic manner.

Notice that all previous works are focused on optimizing location and inventory costs and decisions are taken at a single stage of the supply chain (i.e., warehouses), along with transport or assignment decisions (plant-warehouses and/or warehouses-retailers). In the works described forward, some additional considerations of costs and operations management decisions are integrated into similar ILP formulations.

Kang and Kim [26] investigated an ILP for warehouse location, integrating inventory costs and assignment decisions at the warehouse and retailers. They formulated a nonlinear mixed integer programming model and developed a Lagrangian relaxation-based heuristic.

Silva and Gao [27] introduced a joint replenishment ILP. They proposed a Greedy Randomized Adaptive Search Procedure to solve the problem. Tancrez et al. [28] studied an ILP in three-level supply chain networks. The decision set included distribution centers location, flows allocation, and shipment sizes. They proposed a nonlinear formulation that decomposes into a closed-form equation and a linear programming model when the distribution center flows are fixed. They developed an iterative heuristic that estimates the distribution center flows a priori, solves the linear program, and then improves the distribution center flows estimations. Similarly, Diabat et al. [29] studied a simplified multi-echelon ILP (with location decisions at a single level), where the problem was formulated as a nonlinear mixed integer program and was solved using a Lagrangian relaxation approach.

Shahabi et al. [30] developed a model to coordinate facility location and inventory control in a four-echelon supply chain with hubs, which helps in reducing transportation costs by consolidating shipments. There were three decisions: warehouses and hubs locations, assignment of suppliers and retailers, and inventory control decisions and costs at a single stage (warehouses). A mixed integer nonlinear programming formulation was first presented and then transformed into a compact conic mixed integer programming formulation. Commercial solvers were used to solve the problem.

Shen and Daskin [31], Gaur and Ravindran [32], Miranda and Garrido [33], Mak and Shen [34], and Escalona et al. [35] are a variety of works in which customer or system service levels are integrated and analyzed explicitly based on previous ILP formulations. Genetic Algorithms, Lagrangian

relaxation, and two-step greedy-based heuristics are some of the approaches employed to solve the analyzed ILPs.

Atamtürk et al. [36] introduced a conic integer programming approach to reformulate a variety of ILPs. These reformulations, for different cases of nonlinearities, can be solved directly using standard optimization software packages without the need for designing specific algorithms. They concluded that their approach led to similar or better computational times than other approaches reported in the literature, and they claimed the applicability of their approach to model more general, related problems.

Kaya and Urek [37] proposed a model for a closed loop supply chain that integrates location, inventory and pricing decisions. The location decision is applied to the level of collection-distribution centers. Inventory is modeled without uncertainty in demand but affected by a price-demand function. The mixed integer nonlinear programming model is solved using commercial optimization software for small instances, and several heuristics for large instances. The heuristics include a Piecewise linearization, and hybrids of Simulated Annealing, Tabu Search, Genetic Algorithms, and Variable Neighborhood Search.

Schuster Puga and Tancrez [38] analyze a supply chain network with three levels. The flow of product is from a central plant to distribution centers to retailers. Location decisions are applied to the distribution centers. The model developed considers inventory decisions, and transportation costs follow a full truckload approach. They developed an iterative heuristic to solve the continuous nonlinear programming model. Also, they developed a conic quadratic mixed-integer program to transform the original formulation. They solved instances with up to 1000 retailers and 1000 distribution centers.

Escalona et al. [39] studied a supply chain network with three levels and inventory-location decisions. They add a stratification of customers according to classes based on the service level required. The mixed integer nonlinear programming model is transformed into a conic quadratic mixed-integer problem. Their main objective is to analyze the effect of different inventory policies on the configuration of the supply chain network.

Tapia-Ubeda et al. [40] studied an ILP for a supply chain consisting of two levels, with warehouses shipping product to customers. They developed a mixed integer nonlinear programming model to be solved using a Benders Decomposition algorithm. The nonlinear nature of the original model is transferred to the subproblems, but for the instances solved the subproblems can be solved to optimality without a big effort.

It must be noticed that most of the previous ILP formulations integrate location decisions at one or two stages (warehouse location, plant location and supplier selection, warehouse and consolidating center location, etc.). Inventory costs and decisions are modeled at one or two levels (warehouses, warehouses, and plants, warehouses and retailers, etc.), but none of these previous work simultaneously integrates location and inventory management decisions at two stages in the supply chain. Therefore, this research paper fills an important gap in the field of supply chain network design.

To the best of our knowledge, only a previous work by ourselves has considered the design of such a complex supply chain as in this research, with some differences in the model and more important, in the solution method. Perez Loaiza et al. [41] studied a supply chain of four levels with location and inventory decisions and a bi-objective approach. The mixed integer nonlinear programming model is solved using an evolutionary algorithm. Pareto fronts are compared with respect to solutions obtained with commercial optimization software.

The following part of the literature review discusses linearization strategies for nonlinear problems. You and Grossmann [42] addressed an ILP from the chemical industry, the design of a multiechelon supply chain and inventory system in the presence of uncertain customer demands. A mixed integer nonlinear program modeled the transportation, inventory, and network structure of a multiechelon supply chain. The model had a non-convex objective function. They reformulated the problem as a separable concave program. A spatial decomposition algorithm based on the integration of Lagrangian relaxation and piecewise linear approximation was proposed to solve the model. The transformation assures all the constraints are linear and the only nonlinear terms are univariate concave terms in the objective function. Petridis [43] addressed a multiproduct, multiechelon supply chain network with demand uncertainty. Decisions about the selection of facilities and their capacity are made. Information about the flows of products transferred and the safety stock at each distribution center was derived. The lead time of an order to a customer is computed, using the probabilities of overstocking and understocking. The problem was formulated as a single period mixed integer nonlinear programming problem. Linearization techniques for selected nonlinear terms of the models were explored in order to reduce the computational effort for solving the model. The linearization was achieved by rewriting one of the nonlinear constraints that had a product of a continuous and a binary variable. The objective function was not linearized.

According to the discussed literature review, this paper contributes with an inventory-location model, under continuous review policy, to optimize a four-echelon supply chain network, encompassing multiple suppliers, plants, warehouses, and retailers. The problem is formulated as a mixed integer nonlinear programming model, considering a single item, with stochastic demands across the supply chain network. Decision variables include warehouse and plant location, plant-warehouse assignments, retailer assignment to warehouses, and shipments from suppliers to plants. This ILP integrates inventory costs and decisions at plants and warehouses. As it can be observed from the related literature, the proposed network design model of a four-echelon supply chain has a nonlinear formulation that arises when inventory aspects are integrated into facility location problems, yielding to a nonlinear mixed integer programming model.

Nonlinear mixed integer programming models are extremely hard to solve to optimality, particularly when commercial optimization solvers are employed. They combine the mixed integer programming (NP-Hard) nature with nonlinear and non-convex components. In many cases, researchers

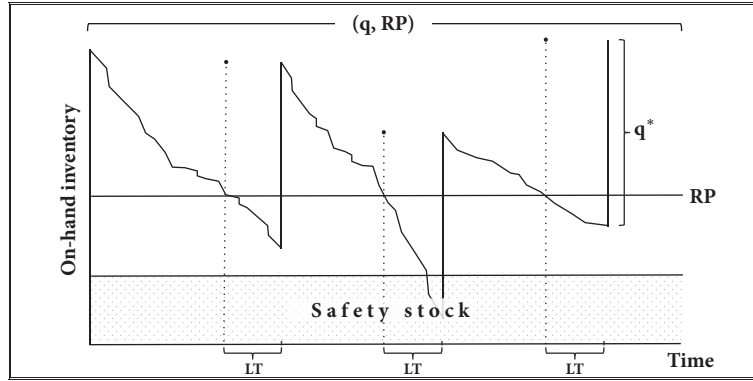


FIGURE 2: Continuous review inventory control policy performance.

and practitioners face two major challenges: local optimum solutions and prohibitive computational times for tackling practical problems. Moreover, these challenges increase when more stages and decision variables are added to the model, such as in this research article. In order to reduce the computational effort, and as a strategy to develop a heuristic framework to tackle this complex problem, a linearization of the model is proposed based on a piecewise linear approximation of the objective function, and a linear reformulation of nonlinear constraints, yielding to a mixed integer linear programming model. The approach is developed following the approach proposed by Diabat and Theodorou [44], who applied this technique in a smaller supply chain network with less supply chain stages and decisions.

3. Problem Description and Modeling

3.1. Continuous Review Based Inventory Control Policy (q, RP). A continuous review based inventory control policy (q, RP) is assumed at plants and warehouses with the aim of dealing with uncertain and stochastic demands (warehouses deal with retailer demands and plants deal with warehouse demands or orders). Figure 2 shows the evolution of the physical inventory level when this inventory control policy is assumed. When the inventory level reaches reorder point RP , an order of q units of the product is requested. The q units of product are received after a fixed and known lead time LT . In other words, RP is the critical inventory level that generates a new order. In the case of warehouses, they generate an order to plants, while in the case of plants they generate a production order or batch. In both cases, a widely accepted approach states that the reorder point at each location, RP , must be set in order to ensure that served demand arising during lead time LT will not be greater than this reorder point, at least with a fixed and known probability. This probability is also known as inventory service level.

The existing inventory level just before the order arrives at each location is known as the safety stock, while the inventory level that is observed over the safety stock is known as cyclic or working inventory.

Warehouse demand depends on customers l assigned to each warehouse k , and plant demand depends on warehouses assigned to each plant j .

To calculate the working inventory cost (WIC), the total inventory cost is estimated by

$$\text{Total inventory cost} = UC \cdot d + OC \cdot \left(\frac{d}{q}\right) + HC \cdot \left(\frac{q}{2}\right), \quad (1)$$

where UC is the purchasing cost, d is the demand, q is the order quantity, OC is the ordering cost, and HC is the inventory holding cost. The first derivative of (1) with respect to q is obtained, equal to zero, and solved for q^* . The optimal order quantity q^* is obtained as

$$q^* = \sqrt{\frac{2 \cdot d \cdot OC}{HC}}. \quad (2)$$

Then, the working inventory costs WIC , the last component of (1), is obtained using q^* from (2) in

$$WIC = HC \cdot \frac{q^*}{2} = \sqrt{2 \cdot OC \cdot HC \cdot d}. \quad (3)$$

Assuming an uncertain demand with normal distribution, the reorder point RP can be calculated according to

$$\text{Prob}(d(LT) \leq RP) = 1 - \alpha, \quad (4)$$

such that

$$RP = d \cdot LT + ZT \cdot \sqrt{u \cdot LT}, \quad (5)$$

where ZT represents the value of the Standard Normal Distribution (0,1) that accumulates a probability of $1-\alpha$ (service level). This value of ZT in (5) determines the safety stock level at each facility. The variance of the demand is given by u , and LT represents the lead time.

RP is the amount of inventory that should be available in each plant and warehouse to absorb fluctuations in demand from retailers during the lead time. Safety stock is the carried

out inventory to prevent stock outs. It describes the balance between the cost of holding inventory and profit foregone as a result of outages. In this case, it was not considered the cost of not having stock, only the safety stock cost (SSC) in each plant or warehouse. SSC is given by

$$SSC = ZT \cdot HC \cdot \sqrt{u \cdot LT}. \quad (6)$$

A similar explanation of these terms can be found in Daskin et al. [15] and Shen et al. [16].

3.2. Description of the Model Formulation. In this section, the mathematical formulation of the problem studied in this paper is presented. Section 3.2.1 discusses the mixed integer nonlinear model for distribution network design, 4EILP-SS. The linearization process for the constraints is explained in this section too. Section 3.2.2 illustrates the linearization process of both safety stock and working inventory costs using an approximation based on piecewise linear functions; thereby a mixed integer linear model is obtained (4EILP-SS-L). The motivation to linearize the original model is based on the characteristics of commercial optimization software. In most of the cases, the standard methods used to solve nonlinear models do not guarantee to find a global optimum solution, and when this is possible the computational time may be too long and unreasonable.

3.2.1. Mixed Integer Nonlinear Model

Notation and Definitions

Sets

- I : Set of suppliers, $i = 1, 2, \dots, |I|$.
- J : Set of potential plant locations, $j = 1, 2, \dots, |J|$.
- K : Set of potential warehouse locations, $k = 1, 2, \dots, |K|$.
- L : Set of retailers, $l = 1, 2, \dots, |L|$.

Parameters

- cs_i : Production capacity of supplier i .
- cw_k : Maximum capacity of warehouse k .
- cp_j : Maximum capacity of plant j .
- d_i : Average demand for each retailer l .
- FCp_j : Fixed cost of opening a plant at site j .
- FCw_k : Fixed cost of opening a warehouse at site k .
- HC_j^p : Inventory holding cost per unit in plant j .
- HC_k^w : Inventory holding cost per unit in warehouse k .
- LT_j^p : Deterministic delivery lead time to plant j .

LT_k^w : Deterministic delivery lead time to warehouse k .

OC_j^p : Ordering cost at plant j .

OC_k^w : Ordering cost at warehouse k .

TC_{ij}^a : Transportation cost per unit from supplier i to plant j .

TC_{jk}^b : Transportation cost per unit from plant j to warehouse k .

TC_{kl}^c : Transportation cost per unit from warehouse k to retailer l .

UC_i^s : Purchase cost per unit at supplier i .

u_i : Demand variance at retailer l .

$Z_{1-\alpha}$: Values of ZT for a given service level, assuming a Normal Distribution.

τ_j : Working inventory cost at plant j .

φ_j : Safety stock at plant j .

ψ_k : Working inventory cost at warehouse k .

ω_k : Safety stock at warehouse k .

Decision Variables

x_k : 1 if warehouse k is opened; 0 otherwise.

y_j : 1 if plant j is opened; 0 otherwise.

w_{jk} : 1 if plant j serves warehouse k ; 0 otherwise.

z_{kl} : 1 if warehouse k serves retailer l ; 0 otherwise.

β_{ij} : Number of units of product produced by supplier i and shipped to plant j .

Auxiliary Variables

δ_j : Average demand assigned to plant j .

η_j : Demand variance assigned to plant j .

λ_k : Average demand assigned to warehouse k .

π_k : Demand variance assigned to warehouse k .

τ_j : Product of ordering cost, inventory holding cost and demand assigned to plant j .

φ_j : Product of lead time and demand variance assigned to plant j .

ψ_k : Product of ordering cost, inventory holding cost and demand assigned to warehouse k .

ω_k : Product of lead time and demand variance assigned to warehouse k .

Thus, the mathematical model for the problem is formulated as

$$\begin{aligned} \min \quad & Z1 \\ & = \sum_{j=1}^{|J|} \left(FCp_j \cdot y_j + \sqrt{\tau_j} + Z_{1-\alpha} \cdot HC_j^p \cdot \sqrt{\varphi_j} + \sum_{i=1}^{|I|} \left((UC_i^s + TC_{ij}^a) \cdot \beta_{ij} \right) + \sum_{k=1}^{|K|} \left(TC_{jk}^b \cdot \lambda_k \cdot w_{jk} \right) \right) \end{aligned}$$

$$+ \sum_{k=1}^{|K|} \left(FCw_k \cdot x_k + \sqrt{\psi_k} + Z_{1-\alpha} \cdot HC_k^w \cdot \sqrt{\omega_k} + \sum_{l=1}^{|L|} (TC_{kl}^c \cdot d_l \cdot z_{kl}) \right), \quad (7)$$

$$\text{subject to } \sum_{k=1}^{|K|} z_{kl} = 1, \quad \forall l = 1, \dots, |L|, \quad (8)$$

$$\sum_{j=1}^{|J|} w_{jk} = x_k, \quad \forall k = 1, \dots, |K|, \quad (9)$$

$$\sum_{i=1}^{|I|} \beta_{ij} \geq y_j, \quad \forall j = 1, \dots, |J|, \quad (10)$$

$$\sum_{l=1}^{|L|} d_l \cdot z_{kl} = \lambda_k, \quad \forall k = 1, \dots, |K|, \quad (11)$$

$$\sum_{l=1}^{|L|} u_l \cdot z_{kl} = \pi_k, \quad \forall k = 1, \dots, |K|, \quad (12)$$

$$\sum_{k=1}^{|K|} \lambda_k \cdot w_{jk} = \delta_j, \quad \forall j = 1, \dots, |J|, \quad (13)$$

$$\sum_{k=1}^{|K|} \pi_k \cdot w_{jk} = \eta_j, \quad \forall j = 1, \dots, |J|, \quad (14)$$

$$\sum_{i=1}^{|I|} \beta_{ij} = \delta_j, \quad \forall j = 1, \dots, |J|, \quad (15)$$

$$\tau_j = 2 \cdot OC_j^p \cdot HC_j^p \cdot \delta_j, \quad \forall j = 1, \dots, |J|, \quad (16)$$

$$\varphi_j = LT_j^p \cdot \eta_j, \quad \forall j = 1, \dots, |J|, \quad (17)$$

$$\psi_k = 2 \cdot OC_k^w \cdot HC_k^w \cdot \lambda_k, \quad \forall k = 1, \dots, |K|, \quad (18)$$

$$\omega_k = LT_k^w \cdot \pi_k, \quad \forall k = 1, \dots, |K|, \quad (19)$$

$$\lambda_k \leq cw_k \cdot x_k, \quad \forall k = 1, \dots, |K|, \quad (20)$$

$$\delta_j \leq cp_j \cdot y_j, \quad \forall j = 1, \dots, |J|, \quad (21)$$

$$\sum_{j=1}^{|J|} \beta_{ij} \leq cs_i, \quad \forall i = 1, \dots, |I|, \quad (22)$$

$$w_{jk}, x_k, y_j, z_{kl} \in \{0, 1\}, \quad \forall j = 1, \dots, |J|, \quad \forall k = 1, \dots, |K|, \quad \forall l = 1, \dots, |L|, \quad (23)$$

$$\beta_{ij} \geq 0, \quad \forall i = 1, \dots, |I|, \quad \forall j = 1, \dots, |J|. \quad (24)$$

The objective function of the optimization model, 4EILP-SS, is to minimize the total cost given by the cost components in (7). These components represent the fixed cost if plant j and warehouse k are open, the cost of safety stock and working inventory at each plant and warehouse, the purchase and transportation cost to serve the average demand shipped

from supplier i to plant j , and the transportation cost from plant j to warehouse k and from there to retailer l . Equation (8) ensures that each retailer is served exactly by one warehouse. Constraints in (9) guarantee that each opened warehouse is served only by one plant. These constraints follow the single source assumption used frequently in the

literature and practice. Equation (10) indicates that suppliers should ship at least one unit to open plants. Equations (11) and (12) compute the transferred demand average and variance from retailers to warehouses. Similarly, nonlinear constraints in (13) and (14) compute the demand average and its variance transferred from warehouses to plants. Equation (15) meets the demand from suppliers to plants. Equations (16) to (19) calculate auxiliary variables used into the objective function. Their use is not essential at this point, but these equations are helpful later to address the nonlinearity of the model. Inequalities in (20), (21), and (22) ensure that the capacities of the warehouses, plants, and suppliers are not exceeded, respectively, for open facilities. Finally, (23) and (24) show the nature of the variables.

A reformulation of the constraints in (13) and (14) is presented forward. In these constraints, the product of an integer auxiliary variable and a binary decision variable occurs, for instance λ_k and w_{jk} in (13). Combining (13) and (11), the first can be reformulated as

$$\sum_{k=1}^{|K|} \sum_{l=1}^{|L|} d_l \cdot w_{jk} \cdot z_{kl} = \delta_j, \quad \forall j = 1, \dots, |J|. \quad (25)$$

Nonlinear constraints are produced by the product of two binary decision variables, i.e., $w_{jk} \cdot z_{kl}$. The nonlinearity in (25) is removed when

$$v_{jkl} \geq w_{jk} + z_{kl} - 1, \quad \forall j = 1, \dots, |J|, \quad \forall k = 1, \dots, |K|, \quad \forall l = 1, \dots, |L|, \quad (26)$$

$$v_{jkl} \leq w_{jk}, \quad \forall j = 1, \dots, |J|, \quad \forall k = 1, \dots, |K|, \quad \forall l = 1, \dots, |L|, \quad (27)$$

$$v_{jkl} \leq z_{kl}, \quad \forall j = 1, \dots, |J|, \quad \forall k = 1, \dots, |K|, \quad \forall l = 1, \dots, |L|, \quad (28)$$

are added, with the introduction of a binary variable v_{jkl} , such that if $w_{jk} = 1$ and $z_{kl} = 1$, then variable $v_{jkl} = 1$, and otherwise v_{jkl} may take the value of zero [45].

If $w_{kl} \cdot z_{jk}$ is substituted by v_{jkl} in (25), the result will be the linear constraints

$$\sum_{k=1}^{|K|} \sum_{l=1}^{|L|} d_l \cdot v_{jkl} = \delta_j, \quad \forall j = 1, \dots, |J|, \quad (29)$$

instead of those in (13). The constraints in (29) involve the product of a parameter d_l and a binary decision variable v_{jkl} . Using the same procedure, constraints in (14) are reformulated and presented as

$$\sum_{k=1}^{|K|} \sum_{l=1}^{|L|} u_l \cdot v_{jkl} = \eta_j, \quad \forall j = 1, \dots, |J|. \quad (30)$$

Once a solution for the model is obtained, the order quantity and reorder point at each open facility can be calculated. Using the results for variables δ_j as demands for plants, with their corresponding values for the ordering cost and holding

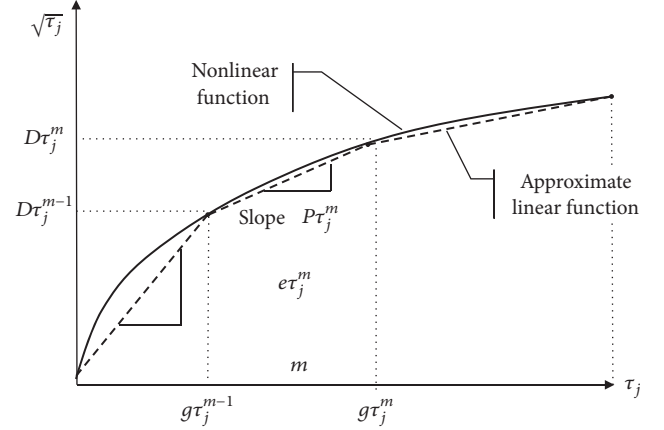


FIGURE 3: Piecewise linear function PLF of WIC.

cost, the order quantity at each plant can be calculated using (2). Using the results for variables δ_j as demands and η_j as demand variances for plants, with their corresponding values for the lead time and service level factor, the reorder point at each plant can be calculated using (5). The same can be done for warehouses using the values of λ_k as demands and π_k as demand variances.

3.2.2. Linear Approximation of the Objective Function. This section presents a linear approximation approach for the safety stock costs SSC and working inventory costs WIC. WIC in plant j is calculated as $\sqrt{\tau_j}$, as derived from (3) and (16). The approximation is based on a piecewise linear function (PLF) defined by two or more straight-lines, presented in different sections of the domain of the function as shown in Figure 3.

Considering a partition of τ_j in M sections, the function $\sqrt{\tau_j}$ is approximated linearly by the expression $D\tau_j^{m-1} \cdot e\tau_j^m + P\tau_j^m \cdot g\tau_j^m$, where $D\tau_j^1 < D\tau_j^2 < D\tau_j^3 \dots < D\tau_j^m$ and $P\tau_j^1 > P\tau_j^2 > P\tau_j^3 \dots > P\tau_j^m$, for $m = 1, \dots, M$ sections. Here, $g\tau_j^m$ is the value of τ_j at the upper bound of section m , $D\tau_j^m$ is the value of $\sqrt{\tau_j}$ at the upper bound of section m , $P\tau_j^m$ is the slope along the section m , $e\tau_j^m$ is equal to 1 if section m of the piecewise linear function of $\sqrt{\tau_j}$ is used or 0 otherwise. Also, $f\tau_j^m = \tau_j - g\tau_j^{m-1}$ are the units of τ_j in any point along section m .

The linear approximation of WIC at each plant is given by

$$\sum_{m=1}^{|M|} (D\tau_j^{m-1} \cdot e\tau_j^m + P\tau_j^m \cdot f\tau_j^m) \approx \sqrt{\tau_j}, \quad (31)$$

$$\forall j = 1, \dots, |J|,$$

$$\sum_{m=1}^{|M|} (f\tau_j^m + g\tau_j^{m-1} \cdot e\tau_j^m) = \tau_j, \quad \forall j = 1, \dots, |J|, \quad (32)$$

$$f\tau_j^m < (g\tau_j^m - g\tau_j^{m-1}) \cdot e\tau_j^m, \quad (33)$$

$$\forall j = 1, \dots, |J|, \quad \forall m = 1, \dots, |M|,$$

TABLE 1: Parameters of the base case, associated with suppliers i .

Parameter	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$	$i = 9$	$i = 10$
$cs_i =$	85	70	90	80	75	65	70	71	66	66
$UC_i^s =$	30	28	30	28	24	25	28	21	28	30

TABLE 2: Parameters of the base case, associated with plants j , and warehouses k .

Plant	$j = 1$	$j = 2$	$j = 3$	Warehouse	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$cp_j =$	150	200	110	$cw_k =$	70	80	75	70	85
$FCp_j =$	187500	250000	137500	$FCw_k =$	131250	150000	97500	131250	121875
$HC_j^p =$	167	134	164	$HC_k^w =$	184	220	190	189	205
$LT_j^p =$	1	3	2	$LT_k^w =$	2	1	2	1	2
$OC_j^p =$	646	657	656	$OC_k^w =$	743	850	552	743	690

TABLE 3: Parameters of the base case, demand, and variance of retailers l .

Retailer	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$	$l = 8$	$l = 9$	$l = 10$	$l = 11$	$l = 12$	$l = 13$
Demand (d_l) =	5	10	10	6	8	5	13	23	12	10	6	7	20
Variance (u_l) =	46	40	25	30	87	50	55	65	30	68	70	76	30
Retailer	$l = 14$	$l = 15$	$l = 16$	$l = 17$	$l = 18$	$l = 19$	$l = 20$	$l = 21$	$l = 22$	$l = 23$	$l = 24$	$l = 25$	
Demand (d_l) =	10	20	9	17	15	15	24	5	25	23	16	14	
Variance (u_l) =	35	86	40	50	25	10	60	68	40	73	45	50	

TABLE 4: Parameters of the base case, transportation unit cost from supplier i to plant j .

TC_{ij}^a	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$	$i = 9$	$i = 10$
$j = 1$	164	151	117	132	153	139	158	151	115	182
$j = 2$	140	123	189	162	136	123	132	136	143	167
$j = 3$	137	155	120	119	167	134	133	127	180	132

$$\sum_{m=1}^{|M|} e\tau_j^m = y_j, \quad \forall j = 1, \dots, |J|, \quad (34)$$

$$e\tau_j^m, y_j \in \{0, 1\},$$

$$f\tau_j^m \geq 0, \quad (35)$$

$$\forall j = 1, \dots, |J|, \quad \forall m = 1, \dots, |M|.$$

Equation (31) represents the linear approximation of the WIC at plant j . Equation (32) ensures that the average demand in plant j is considered for the linear approximation. Equation (33) establishes that only a section m must be used in the linear approximation. Equation (34) guarantees that the linear approximation is carried out only in open plants. Finally, (35) describes the domain of the variables. The same analysis is considered to generate the linear approximation of SSC in each plant and WIC and SSC in each warehouse.

An alternative objective function $Z1'$ is defined. $Z1'$ is equivalent to $Z1$ but considering the linear approximation of WIC and SSC introduced before. The new optimization model, with objective function $Z1'$ and all the linearized constraints, is named 4EILP-SS-L.

TABLE 5: Parameters of the base case, transportation unit cost from plant j to warehouse k .

TC_{jk}^b	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$j = 1$	164	140	137	132	162
$j = 2$	151	123	155	153	136
$j = 3$	117	189	120	119	167

4. Results and Discussion

In this section, five instance sizes with 10 cases each one are presented. The code of the instance size indicates the number of suppliers followed by the number of potential plants, followed by the number of potential warehouses, and the number of retailers. For example, instance size 5-3-5-10 indicates 5 suppliers, 3 potential plants, 5 potential warehouses, and 10 retailers.

A commercial license of LINGO 14 was used to solve every case, with the nonlinear and linear models. The nonlinear model corresponds to the formulation described by (7) to (24). The linear model corresponds to the linearization of the objective function and the constraints applied to the original nonlinear model. A service level of 95% was considered for all the cases. Tables 1–6 show the parameters values of the

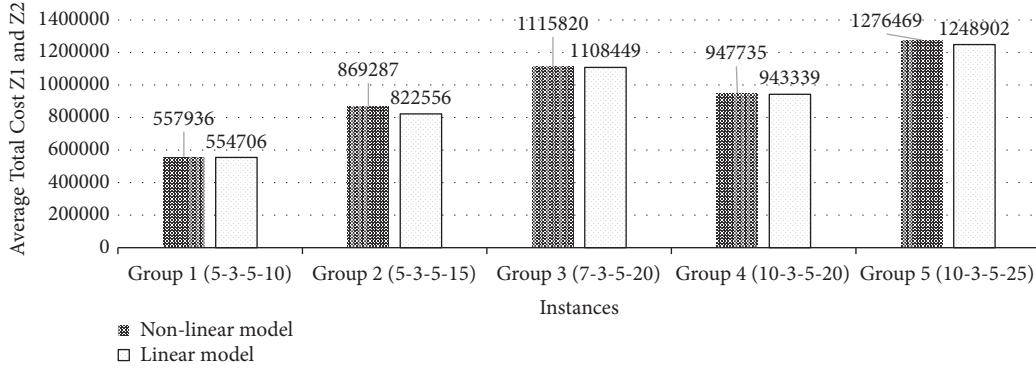


FIGURE 4: Average total cost, comparing the nonlinear versus the linear models.

TABLE 6: Parameters of the base case, transportation unit cost from warehouses k to retailer l .

TC_{kl}^c	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$l = 1$	94	70	78	100	93
$l = 2$	99	86	86	90	89
$l = 3$	86	83	98	80	97
$l = 4$	75	90	77	78	91
$l = 5$	72	75	77	83	87
$l = 6$	80	80	71	99	83
$l = 7$	90	74	98	94	99
$l = 8$	75	75	87	72	98
$l = 9$	74	94	79	80	95
$l = 10$	90	80	84	75	99
$l = 11$	79	95	96	97	79
$l = 12$	84	68	65	97	74
$l = 13$	78	74	95	89	75
$l = 14$	81	73	84	82	89
$l = 15$	98	88	100	98	90
$l = 16$	77	78	98	70	95
$l = 17$	72	85	70	90	66
$l = 18$	98	98	84	86	89
$l = 19$	93	65	83	90	90
$l = 20$	94	71	75	94	90
$l = 21$	89	71	84	85	82
$l = 22$	73	80	76	86	80
$l = 23$	87	84	79	80	70
$l = 24$	81	73	71	70	85
$l = 25$	87	83	89	70	72

base case, which were varied to construct the different cases. The case base has 10 potential suppliers, 3 potential plants, 5 potential warehouses, and 25 potential retailers. For example, to generate one instance of the size 5-3-5-10, 5 out of the 10 available suppliers in the base case were selected randomly, the 3 plants are used, the 5 warehouses are used, and 10 out of the 25 retailers were selected randomly from the base case. To generate one instance of the size 7-3-5-20, 7 out of the 10 available suppliers in the base case were selected randomly, the 3 plants are used, the 5 warehouses are used, and 20

out of the 25 retailers were selected randomly from the base case.

The uncertainty in demand is represented by the variance values shown in Table 3. In all the instances of the linear model, four intervals were used for the linearization, following the results from Diabat and Theodorou [44].

Table 7 shows the computational results of applying the nonlinear and the linear model with a service level of 95% ($ZT_{(0.95)} = 1.64$). Table 7 shows the objective function value Z1, the saved cost from the comparison, the % saving with respect to the solution of the nonlinear model, the run elapsed time for the solution with the linear model, and the cases where the solution was obtained only with the linear model. The comparison denotes the significant benefit (% Saving) that can be obtained by using the linear model instead of the nonlinear model. In this study, the average elapsed run time from the solutions of the linear model was set as a time limit on each group of instances when solving the nonlinear model. In the last column in Table 7, the cases where the nonlinear model does not converge (in the time limit) are shown as well. In order to make a fair comparison between both models, the total cost Z2 was obtained by using the values of the solution obtained by the linear model that optimizes Z1' but evaluated in the original objective function Z1. Let X' be the optimal solution of 4EILP-SS-L, then $Z2 = Z1(X')$.

Table 8 shows the structure of the supply chain according to the results obtained with the models. In many cases, the structures are very similar, but the assignments between customers and warehouses, warehouses and plants, and flows from suppliers produced the differences in costs.

Figure 4 shows the comparison of the average total cost of the nonlinear and the linear models, where a solution was found. In all cases, the solutions of the linear model are cheaper than the solutions from the nonlinear model. The range of the average percent savings (% saving) is from 0.51% to 5.11%. The results of the nonlinear model may be better using a more efficient nonlinear optimizer, especially for the small instances, but it is expected that the linear model will be more efficient for larger instances.

The comparison of the number of solutions obtained between the linear and nonlinear model is presented in Figure 5. The nonlinear model does not converge in all cases with a similar run time than the linear model. The range

TABLE 7: Results of applying the nonlinear model and the linear model considering a service level of 95%.

Case # (Instance size)	Non-Linear Model 4EILP-SS		Linear Model 4EILP-SS-L		Saving	
	Best found objective function value (Z1)	Elapsed run time (sec)	Best found objective function value (Z2)	Elapsed run time (sec)	Saved Cost	% Saving
Case 1 (5-3-5-10)	523,690	5.05	499,861	2.86	23,829	4.55%
Case 2 (5-3-5-10)	451,963	58.68	451,021	3.92	942	0.21%
Case 3 (5-3-5-10)	589,253	490.76	589,246	42.80	7	0.00%
Case 4 (5-3-5-10)	734,372	273.47	733,524	65.91	848	0.12%
Case 5 (5-3-5-10)	514,842	396.13	514,842	3.64	0	0.00%
Case 6 (5-3-5-10)	506,738	13.35	505,464	11.79	1,274	0.25%
Case 7 (5-3-5-10)	511,549	116.7	511,549	8.71	0	0.00%
Case 8 (5-3-5-10)	517,406	98.01	517,406	10.56	0	0.00%
Case 9 (5-3-5-10)	639,303	213.99	639,303	83.68	0	0.00%
Case 10 (5-3-5-10)	584,845	48.24	584,845	15.37	0	0.00%
AVERAGE	557,396	171.44	554,706	24.92		0.51%
Case 11 (5-3-5-15)	745,385	378.24	735,503	191.68	9,882	1.33%
Case 12 (5-3-5-15)	811,106	92.01	727,251	88.86	83,855	10.34%
Case 13 (5-3-5-15)	988,571	419.46	919,670	403.44	68,901	6.97%
Case 14 (5-3-5-15)	974,342	351.94	919,175	343.47	55,167	5.66%
Case 15 (5-3-5-15)	833,043	900.64	832,979	90.82	64	0.01%
Case 16 (5-3-5-15)	845,915	180.74	818,642	160.75	27,273	3.22%
Case 17 (5-3-5-15)	735,539	98.76	735,334	92.62	205	0.03%
Case 18 (5-3-5-15)	917,953	50.82	858,491	50.82	59,462	6.48%
Case 19 (5-3-5-15)	No solution found	-	977,396	191.88	-	Only Linear Solution
Case 20 (5-3-5-15)	971,726	876,959.40	855,959	150.08	115,767	11.91%
AVERAGE	869,287	97,714.66	822,556	174.73		5.11%
Case 21 (7-3-5-20)	No solution found	-	1,051,772	597.58	-	Only Linear Solution
Case 22 (7-3-5-20)	993,942	1,587.54	974,498	461.08	19,444	1.96%
Case 23 (7-3-5-20)	1,274,019	212,939.00	1,269,256	1,636.17	4,763	0.37%
Case 24 (7-3-5-20)	1,285,862	1,636.40	1,268,819	1,454.15	17,043	1.33%
Case 25 (7-3-5-20)	No solution found	-	1,062,143	1,986.87	-	Only Linear Solution
Case 26 (7-3-5-20)	1,051,350	16,621.00	1,051,299	859.15	51	0.00%
Case 27 (7-3-5-20)	976,769	3,850.00	974,422	589.93	2,347	0.24%
Case 28 (7-3-5-20)	1,112,979	28,959.00	1,112,397	611.73	582	0.05%
Case 29 (7-3-5-20)	No solution found	-	1,221,641	1,374.21	-	Only Linear Solution
Case 30 (7-3-5-20)	No solution found	-	1,269,527	693.48	-	Only Linear Solution
AVERAGE	1,115,820	44,265.49	1,108,449	935.37		0.66%
Case 31 (10-3-5-20)	980,256	4,228.34	979,306	2,829.46	950	0.10%
Case 32 (10-3-5-20)	831,262	3,300.00	831,094	1,531.48	168	0.02%
Case 33 (10-3-5-20)	1,106,338	6,568.55	1,106,338	3,234.91	0	0.00%
Case 34 (10-3-5-20)	No solution found	-	1,108,531	2,508.82	-	Only Linear Solution
Case 35 (10-3-5-20)	No solution found	-	1,050,292	1,279.51	-	Only Linear Solution

TABLE 7: Continued.

Case # (Instance size)	Non-Linear Model 4EILP-SS		Linear Model 4EILP-SS-L		Saving	
	Best found objective function value (Z1)	Elapsed run time (sec)	Best found objective function value (Z2)	Elapsed run time (sec)	Saved Cost	% Saving
Case 36 (10-3-5-20)	975,718	1,786.84	975,705	3,503.63	13	0.00%
Case 37 (10-3-5-20)	845,102	3,327.00	824,554	594.98	20,548	2.43%
Case 38 (10-3-5-20)	No solution found	-	976,929	1,151.09	-	Only Linear Solution
Case 39 (10-3-5-20)	No solution found	-	1,109,935	3,214.58	-	Only Linear Solution
Case 40 (10-3-5-20)	No solution found	-	1,109,484	3,532.16	-	Only Linear Solution
AVERAGE	947,735	3,842.15	943,399	2,338.89		0.51%
Case 41 (10-3-5-25)	1,270,846	923.97	1,270,708	982.18	138	0.01%
Case 42 (10-3-5-25)	No solution found	-	1,106,664	1381.60	-	Only Linear Solution
Case 43 (10-3-5-25)	1,281,643	8,552.48	1,278,946	4,756.29	2,697	0.21%
Case 44 (10-3-5-25)	1,280,408	2,266.26	1,280,802	3,530.13	-394	-0.03%
Case 45 (10-3-5-25)	1,417,022	6,783.66	1,266,151	3,295.30	150,871	10.65%
Case 46 (10-3-5-25)	1,140,753	8,476.40	1,130,751	819.56	10,002	0.88%
Case 47 (10-3-5-25)	No solution found	-	1,127,784	1,273.77	-	Only Linear Solution
Case 48 (10-3-5-25)	No solution found	-	1,433,556	7,271.13	-	Only Linear Solution
Case 49 (10-3-5-25)	No solution found	-	1,277,243	3,500.21	-	Only Linear Solution
Case 50 (10-3-5-25)	1,268,140	7,441.08	1,266,054	954.83	2,086	0.16%
AVERAGE	1,276,469	5,740.64	1,248,902	2,389.72		2.38%

of performance of the nonlinear model goes from 50% to 100% among the instances. Using the linear model an optimal solution was obtained in all cases. For these instance sizes, one can be confident in obtaining a good solution with the linear model, which is not the case with the nonlinear model.

5. Conclusions and Final Remarks

This research introduced an inventory-location model, 4EILP-SS, for the network design of a four-echelon supply chain, considering the location of warehouses and plants, transportation cost, inventory costs, and supplier selection. The inventory policy and network design decisions are tactical and strategic decisions that should be analyzed concurrently. This is attained using the aforementioned model. This model can be applied to design complex and long supply chains, like in the automotive industry because of the offshore movement of production facilities from industrialized countries to emerging economies. In many cases, there is infrastructure available in the country, and it is easy to adapt the model to that situation fixing locations for the

available facilities and allowing the model to decide for the new optimum flows and locations of new facilities.

The model proposed becomes nonlinear because of the application of the continuous review inventory policy. This feature added to the computational complexity inherited from the Facility Location Problem makes the problem hard to solve. The mixed integer nonlinear model was reformulated using a piecewise linear function and a reformulation of nonlinear constraints in order to generate a mixed integer linear model. Several instances were solved using the linear and nonlinear models, comparing the results in terms of cost savings, number of solutions obtained, and computational runtime. It was observed that solving the nonlinear model with the commercial software generated local optimal solutions, and in several cases the software either did not converge or resulted in infeasible solutions. The total cost of the original objective function is lower, in most of the cases, with the solution obtained by the linear model than the solution obtained by the nonlinear model.

The main contribution of the paper is two-fold. Firstly, a network optimization model for a particular four level

TABLE 8: Structure of the supply chain applying the nonlinear model and the linear model considering a service level of 95%.

Case # (Instance size)	Non-Linear Model 4EILP-SS		Linear Model 4EILP-SS-L	
	Number of plants selected	Number of warehouses selected	Number of plants selected	Number of warehouses selected
Case 1 (5-3-5-10)	3	4	3	5
Case 2 (5-3-5-10)	3	5	3	5
Case 3 (5-3-5-10)	2	4	2	4
Case 4 (5-3-5-10)	2	3	2	3
Case 5 (5-3-5-10)	3	4	3	5
Case 6 (5-3-5-10)	3	4	3	5
Case 7 (5-3-5-10)	3	4	3	5
Case 8 (5-3-5-10)	3	4	3	5
Case 9 (5-3-5-10)	2	3	2	4
Case 10 (5-3-5-10)	3	4	3	4
Case 11(5-3-5-15)	3	4	3	4
Case 12 (5-3-5-15)	2	4	3	4
Case 13 (5-3-5-15)	2	3	2	3
Case 14 (5-3-5-15)	2	3	2	3
Case 15 (5-3-5-15)	2	4	2	4
Case 16 (5-3-5-15)	2	4	2	4
Case 17 (5-3-5-15)	3	4	3	4
Case 18 (5-3-5-15)	2	3	2	4
Case 19 (5-3-5-15)	No solution found	-	2	3
Case 20 (5-3-5-15)	2	3	2	4
Case 21 (7-3-5-20)	No solution found	-	2	4
Case 22 (7-3-5-20)	3	4	3	4
Case 23 (7-3-5-20)	2	3	2	3
Case 24 (7-3-5-20)	2	3	2	3
Case 25 (7-3-5-20)	No solution found	-	2	4
Case 26 (7-3-5-20)	2	4	2	4
Case 27 (7-3-5-20)	3	4	3	4
Case 28 (7-3-5-20)	2	3	2	4
Case 29 (7-3-5-20)	No solution found	-	2	3
Case 30 (7-3-5-20)	No solution found	-	2	3
Case 31 (10-3-5-20)	2	3	2	4
Case 32 (10-3-5-20)	3	4	3	4
Case 33 (10-3-5-20)	2	3	2	3
Case 34 (10-3-5-20)	No solution found	-	2	3
Case 35 (10-3-5-20)	No solution found	-	2	3
Case 36 (10-3-5-20)	2	3	2	4
Case 37 (10-3-5-20)	3	4	3	4
Case 38 (10-3-5-20)	No solution found	-	2	4
Case 39 (10-3-5-20)	No solution found	-	2	3
Case 40 (10-3-5-20)	No solution found	-	2	3
Case 41 (10-3-5-25)	2	3	2	4
Case 42 (10-3-5-25)	No solution found	-	3	4
Case 43 (10-3-5-25)	2	3	2	4
Case 44 (10-3-5-25)	2	3	2	4
Case 45 (10-3-5-25)	2	3	2	4
Case 46 (10-3-5-25)	2	4	3	4

TABLE 8: Continued.

Case # (Instance size)	Non-Linear Model 4EILP-SS		Linear Model 4EILP-SS-L	
	Number of plants selected	Number of warehouses selected	Number of plants selected	Number of warehouses selected
Case 47 (10-3-5-25)	No solution found	-	3	4
Case 48 (10-3-5-25)	No solution found	-	2	3
Case 49 (10-3-5-25)	No solution found	-	2	4
Case 50 (10-3-5-25)	2	3	2	4

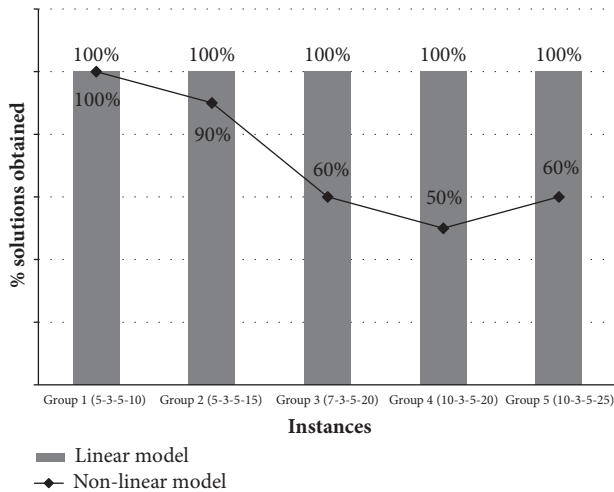


FIGURE 5: Percent of solutions obtained, comparing nonlinear versus linear models.

supply chain, where facility location and inventory management decisions are considered in two stages (plants and warehouses), and transportation or assignment decisions are modeled in three stages of the supply chain, i.e., suppliers-plants, plants-warehouses, and warehouses-customers. Secondly, a linearization approach of the proposed nonlinear model, in order to facilitate its applicability and its effective and efficient implementation, improves both solutions quality and computational times.

The proposed linear model proves to be useful for decision makers interested in analyzing and designing supply chain networks with the structure of a 4EILP-SS. The problem can be analyzed and solved in a short period of time and with significant confidence in the solution quality.

The experiments conducted allowed us to understand the complexity of the problem. For this study, LINGO was used to solve the linear and the nonlinear models, but future work may take advantage of using more efficient commercial software to solve larger instances. As observed, using commercial optimizers may not be the best alternative when it comes to solve very large instances. Therefore, immediate future research should consider the development of other heuristics aimed to tackle large instances (Lagrangian relaxation, Bender decompositions, Cutting planes, and Metaheuristics). Another direction is to extend the pricing approach proposed by Shu et al. [18] to solve a similar problem to the long supply chain proposed in this paper.

In addition, a deeper study may involve the analysis of the relations between parameters like costs, capacities, lead times, and demand variability, to understand their impact in the supply chain network configuration.

Finally, it would be interesting to extend the model to analyze multiobjective situations, more common when designing supply chain networks for real situations.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by Programa para el Mejoramiento del Profesorado PROMEP (Grant ITESUR-001 Charter no. PROMEP/103.5/10/5709) and Universidad Panamericana (Grant no. UP-CI-2018-ING-GDL-06).

References

- [1] M. Mourits and J. J. Evers, "Distribution network design: an integrated planning support framework," *Logistics Information Management*, vol. 9, pp. 45–54, 1995.
- [2] R. Z. Farahani, H. Rashidi Bajgan, B. Fahimnia, and M. Kaviani, "Location-inventory problem in supply chains: A modelling review," *International Journal of Production Research*, vol. 53, no. 12, pp. 3769–3788, 2015.
- [3] OICA, International Organization of Motor Vehicle Manufacturers, Production statistics, 2017, <http://www.oica.net/category/production-statistics/2017-statistics/>.
- [4] H. Hamacher and Z. Drezner, *Facility Location: Applications and Theory*, Springer, Berlin, Germany, 2002.
- [5] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*, Wiley-Interscience, New York, NY, USA, 1st edition, 1995.
- [6] S. Nickel and J. Puerto, *Location Theory: A Unified Approach*, Springer, New York, NY, USA, 2005.
- [7] M. T. Melo, S. Nickel, and F. Saldanha-da-Gama, "Facility location and supply chain management—a review," *European Journal of Operational Research*, vol. 196, no. 2, pp. 401–412, 2009.

- [8] H. A. Eiselt and V. Marianov, *Foundations of Location Analysis*, Springer, New York, NY, USA, 2011.
- [9] H. A. Eiselt and V. Marianov, *Applications of Location Analysis*, Springer, New York, NY, USA, 2015.
- [10] V. Jayaraman, "Transportation, facility location and inventory issues in distribution network design: An investigation," *International Journal of Operations and Production Management*, vol. 18, no. 5, pp. 471–494, 1998.
- [11] L. K. Nozick and M. A. Turnquist, "Integrating inventory impacts into a fixed-charge model for locating distribution centers," *Transportation Research Part E: Logistics and Transportation Review*, vol. 34, no. 3, pp. 173–186, 1998.
- [12] S. J. Erlebacher and R. D. Meller, "The interaction of location and inventory in designing distribution systems," *IIE Transactions*, vol. 32, no. 2, pp. 155–166, 2000.
- [13] L. K. Nozick and M. A. Turnquist, "A two-echelon inventory allocation and distribution center location analysis," *Transportation Research Part E: Logistics and Transportation Review*, vol. 37, no. 6, pp. 425–441, 2001.
- [14] L. K. Nozick and M. A. Turnquist, "Inventory, transportation, service quality and the location of distribution centers," *European Journal of Operational Research*, vol. 129, no. 2, pp. 362–371, 2001.
- [15] M. S. Daskin, C. R. Coullard, and Z. J. M. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, pp. 83–106, 2002.
- [16] Z.-J. M. Shen, C. R. Coullard, and M. S. Daskin, "A joint location-inventory model," *Transportation Science*, vol. 37, no. 1, pp. 40–55, 2003.
- [17] P. A. Miranda and R. A. Garrido, "Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand," *Transportation Research Part E: Logistics and Transportation Review*, vol. 40, no. 3, pp. 183–207, 2004.
- [18] J. Shu, C.-P. Teo, and Z. M. Shen, "Stochastic transportation-inventory network design problem," *Operations Research*, vol. 53, no. 1, pp. 48–60, 2005.
- [19] L. V. Snyder, M. S. Daskin, and C.-P. Teo, "The stochastic location model with risk pooling," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1221–1238, 2007.
- [20] J. Shu, Q. Ma, and S. Li, "Integrated location and two-echelon inventory network design under uncertainty," *Annals of Operations Research*, vol. 181, pp. 233–247, 2010.
- [21] P. A. Miranda and R. A. Garrido, "A simultaneous inventory control and facility location model with stochastic capacity constraints," *Networks and Spatial Economics*, vol. 6, no. 1, pp. 39–53, 2006.
- [22] P. A. Miranda and R. A. Garrido, "Valid inequalities for Lagrangian relaxation in an inventory location problem with stochastic capacity," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 1, pp. 47–65, 2008.
- [23] L. Ozsen, C. R. Coullard, and M. S. Daskin, "Capacitated warehouse location model with risk pooling," *Naval Research Logistics (NRL)*, vol. 55, no. 4, pp. 295–312, 2008.
- [24] P. A. Miranda and G. Cabrera, "Inventory location problem with stochastic capacity constraints under periodic review (R, s, S)," in *Proceedings of the International Conference on Industrial Logistics: Logistics and Sustainability, (ICIL '10)*, pp. 289–296, 2010.
- [25] G. Cabrera, P. A. Miranda, E. Cabrera et al., "Solving a novel inventory location model with stochastic constraints and inventory control policy," *Mathematical Problems in Engineering*, vol. 2013, Article ID 670528, 12 pages, 2013.
- [26] J.-H. Kang and Y.-D. Kim, "Inventory control in a two-level supply chain with risk pooling effect," *International Journal of Production Economics*, vol. 135, no. 1, pp. 116–124, 2012.
- [27] F. Silva and L. Gao, "A joint replenishment inventory-location model," *Networks and Spatial Economics*, vol. 13, no. 1, pp. 107–122, 2013.
- [28] J.-S. Tancrez, J.-C. Lange, and P. Semal, "A location-inventory model for large three-level supply chains," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 2, pp. 485–502, 2012.
- [29] A. Diabat, J.-P. Richard, and C. Codrington, "A Lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design," *Annals of Operations Research*, vol. 203, pp. 55–80, 2013.
- [30] M. Shahabi, S. Akbarinasaji, A. Unnikrishnan, and R. James, "Integrated inventory control and facility location decisions in a multi-echelon supply chain network with hubs," *Networks and Spatial Economics*, vol. 13, pp. 497–514, 2013.
- [31] Z.-J. M. Shen and M. S. Daskin, "Trade-offs between customer service and cost in integrated supply chain design," *Manufacturing and Service Operations Management*, vol. 7, no. 3, pp. 188–207, 2005.
- [32] S. Gaur and A. R. Ravindran, "A bi-criteria model for the inventory aggregation problem under risk pooling," *Computers & Industrial Engineering*, vol. 51, no. 3, pp. 482–501, 2006.
- [33] P. A. Miranda and R. A. Garrido, "Inventory service-level optimization within distribution network design problem," *International Journal of Production Economics*, vol. 122, no. 1, pp. 276–285, 2009.
- [34] H.-Y. Mak and Z. M. Shen, "A two-echelon inventory-location problem with service considerations," *Naval Research Logistics (NRL)*, vol. 56, no. 8, pp. 730–744, 2009.
- [35] P. Escalona, F. Ordóñez, and V. Marianov, "Joint location-inventory problem with differentiated service levels using critical level policy," *Transportation Research Part E: Logistics and Transportation Review*, vol. 83, pp. 141–157, 2015.
- [36] A. Atamtürk, G. Berenguer, and Z.-J. Shen, "A conic integer programming approach to stochastic joint location-inventory problems," *Operations Research*, vol. 60, no. 2, pp. 366–381, 2012.
- [37] O. Kaya and B. Urek, "A mixed integer nonlinear programming model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain," *Computers & Operations Research*, vol. 65, pp. 93–103, 2016.
- [38] M. a. Schuster Puga and J.-S. Tancrez, "A heuristic algorithm for solving large location-inventory problems with demand uncertainty," *European Journal of Operational Research*, vol. 259, no. 2, pp. 413–423, 2017.
- [39] P. Escalona, V. Marianov, F. Ordóñez, and R. Stegmaier, "On the effect of inventory policies on distribution network design with several demand classes," *Transportation Research Part E: Logistics and Transportation Review*, vol. 111, pp. 229–240, 2018.
- [40] F. J. Tapia-Ubeda, P. A. Miranda, and M. Macchi, "A generalized Benders decomposition based algorithm for an inventory location problem with stochastic inventory capacity constraints," *European Journal of Operational Research*, vol. 267, no. 3, pp. 806–817, 2018.

- [41] R. E. Perez Loaiza, E. Olivares-Benitez, P. A. Miranda Gonzalez, A. Guerrero Campanur, and J. L. Martinez Flores, "Supply chain network design with efficiency, location, and inventory policy using a multiobjective evolutionary algorithm," *International Transactions in Operational Research*, vol. 24, no. 1-2, pp. 251–275, 2017.
- [42] F. You and I. E. Grossmann, "Integrated multi-echelon supply chain design with inventories under uncertainty: MINLP models, computational strategies," *AIChE Journal*, vol. 56, no. 2, pp. 419–440, 2010.
- [43] K. Petridis, "Optimal design of multi-echelon supply chain networks under normally distributed demand," *Annals of Operations Research*, vol. 227, no. 1, pp. 63–91, 2015.
- [44] A. Diabat and E. Theodorou, "A location-inventory supply chain problem: Reformulation and piecewise linearization," *Computers & Industrial Engineering*, vol. 90, pp. 381–389, 2015.
- [45] A. Nyberg, I. E. Grossmann, and T. Westerlund, "An efficient reformulation of the multiechelon stochastic inventory system with uncertain demands," *AIChE Journal*, vol. 59, no. 1, pp. 23–28, 2013.

Research Article

Fractional-Derivative Approximation of Relaxation in Complex Systems

Kin M. Li,¹ Mihir Sen,² and Arturo Pacheco-Vega ¹

¹*Department of Mechanical Engineering, California State University-Los Angeles, Los Angeles, CA 90032, USA*

²*Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*

Correspondence should be addressed to Arturo Pacheco-Vega; apacheco@calstatela.edu

Received 27 June 2018; Revised 14 September 2018; Accepted 16 October 2018; Published 8 November 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Kin M. 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.

In this paper, we present a system identification (SI) procedure that enables building linear time-dependent fractional-order differential equation (FDE) models able to accurately describe time-dependent behavior of complex systems. The parameters in the models are the order of the equation, the coefficients in it, and, when necessary, the initial conditions. The Caputo definition of the fractional derivative, and the Mittag-Leffler function, is used to obtain the corresponding solutions. Since the set of parameters for the model and its initial conditions are nonunique, and there are small but significant differences in the predictions from the possible models thus obtained, the SI operation is carried out via global regression of an error-cost function by a simulated annealing optimization algorithm. The SI approach is assessed by considering previously published experimental data from a shell-and-tube heat exchanger and a recently constructed multiroom building test bed. The results show that the proposed model is reliable within the interpolation domain but cannot be used with confidence for predictions outside this region. However, the proposed system identification methodology is robust and can be used to derive accurate and compact models from experimental data. In addition, given a functional form of a fractional-order differential equation model, as new data become available, the SI technique can be used to expand the region of reliability of the resulting model.

1. Introduction

Complex systems are common in mechanical engineering applications, for example, automobiles, washing machines, and thermal power plants. As illustrated schematically in Figure 1, each of these systems is composed of many interconnected subsystems. In principle each subsystem can be studied in isolation and its input and output connected to others. However, for purposes of control we may think of the overall system as being of the single-input single-output (SISO) type, i.e., the interconnections between subsystems are all internal, and there is only a single input $u(t)$ and a single output $y(t)$ of the overall system, where t is time. The control of such systems may either be model based or not [1]. Control procedures that are not based on a mathematical model include techniques such as PID [2], fuzzy logic [3], or other procedures that work on the error signal. For model based control, on the other hand, one should have an approximate mathematical model of the time-dependent

behavior of the system to be able to predict what is going to happen and then control it (approximate because otherwise there would be no need for control). For simplicity we will only consider linear systems. There are two types of inputs that are of interest in complex SISO systems: one is relaxation and the other is periodic. In the former, which is what we will be concerned with here, a step input is applied to the system and the output goes in some dynamic fashion from one constant value to another.

One approach to modeling is from the ground up, i.e., to use first principles to model each component and their interactions to create a so-called white box model. If we think of each subsystem as being governed by a single ordinary differential equation (ODE), then the overall system is governed by a large set of coupled ODEs. If, however, there is a governing partial differential equation (PDE), then that may be considered to be equivalent to an infinite set of ODEs. Modeling is possible only when there is physical and mathematical understanding of the behavior of each

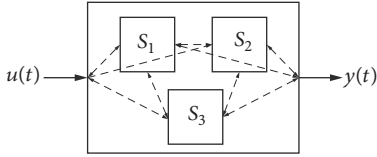


FIGURE 1: Schematic representation of a complex system with subsystems S_1 , S_2 , and S_3 . Dashed lines are interactions. Input and output of the overall system are $u(t)$ and $y(t)$, respectively.

subsystem. In any case, for control of the overall system one ends up working with a large set of ODEs that must be solved in real time. This may be computationally undesirable for control purposes.

In many cases it is advantageous to use existing or continually acquired experimental data to make predictions [4]. This is a black-box approach and its main advantage is that the input-output response of the overall system can be directly obtained, but a disadvantage is that there is no physical understanding of the dynamic behavior compared to the ground-up approach. For example, one can have a procedure based on artificial neural networks which can either be trained beforehand or continuously trained [5]. Alternatively one can fit an analytical model to the data, and this is *system identification* (SI). For control purposes the data-based model should be good enough to approximate reality but also be solvable in real time without intensive computation. The SI process leads to a mathematical model that can make predictions that differ little from reality in the region of interest and is compact enough to be easily solved and used for control purposes. The most common SI procedure is to propose a mathematical model of the relaxation dynamics of the system, usually in the form of differential equations, with certain free constants that are adjusted to best fit measurements. In this manifestation SI is really determination of parameters in the proposed model.

Two new aspects will be explored here in relation to SI.

(a) *Fractional-Order Model*. Usually the proposed mathematical models involve integer-order derivatives, but this lacks the generality provided by derivatives of fractional order [6–8]. In addition, the required mathematical description of the system is directly linked to its complexity, and when the latter increases so does the former. Though any system representation can be achieved using integer-order conservation equations, their solution, in the context of system control, requires extremely large CPU times, and it becomes necessary to develop accurate and compact models that can be used to determine the corresponding behavior in a timely and reliable manner. Since it has been recently demonstrated that fractional-based models are able to describe systems that are complex [9], and a physical understanding of the subsystem processes is not of concern for control of the overall system, the use of suitably defined fractional derivatives is appropriate [10, 11]. There are many different definitions of fractional derivatives available, though there are some common features between them [12]. Initial conditions for

time-dependent systems is an important issue for fractional-order systems: unlike integer-order derivatives where the number of initial conditions corresponds to the order of the differential equation, many of the definitions of fractional derivatives make the equation infinite dimensional so that an infinite number of initial conditions are needed. Here we use the Caputo definition of the fractional derivative since it enables the prescription and application of initial conditions of integer order typically encountered in physical systems, and analytical solutions are often possible.

(b) *Error Minimization by Global Regression*. The mathematical model with the best approximation to experimental data is obtained through a process of error minimization. On the one hand this can be done through an algorithm that searches locally in the neighborhood of an initial guess, a procedure that is effective if there is a single minimum. On the other hand there are procedures that search globally within the permitted range of parameters for one or more minima. One of these techniques is that of genetic algorithms [4] and another is simulated annealing, which is what we will use here [13, 14].

In the present investigation, we propose a methodology of analysis to derive accurate models – based on fractional-order differential equations – that approximate relaxation processes of complex systems. To this end, the paper presents first a brief description of the background information about fractional calculus. Next, the system identification technique, based on global regression of an error-cost function, is introduced in detail. Application to previously published experimental data derived from (1) a heat exchanger and (2) a recently constructed building test facility is later carried out with special emphasis on the accuracy of the model within the region of interest. A vital conclusion is the nonuniqueness in the resulting parameter set that defines both the fractional-order model and the corresponding initial conditions. Although the proposed model is accurate within the interpolation region, it is unreliable outside it.

2. Background on Fractional Calculus

Fractional (noninteger) calculus, which can be thought of as a generalization of the well-known integer calculus, has a long history dating back to Leibniz [9]. Nevertheless, it is not as celebrated by the scientific community as its integer counterpart, mainly due to insufficient enthusiasm about its potential usefulness in applications. Although significant progress in its theoretical basis has been achieved, primarily due to contributions from 18th- and 19th-century mathematicians (e.g., Laplace, Fourier, Abel, and Liouville), it is only in the last two decades that the subject has significantly broadened its applications in physics and engineering. This is reflected by the increased number of publications (i.e., books [9, 15–17], review monographs [10], and scientific articles) devoted to the topic in a variety of fields, like anomalous diffusion [18], semi-infinite tree networks of mechanical, electrical, and hydrodynamic equipment [19, 20], and viscoelastic systems [21], among others. In the context of modeling and control of thermal devices and phenomena, studies, such as those

of Aoki et al. [22], Pineda et al. [23], Gabano and Pointot [24], and Caponetto et al. [25], demonstrate that fractional derivatives provide good approximations for describing the dynamic behavior of heat transfer processes.

In principle, the field of fractional calculus entails the generalization of the concept of a derivative beyond that of the common integer order. It is concerned with the meaning of the “in-between” derivatives. For instance, the 5/4-derivative of a function $f(t)$,

$$f^{(5/4)}(t) = \frac{d^{5/4}f}{dt^{5/4}} \quad (1)$$

can be thought of as a derivative in-between the first and second derivatives. Though it seems straightforward to think of a derivative whose order is between two consecutive integers, the definition of a fractional derivative is not unique and various definitions are possible as long as it satisfies a required set of mathematical rules [12]. Two of the most common definitions are those of Riemann-Liouville [16] and Caputo [26], the latter being the type of derivative used in the present study.

Let us start with the Cauchy formula for repeated integrals of integer order m , for the function $f(t)$, which is defined as

$$J^m f(t) = \frac{1}{(m-1)!} \int_0^t (t-z)^{m-1} f(z) dz; \quad m \in \mathbb{Z}, \quad (2)$$

where J^m is the integral operator. By defining the Gamma function as

$$\Gamma(\xi) = \int_0^\infty e^{-t'} t'^{\xi-1} dt', \quad (3)$$

the term $(m-1)!$ in (2) can be replaced with $\Gamma(\alpha)$, to obtain

$$J^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-z)^{\alpha-1} f(z) dz, \quad (4)$$

where $\alpha > 0$ is an arbitrary positive real number and z is a dummy variable of integration. Note that $J^\alpha f(t) = D^{-\alpha} f(t)$; thus, for a value of α such that $m-1 < \alpha \leq m$

$$D_{RL}^\alpha f(t) = D^m J^{m-\alpha} f(t), \quad (5)$$

which establishes the Riemann-Liouville fractional derivative of order α , as

$$D_{RL}^\alpha f(t) = \frac{d^m}{dt^m} \left[\frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{f(z)}{(t-z)^{\alpha+1-m}} dz \right]. \quad (6)$$

The Caputo definition of the fractional derivative of order α , on the other hand, is defined as [26]

$$D_C^\alpha f(t) = J^{m-\alpha} D^m f(t), \quad (7)$$

so that $D_C^\alpha f(t)$ is explicitly given as

$$D_C^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{f^{(m)}(z) dz}{(t-z)^{\alpha+1-m}}. \quad (8)$$

In (6) and (8), the subscripts *RL* and *C*, respectively, denote Riemann-Liouville and Caputo fractional derivatives.

Regardless of the definition used, the Laplace transform is a powerful technique to solve equations involving fractional derivatives. For instance, the Riemann-Liouville definition in the Laplace domain is

$$\mathcal{L}(D_{RL}^\alpha f(t)) = s^\alpha F(s) - \sum_{k=0}^{m-1} s^k D^{\alpha-k-1} f(0); \quad (9)$$

$$m-1 < \alpha \leq m,$$

where it can be seen that the transformation requires the following initial conditions: $D^{\alpha-k-1} f(0)$ for all $0 \leq k \leq m-1$. It is important to note that, since α does not need to be an integer number, the terms $D^{\alpha-k-1} f(0)$ are all of fractional order. This poses a problem for applications in physical systems, where the initial conditions are typically given in terms of integer orders. On the other hand, the Laplace transform for the Caputo definition is

$$\mathcal{L}(D_C^\alpha f(t)) = s^\alpha F(s) - \sum_{k=0}^{m-1} s^{\alpha-k-1} D^k f(0); \quad (10)$$

$$m-1 < \alpha \leq m,$$

where now the initial conditions necessary to evaluate (10) are prescribed as integer order, $D^k f(0)$, rather than fractional-order derivatives. Therefore, in the subsequent sections of the paper, we use the Caputo definition of a fractional-order derivative.

3. System Identification

The first step in the SI process is to propose the model and then to determine the parameters therein that will approximate – as closely as possible – its output to the target data. For our purposes, the proposed model is the linear fractional differential equation, given as

$$c_1 \frac{d^q y(t)}{dt^q} + c_2 y(t) = u(t), \quad (11)$$

where c_1 and c_2 are constants, q is the order (noninteger, $m-1 < q \leq m$, with $m \in \mathbb{Z}$) of the equation, $u(t)$ is its input, $y(t)$ is the corresponding output, and t is a time-like independent variable. Since the interest is on the conditions of operation that would take the system from one state to another, i.e., relaxation processes, taking $u(t)$ to be the Heaviside function, the Laplace transform of (11) is

$$Y(s) = \frac{\sum_{k=0}^m s^{q-k-1} y^{(k)}(0)}{s^q + c_2/c_1} + \frac{1}{c_1} \frac{s^{-1}}{s^q + c_2/c_1}, \quad (12)$$

which can then be solved if the initial conditions for $y(t)$, and its corresponding derivatives, are provided. The inverse Laplace transform of the above equation leads to the following analytical form for $y(t)$

$$y(t) = \sum_{k=1}^m P_k t^{k-1} E_{q,k} \left(-\frac{c_2}{c_1} t^q \right) + \frac{1}{c_1} t^q E_{q,q+1} \left(-\frac{c_2}{c_1} t^q \right), \quad (13)$$

where E is the typical Mittag-Leffler function [17]. Note that the output from (13) is completely defined if the constants P_k are known. This implies that one must provide m derivatives at $t = 0$ ($P_k = y^{(k-1)}(0)$, $k = 1, 2, \dots, m$); i.e., for $0 \leq q \leq 1$, $y(0)$, and for $1 < q \leq 2$, $y(0)$, and $y'(0)$ are necessary to determine the solution.

To find the unknowns q, c_1, c_2, P_k , for $k = 1, 2, \dots, m$, in (11) and (13), from known data (e.g., N values y_i , $i = 1, \dots, N$), a least-squares method is used to minimize the difference between the sample- and predicted-values of $y(t)$. This is equivalent to minimizing an Euclidean norm, i.e., the variance of the error, given by

$$S_y = \frac{1}{N} \sum_{i=1}^N (y_i^p - y_i^e)^2, \quad (14)$$

where N is the number of data points, usually from experimental measurements; $y_i^p(t)$, for $i = 1, \dots, N$, are values predicted by the model and $y_i^e(t)$, for $i = 1, \dots, N$, refer to the experiments. It is to be noted that $S_y(\mathbf{C})$, with $\mathbf{C} = (q, c_1, c_2, P_k : k = 1, \dots, m)$ being the vector of unknown parameters, is a smooth manifold in a $(k + 3)$ -dimensional space. The goal is now to search for the values of the parameter set \mathbf{C} , such that $S_y(\mathbf{C})$ is a minimum. This process can be carried out either using local – such as gradient-based – methods or global optimization techniques like evolutionary or deterministic algorithms, each with advantages and drawbacks [27–31]. However, as we will show in a later section, under certain cases (14) has multiple local minima, stemming from its nonlinear nature with respect to the arguments, thus leading us to seek the global minimum which will provide the optimal values of the parameter set \mathbf{C} required in the model (11), and the corresponding initial conditions. This optimization procedure is carried out here by the *simulated annealing* (SA) technique.

Description. The SA technique is inspired by the molecular calculation of the cooling of a physical system in which random agitation is used to avoid entrapment in local extrema [29]. A starting point in the space of unknowns, \mathbf{C}_0 , is randomly selected, and a cycle of random moves along each coordinate direction is then performed. The new point \mathbf{C} is accepted if it gives a better value of S_y . If it is worse, it is accepted only with a certain probability, $\exp(-\Delta S_y/T^*)$, where ΔS_y is the change in value of S_y , and T^* is a dynamic parameter that is analogous to the temperature of a system being cooled [32]. The process is repeated with decreasing T^* , and step size $\Delta \mathbf{C}$ until convergence within a certain tolerance is reached. The procedure described above [33] has been successfully implemented and used in the context of heat transfer correlations by Pacheco-Vega et al. [34], and it is the one followed here.

It is to be noted that other global optimization algorithms can also be used in the search for the set of parameters \mathbf{C} . These include the popular genetic algorithm (GA) [28] and the interval method (IM) [35], among several alternatives, which have been successfully used in a variety of applications. Though each technique has advantages over the others (particularly in terms of CPU time necessary to find

a solution), the GA will only find the region – with a high degree of certainty – where the global optimum is located. From this perspective, only IMs mathematically guarantee that the global optimum has been found, while the SA is only probabilistically – but not deterministically – guaranteed to find such an optimum solution.

SI using the fractional model in (11), herein referred to as FOSI, is applied to experimental data from two complex thermal systems: (1) shell-and-tube heat exchanger data analyzed by Mayes [36] and (2) a multiroom building test facility. Preliminary analysis on the applicability of the FOSI methodology to a set of analytical problems and to heat exchanger data has been recently reported in Li et al. [37]. In what follows, the search domain for q , in (11), is restricted to $q \in [0, m]$ with $m = 2$.

4. Shell-and-Tube Heat Exchanger Data

4.1. Description. Details about the experimental setup in the thermal systems laboratory at the University of Notre Dame, and the corresponding data obtained with it, have been reported in Mayes [36]; here we provide only a brief description of the problem at hand. The heat exchanger, a schematic of which is shown in Figure 2, corresponds to the common shell-and-tube configuration. The figure illustrates flow directions of the cold and hot fluids, \dot{m}_c and \dot{m}_h , respectively, along with their corresponding terminal inlet, T_c^{in} and T_h^{in} , and outlet, T_c^{out} and T_h^{out} , bulk temperatures. The reported tests [36] were carried out under time-dependent conditions until thermal equilibrium in the device was achieved. Subsequently, while maintaining constant values of T_c^{in} , T_h^{in} , and \dot{m}_h , a step change in \dot{m}_c , i.e., $\Delta \dot{m}_c$, was applied. A data acquisition system (DAQ) and a personal computer (PC) were used to record and store the time-dependent values of the inlet and outlet temperatures of the two fluids.

For purposes of analysis, we take the approach of [36] and work with normalized variables. The dimensionless outlet temperature of the hot fluid θ and time variable τ are defined as

$$\theta(t) = \frac{T_h^{out}(t) - T_h^{out}(0)}{T_h^{out}(\infty) - T_h^{out}(0)}; \quad \tau = \frac{t}{\tau_r}, \quad (15)$$

where $T_h^{out}(0)$ and $T_h^{out}(\infty)$ are, respectively, the values of hot-fluid outlet temperatures at the initial and steady states; τ_r is the rise time, defined as the time required for $T_h^{out}(t)$ to reach 85% of $T_h^{out}(\infty)$. Thus, $\theta(\tau) \in [0, 1]$, with $\tau \in [0, \infty)$. The experiments showing the temporal evolution of $\theta(\tau)$ are illustrated graphically in Figure 3(a), where it is clear that some type of relaxation process from one state (with θ having a small positive value) to another (where θ is close to 0.8) took place. Since the goal here is to find the parameters in the fractional model of the form given by (11) which fit the experimental data the best, for clarity we define $\theta(\tau) \equiv y(t)$ and $\tau \equiv t$, so that the fractional-order differential equation becomes

$$c_1 \frac{d^q \theta(\tau)}{d\tau^q} + c_2 \theta(\tau) = u(\tau), \quad (16)$$

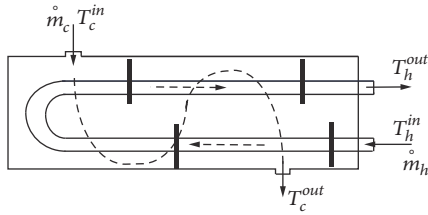


FIGURE 2: Schematic of a shell-and-tube heat exchanger.

where c_1 and c_2 are constants, q is the order in the search domain $q \in [0, m]$, $m = 2$, and $u(t)$ is the input resulting from the step change in the mass flow rate of the cold fluid $\Delta \dot{m}_c$. Once the model has been defined (i.e., Eq. (16)), the FOSI technique is applied to the normalized data.

In this case, the variance of the error is given as

$$S_\theta = \frac{1}{N} \sum_{i=1}^N (\theta_i^p - \theta_i^e)^2, \quad (17)$$

where p and e correspond to predictions and experiments, respectively, and N is the number of data sets. The set of unknown parameters that will be considered in the forthcoming sections will be either $\mathbf{C} = (q, c_1, c_2, P_2 = \theta'(0))$ or $\mathbf{C} = (q, c_1, c_2, P_1 = \theta(0), P_2 = \theta'(0))$. Notice that, when seeking the optimum values in the parameter set, either one $P_2 = \theta'(0)$ or two initial conditions $P_1 = \theta(0)$ and $P_2 = \theta'(0)$ are included; i.e., we have relaxed the restriction regarding the number of unknown parameters and now included either one or two initial conditions in the search. The inclusion of the initial conditions in the parameter search brings the advantage of finding the most appropriate values for them directly from the operating conditions of the system and, therefore, provides a better approximation from the model thus obtained. However, as it will be shown in the next section, this generates multiplicity of solutions of the parameter set, some of which may not be physically feasible.

4.2. Initial Conditions and Nonuniqueness. Initial conditions provide the set of values of the system at the point of departure towards a new set of states. Mathematically, these conditions can be easily set up so that the model can be solved. Experimentally, however, this may not be the case since it is difficult to measure the value of the function (e.g., temperature) at a specific time (e.g., $t = 0$), and even more difficult to establish its rate of change, i.e., the derivative. Sensitivity of the sensors, and even where they are placed, plays a crucial role and increases the uncertainty of the corresponding measurement.

As an example of this situation, let us focus on the experimental measurements of the shell-and-tube heat exchanger described before. The system was analyzed in [36] for dynamic conditions of the outlet temperatures after some thermal equilibrium in the device was achieved. Figure 3(a) shows the experimental data (digitized from [36]) corresponding to the evolution of the dimensionless temperature $\theta(\tau)$. From the figure it can be observed that the general trend in the values of the experimental data near $\tau = 0$

TABLE 1: Model parameters from heat exchanger data.

Model	q	c_1	c_2	$P_2 = \theta'(0)$	$S_\theta \times 10^4$
A	1	-2.4903	0.8055	–	20
B	1.6708	0.3313	1.8067	-0.1093	6.94

($\tau \in [0, 0.01]$) is to decrease in magnitude before showing a sustained relaxation-type increase (with some variation about the mean values) to about $\theta = 0.8$ at $\tau = 1$. This seems to indicate that, unless sudden changes in temperature occurred in the laboratory, the heat-exchange device might not have actually reached the expected thermal equilibrium, and it raises the question about the actual level of uncertainty that may be present while attempting to accurately establish the initial conditions for complex systems since the values $\theta(0) = 0$ and $\theta'(0) = 0$, in this case, are not feasible.

An example is the case of a sensor measuring temperature that either is not appropriately calibrated or that has a slow response time-constant, then the initial states for the temperature or its rate of change will not be correctly established. In other cases, the system may be so complex that establishing actual initial conditions is virtually impossible. A car comprising a large number of subsystems may have initial states for each of those components that are different than those of the overall system, and deciding which initial state is the most appropriate (and thus, the one that should be used) may be extremely difficult. A similar situation occurs in a fluid, where initial conditions must be specified at all locations. From this perspective, the inclusion of the initial conditions in the unknown parameter set may be advantageous since the optimal values of these states can be directly computed from the data via SI. On the other hand, mathematically, by increasing the number of adjustable parameters it may be possible to obtain a better approximation from the model. This is what we have done within the context of the two thermal systems considered here.

In what follows, we apply the global-regression-based FOSI to the heat exchanger experimental data to obtain the parameters in (16) and – when necessary – the corresponding initial conditions that minimize the variance of the error [Eq. (17)]. The search for the parameters in $\mathbf{C} = (q, c_1, c_2, P_2 = \theta'(0))$ is constrained to the following ranges: $q \in [0, 2]$, $\{c_1, c_2\} \in [-10, 10]$, and $\theta'(0) \in [-10, 10]$. The initial condition is kept as $P_1 = \theta(0) = 0$, as established in [36]. For purposes of comparison, a gradient-based local optimization algorithm (LOA) is also applied to seek the best-possible parameters in \mathbf{C} . In applying the LOA technique, the search is constrained only for the order $q \in [0, 2]$, while the domain for all other parameters remains unconstrained.

The results from the two fractional-order models are shown in Figure 3(a) and Table 1. From the figure it can be seen that the model obtained by global regression (model B) provides a much better approximation to the experimental data than that from the local-regression procedure (model A). Although both models are somewhat closer in the middle regions, i.e., $t \in [0.2, 0.8]$, their predictions deviate significantly at the two ends of the relaxation process, with the SA-based model (also referred to as a four-parameter model

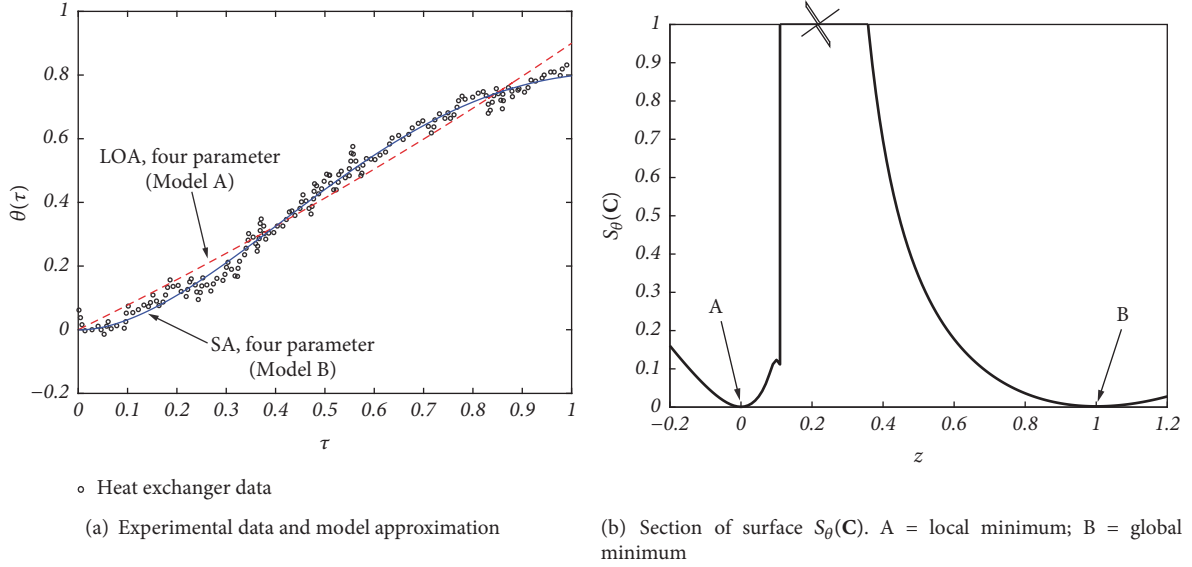


FIGURE 3: Multiplicity of solutions in the parameter set $\mathbf{C} = (q, c_1, c_2, P_2 = \theta'(0))$.

TABLE 2: Model parameters from shifted heat exchanger data.

	q	c_1	c_2	$P_1 = \theta(0)$	$P_2 = \theta'(0)$	$S_\theta \times 10^4$
θ_+	1.5061	0.2717	1.3858	0.005	-0.8368	6.6471
θ	1.5061	0.2621	1.3373	0.0313	-0.8366	6.6470
θ_-	1.5061	0.2532	1.2919	0.0576	-0.8364	6.6471

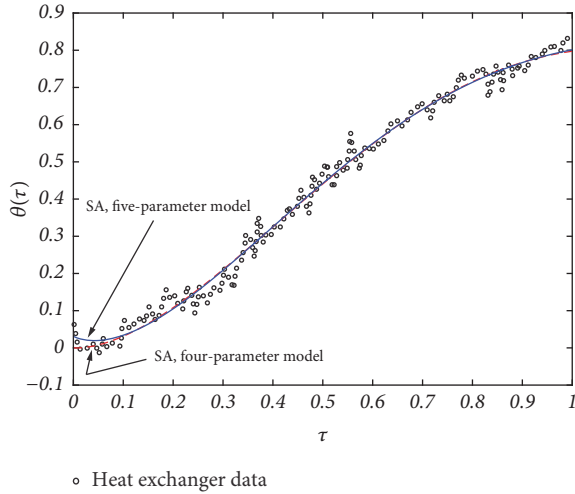


FIGURE 4: Comparison of prediction of $\theta(\tau)$ between four- and five-parameter-set models. (—) five-parameter model, RMS error is 6.647×10^{-4} ; (---) four-parameter model, RMS error is 6.939×10^{-4} .

in Figure 3(a) since the vector \mathbf{C} comprised four parameters) being much closer to the original data. On the other hand, from the table it can be seen that the two fractional models have very different values of the parameters q , c_1 , and c_2 , and particularly the fact that $\theta'(0) = P_2$ only appears in the fractional-order model obtained via global regression (model B) since its value of q lies between $m = 1$ and $m = 2$. This

outcome is indicative of the nonuniqueness of the resulting fractional differential equation model, along with the initial condition considered in the parameter set \mathbf{C} .

Figure 3(b) shows a section of the S_θ hypersurface that passes through the two minima A and B. For clarity, a location coordinate z is defined such that $\mathbf{C} = \mathbf{C}_A(1 - z) + \mathbf{C}_B z$, where $\mathbf{C}_A = (q, c_1, c_2, P_2 = \theta'(0))_A$ and $\mathbf{C}_B = (q, c_1, c_2, P_2 = \theta'(0))_B$, corresponding to the two minima. The value of the cost function S_θ from the LOA-based model (model A) is 3 times higher than that of the SA model (model B in Table 1). The above result leads us to ask which model is better, and a possible answer may be the model equation that provides the minimum error, i.e., the global minimum. This is accomplished in the present study through the implementation of SA in the SI methodology.

By using the heat exchanger data described before, and relaxing the restriction on the number of parameters in \mathbf{C} to now include the additional initial condition $\theta(0) = P_1$, $\mathbf{C} = (q, c_1, c_2, P_1 = \theta(0), P_2 = \theta'(0))$, i.e., a five-parameter fractional-order model, then it is clear as illustrated in Figure 4 that not only the overall solution but also the value of $\theta(0)$ is much closer to the target data than that of the four-parameter model. It is to be noted that, although not shown explicitly, there is also a multiplicity of solutions in $S_\theta(\mathbf{C})$, the global optimum being found by the SA algorithm. The multiplicity in the parameter set seems to be due to the fact that the initial condition(s) is(are) included in the search rather than the model itself. Finally, Table 2 shows, quantitatively, the improved accuracy of the five-parameter

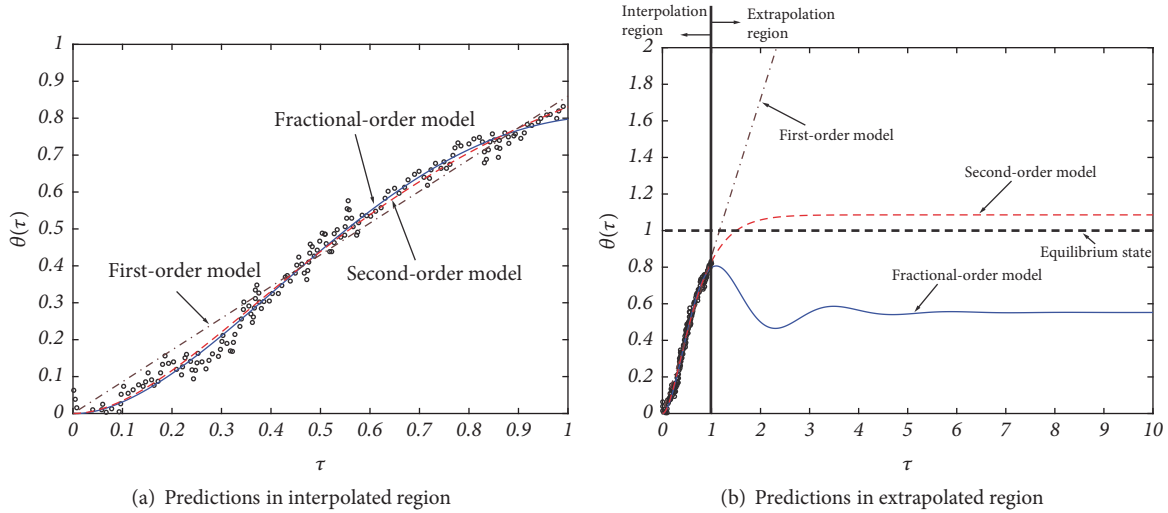


FIGURE 5: Comparison in predictions from fractional- and integer-order fitted models.

fractional model with a value of $S_\theta = 6.647 \times 10^{-4}$, which is an improvement of 4.5% with respect to that of the model based on four parameters.

4.3. Reliability of the Model. The use of fractional-order approximation models stems from the fact that as the complexity of a system increases, so does its required mathematical representation. Though this can be achieved starting from conservation equations, for purposes of control it is important that the plant model is compact, efficient, and sufficiently accurate. For instance, a model based on the Navier-Stokes equations may be more accurate than an approximate model, but would be computationally intensive for control (for design and prediction, on the other hand, the mathematical model needs to be accurate). An approximate model, however, may also serve the purpose of bringing the output close to the desired state from where some other form of nonmodel control (e.g., PID) can take over. From this perspective, the fractional-order model given in (11) and (16) can be regarded simply as an approximation to the governing equations (PDEs), and certainly it is not a model based on first principles (i.e., derived from the laws of physics for a specific problem), but is developed based on a curve-fitting process from a set of experimental data in a specific domain. Thus, it follows that the resulting mathematical model can only be expected to perform well in the domain of the data from which it was derived (i.e., interpolation). In other words, the model may not work – and it does *not* have to work – beyond the domain in which the data apply (i.e., within the cloud of training data). Making predictions outside the domain of the data from which the model was derived would correspond to extrapolation, and therefore unreliable.

This data-driven model interpolation/extrapolation issue is particularly problematic for the case of complex systems, an example of which is heat exchangers, and the phenomena associated with them. In these thermal systems, complexity arises from geometrical configurations, the large number of parameters involved in their operation, and the nonlinear

nature of the system. In the latter case, for example, there is nonlinearity due to variation of properties (e.g., density, viscosity, and thermal conductivity), with temperature. Nonlinearity in the phenomena then generates the possibility of bifurcations, including instability and transition to turbulence. Finally, the conservation equations themselves, which provide the most accurate description of the system, are nonlinear (e.g., in the advective terms). It is apparent that nonlinearities associated with the system may be a main reason for constraining predictions from data-driven models to within the interpolation region only.

The issue of interpolation vs. extrapolation is prevalent in models derived from experimental data. For instance, techniques such as artificial neural networks and correlation equations, among others, are unreliable for extrapolation; yet, they are extremely useful for predictions *within* the region of training data of the system behavior at operating points different from those used to derive the model. This is illustrated in Figure 5 in the context of the current investigation, which shows the results from the fractional-order model given in (16) and an ordinary second-order linear oscillator given by

$$c_1 \frac{d^2\theta(\tau)}{d\tau^2} + c_3 \frac{d\theta(\tau)}{d\tau} + c_2\theta(\tau) = u(\tau), \quad (18)$$

with $y \equiv \theta$, zero initial conditions, and fitted constants c_1 , c_2 , and c_3 obtained from the same data. The figure shows the evolution of $\theta(\tau)$ for the ranges $\tau \in [0, 1]$ and $\tau \in [1, 10]$. For completeness, the results from a first-order model are also included. From Figure 5(a), the results show that, in the range $\tau \in [0, 1]$, both models predict the behavior of the system very well, with the fractional-order model being more accurate ($S_\theta = 6.94 \times 10^{-4}$ and $S_\theta = 7.72 \times 10^{-4}$, respectively), and both show (see Figure 5(b)) some discrepancy with respect to the expected behavior in the range $\tau \in [1, 10]$, with the second-order model being qualitatively more accurate than the fractional-order. However, in this range the actual values

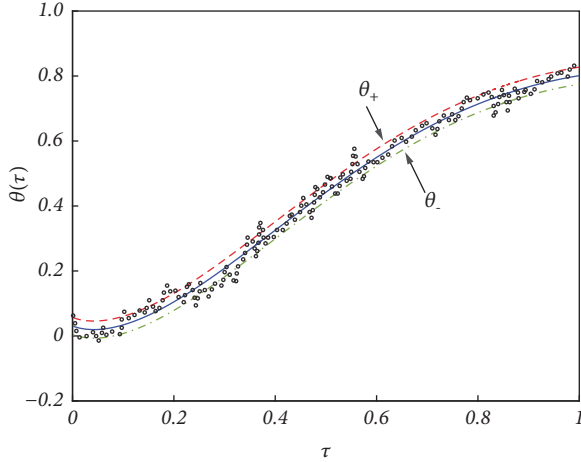


FIGURE 6: Range of applicability of five-parameter set $\mathbf{C} = (q, c_1, c_2, P_1 = \theta(0), P_2 = \theta'(0))$ model. $\theta_+ = \theta + \epsilon_{rms}$; $\theta_- = \theta - \epsilon_{rms}$.

of the expected behavior from both models are incorrect, indicating a lack of reliability.

The issue of reliability of mathematical models is an active research topic [38], and several studies have attempted to establish the region of reliability by defining it either as a hyperbox or by using the convex hull of the data. Others have concluded that the number and distribution of the experimental data are important factors in the reliability of empirical models [39, 40]. In fact, in the context of modeling with artificial neural networks, Pacheco-Vega et al. [41] discovered that even inside the cloud of data – and its corresponding convex hull – extrapolation could exist if the number and distribution of data points in a high-dimensional parameter space were limited in number and location, and proposed a methodology to establish the upper limits in the prediction error from such models.

4.4. Range of Applicability. In practice, experimental uncertainty may be present due to nonideal sensor measurement, thus it may be necessary to establish the applicability of the resulting model for the system under analysis. In the present case, we use the experimental data, along with a definition based on the root-mean-square (RMS) error, to establish the range of applicability for the five-parameter fractional-order model, i.e., (16), with fitted $\theta(0)$ and $\theta'(0)$.

The first step is to define the RMS error as

$$\epsilon_{rms} = \left[\frac{1}{N} \sum_{i=1}^N (\theta_i^p - \theta_i^e)^2 \right]^{1/2}. \quad (19)$$

This error provides the baseline value for the range of applicability from the original predictions. The next step is to shift the data up and down by a value ϵ_{rms} , as

$$\{\theta_+, \theta_-\} = \theta \pm \epsilon_{rms}. \quad (20)$$

The global-regression-based SI is then applied to the shifted data and the optimal parameters obtained. The results are given in Table 2 and Figure 6. Table 2 presents a quantitative

account of the results showing the range of applicability of the fractional differential equation model, where the order of the equation remains essentially unaltered, while the other θ_+ - and θ_- -based shifted parameters provide the values corresponding to the upper and lower bounds. Figure 6 supplies a graphical account of the same results, where it is clear that the model is able to accurately predict the behavior of the system.

5. Multiroom Building Data

The second test deals with a more complex multiroom building system. A subscaled two-floor building facility, designed specifically to be a scaled model of an actual building for energy-related studies, is located in the thermofluids laboratory at the California State University, Los Angeles, and is used to conduct experiments for thermal analysis and control [42]. A schematic of the experimental facility, with overall dimensions of 1.2 m \times 0.92 m \times 1.1 m, and its photographic depiction are shown in Figure 7. The test bed has a two-floor configuration with four rooms in each floor. Its structure is made of wood with interior walls covered with drywall and insulation (additional details are in [42]). Average temperatures of air in the rooms are measured by type-K thermocouples, while incandescent light bulbs serve as internal heat sources. Supply and return vents are installed in each room, and an external cooling unit is used to provide cold air through a set of ducts, each connected to the corresponding supply vent. Flow rates of air delivered into each room can be modified by a set of dampers (valves) through Arduino microcontrollers. Time-dependent information of the room temperatures, T , is collected in two DAQ boards and stored in a PC for further analysis. LabVIEW serves to interface the controller and the experimental system. The facility has been recently studied by Baltazar et al. [43] in the context of thermal control with fuzzy logic.

The average air temperature measurements for the eight rooms of the building are illustrated in Figure 8 which shows that each room is initially heated to a temperature above 27°C and then cooled down for 30 min [43]. From the figure it can be seen that all temperature curves depict three features: (1) the cooling rate in each room is different due to uneven delivery of airflow by the dampers; (2) there is a nearly exponential decay in the average temperatures, which is expected due to the convection process that takes place in each room; and (3) the decay in the temperatures is altered by two peaks, at $t = 17$ min and $t = 25$ min, resulting from the influence that the laboratory temperature has on the operation of the cooling unit.

For the analysis that follows, room 1 is selected because of its smooth transition from a maximum temperature, T_{max} , to a temperature considered as minimum, T_{min} , just before the peak occurs at t_{max} . As before, it is useful to normalize its mean temperature $T(t)$ and the time variable t , as

$$\theta(t) = \frac{T(t) - T_{min}}{T_{max} - T_{min}}; \quad \tau = \frac{t}{t_{max}}, \quad (21)$$

where θ is the dimensionless average temperature for air, τ the dimensionless time, $T_{max} = 28^\circ\text{C}$, $T_{min} = 18^\circ\text{C}$, and

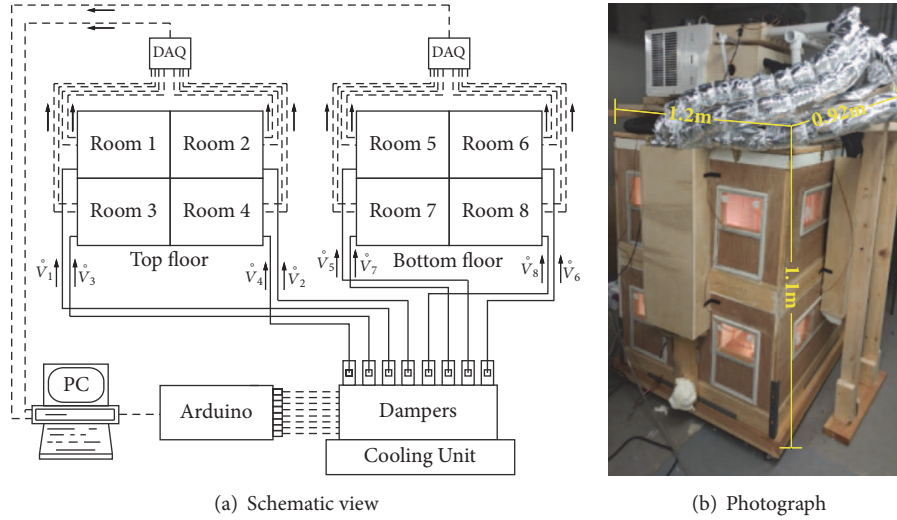


FIGURE 7: Subscaled building test facility.

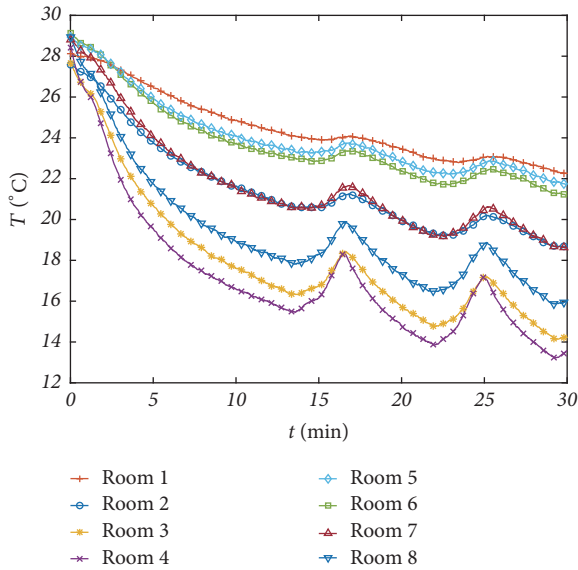


FIGURE 8: Time evolution of average room temperatures in multi-room building device.

$t_{max} = 17$ min. With the aforementioned definitions, the linear fractional-order model is again defined as

$$c_1 \frac{d^q \theta(\tau)}{d\tau^q} + c_2 \theta(\tau) = u(\tau), \quad (22)$$

where $u(\tau)$ results from the sudden change in the delivery of cooling air flow from zero to a maximum.

The FOSI methodology is now applied to the experimental temperature measurements of room 1 of the test facility. Again, the variance of the error is $S_\theta = [\sum_{i=1}^N (\theta_i^p - \theta_i^e)^2] / N$, the parameter set is $\mathbf{C} = (q, c_1, c_2, P_1 = \theta(0), P_2 = \theta'(0))$, and the parameter search is constrained to the following ranges: $q \in [0, 2]$, $\{c_1, c_2\} \in [-10, 10]$, $\theta(0) \in [-10, 10]$, and $\theta'(0) \in [-10, 10]$. Using the same conditions, LOA is also used for

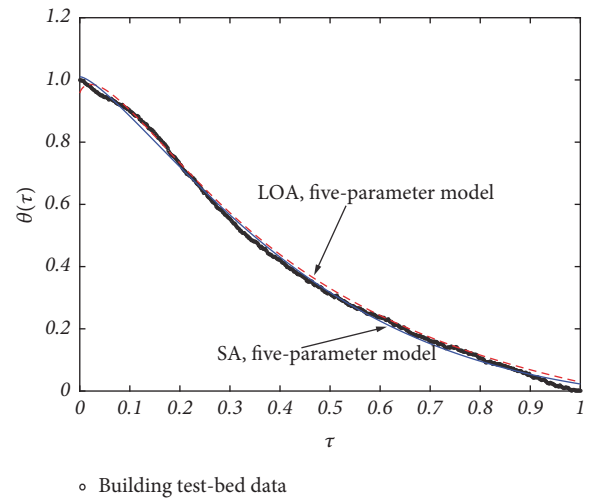


FIGURE 9: Prediction of time evolution of average room temperature for room 1 from fractional-order models.

purposes of comparison. The results from the two methods are shown in Figure 9 and Table 3, where it is clear that, once again, multiplicity of solutions for the parameter set \mathbf{C} , and $S_\theta(\mathbf{C})$, occurs. From the figure it can be observed that the fractional-order model obtained by the FOSI, SA technique, accurately predicts the dynamic behavior of the temperature dynamics in room 1 for the entire time range. It is particularly remarkable that the conditions of θ and θ' at $\tau = 0$ are very close to those of the data. On the other hand, the results from the LOA-based model are close to the measurements in the mid-section, but deviate significantly at the extremes, particularly near the initial times.

These results are confirmed quantitatively in Table 3, where it is noticed that the fractional-order model obtained via global regression outperforms the corresponding LOA model by close to 40% improvement in the accuracy of the predictions. Moreover, the values of the parameters

TABLE 3: Model parameters for the relaxation process in room 1 of a building test facility.

Model	q	c_1	c_2	$P_1 = \theta(0)$	$P_2 = \theta'(0)$	$S_\theta \times 10^4$
SA	1.2319	3.4428	10	1.0223	-0.1043	2.04
LOA	1.0266	-0.0213	-0.1052	0.9577	47.5124	2.86

obtained from global regression seem to be consistent with the experimental data, particularly those of the rate of change in temperature θ' , at times close to zero, whereas those computed by the LOA technique can be associated with physically unfeasible conditions, as mentioned in the previous section.

Finally, it is important to note that, though commonly done with integer-order calculus, a physical interpretation of the fractional derivative in the modeling of the two thermal systems analyzed here is not necessary since the corresponding fractional-order model – in each case – is merely a stand-in for the true mathematical model. Some efforts have been made in the physical interpretation of fractional-order derivatives (see [44, 45], for example), but at the moment they do not have the acceptance that terms such as velocity and acceleration have in mechanics, for instance. The issue of the physical interpretation of fractional-order models would become more important if initial conditions were being assigned based on the initial state of the plant.

6. Concluding Remarks

Complex systems, of which heat exchangers and multiroom buildings are examples, can be found in widespread applications. For their performance control it is necessary to obtain an accurate description of their dynamic behavior. In this regard, fractional-order differential equations, derived via SI, may provide compact and efficient models of complex systems. In this work, a fractional-order-based system SI procedure grounded on global regression has been proposed to build accurate models from data. The SI methodology seeks the optimum values of a parameter set that includes the fractional order of the differential equation, its parametric constants, and, when necessary, the initial conditions. This last case arises when accurate knowledge of the initial conditions, whether the value of the function or its rate of change, is not possible, and such conditions have to be included as part of the search. It is to be noted, however, that there may be cases when some values of these conditions may not be physically feasible.

The application of the SI methodology to experimental measurements from a shell-and-tube heat exchanger and a multiroom building has confirmed that the approach is accurate and robust. The fractional-based model obtained via global regression provides better approximations to the data than those obtained by SI with local optimization algorithms. The results show that not only the apparent multiplicity in the parameter set that includes the order of the proposed model equation along with its constants and the initial conditions, as provided by the results from the global regression analysis, is due to the fact that the initial condition(s) is(are) included in the search, rather than in the model itself, but also

the proposed fractional-order differential equation model is reliable within the interpolation region, but cannot be used with confidence for predictions outside this region. The SI methodology proposed here is robust and can be used to derive accurate and compact models from experimental data. In addition, given a functional form of a fractional-order differential equation model, as new data become available the SI technique can be used to expand the region of reliability (interpolation) of the resulting model. The methodology described here can be extended to model other complex physical systems.

Nomenclature

- J^α : Fractional integral operator of order α
- c_1, c_2 : Coefficient for the linear fractional differential equation
- D_{RL}^α : Riemann-Liouville fractional derivative of order α
- D_C^α : Caputo fractional derivative of order α
- $E_{i,j}$: Two-parameter Mittag-Leffler function
- m : m -th integer order
- \dot{m} : Mass flow rate
- P_k : k -th initial condition
- q : Fractional order
- S_y, S_θ : Variance of the error
- N : Number of experimental data points
- C : Vector of search parameters
- T : Fluid bulk temperature [K]
- t : Time [s]
- $u(t)$: Input variable
- $y(t)$: Output variable

Greek Symbols

- ϵ : rms error
- $\Gamma(\xi)$: Gamma function
- θ : Dimensionless temperature
- ξ : Independent variable
- τ : Dimensionless time

Subscripts and Superscripts

- c : Cold fluid
- e : Experimental value
- h : Hot fluid
- in : Inlet
- out : Outlet
- p : Predicted value

Abbreviations

DAQ: Data acquisition system
 FDE: Fractional-order differential equation
 FOSI: Fractional-order system identification
 LOA: Local optimization algorithm
 ODE: Ordinary differential equation
 SA: Simulated annealing algorithm
 SI: System identification
 PDE: Partial differential equation.

Data Availability

The data used in this study can be made available, after publication, via a hyperlink to the corresponding author's personal website.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Kin M. Li was the recipient of a CREST-CEaS fellowship, for which we are grateful. The comments from an anonymous referee have greatly influenced the final version of this paper and are very much appreciated. This work has been supported by an NSF HRD-1547723 grant.

References

- [1] Z. S. Hou and Z. Wang, "From model-based control to data-driven control: survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.
- [2] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Pearson, Upper Saddle River, 13th edition, 2016.
- [3] A. Pacheco-Vega, C. Ruiz-Mercado, K. Peters, and L. E. Vilchiz, "On-line fuzzy-logic-based temperature control of a concentric-tube heat exchanger facility," *Heat Transfer Engineering*, vol. 30, no. 14, pp. 1208–1215, 2009.
- [4] A. Pacheco-Vega, G. Diaz, M. Sen, and K. T. Yang, *Applications of artificial neural networks and genetic methods in thermal engineering*, R. P. Chhabra, Ed., chapter 4.27, CRC Handbook of Thermal Engineering, 2nd edition, 2017.
- [5] G. Díaz, M. Sen, K. T. Yang, and R. L. McClain, "Adaptive neurocontrol of heat exchangers," *Journal of Heat Transfer*, vol. 123, no. 3, pp. 556–562, 2001.
- [6] R. L. Bagley and R. A. Calico, "Fractional order state equations for the control of viscoelastically damped structures," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 2, pp. 304–311, 1991.
- [7] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-Order Systems and Controls*, Advances in Industrial Control, Springer, London, UK, 2010.
- [8] V. E. Tarasov, *Fractional Dynamics: Applications of Fractional Calculus to Dynamics of Particles, Fields and Media*, Springer, Berlin, Germany, 2010.
- [9] S. Das, *Functional Fractional Calculus*, Springer, New Delhi, India, 2011.
- [10] J. T. Machado, V. Kiryakova, and F. Mainardi, "Recent history of fractional calculus," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 3, pp. 1140–1153, 2011.
- [11] E. C. de Oliveira and J. A. Tenreiro Machado, "A review of definitions for fractional derivatives and integral," *Mathematical Problems in Engineering*, vol. 2014, Article ID 238459, 6 pages, 2014.
- [12] M. D. Ortigueira and J. A. Tenreiro Machado, "What is a fractional derivative?" *Journal of Computational Physics*, vol. 293, pp. 4–13, 2015.
- [13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [14] I.-K. Jeong and J.-J. Lee, "Adaptive simulated annealing genetic algorithm for system identification," *Engineering Applications of Artificial Intelligence*, vol. 9, no. 5, pp. 523–532, 1996.
- [15] K. B. Oldham and J. Spanier, *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Academic Press, New York, NY, USA, 1974.
- [16] S. G. Samko, A. A. Kilbas, and O. I. Marichev, *Fractional Integrals and Derivatives: Theory and Applications*, Gordon and Breach Science Publishers, 1993.
- [17] I. Podlubny, *Fractional Differential Equations*, vol. 198 of *Mathematics in Science and Engineering*, Academic Press, San Diego, Calif, USA, 1999.
- [18] L. Vlahos, H. Isliker, Y. Kominis, and K. Hizanidis, "Normal and anomalous diffusion: A tutorial," 2008, <https://arxiv.org/abs/0805.0419>.
- [19] B. Goodwine, "Modeling a multi-robot system with fractional-order differential equations," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1763–1768, Hong Kong, China, May 2014.
- [20] M. Sen, J. P. Hollkamp, F. Semperlotti, and B. Goodwine, "Implicit and fractional-derivative operators in infinite networks of integer-order components," *Chaos, Solitons and Fractals*, vol. 114, pp. 186–192, 2018.
- [21] K. A. Lazopoulos, D. Karaoulanis, and A. K. Lazopoulos, "On fractional modelling of viscoelastic mechanical systems," *Mechanics Research Communications*, vol. 78, pp. 1–5, 2016.
- [22] Y. Aoki, M. Sen, and S. Paolucci, "Approximation of transient temperatures in complex geometries using fractional derivatives," *Heat and Mass Transfer*, vol. 44, no. 7, pp. 771–777, 2008.
- [23] S. M. Pineda, G. Diaz, and C. F. Coimbra, "Approximation of transient 1D conduction in a finite domain using parametric fractional derivatives," *Journal of Heat Transfer*, vol. 133, no. 7, p. 071301, 2011.
- [24] J.-D. Gabano and T. Poinot, "Fractional modelling and identification of thermal systems," *Signal Processing*, vol. 91, no. 3, pp. 531–541, 2011.
- [25] R. Caponetto, F. Sapuppo, V. Tomasello, G. Maione, and P. Lino, "Fractional-order identification and control of heating processes with non-continuous materials," *Entropy*, vol. 18, no. 11, article 398, 2016.
- [26] M. Caputo, "Linear models of dissipation whose Q is almost frequency independent-II," *The Geophysical Journal of the Royal Astronomical Society*, vol. 13, no. 5, pp. 529–539, 1967.
- [27] G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design: With Applications*, National Science Review, Mineola, New York, NY, USA, 1984.
- [28] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

- [29] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, UK, 1989.
- [30] F. Archetti and F. Schoen, "A survey on the global optimization problem: General theory and computational approaches," *Annals of Operations Research*, vol. 1, no. 2, pp. 87–110, 1984.
- [31] A. Gotmare, S. S. Bhattacharjee, R. Patidar, and N. V. George, "Swarm and evolutionary computing algorithms for system identification and filter design: A comprehensive review," *Swarm and Evolutionary Computation*, vol. 32, pp. 68–84, 2017.
- [32] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [33] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, 1987.
- [34] A. Pacheco-Vega, M. Sen, and K. T. Yang, "Simultaneous determination of in- and over-tube heat transfer correlations in heat exchangers by global regression," *International Journal of Heat and Mass Transfer*, vol. 46, no. 6, pp. 1029–1040, 2003.
- [35] E. R. Hansen, *Global Optimization using Interval Analysis*, vol. 165, Marcel Dekker, New York, NY, USA, 1992.
- [36] J. Mayes, *Reduction and Approximation in Large and Infinite Potential-Driven Flow Networks [Ph.D. thesis]*, University of Notre Dame, Notre Dame, Indiana, 2012.
- [37] K. M. Li, M. Sen, and A. Pacheco-Vega, "Fractional-Order-Based System Identification for Heat Exchangers," in *Proceedings of the 3rd World Congress on Momentum, Heat and Mass Transfer*, Budapest, Hungary, April 2018.
- [38] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," *National Science Review*, vol. 1, no. 2, pp. 293–314, 2014.
- [39] P. Courriou, "Three algorithms for estimating the domain of validity of feedforward neural networks," *Neural Networks*, vol. 7, no. 1, pp. 169–174, 1994.
- [40] P. Niyogi and F. Girosi, "Generalization bounds for function approximation from scattered noisy data," *Advances in Computational Mathematics*, vol. 10, no. 1, pp. 51–80, 1999.
- [41] A. Pacheco-Vega, M. Sen, K. T. Yang, and R. L. McClain, "Neural network analysis of fin-tube refrigerating heat exchanger with limited experimental data," *International Journal of Heat and Mass Transfer*, vol. 44, no. 4, pp. 763–770, 2001.
- [42] A. Baghdasarian, O. Ramos, J. Ruvalcaba et al., "Design and instrumentation of an experimental test-bed for research in energy efficiency in buildings," in *Proceedings of the ASME 2015 International Mechanical Engineering Congress and Exposition, IMECE 2015*, Houston, USA, November 2015.
- [43] J. Baltazar, A. Yarian, and A. Pacheco-Vega, "Development of P-PD- and PID-fuzzy SISO controllers of a sub-scaled multi-room building test-bed," in *Proceedings of the 3rd Thermal and Fluids Engineering Conference (TFEC)*, Tampa, USA, March, 2018.
- [44] J. F. Gómez-Aguilar, R. Razo-Hernández, and D. Granados-Lieberman, "A physical interpretation of fractional calculus in observables terms: analysis of the fractional time constant and the transitory response," *Revista Mexicana de Física*, vol. 60, no. 1, pp. 32–38, 2014.
- [45] V. E. Tarasov, "Interpretation of fractional derivatives as reconstruction from sequence of integer derivatives," *Fundamenta Informaticae*, vol. 151, no. 1-4, pp. 431–442, 2017.

Research Article

Maximizing the Coverage of Roadmap Graph for Optimal Motion Planning

Jae-Han Park ¹ and Tae-Woong Yoon ²

¹Robotics R&D Group, Korea Institute of Industrial Technology (KITECH), Ansan, Republic of Korea

²School of Electrical Engineering, Korea University, Seoul, Republic of Korea

Correspondence should be addressed to Tae-Woong Yoon; twy@korea.ac.kr

Received 26 April 2018; Revised 15 September 2018; Accepted 4 October 2018; Published 8 November 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Jae-Han Park and Tae-Woong Yoon. 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.

Automated motion-planning technologies for industrial robots are critical for their application to Industry 4.0. Various sampling-based methods have been studied to generate the collision-free motion of articulated industrial robots. Such sampling-based methods provide efficient solutions to complex planning problems, but their limitations hinder the attainment of optimal results. This paper considers a method to obtain the optimal results in the roadmap algorithm that is representative of the sampling-based method. We define the coverage of a graph as a performance index of its optimality as constructed by a sampling-based algorithm and propose an optimization algorithm that can maximize graph coverage in the configuration space. The proposed method was applied to the model of an industrial robot, and the results of the simulation confirm that the roadmap graph obtained by the proposed algorithm can generate results of satisfactory quality in path-finding tests under various conditions.

1. Introduction

Automatic motion planning for robots is an important technology for next-generation manufacturing systems, such as Industry 4.0. Sampling-based algorithms have been widely used in robot motion-planning problems because they are efficient at providing reasonable solutions [1–4]. Sampling-based algorithms determine feasible paths for the robot's motion using information from a graph that consists of randomly sampled nodes and connected edges in the given configuration space. Such randomized approaches have a strong advantage in terms of quickly providing solutions to complex problems, such as in a high-dimensional configuration space. However, sampling-based methods guarantee probabilistic completeness, and thus they require a large number of sampling nodes to find paths for the motion of the robot in arbitrary cases [2]. The *probabilistic roadmap (PRM)* [3] and *rapidly exploring random trees (RRT)* [5] are representative of such sampling-based motion-planning algorithms.

The PRM method is composed of learning and query phases. In the query phase, collision-free paths for the motion

of the robot are obtained by using information from a graph known as the roadmap [2, 3]. The learning phase is the process of constructing a graph that represents the morphology of the given configuration space. In this phase, it is important to construct the graph appropriately to obtain the shortest path for the collision-free motion of the robot. Developments have recently been reported on the optimality of sampling-based algorithms [6–9]. For example, the PRM* algorithm provides criteria for connecting edges that can guarantee asymptotic optimality in terms of the probability of finding path.

An important factor to consider when constructing a graph for optimal motion planning is the node sampling strategy used. Sampling-based algorithms were originally designed with a random sampling of uniform distribution [2, 10]. However, a few drawbacks have been discovered in these uniform sampling strategies, such as the narrow passage problem and failure to capture the true connectivity of spaces of arbitrary shapes [9–11]. Thus, various sampling schemes, such as the Voronoi diagram [11], Gaussian random sampling [12], the bridge test [13], and visibility PRM [14]

algorithms, have been proposed to overcome drawbacks in the uniform sampling method. Nevertheless, the effect of sampling strategies on optimal motion planning remains unclear. The node sampling strategy that fully captures the shape of the free configuration space remains an important problem in optimal motion planning [9, 10]. In the query phase of the roadmap method, the shortest path is obtained as the optimal path using graph information. Thus, constructing an appropriate graph is closely related to the problem of optimal motion planning.

Another area highly relevant to our work here is research on optimization techniques. In recent years, with growing interest in machine learning and Artificial Intelligence, techniques of distributed optimization that can handle large amounts of data and high-dimensional variables have emerged as an important issue. In [15, 16], distributed optimization techniques were considered for multi-agent systems in complex networks. Secure and resilient techniques [17–19] have also been developed to guarantee the stability of distributed algorithms in adversarial environments. As highly relevant to our work here, coverage maximization was studied using methods of distributed optimization in applications of sensor networks [20–22].

This paper proposes an algorithm to optimize the roadmap graph that can cover arbitrary morphologies of the free configuration space to maximize coverage. Our previous study [23] considered an adaptation algorithm for geometric graphs in an intuitive manner. However, in this study, we reanalyze that adaptive graph algorithm and present its results in terms of the maximization of graph coverage. We found that the optimization algorithm updates the positions of the nodes along the direction of expanding the region of covering until it reaches equilibrium, where expansion and contraction factors are balanced. The results of a simulation to test the proposed optimization algorithm show that it can construct a suitable graph to describe arbitrary configuration spaces, and the resulting graph can generate paths of shorter length for various test queries.

2. Problem Statement

The *configuration space* $\mathbf{Q} \subset \mathbf{R}^n$ is the space of all possible configurations of the system, where n is the number of degrees of freedom (DOF) of the system and \mathbf{R}^n is n -dimensional Euclidean space [1]. In the formulation of the configuration space, the robot can be abstracted as a point $\mathbf{q} \in \mathbf{Q}$. *Free configuration* is defined as one where the robot $R(\mathbf{q})$ does not intersect with any obstacle in the workspace. The set of all free configurations is defined as the *free configuration space* $\mathbf{Q}_{free} \subset \mathbf{Q}$, and it can be determined by checking for collisions between obstacles and the robot in the workspace. Every workspace obstacle WO_i is mapped as an obstacle in the configuration space $QO_i = \{\mathbf{q} \in \mathbf{Q} \mid R(\mathbf{q}) \cap WO_i \neq \emptyset\}$ [2]. Figure 1 shows the mapping relations between the workspace and the configuration space for a two-link articulated robot.

The problem of motion planning requires as solution a feasible path in the free configuration space for a specified initial motion and a goal. The *path planning problem* can then be defined as follows:

Definition 1 ((path planning problem) [1]). Given a set of collision-free configurations $\mathbf{Q}_{free} \subset \mathbf{Q}$, and initial and goal configurations $\mathbf{q}_{init}, \mathbf{q}_{goal} \in \mathbf{Q}_{free}$, find a continuous curve $\sigma \in \Sigma = \{t \mid t : [0, 1] \rightarrow \mathbf{Q}_{free}\}$, where $\sigma(0) = \mathbf{q}_{init}$ and $\sigma(1) = \mathbf{q}_{goal}$.

In this study, we employ the *roadmap* method for path planning. The roadmap represents the configuration space topologically by a network of one-dimensional (1D) curves. It is defined as follows.

Definition 2 ((roadmap) [2]). A union of 1D curves is a *roadmap* \mathbf{RM} if, for all \mathbf{q}_{init} and \mathbf{q}_{goal} in \mathbf{Q}_{free} that can be connected by a path, the following properties hold:

- (1) **Accessibility:** there exists a path from $\mathbf{q}_{init} \in \mathbf{Q}_{free}$ to some $\mathbf{q}'_{init} \in \mathbf{RM}$,
- (2) **Departability:** there exists a path from $\mathbf{q}'_{goal} \in \mathbf{RM}$ to $\mathbf{q}_{goal} \in \mathbf{Q}_{free}$,
- (3) **Connectivity:** there exists a path in \mathbf{RM} between \mathbf{q}'_{init} and \mathbf{q}'_{goal} .

The result of the roadmap method is a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ consisting of a set of N nodes $\mathbf{V} = \{\mathbf{q}_i \in \mathbf{Q}_{free} \mid \forall i \in \{1, \dots, N\}\}$ and a set of edges $\mathbf{E} = \{e_{ij} = (\mathbf{q}_i, \mathbf{q}_j) \mid \mathbf{q}_i, \mathbf{q}_j \in \mathbf{V}\}$. Edge e_{ij} is created if the line segment between nodes \mathbf{q}_i and \mathbf{q}_j persists in \mathbf{Q}_{free} , i.e., $t\mathbf{q}_i + (1-t)\mathbf{q}_j \in \mathbf{Q}_{free} \forall t \in [0, 1]$.

To obtain a feasible path for the robot's motion using information from the roadmap graph, the three properties in Definition 2 need to be met. Furthermore, for optimal planning, the best roadmap must be determined among a set of feasible roadmaps. We now employ roadmap coverage as a performance index to assess the optimality of the roadmap graph.

To describe roadmap coverage, the *covering region of node* \mathbf{q}_i for a certain *neighbor radius* r_η , is defined as follows:

$$\mathbf{B}_i = \mathbf{B}(\mathbf{q}_i, r_\eta) = \left\{ \mathbf{q} \in \mathbf{R}^n \mid \|\mathbf{q} - \mathbf{q}_i\|_2 \leq \frac{r_\eta}{2}, \mathbf{q}_i \in \mathbf{V} \right\}. \quad (1)$$

Let $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_N\}$ be the family of sets of \mathbf{B}_i ; subsequently, the *roadmap coverage* for neighbor radius r_η is defined as the union of \mathbf{B}_i s as follows.

Definition 3 (roadmap coverage). *Roadmap coverage* \mathbf{C} for roadmap \mathbf{G} is the region of the union of the family of sets \mathcal{B} :

$$\mathbf{C}(\mathbf{G}) = \bigcup \mathcal{B} = \mathbf{B}_1 \bigcup \mathbf{B}_2 \bigcup \dots \bigcup \mathbf{B}_N. \quad (2)$$

Figure 2 shows an example of a roadmap and its coverage for a configuration space.

Let $\mu(\cdot)$ be the Lebesgue measure; by the inclusion-exclusion principle [25, 26], the volume of roadmap coverage $\mu(\mathbf{C})$ can be represented as follows:

$$\mu(\mathbf{C}(\mathbf{G})) = \mu\left(\bigcup \mathcal{B}\right) = \sum_{i=1}^N (-1)^{i-1} S_i, \quad (3)$$

where $S_i = \sum_{1 \leq j_1 < j_2 < \dots < j_i \leq N} \mu(\mathbf{B}_{j_1} \cap \mathbf{B}_{j_2} \cap \dots \cap \mathbf{B}_{j_i})$.

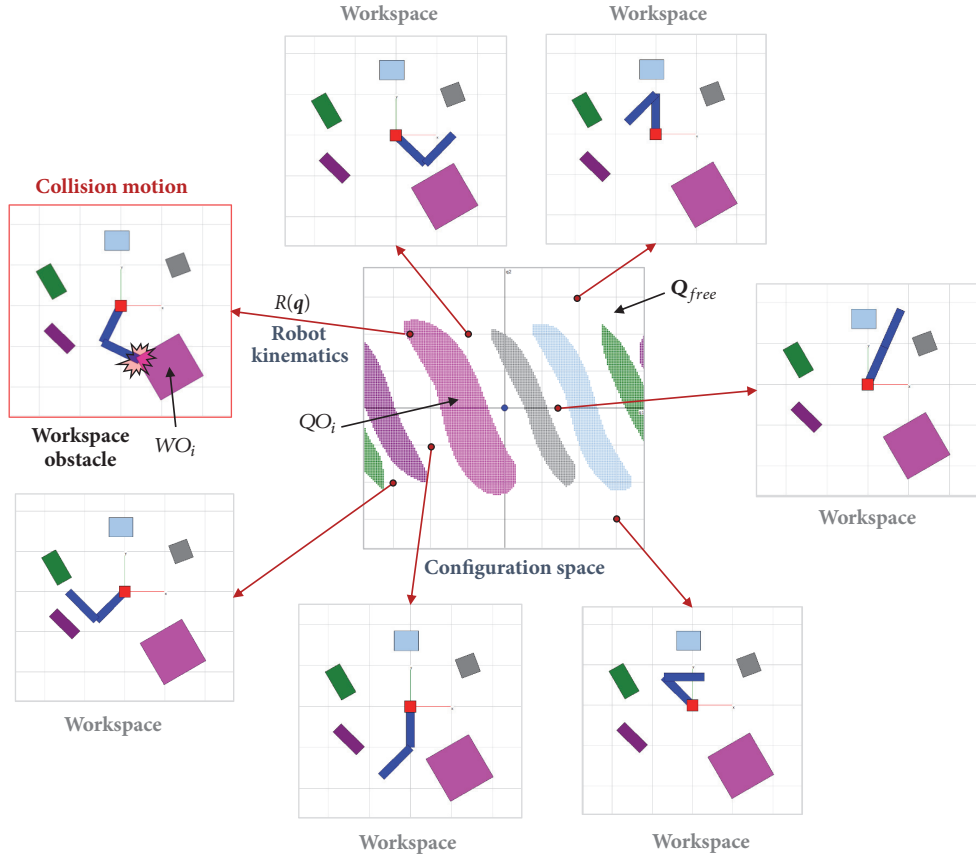


FIGURE 1: Mapping relations between the workspace and the configuration space.

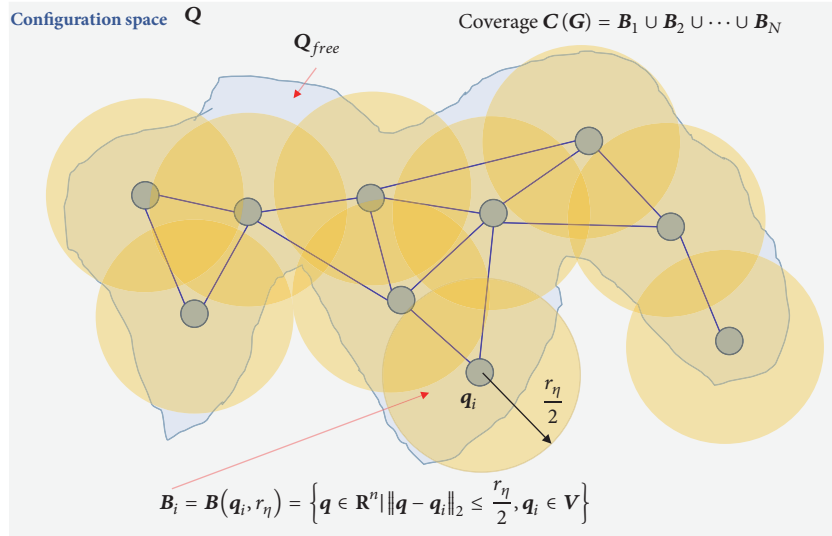


FIGURE 2: Example of a roadmap and its coverage.

According to the properties in Definition 2, the optimality of the roadmap can be considered to be the capability of finding paths for an arbitrarily set initial point and goal in the configuration space. To satisfy this requirement, roadmap

coverage should cover the entire free configuration space. As a quantitative representation of this performance index, we define the *optimal roadmap* as the graph that maximizes the volume of roadmap coverage as follows.

Definition 4 (optimal roadmap). The *optimal roadmap* \mathbf{G}^* is the graph that maximizes the volume of roadmap coverage within the finite free configuration space:

$$\begin{aligned} \mathbf{G}^* &= \underset{\mathbf{V}=\{q_1, q_2, \dots, q_N\}}{\text{maximize}} && \mu(\mathbf{C}(\mathbf{G})) \\ &\text{subject to} && \mathbf{q}_i \in \mathbf{Q}_{free}, \quad i = 1, \dots, N. \end{aligned} \quad (4)$$

The volume of roadmap coverage is primarily related to the geometric structure of the graph. It is complicated to calculate, especially in case the configuration space is high dimensional and the roadmap consists of a large number of nodes. Thus, an alternative performance index that uses the upper or lower bound for the volume of roadmap coverage is needed to render the optimization problem solvable.

Let the sum of terms consisting of two or more intersections be $I(\mathcal{B}) = \sum_{i=2}^N (-1)^i S_i$. Therefore, (3) can be written as

$$\begin{aligned} \mu(\mathbf{C}(\mathbf{G})) &= \mu\left(\bigcup \mathcal{B}\right) = \sum_{i=1}^N (-1)^{i-1} S_i \\ &= S_1 + \sum_{i=2}^N (-1)^{i-1} S_i = \sum_{i=1}^N \mu(\mathbf{B}_i) - I(\mathcal{B}), \end{aligned} \quad (5)$$

where $\sum_{i=1}^N \mu(\mathbf{B}_i) = N \cdot \Gamma(n/2 + 1)^{-1} (\sqrt{\pi} \cdot r_\eta)^n$, and $\Gamma(x)$ is the gamma function. The problem of designing the optimal roadmap can be transformed into a minimization problem as follows:

$$\begin{aligned} \mathbf{G}^* &= \underset{\mathbf{V}=\{q_1, q_2, \dots, q_N\}}{\text{minimize}} && I(\mathcal{B}) \\ &\text{subject to} && \mathbf{q}_i \in \mathbf{Q}_{free}, \quad i = 1, \dots, N. \end{aligned} \quad (6)$$

Moreover, the upper bound of $I(\mathcal{B})$, which is the sum of the intersection terms, can be expressed as follows [Section A.1]:

$$I(\mathcal{B}) \leq \kappa \cdot S_2 = \kappa \cdot \left(\sum_{1 \leq i < j \leq N} \mu(\mathbf{B}_i \cap \mathbf{B}_j) \right), \quad (7)$$

where $\kappa = 2/N(N-1) \cdot \sum_{i=2}^N \left(\prod_{j=2}^i \binom{N}{j} \right)$ is a finite constant that depends on the number of nodes N .

This upper bound on $I(\mathcal{B})$ can be applied to the following *suboptimal roadmap* problem.

Definition 5 (suboptimal roadmap). The *suboptimal roadmap* $\widehat{\mathbf{G}}^*$ is the graph that minimizes the upper bound of $I(\mathcal{B})$ in (7) such that

$$\begin{aligned} \widehat{\mathbf{G}}^* &= \underset{\mathbf{V}=\{q_1, q_2, \dots, q_N\}}{\text{minimize}} && \sum_{1 \leq i < j \leq N} \mu(\mathbf{B}_i \cap \mathbf{B}_j) \\ &\text{subject to} && \mathbf{q}_i \in \mathbf{Q}_{free}, \quad i = 1, \dots, N. \end{aligned} \quad (8)$$

3. Coverage Maximization Algorithm

Let $\Phi(\mathbf{q}_i, \mathbf{q}_j)$ be the volume of the region of intersection between nodes, $\mu(\mathbf{B}_i \cap \mathbf{B}_j)$. By replacing the constraints $\mathbf{q}_i \in \mathbf{Q}_{free}$ with the collision detection function $s(\mathbf{q}_i)$, the optimization problem in Definition 5 can be represented as follows:

$$\begin{aligned} \widehat{\mathbf{G}}^* &= \underset{\mathbf{V}=\{q_1, q_2, \dots, q_N\}}{\text{minimize}} && \sum_{1 \leq i < j \leq N} \Phi(\mathbf{q}_i, \mathbf{q}_j) \\ &\text{subject to} && s(\mathbf{q}_i) = 0, \quad i = 1, \dots, N, \end{aligned} \quad (9)$$

where $s(\mathbf{q}_i)$ is the collision detection function defined as

$$s(\mathbf{q}_i) = \begin{cases} 1 & \text{(collision detected)} \\ 0 & \text{(otherwise)}. \end{cases} \quad (10)$$

The volume of the region of intersection between n -dimensional hyperspheres with respect to the neighbor radius r_η is

$$\Phi(\mathbf{q}_i, \mathbf{q}_j) = \begin{cases} 2 \cdot \int_{d_{ij}/2}^{r_\eta/2} \frac{\sqrt{\pi}^{n-1}}{\Gamma(1 + (n-1)/2)} \sqrt{(r_\eta/2)^2 - t^2}^{n-1} dt & (0 \leq d_{ij} < r_\eta) \\ 0 & (d_{ij} \geq r_\eta), \end{cases} \quad (11)$$

where $d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|_2$. See [Section A.2] for details.

Using the *Lagrange multiplier* λ , the optimization problem in (9) can be transformed as follows [27]:

$$\widehat{\mathbf{G}}^* = \underset{\mathbf{H}}{\text{minimize}} \quad \mathbf{H}, \quad (12)$$

where $\mathbf{H} = \sum_{1 \leq i < j \leq N} \Phi(\mathbf{q}_i, \mathbf{q}_j) + \sum_{i=1}^N \lambda_i s(\mathbf{q}_i)$.

The solution of this minimization problem is a node set that makes the gradient of \mathbf{H} to zero, i.e., $\nabla \mathbf{H} = \sum_{1 \leq i < j \leq N} \nabla \Phi(\mathbf{q}_i, \mathbf{q}_j) + \sum_{i=1}^N \lambda_i \nabla s(\mathbf{q}_i) = 0$.

Let $\mathbf{x} = [\mathbf{q}_i] \in \mathbf{R}^{nN}$ be the stacked vector of N nodes. Using the steepest descent method [27], the algorithm to minimize (9) can be described as follows:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{x}[k] - \varepsilon \nabla \mathbf{H}_k, \\ \nabla \mathbf{H}_k &= \sum_{1 \leq i < j \leq N} \nabla \Phi(\mathbf{q}_i[k], \mathbf{q}_j[k]) \\ &\quad + \sum_{i=1}^N \lambda_i[k] \nabla s(\mathbf{q}_i[k]), \end{aligned} \quad (13)$$

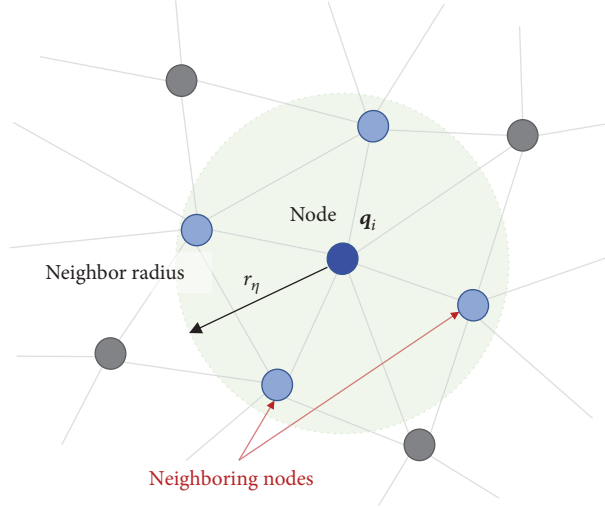


FIGURE 3: Nodes neighboring \mathbf{q}_i within radius r_η [23].

where k is the iteration index, and $\varepsilon > 0$ is the step size of each iteration. By the penalty parameter $c[k]$, the Lagrange multiplier $\lambda_i[k]$ can be determined by the following updating formula [27]:

$$\lambda_i[k+1] = \lambda_i[k] + c[k] s(\mathbf{q}_i[k]), \quad (14)$$

where the penalty parameter is chosen to satisfy $c[k+1] \geq c[k]$.

If the distance $d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|_2$ between nodes \mathbf{q}_i and \mathbf{q}_j is greater than the neighbor radius r_η , the volume of the region of intersection becomes zero, i.e., $\Phi(\mathbf{q}_i, \mathbf{q}_j) = 0$. Also $\nabla\Phi(\mathbf{q}_i, \mathbf{q}_j)$ for node \mathbf{q}_i is valid only when \mathbf{q}_j is located within r_η . Thereby, as shown in Figure 3, we define the *neighboring node* set located within the neighbor radius as

$$\eta_i = \{\mathbf{q}_j \in \mathbf{V} \mid \|\mathbf{q}_j - \mathbf{q}_i\|_2 \leq r_\eta, j \neq i\}. \quad (15)$$

Furthermore, using the *decomposition method* [28–30], the process in (13) can be transformed into a decentralized form that is processed at each node as follows:

$$\begin{aligned} \mathbf{q}_i[k+1] = & \mathbf{q}_i[k] - \varepsilon \sum_{\mathbf{q}_j[k] \in \eta_i} \nabla\Phi(\mathbf{q}_i[k], \mathbf{q}_j[k]) \\ & - \varepsilon \lambda_i[k] \nabla s(\mathbf{q}_i[k]). \end{aligned} \quad (16)$$

$\nabla s(\mathbf{q}_i)$ in (16) is, however, not available because $s(\mathbf{q}_i)$ is discontinuous. In such cases, the gradient of $s(\mathbf{q}_i)$ can be estimated stochastically through *response surface method* (RSM) [31–33]. To apply the RSM algorithm, we define the *sensing node* $\Delta\mathbf{q} \in \mathbf{R}^n$ as follows.

Definition 6 (sensing node). A *sensing node* $\Delta\mathbf{q}$ is a point evenly sampled by Poisson-disk sampling [34] on $\partial\mathbf{B}_{r_s} = \{\mathbf{q} \in \mathbf{R}^n \mid \|\mathbf{q}\|_2 = r_s\}$, which is the surface of an n -dimensional hypersphere of radius r_s . The sensing nodes detect the

deformation of the configuration space around each node. The set of sensing nodes \mathbf{S} can be expressed as

$$\begin{aligned} \mathbf{S} = \{ & \Delta\mathbf{q}_l \in \partial\mathbf{B}_{r_s} \mid l = 1, \dots, n_s, \|\Delta\mathbf{q}_i - \Delta\mathbf{q}_j\|_2 \\ & > r_p \text{ for } \forall i, j \in \{1, \dots, n_s\}\}, \end{aligned} \quad (17)$$

where r_p and n_s denote the sampling radius of the Poisson disk and the number of sensing nodes, respectively.

Figure 4(a) shows an example of sensing nodes sampled on the surface of a sphere, and Figure 4(b) shows a node and its sensing nodes in 3D configuration space.

Let $\mathbf{M} = [\Delta\mathbf{q}_l^T] \in \mathbf{R}^{n_s \times n}$ be a matrix composed of sensing nodes and $\mathbf{w}_i = [s(\mathbf{q}_i + \Delta\mathbf{q}_l)] \in \mathbf{R}^{n_s}$ be a stacked vector representing sensing information. If the sensing nodes are selected symmetrically such that $\sum_{l=1}^{n_s} \Delta\mathbf{q}_l = \mathbf{0}$, the estimated gradient $\nabla\hat{s}(\mathbf{q}_i)$ can be calculated by the RSM as follows:

$$\nabla s(\mathbf{q}_i) \approx \hat{\nabla}s(\mathbf{q}_i) = \nabla\hat{s}(\mathbf{q}_i) = \mathbf{M}^+ \mathbf{w}_i, \quad (18)$$

where $\mathbf{M}^+ = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T$ is the pseudoinverse matrix of \mathbf{M} and $\mathbf{M}^T \mathbf{M}$ is a nonsingular matrix. See [Section A.3] for details.

$\Phi(\mathbf{q}_i, \mathbf{q}_j)$ represents the interaction between two nodes \mathbf{q}_i and \mathbf{q}_j , and it is a function of the distance $d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|_2$. The gradient of $\Phi(\mathbf{q}_i, \mathbf{q}_j)$ can be described as follows [35]:

$$\nabla\Phi(\mathbf{q}_i, \mathbf{q}_j) = \frac{\partial\Phi(d_{ij})}{\partial d_{ij}} \cdot \frac{\mathbf{q}_i - \mathbf{q}_j}{d_{ij}}. \quad (19)$$

Then, it follows from (11) that

$$\begin{aligned} & \frac{\partial\Phi(d_{ij})}{\partial d} \\ & = \begin{cases} -2 \frac{\sqrt{\pi/4}^{n-1}}{\Gamma(1+(n-1)/2)} \sqrt{r_\eta^2 - d_{ij}^2}^{n-1} & (0 \leq d_{ij} < r_\eta) \\ 0 & (d_{ij} \geq r_\eta). \end{cases} \end{aligned} \quad (20)$$

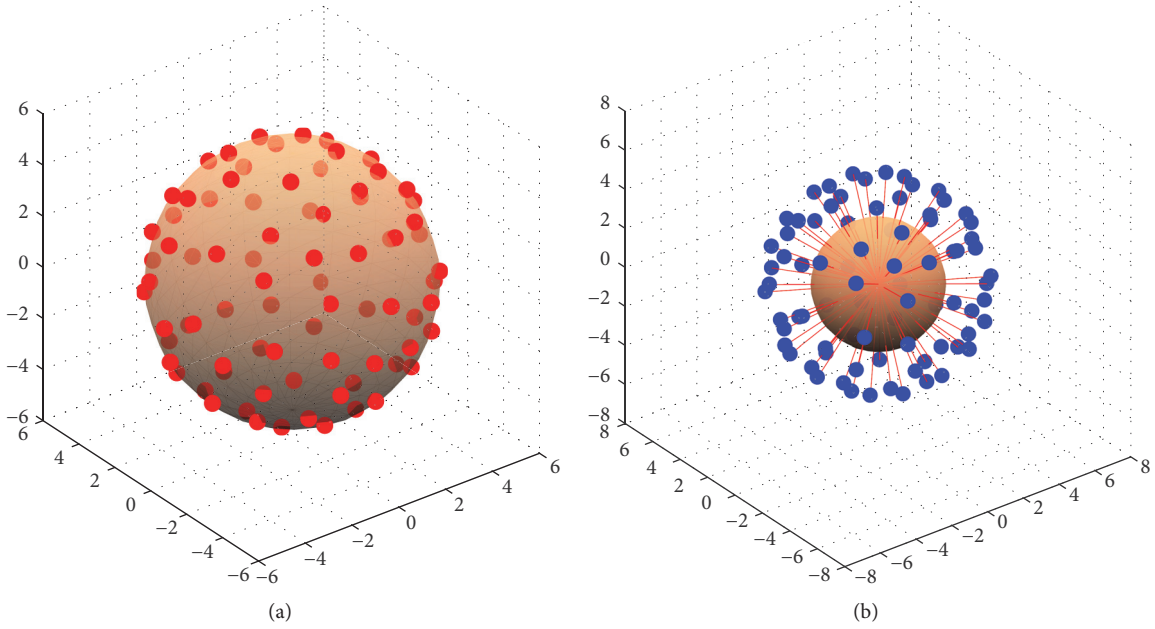


FIGURE 4: Example of sampled sensing nodes and a node in 3D configuration space. (a) Sensing nodes sampled on $\partial\mathbf{B}_{r_s}$ ($r_s = 5$). (b) A node and its sensing nodes.

It is noteworthy that $\partial\Phi(d_{ij})/\partial d \leq 0$. We now set a non-negative value of $g(d_{ij})$ as

$$g(d_{ij}) = \begin{cases} -\frac{1}{d_{ij}} \cdot \frac{\partial\Phi(d_{ij})}{\partial d_{ij}} & (0 < d_{ij} < r_\eta) \\ 0 & (\text{otherwise}). \end{cases} \quad (21)$$

Subsequently, the distributed optimization algorithm in (16) can be reformulated as follows:

$$\begin{aligned} \mathbf{q}_i[k+1] &= \mathbf{q}_i[k] \\ &+ \varepsilon \sum_{\mathbf{q}_j[k] \in \eta_i} g(d_{ij}[k]) (\mathbf{q}_i[k] - \mathbf{q}_j[k]) \\ &- \varepsilon \lambda_i[k] \mathbf{M}^+ \mathbf{w}_i[k]. \end{aligned} \quad (22)$$

In this algorithm, the major variable influencing the amount of computation needed is the number of nodes N . The main computational tasks of the optimization process are the construction of the Laplacian matrix and the computation of the sensing vector. In the original optimization process in (13), computational complexities according to the number of nodes N are estimated as quadratic $O(N^2)$ for the Laplacian matrix and linear $O(N)$ for the sensing vector, respectively. However, as this algorithm can be decomposed as shown in (16), if a many-core processor such as a General-Purpose computing on Graphics Processing Unit (GPGPU) is available, (22) can be computed at each node in a separate processor in parallel. Thus, if the optimization algorithm can be executed by using parallel computation, the computational complexity of the Laplacian matrix and the configuration of the sensing vector are linear $O(N)$ and constant $O(c)$, respectively.

The algorithm in (22) is an iterative process that can converge to a suboptimal roadmap by detecting the boundary of the state of collision of the configuration space through the sensing vector \mathbf{w}_i . Even if the morphology of the configuration space changes, the structure of the roadmap graph can adapt to it accordingly because the optimization process is continuously performed using the sensing node. In the general optimization algorithm, the Lagrange multiplier $\lambda_i[k]$ is updated at every iteration. This is suitable in fixed configuration spaces, but it is difficult to apply to changeable spaces because if the space changes, the updated $\lambda_i[k]$ becomes useless and needs to be recalculated to obtain the initial conditions. We set the Lagrange multiplier to a constant value λ as an iteration gain. Further, we set the same value for each node such that $\lambda_1 = \lambda_2 = \dots = \lambda_N = \lambda$. The results of a simulation show that the algorithm works well with a fixed λ even with a changing configuration space.

4. Equilibrium State Analysis

This section analyzes the states of equilibrium of the algorithm in (22). For the equilibrium analysis, the differential equation for the vector $\mathbf{x} = [\mathbf{q}_i] \in \mathbf{R}^{nN}$ should be considered.

Let the vector $\mathbf{w} = [\mathbf{w}_i] \in \mathbf{R}^{n \times N}$ be the stacked vector of sensing vector \mathbf{w}_i at each node, and let the *Laplacian matrix* $\mathbf{L} = [L_{ij}] \in \mathbf{R}^{N \times N}$ for $g(d_{ij}) \geq 0$ be as follows:

$$\mathbf{L} = [L_{ij}], \quad L_{ij} = \begin{cases} -g(d_{ij}) & i \neq j \\ \sum_{k=1, i \neq k}^N g(d_{ik}) & i = j. \end{cases} \quad (23)$$

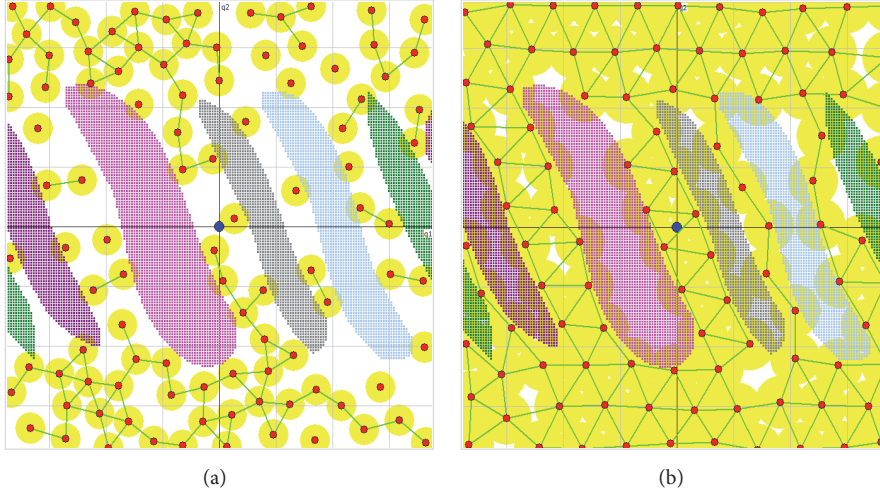


FIGURE 5: Examples of states of equilibrium: The yellow region shows the coverage of the roadmap. (a) An ill-structured roadmap. (b) A properly constructed roadmap.

With the Lagrange multipliers λ_i set to $\lambda_1 = \lambda_2 = \dots = \lambda_N = \lambda$, the decentralized differential equation shown in (22) can be integrated for $\mathbf{x} \in \mathbf{R}^{nN}$ as follows [Section A.4]:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{x}[k] + \varepsilon (\mathbf{L}[k] \otimes \mathbf{I}_n) \mathbf{x}[k] \\ &\quad - \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}[k] \\ &= (\mathbf{I}_{nN} + \varepsilon (\mathbf{L}[k] \otimes \mathbf{I}_n)) \mathbf{x}[k] \\ &\quad - \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}[k], \end{aligned} \quad (24)$$

where \otimes is the *Kronecker product operator* and $\mathbf{I}_k \in \mathbf{R}^{k \times k}$ is a k -dimensional identity matrix.

Let \mathbf{x}^* be the whole-state variable in equilibrium, and let \mathbf{w}^* and \mathbf{L}^* be the entire sensing vector and the Laplacian matrix at that time, respectively. Subsequently, the differential equation at equilibrium is

$$\begin{aligned} \mathbf{x}^* &= (\mathbf{I}_{nN} + \varepsilon (\mathbf{L}^* \otimes \mathbf{I}_n)) \mathbf{x}^* - \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}^* \\ &= \mathbf{x}^* + \varepsilon (\mathbf{L}^* \otimes \mathbf{I}_n) \mathbf{x}^* - \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}^*. \end{aligned} \quad (25)$$

In (25), the condition that renders the whole-state vector \mathbf{x} in equilibrium is

$$\varepsilon (\mathbf{L}^* \otimes \mathbf{I}_n) \mathbf{x}^* - \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}^* = \mathbf{0}. \quad (26)$$

The cases of states of equilibrium that can meet such a condition can be summarized as follows.

Remark 7 (equilibrium conditions of suboptimal roadmap problem). The *conditions of equilibrium states* for the suboptimal roadmap algorithm are classified into three cases as follows:

Case 1 ($\mathbf{L}^* = \mathbf{0}, \mathbf{w}^* = \mathbf{0}$). All nodes and their corresponding sensing nodes are in \mathbf{Q}_{free} . Furthermore, no neighbor node exists because the neighbor radius r_η is small for $\mu(\mathbf{Q}_{free})$,

such that $\mu(\mathbf{Q}_{free}) > N \cdot \mu(\mathbf{B}(\mathbf{q}_i, r_\eta))$. Figure 5(a) shows an example of this case.

Case 2 ($\mathbf{L}^* \neq \mathbf{0}, \mathbf{x}^* \in \text{Null}(\mathbf{L}^* \otimes \mathbf{I}_n), \mathbf{w}^* = \mathbf{0}$). In this case, \mathbf{x} is a null-space element of the matrix $\mathbf{L}^* \otimes \mathbf{I}_n$, and all sensing nodes are in \mathbf{Q}_{free} . If the graph is formed by a single connected component, the null-space elements are $\mathbf{x} = [\mathbf{q}_i]$, where $\mathbf{q}_1 = \mathbf{q}_2 = \dots = \mathbf{q}_N$ [36]. Such states are rare in dynamical processes. Thus, herein, we do not consider this type of equilibrium.

Case 3 ($\mathbf{L}^* \neq \mathbf{0}, \mathbf{x}^* \notin \text{Null}(\mathbf{L}^* \otimes \mathbf{I}_n), \mathbf{w}^* \neq \mathbf{0}$). The neighbor nodes and information regarding the sensing nodes exist. In this case, the equilibrium is a point, and this is the desirable solution to the suboptimal roadmap problem. Figure 5(b) shows an example of this case. In this state, the expanding factor between the neighbor nodes and the factor of sensing nodes that detect the boundary of collision are in balance as follows:

$$(\mathbf{L}^* \otimes \mathbf{I}_n) \mathbf{x}^* = \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}^*. \quad (27)$$

The maximum rank of the Laplacian matrix is $n-1$ [37]. Although information concerning the Laplacian matrix \mathbf{L}^* and sensing vector \mathbf{w}^* in equilibrium state is given, the state vector \mathbf{x}^* cannot be calculated using (27). However, if the graph is a single connected component such as $\text{rank}(\mathbf{L}^*) = n-1$ [36, 37], the state vector \mathbf{x}^* can be obtained with some additional constraints.

If a constraint such as $\sum_{i=1}^N \mathbf{q}_i = \mathbf{0}$ is added, the state vector in equilibrium can be calculated as follows:

$$\mathbf{x}^* = \left(\left(\begin{bmatrix} \mathbf{L}^* \\ \frac{1}{N} \cdot \mathbf{1}_N^T \end{bmatrix} \otimes \mathbf{I}_n \right)^+ \begin{bmatrix} \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}^* \\ \mathbf{0} \end{bmatrix} \right), \quad (28)$$

where the symbol “+” indicates the pseudoinverse matrix and $\mathbf{1}_N$ is an N -dimensional vector.

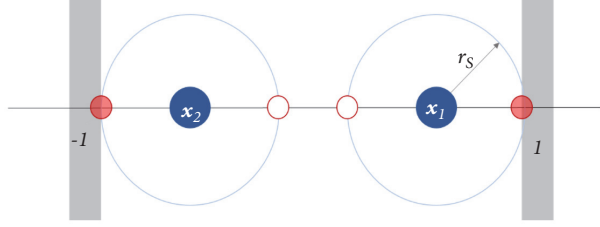


FIGURE 6: Structure of roadmap for Example 8.

We now verify this result through a simple example of a roadmap with two nodes in a 1D configuration space as shown in Figure 6.

Example 8 (two-node roadmap in 1D space). Obtain the equilibrium point $\mathbf{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$ for $\mathbf{Q}_{free} = \{x \in \mathbf{R} \mid -1 \leq x \leq 1\}$, $r_s = 0.6$, $\lambda = 1$, $\mathbf{L}^* = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, and $\mathbf{w}^* = [1 \ 0 \ 0 \ 1]^T$.

The sensing node matrix is $\mathbf{M} = \begin{bmatrix} 0.6 \\ -0.6 \end{bmatrix}$ and its pseudo-matrix is $\mathbf{M}^+ = [0.8333 \ -0.8333]$. The configuration space is symmetric to the origin. Thus, a constraint on the center of mass can be added, such as $(x_1 + x_2)/2 = 0$. Subsequently, the equilibrium point can be calculated using (28) as follows:

$$\mathbf{x}^* = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 1/2 & 1/2 \end{bmatrix}^+ \cdot \begin{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes [0.8333 \ -0.8333] \right) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ 0 \end{bmatrix} \quad (29)$$

$$= \begin{bmatrix} 0.4167 \\ -0.4167 \end{bmatrix}$$

Hence, the roadmap at equilibrium is as shown in Figure 7.

Figure 8 shows the phase portrait of the optimization process for various initial conditions. As shown in the phase portrait, the states of \mathbf{x} converge for all initial conditions to $\mathbf{x}^* = [0.4167 \ -0.4167]^T$ or $\mathbf{x}^* = [-0.4167 \ 0.4167]^T$, as calculated in (29). In case of convergence to $\mathbf{x}^* = [-0.4167 \ 0.4167]^T$, the sensing vector is $\mathbf{w}^* = [0 \ 1 \ 1 \ 0]^T$.

Figure 9 shows the phase portrait of example 1 for $r_\eta = 0.6$, where the neighbor radius is small for the volume of the configuration space. In this case, the equilibrium condition of the optimization process is the first case of Remark 7. The state of equilibrium is not a point but a region here, as shown by the two symmetrical triangles in Figure 9.

5. Regulation of Internal Repulsion

From the differential equation in (24), the types of equilibrium can be summarized into three cases. We also confirmed that the equilibrium of Case 3, such that

$$\begin{aligned} \mathbf{L}^* &\neq \mathbf{0}, \\ \mathbf{x}^* &\notin \text{Null}(\mathbf{L}^* \otimes \mathbf{I}_n), \\ \mathbf{w}^* &\neq \mathbf{0}, \end{aligned} \quad (30)$$

is the desirable solution to the suboptimal roadmap problem. To construct the appropriate roadmap, the Laplacian matrix must not be a zero matrix. To investigate this issue, the *internal repulsion* is defined as the quantitative information of the Laplacian matrix.

Definition 9 (internal repulsion). The *internal repulsion* P is the sum of the repulsion factors between neighbor nodes in the graph. It can be defined as the *entry-wise 1-norm of the Laplacian matrix* as follows:

$$P = \frac{1}{2} \|\mathbf{L}\|_1 = \frac{1}{2} \|\text{vec}(\mathbf{L})\|_1 = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N |L_{ij}|. \quad (31)$$

In the Laplacian matrix, the diagonal element is the sum of the row elements except itself, such that $|L_{ii}| = \sum_{i=1, i \neq j}^N |L_{ij}|$. Considering (23), the internal repulsion can be represented as follows:

$$\begin{aligned} P &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N |L_{ij}| = \frac{1}{2} \sum_{i=1}^N 2 \sum_{j=1, i \neq j}^N |g(d_{ij})| \\ &= \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(d_{ij}). \end{aligned} \quad (32)$$

Equilibrium in Case 1 can occur when the neighbor radius is small in comparison with the volume of the free configuration space, such that $\mu(\mathbf{Q}_{free}) > N \cdot \mu(\mathbf{B}(\mathbf{q}_i, r_\eta))$. Thus, whether the Laplacian matrix is a zero matrix is intimately related to the volume of the free configuration space $\mu(\mathbf{Q}_{free})$ and neighbor radius r_η . Therefore, to determine the properties of convergence of the algorithm, we observe the steady-state value of internal repulsion with respect to various neighbor radii.

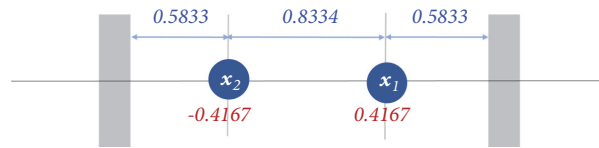


FIGURE 7: State of equilibrium for Example 8.

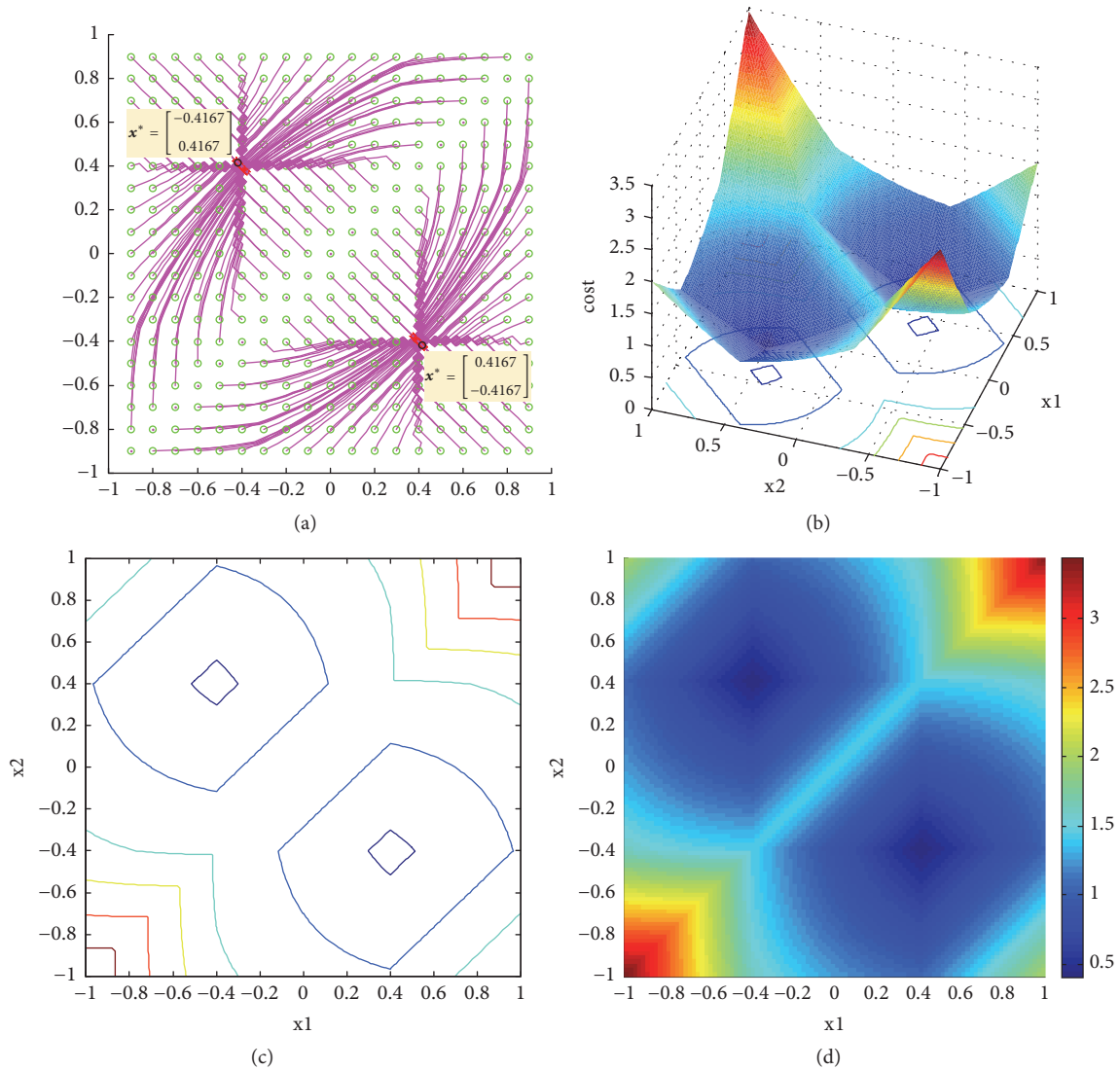


FIGURE 8: Phase portrait of the optimization process of Example 8. (a) State trajectories for each initial condition. (b) Cost graph of function H . (c) Contour graph of the cost function. (d) Cost graph.

Figure 10 shows the maximum and minimum values of internal repulsion in the steady state for various neighbor radii. Figure 11 shows the structure of the roadmap in the steady state for some cases. In the internal repulsion graph, the maximum and minimum values are identical when the neighbor radius is smaller than 57. In this case, the internal repulsion and the nodes remain in a certain state, which means that the structure and the state of the roadmap converge to equilibrium. However, when the internal repulsion converges to zero, in the case of a neighbor radius smaller

than 40, this means that the roadmap does not encompass the entire free configuration space, as shown in Figures 11(a) and 11(b). When the neighbor radius is 47, as shown in Figure 11(c), the structure of the roadmap encompasses the entire free configuration space well because the state of the roadmap converges with the appropriate internal repulsion, such as in the equilibrium of Case 3. When the neighbor radius is larger than 57, the maximum and minimum values are different. This means that the state of the nodes and the internal repulsion oscillate without converging. In this

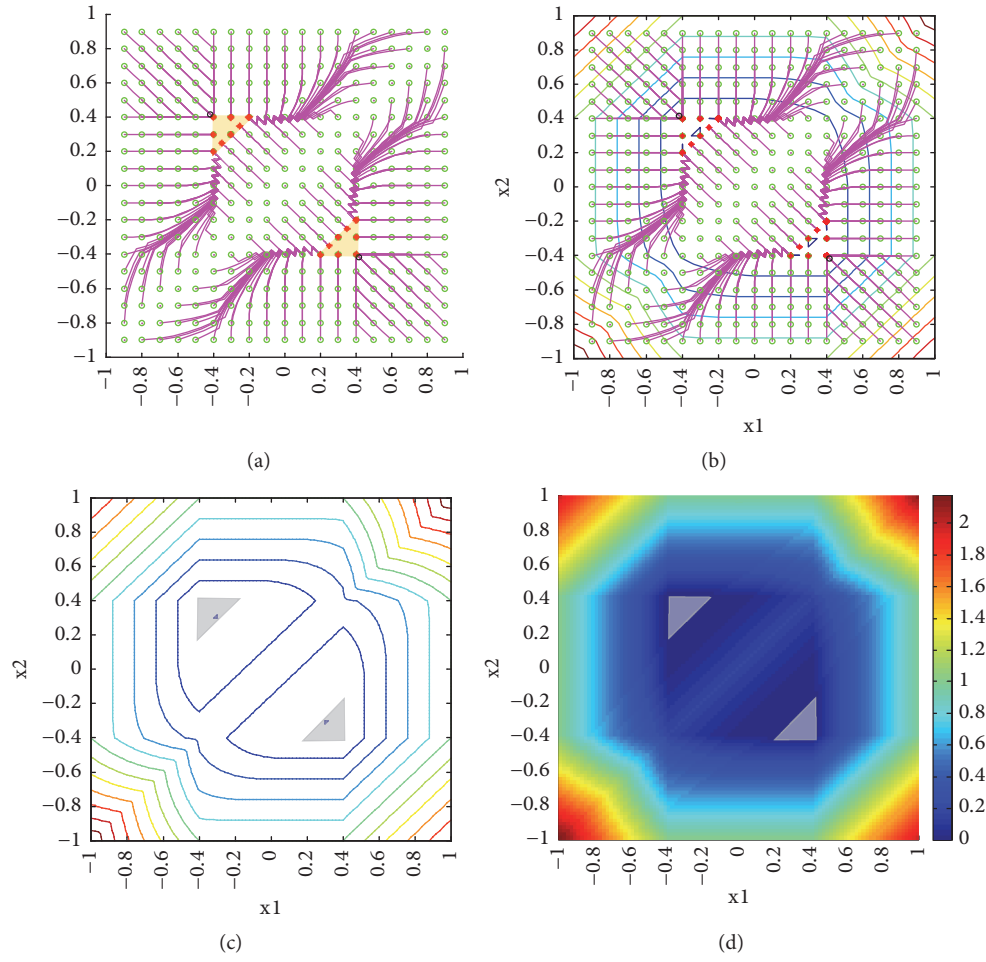


FIGURE 9: Phase portrait of the optimization process of Example 8 for $r_\eta = 0.6$. (a) State trajectories for each initial condition. (b) State trajectories and contour graph. (c) Contour graph. (d) Cost graph of function H .

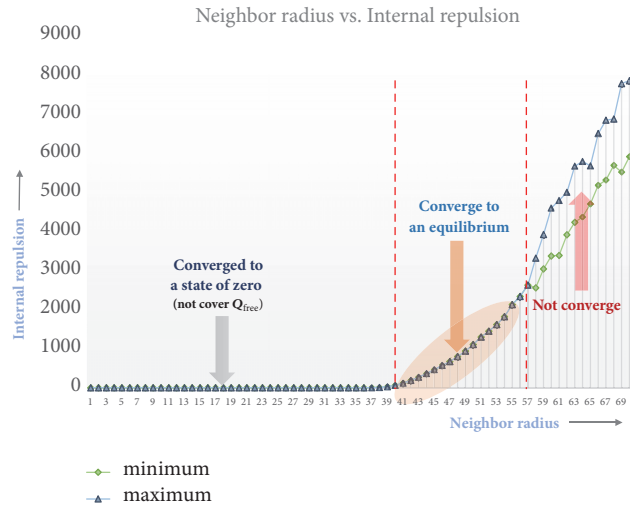


FIGURE 10: Graph of internal repulsion with respect to neighbor radius r_η .

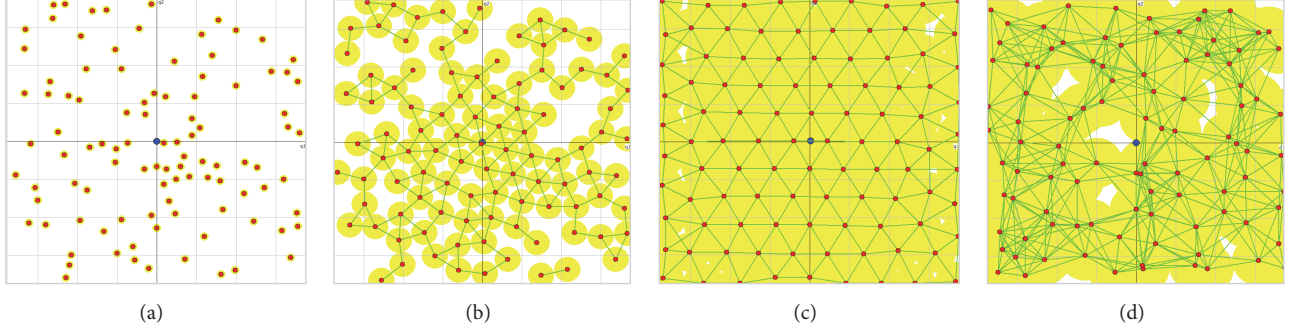


FIGURE 11: States of roadmap after 100 iterations. (a) $r_\eta = 10$ (no action); (b) $r_\eta = 30$ (ill-structured graph); (c) $r_\eta = 47$ (well-structured graph); (d) $r_\eta = 60$ (does not converge).

case, the algorithm cannot properly construct the roadmap because the state does not converge to equilibrium, as shown in Figure 11(d).

The deformation of the configuration space can induce changes in the volume of the free configuration space. When the volume of the free configuration space changes, the roadmap may no longer be constructed properly because the neighbor radius may be large or small relative to the changed volume of the free configuration space. Thus, the parameter of the neighbor radius should be adjusted with respect to the volume of the free configuration space. Generally, the volume of the free configuration space is difficult to calculate. However, the internal repulsion is highly related to it and is computable by executing the algorithm. With the regulation of internal repulsion by a desirable value, the roadmap can be constructed stably regardless of the volume of the free configuration space. To regulate internal repulsion, the controller that adjusts the neighbor radius should satisfy the following conditions.

Remark 10 (conditions for internal repulsion regulation). If the repulsion function and the regulation controller meet the following conditions for $\mathbf{L} \neq \mathbf{0}$, the *internal repulsion regulation* can be achieved. See [Section A.5] for proof.

(1) **Repulsion function condition:**

$$\frac{\partial g(r_\eta, \rho_{ij})}{\partial r_\eta} > 0 \quad (\exists \rho_{ij}, \rho_{ij} < r_\eta), \quad (33)$$

where r_η is added to the repulsion function as a variable because it is the control value in internal repulsion regulation, and ρ_{ij} is the expectation of neighbor distance, i.e., $\rho_{ij} = \mathbf{E}[d_{ij}]$.

(2) **Regulation controller condition:**

$$r_\eta = G_{IRR} \int e_P dt + r_\eta(t_0), \quad (e_P = P_d - P_\rho), \quad (34)$$

where $G_{IRR} > 0$ is controller gain, P_d is the desired internal repulsion value, and P_ρ is the internal repulsion by the expectation of neighbor distance, i.e., $P_\rho = \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta, \mathbf{E}[d_{ij}])$.

In the suboptimal roadmap algorithm, the gradient of the repulsion function satisfies *Condition (1)* because it is calculated from (21) as follows:

$$\begin{aligned} \frac{\partial g}{\partial r_\eta} &= \begin{cases} \frac{4}{d_{ij}} \frac{\sqrt{\pi/4}^{n-1}}{\Gamma(1 + (n-1)/2)} r_\eta \sqrt{r_\eta^2 - d_{ij}^2}^{n-3} & (0 < d_{ij} < r_\eta) \\ 0 & \text{otherwise.} \end{cases} \quad (35) \end{aligned}$$

Thus, the internal repulsion can be regulated to the desired value using the controller of *Condition (2)*. Such a controller in (34) can be implemented as a discrete-time system as follows:

$$\begin{aligned} r_\eta[k] &= G_{IRR} \\ &\cdot \sum_{l=k_0}^k \varepsilon \cdot \left(P_d - \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta[l], \bar{\rho}_{ij}[l]) \right) \\ &+ r_\eta[k_0], \end{aligned} \quad (36)$$

where ε is the sampling time or step size of the discrete-time system, and $\bar{\rho}_{ij}[k]$ is the estimation of ρ_{ij} that can be calculated as the average of d_{ij} for T_R intervals as $\bar{\rho}_{ij}[k] = (1/(T_R + 1)) \sum_{j=k-T_R}^k d_{ij}[j]$.

Figure 12 shows the efficacy of internal repulsion regulation in the case of a sudden change in free configuration space. At the initial condition, no obstacle exists and the free configuration space ratio $\mu(\mathbf{Q}_{free})/\mu(\mathbf{Q})$ is 100%. After the process reaches the state of equilibrium, some obstacles are added so that the volume of the free configuration space is reduced to 69.3%. The graphs of the internal repulsion for this situation are depicted in Figure 12 with respect to the time series. The green line represents the fixed neighbor radius and the blue line corresponds to the internal repulsion regulation. As shown in the graph, if no obstacles exist in the configuration space, both roadmaps converge to a similar structure. However, after the addition of obstacles into the configuration space, the roadmap with the fixed neighbor radius cannot converge and exhibits unstable oscillation, as depicted by the green line. However, the blue line graph shows

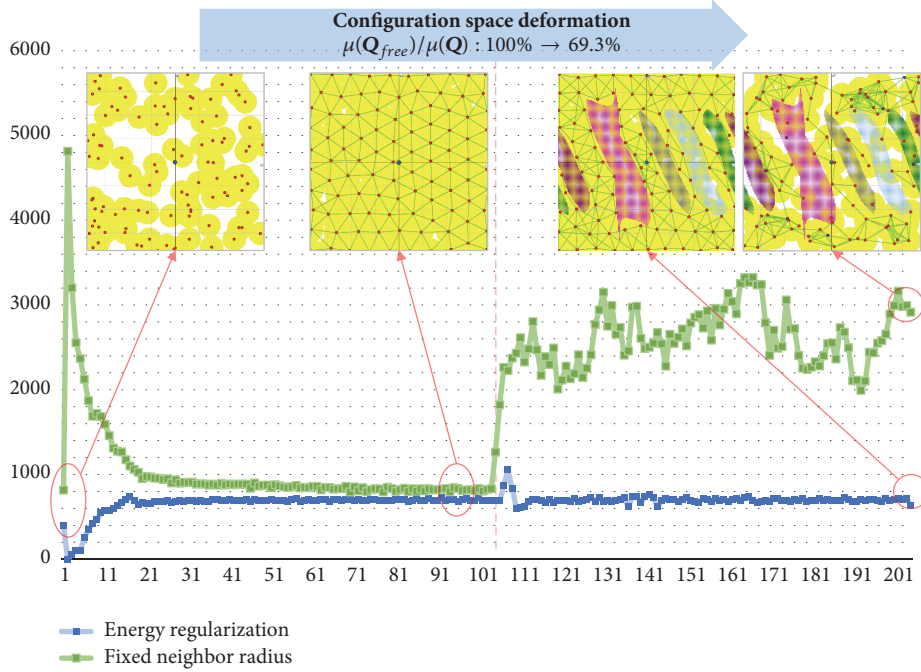


FIGURE 12: Graphs of internal repulsion for a change in the configuration space with a fixed neighbor radius ($r_\eta = 50$) and with internal repulsion regulation ($P_d = 700$).

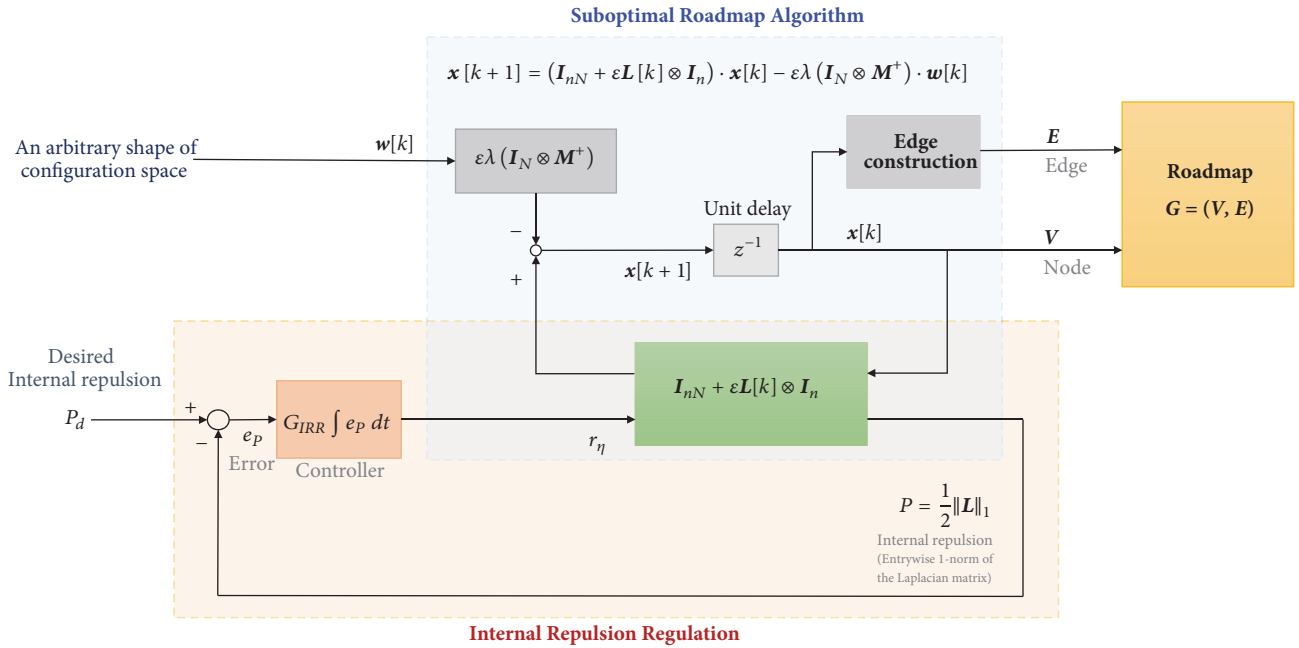


FIGURE 13: Overall structure of the suboptimal roadmap algorithm with internal repulsion regulation.

that the roadmap converges to equilibrium by regulating the internal repulsion uniformly regardless of the volume of the free configuration space.

Figure 13 shows an overall diagram of the optimization algorithm with internal repulsion regulation. This system features two types of differential equations. One is the differential equation used to maximize coverage that can construct

an appropriate graph against deformations in the configuration space, and the other is that for internal repulsion regulation of the roadmap regardless of the volume of the free configuration space. With these differential equations, the suboptimal roadmap algorithm can construct the proper roadmap graph that covers arbitrary shapes of the free configuration space.

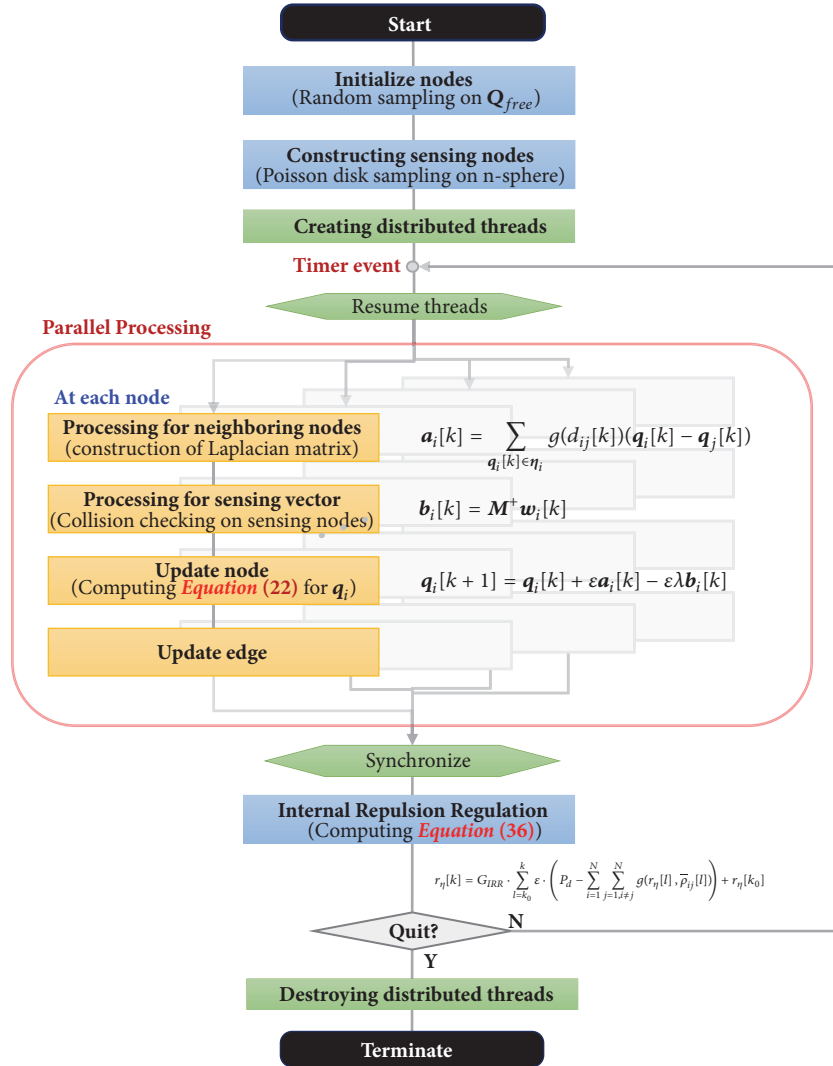


FIGURE 14: Flowchart for processing the suboptimal roadmap algorithm with internal repulsion regulation.

The suboptimal roadmap algorithm can be optimized by dividing its complex procedures using parallel computing techniques, such as multithreading or GPU processing. Thus, (22) is computed concurrently at each node. Figure 14 shows the flowchart for processing the suboptimal roadmap algorithm with internal repulsion regulation. The four procedures in the red border are performed in parallel at each node.

6. Case Study

To verify the effectiveness of the proposed method, we applied the suboptimal roadmap algorithm to a robot system in a car manufacturing assembly line. The target robot was the MPX3500, a painting robot system of the Yaskawa Electric Corporation (YEC) [24]. As shown in Figure 15, the MPX3500 robot had six axes; the S, L, and U axes rendered translation motions of the end effector. The other axes—R, B, and T—provided orientation motions. In conventional industrial robots, these axes are located around the end effector. In this application, only three axes—the S, L, and

U axes—are considered as the orientation-related motions of the end effector do not affect motion related to collisions. Hence, the configuration space of the MPX3500 was a 3D space, and the constructed roadmap could be visualized with a geometrical representation.

The first step in applying the motion-planning algorithm is to create the collision-detection function $s(\mathbf{q})$. We implemented it using the *oriented bounding box* (OBB) method, which yields the best efficiency in terms of both accuracy and computation time [38]. The suboptimal roadmap algorithm was applied to plan the motion of the MPX3500 robot in the automobile manufacturing line. Figure 16 shows the simulation environment, where a model of a car and some obstacles were placed. To observe the performance of the constructed roadmap with respect to the number of nodes N , we varied N from 50 to 300 and set the step size and the Lagrange multiplier to $\epsilon = 0.01$ and $\lambda = 25$, respectively. The number of sensing nodes and the radius of the sensing node were set to $n_s = 100$ and $r_s = 5$, respectively, and the desired internal repulsion was set to $P_d = 100$.

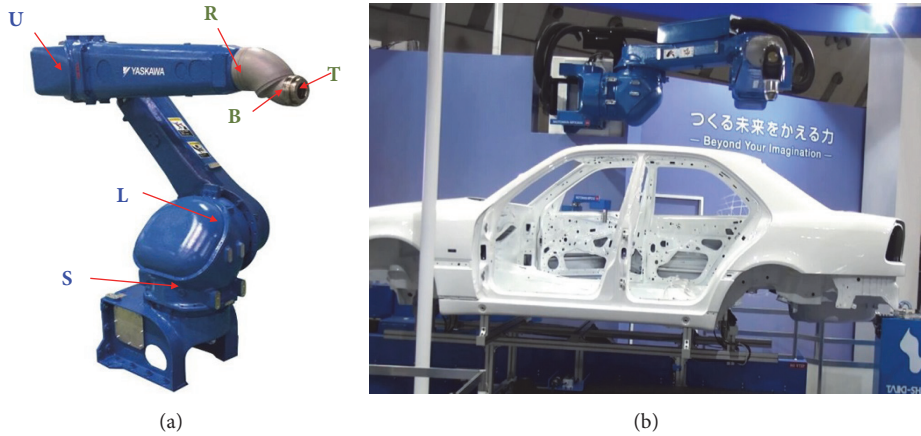


FIGURE 15: YEC-MPX3500 painting robot system [24]. (a) MPX3500 robot. (b) Scene of its operation at an automobile painting line.

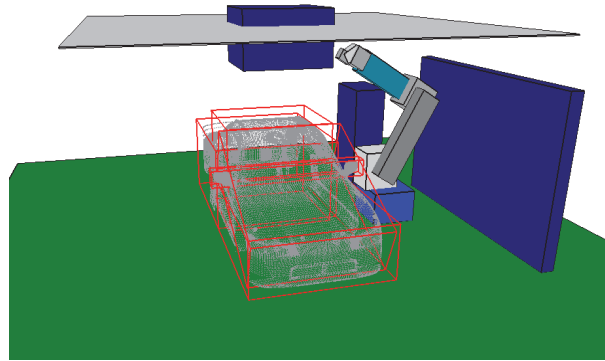


FIGURE 16: Workspace of an automobile manufacturing line. The car model and some obstacles are placed in the simulation environment.

Figure 17 shows the state of the roadmap for the iterations of the suboptimal roadmap algorithm for 150 nodes. Figure 17(a) displays the initial state of the roadmap, in which 150 nodes were scattered randomly in the configuration space. Figures 17(b) and 17(c) show the states of the roadmap after the first and the fifth iterations, respectively. These states are the initial steps of the execution of the algorithm. Figures 17(d), 17(e), and 17(f) show the states of the roadmap after the 25th, 50th, and 100th iterations, respectively. After 25 iterations, no considerable changes were observed in the state of the roadmap, which indicates that the iteration process might have arrived at a minimum state.

Figure 18 shows roadmaps of the suboptimal roadmap algorithm with respect to the number of nodes after 100 iterations. As shown in the figures, the appearances of the roadmap graphs are similar. However, the density of nodes is different with respect to the number of nodes. Although the quality of the results of planning depends on the density of nodes, near-optimal solutions were obtained even for a low density of nodes because the nodes were dispersed evenly.

Figures 19 and 20 show examples of the results of motion planning using the optimal roadmap, PRM*, and RRT algorithms in the test environment. Figure 19(a) shows the initial motion and (b) shows the goal motion of the robot.

(c) and (d) present the compared paths obtained by the three methods in the configuration space. Figure 20 shows the three trajectories of the robot's motion according to the planned path in Figure 19 in the workspace. As shown in the figures, the paths obtained by the suboptimal roadmap method were the shortest for the two workspaces; those obtained by the PRM* algorithm were the next shortest, and the RRT algorithm offered suboptimal results, such as paths involving detours, as confirmed in Figures 19(c) and 19(d).

Figure 21 shows comparison graphs of the results of motion planning using different algorithms—the suboptimal roadmap, PRM*, and RRT—with respect to the number of nodes N . Using roadmaps with varying number of nodes, each calculation was performed 10 times to plan for 100 pairs of motions of arbitrary initial points and goals. For all roadmaps, the results of the suboptimal roadmap algorithm yielded the lowest value of accumulated length of the planned path. This indicates that the paths obtained using the proposed method were closest to the optimal results for the given motion-planning problem. Figure 22 shows the graph of the cumulative length of the paths obtained with the suboptimal roadmap and the PRM* algorithm with respect to the number of nodes. As shown in the figure, the graph of the suboptimal roadmap algorithm is always lower than that of the PRM*. This indicates that better motion-planning results

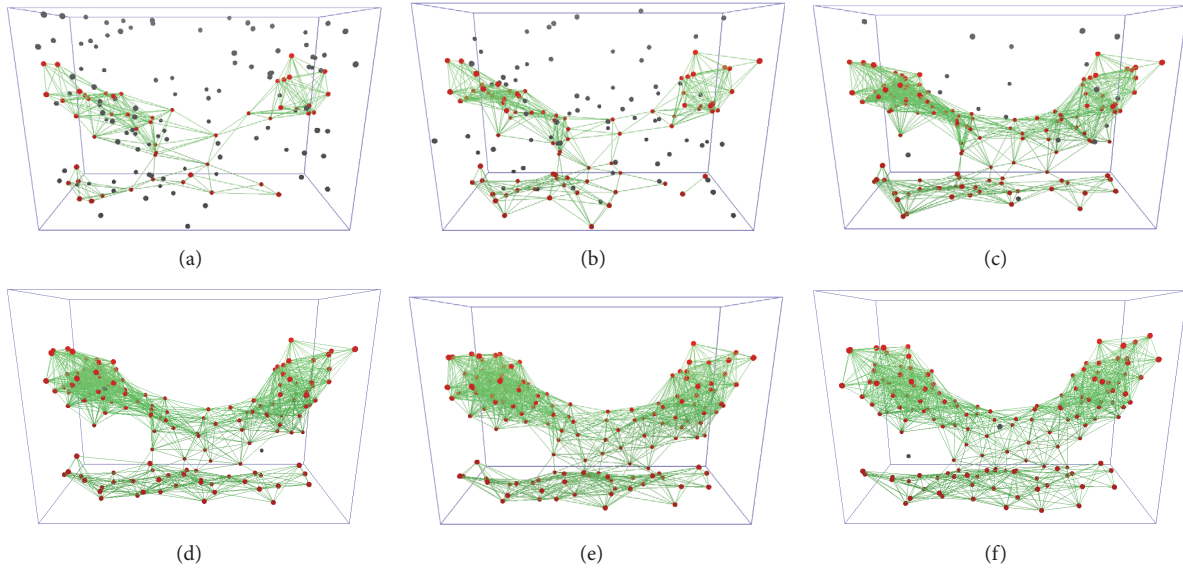


FIGURE 17: Operating procedures of the suboptimal roadmap algorithm in the configuration space (150 nodes). (a) Initial state. (b) 1st iteration. (c) 5th iteration. (d) 25th iteration. (e) 50th iteration. (f) 100th iteration.

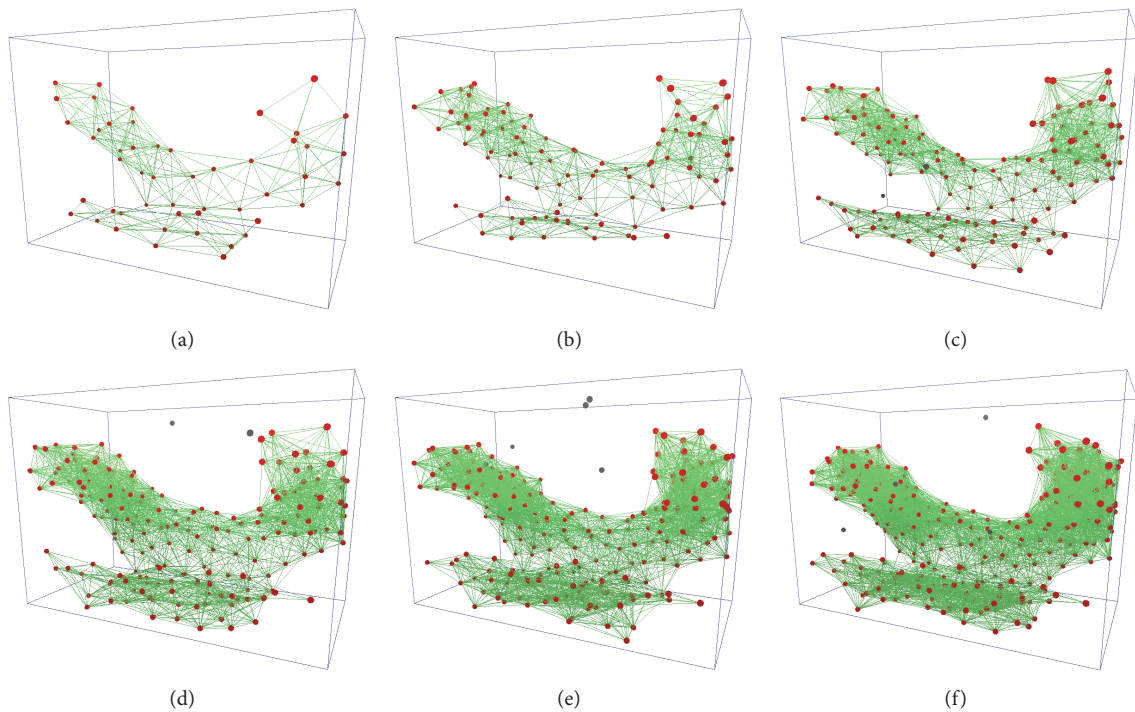


FIGURE 18: Roadmaps constructed by the suboptimal roadmap algorithm with respect to the number of nodes after 100 iterations. (a) $N = 50$; (b) $N = 100$; (c) $N = 150$; (d) $N = 200$; (e) $N = 250$; (f) $N = 300$.

can be obtained with the suboptimal roadmap algorithm with a small number of nodes. Moreover, the ratios of decrement of the suboptimal roadmap graphs were smaller than those of PRM*. This means that the motion-planning performance of the suboptimal roadmap algorithm was less sensitive to the number of nodes than current sampling-based methods.

7. Conclusion

This paper considered the problem of constructing a proper roadmap graph that fully encompasses the arbitrary morphologies of the free configuration space. To this end, the coverage of the graph was defined as a performance index on the optimality of a graph constructed by the sampling-based

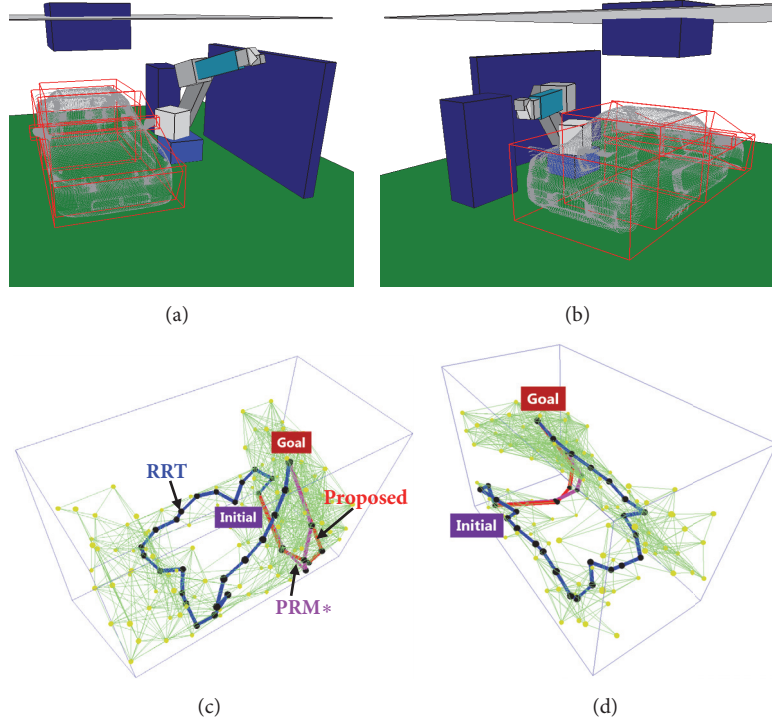


FIGURE 19: Comparison of the results of motion planning in the test environment. (a) Initial configuration. (b) Goal configuration. (c), (d) Motion paths obtained by the proposed method (red), PRM* (magenta), and RRT (blue) algorithms in the configuration space.

algorithm. We then proposed an optimization algorithm that can maximize graph coverage in the configuration space. Because of the computational complexity of coverage in practice, the proposed algorithm was designed using the concept of the suboptimal roadmap that minimizes the upper bound of the volume of intersection of the graph-covering region. Equilibrium conditions for the optimization algorithm were derived from the iteration equation. Among the results, we found the equilibrium condition that can construct an appropriate roadmap in an arbitrary configuration space. Furthermore, a method for regulating internal repulsion was proposed to utilize the suboptimal roadmap algorithm regardless of changes to the volume of the free configuration space. The convergence of the algorithm was found to be highly relevant to the volume of the free configuration space, accompanied by the number of nodes and the neighboring radius. Thus, to represent these relations quantitatively, internal repulsion was defined as the sum of repulsion factors between neighboring nodes in the graph. Further, we proposed an integrator-type controller that adjusts the neighboring radius to regulate internal repulsion according to the desired value. The feasibility of the proposed method was confirmed through a case study involving a robot. We applied the method to an industrial robot used in car manufacturing assembly lines. The results of the case study confirmed that the roadmap graph obtained by the proposed algorithm can produce more optimal results for paths of motion of the robot in the simulation environment.

Appendix

A.

A.1. Upper Bound of Intersection Volume $I(\mathcal{B})$

Theorem A.1 (upper bound of intersection volume). *In a family of sets $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_N\}$ composed of N sets, the upper bound of the intersection volume $I(\mathcal{B})$ can be represented by the multiplication of the constant κ and the sum of the intersection volume of two sets as follows:*

$$I(\mathcal{B}) \leq \kappa \cdot \left(\sum_{1 \leq i < j \leq N} \mu(\mathbf{B}_i \cap \mathbf{B}_j) \right), \quad (\text{A.1})$$

where $\kappa = 2/N(N-1) \cdot \sum_{i=2}^N \left(\prod_{j=2}^i \binom{N}{j} \right)$ is a finite constant related to number of sets N .

Proof. Let the sum of intersection volumes of i elements in \mathcal{B} be

$$S_i = \sum_{1 \leq j_1 < j_2 < \dots < j_i \leq N} \mu(\mathbf{B}_{j_1} \cap \mathbf{B}_{j_2} \cap \dots \cap \mathbf{B}_{j_i}); \quad (\text{A.2})$$

subsequently, the intersection volume $I(\mathcal{B}) = \sum_{i=2}^N (-1)^i S_i$ meets the following condition:

$$I(\mathcal{B}) = \sum_{i=2}^N (-1)^i S_i \leq \sum_{i=2}^N S_i. \quad (\text{A.3})$$

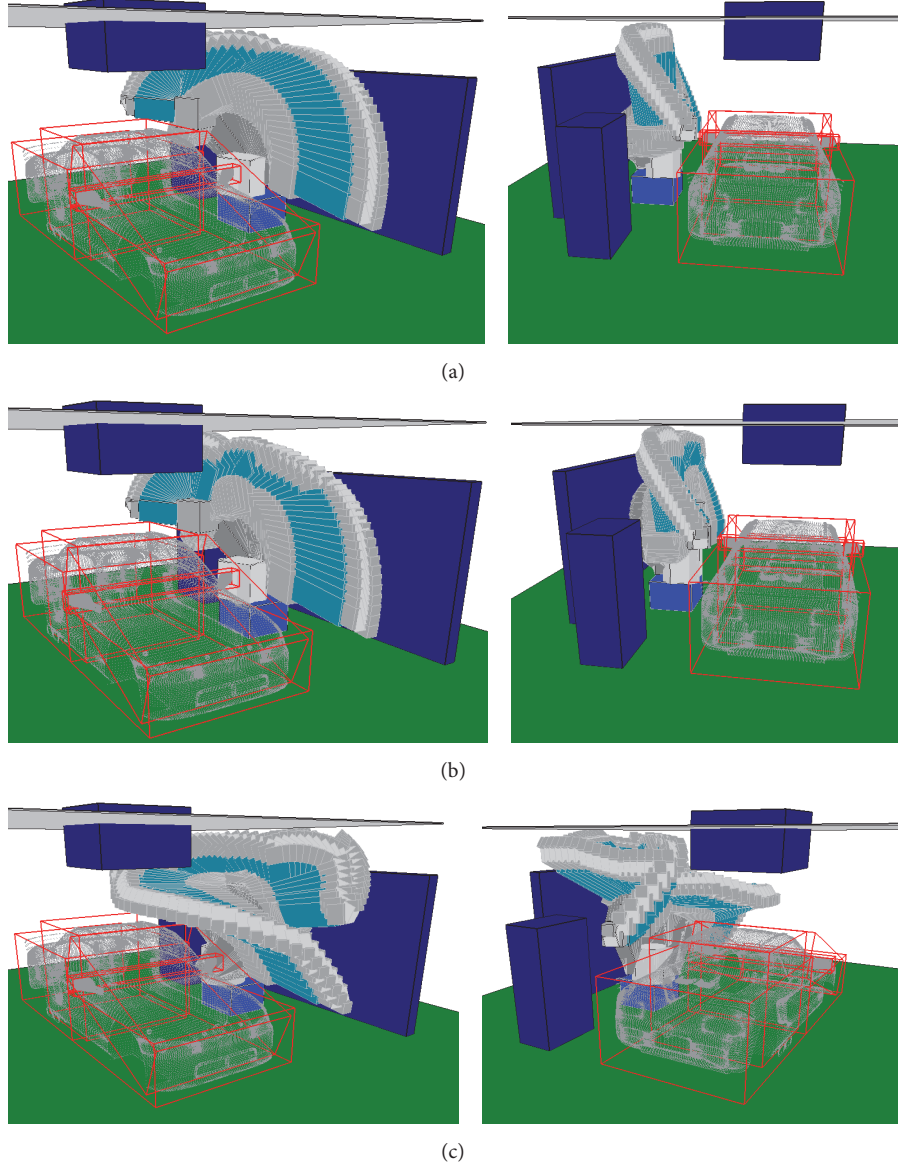


FIGURE 20: Trajectories of the robot's motion in the workspace with respect to the planning results of Figure 19. (a) Proposed method. (b) PRM*. (c) RRT.

Because $S_{i+1} \leq \binom{N}{i+1} \cdot S_i$ by Lemma A.2, the following inequality is met for S_i :

$$S_i \leq \binom{N}{2}^{-1} \left(\prod_{j=2}^i \binom{N}{j} \right) \cdot S_2, \quad (2 \leq i \leq N); \quad (\text{A.4})$$

further, the following inequality is also satisfied for $\sum_{i=2}^N S_i$ ($\geq I(\mathcal{B})$):

$$\begin{aligned} \sum_{i=2}^N S_i &\leq \binom{N}{2}^{-1} \left(\prod_{j=2}^2 \binom{N}{j} \right) \cdot S_2 + \binom{N}{2}^{-1} \\ &\cdot \left(\prod_{j=2}^3 \binom{N}{j} \right) \cdot S_2 + \dots + \binom{N}{2}^{-1} \left(\prod_{j=2}^N \binom{N}{j} \right) \end{aligned}$$

$$\cdot S_2 = \binom{N}{2}^{-1}$$

$$\cdot \left\{ \prod_{j=2}^2 \binom{N}{j} + \prod_{j=2}^3 \binom{N}{j} + \dots + \prod_{j=2}^N \binom{N}{j} \right\} \cdot S_2$$

$$= \frac{2}{N(N-1)}$$

$$\cdot \sum_{i=2}^N \left(\prod_{j=2}^i \binom{N}{j} \right) \cdot \left(\sum_{1 \leq i < j \leq N} \mu(\mathbf{B}_i \cap \mathbf{B}_j) \right).$$

(A.5)

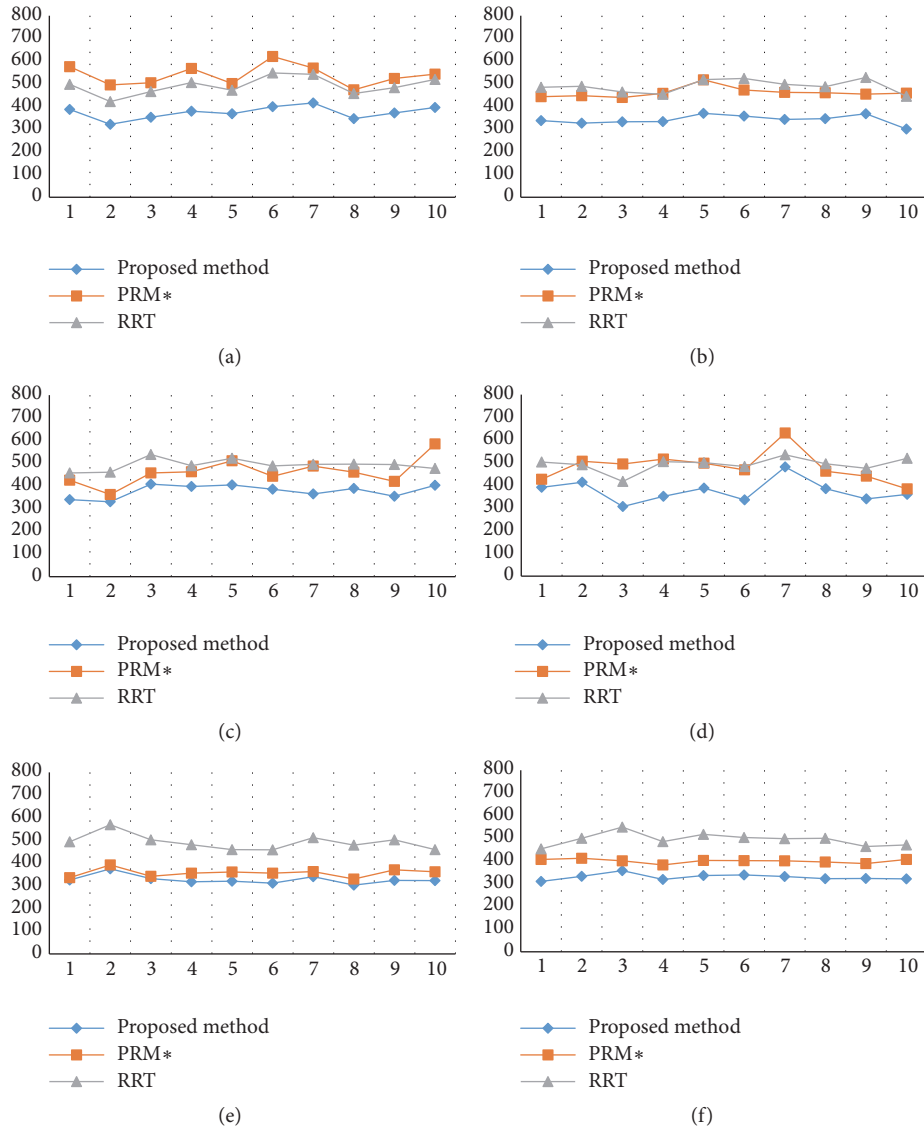


FIGURE 21: Results of comparison of path qualities obtained using the proposed method, PRM*, and RRT algorithms with respect to the number of nodes. (a) $N = 50$; (b) $N = 100$; (c) $N = 150$; (d) $N = 200$; (e) $N = 250$; (f) $N = 300$.

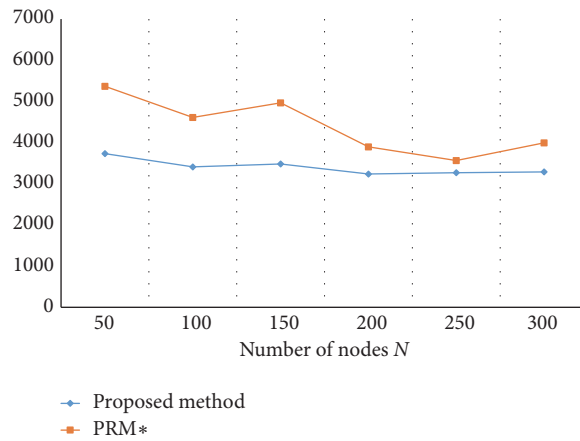


FIGURE 22: Results of evaluation of the total cumulative path of the proposed method and PRM* algorithm with respect to the number of nodes in the test environment.

Hence, the intersection volume $I(\mathcal{B})$ satisfies the inequality of (A.1). \square

Lemma A.2. S_{i+1} meets the following inequality for S_i :

$$S_{i+1} \leq \binom{N}{i+1} \cdot S_i. \quad (\text{A.6})$$

Proof. Let the k -th intersection set in S_i be $\mathbf{C}_i(k) = \mathbf{B}_{j_1(k)} \cap \mathbf{B}_{j_2(k)} \cap \dots \cap \mathbf{B}_{j_i(k)}$ and its volume be $\mu(\mathbf{C}_i(k))$; therefore, S_i can be expressed as follows:

$$\begin{aligned} S_i &= \sum_{1 \leq j_1 < j_2 < \dots < j_i \leq N} \mu(\mathbf{B}_{j_1} \cap \mathbf{B}_{j_2} \cap \dots \cap \mathbf{B}_{j_i}) \\ &= \sum_{k=1}^{\binom{N}{i}} \mu(\mathbf{C}_i(k)). \end{aligned} \quad (\text{A.7})$$

In S_{i+1} , as $\mathbf{C}_{i+1}(k) = \mathbf{C}_i(l_k) \cap \mathbf{B}_{j_{i+1}(k)}$ for $l_k \in \{1, \dots, \binom{N}{i}\}$, $\mathbf{C}_{i+1}(k) \subseteq \mathbf{C}_i(l_k)$ can be represented as

$$\begin{aligned} \mu(\mathbf{C}_{i+1}(k)) &= \alpha_{i+1}(k) \cdot \mu(\mathbf{C}_i(l_k)) \\ &\text{for } 0 \leq \alpha_{i+1}(k) \leq 1. \end{aligned} \quad (\text{A.8})$$

Thus, S_{i+1} can be expressed by the sum of $\binom{N}{i+1}$ terms of $\alpha_{i+1} \cdot \mu(\mathbf{C}_i)$ as follows:

$$S_{i+1} = \sum_{k=1}^{\binom{N}{i+1}} \mu(\mathbf{C}_{i+1}(k)) = \sum_{k=1}^{\binom{N}{i+1}} \alpha_{i+1}(k) \cdot \mu(\mathbf{C}_i(l_k)). \quad (\text{A.9})$$

Let $\beta_{i+1}(l)$ be the sum of $\alpha_{i+1}(k)$ that is applied to the same $\mu(\mathbf{C}_i(l))$; therefore, S_{i+1} can be reformulated by the sum of $\binom{N}{i}$ terms of $\beta_{i+1} \cdot \mu(\mathbf{C}_i)$ as follows:

$$\begin{aligned} S_{i+1} &= \sum_{k=1}^{\binom{N}{i+1}} \alpha_{i+1}(k) \cdot \mu(\mathbf{C}_i(k)) \\ &= \sum_{l=1}^{\binom{N}{i}} \beta_{i+1}(l) \cdot \mu(\mathbf{C}_i(l)), \end{aligned} \quad (\text{A.10})$$

where $\sum_{k=1}^{\binom{N}{i+1}} \alpha_{i+1}(k) = \sum_{k=1}^{\binom{N}{i}} \beta_{i+1}(k)$.

S_{i+1} can be represented by the inner product of two vectors $\boldsymbol{\beta}_{i+1} = [\beta_{i+1}(k)] \in \mathbf{R}^{\binom{N}{i}}$ and $\boldsymbol{\mu}_i = [\mu(\mathbf{C}_i(k))] \in \mathbf{R}^{\binom{N}{i}}$ as follows:

$$S_{i+1} = \sum_{l=1}^{\binom{N}{i}} \beta_{i+1}(l) \cdot \mu(\mathbf{C}_i(l)) = \boldsymbol{\beta}_{i+1}^T \cdot \boldsymbol{\mu}_i. \quad (\text{A.11})$$

Because $\beta_{i+1}(k), \mu(\mathbf{C}_i(k)) \geq 0$, $S_{i+1} = \boldsymbol{\beta}_{i+1}^T \cdot \boldsymbol{\mu}_i \leq \|\boldsymbol{\beta}_{i+1}\|_1 \cdot \|\boldsymbol{\mu}_i\|_1$ is satisfied by the Cauchy-Schwarz inequality. $\|\boldsymbol{\mu}_i\|_1 = \sum_{k=1}^{\binom{N}{i}} \mu(\mathbf{C}_i(k)) = S_i$ and

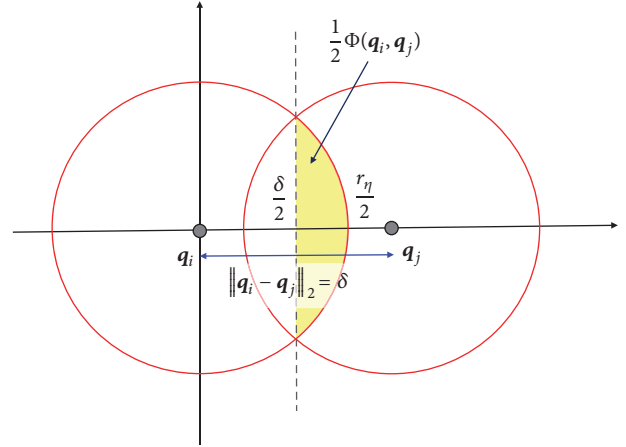


FIGURE 23: Two spheres and their region of intersection.

$$\begin{aligned} \|\boldsymbol{\beta}_{i+1}\|_1 &= \sum_{k=1}^{\binom{N}{i+1}} \beta_{i+1}(k) = \sum_{k=1}^{\binom{N}{i+1}} \alpha_{i+1}(k) \leq \binom{N}{i+1}, \\ &\quad (\because 0 \leq \alpha_{i+1}(k) \leq 1). \end{aligned} \quad (\text{A.12})$$

Hence, the inequality in (A.6) is satisfied as follows:

$$S_{i+1} = \boldsymbol{\beta}_{i+1}^T \cdot \boldsymbol{\mu}_i \leq \|\boldsymbol{\beta}_{i+1}\|_1 \cdot \|\boldsymbol{\mu}_i\|_1 \leq \binom{N}{i+1} \cdot S_i. \quad (\text{A.13})$$

\square

A.2. Volume of Region of Intersection between n -Dimensional Hyperspheres [39]. The volume of the n -dimensional hypersphere can be calculated by integrating the volume of an $n-1$ -dimensional hypersphere along a certain axis. As shown in Figure 23, the region of intersection between spheres can be calculated by integrating the volume of the $n-1$ -dimensional sphere from $d_{ij}/2$ ($d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|_2$) to $r_\eta/2$ along a certain axis. The volume of the $n-1$ -dimensional hypersphere of radius r is

$$V_{n-1}(r) = \frac{\sqrt{\pi}^{n-1}}{\Gamma(1 + (n-1)/2)} r^{n-1}, \quad (\text{A.14})$$

where $\Gamma(x)$ is the gamma function [40].

To obtain the volume of the n -dimensional hypersphere of radius $r_\eta/2$, the $n-1$ hypersphere is integrated for radius $r = \sqrt{(r_\eta/2)^2 - t^2}$ on $-r_\eta/2 \leq t \leq r_\eta/2$ [39]. Thus, the integral of the yellow region in Figure 23 can be calculated by integrating the volume of the $n-1$ sphere as follows:

$$\begin{aligned} &\int V_{n-1}(r) dr \\ &= \int_{d_{ij}/2}^{r_\eta/2} \frac{\sqrt{\pi}^{n-1}}{\Gamma(1 + (n-1)/2)} \sqrt{\left(\frac{r_\eta}{2}\right)^2 - t^2}^{n-1} dt. \end{aligned} \quad (\text{A.15})$$

In Figure 23, the yellow area calculated by the integration is the half-region of the area of intersection; thus, the volume of intersection is twice the calculated volume.

$$\Phi(\mathbf{q}_i, \mathbf{q}_j) = \begin{cases} 2 \cdot \int_{d_{ij}/2}^{r_\eta/2} \frac{\sqrt{\pi}^{n-1}}{\Gamma(1 + (n-1)/2)} \sqrt{\left(\frac{r_\eta}{2}\right)^2 - t^2}^{n-1} dt & (0 \leq d_{ij} < r_\eta) \\ 0 & (d_{ij} \geq r_\eta) \end{cases} \quad (\text{A.16})$$

A.3. Gradient Estimation of the Collision-Detection Function. Function $s(\mathbf{q})$ checks for collisions in the workspace for configuration $\mathbf{q} \in \mathbf{Q}$. As $s(\mathbf{q})$ is a discontinuous function, its gradient cannot be calculated. The gradient of $s(\mathbf{q})$ can be estimated stochastically using the *response surface method* (RSM) [31–33].

In the RSM, for $\mathbf{q}_i \in \mathbf{Q}$, $\boldsymbol{\varphi} \in \mathbf{R}^n$, $s(\mathbf{q})$ is assumed to be a linear model in the neighborhood of \mathbf{q}_i as follows:

$$\hat{s}(\mathbf{q}) = \varphi_0 + \boldsymbol{\varphi}^T (\mathbf{q} - \mathbf{q}_i). \quad (\text{A.17})$$

The gradient of the function above, which is an estimated gradient of $s(\mathbf{q})$ at $\mathbf{q} = \mathbf{q}_i$ is

$$\hat{\nabla}s(\mathbf{q})\big|_{\mathbf{q}=\mathbf{q}_i} = \nabla\hat{s}(\mathbf{q}_i) = \boldsymbol{\varphi}. \quad (\text{A.18})$$

In the RSM algorithm, $\nabla\hat{s}(\mathbf{q}_i)$ can be estimated using p design points \mathbf{q}_m ($m = 1, \dots, p$) around node \mathbf{q}_i ; here, the n_s ($> n$) sensing nodes of \mathbf{q}_i correspond to the design points, i.e., $\mathbf{q}_m = \mathbf{q}_i + \Delta\mathbf{q}_m$ ($m = 1, \dots, n_s$). By stacking the $s(\mathbf{q})$ information of n_s design points, (A.17) can be represented in matrix form as follows:

$$\begin{aligned} \begin{bmatrix} s(\mathbf{q}_i + \Delta\mathbf{q}_1) \\ s(\mathbf{q}_i + \Delta\mathbf{q}_2) \\ \vdots \\ s(\mathbf{q}_i + \Delta\mathbf{q}_{n_s}) \end{bmatrix} &= \begin{bmatrix} \varphi_0 + \boldsymbol{\varphi}^T (\mathbf{q}_i + \Delta\mathbf{q}_1 - \mathbf{q}_i) \\ \varphi_0 + \boldsymbol{\varphi}^T (\mathbf{q}_i + \Delta\mathbf{q}_2 - \mathbf{q}_i) \\ \vdots \\ \varphi_0 + \boldsymbol{\varphi}^T (\mathbf{q}_i + \Delta\mathbf{q}_{n_s} - \mathbf{q}_i) \end{bmatrix} \\ &= \begin{bmatrix} 1 & \Delta\mathbf{q}_1^T \\ 1 & \Delta\mathbf{q}_2^T \\ \vdots & \vdots \\ 1 & \Delta\mathbf{q}_{n_s}^T \end{bmatrix} \begin{bmatrix} \varphi_0 \\ \boldsymbol{\varphi} \end{bmatrix}. \end{aligned} \quad (\text{A.19})$$

Using the sensing vector $\mathbf{w}_i = [s(\mathbf{q}_i + \Delta\mathbf{q}_m)] \in \mathbf{R}^{n_s}$ at node \mathbf{q}_i , the parameter $\hat{\boldsymbol{\varphi}}_I = \begin{bmatrix} \varphi_0 \\ \boldsymbol{\varphi} \end{bmatrix} \in \mathbf{R}^{n+1}$ is estimated by the least-squares algorithm as follows [32]:

$$\hat{\boldsymbol{\varphi}}_I = (\mathbf{M}_I^T \mathbf{M}_I)^{-1} \mathbf{M}_I^T \mathbf{w}_i, \quad (\text{A.20})$$

where $\mathbf{M} = [\Delta\mathbf{q}_m^T] \in \mathbf{R}^{n_s \times n}$ and $\mathbf{M}_I = [\mathbf{1}_{n_s} \quad \mathbf{M}] \in \mathbf{R}^{n_s \times (n+1)}$.

Further, $d_{ij} \geq 0$ and $\Phi(\mathbf{q}_i, \mathbf{q}_j) = 0$ for $d_{ij} \geq r_\eta$; thus, the volume of the region of intersection between the n -dimensional hyperspheres can be summarized as follows:

The matrix $\mathbf{M}_I^T \mathbf{M}_I$ is

$$\begin{aligned} \mathbf{M}_I^T \mathbf{M}_I &= \begin{bmatrix} 1 & \cdots & 1 \\ & \mathbf{M}^T & \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ \mathbf{M} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{m=1}^{n_s} 1 & \sum_{m=1}^{n_s} \Delta\mathbf{q}_m^T \\ \sum_{m=1}^{n_s} \Delta\mathbf{q}_m & \mathbf{M}^T \mathbf{M} \end{bmatrix}; \end{aligned} \quad (\text{A.21})$$

if $\sum_{m=1}^{n_s} \Delta\mathbf{q}_m = \mathbf{0}$, then $\mathbf{M}_I^T \mathbf{M}_I = \begin{bmatrix} n_s & \mathbf{0}^T \\ \mathbf{0} & \mathbf{M}^T \mathbf{M} \end{bmatrix}$; therefore, (A.21) becomes

$$\begin{aligned} \hat{\boldsymbol{\varphi}}_I &= \begin{bmatrix} \hat{\varphi}_0 \\ \hat{\boldsymbol{\varphi}} \end{bmatrix} = \begin{bmatrix} n_s & \mathbf{0}^T \\ \mathbf{0} & \mathbf{M}^T \mathbf{M} \end{bmatrix}^{-1} \begin{bmatrix} 1 & \cdots & 1 \\ & \mathbf{M}^T & \end{bmatrix} \mathbf{w}_i \\ &= \begin{bmatrix} n_s^{-1} \cdot \sum_{m=1}^{n_s} s(\mathbf{q}_i + \Delta\mathbf{q}_m) \\ (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{w}_i \end{bmatrix}. \end{aligned} \quad (\text{A.22})$$

Hence, the estimated gradient of $s(\mathbf{q})$ is

$$\hat{\boldsymbol{\varphi}} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{w}_i. \quad (\text{A.23})$$

A.4. Derivation of the Differential Equation for the Entire Optimization Variable $\mathbf{x} \in \mathbf{R}^{nN}$. Let the Lagrange multipliers λ_i be $\lambda_1 = \lambda_2 = \dots = \lambda_N = \lambda$; then, the decentralized optimization algorithm can be formulated as follows:

$$\begin{aligned} \mathbf{q}_i[k+1] &= \mathbf{q}_i[k] \\ &+ \varepsilon \sum_{\mathbf{q}_j[k] \in \boldsymbol{\eta}_i} g(d_{ij}[k]) (\mathbf{q}_i[k] - \mathbf{q}_j[k]) \\ &- \varepsilon \lambda \mathbf{M}^+ \mathbf{w}_i[k], \end{aligned} \quad (\text{A.24})$$

where d_{ij} is the distance between nodes $\mathbf{q}_i, \mathbf{q}_j$; $\boldsymbol{\eta}_i$ is the set of neighbors of \mathbf{q}_i ; and $g(d_{ij})$ is a function defined in (21).

Because $g(d_{ij}) = 0$ for $\mathbf{q}_j \notin \boldsymbol{\eta}_i$, (A.24) becomes

$$\begin{aligned} \mathbf{q}_i[k+1] &= \mathbf{q}_i[k] \\ &+ \varepsilon \sum_{j=1, i \neq j}^N g(d_{ij}[k]) (\mathbf{q}_i[k] - \mathbf{q}_j[k]) \\ &- \varepsilon \lambda \mathbf{M}^+ \mathbf{w}_i[k] = \mathbf{q}_i[k] \\ &+ \varepsilon \left(\left(\sum_{j=1, i \neq j}^N g(d_{ij}[k]) \right) \mathbf{q}_i[k] \right. \\ &\left. - \sum_{j=1, i \neq j}^N g(d_{ij}[k]) \mathbf{q}_j[k] \right) - \varepsilon \lambda \mathbf{M}^+ \mathbf{w}_i[k]. \end{aligned} \quad (\text{A.25})$$

Let the row vector \mathbf{L}_i be

$$\mathbf{L}_i = [L_{ij}] \in \mathbf{R}^{1 \times N}, \quad L_{ij} = \begin{cases} -g(d_{ij}) & i \neq j \\ \sum_{k=1, i \neq k}^N g(d_{ik}) & i = j, \end{cases} \quad (\text{A.26})$$

(A.25) can be represented for $\mathbf{x} = [\mathbf{q}_i] \in \mathbf{R}^{nN}$ as follows:

$$\begin{aligned} \mathbf{q}_i[k+1] &= \mathbf{q}_i[k] + \varepsilon (\mathbf{L}_i[k] \otimes \mathbf{I}_n) \mathbf{x}[k] \\ &- \varepsilon \lambda \mathbf{M}^+ \mathbf{w}_i[k], \end{aligned} \quad (\text{A.27})$$

where \otimes denotes the Kronecker product operator.

The equations of iteration performed at every node can be integrated such that

$$\begin{aligned} \begin{bmatrix} \mathbf{q}_1[k+1] \\ \mathbf{q}_2[k+1] \\ \vdots \\ \mathbf{q}_N[k+1] \end{bmatrix} &= \begin{bmatrix} \mathbf{q}_1[k] \\ \mathbf{q}_2[k] \\ \vdots \\ \mathbf{q}_N[k] \end{bmatrix} \\ &+ \varepsilon \left(\begin{bmatrix} \mathbf{L}_1[k] \\ \mathbf{L}_2[k] \\ \vdots \\ \mathbf{L}_N[k] \end{bmatrix} \otimes \mathbf{I}_n \right) \mathbf{x}[k] \quad (\text{A.28}) \\ &- \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \begin{bmatrix} \mathbf{w}_1[k] \\ \mathbf{w}_2[k] \\ \vdots \\ \mathbf{w}_N[k] \end{bmatrix} \end{aligned}$$

Let $\mathbf{w} = [\mathbf{w}_i] \in \mathbf{R}^{n \times N}$ be the stacked vector of N sensing vectors \mathbf{w}_i ; then, using the Laplacian matrix $\mathbf{L} = [\mathbf{L}_i] \in \mathbf{R}^{N \times N}$,

(A.28) can be reformulated for the entire variable $\mathbf{x} \in \mathbf{R}^{nN}$ as follows:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{x}[k] + \varepsilon (\mathbf{L}[k] \otimes \mathbf{I}_n) \mathbf{x}[k] \\ &- \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}[k] \\ &= (\mathbf{I}_{nN} + \varepsilon \mathbf{L}[k] \otimes \mathbf{I}_n) \mathbf{x}[k] \\ &- \varepsilon \lambda (\mathbf{I}_N \otimes \mathbf{M}^+) \mathbf{w}[k]. \end{aligned} \quad (\text{A.29})$$

A.5. Proof of Convergence For Internal Repulsion Regulation. This section proves Remark 10.

Proof. Because the neighbor radius r_η is added to the repulsion function as a variable, the internal repulsion is

$$P = \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta, d_{ij}). \quad (\text{A.30})$$

The neighbor distance d_{ij} can be considered a random variable with expectation $\mathbf{E}[d_{ij}] = \rho_{ij}$; thus, we define P_ρ as the internal repulsion by expectation of neighbor distance as follows:

$$P_\rho = \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta, \mathbf{E}[d_{ij}]) = \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta, \rho_{ij}). \quad (\text{A.31})$$

Let e_P be the tracking error of P_ρ for the desired value P_d :

$$e_P = P_d - P_\rho = P_d - \sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta, \rho_{ij}). \quad (\text{A.32})$$

Using the quadratic form of e_P , we define the Lyapunov candidate function V as follows:

$$V = \frac{1}{2} (P_d - P_\rho)^2 = \frac{1}{2} e_P^2. \quad (\text{A.33})$$

The derivative of V is

$$\begin{aligned} \frac{dV}{dt} &= (P_d - P_\rho) \cdot \left(-\frac{d}{dt} \left(\sum_{i=1}^N \sum_{j=1, i \neq j}^N g(r_\eta, \rho_{ij}) \right) \right) \\ &= -e_P \cdot \left(\sum_{i=1}^N \sum_{j=1, i \neq j}^N \frac{\partial g}{\partial r_\eta} \right) \cdot \frac{dr_\eta}{dt}. \end{aligned} \quad (\text{A.34})$$

We set the derivative of r_η to $G_{IRR} e_P$ for a positive gain G_{IRR} , and the differentiation of the repulsion function to always be greater than zero for $0 < \rho_{ij} < r_\eta$:

$$\frac{dr_\eta}{dt} = G_{IRR} e_P, \quad (\text{A.35})$$

$$\frac{\partial g(r_\eta, \rho_{ij})}{\partial r_\eta} > 0 \quad (\exists \rho_{ij}, \rho_{ij} < r_\eta).$$

Therefore, the derivative of V is always smaller than zero for $L \neq \mathbf{0}$:

$$\frac{dV}{dt} = -G_{IRR} e_P^2 \cdot \left(\sum_{i=1}^N \sum_{j=1, i \neq j}^N \frac{\partial g}{\partial r_{ij}} \right) < 0. \quad (\text{A.36})$$

If the conditions of (A.35) are satisfied, V converges to zero. Thus, the tracking error for internal repulsion by the expectation of neighbor distance goes to zero:

$$e_P = P_d - P_\rho \longrightarrow 0 \quad \text{as } t \longrightarrow \infty. \quad (\text{A.37})$$

□

Data Availability

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

Conflicts of Interest

The authors have no conflicts of interest to declare regarding the publication of this paper.

Acknowledgments

This research was financially supported by the Ministry of Trade, Industry, and Energy (MOTIE) and the Korea Evaluation Institute of Industrial Technology (KEIT) through the Industrial Strategic Technology Development Program (10080636).

References

- [1] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- [2] H. Cheset, K. M. Lynch, S. Hutchinson et al., *Principles of Robot Motion: Theory, Algorithms, and Implementations*, The MIT Press, Cambridge, UK, 2005.
- [3] L. E. Kavarakis, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration space," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [4] J. Barraquand, L. Kavraki, J.-C. Latombe, R. Motwani, T.-Y. Li, and P. Raghavan, "A random sampling scheme for path planning," *International Journal of Robotics Research*, vol. 16, no. 6, pp. 759–774, 1997.
- [5] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," in *Proceedings of the IEEE ICRA*, vol. 2, pp. 995–1001, 2000.
- [6] D. Hsu, J.-C. Latombe, and H. Kurniawati, "On the probabilistic foundations of probabilistic roadmap planning," *International Journal of Robotics Research*, vol. 25, no. 7, pp. 627–643, 2006.
- [7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [8] J. D. Marble and K. E. Bekris, "Asymptotically near-optimal planning with probabilistic roadmap spanners," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 432–444, 2013.
- [9] D. Balkcom, A. Kannan, Y.-H. Lyu, W. Wang, and Y. Zhang, "Metric Cells: Towards Complete Search for Optimal Trajectories," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015*, pp. 4941–4948, Germany, October 2015.
- [10] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: a review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
- [11] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif, "Narrow passage sampling for probabilistic roadmap planning," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1105–1115, 2005.
- [12] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1018–1023, 1999.
- [13] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the space," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1024–1031, 1999.
- [14] T. Siméon, J.-P. Laumond, and C. Nissoux, "Visibility-based probabilistic roadmaps for motion planning," *Advanced Robotics*, vol. 14, no. 6, pp. 477–493, 2000.
- [15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [16] J. Yan and H. Yu, "Distributed Optimization of Multiagent Systems in Directed Networks with Time-Varying Delay," *Journal of Control Science and Engineering*, vol. 2017, Article ID 7937916, 9 pages, 2017.
- [17] S. Weerakkody, X. Liu, S. H. Son, and B. Sinopoli, "A graph-theoretic characterization of perfect attackability for secure design of distributed control systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 60–70, 2017.
- [18] R. Zhang and Q. Zhu, "Secure and resilient distributed machine learning under adversarial environments," in *Proceedings of the 18th International Conference on Information Fusion*, pp. 644–651, 2015.
- [19] S. Sundaram and B. Ghahesifard, "Distributed Optimization Under Adversarial Nodes," *IEEE Transactions on Automatic Control*, 2018.
- [20] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.
- [21] A. Kwok and S. Martinez, "Unicycle coverage control via hybrid modeling," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 55, no. 2, pp. 528–532, 2010.
- [22] H. Mahboubi and A. G. Aghdam, "Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: relocation by virtual force," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 736–748, 2017.
- [23] J.-H. Park, J.-H. Bae, and M.-H. Baeg, "Adaptation algorithm of geometric graphs for robot motion planning in dynamic environments," *Mathematical Problems in Engineering*, vol. 2016, Article ID 3973467, 19 pages, 2016.
- [24] <https://www.yaskawa.eu.com/en/products/robotic/motoman-robots/productdetail/product/mpx3500/>.
- [25] K. Ferland, *Discrete Mathematics*, CENGAGE Learning, Belmont, NY, USA, 2008.
- [26] T. Tao, *An Introduction to Measure Theory*, American Mathematical Society, 2011.

- [27] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Massachusetts, MASS, USA, 1996.
- [28] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Massachusetts, MASS, USA, 2014.
- [29] C. Langbort, L. Xiao, R. D'Andrea, and S. Boyd, "A decomposition approach to distributed analysis of networked systems," in *Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC)*, pp. 3980–3985, Bahamas, Caribbean, December 2004.
- [30] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [31] P. Sadegh, "Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation," *Automatica*, vol. 33, no. 5, pp. 889–892, 1997.
- [32] K. Marti, *Stochastic Optimization Methods*, Springer, Berlin Heidelberg, 3rd edition, 2008.
- [33] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*, John Wiley & Sons, 3rd edition, 2009.
- [34] M. S. Ebeida, S. A. Mitchell, A. Patney, A. A. Davidson, and J. D. Owens, "A simple algorithm for maximal poisson-disk sampling in high dimensions," *Computer Graphics Forum*, vol. 31, no. 2, pp. 785–794, 2012.
- [35] E. A. Jackson, *Equilibrium Statistical Mechanics*, Prentice-Hall, New Jersey, NJ, USA, 1968.
- [36] L. Kocarev, *Consensus and Synchronization in Complex Networks*, Springer, Berlin Heidelberg, 2013.
- [37] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical Processes on Complex Networks*, Cambridge University Press, Cambridge, UK, 2008.
- [38] C. Ericson, *Real-Time Collision Detection*, CRC Press, New York, NY, USA, 2004.
- [39] W. F. Basener, *Topology and Its Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2006.
- [40] X. Wang, "Volumes of Generalized Unit Balls," *Mathematics Magazine*, vol. 78, no. 5, pp. 390–395, 2005.

Research Article

Adaptive Fuzzy Control for Stochastic Pure-Feedback Nonlinear Systems with Unknown Hysteresis and External Disturbance

Xikui Liu,¹ Yingying Ge,² and Yan Li ³

¹College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China

²College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

³College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Yan Li; liyanhd@163.com

Received 25 June 2018; Revised 13 September 2018; Accepted 25 October 2018; Published 8 November 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Xikui Liu 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.

This paper solves the tracking control problem of a class of stochastic pure-feedback nonlinear systems with external disturbances and unknown hysteresis. By using the mean-value theorem, the problem of pure-feedback nonlinear function is solved. The direction-unknown hysteresis problem is solved with the aid of the Nussbaum function. The external disturbance problems can be solved by defining new Lyapunov functions. Using the backstepping technique, a new adaptive fuzzy control scheme is proposed. The results show that the proposed control scheme ensures that all signals of the closed-loop system are semiglobally uniformly bounded and the tracking error converges to the small neighborhood of origin in the sense of mean quartic value. Simulation results illustrate the effectiveness of the proposed control scheme.

1. Introduction

Hysteresis is widely found in mechanical equipment, which severely limits performance of the system and even leads to system instability. Therefore, the control problem of the system with hysteresis has been paid more and more attention. For the adaptive control system in [1–7], scholars study problems in different directions, such as hysteresis input in [1], dead zone input in [2], and time-delay input in [3]. There are several common hysteresis phenomena. The author solves the backlash-like hysteresis problem in [4]. The authors solve a class of traditional P-I hysteresis problems and propose an adaptive backstepping scheme in [5]. Scholars have studied a class of nonlinear systems with generalized P-I hysteresis inputs in [6, 7]. In addition, scholars have studied unmodeled dynamics deterministic systems in [8] and uncertain nonsmooth deterministic systems in [9]. The finite-time control problem of nonlinear deterministic systems is studied in [10]. However, the system studied above is a deterministic system and ignores the effects of stochastic disturbance.

Stochastic disturbance often occur in many systems, and the adaptive control problems of stochastic nonlinear systems

are more difficult than those of deterministic nonlinear systems. Stochastic disturbance is added to the system, and differential operations on Lyapunov functions are more complicated. The research of stochastic nonlinear systems has been increasingly discussed in [11–21]. A finite time control method of switched stochastic systems is proposed in [11]. The control problem of nonlinear stochastic systems is discussed in [12–14]. Using the mean value theorem to solve the pure-feedback nonlinear function, the complexity of the system is increased. Further, more researchers have studied other types of stochastic nonlinear systems, such as from stochastic systems with unknown backlash-like hysteresis in [15] to pure-feedback stochastic nonlinear systems with unknown dead-zone input in [16], from SISO systems in [17] to pure-feedback MIMO systems in [18]. The stochastic systems with time-varying delays are proposed in [19, 20] and the stochastic systems with unknown direction hysteresis are proposed in [21]. The above studies have considered the effect of stochastic disturbance, without considering the external disturbances.

However, external disturbance often exists in practice. External disturbance cannot be ignored; it is also a source of

system instability in practice. Scholars have extended the system without external disturbances in [22–25] to systems with external disturbances in [26–30], such as a determination system with external disturbances in [29] and a stochastic system with external disturbances in [30]. The system with external disturbances makes the design of the controller more difficult.

In this paper, the control problem of pure-feedback stochastic nonlinear system with external disturbance and unknown hysteresis is studied. For the determination system with external disturbance in [26], stochastic terms are not considered. The adaptive fuzzy control problem for stochastic nonlinear systems is studied in [14], without considering external disturbances. Therefore, a more general nonlinear system is processed in this paper. Furthermore, the difficulty is to deal with unknown direction hysteresis in [21]. The difficulty of this paper is how to solve the influence of external disturbance on the unknown direction hysteresis and ensure the stability of the nonlinear system. This problem can be solved by designing appropriate Lyapunov functions. The major contributions of this paper are described below:

- (1) The tracking control problem of the stochastic pure-feedback nonlinear systems with stochastic disturbances, direction-unknown hysteresis, and external disturbances is solved in this paper.
- (2) In the n th step of the backstepping design, the Lyapunov function with the external disturbance term $\Delta(t)$ is defined, and the external disturbance problem is solved. A new adaptive control scheme is proposed.

The remainder of this article is as follows. The second part puts forward the preparation work and the problem formulation. The third part is the design process of the adaptive control method. The fourth part gives the simulation example. The fifth part summarizes the full text.

2. Preparation and Problem Formulation

2.1. Preliminary Knowledge. The stochastic nonlinear system is expressed as follows:

$$dx = f(x, t) dt + h(x, t) dw, \quad (1)$$

where $x \in R^n$ is the state variable, $f: R^n \times R^+ \rightarrow R^n$, $h: R^n \times R^+ \rightarrow R^{n \times r}$ are continuous functions. w indicates that an independent r -dimension standard Brownian motion, which is defined on the complete probability space.

Definition 1 (see [31]). For a quadratic continuous differentiable function $V(x, t)$, define a derivative operator L expressed as follows:

$$LV = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f + \frac{1}{2} Tr \left\{ h^T \frac{\partial^2 V}{\partial x^2} h \right\}, \quad (2)$$

where Tr is a trace of matrix.

Remark 2. The $\frac{\partial^2 V}{\partial x^2}$ in $It\hat{o}$ correction term $(1/2)Tr\{h^T(\frac{\partial^2 V}{\partial x^2})h\}$ makes the design of the control scheme

in the stochastic system more complicated than the design of the control scheme in the determined system.

Lemma 3 (see [32]). For the stochastic system (1), let $V(t)$ and $\zeta(t)$ are smooth functions defined on $[0, t_f]$, $V(t) \geq 0$, $\forall t \in [0, t_f]$; the function ξ satisfies $0 < |\xi(t)| \leq l < \infty$ (l is a constant); $N(\cdot)$ is a Nussbaum-type function. The following inequality is satisfied:

$$V(t) \leq e^{-ct} \int_0^t (\xi(t) N(\zeta) \dot{\zeta} + \dot{\zeta}) e^{ct} d\tau + \eta + M(t), \quad (3)$$

$$\forall t \in [0, t_f],$$

where η is a nonnegative variable, $M(t)$ is a real valued continuous local martingale. Then the functions $V(t)$, $\zeta(t)$ and $\int_0^t (\xi(t) N(\zeta) \dot{\zeta} + \dot{\zeta}) d\tau$ are bounded on $[0, t_f]$.

Lemma 4 (see [33]). For $\forall(x, y) \in R^2$, the following inequality is established:

$$xy \leq \frac{\varepsilon^p}{p} |x|^p + \frac{1}{q\varepsilon^q} |y|^q, \quad (4)$$

where $\varepsilon > 0$, $p > 1$, and $(p-1)(q-1) = 1$.

Lemma 5 (see [3]). Consider the following dynamic system:

$$\dot{\hat{\theta}}(t) = -\gamma \hat{\theta}(t) + k\rho(t), \quad (5)$$

where γ and k are positive constants and $\rho(t)$ is a positive function. Then for $\forall t \geq t_0$ and any bounded initial condition $\hat{\theta}(t_0) \geq 0$, we have $\hat{\theta}(t) \geq 0$.

2.2. Problem Formulation. This paper considers the following stochastic pure-feedback nonlinear system:

$$dx_i = f_i(\bar{x}_i, x_{i+1}) dt + \psi_i^T(\bar{x}_i) dw, \quad 1 \leq i \leq n-1,$$

$$dx_n = (f_n(\bar{x}_n, u) + \Delta(t)) dt + \psi_n^T(\bar{x}_n) dw, \quad (6)$$

$$y = x_1,$$

where $x = [x_1, x_2, \dots, x_n]^T \in R^n$ and $\bar{x}_i = [x_1, x_2, \dots, x_i]^T \in R^i$ are the state vectors, $y \in R$ is system output, and w is defined as (1). $f_i(\cdot): R^i \rightarrow R$ and $\psi_i(\cdot): R^i \rightarrow R^r$ ($i = 1, 2, \dots, n$) are unknown nonlinear functions. $\Delta(t)$ is a bounded external disturbance. $u \in R$ is the system input and the output of an unknown Bouc-Wen hysteresis is defined as follows [1]:

$$u = H(v) = \mu_1 v + \mu_2 \zeta, \quad (7)$$

where μ_1 and μ_2 are unknown constants and have the same sign. $v(t) \in R$ is the input of the hysteresis. ζ is the auxiliary variable, $\zeta(t_0) = 0$. In [1] we know that ζ is bounded and can be expressed as:

$$|\zeta| \leq \sqrt{\frac{1}{(\beta + \chi)}}, \quad (8)$$

where β, χ, n are the unknown hysteresis parameters, and $\beta > |\chi|, n > 1$.

Remark 6. This article has stochastic term $\psi_i^T(\bar{x}_i)dw$, and the hysteresis output u is different from [29]. If we ignore the external disturbance $\Delta(t)$, the results of this paper are the same as [21]. Therefore, this paper considers a more general nonlinear system.

For the system (6), define

$$g_i(\bar{x}_i, x_{i+1}) = \frac{\partial f_i(\bar{x}_i, x_{i+1})}{\partial x_{i+1}}, \quad i = 1, 2, \dots, n, \quad (9)$$

with $x_{i+1} = u$.

Assumption 7. For $1 \leq i \leq n$, there is an unknown constant b_M such that

$$1 \leq g_i(\bar{x}_i, x_{i+1}) \leq b_M < \infty, \quad \forall (\bar{x}_i, x_{i+1}) \in R^i \times R. \quad (10)$$

By using the mean-value theorem, the pure-feedback nonlinear functions in (6) can be expressed as

$$f_i(\bar{x}_i, x_{i+1}) - f_i(\bar{x}_i, x_{i+1}^0) = g_i(\bar{x}_i, \eta_i)(x_{i+1} - x_{i+1}^0), \quad (11)$$

where $x_{n+1}^0 = u^0 = H(v^0)$, $x_{n+1} = u = H(v)$, and η_i is point between x_{i+1} and x_{i+1}^0 .

Substituting (11) into (6), the control system can be rewritten as

$$\begin{aligned} dx_i &= (f_i(\bar{x}_i, x_{i+1}^0) + g_i(\bar{x}_i, \eta_i)(x_{i+1} - x_{i+1}^0)) dt \\ &\quad + \psi_i^T(\bar{x}_i) dw, \quad 1 \leq i \leq n-1, \\ dx_n &= (f_n(\bar{x}_n, u^0) + g_n(\bar{x}_n, \eta_n)(u - u^0) + \Delta(t)) dt \\ &\quad + \psi_n^T(\bar{x}_n) dw, \\ y &= x_1. \end{aligned} \quad (12)$$

2.3. Fuzzy Logic Systems. In order to approximate a continuous function $f(x)$ with a fuzzy logic system, consider the following fuzzy rules:

R^l : If x_1 is F_1^l and \dots and x_n is F_n^l .

Then y is G^l , $l = 1, 2, \dots, N$,

where $x = [x_1, x_2, \dots, x_n]^T \in R^n$ is the input of system, $y \in R$ is the output of the system, F_i^l and G^l are fuzzy sets in R , and N is the number of rules. The output form of the fuzzy logic system is as follows:

$$y(x) = \frac{\sum_{l=1}^N \Phi_l \prod_{i=1}^n \mu_{F_i^l}(x_i)}{\sum_{l=1}^N \left[\prod_{i=1}^n \mu_{F_i^l}(x_i) \right]}, \quad (13)$$

where

$$\begin{aligned} \Phi_l &= \max_{y \in R} \mu_{G^l}(y), \\ \Phi &= (\Phi_1, \Phi_2, \dots, \Phi_N)^T. \end{aligned} \quad (14)$$

Letting

$$\xi_l(x) = \frac{\prod_{i=1}^n \mu_{F_i^l}(x_i)}{\sum_{l=1}^N \left[\prod_{i=1}^n \mu_{F_i^l}(x_i) \right]}, \quad (15)$$

$$\xi(X) = (\xi_1(x), \xi_2(x), \dots, \xi_N(x))^T,$$

the fuzzy system is written as

$$y(x) = \Phi^T \xi(x). \quad (16)$$

Lemma 8 (see [27]). *Let $f(x)$ be a continuous function defined on a compact set Ω . Then, for $\forall \varepsilon > 0$, there exists a fuzzy logic system $\Phi^T \xi(x)$ such that*

$$\sup_{x \in \Omega} |f(x) - \Phi^T \xi(x)| \leq \varepsilon. \quad (17)$$

The goal of this paper is to design an adaptive controller, so that the system output y converges to the reference signal y_d and all signals of the closed-loop system are bounded.

Define a vector function as $\bar{y}_d^{(i)} = [y_d, y_d^{(1)}, \dots, y_d^{(i)}]^T$, $i = 1, 2, \dots, n$, where $y_d^{(i)}$ denotes the i th order derivative of y_d .

Assumption 9 (see [21]). The reference signal $y_d(t)$ and its order derivatives up to the n th time are continuous and bounded.

3. Adaptive Control Design

In this part, the adaptive fuzzy control is proposed by the backstepping technique, and the following coordinate transformation is defined to develop the backstepping technique:

$$\begin{aligned} z_1 &= y - y_d, \\ z_i &= x_i - \alpha_{i-1}, \quad i = 2, \dots, n, \end{aligned} \quad (18)$$

where α_{i-1} is an intermediate function to be determined next.

In each step of the backstepping method, a fuzzy logic system $\Phi_l^T \xi_l(X_i)$ will be used to approximate an unknown function f_i . We define a constant $\theta_i = \|\Phi_i\|^2$, the estimation error is $\tilde{\theta}_i = \theta_i - \hat{\theta}_i$, $\hat{\theta}_i$ as the estimation of θ_i , $i = 1, 2, \dots, n$.

Step 1. For stochastic pure-feedback systems (12), according to $z_1 = x_1 - y_d$, we know that dynamic error is satisfied

$$\begin{aligned} dz_1 &= (f_1(\bar{x}_1, x_2^0) + g_1(\bar{x}_1, \eta_1)(x_2 - x_2^0) - \dot{y}_d) dt \\ &\quad + \psi_1^T dw. \end{aligned} \quad (19)$$

We choose Lyapunov function as follows:

$$V_1 = \frac{1}{4} z_1^4 + \frac{1}{2\lambda_1} \tilde{\theta}_1^2, \quad (20)$$

where λ_1 is a positive constant. By (2), (18), and (19), one has

$$\begin{aligned} LV_1 &= z_1^3 (f_1(\bar{x}_1, x_2^0) + g_1(\bar{x}_1, \eta_1)(z_2 + \alpha_1 - x_2^0) - \dot{y}_d) \\ &\quad + \frac{3}{2} z_1^2 \psi_1^T \psi_1 - \frac{1}{\lambda_1} \tilde{\theta}_1 \dot{\tilde{\theta}}_1. \end{aligned} \quad (21)$$

Applying Lemma 4 and Assumption 7, the following inequalities hold:

$$\frac{3}{2}z_1^2\psi_1^T\psi_1 \leq \frac{3}{4}l_1^{-2}z_1^4\|\psi_1\|^4 + \frac{3}{4}l_1^2, \quad (22)$$

$$g_1(\bar{x}_1, \eta_1)z_1^3z_2 \leq \frac{3}{4}b_Mz_1^4 + \frac{b_M}{4}z_2^4, \quad (23)$$

where l_1 is a positive constant. Substituting (22) and (23) into (21), we can get

$$\begin{aligned} LV_1 \leq & z_1^3 \left(f_1(\bar{x}_1, x_2^0) + g_1(\bar{x}_1, \eta_1)(\alpha_1 - x_2^0) - \dot{y}_d \right. \\ & \left. + \frac{3}{4}b_Mz_1 + \frac{3}{4}l_1^{-2}z_1\|\psi_1\|^4 \right) + \frac{3}{4}l_1^2 + \frac{b_M}{4}z_2^4 - \frac{1}{\lambda_1} \\ & \cdot \bar{\theta}_1\dot{\hat{\theta}}_1. \end{aligned} \quad (24)$$

Defining a new function $\bar{f}_1 = f_1 + (3/4)b_Mz_1 - \dot{y}_d + (3/4)l_1^{-2}z_1\|\psi_1\|^4$, then the above inequality can be rewritten as

$$\begin{aligned} LV_1 \leq & z_1^3\bar{f}_1 + z_1^3g_1(\alpha_1 - x_2^0) + \frac{3}{4}l_1^2 + \frac{b_M}{4}z_2^4 \\ & - \frac{1}{\lambda_1}\bar{\theta}_1\dot{\hat{\theta}}_1. \end{aligned} \quad (25)$$

Because \bar{f}_1 contains the unknown function f_1 and ψ_1 , it cannot be directly controlled in practice. Therefore, according to Lemma 8, for any given $\varepsilon_1 > 0$, there exists a fuzzy logic system $\Phi_1^T\xi_1(X_1)$ such that

$$\bar{f}_1 = \Phi_1^T\xi_1(X_1) + \delta_1(X_1), \quad |\delta_1(X_1)| \leq \varepsilon_1, \quad (26)$$

where $X_1 = (x_1, y_d, \dot{y}_d)$. According to Lemma 4, it follows that

$$\begin{aligned} z_1^3\bar{f}_1 &= z_1^3\Phi_1^T\xi_1(X_1) + z_1^3\delta_1(X_1) \\ &\leq \frac{z_1^6}{2a_1^2}\|\Phi_1\|^2\xi_1^T\xi_1 + \frac{1}{2}a_1^2 + \frac{3}{4}z_1^4 + \frac{1}{4}\varepsilon_1^4 \\ &= \frac{z_1^6}{2a_1^2}\theta_1\xi_1^T\xi_1 + \frac{1}{2}a_1^2 + \frac{3}{4}z_1^4 + \frac{1}{4}\varepsilon_1^4, \end{aligned} \quad (27)$$

where a_1 is a positive parameter. We choose the following virtual control signal and adaptive law:

$$\alpha_1 = -\left(k_1 + \frac{3}{4}\right)z_1 - \frac{1}{2a_1^2}\hat{\theta}_1z_1^3\xi_1^T\xi_1 + x_2^0, \quad (28)$$

$$\dot{\hat{\theta}}_1 = \frac{\lambda_1}{2a_1^2}z_1^6\xi_1^T\xi_1 - \gamma_1\hat{\theta}_1, \quad \hat{\theta}_1(0) \geq 0, \quad (29)$$

where k_1 and γ_1 are positive constants. Based on (28), Assumption 7, one has

$$z_1^3g_1(\alpha_1 - x_2^0) \leq -\left(k_1 + \frac{3}{4}\right)z_1^4 - \frac{1}{2a_1^2}z_1^6\hat{\theta}_1\xi_1^T\xi_1. \quad (30)$$

Substituting (27), (29), and (30) into (25), we have

$$LV_1 \leq -k_1z_1^4 + \frac{b_M}{4}z_2^4 + \frac{3}{4}l_1^2 + \frac{1}{4}\varepsilon_1^4 + \frac{\gamma_1}{\lambda_1}\bar{\theta}_1\hat{\theta}_1 + \frac{1}{2}a_1^2. \quad (31)$$

It is noted that

$$\frac{\gamma_1}{\lambda_1}\bar{\theta}_1\hat{\theta}_1 \leq -\frac{\gamma_1}{2\lambda_1}\bar{\theta}_1^2 + \frac{\gamma_1}{2\lambda_1}\theta_1^2. \quad (32)$$

Substituting (32) into (31), we have

$$LV_1 \leq -k_1z_1^4 - \frac{\gamma_1}{2\lambda_1}\bar{\theta}_1^2 + \varrho_1 + \frac{b_M}{4}z_2^4, \quad (33)$$

where $\varrho_1 = (3/4)l_1^2 + (1/2)a_1^2 + (1/4)\varepsilon_1^4 + (\gamma_1/2\lambda_1)\theta_1^2$.

Step 2. Since $z_2 = x_2 - \alpha_1$ and *Itô* formula, one has

$$\begin{aligned} dz_2 &= \left(f_2(\bar{x}_2, x_3^0) + g_2(\bar{x}_2, \eta_2)(x_3 - x_3^0) - L\alpha_1 \right) dt \\ &\quad + \left(\psi_2 - \frac{\partial\alpha_1}{\partial x_1}\psi_1 \right)^T dw, \end{aligned} \quad (34)$$

with

$$\begin{aligned} L\alpha_1 &= \frac{\partial\alpha_1}{\partial x_1} \left[f_1(\bar{x}_1, x_2^0) + g_1(\bar{x}_1, \eta_1)(x_2 - x_2^0) \right] \\ &\quad + \frac{\partial\alpha_1}{\partial\hat{\theta}_1}\dot{\hat{\theta}}_1 + \sum_{i=0}^1 \frac{\partial\alpha_1}{\partial y_d^{(i)}}y_d^{(i+1)} + \frac{1}{2} \frac{\partial^2\alpha_1}{\partial x_1^2}\psi_1^T\psi_1. \end{aligned} \quad (35)$$

Choose stochastic Lyapunov function as

$$V_2 = V_1 + \frac{1}{4}z_2^4 + \frac{1}{2\lambda_2}\bar{\theta}_2^2, \quad (36)$$

where λ_2 is a positive design constant. Using the similar procedure as (21), it follows that

$$\begin{aligned} LV_2 &= LV_1 + z_2^3 \left(f_2(\bar{x}_2, x_3^0) \right. \\ &\quad \left. + g_2(\bar{x}_2, \eta_2)(z_3 + \alpha_2 - x_3^0) - L\alpha_1 \right) + \frac{3}{2}z_2^2 \left(\psi_2 \right. \\ &\quad \left. - \frac{\partial\alpha_1}{\partial x_1}\psi_1 \right)^T \left(\psi_2 - \frac{\partial\alpha_1}{\partial x_1}\psi_1 \right) - \frac{1}{\lambda_2}\bar{\theta}_2\dot{\hat{\theta}}_2. \end{aligned} \quad (37)$$

It is noticed that

$$\frac{3}{2}z_2^2 \left\| \psi_2 - \frac{\partial\alpha_1}{\partial x_1}\psi_1 \right\|^2 \leq \frac{3}{4}l_2^{-2}z_2^4 \left\| \psi_2 - \frac{\partial\alpha_1}{\partial x_1}\psi_1 \right\|^4 + \frac{3}{4}l_2^2, \quad (38)$$

$$g_2(\bar{x}_2, \eta_2)z_2^3z_3 \leq \frac{3}{4}b_Mz_2^4 + \frac{b_M}{4}z_3^4, \quad (39)$$

where l_2 is a positive parameter. Substituting (33), (38), and (39) into (37), we have

$$\begin{aligned} LV_2 &\leq -k_1z_1^4 - \frac{\gamma_1}{2\lambda_1}\bar{\theta}_1^2 + \varrho_1 + z_2^3 \left(f_2(\bar{x}_2, x_3^0) \right. \\ &\quad \left. + g_2(\bar{x}_2, \eta_2)(\alpha_2 - x_3^0) - L\alpha_1 + b_Mz_2 \right) \\ &\quad + \frac{3}{4}l_2^{-2}z_2 \left\| \psi_2 - \frac{\partial\alpha_1}{\partial x_1}\psi_1 \right\|^4 + \frac{3}{4}l_2^2 + \frac{b_M}{4}z_3^4 - \frac{1}{\lambda_2} \\ &\quad \cdot \bar{\theta}_2\dot{\hat{\theta}}_2. \end{aligned} \quad (40)$$

Define a function as $\bar{f}_2 = f_2 - L\alpha_1 + b_M z_2 + (3/4)l_2^2 z_2 \|\psi_2 - (\partial\alpha_1/\partial x_1)\psi_1\|^4$. Furthermore, (40) can be rewritten as

$$LV_2 \leq -k_1 z_1^4 - \frac{\gamma_1}{2\lambda_1} \bar{\theta}_1^2 + \varrho_1 + z_2^3 \bar{f}_2 + z_2^3 g_2 (\alpha_2 - x_3^0) + \frac{3}{4} l_2^2 + \frac{b_M}{4} z_3^4 - \frac{1}{\lambda_2} \tilde{\theta}_2 \dot{\hat{\theta}}_2. \quad (41)$$

Since \bar{f}_2 contains the unknown function f_1 , ψ_1 , and ψ_2 , it is not possible in practice. Thus, the fuzzy logic system $\Phi_2^T \xi_2(X_2)$ is used to approximate \bar{f}_2 , where $X_2 = [\bar{x}_2^T, \hat{\theta}_1, \bar{y}_d^{(2)T}]^T \in \Omega_{Z_2}$. Due to Lemma 8, \bar{f}_2 can be written as

$$\bar{f}_2 = \Phi_2^T \xi_2(X_2) + \delta_2(X_2), \quad |\delta_2(X_2)| \leq \varepsilon_2, \quad (42)$$

where ε_2 is any given positive constant. Repeating the method of (27), we have

$$z_2^3 \bar{f}_2 \leq \frac{z_2^6}{2a_2^2} \theta_2 \xi_2^T \xi_2 + \frac{1}{2} a_2^2 + \frac{3}{4} z_2^4 + \frac{1}{4} \varepsilon_2^4, \quad (43)$$

where a_2 is a positive parameter. We choose the following virtual control signal and adaptive law:

$$\alpha_2 = -\left(k_2 + \frac{3}{4}\right) z_2 - \frac{1}{2a_2^2} \hat{\theta}_2 z_2^3 \xi_2^T \xi_2 + x_3^0, \quad (44)$$

$$\dot{\hat{\theta}}_2 = \frac{\lambda_2}{2a_2^2} z_2^6 \xi_2^T \xi_2 - \gamma_2 \hat{\theta}_2, \quad \hat{\theta}_2(0) \geq 0, \quad (45)$$

where k_2 , γ_2 are design constants. Similar to (30), the following inequality is obtained:

$$z_2^3 g_2 (\alpha_2 - x_3^0) \leq -\left(k_2 + \frac{3}{4}\right) z_2^4 - \frac{1}{2a_2^2} z_2^6 \hat{\theta}_2 \xi_2^T \xi_2. \quad (46)$$

Substituting (43), (45) and (46) into (41), we have

$$LV_2 \leq -k_1 z_1^4 - \frac{\gamma_1}{2\lambda_1} \bar{\theta}_1^2 + \varrho_1 - k_2 z_2^4 + \frac{b_M}{4} z_3^4 + \frac{3}{4} l_2^2 + \frac{1}{4} \varepsilon_2^4 + \frac{\gamma_2}{\lambda_2} \tilde{\theta}_2 \hat{\theta}_2 + \frac{1}{2} a_2^2. \quad (47)$$

It is noted that

$$\frac{\gamma_2}{\lambda_2} \tilde{\theta}_2 \hat{\theta}_2 \leq -\frac{\gamma_2}{2\lambda_2} \bar{\theta}_2^2 + \frac{\gamma_2}{2\lambda_2} \theta_2^2, \quad (48)$$

(47), can be rewritten in the form

$$LV_2 \leq -\sum_{j=1}^2 \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^2 \varrho_j + \frac{b_M}{4} z_3^4, \quad (49)$$

where $\varrho_j = (\gamma_j/2\lambda_j)\theta_j^2 + (3/4)l_j^2 + (1/4)\varepsilon_j^4 + (1/2)a_j^2$, $j = 1, 2$.

Step i ($3 \leq i \leq n-1$). According to $z_i = x_i - \alpha_{i-1}$ and *Itô* formula, one has

$$dz_i = (f_i(\bar{x}_i, x_{i+1}^0) + g_i(\bar{x}_i, \eta_i)(x_{i+1} - x_{i+1}^0) - L\alpha_{i-1}) dt + \left(\psi_i - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_j \right)^T dw, \quad (50)$$

with

$$L\alpha_{i-1} = \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} [f_j + g_j(x_{j+1} - x_{j+1}^0)] + \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}_j} \dot{\hat{\theta}}_j + \sum_{j=0}^{i-1} \frac{\partial \alpha_{i-1}}{\partial y_d^{(j)}} y_d^{(j+1)} + \frac{1}{2} \sum_{p,q=1}^{i-1} \frac{\partial^2 \alpha_{i-1}}{\partial x_p \partial x_q} \psi_p^T \psi_q. \quad (51)$$

We consider the following Lyapunov function:

$$V_i = V_{i-1} + \frac{1}{4} z_i^4 + \frac{1}{2\lambda_i} \bar{\theta}_i^2, \quad (52)$$

where λ_i is a positive constant. Using the similar procedure as (21), it follows that

$$LV_i = LV_{i-1} + z_i^3 (f_i + g_i(z_{i+1} + \alpha_i - x_{i+1}^0) - L\alpha_{i-1}) + \frac{3}{2} z_i^2 \left(\psi_i - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_j \right)^T \left(\psi_i - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_j \right) - \frac{1}{\lambda_i} \tilde{\theta}_i \dot{\hat{\theta}}_i. \quad (53)$$

It is noticed that

$$\frac{3}{2} z_i^2 \left\| \psi_i - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_j \right\|^2 \leq \frac{3}{4} l_i^2 + \frac{3}{4} l_i^{-2} z_i^4 \left\| \psi_i - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_j \right\|^4, \quad (54)$$

$$g_i(\bar{x}_i, \eta_i) z_i^3 z_{i+1} \leq \frac{3}{4} b_M z_i^4 + \frac{b_M}{4} z_{i+1}^4, \quad (55)$$

where l_i is a positive constant. Using the similar procedure as (24), we have

$$LV_i \leq -\sum_{j=1}^{i-1} \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^{i-1} \varrho_j + z_i^3 \left(f_i + g_i(\alpha_i - x_{i+1}^0) - L\alpha_{i-1} + b_M z_i \right) + \frac{3}{4} l_i^{-2} z_i \left\| \psi_i - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_j \right\|^4 + \frac{3}{4} l_i^2 + \frac{b_M}{4} z_{i+1}^4 - \frac{1}{\lambda_i} \tilde{\theta}_i \dot{\hat{\theta}}_i. \quad (56)$$

Define a function as $\bar{f}_i = f_i - L\alpha_{i-1} + b_M z_i + (3/4)l_i^{-2} z_i \|\psi_i - \sum_{j=1}^{i-1} (\partial\alpha_{i-1}/\partial x_j)\psi_j\|^4$. Furthermore, (56) can be rewritten as

$$LV_i \leq -\sum_{j=1}^{i-1} \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^{i-1} \varrho_j + z_i^3 \bar{f}_i + z_i^3 g_i(\alpha_i - x_{i+1}^0) + \frac{3}{4} l_i^2 + \frac{b_M}{4} z_{i+1}^4 - \frac{1}{\lambda_i} \tilde{\theta}_i \dot{\hat{\theta}}_i. \quad (57)$$

Similarly, The fuzzy logic system $\Phi_i^T \xi_i(X_i)$ is used to approximate \bar{f}_i , where $X_i = [\bar{x}_i^T, \bar{\theta}_{i-1}^T, \bar{y}_d^{(i)T}]^T \in \Omega_{Z_i}$, with $\bar{\theta}_{i-1} = [\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_{i-1}]^T$. According to Lemma 8, \bar{f}_i can be expressed as

$$\bar{f}_i = \Phi_i^T \xi_i(X_i) + \delta_i(X_i), \quad |\delta_i(X_i)| \leq \varepsilon_i, \quad (58)$$

where ε_i is any given positive constant. According to the method of (27), we can get

$$z_i^3 \bar{f}_i \leq \frac{z_i^6}{2a_i^2} \theta_i \xi_i^T \xi_i + \frac{1}{2} a_i^2 + \frac{3}{4} z_i^4 + \frac{1}{4} \varepsilon_i^4, \quad (59)$$

where a_i is a design constant. We choose the following virtual control signal and adaptive law:

$$\alpha_i = -\left(k_i + \frac{3}{4}\right) z_i - \frac{1}{2a_i^2} \hat{\theta}_i z_i^3 \xi_i^T \xi_i + x_{i+1}^0, \quad (60)$$

$$\dot{\hat{\theta}}_i = \frac{\lambda_i}{2a_i^2} z_i^6 \xi_i^T \xi_i - \gamma_i \hat{\theta}_i, \quad \hat{\theta}_i(0) \geq 0, \quad (61)$$

where k_i and γ_i are positive parameters. Similar to (30), we have

$$z_i^3 g_i(\alpha_i - x_{i+1}^0) \leq -\left(k_i + \frac{3}{4}\right) z_i^4 - \frac{1}{2a_i^2} z_i^6 \hat{\theta}_i \xi_i^T \xi_i. \quad (62)$$

Similar to (31), we have

$$\begin{aligned} LV_i \leq & -\sum_{j=1}^{i-1} \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^{i-1} \varrho_j - k_i z_i^4 + \frac{b_M}{4} z_{i+1}^4 \\ & + \frac{3}{4} l_i^2 + \frac{1}{4} \varepsilon_i^4 + \frac{\gamma_i}{\lambda_i} \bar{\theta}_i \hat{\theta}_i + \frac{1}{2} a_i^2. \end{aligned} \quad (63)$$

It is noted that

$$\frac{\gamma_i}{\lambda_i} \bar{\theta}_i \hat{\theta}_i \leq -\frac{\gamma_i}{2\lambda_i} \bar{\theta}_i^2 + \frac{\gamma_i}{2\lambda_i} \theta_i^2. \quad (64)$$

The above inequality can be rewritten as

$$LV_i \leq -\sum_{j=1}^i \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^i \varrho_j + \frac{b_M}{4} z_{i+1}^4, \quad (65)$$

where $\varrho_j = (\gamma_j/2\lambda_j)\bar{\theta}_j^2 + (3/4)l_j^2 + (1/4)\varepsilon_j^4 + (1/2)a_j^2$, $j = 1, 2, \dots, i$.

Step n. Based on the coordinate transformation $z_n = x_n - \alpha_{n-1}$ and *Itô* formula, we can get

$$\begin{aligned} dz_n = & \left(f_n(\bar{x}_n, u^0) + g_n(\bar{x}_n, \eta_n)(u - u^0) + \Delta(t) \right. \\ & \left. - L\alpha_{n-1} \right) dt + \left(\psi_n - \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \psi_j \right)^T dw, \end{aligned} \quad (66)$$

with

$$\begin{aligned} L\alpha_{n-1} = & \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \left[f_j + g_j(x_{j+1} - x_{j+1}^0) \right] \\ & + \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial \hat{\theta}_j} \dot{\hat{\theta}}_j + \sum_{j=0}^{n-1} \frac{\partial \alpha_{n-1}}{\partial y_d^{(j)}} y_d^{(j+1)} \\ & + \frac{1}{2} \sum_{p,q=1}^{n-1} \frac{\partial^2 \alpha_{n-1}}{\partial x_p \partial x_q} \psi_p^T \psi_q. \end{aligned} \quad (67)$$

Consider stochastic Lyapunov function as follows:

$$V_n = V_{n-1} + \frac{1}{4} z_n^4 + \frac{1}{2\lambda_n} \bar{\theta}_n^2 + \frac{1}{2\sigma} \tilde{\Delta}^2, \quad (68)$$

where λ_n and σ are positive constants. Denote $\hat{\Delta}$ as the estimation of Δ , and the estimation error is $\tilde{\Delta} = \Delta - \hat{\Delta}$. Similar to procedure (21), it follows that

$$\begin{aligned} LV_n = & LV_{n-1} \\ & + z_n^3 \left(f_n + g_n(H(v) - u^0) + \Delta - L\alpha_{n-1} \right) + \frac{3}{2} \\ & \cdot z_n^2 \left(\psi_n - \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \psi_j \right)^T \\ & \cdot \left(\psi_n - \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \psi_j \right) - \frac{1}{\lambda_n} \bar{\theta}_n \dot{\hat{\theta}}_n - \frac{1}{\sigma} \tilde{\Delta} \dot{\tilde{\Delta}}. \end{aligned} \quad (69)$$

It is noticed that

$$\begin{aligned} & \frac{3}{2} z_n^2 \left\| \psi_n - \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \psi_j \right\|^2 \\ & \leq \frac{3}{4} l_n^2 + \frac{3}{4} l_n^{-2} z_n^4 \left\| \psi_n - \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \psi_j \right\|^4, \end{aligned} \quad (70)$$

where l_n is a positive constant. Based on (7), the the following inequality holds

$$\begin{aligned} g_n z_n^3 (H(v) - u^0) = & g_n z_n^3 \mu_1 (v - v^0) \\ & + g_n z_n^3 \mu_2 (\varsigma - \varsigma^0). \end{aligned} \quad (71)$$

As [21], an even Nussbaum-type function $N(\zeta) = \zeta^2 \cos \zeta$ is defined, and the following equality holds:

$$v = -N(\zeta) \bar{v} + v^0, \quad (72)$$

$$\dot{\zeta} = -\gamma \bar{v} z_n^3, \quad (73)$$

where \bar{v} is the auxiliary virtual controller and γ is a positive parameter. Then, the following equality can be obtained:

$$\begin{aligned} g_n z_n^3 \mu_1 (v - v^0) = & z_n^3 g_n \mu_1 (-N(\zeta) \bar{v} + v^0 - v^0) \\ = & -z_n^3 g_n \mu_1 N(\zeta) \bar{v} - z_n^3 \bar{v} + z_n^3 v^0 \\ = & -z_n^3 (g_n \mu_1 N(\zeta) + 1) \bar{v} + z_n^3 v^0. \end{aligned} \quad (74)$$

According to Lemma 4 and (8), it follows that

$$g_n z_n^3 \mu_2 (\varsigma - \varsigma^0) \leq \frac{3}{4} b_M \mu_2^{4/3} z_n^4 + \frac{1}{2} b_M (\beta + \chi)^{-4/n}. \quad (75)$$

Substituting (73)-(75) into (71), we have

$$\begin{aligned} g_n z_n^3 (H(v) - u^0) &\leq \frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} + z_n^3 \bar{v} \\ &\quad + \frac{3}{4} b_M \mu_2^{4/3} z_n^4 \\ &\quad + \frac{1}{2} b_M (\beta + \chi)^{-4/n}. \end{aligned} \quad (76)$$

According to (65) with $(i = n - 1)$, (70), and (76), (69) can be rewritten as follows:

$$\begin{aligned} LV_n &\leq -\sum_{j=1}^{n-1} \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^{n-1} \varrho_j + z_n^3 \left(f_n \right. \\ &\quad \left. + \frac{1}{4} b_M z_n + \bar{v} + \frac{3}{4} b_M \mu_2^{4/3} z_n - L\alpha_{n-1} + \Delta \right. \\ &\quad \left. + \frac{3}{4} l_n^2 z_n \left\| \psi_n - \sum_{j=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_j} \psi_j \right\|^4 \right) + \frac{1}{\gamma} (g_n \mu_1 N(\zeta) \\ &\quad + 1) \dot{\zeta} + \frac{1}{2} b_M (\beta + \chi)^{-4/n} + \frac{3}{4} l_n^2 - \frac{1}{\lambda_n} \bar{\theta}_n \dot{\theta}_n - \frac{1}{\sigma} \\ &\quad \cdot \bar{\Delta}_n \dot{\hat{\Delta}}. \end{aligned} \quad (77)$$

Define a function as $\bar{f}_n = (1/4)b_M z_n + f_n + (3/4)b_M \mu_2^{4/3} z_n + \Delta - L\alpha_{n-1} + (3/4)l_n^2 z_n \|\psi_n - \sum_{j=1}^{n-1} (\partial \alpha_{n-1} / \partial x_j) \psi_j\|^4$. Furthermore, (77) can be rewritten as

$$\begin{aligned} LV_n &\leq -\sum_{j=1}^{n-1} \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^{n-1} \varrho_j + z_n^3 \bar{f}_n + z_n^3 \bar{v} \\ &\quad + \frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} + \frac{1}{2} b_M (\beta + \chi)^{-4/n} \\ &\quad + \frac{3}{4} l_n^2 - \frac{1}{\lambda_n} \bar{\theta}_n \dot{\theta}_n - \frac{1}{\sigma} \bar{\Delta}_n \dot{\hat{\Delta}}. \end{aligned} \quad (78)$$

For any positive constant $\varepsilon_n > 0$, the fuzzy logic system $\Phi_n^T \xi_n(X_n)$ existed, such that

$$\bar{f}_n = \Phi_n^T \xi_n(X_n) + \delta_n(X_n), \quad |\delta_n(X_n)| \leq \varepsilon_n, \quad (79)$$

Similarly, we can obtain

$$z_n^3 \bar{f}_n \leq \frac{z_n^6}{2a_n^2} \theta_n \xi_n^T \xi_n + \frac{1}{2} a_n^2 + \frac{3}{4} z_n^4 + \frac{1}{4} \varepsilon_n^4, \quad (80)$$

where a_n is a positive parameter. We choose the following virtual control signal and adaptive law:

$$\bar{v} = -\left(k_n + \frac{3}{4}\right) z_n - \frac{1}{2a_n^2} \bar{\theta}_n z_n^3 \xi_n^T \xi_n, \quad (81)$$

$$\dot{\hat{\theta}}_n = \frac{\lambda_n}{2a_n^2} z_n^6 \xi_n^T \xi_n - \gamma_n \hat{\theta}_n, \quad (82)$$

$$\dot{\hat{\Delta}} = -\sigma_d \hat{\Delta}, \quad (83)$$

where k_n , γ_n , and σ_d are positive constants. Substituting (80)-(83) into (78), we have

$$\begin{aligned} LV_n &\leq -\sum_{j=1}^{n-1} \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^{n-1} \varrho_j \\ &\quad + \frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} + \frac{1}{2} b_M (\beta + \chi)^{-4/n} \\ &\quad + \frac{3}{4} l_n^2 + \frac{1}{2} a_n^2 + \frac{1}{4} \varepsilon_n^4 - k_n z_n^4 + \frac{\gamma_n}{\lambda_n} \bar{\theta}_n \dot{\theta}_n + \frac{\sigma_d}{\sigma} \bar{\Delta}_n \dot{\hat{\Delta}}. \end{aligned} \quad (84)$$

Furthermore,

$$\begin{aligned} \frac{\gamma_n}{\lambda_n} \bar{\theta}_n \dot{\theta}_n &\leq -\frac{\gamma_n}{2\lambda_n} \bar{\theta}_n^2 + \frac{\gamma_n}{2\lambda_n} \theta_n^2, \\ \frac{\sigma_d}{\sigma} \bar{\Delta}_n \dot{\hat{\Delta}} &\leq -\frac{\sigma_d}{2\sigma} \bar{\Delta}^2 + \frac{\sigma_d}{2\sigma} \Delta^2. \end{aligned} \quad (85)$$

The above inequality can be rewritten as

$$\begin{aligned} LV_n &\leq -\sum_{j=1}^n \left(k_j z_j^4 + \frac{\gamma_j}{2\lambda_j} \bar{\theta}_j^2 \right) + \sum_{j=1}^n \varrho_j - \frac{\sigma_d}{2\sigma} \bar{\Delta}^2 \\ &\quad + \frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} + \frac{1}{2} b_M (\beta + \chi)^{-4/n}, \end{aligned} \quad (86)$$

where $\varrho_j = (\gamma_j/2\lambda_j)\theta_j^2 + (3/4)l_j^2 + (1/4)\varepsilon_j^4 + (1/2)a_j^2$, $j = 1, 2, \dots, n-1$, and $\varrho_n = (\gamma_n/2\lambda_n)\theta_n^2 + (3/4)l_n^2 + (1/4)\varepsilon_n^4 + (1/2)a_n^2 + (\sigma_d/2\sigma)\Delta^2$.

The control design of the adaptive fuzzy logic system has been completed by using the backstepping technique. The main theorem is described below.

Theorem 10. Consider the stochastic pure-feedback nonlinear system (6) with Assumptions 7–9. For bounded initial conditions, combine with the virtual control signal and the adaptation law (60)-(61) and (81)-(83) guarantee that

- (i) all the signals of the closed-loop system are semi-globally uniformly bounded on $[0, t_f)$, $\forall t_f > 0$;
- (ii) the steady-state tracking errors z_j converge to Ω_Z in the sense of mean quartic value, which is defined as

$$\Omega_Z = \left\{ z_j \mid \sum_{j=1}^n E[|z_j|^4] \leq 4\rho \right\}, \quad (87)$$

where ρ is defined in (97).

Proof. (i) Let Lyapunov function as $V = V_n$, defining $c = \min\{\sigma_d, 4k_j, \gamma_j, j = 1, 2, \dots, n\}$ and $d = \sum_{j=1}^n \varrho_j + (1/2)b_M(\beta + \chi)^{-4/n}$. According to (86), it follows that

$$LV \leq -cV + \frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} + d, \quad t \geq 0. \quad (88)$$

Multiplying V by e^{ct} and based on $It\hat{o}$ formula, one has

$$d(e^{ct}V) = e^{ct}(cV + LV)dt + e^{ct}M(t)dw, \quad (89)$$

where

$$M(t) = \frac{\partial V}{\partial z_1} \psi_1^T(x_1) + \sum_{i=2}^n \frac{\partial V}{\partial z_i} \left(\psi_i(\bar{x}_i) - \sum_{j=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_j} \psi_i(\bar{x}_i) \right)^T. \quad (90)$$

According to (88)-(89), the following inequality can be obtained:

$$d(e^{ct}V) \leq e^{ct} \left(\frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} + d \right) dt + e^{ct} M(t) dw. \quad (91)$$

Integrating (91) on $[0, t]$, one has

$$e^{ct}V(t) - V(0) \leq \int_0^t e^{c\tau} \frac{1}{\gamma} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau + \frac{d}{c} (e^{ct} - 1) + \int_0^t e^{c\tau} M(\tau) dw. \quad (92)$$

Furthermore,

$$V(t) \leq e^{-ct} \frac{1}{\gamma} \int_0^t e^{c\tau} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau + e^{-ct} V(0) + \frac{d}{c} (1 - e^{-ct}) + e^{-ct} \int_0^t e^{c\tau} M(\tau) dw, \quad (93)$$

$$\forall t \in [0, t_f].$$

Let $\xi(t) = g_n \mu_1$; according to Assumption 7 and the definition of μ_1 , we have $0 < |\xi(t)| \leq b_M |\mu_1| < +\infty$. Thus, according to the boundedness of $\xi(t)$ and Lemma 3, we can get that $V(t)$ and ζ and $\int_0^t (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau$ are bounded on $[0, t_f]$.

Next according to the definition of $V(t)$, we can get that z_j and $\hat{\theta}_j$ and $\hat{\Delta}$ are bounded on $[0, t_f]$. Thereby, $\hat{\theta}_j$ and $\hat{\Delta}$ are also bounded on $[0, t_f]$. Due to $z_1 = x_1 - y_d$ and y_d being bounded, we can get that x_1 is bounded. Based on α_1 is the function of z_1 and $\hat{\theta}_1$, thus $x_2 = z_2 + \alpha_1$ is also bounded. Furthermore, we know that $x_j, j = 3, 4, \dots, n$ is bounded. Thus, all the signals of the closed-loop system are semiglobally uniformly bounded on $[0, t_f], \forall t_f > 0$.

(ii) Define $\beta = \sup(1/\gamma) \int_0^t (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau$.

$$\begin{aligned} & e^{-ct} \frac{1}{\gamma} \int_0^t e^{c\tau} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau \\ &= \frac{1}{\gamma} \int_0^t e^{c(\tau-t)} (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau \\ &\leq \frac{1}{\gamma} \int_0^t (g_n \mu_1 N(\zeta) + 1) \dot{\zeta} d\tau \leq \beta, \quad t \geq 0. \end{aligned} \quad (94)$$

By taking expectation on (93) and applying (94), $E[w] = 0$, we can get

$$E[V(t)] \leq \beta + e^{-ct} E[V(0)] + \frac{d}{c} (1 - e^{-ct}), \quad t \geq 0, \quad (95)$$

and therefore,

$$E[V(t)] \leq \beta + \frac{d}{c}, \quad t \rightarrow +\infty. \quad (96)$$

Let

$$\rho = \beta + \frac{d}{c}, \quad (97)$$

and further, due to $V = V_n$ in the definition of (68), we have

$$E \left(\sum_{j=1}^n z_j^4 \right) \leq 4E[V(t)] \leq 4\rho, \quad t \rightarrow +\infty. \quad (98)$$

Thus, the steady-state tracking errors z_j converge to Ω_Z in the sense of mean quartic value. \square

4. Simulation Example

In this section, a simulation example is given to prove the effectiveness of the proposed adaptive control scheme.

Example. Consider a second-order pure-feedback stochastic system with unknown hysteresis and external disturbance

$$\begin{aligned} dx_1 &= (x_1^2 + 5x_2 + 0.5 \sin x_1^2) dt + \sin x_1 dw, \\ dx_n &= (x_2^2 \sin x_2^2 + 3u + \sin t^2) dt + \cos(x_1 x_2) dw, \\ y &= x_1, \end{aligned} \quad (99)$$

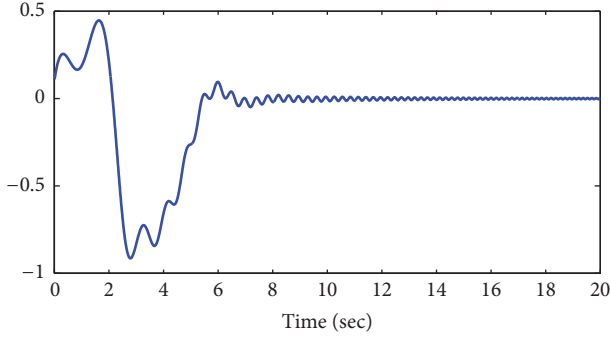
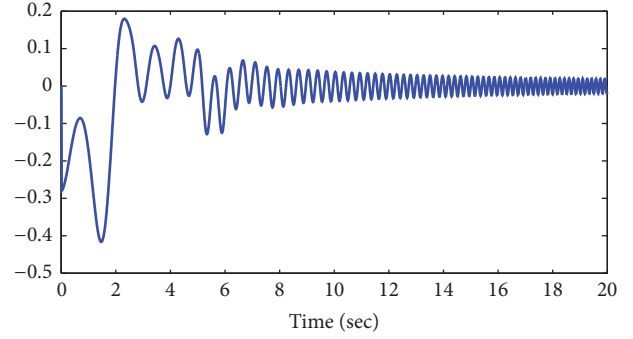
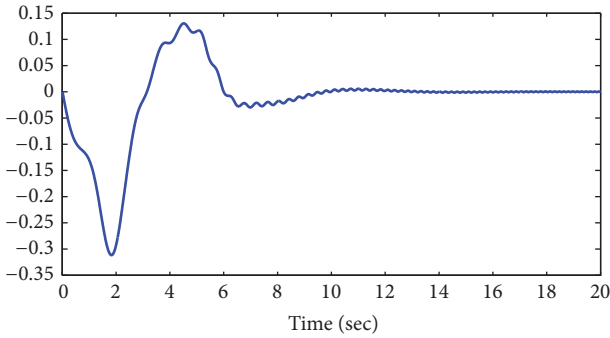
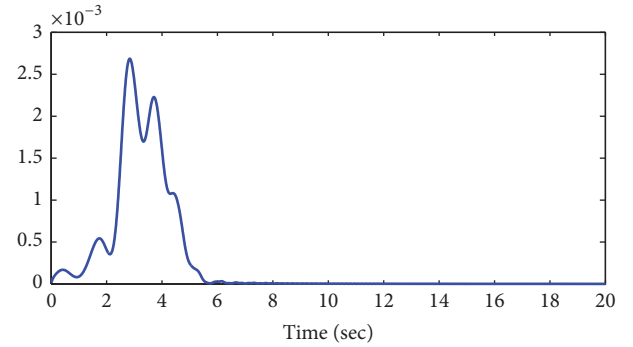
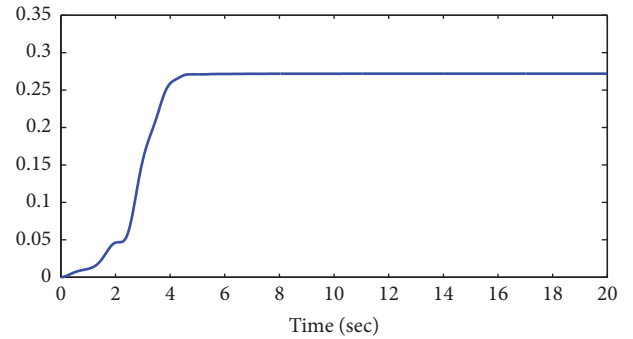
where x_1 and x_2 are the state variable, t is the time variable, y is the system output, and u is the unknown hysteresis output. These hysteresis parameters are chosen as $\mu_1 = 0.5$, $\mu_2 = 1.2$, $\beta = 1.5$, $\chi = 0.8$, and $n = 2$. It is obvious that (99) satisfies the assumptions of 7-9. The system control target is that the system output y is tracked to the reference signal $y_d = 0.6 \sin(0.5x) + 0.5 \sin(x)$.

For all state variables, the fuzzy sets are defined on the interval $[-1.5, 1.5]$, the fuzzy membership functions are as follows:

$$\begin{aligned} \mu_{F_1^1}(x) &= \exp(-0.5(x + 1.5)^2), \\ \mu_{F_1^2}(x) &= \exp(-0.5(x + 1)^2), \\ \mu_{F_1^3}(x) &= \exp(-0.5(x + 0.5)^2), \\ \mu_{F_1^4}(x) &= \exp(-0.5x^2), \\ \mu_{F_1^5}(x) &= \exp(-0.5(x - 0.5)^2), \\ \mu_{F_1^6}(x) &= \exp(-0.5(x - 1)^2), \\ \mu_{F_1^7}(x) &= \exp(-0.5(x - 1.5)^2). \end{aligned} \quad (100)$$

The virtual control signal and the adaptive law are selected as follows:

$$\alpha_1 = - \left(k_1 + \frac{3}{4} \right) z_1 - \frac{1}{2a_1^2} \hat{\theta}_1 z_1^3 \xi_1^T \xi_1 + x_2^0,$$

FIGURE 1: Tracking error $z_1 = y - y_d$.FIGURE 3: The true control input \bar{v} .FIGURE 2: State variable x_2 .FIGURE 4: The adaptive parameter $\hat{\theta}_2$.FIGURE 5: Estimation of disturbance bound $\hat{\Delta}$.

$$\begin{aligned}
 \dot{\hat{\theta}}_1 &= \frac{\lambda_1}{2a_1^2} z_1^6 \xi_1^T \xi_1 - \gamma_1 \hat{\theta}_1, \quad \hat{\theta}_1(0) \geq 0, \\
 \bar{v} &= -\left(k_2 + \frac{3}{4}\right) z_2 - \frac{1}{2a_2^2} \hat{\theta}_2 z_2^3 \xi_2^T \xi_2, \\
 \dot{\hat{\theta}}_2 &= \frac{\lambda_2}{2a_2^2} z_2^6 \xi_2^T \xi_2 - \gamma_2 \hat{\theta}_2, \\
 \dot{\hat{\Delta}} &= -\sigma_d \hat{\Delta},
 \end{aligned} \tag{101}$$

where $z_1 = x_1 - y_d$, $z_2 = x_2 - \alpha_1$. The Nussbaum function is chosen as $N(\zeta) = \zeta^2 \cos \zeta$; its control law is $v = -N(\zeta)\bar{v} + v^0$, $\dot{\zeta} = -\gamma\bar{v}z_2^3$. The control parameters are designed as $k_1 = k_2 = 10$, $a_1 = a_2 = 2$, $\lambda_1 = \lambda_2 = 8$, $\gamma_1 = \gamma_2 = 0.5$, $\gamma = 8$, $\sigma_d = 0.25$, and $x_2^0 = v^0 = 0$. The initial condition is selected as $[x_1(0), x_2(0)]^T = [0.1, 0.2]^T$, $\hat{\Delta}(0) = 0$, and $[\hat{\theta}_1(0) = \hat{\theta}_2(0)]^T = [0, 0]^T$. The simulation results are shown in Figures 1–5. Figure 1 shows the tracking error $z_1 = y - y_d$. Figure 2 shows the state variable x_2 . Figure 3 shows the true control input \bar{v} . Figure 4 presents the adaptive parameter $\hat{\theta}_2$. Figure 5 presents estimation of disturbance bound $\hat{\Delta}$.

5. Conclusion

In this paper, we study a class of stochastic pure-feedback nonlinear systems with bounded external disturbance and

unknown hysteresis. This paper holds that stochastic disturbance and external disturbance exist simultaneously. By using the characteristics of the Nussbaum function, the unknown hysteresis problem is solved. Based on the approximation ability of fuzzy logic system, a new adaptive fuzzy control scheme is proposed. The scheme ensures that all signals of the closed-loop system are bounded and the tracking error converges to small domain of the origin. Finally, the simulation results show the effectiveness of the proposed scheme.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61573227), Shandong Provincial Natural Science Foundation (nos. ZR2018MF013, ZR2016FM48), the Research Fund for the Taishan Scholar Project of Shandong Province of China, SDUST Research Fund (no. 2015TDJH105), and the Fund for Postdoctoral Application Research Project of Qingdao (01020120607).

References

- [1] J. Zhou, C. Wen, and T. Li, "Adaptive output feedback control of uncertain nonlinear systems with hysteresis nonlinearity," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2627–2633, 2012.
- [2] M.-C. Pai, "Chaos control of uncertain time-delay chaotic systems with input dead-zone nonlinearity," *Complexity*, vol. 21, no. 3, pp. 13–20, 2015.
- [3] M. Wang, S. Zhang, B. Chen, and F. Luo, "Direct adaptive neural control for stabilization of nonlinear time-delay systems," *Science China Information Sciences*, vol. 53, no. 4, pp. 800–812, 2010.
- [4] H. Wang, P. X. Liu, and S. Liu, "Adaptive neural synchronization control for bilateral teleoperation systems with time delay and backlash-like hysteresis," *IEEE Transactions on Cybernetics*, 2017.
- [5] C.-Y. Su, Q. Wang, X. Chen, and S. Rakheja, "Adaptive variable structure control of a class of nonlinear systems with unknown Prandtl-Ishlinskii hysteresis," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 2069–2074, 2005.
- [6] G. Gu, L. Zhu, and C. Su, "Modeling and Compensation of Asymmetric Hysteresis Nonlinearity for Piezoceramic Actuators With a Modified Prandtl-Ishlinskii Model," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1583–1595, 2014.
- [7] C. Hua and Y. Li, "Output feedback prescribed performance control for interconnected time-delay systems with unknown Prandtl-Ishlinskii hysteresis," *Journal of The Franklin Institute*, vol. 352, no. 7, pp. 2750–2764, 2015.
- [8] H. Wang, P. X. Liu, S. Li, and D. Wang, "Adaptive Neural Output-Feedback Control for a Class of Nonlower Triangular Nonlinear Systems With Unmodeled Dynamics," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.
- [9] X. Zhao, X. Wang, G. Zong, and H. Li, "Fuzzy-approximation-based adaptive output-feedback control for uncertain nonsmooth nonlinear systems," *IEEE Transactions on Fuzzy Systems*, pp. 1–12, 2018.
- [10] F. Wang, B. Chen, C. Lin, J. Zhang, and X. Meng, "Adaptive Neural Network Finite-Time Output Feedback Control of Quantized Nonlinear Systems," *IEEE Transactions on Cybernetics*, 2017.
- [11] F. Wang, B. Chen, Y. Sun, and C. Lin, "Finite time control of switched stochastic nonlinear systems," *Fuzzy Sets and Systems*, 2018.
- [12] H. Wang, K. Liu, X. Liu, B. Chen, and C. Lin, "Neural-based adaptive output-feedback control for a class of nonstrict-feedback stochastic nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1977–1987, 2015.
- [13] X. Liu, Y. Li, and W. Zhang, "Stochastic linear quadratic optimal control with constraint for discrete-time systems," *Applied Mathematics and Computation*, vol. 228, pp. 264–270, 2014.
- [14] H. Li, L. Bai, Q. Zhou, R. Lu, and L. Wang, "Adaptive fuzzy control of stochastic nonstrict-feedback nonlinear systems with input saturation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2185–2197, 2017.
- [15] H. Q. Wang, B. Chen, K. F. Liu, X. P. Liu, and C. Lin, "Adaptive neural tracking control for a class of nonstrict-feedback stochastic nonlinear systems with unknown backlashlike hysteresis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 947–958, 2014.
- [16] H. Wang, B. Chen, and C. Lin, "Adaptive fuzzy control for pure-feedback stochastic nonlinear systems with unknown dead-zone input," *International Journal of Systems Science*, vol. 2013, Article ID 773470, 2013.
- [17] Y. Gao, S. Tong, and Y. Li, "Fuzzy adaptive output feedback DSC design for SISO nonlinear stochastic systems with unknown control directions and dead-zones," *Neurocomputing*, vol. 167, pp. 187–194, 2015.
- [18] Y.-J. Liu and S. C. Tong, "Adaptive fuzzy identification and control for a class of nonlinear pure-feedback MIMO systems with unknown dead zones," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1387–1398, 2015.
- [19] Q. Zhou, P. Shi, S. Xu, and H. Li, "Observer-based adaptive neural network control for nonlinear stochastic systems with time delay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 71–80, 2013.
- [20] Z. X. Yu, S. G. Li, and H. B. Du, "Razumikhin-Nussbaum-lemma-based adaptive neural control for uncertain stochastic pure-feedback nonlinear systems with time-varying delays," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 11, pp. 1214–1239, 2013.
- [21] F. Wang, Z. Liu, Y. Zhang, and C. L. P. Chen, "Adaptive fuzzy control for a class of stochastic pure-feedback nonlinear systems with unknown hysteresis," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 140–152, 2016.
- [22] X. Zhang, X. Liu, and Y. Li, "Adaptive fuzzy tracking control for nonlinear strict-feedback systems with unmodeled dynamics via backstepping technique," *Neurocomputing*, vol. 235, pp. 182–191, 2017.
- [23] N. Wang, J.-C. Sun, and Y.-C. Liu, "Direct adaptive self-structuring fuzzy control with interpretable fuzzy rules for a class of nonlinear uncertain systems," *Neurocomputing*, vol. 173, pp. 1640–1645, 2016.
- [24] Y. Hou and S. Tong, "Adaptive fuzzy output-feedback control for a class of nonlinear switched systems with unmodeled dynamics," *Neurocomputing*, vol. 168, pp. 200–209, 2015.
- [25] G. Nagamani, S. Ramasamy, and P. Balasubramaniam, "Robust dissipativity and passivity analysis for discrete-time stochastic neural networks with time-varying delay," *Complexity*, vol. 21, no. 3, pp. 47–58, 2016.
- [26] N. Wang, J.-C. Sun, and M. J. Er, "Tracking-error-based universal adaptive fuzzy control for output tracking of nonlinear systems with completely unknown dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 869–883, 2018.

- [27] B. Chen, X. Liu, K. Liu, and C. Lin, "Direct adaptive fuzzy control of nonlinear strict-feedback systems," *Automatica*, vol. 45, no. 6, pp. 1530–1535, 2009.
- [28] Z. Liu, F. Wang, Y. Zhang, X. Chen, and C. L. P. Chen, "Adaptive tracking control for a class of nonlinear systems with a fuzzy dead-zone input," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 193–204, 2015.
- [29] B. Ren, S. S. Ge, C. Su, and T. H. Lee, "Adaptive neural control for a class of uncertain nonlinear systems in pure-feedback form with hysteresis input," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 431–443, 2009.
- [30] S. Tong, Y. Li, Y. Li, and Y. Liu, "Observer-based adaptive fuzzy backstepping control for a class of stochastic nonlinear strict-feedback systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 6, pp. 1693–1704, 2011.
- [31] S.-J. Liu, J.-F. Zhang, and Z.-P. Jiang, "Decentralized adaptive output-feedback stabilization for large-scale stochastic nonlinear systems," *Automatica*, vol. 43, no. 2, pp. 238–251, 2007.
- [32] S. S. Ge, F. Hong, and T. H. Lee, "Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 499–516, 2004.
- [33] H. Wang, B. Chen, X. Liu, K. Liu, and C. Lin, "Robust adaptive fuzzy tracking control for pure-feedback stochastic nonlinear systems with input constraints," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2093–2104, 2013.

Research Article

Interface Data Modeling to Detect and Diagnose Intersystem Faults for Designing and Integrating System of Systems

Kyung-Min Seo  and **Kwang-Phil Park**

Naval and Energy System R&D Institute, Daewoo Shipbuilding and Marine Engineering, Seoul 04521, Republic of Korea

Correspondence should be addressed to Kyung-Min Seo; kmseo.kumsung@gmail.com

Received 25 April 2018; Revised 16 July 2018; Accepted 1 August 2018; Published 14 October 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Kyung-Min Seo and Kwang-Phil Park. 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.

In system of systems engineering, system integrators are in charge of compatible and reliable interfaces between subsystems. This study explains a systematic solution to identify and diagnose interface faults during designing and integrating systems of systems. Because the systems targeted in this study are real underwater vessels, we first have anatomized 188 interface data transferred between 22 subsystems of them. Based on this, two interface data models are proposed, which include data sets regarding messages and inner fields and transition and decision functions for them. Specifically, a structure model at the message level evaluates how inner fields belong to a message, and a logic model at the field level assesses how each field is interpreted and if the interpreted value is understandable. The software that supports the modeling is implemented using the following concepts: (1) a model-view-viewmodel pattern for overall software design and (2) a computer network for representing sequential properties of field interpretations. The proposed modeling and software facilitate diagnostic decisions by checking the consistency between interface protocols and obtained real data. As a practical use, the proposed work was applied to an underwater shipbuilding project. Within 10 interfaces, 14 fault cases were identified and diagnosed. They were gradually resolved during the system design and integration phases, which formed the basis of successful submarine construction.

1. Introduction

A complex system such as automotive, marine, or aerospace system of systems (SoS) contains diverse subsystems that must be designed and integrated to work together [1–3]. In an underwater vessel, for example, an inertial navigation system (INS) receives speed over water and locational information from an electromagnetic log (EM log) and Global Positioning System (GPS), respectively [4]. They enable the INS to enhance the computational accuracy of its orientation and velocity. In this context, the input data of the INS have been utilized for computing the INS's outputs precisely; besides, the outputs also are a basis of estimating the geographic location of the vessel in water. Thus, understandable and reliable interfaces between the subsystems are the main prerequisite to organize the subsystems as an integrated system at the corporate level [5–7].

When designing and integrating the complex SoS, a system integrator has difficulty figuring out interface faults for the following reasons. First, both subsystems of an interface are commonly developed by different manufacturers, which gives rise to disparate implementations of the same interface protocols [8]. Furthermore, for easy modifiability and scalability, the manufacturers still prefer customized protocols to those that are standardized [9, 10]. In this regard, the protocols are occasionally revised during the system design phase as well as the integration phase.

This study suggests a systematic solution about how a subsystem successfully interacts with a counterpart one when they are designed and integrated for the whole system. Specifically, we have focused on resolving the interface faults (i.e., anomalies) during node-to-node delivery over the digital network. Our goal in this study is to ensure compatible and reliable interfaces by checking the

consistency between the interface protocols and obtained interface data.

In order to transfer sensitive information, a sending system encodes interface data (i.e., messages) so only authorized receivers can understand [11]. The encoding rules contain what payloads (i.e., fields in this study) are structured in a message or how each field is logically converted, which are described in the interface protocols [12]. Because the protocols including the structural and logical rules are diverse and complicated, they should be preferentially explored for overcoming the interface faults. To this end, we have analyzed interface protocols used in domestic ship systems that are already in operation or under construction. Messages transmitted via radar, navigation, acoustic, and optical sensor systems as well as several control systems were investigated in this study. We atomized 188 message types that interacted between 22 subsystems for message structures and field logics.

Based on the preanalysis, we proposed two modeling formulas, which contain data sets and functions [13]. A structure model at the message level formalizes how many fields belong to a message and what makes it to be structured, and a logic model at the field level assesses how each field is interpreted and whether the interpreted value of the field is understandable or not for receiving systems. The models fundamentally receive interface data as input. Then, they detect and diagnose the faults via the transition functions and output the results through the decision functions. In the proposed modeling, we practically classified five structure types at the message level and modularized several transition functions at the field level. With the proposed formalism, a modeler can specify the interface protocols and diagnose the faults containing messages and inner fields in a systematic rather than an ad hoc manner [14–16].

Over the last decade, several studies for fault detection and diagnosis methods have been developed for various systems and applications. Some researchers have developed system models for representing real systems by checking model-predicted outputs and obtained system outputs [17–19], and others have centered on output signals of the systems to analyze their features or patterns for fault detection and diagnosis [20–22]. This study combined these two methods. We focused on input/output (I/O) signals within digital interfaces; at the same time, the signals are explicitly formalized in the two-level models to detect and diagnose interface faults of an arbitrary interface. To the best of our knowledge, no work has been reported toward focusing on fault detection and diagnosis during the system design and integration phases.

To realize the proposed models in an effective way, we have used the model-view-viewmodel (MVVM) design pattern in Windows Presentation Foundation (WPF) technology [23]. In the developed software, block libraries for modeling elements have been provided to illustrate the benefits of a graphical modeling environment. In addition, a computer network concept has been applied, which is based on the concept of using nodes and connections to create an overall logic modeling. Thus, it facilitates intuitive modeling

via libraries regarding structural delimiters and logical operations and allows flexible modeling through their creation and revision.

As a practical use, the proposed work was applied to an underwater shipbuilding project, namely, a submarine renovation project [24]. Ten digital interfaces connected to improved subsystems were examined to resolve the interface faults. Seven tests were performed at sea to find the faults for various operational situations, and two tests that allow the ideal preparation for sea-trial tests were conducted in a harbor. The empirical results showed that 14 fault cases, which are either structural or logical, were detected and diagnosed during designing and integrating the renovated submarine. These incorrect patterns in the interfaces were successfully resolved during this project.

The study is organized as follows. Section 2 describes our focus of fault scope, and Section 3 analyzes previous works. Section 4 proposes modeling methods and realization of the modeling as a software tool. Section 5 explains and discusses an application for the shipbuilding project. Finally, Section 6 presents our conclusions.

2. Background

A fault is defined as an unpermitted deviation of at least one characteristic property or parameter of a system from the acceptable, usual, or standard condition [25]. Because the various cases of faults can occur when a system is under development as well as in operation, the fault scope interested in this study needs to be clarified here.

2.1. Interface Faults in Complex System Development.

Figure 1 shows a simplified illustration of how a fault is identified and diagnosed in complex shipbuilding engineering. As explained in the introduction, the basic concept for resolving the fault is to evaluate the consistency by comparing with the interface data and the interface protocol including structural and logical rules.

The ship system as an SoS is generally composed of diverse sensing equipment and dynamic systems, which are incorporated into an integrated system [26, 27]. It has been noted that the majority of end systems such as sensors or actuators do not plug directly into the central network [28]. Instead, they connect to a local proxy with each point-to-point link, which in turn is distributed across a central bus network. The signal-processing unit, data integration system (DIS), and integrated management system in Figure 1 act as such proxies.

When designing and integrating the subsystems for the overall system, faults can be found in the central network as well as the outside of the network, specifically in the point-to-point links between the local subsystems [29]. In this study, we focused on the local faults rather than the central faults for the following reasons. First, the local faults occur more frequently than the central faults due to disparate implementation of the same protocol. This problem accords with the current industrial tendency that interoperability testing for

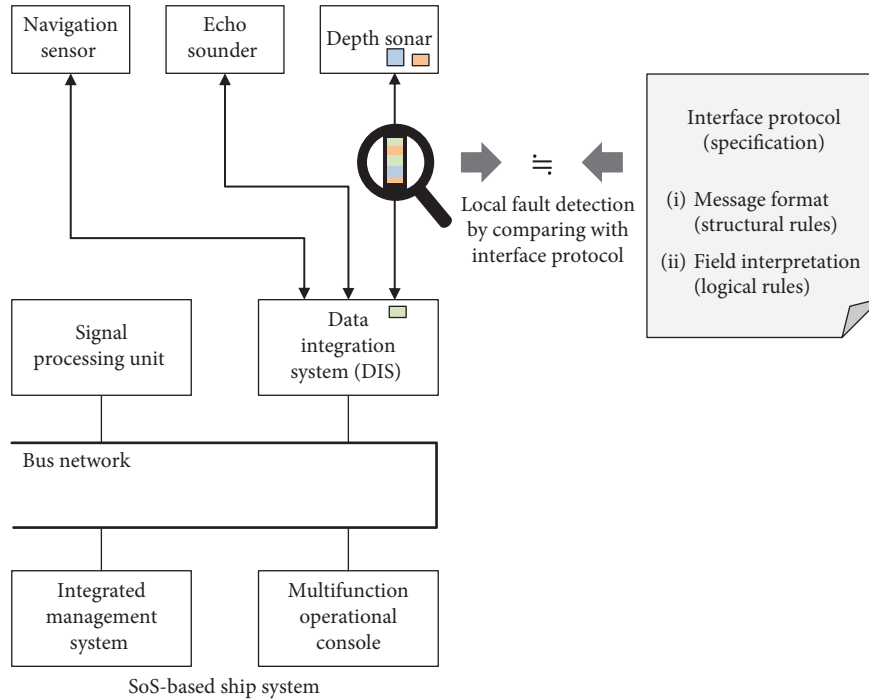


FIGURE 1: Local faults in SoS-based ship system.

communication between the connected systems becomes more important [30, 31].

Next, the local faults need to be preferentially identified, because these failures obviously influence the central part [32]. For example, if an EM log sends speed information with the wrong unit or resolution, other subsystems using the speed (e.g., navigation sensor or echo sounder) have abnormal behaviors sequentially. Thus, the local faults often have been ascribed to uncontrolled, unanticipated, and unwanted interactions between the subsystems [33, 34].

In this respect, this study focuses on the local faults during node-to-node delivery over digital interfaces when designing and integrating the SoS. The local faults were classified into structural and logical levels, which are explained in the following subsection.

2.2. Classification of Local Faults. Figure 2 illustrates exchanging digital data between two subsystems. In digital communication, the data are a sequential stream of bytes at the physical layer [35]. In this study, we assumed that the byte stream was already transformed into manageable data units (i.e., messages).

An individual message has a common format to be distinguished with different types of messages [36]. For example, a message may be determined to have a fixed-length structure or may include several fields for transmitted information as well as supplemental delimiters such as a header and a footer. Since the message is usually encoded for information security, it eventually needs a conversion process that returns into the original sequence of information [36]. These structural and logical rules are comprised in

communication protocol, which have to do with an agreement between both-sided manufacturers and a system integrator.

As shown in Figure 2, the faults in an individual interface data are hierarchically classified into two levels: a structural and a logical fault. The structural fault is incorrectness about the exterior of a message. The wrong length of the message or different delimiters in the message are the structural fault. On the contrary, the logical fault occurs when a field conveyed in the message is semantically incorrect. Uninterpretable data, unmeasurable values, or unspecified status information in the field corresponds to this level.

Because the structural and logical rules are diverse and complicated, they should be preferentially explored to detect and diagnose their faults. To this end, we analyzed interface protocols for real messages transferred in domestic naval vessels. Table 1 shows the preinvestigation regarding two types of real submarine systems. The first-generation submarines have been in operation domestically, and those that are third-generation are under construction. In total, 188 message types in 22 subsystems were anatomized for message formats and field logics. We generalized how the messages were constructed to distinguish from others and which rules were required to interpret the fields for meaningful information.

In summary, this study introduces a practical concept for identifying and diagnosing the structural and the logical faults during the design and integration of the complex SoS. Based on the preanalysis, we formalize a structure model at the message level and a logic model at the field level. The proposed models are implemented to a software tool, which facilitates intuitive and flexible modeling to detect and diagnose the interface faults.

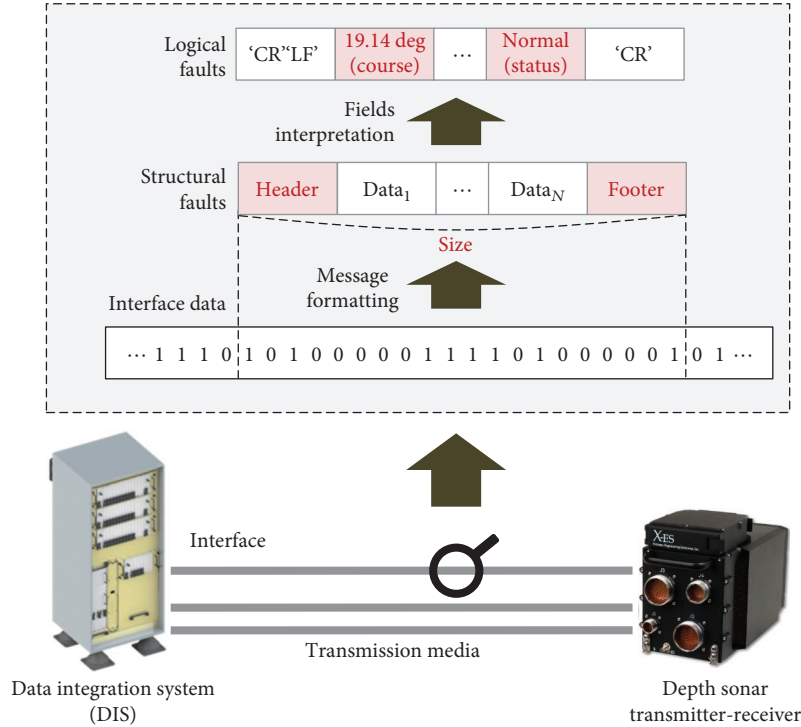


FIGURE 2: Classification of local faults in point-to-point link: structural and logical faults.

TABLE 1: Preinvestigated interface protocols in 2 types of submarines.

Submarine type	Number of subsystems	Number of interfaces	Number of messages
First-generation class	11	12	74
Third-generation class	11	31	114
Total number	22	43	188

Subsystems to be analyzed are (1) radar, navigation, acoustic, and optical sensor systems and (2) control systems such as plotting boards and weapons control systems.

3. Literature Review

Over the last decade, several studies for fault detection and diagnosis have been developed for various systems and applications. In this section, we have classified them into three approaches, which are summarized in Table 2.

In model-based approaches, system models are developed to describe the relationships among main system variables [39–41]. Based on the models, fault diagnosis algorithms have been developed to monitor the consistency between the measured outputs of the practical systems and the model-predicted outputs [42]. For example, Cai et al. [17] used object-oriented Bayesian networks (OOBN) to model complex systems. The OOBN-based modeling is classified into structure and parameter modeling that are built with sensor historical data and expert knowledge. Lamperti and Zhao [18] focused on the diagnosis of active systems, and the diagnosis of rules in the proposed finite system machine (FSM) has been specified based on associations

between context-sensitive faults and regular expressions. Poon et al. [19] used a model-based state estimator to generate an error residual that captures the difference between the measured and estimator outputs. These model-based approaches require explicit models, whose accuracy determines the diagnosis performance.

On the contrary, signal-based approaches decide diagnostic decisions based on features or patterns of the extracted signals rather than the system models [43, 44]. For example, Loza et al. [20] proposed a nonhomogeneous high-order sliding mode observer to estimate sensor signal faults in finite time and in the presence of bounded disturbances. In Do and Chong’s work [21], the vibration signal was translated into an image; the local features were then extracted from the image using scale-invariant feature transform for fault detection and isolation under a pattern classification framework. Pan et al. [22] proposed an acoustic fault detection method, which was addressed for the gearbox based on the improved frequency-domain blind deconvolution flow. The signal-based approaches generally extract the major features of the output signals for fault diagnosis, but they pay less attention to system inputs [45].

This study has combined the two approaches. To detect and diagnose interface digital signals, signal patterns under a normal status were generalized, which were known from the interface protocols. Input signals as well as output signals were targeted because the interfaces interested in this study generally had both-sided signals. Then, the patterns were formalized with mathematical models (i.e., interface data models). The proposed models had explicit sets and functions, and the fault diagnosis within the modeling was carried

TABLE 2: Summary of related works.

Approach	Previous work	Motivation	Method	Application
Model-based approach	Cai et al. [17]	To eliminate system faults immediately once they occur in complex systems	Structure and parameter models using object-oriented Bayesian networks are proposed.	Subsea production system
	Lamperti and Zhao [18]	To diagnose component faults in complex discrete-event systems during the system evolution	A finite-state machine model is built for fault diagnosis.	Military system
	Poon et al. [19]	To resolve faults in components and sensors in switching power converters	A model-based state estimator is proposed based on a library of fault signatures for possible component and sensor faults in all 4 converters.	Switching power converter system
Signal-based approach	Loza et al. [20]	To reconstruct signal faults as early as possible	A nonhomogeneous high-order sliding mode-based observation approach is proposed.	Aircraft transport system
	Do and Chong [21]	To detect and diagnose vibration signals of the inductor motor	The scale invariant feature transform algorithm is proposed to generate the faulty symptoms.	Three-phase AC motor
	Pan et al. [22]	To analyze acoustical signal for monitoring normal operation of gearbox	The complex-valued fixed point algorithm was used for frequency domain signal.	Power transfer system
Interoperability testing	Vijayaraghavan et al. [49]	To provide a common means for communication between devices	A data exchange standard was proposed.	Manufacturing system
	Shin et al. [38]	To analyze the operating situations of the systems at the system-integration phase	A message-description language was used to convert the raw interface data into the interpreted data format.	Ship system

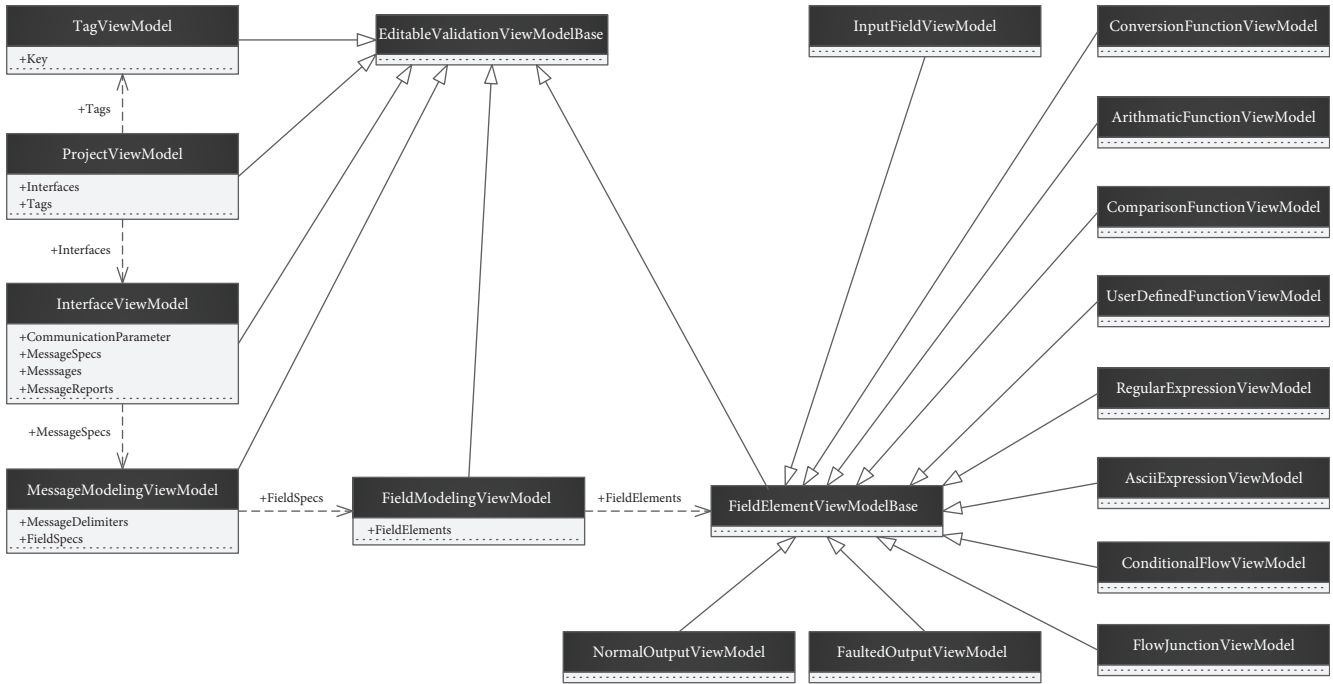


FIGURE 3: Simplified class diagram of developed software for fault detection and diagnosis.

out by checking the consistency between the structural and logical patterns and the measured signals. In short, we focused on the interface signals and developed explicit data models with deterministic criteria [46, 47].

Most of all, the above studies have been utilized in operating the complex systems. For system design and integration phases, the interoperability testing is a similar concept to our approach [37, 48]. For example, Vijayaraghavan et al. [49] proposed an open communication standard for data interoperability. The proposed open protocol provides the mechanism for process and system monitoring and optimization concerning resources. Shin et al. [38], which is similar to that of the present study, involved the development of an analysis tool to confirm the integrated performance of the complex system. To analyze the data, a message-description language was used to convert the raw interface data into the interpreted data format. Despite their practical contributions, however, they cannot diagnose the faults within the interfaces. To the best of our knowledge, no work has been reported toward focusing on fault detection and diagnosis during the system design and integration phases.

4. Proposed Work

4.1. Software Architecture. Having introduced the concept of interface data and their faults, the overall design of the developed software will be introduced in this subsection. The focus of the software is to (1) represent hierarchies of the interface data (i.e., project, interface, message, and field) and (2) realize structure and logic modeling with graphical user interface (GUI).

To provide flexible GUI for modelers, a specific software design pattern was used. The MVVM design pattern in WPF technology facilitates to decouple the GUI from model logic

and data [50, 51]. The model in the MVVM pattern is an implementation of the application's domain model that includes a data model along with business logic, and the view is responsible for defining the layout and appearance of what the user sees on the screen. The view model acts as an intermediary between the view and the model and is responsible for handling the view logic. Because the view model retrieves data from the model and then makes the data available to the view, in this subsection, we will focus on view models to realize the proposed modeling.

A class diagram for major view models of the developed software is described in Figure 3. Two *ViewModelBase* classes serve as base classes for other view model classes. The left part of Figure 3 indicates hierarchies for interface data modeling from a project to a field, and the right part shows specific elements to model a field logically.

The *ProjectViewModel* class takes charge of resolving interface faults for a particular project. After a project is determined, *TagViewModel* manages multiple tests in the project. In the following application section, nine tests for the submarine renovation project were managed by this class. The *InterfaceViewModel* ensures operations regarding evaluating an individual interface (e.g., loading interface data, analyzing them with interface protocols, and visualizing fault results). Therefore, it is mainly composed of the following properties: *Messages* for the interface data, *MessageSpecs* for the interface protocols, and *MessageReports* for the fault reports. Finally, the *MessageModelingViewModel* facilitates structure modeling at the message level, and the *FiledModelingViewModel* enables logic modeling at the field level. On this wise, the architecture fundamentally facilitates a hierarchical modeling: an interface provides the means for an arbitrary number of messages and a message also comprises multiple fields.

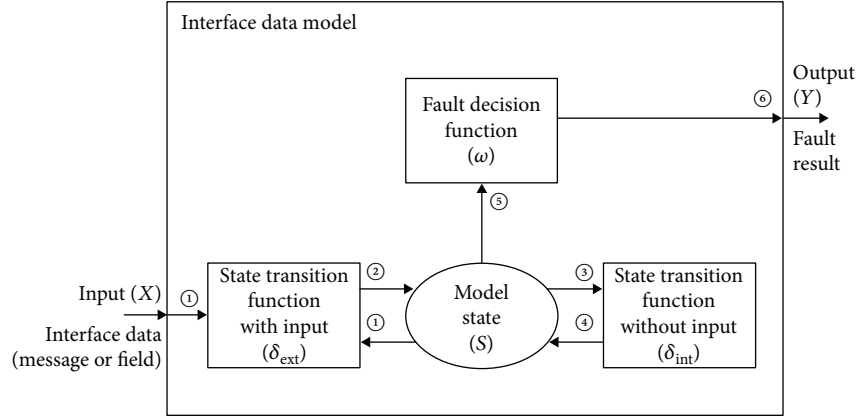


FIGURE 4: Elements of proposed interface data model.

In particular, the logic model at the field level recognizes logical rules and diagnoses faults by comparing received field data to the rules. The logical rules generally contain multiple steps to decode the raw data to interpretable information. Therefore, to model a field in stages, the developed software provides eight elements for the rules, which are illustrated in the right parts of Figure 3 (from *ConversionFunctionView-Model* to *FlowJunctionViewModel*). The remaining view models (i.e., *InputFieldViewModel*, *NormalOutputViewModel*, and *FaultOutputViewModel*) are for inputs and outputs of the logic model. Because these elements have their own views in the MVVM pattern, the developed software offers a graphical modeling approach that helps the modelers visualize every element to model a field. In the following subsections, we explain methodological aspects for structure and logic modeling at the message and the field levels, respectively.

4.2. Structure Modeling at Message Level. Figure 4 illustrates elements of the proposed model that is either a structure or a field model. The model fundamentally receives interface data (X in Figure 4) as input and sends fault results (Y) as output. Inside the model are one total state and three functions. These are depicted with circle and squares in Figure 4. The model state (S) is updated after performing two transition functions, which contain received data, conditions for the rules, and functional results.

If an input occurs, δ_{ext} interprets the input data with the rules and updates S (① and ② in Figure 4). As explained in the previous subsection, the logical rules need multiple interpretations. In this instance, after δ_{ext} , δ_{int} updates S without any input. Note that δ_{int} is carried out sequentially until the interpretations are completed (③ and ④). This is a general situation for logic modeling at the field level, which will be explained in the following subsection. Finally, when all δ_{ext} and δ_{int} are fulfilled, ω decides fault results based on the updated state and outputs the decision (⑤ and ⑥).

The proposed models are derived from the discrete event system specification (DEVS), which is a general mathematical representation for discrete event systems [52–54]. The main difference between the DEVS and our models is targets

to be modeled. The DEVS focuses on the system itself. Thus, it should represent behaviors of the system as time passes. On the contrary, because our models aim at the interface data rather than the system, they have no concept of time. In other words, our models are static in which the output depends on the input at the same time.

According to Figure 4, the proposed structure model is 5-tuple consisting the following:

$$SM_{\text{Message}} = \langle X, Y, S, \delta_{ext}, \omega \rangle$$

X is an input set of n fields comprised of a message, where

$$X = \{(f_i, l_i) \mid 1 \leq i \leq n\},$$

f_i is the value of the i -th field within the message,

l_i is the length of f_i ;

Y is an output set of structural faults, where

$$Y = \{(d_j, r_j) \mid 1 \leq j \leq k, d_j \in D, r_j \in \{true, false\}\},$$

d_j is the j -th delimiter in D ,

r_j is the fault result for d_j ;

$S = D \times R$ is a total state set, where

D is a set of delimiters for structural rules,

$R = \{true, false\}$ is a set of transition results;

$\delta_{ext} : 2^X \times D \rightarrow S$ is the message transition function;

$\omega : S \rightarrow Y$ is the fault decision function.

Components of SM_{Message} are contained within $\langle \rangle$. Every notation in SM_{Message} is based on set theory. For example, $\{ \}$ means a set; \times indicates the Cartesian product (i.e., all possible ordered pairs); 2^X means the power set of X ; and \rightarrow means the function mapping.

The structure model evaluates how many fields belong to a message and what distinguishes the message. Specifically, the external transition function, δ_{ext} , receives all the fields comprising a message and appropriate delimiters.

TABLE 3: Examples of X and D depending on type of message structure.

Structure type	$x \in 2^X$	$d \in D$
Header and footer	(f_1, l_1)	d_{hdr}
	(f_n, l_n)	d_{ftr}
Header and length	(f_1, l_1)	d_{hdr}
	X or (f_i, l_i)	d_{length}
Header	(f_1, l_1)	d_{hdr}
Footer	(f_n, l_n)	d_{ftr}
Length	X or (f_i, l_i)	d_{length}

f_1 and f_n mean the first and last fields, respectively. (f_i, l_i) means a specific field containing the length information of the message.

And it updates the transition result, indicating if the message has a correct format. Because one or more fields are mapped into one delimiter, δ_{ext} needs X in the form of the power set. Note that the internal state transition function is not required in the structure model. Finally, the fault decision function, ω , checks the current state and produces a fault result.

Based on the preinvestigation in Table 1, the message structures were divided into five types, which is shown in Table 3. The structure types are classified depending on how to use the following delimiters: a header, a footer, and a message length. In case of types with the header and the footer, ω checks whether the first and the last fields are satisfied with the header and the footer, respectively. The length is known in two ways: (1) it is computed by adding length of all the fields, or (2) it can be found within a specific field, for example, (f_i, l_i) in Table 3. The structure model is relatively simple to design because it decides the correctness of the message exterior. As explained in the previous subsection, the *MessageModelingViewModel* in Figure 3 realizes the structure model.

The following specifications are a modeling example for a message in the GPS that will be explained in Section 5.3:

$$SM_{GPS-1} = \langle X, Y, S, \delta_{ext}, \omega \rangle$$

$$X = \{(f_1, l_1), (f_2, l_2), (f_3, l_3), (f_4, l_4), (f_5, l_5), (f_6, l_6), (f_7, l_7), (f_8, l_8), (f_9, l_9)\}$$

$$Y = \{(d_{hdr}, r_{hdr}), (d_{ftr}, r_{ftr})\};$$

$$S = \{d_{hdr}, d_{ftr}\} \times \{true, false\}, \text{ where}$$

$$d_{hdr} = 0x3A58 \text{ as a hexadecimal number,}$$

$$d_{ftr} = 0x0D0A \text{ as a hexadecimal number;}$$

$$\delta_{ext} : (f_1, l_1) \times (d_{hdr}) \rightarrow \begin{cases} (d_{hdr}, true) & \text{if } f_1 = d_{hdr}, \\ (d_{hdr}, false) & \text{if } f_1 \neq d_{hdr}; \end{cases}$$

$$(f_9, l_9) \times (d_{ftr}) \rightarrow \begin{cases} (d_{ftr}, true) & \text{if } f_9 = d_{ftr}, \\ (d_{ftr}, false) & \text{if } f_9 \neq d_{ftr}; \end{cases}$$

$$\omega : \begin{cases} (d_{hdr}, true) \rightarrow (d_{hdr}, true); \\ (d_{hdr}, false) \rightarrow (d_{hdr}, false); \\ (d_{ftr}, true) \rightarrow (d_{ftr}, true); \\ (d_{ftr}, false) \rightarrow (d_{ftr}, false). \end{cases}$$

4.3. *Logic Modeling at Field Level.* The logic model assesses if each field is understandable for receiving systems. Because the interpretation process is complicated, the initial value of the field data needs a sequence of transition functions to decode to an understandable information. The proposed logic model is formalized as follows:

$$LM_{Field} = \langle X, Y, S, \delta_{ext}, \delta_{int}, \omega \rangle$$

X is an input set of fields, where

$$X = \{v_{tgt}, v_{ref}\},$$

v_{tgt} is the initial value of the target field,

v_{ref} is the initial value of the reference field;

Y is an output set of logical faults, where

$$Y = \{(v, r) \mid v = v_{tgt} \in X, r \in \{true, false\}\},$$

v is the initial data of the target field,

r is the fault result of the target field;

$S = F \times P \times R$ is a total state set, where

$F = \{(f_{int}, f_{dec}) \mid f_{int} = v_{tgt} \in X\}$ is a set of values for the target field, where

f_{int} is the initial value of the target field,

f_{dec} is the decoded value of f_{int} after the previous transition function,

$P = \{p_j \mid 1 \leq j \leq n\}$ is a set of parameters for logical rules,

$R = \{(r_t, r_f) \mid r_t, r_f \in \{false, idle, true\}\}$ is a set of

transition results, where

r_t is the transient result after the interim transition function,

r_f is the final result after the last transition function;

$\delta_{ext} : X \times P \times R \rightarrow F \times R$ is the field transition function with X ;

$\delta_{int} : F \times P \times R \rightarrow F \times R$ is the field transition function

without X ;

$\omega : F \times R \rightarrow Y$ is the fault decision function.

A key element of LM_{Field} is the field transition functions, (i.e., δ_{ext} and δ_{int}); thus, we have summarized their practical

TABLE 4: Examples of δ classified into 5 operations.

Category	δ	Description
Conversion operation	Type conversion (δ^1)	Changing one data type into another, for example, integer to string
Arithmetic operation	Addition (δ^2)	Fundamental numerical functions, for example, +, -, \times , \div
	Subtraction (δ^3)	
	Multiplication (δ^4)	
	Division (δ^5)	
Comparison operation	Greater than (δ^6)	Inequality functions, for example, >, \geq , <, \leq
	Greater than or equal (δ^7)	
	Less than (δ^8)	
	Less than or equal (δ^9)	
	Equal (δ^{10})	
Special operation	Not equal (δ^{11})	Equality functions, for example, =, \neq
	User-defined equation (δ^{12})	User-defined equation in a complex form including arithmetical, logical, and bit-shift operations, for example, $(\sin x + 10) \times 30$, $(x < 90) \& (x > -90)$, $[x] << 7$
	Regular expression (δ^{13})	Comparison of standard textual syntax for representing patterns of text, for example, $[p - s][0 - 5][0 - 9]$, $[0 - 3][0 - 5][0 - 9]$
Flow control operation	ASCII expression (δ^{14})	Comparison of ASCII codes for example: a specific field \equiv "STS"
	Conditional flow (δ^{15})	Conditional function, for example, If-else statement

δ is for either δ_{ext} or δ_{int} .

types in Table 4. In common with Table 3, the functions also were induced from Table 1.

The five categories in Table 4 show that three types of operations are basically provided to (1) convert the data type of the field to another, (2) compute it arithmetically, and (3) compare it with a criterion. To express the complicated patterns in the fields, we identified 3 special functions: user-defined equation, regular expression [55], and ASCII expression (δ_{12} to δ_{14} in Table 4). For example, we assume that a bearing field has three ASCII characters to represent a three-digit number. The field is additionally promised that the seventh bit of the third character is always assigned if the bearing is not newly updated. Otherwise, under normal states, the field is interpreted as the three-digit number from zero to 359. In this case, we simplified logic modeling by using two regular expressions: to compare the field with patterns, that is, $[0 - 3][0 - 5][0 - 9]$ for a normal state and $[p - s][0 - 5][0 - 9]$ for an abnormal state. Finally, these operations are parallel in using a flow control operation to make a conditional statement. Parameters for the operations are specified in P in LM_{Field} .

Let us explain how the transition functions in Table 4 are used in a real case. As explained previously, an Electronic Support Measure (ESM) sends bearing information to be encoded with three ASCII characters. The following steps show the overall procedures that the receiving system decodes the bearing information:

- (1) The receiving system first identifies availability of the bearing field by checking another reference field within the same message (δ_{ext}^{10} , δ_{ext}^{15}).
- (2) If the bearing field is not available, the system confirms that its hexadecimal values are all 0x20 (δ_{int}^{14}).
- (3) If the bearing field is available, the system next identifies if the field is newly updated by comparing its values with predefined patterns ($\delta_{\text{int}}^{13,1}$).
- (4) If the bearing field is proved to be newly updated, the system checks if the numeric value is within the valid range from zero to 360 ($\delta_{\text{int}}^{13,2}$).
- (5) The system finally converses its current data type, that is, ASCII characters, into unsigned integers (δ_{int}^1).

These steps are performed sequentially according to the results of the previous step. In this case, for logic modeling, six transition functions were used including one conditional flow function. The overall specifications are as follows (the bold fonts in δ mean updated parts):

$$LM_{\text{bearing}} = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \omega \rangle$$

$$X = \{v_{\text{brg}}, v_{\text{ref}}\}, \text{ where}$$

v_{brg} is the initial value of the targeted bearing field,

v_{ref} is the initial value of the reference field for checking the availability of v_{brg} ;

$$Y = \{(v_{\text{brg}}, \text{true}), (v_{\text{brg}}, \text{false})\};$$

$$S = \{(f_{\text{int}}, f_{\text{dec}})\} \times \{p_1, p_2, p_3, p_4\} \times \{(r_i, r_f)\}, \text{ where}$$

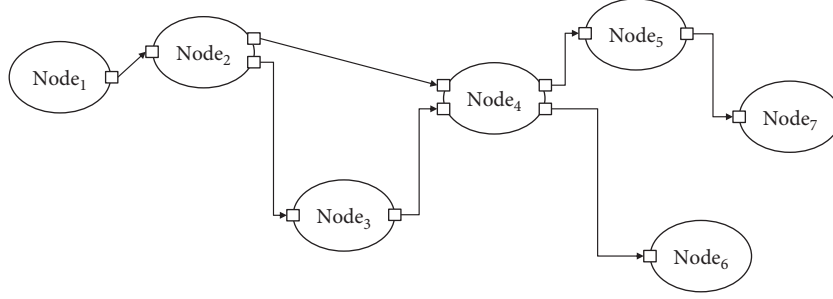


FIGURE 5: Sequential property for interpreting field.

$p_1 = 0 \times 20$ as a hexadecimal number,
 $p_2 = [p - s][0 - 5][0 - 9]$ as a regular expression,
 $p_3 = [0 - 3][0 - 5][0 - 9]$ as a regular expression,
 $p_4 =$ unsigned integer as data type;

$$\begin{aligned}
 \delta_{\text{ext}}^{10} : & (v_{\text{ref}}) \times (p_1) \times (\text{idle}, \text{idle}) \\
 & \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{true}, \mathbf{false}) \text{ if } v_{\text{brg}} \text{ is not} \\
 & \text{available,} \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{false}, \mathbf{false}) \text{ if } v_{\text{brg}} \text{ is} \\
 & \text{available;}
 \end{aligned}$$

$$\begin{aligned}
 \delta_{\text{ext}}^{15} : & \\
 & (v_{\text{tgt}}) \times \emptyset \times (\mathbf{true}, \mathbf{false}) \\
 & \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{true}, \mathbf{false}), \\
 & (v_{\text{tgt}}) \times \emptyset \times (\mathbf{false}, \mathbf{false}) \\
 & \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{false}, \mathbf{false});
 \end{aligned}$$

$$\begin{aligned}
 \delta_{\text{int}}^{14} : & \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (p_1) \times (\mathbf{true}, \mathbf{false}) \\
 & \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{true}, \mathbf{true}) \text{ if } v_{\text{tgt}} \text{ is correct,} \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{false}, \mathbf{false}) \text{ if } v_{\text{tgt}} \text{ is not correct;}
 \end{aligned}$$

$$\begin{aligned}
 \delta_{\text{int}}^{13.1} : & \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (p_2) \times (\mathbf{false}, \mathbf{false}) \\
 & \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{true}, \mathbf{true}) \text{ if the comparison} \\
 & \text{is correct,} \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{false}, \mathbf{false}) \text{ if the comparison} \\
 & \text{is not correct;}
 \end{aligned}$$

$$\begin{aligned}
 \delta_{\text{int}}^{13.2} : & \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (p_3) \times (\mathbf{false}, \mathbf{false}) \\
 & \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{true}, \mathbf{false}) \text{ if the comparison is} \\
 & \text{correct,} \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{false}, \mathbf{true}) \text{ if the comparison is} \\
 & \text{not correct;}
 \end{aligned}$$

$$\begin{aligned}
 \delta_{\text{int}}^1 : & (f_{\text{int}}, f_{\text{dec}}) \times (p_4) \times (\mathbf{true}, \mathbf{false}) \rightarrow (f_{\text{int}}, f_{\text{dec}}) \times \\
 & (\mathbf{true}, \mathbf{true}); \\
 \omega : & (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{true}, \mathbf{true}) \rightarrow (v_{\text{brg}}, \mathbf{true}), \\
 & (f_{\text{int}}, f_{\text{dec}}) \times (\mathbf{false}, \mathbf{true}) \rightarrow (v_{\text{brg}}, \mathbf{false}).
 \end{aligned}$$

To realize the sequential characteristics of the logic modeling as a software, a computer network concept is applied. Figure 5 shows a schematic illustration of a network configuration, which is a collection of nodes and connections, and the connectors in the nodes are anchor points to attach connections between the nodes. For example, Node₁ corresponds to X of LM_{Field} , Node₆ and Node₇ are relevant to Y , and the others are represented by two transition functions: δ_{ext} or δ_{int} . A major difference from the typical computer network is that the network in Figure 5 is a one-way communication and not a two-way interaction; that is, all the connections have directions to pass the data to the node at right.

Figure 6 shows a class diagram for logic modeling based on the network configuration. The *DiagramViewModel* visualizes and edits the overall modeling of a field. *Nodes* and *Connections* as properties of this class specify the collections of nodes and connections to be displayed in the logic modeling. In *NodeViewModel*, *InputConnectors*, and *OutputConnectors* are the collection of connectors that specify the node's connection anchor points, and *AttachedConnections* retrieves a collection of the connections that are attached to the node. The *Element* determines the type of the node. The *ConnectionViewModel* describes a connection between both-sided nodes, specifically two connectors in each node (i.e., the *SourceConnector* and the *DestConnector*). This connection continuously monitors its source and destination connectors. Finally, the *ConnectorViewModel* indicates an anchor point on a node for attaching a connection. The *ParentNode* in this class references the node that owns the connector.

Figure 7 shows the modeling execution of the bearing field previously described, that is, LM_{bearing} . The developed software provides two views: a list view in the form of the ribbon command bar and a model view for building the model. The list view provides block libraries of modeling elements, in particular transition functions in Table 4 (the red box in Figure 7). Using the libraries, a modeler can

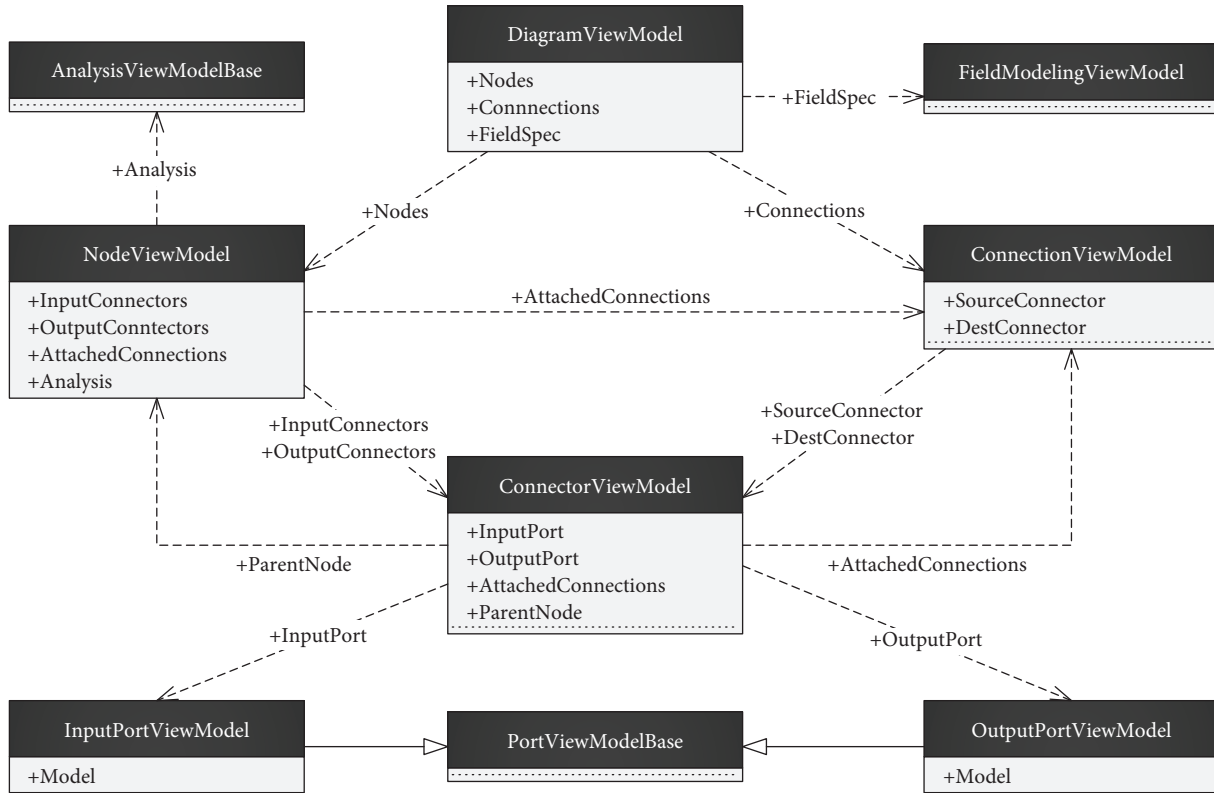


FIGURE 6: Class diagram for logic modeling using network configuration.

graphically build and edit the logic model in the model view. For example, he/she can choose an appropriate element from the list view based on the model design and drag it to the model view. By connecting the two-sided elements with lines, the modeler can easily build a sequence of transitions and decision functions. In Figure 7, yellow boxes are realizations of LM_{bearing} . In this manner, the developed software facilitates intuitive modeling via the block libraries and allows flexible modeling through the addition and deletion of the modeling elements.

5. Application

The objective of this application is to demonstrate how we can detect and diagnose interface faults when designing and integrating a complex SoS. The targeted system is an underwater vessel (i.e., a submarine system). The faults were (1) incorrect interface protocols during the design phase and (2) abnormal values in interface data during the integration phase.

5.1. Shipbuilding Project Overview. Due to budget constraints, the Navy can no longer afford to build new ships beyond its existing military force [28]. In this context, the product improvement program (PIP) is a good alternative. The PIP incorporates improvements of partial systems to enhance overall system performance. Because it reduces the procurement time and lowers the maintenance costs compared to the development of an entirely new system,

PIP has become an industrial trend in several industrial fields [56–59].

Since late 2014, South Korean shipbuilder, Daewoo Shipbuilding and Marine Engineering (DSME), has undergone a PIP for three submarine systems that corresponded to the first-generation class of Table 1 [24]. The submarines' onboard subsystems including navigation, acoustic, optical, and radar sensors as well as combat systems have been renovated. The PIP will be finalized in 2019.

For this PIP, the compatibility of the improved subsystems with existing ones at the I/O level is a key consideration. Thus, two phases in the system development life cycle (SDLC) require interface fault-handling activities. To be specific, the validity of the interface protocols needs to be assured to accurately represent the interface data between the linked subsystems. Then, the developed subsystems should be verified at the I/O level by comparing the interface data with the valid protocols. In this respect, the DSME has carried out several tests to resolve interface faults using the proposed method and software at the design phase as well as the integration phase. More detailed descriptions for this PIP were informed in our previous work [12].

5.2. Design of Tests. As shown in Table 5, the proposed modeling and software have been utilized for nine shipboard tests over the last three years. Until the first half of 2017, preliminary and critical design phases had been proceeded for the first renovated submarine. During this period, eight tests were conducted. Thereafter, all the subsystems were

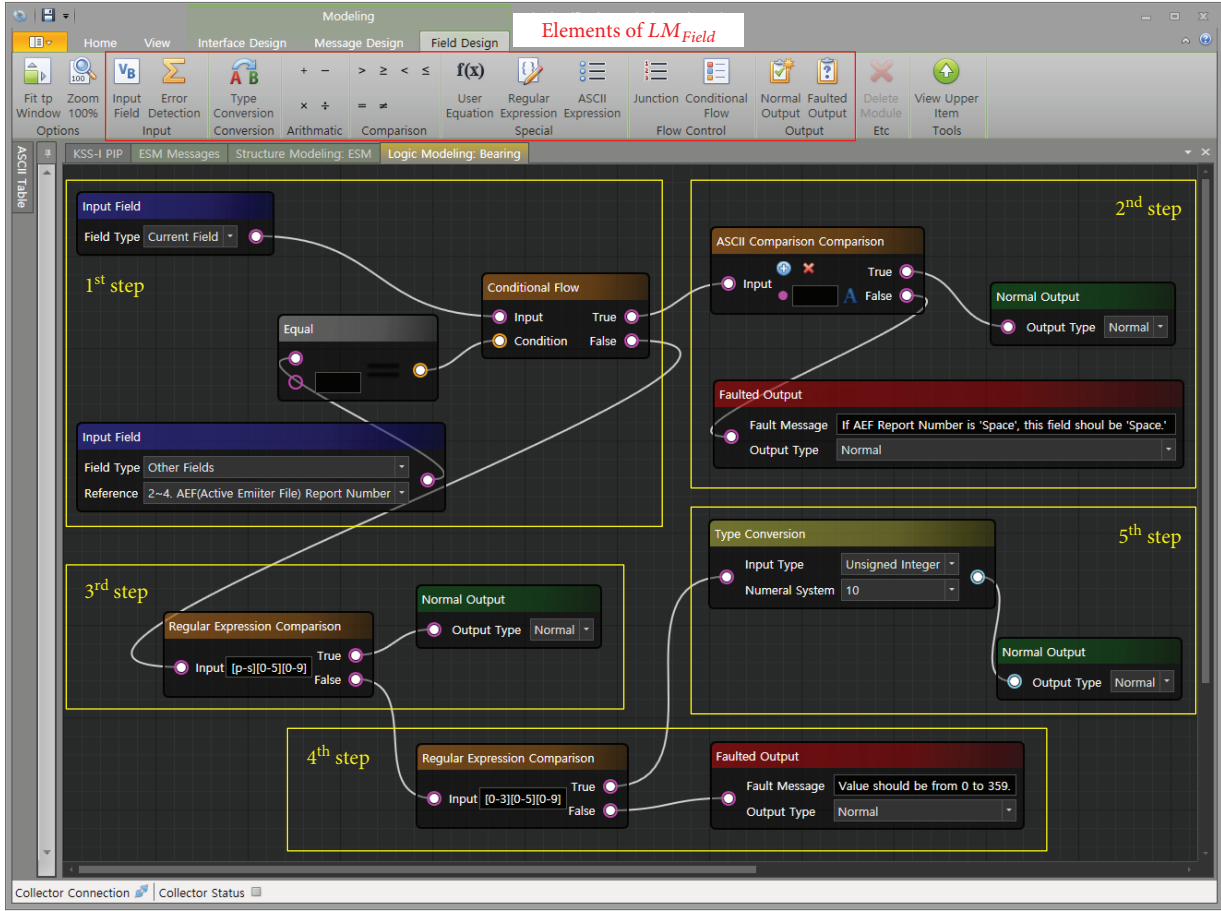


FIGURE 7: Logic modeling of bearing field in ESM message.

TABLE 5: Overall design of tests during system design and integration phases.

SDLC	Test no.	When (yy.mm)	Test site	Subsystems of local interfaces	Objective	Test period
System design phase	Test ₁	15.03	At sea	Eight sensor and one control systems	To train for detecting a surface target and evaluate target motion analysis	72 hours
	Test ₂	15.05	At sea	Eight sensor and one control systems		72 hours
	Test ₃	15.09	At the harbor	One additional control system	To train command and control for underwater weapon engagement	6 hours
	Test ₄	15.12	At sea	Eight sensor and one control systems	To measure accuracy of passive sonars to detect a surface target	72 hours
	Test ₅	16.08	At sea	Eight sensor and one control systems	To acquire navigation data	72 hours
	Test ₆	16.11	At sea	Eight sensor and one control systems	To measure underwater radiation noise	72 hours
	Test ₇	16.11	At sea	Eight sensor and one control systems	To measure self-noise of sonar systems	72 hours
	Test ₈	17.04	At sea	Eight sensor and one control systems		72 hours
System integration phase	Test ₉	18.01	At the harbor	Four sensor systems	To evaluate systems integration	6 hours

yy and mm in (yy.mm) mean year and month, respectively. Subsystems in the fourth column are connected to local proxies such as the DIS, the integrated management system, or the signal processing system. These tests are extension from our previous study [12].

Key	Interface Name	Description	Device 1	Device 2	Transmission Method
SPS	SPS	DU7	SPS	DDS	RS-232
INS	INS	INS	INS	DDS	RS-422
GPS	GPS	DUB	GPS	DDS	RS-232
EMLOG	EMLOG	DU2	EMLOG	DDS	RS-232
DEPTH	DEPTH SONAR		DEPTH SONAR	DDS	RS-232
ESM	ESM	DU6	ESM	DDS	RS-232
PLT	PLT	DU4	PLT	DDS	RS-232
ESD	ECHO SOUNDER		ECHO SOUNDER	DDS	RS-422
CTD	CTD		CTD	DDS	RS-422
WTSRC	WTSRC		WTSRC	DDS	RS-232

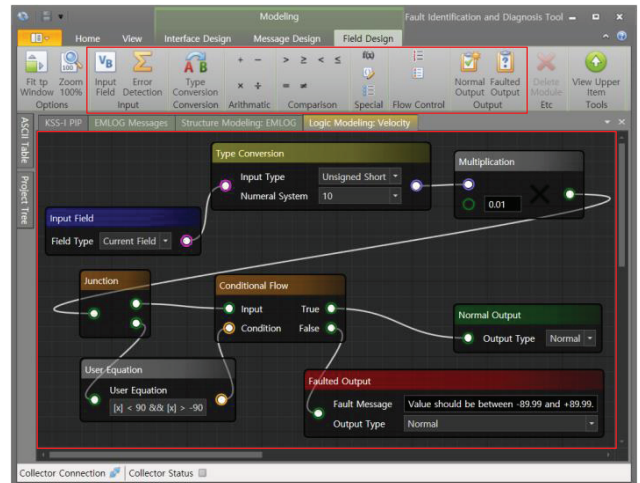
(a) Modeling of 9 groups of interfaces in shipbuilding system

Key	Name	Description
GPSMsgM	Data Block 01 (M)	Satellite Fix Time Mark Output
GPSMsgX	Data Block 02 (X)	Satellite Fix Position Output
GPSMsgQ	Data Block 03 (Q)	Satellite Fix Quality Output (Part I)
GPSMsgU	Data Block 04 (U)	Satellite Fix Quality Output (Part II)
GPSMsgO	Data Block 05 (O)	Satellite Dead Reckoning Output
GPSMsgR	Data Block 06 (R)	Status Reply Output
GPSMsgB	Data Block 08 (B)	Omega Fix Time Output
GPSMsgC	Data Block 09 (C)	Omega Fix Position Output
GPSMsgD	Data Block 10 (D)	Omega Fix Quality Output
GPSMsgE	Data Block 11 (E)	Satellite/Omega Dead Reckoning
GPSMsgI	Data Block 30 (I)	GPS Navigational Data
GPSMsgZ	Data Block 31 (Z)	GPS Status Data
CMSMsgT	Data Block 11 (T)	Status Request
CMSMsgN	Data Block 12 (N)	Velocity
CMSMsgK	Data Block 13 (K)	Control

(b) Modeling of 15 groups of messages transferred in GPS

#	Length	Type	No. of Byte	Type	Fixed Value(Hex)	Name	Description	Date	Type	Data Length
1	Fixed Length	1-2	Key	3A-5B		Header			Ascii	2
2	Fixed Length	3-5	Data			Day			Ascii	3
3	Fixed Length	6-10	Data			GMT			Ascii	5
4	Fixed Length	11-16	Data			Latitude			Ascii	6
5	Fixed Length	17-22	Data			Longitude			Ascii	6
6	Fixed Length	23	Data			Quality flag			Ascii	1
7	Fixed Length	24-29	Data			Reserved			Ascii	6
8	Fixed Length	30	Data			Checksum			Ascii	1
9	Fixed Length	31-32	Fixed V...	0D-0A		Footer			Ascii	2

(c) Modeling of 9 fields in GPS position message



(d) Modeling of velocity field in EM log message

FIGURE 8: Interface data modeling using developed software.

completely developed; thus, they have been integrated to the submarine system as of late 2017. During this time, we carried out one fault test. Most tests were conducted at sea to resolve the faults for various operational situations.

Among 10 subsystems to be tested, eight were sensor systems including acoustic and optical sensors (e.g., echo sounder, depth sonar, periscope, CTD device, EM log, and ESM) and the navigation suite of sensors (e.g., INS and GPS). Two control systems were a plotting board system and a weapon control system. The weapon control system was an additionally renovated system during the PIP. This means that test₃ in Table 5 was unexpected and belatedly determined just two months before the test.

5.3. Interface Data Modeling. For detection and diagnosis of the interface faults, we first modeled message structures and field logics in each interface. Figure 8 shows some modeling results using the developed software. Ten interfaces between the 10 subsystems and local proxies were targeted, which are based on serial communications such

as RS-232 and RS-422 (Figure 8(a)). As an example of the interface between the GPS and the DIS, 15 groups of messages were transferred (Figure 8(b)). One message, Data Block 2, was modeled with nine fields including d_{hdr} and d_{ftr} (Figure 8(c)). These figures are the realization of SM_{GPS-1} described in Section 4.2.

Figure 8(d) shows logic modeling of the velocity field in the EM log message. In this field, two main interpretations are required: (1) to convert a hexadecimal number to a floating-point number and (2) to check the valid range of the value. Specifically, the field data has initially a hexadecimal number; thus, it needs to be first converted to a decimal number. Then the decimal number is multiplied by 0.01 to represent two decimal places. The value is finally checked whether it is within the valid range from -90 to $+90$. If the result is out of range, the field is diagnosed with a logical fault. The logic modeling for this process can be expressed in a combination of various transition functions. In Figure 8(d), four transition functions (i.e., δ_{ext}^1 , δ_{int}^4 , δ_{int}^{12} , and δ_{int}^{15}) were used to model the logic of the velocity field.

TABLE 6: Main results of interface data modeling.

Interface type	Message type	Structure modeling (SM)					Field division unit	Logic modeling (LM)		
		$\sum_{i=1}^n l_i \in X$	$N(X)$	d_{hdr}	d_{ftr}	d_{length}		$\frac{\sum_{i=1}^{N(\text{SM},X)} [N(\delta_{\text{ext}}) + N(\delta_{\text{int}})]}{N(\text{SM},X)}$	Decoded value Number Character	
Sensor _A	Msg _{A-1}	14	6	O	O		Byte	4.33	O	O
	Msg _{A-2}	9	4	O	O		Byte	2.50	O	
Sensor _B	Msg _{B-1}	28	20	O	O		Bit	3.00	O	O
	Msg _{B-2}	30	27			O	Bit	3.44	O	O
Sensor _C	Msg _{C-1}	4	2	O	O		Byte	2.00	O	
	Msg _{C-2}	32	9	O	O		Byte	2.56	O	
	Msg _{C-3}	44	16	O	O		Byte	2.31	O	
	Msg _{C-4}	50	4	O	O		Byte	2.78		O
	Msg _{C-5}	54	18	O	O		Byte	2.56	O	O
	Msg _{C-6}	6	3	O	O		Byte	3.00		O
	Msg _{C-7}	4	2	O	O		Byte	3.30	O	
	Msg _{C-8}	32	9	O	O		Byte	4.63	O	O
	Msg _{C-9}	44	10	O	O		Byte	2.78	O	O
	Msg _{C-10}	54	16	O	O		Byte	2.75	O	O
	Msg _{C-11}	62	15	O	O		Byte	3.32	O	
	Msg _{C-12}	52	17	O	O		Byte	2.50	O	O
	Msg _{C-13}	6	3	O	O		Byte	2.50		O
	Msg _{C-14}	24	7	O	O		Byte	3.33	O	
	Msg _{C-15}	6	4	O	O		Byte	3.75		O
Sensor _D	Msg _{D-1}	17	7	O	O		Byte	5.29	O	O
Sensor _E	Msg _{E-1}	Variable length	4	O	O		Byte	3.00	O	O
	Msg _{E-2}	9	5	O	O		Byte	3.00	O	O
Sensor _F	Msg _{F-1}	236	67	O	O		Byte	3.75	O	O
Control _G	Msg _{G-1}	7	4	O	O		Byte	5.12		O
	Msg _{G-2}	284	85	O	O		Byte	3.33	O	O
Sensor _H	Msg _{H-1}	26	16	O	O		Byte	7.25	O	O
	Msg _{H-2}	75	17	O	O		Byte	3.25	O	O
Sensor _I	Msg _{I-1}	3	7	O	O		Bit	2.25	O	
	Msg _{I-2}	3	7	O	O		Bit	2.25	O	
	Msg _{I-3}	3	7	O	O		Bit	2.25	O	
	Msg _{I-4}	3	7	O	O		Bit	2.25	O	
Control _J	Msg _{J-1}	18	90	O	O		Bit	3.75	O	O

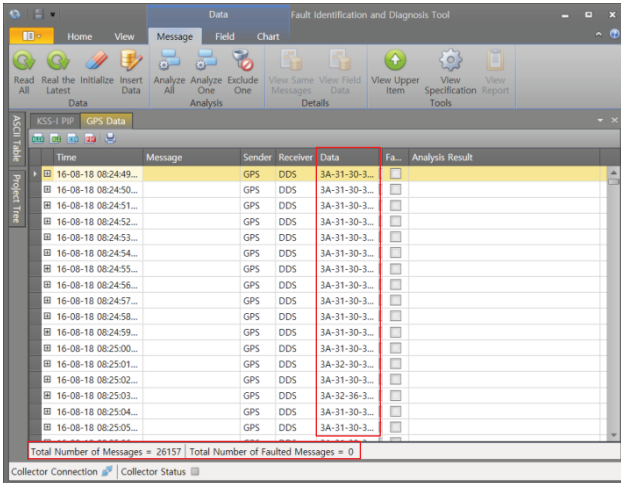
Interface names consist of subsystems connected to local proxies and their identifiers: subsystem identifier. ‘‘O’’ in columns for the delimiter and interpreted value means to be applicable as a positive answer.

Table 6 summarizes key results of overall structure and logical modeling. Thirty-two messages in the interfaces were modeled whose lengths are 3 to 284 bytes. The number of fields in each message, that is, $N(X \in SM)$, increases if the message length is longer or if the fields are separated by bit units. For example, Msg_{G-2}, the longest message, has 85 fields in 284 bytes. Whereas, Msg_{J-1} has 90 fields in only 18 bytes because it is divided by bit units as a typical example of customized communication protocols.

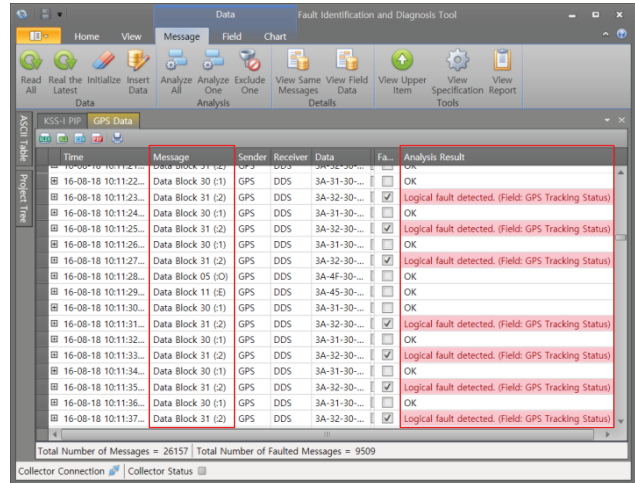
Now, let us examine Msg_{B-2} and Msg_{E-1} to explain specific types of message delimiters. First, since Msg_{B-2} has a fixed-length without any header and footer, it should be classified with a message length (i.e., d_{length}). On the other hand, the

length of Msg_{E-1} is variable because its depth field has a floating-point number with 3 to 6 bytes. The variable length was not actually recognized before test₅, which will be explained in the following subsection. After test₅, Msg_{E-1} was accepted that it cannot be classified with the message length; instead, it should be classified into the header and the footer. Except for these cases, all the messages are generally modeled with d_{hdr} and d_{ftr} . To sum up, the messages in this study used two structural types: (1) d_{hdr} and d_{ftr} and (2) d_{length} .

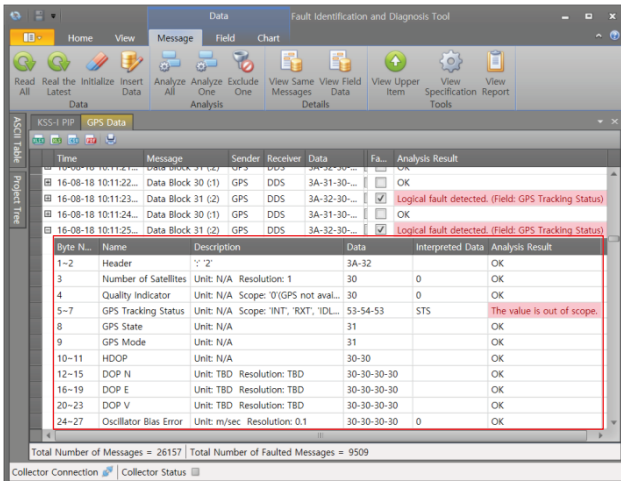
For logic modeling, the ninth column of Table 6 shows the average number of transition functions (i.e., δ_{ext} and δ_{int}) to be used for modeling fields in each message. For example, 5.29 in Msg_{D-1} means that more than five transition



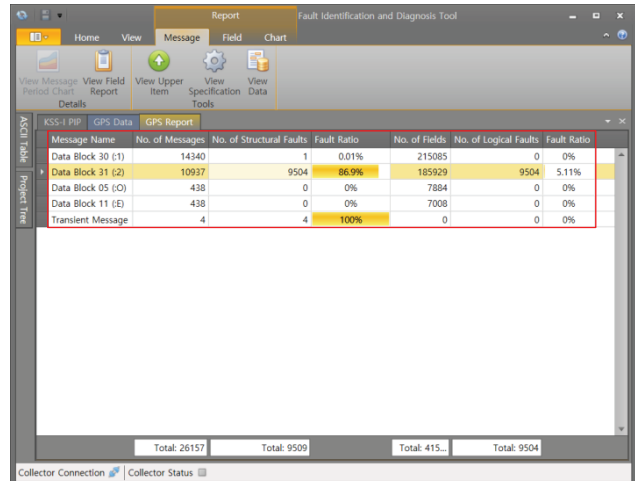
(a) Read of interface signals: GPS messages



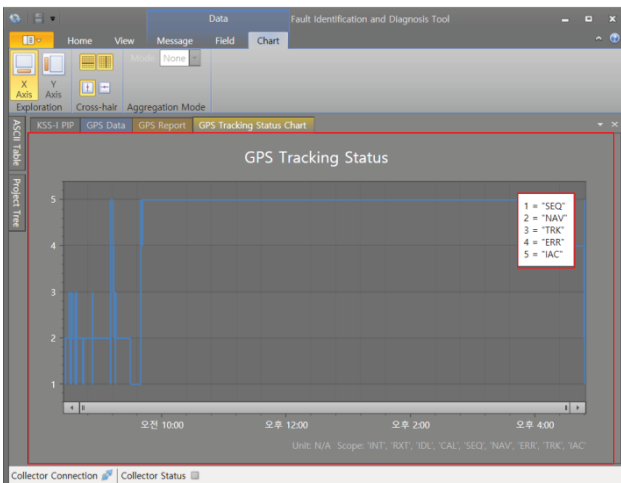
(b) Signal identification and diagnosis at message level



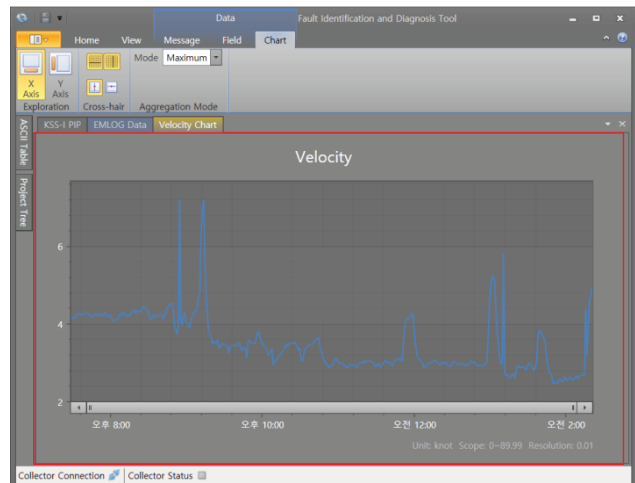
(c) Signal identification and diagnosis at field level



(d) Report of faults identification in message and field levels



(e) Analysis of time series data: Tracking status field



(f) Analysis of time series data: Velocity field

FIGURE 9: Results of fault detection and diagnosis using developed software.

functions were used to model each field. In this column, most messages have numbers larger than 2, which means that a group of transition functions was used to interpret the field

and diagnose faults. Finally, the decoded values could be either numbers such as velocity, yaw, pitch, and depth or characters (e.g., textual message or behavioral mode).

TABLE 7: Overall test results: fault-detection results.

Test no.	Evaluation index	Interface									
		Sensor _A	Sensor _B	Sensor _C	Sensor _D	Sensor _E	Sensor _F	Control _G	Sensor _H	Sensor _I	Control _J
Test ₁	NI _{total}	677,229	1,744,894	288,906	314,987	21,403	2555	39,529	618,028	4,105,489	N/A
	NI _{fault}	0	0	0	0	0	1327	14,366	0	0	N/A
Test ₂	NI _{total}	381,452	741,652	66,487	134,920	102,586	14,013	16,862	162,438	1,537,041	N/A
	NI _{fault}	0	0	0	0	76,828	0	1168	0	0	N/A
Test ₃	NI _{total}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	126,533
	NI _{fault}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0
Test ₄	NI _{total}	1,294,454	1,524,532	188,250	277,205	2,691,498	52,296	34,650	554,447	2,704,102	N/A
	NI _{fault}	0	0	0	0	0	0	8213	0	0	N/A
Test ₅	NI _{total}	1,021,158	1,238,562	26,157	225,154	2,132,133	12,911	28,144	450,184	1,843,669	N/A
	NI _{fault}	0	0	9509	0	0	0	0	0	00	N/A
Test ₆	NI _{total}	578,971	2,070,717	138,547	376,578	128,331	3830	47,072	489,510	4,752,324	N/A
	NI _{fault}	0	0	30,193	0	0	0	0	0	0	N/A
Test ₇	NI _{total}	197,753	629,961	56,431	112,904	85,332	68	14,003	214,499	953,452	N/A
	NI _{fault}	0	0	0	0	0	0	0	0	0	N/A
Test ₈	NI _{total}	1,151,185	2,076,729	188,025	372,984	3,664,990	991	39,169	750,569	3,610,105	N/A
	NI _{fault}	0	0	0	0	0	0	0	0	0	N/A
Test ₉	NI _{total}	N/A	11,004	N/A	N/A	10,582	N/A	N/A	3496	8877	N/A
	NI _{fault}	N/A	2331	N/A	N/A	0	N/A	N/A	3496	0	N/A

NI_{total} is the number of obtained messages from each interface. NI_{fault} is the number of faulted messages from each interface. Some interface is not applicable for a specific test, which is represented by N/A.

5.4. Test Results. Figure 9 shows some results of test₅ and test₆ using the developed software. All the figures except Figure 9(e) are relevant to the GPS messages.

Figure 9(a) shows the messages transmitted between the GPS and the DIS. During the 72 hours, 26,157 messages were monitored, which are chronologically arranged in the main table. Because the messages had not been evaluated yet, two columns—Message and Analysis Result—are empty, and the number of faulted messages at the bottom of the table is also zero. By pushing the Analyze All icon in the ribbon bar, the messages were analyzed. In the main table of Figure 9(b), during 15 seconds, two message types (Data Block 05 and Data Block 11) were identified just once, and two types (Data Block 30 and Data Block 31) were distinguished continuously. After the analysis, all the messages regarding Data Block 31 were diagnosed with logical faults, of whose rows in Analysis Result are shaded in red. Of the 26,157 messages, 9509 messages were faulted, which are indicated at the bottom of the table.

To examine where and why the faults occurred in each message, the message can be opened out so that every field is displayed. In Figure 9(c), the opened message has a logical fault at the fourth field due to the unexpected value. To be specific, the fourth field was modeled not to send STS. However, during test₅, the relevant system actually sent that value, which leads to a contradiction between the modeling and the real data. Finally, the overall results were shown by pushing the View Report icon. In Figure 9(d), more than 80% of messages in Data Block 32 were faulted during test₅.

The software also provides a time series chart for an interpreted value of each field. Figure 9(e) shows a numeric chart for the velocity value in the EM log message, and Figure 9(f) represents a chart for tracking status in the GPS message. These charts facilitate the trend of the values according to the progressed time at a glance. For example, test₅ was for acquiring navigation data at various velocities, and the chart exactly visualizes when the velocity is changed. The INS needs the GPS data to calibrate the navigation data, and the GPS status can be found on the chart regularly.

Table 7 summarizes (1) how many messages were acquired from interfaces of all the tests and (2) how many faults were detected among them. The numbers of the obtained messages from each interface (i.e., NI_{total} in Table 7) are all different for the following two main reasons. First, each interface has different message types as well as the types have different transmission cycles. For example, Control_G has two message types (i.e., Msg_{G-1} and Msg_{G-2}) and they are transferred every eight seconds. On the other hand, Sensor_I has four message types with 0.125-second cycles. In this case, NI_{total} of Sensor_I is arithmetically 128 times more than that of Control_G if they operate in the same amount of time ($128 = (4/2) \times (8/0.125)$). Next, because scenarios of the tests are all distinguished, the subsystems are operated situationally. The periscope system for Sensor_A is normally operated during the vessel moves above the specific depth (i.e., periscope depth). This means that any messages in Sensor_A will not be transferred when the vessel dives below the depth. In this context, we can assume that test₂ was carried out below the periscope depth over a longer period than test₁.

TABLE 8: Overall test results: fault-diagnosis results.

Test number	Fault case	Interface type	Message type	Interface data	Fault diagnosis	Fault category
Test ₁	Fault ₁	Sensor _F	Msg _{F-1}	02-30-31-30-30-39-33-37- ...-35-31-20-20-20-...-20- 20-20-30-30-33-35-...-03	The modeling should be revised that all the fields regarding a specific section are full of “0x20” if no targets are detected in the section.	Logical fault
	Fault ₂	Control _G	Msg _{G-2}	0D-0A-48-32-35-31-31- ...-20-2B-30-38-35-33- ...-53-2B-...	The modeling should be revised that the interpreted value of the sign field can be “+” although the precondition field is unavailable.	Logical fault
	Fault ₃	Control _G	Msg _{G-2}	0D-0A-48-32-35-31-31- ...-20-2B-30-38-35-33- ...-53-2B-...	The modeling should be revised that the speed field has meaningful information although the precondition field is unavailable.	Logical fault
	Fault ₄	Control _G	Msg _{G-2}	0D-0A-48-32-35-31-31- ...-20-2B-30-38-35-33- ...-53-2B-4F-53-2B-...	The modeling should be revised that the interpreted value of the headline sonar field is “O” instead of “0” if sonar systems are available.	Logical fault
	Fault ₅	Control _G	Msg _{G-2}	0D-0A-48-32-35-31-31- ...-00-00-00-00-20-00-00- 00-...	The modeling should be revised that every byte of the range field can be 0x00 as well as [0x30, 0x39].	Logical fault
Test ₂	Fault ₆	Sensor _E	Msg _{E-1}	2A-30-30-30-31-32-30-31- 36-0D-0A	The delimiters should be changed from header and length to header and footer for variable lengths.	Structural fault
	Fault ₇	Sensor _E	Msg _{E-1}	2A-30-30-30-31-2D-36-34- 34-35-33-39-0D-0A	The pressure field should be revised to have 4 to 7 bytes including sign characters optionally.	Logical fault
	Fault ₈	Control _G	Msg _{G-2}	0D-0A-48-32-35-31-31- ...-00-00-00-...-30-30-35-40	The modeling should be revised that the target field is full of “0x00” if the relevant target is not identified.	Logical fault
Test ₄	Fault ₉	Control _G	Msg _{G-1}	<i>05-0A-41-43-4B-20-40</i>	The modeling should be revised that multiple headers, that is, “0D-0A” and “05-0A” are allowed.	Structural fault
Test ₅	Fault ₁₀	Sensor _C	Msg _{C-12}	3A-32-30-30-53-43-53-31- 31-30-...30-30-53-0D-0A	The modeling should be revised that the interpreted value of the status field contains “STS.”	Logical fault
Test ₆	Fault ₁₁	Sensor _C	Msg _{C-12}	3A-32-30-30-49-41-43-31- 31-30-...30-30-53-0D-0A	The modeling should be revised that the interpreted value of the status field contains “IAC.”	Logical fault
	Fault ₁₂	Sensor _C	Msg _{C-12}	3A-32-30-30-54-52-4B-31- 31-30-...30-30-53-0D-0A	The modeling should be revised that the interpreted value of the status field contains “TRK.”	Logical fault
Test ₉	Fault ₁₃	Sensor _B	Msg _{B-1}	03-01-01-00-00-00-00- ...-1B-C3-79-58-84-8A-00- 00-00-3F	Sensor _B needs to be refined to send “0x00” or “0x54” for the test field although the corresponding system is initialized.	Logical fault
	Fault ₁₄	Sensor _H	Msg _{H-1}	<i>0D-0A-20-20-20-...-38-43- 35-03</i>	Sensor _H should be refined to send the message with an accurate header.	Structural fault

Data in italics in interface data mean parts for fault diagnosis. Square brackets in fault diagnosis are used for regression expression. These results are extended from our previous study [12].

In Table 7, the interfaces where local faults were detected are marked italics. The number of the local faults (i.e., NI_{fault}) was counted if $r_j \in SM.Y$ or $r \in LM.Y$ has *False* once in an individual message. For example, in Sensor_C of test₅, 9509 messages were detected to be structurally or logically faulty among 26,157 messages (this is the case of Figure 9(d)). Synthetically, six interfaces except Sensor_A, Sensor_D, Sensor_F, and Control_J were faulted. Note that Sensor_C and Control_G had local faults in more than two tests. This implies that the causes of the faults are distinct according to the test, which will be explained in Table 8.

To evaluate the relative magnitude of the detected faults in each interface, Figure 10 illustrates fault ratios. In Sensor_E, Sensor_F, and Control_J, more than half of the messages were

faulted. In Control_J of test₉, the 100-percent ratio means all the messages in this interface failed to be interpretable. Although Sensor_C in test₆ has more faulted messages than the case of Sensor_F in test₁, the fault ratio of Sensor_F is twice higher than that of Sensor_C.

Table 8 summarizes diagnostic results of the faults in the overall tests. In total, fourteen fault cases were diagnosed within seven message types: three cases are for structural faults and 11 are relevant to logical faults. The structural faults, which $r_j \in SM.Y$ is *False*, came from incorrect headers and length. The logical faults have three diagnoses: (1) wrong field interpretations, (2) missed status information, and (3) incorrect relations between neighboring fields. Specifically, Msg_{E-1} had structural and logical faults simultaneously.

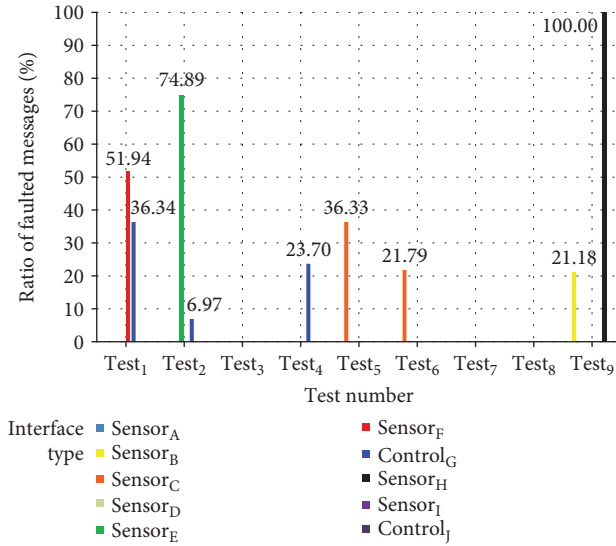


FIGURE 10: Fault ratio at interface level.

Before test₂, Msg_{E-1} was modeled with a header and length for delimiters. However, real messages could not be classified with the current delimiters due to their variable lengths; thus, the modeling was revised to use a header and footer. Then, we looked over logic modeling, focusing on which field influenced the variability. It was proved that the pressure field could be represented with 4 to 7 bytes, including sign characters.

Figure 11 shows how the fault cases are influenced on the messages with the same type. Because NM_{total} in Figure 11 is regarding the same message type, it is a subset of NI_{total} in Table 7. From Figure 11, we summarized the following findings. First, eight fault cases cause more than half of the faulted messages. For example, fault₅ brought about more than 14,000 faulted messages during 72 hours. Next, fault₂ and fault₃ cause the same number of faulted messages, which means that they were complementary and occurred at the same time. Additionally, fault₇ and fault₁₂ were relatively difficult to detect and diagnose because they were scenario-dependent faults. If the scenarios are different, the results will be different. This means that test scenarios to cover all the cases are also important. Finally, the number of faulted messages increased as time passed since one fault case occurred.

5.5. Discussion. For synthesized analysis, Figure 12 summarizes how many fault cases were diagnosed and resolved in each test. Note that the numbers in this graph are not total numbers of faulted messages.

Invalid interface protocols led to unforeseen incompatibilities between subsystems that could not be revealed until they were integrated. The first eight tests were carried out to validate the interface protocols at the system design phase. During the tests, the structure and the logic models had been gradually revised by fixing the current faults for the next test. For example, Msg_{G-2} in Table 7, whose field has “0” as an interpretation value before test₅, modified the interpreted

value after the test. Consequently, the number of fault cases decreased as the tests progressed. Because the seventh and eighth tests had no interface faults, the interface protocols were almost fully assured to be validated.

Let us examine test₃ as a special case. As explained in Section 5.2, it was not considered initially in our tests. Nevertheless, the interface data for the weapon control system could be evaluated because we analyzed various types of interface data in Table 1, generalized their properties, and formalized them with a mathematical form. Fortunately, no faults were detected in this interface. Indeed, formal representations of interface data and flexible modeling using the software are particularly beneficial in arbitrary system developments.

Finally, the faults in test₉ made the corresponding subsystems resolve their unexpected behaviors at the system integration phase. For example, Sensor_B needed transient time for initialization, and during this period, it should have been revised to send an appropriate value in the test field. As the system development progressed, integration problems became harder and more expensive to solve, so it was paramount to figure out potential faults as early as possible [21]. In this application, only two faults were found in the ninth test, which means that the previous eight tests significantly reduced the integration problems.

To sum up, the faults were identified until test₈, induced a revision of the structural and logical modeling. In other words, the DSME as a shipbuilding integrator continuously revised and validated the communication protocols based on the results of the eight tests. After that, the DSME verified the developed systems via resolving the faults in test₉. These fault-resolving activities had been conducted during the system design and integration phases, which is a clear difference between the previous fault-resolving studies. The proposed work played a vital role in the overall submarine renovation project.

6. Conclusion

In this study, we are mainly concerned with intersystem faults whose results are observable outside the systems. Our goal was to find patterns in the interface data that do not conform to expected behaviors.

The main contribution of this study is theoretical and practical. From the theoretical viewpoint, we categorized the interface faults into structural and logical levels, and they were evaluated based on mathematical modeling formalism. The core concept in the formalism is to support explicit functions for transitions and fault decisions. Thus, the proposed formalism could be applicable to customized protocols as well as standardized ones, which is suitable for arbitrary system development. From the practical perspective, the developed software facilitates graphical modeling via creation, arrangement, and revision of the modeling elements. The system integrator could constantly evaluate and supplement the interface protocols at the design phase and the interacted subsystems at the integration phase. It has been successfully utilized for a submarine renovation project.

Fault type	Message type	NM _{total}	NM _{fault}	Fault ratio at message level (%) (NM _{fault} /NM _{total} × 100)
Fault ₁	Msg _{F-1}	2555	1327	51.94%
Fault ₂	Msg _{G-2}	19,764	14,366	72.69%
Fault ₃	Msg _{G-2}	19,764	14,366	72.69%
Fault ₄	Msg _{G-2}	19,764	5662	28.65%
Fault ₅	Msg _{G-2}	19,764	14,366	72.69%
Fault ₆	Msg _{E-1}	102,586	76,528	74.60%
Fault ₇	Msg _{E-1}	102,586	1203	1.17%
Fault ₈	Msg _{G-2}	8431	1168	13.85%
Fault ₉	Msg _{G-1}	17,330	8213	47.39%
Fault ₁₀	Msg _{C-12}	10,937	9504	86.90%
Fault ₁₁	Msg _{C-12}	35,402	30,060	84.91%
Fault ₁₂	Msg _{C-12}	35,402	133	0.38%
Fault ₁₃	Msg _{B-1}	10,812	2331	21.56%
Fault ₁₄	Msg _{H-1}	3496	3496	100.00%

NM_{total} is the number of obtained messages with the same message type from each interface.
 NM_{fault} is the number of faulted messages with the same message type from each interface.

FIGURE 11: Number of faulted messages and fault ratios at message level.

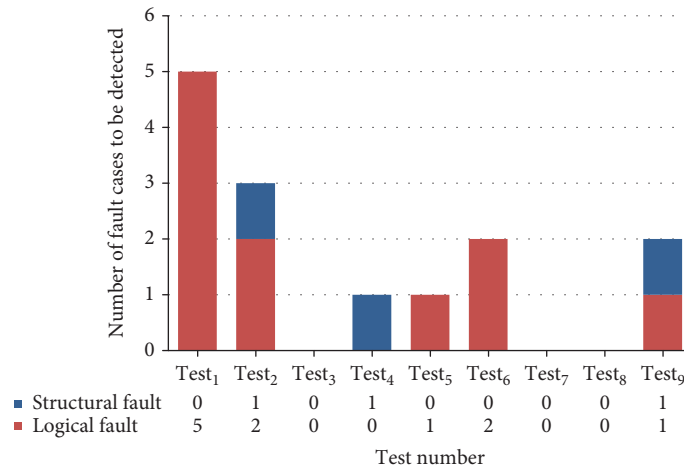


FIGURE 12: Number of fault cases in tests.

All the works in this study were based on real data acquired from submarine systems. The interface faults regarding incorrect design and abnormal implementation can be resolved during designing and integrating complex systems. The proposed work have facilitated to reduce system development time and avoid dangerous situations during a shipbuilding project. The faults interested in this study are relevant to individual interface data; thus, detection and diagnosis of a sequence of multiple interface data will remain for future work.

Abbreviations

ASCII: American Standard Code for Information Interchange
 CTD: Conductivity, temperature, and depth

DEVS: Discrete event system specification
 DIS: Data integration system
 DSME: Daewoo Shipbuilding and Marine Engineering
 EM log: Electromagnetic log
 ESM: Electronic Support Measure
 FSM: Finite system machine
 GPS: Global Positioning System
 GUI: Graphical user interface
 INS: Inertial navigation system
 I/O: Input/output
 MVVM: Model-view-viewmodel
 OOBN: Object-oriented Bayesian networks
 PIP: Product improvement program
 SLDC: System development life cycle
 SoS: System of systems
 WPF: Windows Presentation Foundation.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. L. Butterfield, J. S. Pearlman, and S. C. Vickroy, "A system-of-systems engineering GEOSS: architectural approach," *IEEE Systems Journal*, vol. 2, no. 3, pp. 321–332, 2008.
- [2] L. B. Rainey and A. Tolk, Eds., *Modeling and Simulation Support for System of Systems Engineering Applications*, Wiley, Hoboken, NJ, USA, 2015.
- [3] M. Jamshidi, *Systems of Systems Engineering: Principles and Application*, CRC Press, Boca Raton, FL, USA, 2008.
- [4] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech house, London, UK, 2013.
- [5] H. Panetto and A. Molina, "Enterprise integration and interoperability in manufacturing systems: trends and issues," *Computers in Industry*, vol. 59, no. 7, pp. 641–646, 2008.
- [6] P. Graignic, T. Vosgien, M. Jankovic, V. Tuloup, J. Berquet, and N. Troussier, "Complex system simulation: proposition of a MBSE framework for design-analysis integration," *Proceedia Computer Science*, vol. 16, pp. 59–68, 2013.
- [7] A. M. Madni and M. Sievers, "Systems integration: key perspectives, experiences, and challenges," *Systems Engineering*, vol. 17, no. 1, pp. 37–51, 2014.
- [8] L. Sassaman, M. L. Patterson, S. Bratus, and M. E. Locasto, "Security applications of formal language theory," *IEEE Systems Journal*, vol. 7, no. 3, pp. 489–500, 2013.
- [9] D. C. Sturman and G. A. Agha, "A protocol description language for customizing failure semantics," in *Proceedings of IEEE 13th Symposium on Reliable Distributed Systems*, pp. 148–157, Dana Point, CA, USA, October 1994.
- [10] D. T. Nguyen, Y. Chae, and Y. Park, "Enhancement of data rate and packet size in image sensor communications by employing constant power 4-PAM," *IEEE Access*, vol. 6, pp. 8000–8010, 2018.
- [11] W. Stallings, *Data and Computer Communications*, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2004.
- [12] K.-M. Seo, K.-P. Park, and B.-J. Lee, "Achieving data interoperability of communication interfaces for combat system engineering," *IEEE Access*, vol. 5, pp. 17938–17951, 2017.
- [13] J. Wang, *Handbook of Finite State Based Models and Applications*, CRC Press, Boca Raton, FL, USA, 2013.
- [14] K.-M. Seo, C. Choi, T. G. Kim, and J. H. Kim, "DEVS-based combat modeling for engagement-level simulation," *Simulation*, vol. 90, no. 7, pp. 759–781, 2014.
- [15] M. Hofmann, J. Palić, and G. Mihelcic, "Epistemic and normative aspects of ontologies in modelling and simulation," *Journal of Simulation*, vol. 5, no. 3, pp. 135–146, 2017.
- [16] S. Y. Diallo, J. J. Padilla, R. Gore, H. Herencia-Zapana, and A. Tolk, "Toward a formalism of modeling and simulation using model theory," *Complexity*, vol. 19, no. 3, 63 pages, 2014.
- [17] B. Cai, H. Liu, and M. Xie, "A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks," *Mechanical Systems and Signal Processing*, vol. 80, pp. 31–44, 2016.
- [18] G. Lamperti and X. Zhao, "Diagnosis of active systems by semantic patterns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 8, pp. 1028–1043, 2014.
- [19] J. Poon, P. Jain, I. C. Konstantakopoulos, C. Spanos, S. K. Panda, and S. R. Sanders, "Model-based fault detection and identification for switching power converters," *IEEE Transactions on Power Electronics*, vol. 32, no. 2, pp. 1419–1430, 2017.
- [20] A. F. de Loza, D. Henry, J. Cieslak, A. Zolghadri, and J. Dávila, "Sensor fault diagnosis using a non-homogeneous high-order sliding mode observer with application to a transport aircraft," *IET Control Theory & Applications*, vol. 9, no. 4, pp. 598–607, 2015.
- [21] V. T. Do and U.-P. Chong, "Signal model-based fault detection and diagnosis for induction motors using features of vibration signal in two-dimension domain," *Strojniški vestnik – Journal of Mechanical Engineering*, vol. 57, no. 9, pp. 655–666, 2011.
- [22] N. Pan, X. Wu, Y.-L. Chi, X. Liu, and C. Liu, "Combined failure acoustical diagnosis based on improved frequency domain blind deconvolution," *Journal of Physics: Conference Series*, vol. 364, article 012078, 2012.
- [23] X. Li, D. Chang, H. Pen, X. Y. Liu, and Y. Yao, "Application of MVVM design pattern in MES," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1374–1378, Shenyang, China, June 2015.
- [24] The Korea Economic Daily, "DSME wins 179 Bil. Won project to renovate 3 submarines," 2014, <http://english.hankyung.com/business/2014/07/22/1314111/spanclasskeywordd smespan-wins-179-bil-won-project-to-spanclasskeywordreno vatespan-3-submarines>.
- [25] D. V. Schrick, "Remarks on terminology in the field of supervision, fault detection and diagnosis," in *Proceedings of the IFAC Symposium on Fault Detection, Supervision Safety for Technical Processes*, pp. 959–964, Kingston upon Hull, UK, August 1997.
- [26] M. R. Khaefi, J.-Y. Im, and D.-S. Kim, "An efficient DDS node discovery scheme for naval combat system," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pp. 1–8, Luxembourg, September 2015.
- [27] K.-M. Seo, W. Hong, and T. G. Kim, "Enhancing model composability and reusability for entity-level combat simulation: a conceptual modeling approach," *Simulation*, vol. 93, no. 10, pp. 825–840, 2017.
- [28] NAVSEA, "Combat system engineering and integration," 2017, http://www.navsea.navy.mil/Portals/103/Documents/NSWC_Dahlgren/LeadingEdge/CSEI/CombSys.pdf.
- [29] P.-Y. Chen, S. Yang, and J. A. McCann, "Distributed real-time anomaly detection in networked industrial sensing systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3832–3842, 2015.
- [30] C. H. Porter, C. Villalobos, D. Holzworth et al., "Harmonization and translation of crop modeling data to ensure interoperability," *Environmental Modelling & Software*, vol. 62, no. 2014, pp. 495–508, 2014.
- [31] B. S. Ahmed, K. Z. Zamli, W. Afzal, and M. Bures, "Constrained interaction testing: a systematic literature study," *IEEE Access*, vol. 5, pp. 25706–25730, 2017.

- [32] S. Lin, Y. Wang, and L. Jia, "System reliability assessment based on failure propagation processes," *Complexity*, vol. 2018, Article ID 9502953, 19 pages, 2018.
- [33] S. D. Eppinger, N. R. Joglekar, A. Olechowski, and T. Teo, "Improving the systems engineering process with multilevel analysis of interactions," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 28, no. 4, pp. 323–337, 2014.
- [34] E. Santos and Y. Zhao, "Automatic emergence detection in complex systems," *Complexity*, vol. 2017, Article ID 3460919, 24 pages, 2017.
- [35] B. Sklar, *Digital Communications Fundamentals and Applications*, Prentice-Hall, Upper Saddle River, NJ, USA, 2001.
- [36] B. A. Forouzan, *Data Communications and Networking*, McGraw-Hill, New York, NY, USA, 2006.
- [37] Z. L. Wang, X. Yin, and C. M. Jing, "A formal method to real-time protocol interoperability testing," *Science in China Series F: Information Sciences*, vol. 51, no. 11, pp. 1723–1744, 2008.
- [38] S.-C. Shin, J.-G. Shin, and D.-K. Oh, "Development of data analysis tool for combat system integration," *International Journal of Naval Architecture and Ocean Engineering*, vol. 5, no. 1, pp. 147–160, 2013.
- [39] R. Isermann, "Model-based fault-detection and diagnosis – status and applications," *Annual Reviews in Control*, vol. 29, no. 1, pp. 71–85, 2005.
- [40] X. Liu, Z. Gao, and M. Z. Q. Chen, "Takagi–Sugeno fuzzy model based fault estimation and signal compensation with application to wind turbines," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5678–5689, 2017.
- [41] Z. Gao, X. Liu, and M. 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, 2015.
- [42] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques – part I: fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.
- [43] V. H. Ferreira, R. Zanghi, M. Z. Fortes et al., "A survey on intelligent system application to fault diagnosis in electric power system transmission lines," *Electric Power Systems Research*, vol. 136, pp. 135–153, 2016.
- [44] I. V. de Bessa, R. M. Palhares, M. F. S. V. D'Angelo, and J. E. Chaves Filho, "Data-driven fault detection and isolation scheme for a wind turbine benchmark," *Renewable Energy*, vol. 87, pp. 634–645, 2016.
- [45] X. Shuiqing, Z. Ke, C. Yi, H. Yigang, and F. Li, "Gear fault diagnosis in variable speed condition based on multiscale chirplet path pursuit and linear canonical transform," *Complexity*, vol. 2018, Article ID 3904598, 8 pages, 2018.
- [46] B. S. Kim, B. G. Kang, S. H. Choi, and T. G. Kim, "Data modeling versus simulation modeling in the big data era: case study of a greenhouse control system," *Simulation*, vol. 93, no. 7, pp. 579–594, 2017.
- [47] S. J. E. Taylor, A. Khan, K. L. Morse et al., "Grand challenges for modeling and simulation: simulation everywhere—from cyberinfrastructure to clouds to citizens," *Simulation*, vol. 91, no. 7, pp. 648–665, 2015.
- [48] W. A. Khan, A. M. Khattak, M. Hussain et al., "An adaptive semantic based mediation system for data interoperability among health information systems," *Journal of Medical Systems*, vol. 38, no. 8, p. 28, 2014.
- [49] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld, and P. Warndorf, "Improving machine tool interoperability using standardized interface protocols: MTConnect™," in *Proceedings of 2008 International Symposium on Flexible Automation*, pp. 1–6, Atlanta, GA, USA, June 2008.
- [50] E. Sorensen and M. I. Mikailcsc, "Model-view-ViewModel (MVVM) design pattern using Windows Presentation Foundation (WPF) technology," *MegaByte Journal*, vol. 9, no. 4, pp. 1–19, 2010.
- [51] R. Garofalo, *Building Enterprise Applications with Windows Presentation Foundation and the Model View View Model Pattern*, Microsoft Press, 2011.
- [52] B. P. Ziegler, T. G. Kim, and H. Praehofer, *Theory of Modeling and Simulation Integrating Discrete Event and Continuous Complex Dynamic Systems*, Academic Press, 2000.
- [53] B.-G. Kang, K.-M. Seo, and T. G. Kim, "Communication analysis of network-centric warfare via transformation of system of systems model into integrated system model using neural network," *Complexity*, vol. 2018, Article ID 6201356, 16 pages, 2018.
- [54] S. Choi, K.-M. Seo, and T. Kim, "Accelerated simulation of discrete event dynamic systems via a multi-fidelity modeling framework," *Applied Sciences*, vol. 7, no. 10, p. 1056, 2017.
- [55] A. Backurs and P. Indyk, "Which regular expression patterns are hard to match?," in *Proceedings of the 57th Annual Symposium on Foundations of Computer Science*, pp. 457–466, New Brunswick, NJ, USA, October 2016.
- [56] Defense-Aerospace, "France awards contract for the mid-life upgrade of its Mirage 2000D fighters," 2016, <http://www.defense-aerospace.com/article-view/release/175655/france-awards-mlu-contract-for-mirage-2000d-fleet.html>.
- [57] Navy Recognition, "Navantia & Indra to modernize Indonesian Navy corvette KRI Malahayati combat system," 2016, August 2017, <http://www.navyrecognition.com/index.php/news/defence-news/2016/november-2016-navy-navalforces-defense-industry-technology-maritime-security-global-news/4559-navantia-indra-to-modernize-indonesian-navy-corvette-kri-malahayaticombat-system.html>.
- [58] Black & Veatch, "Plant improvement engineering services," 2018, <https://www.bv.com/docs/energy-brochures/plant-improvement-engineering-services.pdf>.
- [59] Austin Energy, "Pumping system improvement project saves energy and improves performance at a power plant project summary," 2018, <https://www.nrel.gov/docs/fy05osti/37537.pdf>.

Research Article

Networked Control System Design for Turbofan Aeroengines with Aging and Deterioration

Ruichao Li ¹, Sing Kiong Nguang ², Yingqing Guo ¹ and Yifeng Chen ¹

¹Northwestern Polytechnical University, Xi'an, Shaanxi, China

²The University of Auckland, Auckland, New Zealand

Correspondence should be addressed to Ruichao Li; liruichao@mail.nwpu.edu.cn

Received 30 May 2018; Accepted 19 August 2018; Published 11 October 2018

Academic Editor: Yan-Ling Wei

Copyright © 2018 Ruichao 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.

This paper is concerned with designing a networked controller for a mixed flow two-spool turbofan aeroengine with aging and deterioration. Firstly, the state-space representation of the aeroengine considering aging and deterioration is identified, by which the engine system with aging and deterioration is modeled as an uncertain linear system. Then based on this uncertain linear system, theoretical results from the networked control systems and the regional pole assignment are introduced to formulate the networked engine control design in the form of linear matrix inequalities (LMIs). By solving these LMIs simultaneously, a networked engine controller is obtained which guarantees both the robustness against delay/dropout and the satisfactory dynamic performance. Finally, the proposed method is applied to an aerothermodynamic component-level engine simulator to demonstrate its validity and applicability. The corresponding delay/dropout margin is also calculated, which provides reference for the future development of the distributed engine control system.

1. Introduction

The next revolution in turbine engine control systems would be the physical distribution of control functionality, known as distributed engine control (DEC) [1]. The concept of DEC is borrowed from the distributed control system in automated industry, which is featured by the application of advanced data buses, smart sensors, and smart actuators [2, 3]. The realization of DEC is beneficial to the implementation of the advanced control mode and control algorithm, such as active control, fuzzy control, and model-based control, which bring advantages of weight reduction, modularity, high reliability, high intelligence, etc. [4–10].

However, due to immaturity of the high-temperature electronics [11, 12], transition from the centralized control system to the fully distributed control system will be a gradual process. An intermediate distributed system approach is called the networked engine control (NEC) system, in which the data bus is introduced to replace the analog signal transmission. The A/D conversion functions (sampling, shaping, and quantization) of the centralized controller are moved

into smart sensors while the control logic still remains as the central form (single controller).

Due to the introduction of the data bus, transmission delay and packet dropout are inevitable in a NEC system. Depending on how network-induced delays and packet dropouts are handled, a number of methods are developed on stability analysis and control design for networked control systems (NCSs). Among them, time-delay system approach and switched system approach are the two mainstream methods. The time-delay system approach yue 2004 state, Jiang 2008A, e.g., [13, 14], models the packet dropout as input-delay, and therefore, the maximum allowable number of consecutive packet dropouts is calculated. The switched system approach models the NCS as a switched system with arbitrary switching Zhang 2008 modelling, Zhang 2009 robust, Donkers 2011 stability, Kruszewski 2012 switched, e.g., [15–18], or known probability switching [19, 20], which is usually applied in the discrete-time domain.

As a safety-critical power device, it is crucial to figure out to what degree the control system could tolerate the delay and the packet dropout. To settle this problem, different

kinds of NEC system models are established by scholars in the aeroengine control field. Constant delay and deterministic dropout are considered by [21, 22] who calculate the worst case bound on the number of consecutive dropouts. The case of stochastic dropout has also been considered. Reference [23] models the packet dropout as a Bernoulli process and calculates the maximum allowable packet dropping probability for the decentralized NEC system. Random delays in the NEC system are considered by [24, 25].

Although a lot of work has been done on the NEC systems, the requirement of the dynamic performance has not been fully considered. Since previous results are developed based on the Lyapunov stability theory, the overall system is Lyapunov stable under the network-induced factors but it may still have a very small decay rate if no constraints are imposed. Besides, the effect of the uncertainty has also not been considered in the design stage; thus, previous results are still far away from practical application. Based on the discussion above, the goal of this paper is to design a NEC system satisfying the following control requirements:

- (1) Robustness against the transmission delay and packet dropout
- (2) Satisfactory dynamical performance under different degrees of aging and deterioration
- (3) Zero tracking error

To satisfy these control requirements, theoretical results from the networked control systems and the regional pole placement [26, 27] which have been studied previously are considered as solutions. In this paper, these theories are further extended to an uncertain case and formulated in the form of LMIs which are numerically tractable. Besides, efforts are also made to establish the uncertain description for aeroengines to make the proposed method applicable.

This paper is organized as follows. Section 2 introduces the modeling process of the aeroengine. Section 3 presents the stability theory and controller design of the NEC system. In Section 4, the robust networked controller and its corresponding delay/dropout margin are calculated. Then simulations are performed to demonstrate the effectiveness of the proposed method. Discussions are also extended on further simulation results. Finally, Section 5 presents conclusions. The symbols used in this paper are listed in Table 1.

2. Modeling of the Aeroengine

In this section, the working principle of the two-spool turbofan aeroengine and its commercial simulator is introduced first. Then by treating the complex simulator as a black box, a group of linear models is identified. These linear models are further redescribed as a nominal model with norm-bounded uncertainty to facilitate the controller design.

TABLE 1: Nomenclature.

Symbol	Type	Meaning
$\ \cdot\ $	Function	Euclidean vector norm
*	Function	Ellipsis for the symmetric terms
$\text{diag}(\dots)$	Function	Yielding a block diagonal matrix
$\text{sym}(\mathbf{A})$	Function	Denoting $\mathbf{A} + \mathbf{A}^T$
\mathbf{I}	Matrix	Identity matrix of appropriate dimensions
u, x, y, w	Vector	Input, state, output, perturbation
$x_{ss}, y_{ss}, u_{ss}, w_{ss}$	Vector	Steady-state values of u, x, y, w
$\delta u, \delta x, \delta y$	Vector	Deviation between u, x, y and x_{ss}, y_{ss}, u_{ss}
u_i, x_i, y_i, w_i	Scalar	The i th element in u, x, y, w
z	Vector	State of the integrator
\bar{x}	Vector	Augmented state
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$	Matrix	Identified system matrix
$\bar{\mathcal{A}}, \bar{\mathcal{B}}$	Matrix	Augmented identified system matrix
$\mathcal{A}_l, \mathcal{B}_l, \mathcal{C}_l, \mathcal{D}_l$	Matrix	Identified system matrix under the l th health condition
$\bar{\mathcal{A}}_l, \bar{\mathcal{B}}_l$	Matrix	Augmented identified system matrix
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	Matrix	Nominal system matrix
$\Delta\mathbf{A}, \Delta\mathbf{B}, \Delta\mathbf{C}, \Delta\mathbf{D}$	Matrix	Norm-bounded uncertainty
$\mathbf{K}_x, \mathbf{K}_z, \mathbf{K}$	Matrix	Feedback gain matrix
\mathcal{L}, \mathcal{M}	Set	Set whose members contain a group of $\mathcal{A}_l, \mathcal{B}_l, \mathcal{C}_l, \mathcal{D}_l$
$\bar{\mathcal{L}}, \bar{\mathcal{M}}$	Set	Set whose members contain a group of $\bar{\mathcal{A}}_l, \bar{\mathcal{B}}_l$
\mathcal{N}	Matrix	Interval matrix
$\mathbf{G}_i, \mathbf{E}_i (i \in \{a, b, c, d\})$	Matrix	Constant matrix
$\bar{\mathbf{G}}, \bar{\mathbf{E}}_a, \bar{\mathbf{E}}_b$	Matrix	Augmented constant matrix
$\mathbf{F}_i (i \in \{a, b, c, d\})$	Matrix	Uncertain matrix satisfying $\mathbf{F}_i^T \mathbf{F}_i \leq \mathbf{I}$
$\bar{\mathbf{F}}$	Matrix	Augmented uncertain matrix satisfying $\bar{\mathbf{F}}^T \bar{\mathbf{F}} \leq \mathbf{I}$
h	Scalar	Sampling period
η	Scalar	Delay/dropout margin
τ_{\max}	Scalar	Maximum allowable transmission delay
n_{\max}	Scalar	Maximum allowable number of consecutive packet dropouts
$\mathcal{C}(-q, r)$	Set	A disk region of which the center is $-q$ and the radii is r

2.1. *Mixed Flow Two-Spool Turbofan Aeroengine.* The cross-section of a mixed flow two-spool turbofan aeroengine is given in Figure 1. In a two-spool aeroengine, the compressors

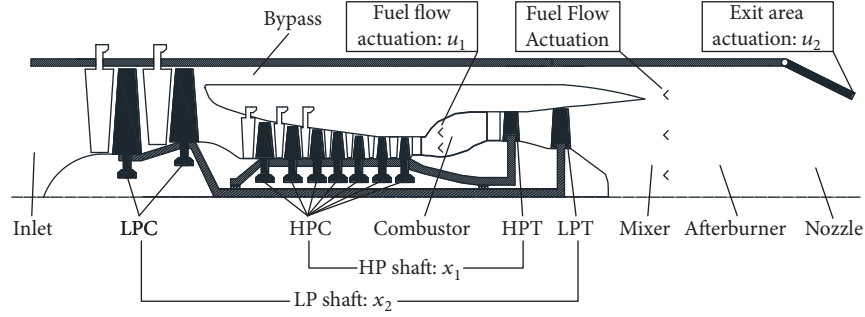


FIGURE 1: Cross-section of a mixed flow two-spool turbofan aeroengine.

TABLE 2: Structure and physical meaning of the identified model.

Signal type	Notation	Designed value	Physical meaning
Input u	u_1	5813 kg/h	Main combustor fuel flow
	u_2	0.26 m ²	Nozzle exit area
State x	x_1	8880 rpm	Rotation speed of the LP shaft
	x_2	16419 rpm	Rotation speed of the HP shaft
Output y	y_1	8880 rpm	Being the same with x_1
	y_2	10.65	Turbine pressure ratio (ratio of static pressure at the HPC exit and total pressure at the LPT exit)
Health parameter w	w_1, \dots, w_8	1.00, \dots , 1.00	Flow modifiers and efficiency modifiers of the LPC, HPC, HPT, and LPT

and turbines are linked by concentric shafts which rotate independently, namely, the low-pressure compressor (LPC) is driven by the low-pressure turbine (LPT) and the high-pressure compressor (HPC) is powered by the high-pressure turbine (HPT).

In [28], using GasTurb modeling software, an aerothermodynamic component-level aeroengine simulator is established for the mixed flow two-spool turbofan aeroengine with realistic structural parameter and features of compressors/turbines. The turbine engine modeling software, GasTurb, is a commercial software [29, 30] which is widely used for modeling and simulating gas turbine aeroengines. For the convenience in controller design, the aeroengine simulator developed in [28] is compiled under the MATLAB/Simulink environment further. However, due to the limited space and the modeling of simulators being not the main focus of the paper, the modeling process of the simulator is not included in this paper. Readers who are interested could obtain a general knowledge about the modeling process from the gas turbine textbook (see chap. 2.5 of [28, 31]).

The nonlinear state and output equations of the aeroengine simulator are expressed as

$$\begin{cases} \dot{x} = f(x, u, w), \\ y = g(x, u, w), \end{cases} \quad (1)$$

where f and g are nonlinear functions of the state vector x , the input vector u , and the health parameter vector w as given in Table 2. The degradation is due to usage and aging results in the higher fuel consumption and exhaust gas temperature, which further yields a shorter component life and a higher cost. Generally, the degradation is reflected as changes in flow characteristics and efficiencies of the rotational components. Therefore, a group of multipliers, which are referred as “health parameters,” are introduced. For instance, $w_i = 1$ ($i = 1, \dots, 8$) denotes a new component while a value being lower than 1 denotes a degraded one. For the engine which needs an overhaul, the health parameters may degrade to a maximum of 0.99.

2.2. State-Space Models Obtained by the System Identification. Considering the complexity of the simulator, the system identification technique is utilized to obtain a state-space representation. At the steady-state operating point $(x_{ss}, y_{ss}, u_{ss}, w_{ss})$, the simulator is identified as

$$\begin{cases} \dot{x}(t) = \mathcal{A} \cdot \delta x(t) + \mathcal{B} \cdot \delta u(t), \\ \delta y(t) = \mathcal{C} \cdot \delta x(t) + \mathcal{D} \cdot \delta u(t), \end{cases} \quad (2)$$

where symbol δ denotes the deviation from the steady-state point.

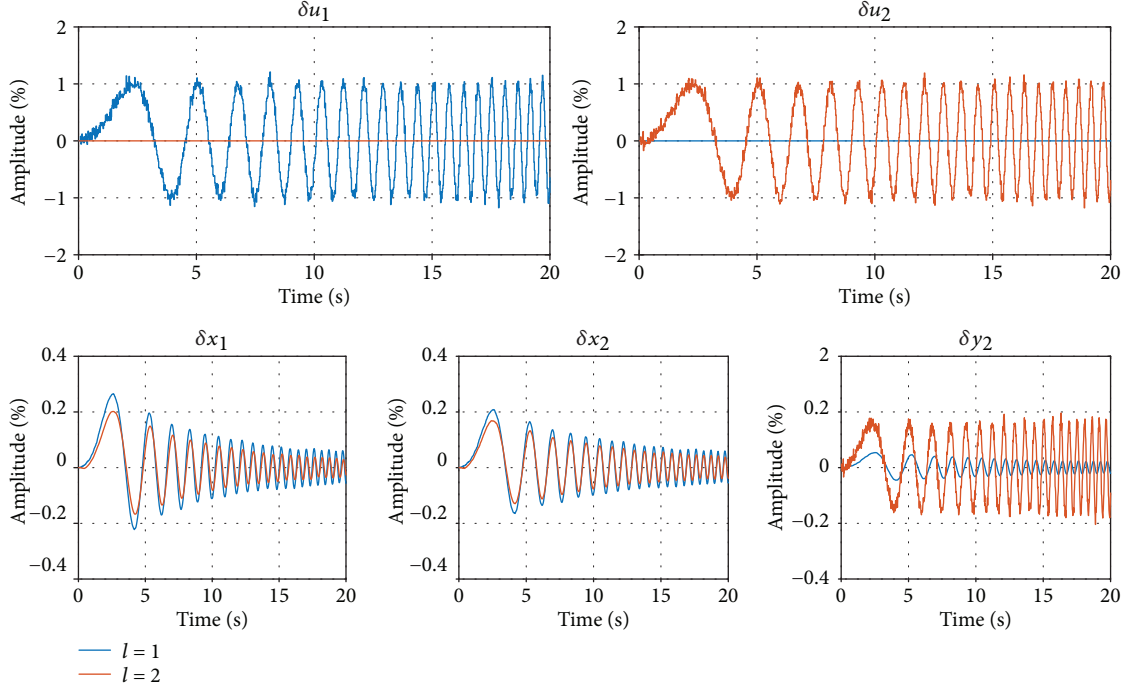


FIGURE 2: Data generated for the system identification.

Take the “full health” status $w = [1, \dots, 1]^T$ as an example. To apply the system identification technology, outputs of the aeroengine simulators y_1 and y_2 are first driven to their designed value given in Table 2; then two excitation signals $\delta u^l(t) (l=1, 2)$ are added separately to the input channels to generate $\delta x^l(t)$ and $\delta y^l(t) (l=1, 2)$. These two excitation signals are taken, respectively, as

$$\begin{aligned} \delta u^1(t) &= \begin{bmatrix} \text{chirp}(t) \\ 0 \end{bmatrix}^T + \omega(t), \\ \delta u^2(t) &= \begin{bmatrix} 0 \\ \text{chirp}(t) \end{bmatrix}^T + \omega(t), \end{aligned} \quad (3)$$

where $\text{chirp}(t)$ is a swept-frequency cosine signal with amplitude being $1\% u_{\text{designed}}$. As stated in chap. 3.1 of [31], frequencies of the dominant dynamics are generally not higher than 2 Hz for aeroengines. Therefore, the frequency of $\text{chirp}(t)$ is set to increase linearly from 0 to 2 Hz within 20 s, which is adequate to extract the spectral characteristics of rotors. Moreover, zero-mean white Gaussian noises $\omega(t)$ are also added to the excitation signals such that the signal-to-noise ratio (SNR) satisfies

$$\text{SNR} = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2} = 100, \quad (4)$$

in which σ denotes the standard deviation.

The excitation signals $\delta u^l = [u_1^l(t), u_2^l(t)]^T (l=1, 2)$ and the deviations $\delta x_1^l, \delta x_2^l,$ and $\delta y_2^l (l=1, 2)$ generated from the aeroengine simulator are depicted in Figure 2, which have already been normalized by their designed value. With these two samples, the system matrices $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ under the “full-health” status are identified by the method proposed in [32] as

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} -5.792 & 4.818 \\ 0.612 & -4.390 \end{bmatrix}, \\ \mathcal{B} &= \begin{bmatrix} 0.628 & 0.363 \\ 0.734 & 0.579 \end{bmatrix}, \\ \mathcal{C} &= \begin{bmatrix} 1 & 0 \\ -0.578 & 1.900 \end{bmatrix}, \\ \mathcal{D} &= \begin{bmatrix} 0 & 0 \\ 0.033 & 0.707 \end{bmatrix}, \end{aligned} \quad (5)$$

For simplicity, it is assumed that all health parameters $w_i (i=1, \dots, 8)$ degrade simultaneously. Then by varying w_i from 1 to 0.99 evenly and still maintaining y at its designed value y_{designed} given in Table 3, a group of matrices $\mathcal{A}_l, \mathcal{B}_l, \mathcal{C}_l,$ and $\mathcal{D}_l (l=1, 2, \dots, 11)$ is obtained via the system identification method above, which constitutes the set \mathcal{L} below

$$\mathcal{L} \triangleq \left\{ \Lambda | \Lambda = \begin{bmatrix} \mathcal{A}_l & \mathcal{B}_l \\ \mathcal{C}_l & \mathcal{D}_l \end{bmatrix}, \quad l=1, 2, \dots, 11 \right\}. \quad (6)$$

TABLE 3: Delay/dropout margin of each disk region ($r = 0.6q$).

	$-q$	-6	-7	-8	-9	-10	-11	-12	-13
η_{\max}	0.143	0.121	0.104	0.090	0.079	0.070	0.062	0.056	
$-q$	-14	-15	-16	-17	-18	-19	-20		
η_{\max}	0.050	0.046	0.041	0.038	0.034	0.031	0.028		

Due to the limited space, the values of these system matrices are not listed here.

2.3. Derivation of the Norm-Bounded Uncertainty. In this paper, degradations of the health parameters w (representing aging and deterioration) are viewed as the source of uncertainties for the identified state-space models. The degradation has two adverse effects on the identified linear models: On the one hand, the degradation is reflected in the drift of the steady-state value, and on the other hand, it causes the perturbation of the system matrices.

Since the identified system matrices \mathcal{A} , \mathcal{B} , \mathcal{C} , and \mathcal{D} vary with w , they are supposed to be decomposed as

$$\begin{aligned}\mathcal{A} &= \mathbf{A} + \Delta\mathbf{A}, \\ \mathcal{B} &= \mathbf{B} + \Delta\mathbf{B},\end{aligned}$$

$$\mathcal{M} \triangleq \left\{ \Lambda \mid \Lambda = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_a \mathbf{F}_a \mathbf{E}_a & \mathbf{G}_b \mathbf{F}_b \mathbf{E}_b \\ \mathbf{G}_c \mathbf{F}_c \mathbf{E}_c & \mathbf{G}_d \mathbf{F}_d \mathbf{E}_d \end{bmatrix}, \quad \mathbf{F}_i^T \mathbf{F}_i \leq \mathbf{I}, i \in \{a, b, c, d\} \right\}. \quad (9)$$

To the best of our knowledge, there exists no transformation method such that \mathcal{L} equals \mathcal{M} .

Therefore, efforts are made to find a set \mathcal{M} which could encompass \mathcal{L} , namely, $\mathcal{L} \subseteq \mathcal{M}$. This is handled via the concept of *interval matrix* introduced below.

Definition 1 [33]. With the given matrices $\mathbf{P} \triangleq (p_{ij})_{n \times n}$, $\mathbf{Q} \triangleq (q_{ij})_{n \times n}$, and $\mathcal{A} \triangleq (a_{ij})_{n \times n}$, notation $\mathcal{N}[\mathbf{P}, \mathbf{Q}]$ is used to denote the set of matrix \mathcal{A} which satisfies $p_{ij} \leq a_{ij} \leq q_{ij}$, namely,

$$\mathcal{N}[\mathbf{P}, \mathbf{Q}] \triangleq \left\{ \mathcal{A} \in \mathbb{R}^{n \times n} \mid p_{ij} \leq a_{ij} \leq q_{ij}, \quad i, j = 1, 2, \dots, n \right\}. \quad (10)$$

Then $\mathcal{N}[\mathbf{P}, \mathbf{Q}]$ is called interval matrix, in which \mathbf{P} and \mathbf{Q} are the lower- and upper-bound matrices, respectively.

Based on the lemma given below, the interval matrix $\mathcal{N}[\mathbf{P}, \mathbf{Q}]$ can be converted equivalently to a matrix with norm-bounded uncertainty. Therefore, the uncertain representation is obtained by seeking a lower-bound matrix \mathbf{P} and an upper-bound matrix \mathbf{Q} such that $\mathcal{A}_l \in \mathcal{N}[\mathbf{P}, \mathbf{Q}] (l = 1, 2, \dots, 11)$.

$$\begin{aligned}\mathcal{C} &= \mathbf{C} + \Delta\mathbf{C}, \\ \mathcal{D} &= \mathbf{D} + \Delta\mathbf{D},\end{aligned} \quad (7)$$

in which $\Delta\mathbf{A}$, $\Delta\mathbf{B}$, $\Delta\mathbf{C}$ and $\Delta\mathbf{D}$ are norm-bounded uncertainties satisfying

$$\begin{aligned}\Delta\mathbf{A} &= \mathbf{G}_a \mathbf{F}_a \mathbf{E}_a, \\ \Delta\mathbf{B} &= \mathbf{G}_b \mathbf{F}_b \mathbf{E}_b, \\ \Delta\mathbf{C} &= \mathbf{G}_c \mathbf{F}_c \mathbf{E}_c, \\ \Delta\mathbf{D} &= \mathbf{G}_d \mathbf{F}_d \mathbf{E}_d,\end{aligned} \quad (8)$$

where \mathbf{G}_i and $\mathbf{E}_i (i \in \{a, b, c, d\})$ are constant matrices of appropriate dimensions. \mathbf{F}_i is the uncertain matrix with Lebesgue measurable elements that satisfy $\mathbf{F}_i^T \mathbf{F}_i \leq \mathbf{I}$.

When obtaining the uncertain state-space representation (7), the difficulty lies in how to decompose a group of matrices \mathcal{A}_l further into a uniform norm-bounded description $\mathbf{A} + \mathbf{G}_a \mathbf{F}_a \mathbf{E}_a$, namely, transforming \mathcal{L} to the set \mathcal{M} defined below.

Lemma 1 [34]. *The interval matrix $\mathcal{N}[\mathbf{P}, \mathbf{Q}]$ is equivalent to the set \mathcal{M}_a defined below.*

$$\mathcal{M}_a \triangleq \left\{ \mathcal{A} \in \mathbb{R}^{n \times n} \mid \mathcal{A} = \mathbf{A} + \mathbf{G}_a \mathbf{F}_a \mathbf{E}_a, \mathbf{F}_a^T \mathbf{F}_a \leq \mathbf{I} \right\}, \quad (11)$$

in which matrices $\mathbf{G}_a \in \mathbb{R}^{n \times n^2}$, $\mathbf{F}_a \in \mathbb{R}^{n^2 \times n^2}$, and $\mathbf{E}_a \in \mathbb{R}^{n^2 \times n}$ are taken as

$$\begin{aligned}\mathbf{A} &= \frac{(\mathbf{P} + \mathbf{Q})}{2}, \\ \mathbf{H} &\triangleq (h_{ij})_{n \times n} = \frac{(\mathbf{Q} - \mathbf{P})}{2}, \\ \mathbf{G}_a &= \left[\sqrt{h_{11}} \mathbf{e}_1 \cdots \sqrt{h_{1n}} \mathbf{e}_1 \cdots \sqrt{h_{n1}} \mathbf{e}_n \cdots \sqrt{h_{nn}} \mathbf{e}_n \right], \\ \mathbf{E}_a &= \left[\sqrt{h_{11}} \mathbf{e}_1 \cdots \sqrt{h_{1n}} \mathbf{e}_n \cdots \sqrt{h_{n1}} \mathbf{e}_1 \cdots \sqrt{h_{nn}} \mathbf{e}_n \right]^T, \\ \mathbf{e}_i &= \left[\underbrace{0, \dots, 0, 1, 0, \dots, 0}_i \right]^T \in \mathbb{R}^n, \quad (i = 1, 2, \dots, n), \\ \mathbf{F}_a &= \text{diag} (\varepsilon_{11}, \dots, \varepsilon_{1n}, \dots, \varepsilon_{n1}, \dots, \varepsilon_{nn}), \quad |\varepsilon_{ij}| \leq 1.\end{aligned} \quad (12)$$

Remark 1. The equivalence of $\mathcal{N}[\mathbf{P}, \mathbf{Q}]$ and \mathcal{M}_a has been proved in [34] and thus is omitted here. Readers could verify equalities $\mathbf{P} = \mathbf{A} - \mathbf{G}_a \mathbf{I}_{n^2} \mathbf{E}_a$ and $\mathbf{Q} = \mathbf{A} + \mathbf{G}_a \mathbf{I}_{n^2} \mathbf{E}_a$ for ease of understanding.

With the given matrices $\mathcal{A}_l (l = 1, 2, \dots, 11)$, by selecting “tight” bound matrices \mathbf{P} and \mathbf{Q} such that $\mathcal{A}_l \in \mathcal{N}[\mathbf{P}, \mathbf{Q}]$, matrices \mathbf{A} , \mathbf{G}_a , and \mathbf{E}_a are calculated according to Lemma 1 as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -5.248 & 4.080 \\ 1.012 & -4.933 \end{bmatrix}, \\ \mathbf{G}_a &= \begin{bmatrix} 0.738 & 0.859 & 0 & 0 \\ 0 & 0 & 0.632 & 0.737 \end{bmatrix}, \\ \mathbf{E}_a &= \begin{bmatrix} 0.738 & 0 & 0.632 & 0 \\ 0 & 0.859 & 0 & 0.737 \end{bmatrix}^T. \end{aligned} \quad (13)$$

Similarly, matrices \mathbf{B} , \mathbf{C} , \mathbf{D} , \mathbf{G}_i , and $\mathbf{E}_i (i \in \{b, c, d\})$ in (7) are calculated as

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} 0.631 & 0.389 \\ 0.743 & 0.576 \end{bmatrix}, \\ \mathbf{G}_b &= \begin{bmatrix} 0.061 & 0.160 & 0 & 0 \\ 0 & 0 & 0.091 & 0.060 \end{bmatrix}, \\ \mathbf{E}_b &= \begin{bmatrix} 0.061 & 0 & 0.091 & 0 \\ 0 & 0.160 & 0 & 0.060 \end{bmatrix}^T, \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 \\ -0.759 & 2.107 \end{bmatrix}, \\ \mathbf{G}_c &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.425 & 0.455 \end{bmatrix}, \\ \mathbf{E}_c &= \begin{bmatrix} 0 & 0 & 0.425 & 0 \\ 0 & 0 & 0 & 0.455 \end{bmatrix}^T, \\ \mathbf{D} &= \begin{bmatrix} 0 & 0 \\ 0.034 & 0.700 \end{bmatrix}, \\ \mathbf{G}_d &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.042 & 0.080 \end{bmatrix}, \\ \mathbf{E}_d &= \begin{bmatrix} 0 & 0 & 0.042 & 0 \\ 0 & 0 & 0 & 0.080 \end{bmatrix}^T. \end{aligned} \quad (14)$$

Therefore, the set \mathcal{M} encompassing the set \mathcal{L} is obtained to redescribe a group of linear models (6) as a nominal matrix (9) with norm-bounded uncertainties.

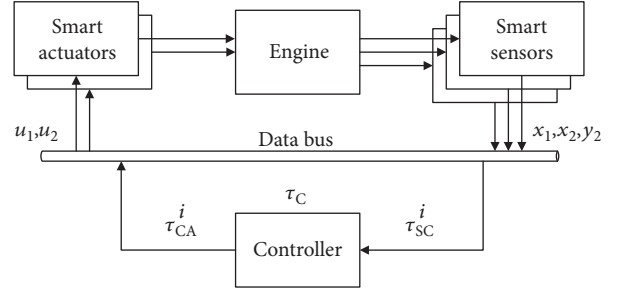


FIGURE 3: Schematic of the NEC system architecture.

3. Networked Controller Design for the Aeroengine

In this section, the mathematical expression of the NEC system is presented which takes the transmission delay and the packet dropout into account simultaneously. Then a robust networked engine controller is designed with requirements of robustness against delay/dropout and the satisfactory dynamic performance being met with the NCS stability theory and the regional pole placement, respectively. The former will be presented in Section 3.2 while the latter in Section 3.3. Then by combining linear matrix inequalities (LMIs) from these two theories, the controller design is synthesized in Section 3.4.

3.1. Networked Engine Control System. A schematic of the NEC system architecture is shown in Figure 3, where $\tau_{SC}^i (i = 1, 2, 3)$, $\tau_{CA}^i (i = 1, 2)$, and τ_C denote the transmission delay between the sensors and the controller, the transmission delay between the controller and the actuators, and the calculation time of the controller, respectively.

To eliminate the potential tracking error due to reasons like modeling biases or plant disturbances, integrators are incorporated in the controller. Based on [35], by putting integrators at the output y , format of the controller is taken as

$$\delta u(t) = \mathbf{K}_x \cdot \delta x(t) + \mathbf{K}_z \cdot z(t), \quad (15)$$

where z is the state of the integrator satisfying

$$z(t) = \int e(t) dt = \int [(\delta y)_{\text{cmd}} - \delta y(t)] dt. \quad (16)$$

Supposed that the sensors are clock driven, the controller and actuators are event driven and the data are transmitted with a single packet and the integrated network-induced delay τ can be approximated as $\tau = \max_i(\tau_{SC}^i) + \tau_C + \max_j(\tau_{CA}^j)$. When packet dropout occurs, the latest packet is utilized for computation and actuation.

Then combining (2) with (15) and following the modeling process in [13], the closed-loop system is expressed as

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathcal{A} \cdot \delta x(t) + \mathcal{B} \cdot \delta u(t), \\ \delta \mathbf{y}(t) = \mathcal{C} \cdot \delta x(t) + \mathcal{D} \cdot \delta u(t), \\ t \in [i_k h + \tau_k, i_{k+1} h + \tau_{k+1}), \\ \delta \mathbf{u}(t) = \mathbf{K}_x \cdot \delta x(t - \tau_k) + \mathbf{K}_z \cdot z(t - \tau_k), \\ t = i_k h + \tau_k, i_k \in \{1, 2, 3, \dots\}, k = 1, 2, \dots, \end{cases} \quad (17)$$

in which h is the sampling period. τ_k is the transmission delay from the sampling instant $i_k h$ to the actuating instant $i_k h + \tau_k$. $\{i_k\}$ is a nondecreasing sequence satisfying $i_{k+1} \geq i_k$ if the data packet disorder is ignored.

More specifically, $i_{k+1} = i_k + 1$ means that no packet dropout occurs while $i_{k+1} = i_k$ means a packet loss in the transmission. By the definition of the symbol $\{i_k\}$, it can be seen that (17) models the transmission delay and the packet dropout simultaneously.

Furthermore, by defining an augmented state vector as $\bar{x} \triangleq [x \quad z]^T$ and considering the output reference (δy) to be zero here, the closed-loop system (17) can be expressed compactly as

$$\begin{cases} \dot{\bar{x}}(t) = \bar{\mathcal{A}} \cdot \delta \bar{x}(t) + \bar{\mathcal{B}} \cdot \delta u(t), \\ t \in [i_k h + \tau_k, i_{k+1} h + \tau_{k+1}), \\ \delta u(t) = \mathbf{K} \cdot \delta \bar{x}(t - \tau_k), \\ t = i_k h + \tau_k, i_k \in \{1, 2, 3, \dots\}, k = 1, 2, \dots, \end{cases} \quad (18)$$

in which $\mathbf{K} \triangleq [\mathbf{K}_x \quad \mathbf{K}_z]$. $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ are system matrices that can still be decomposed into a norm-bounded form as

$$\begin{aligned} \bar{\mathbf{A}} &= \bar{\mathbf{A}} + \Delta \bar{\mathbf{A}}, \\ \bar{\mathbf{B}} &= \bar{\mathbf{B}} + \Delta \bar{\mathbf{B}}, \\ [\Delta \bar{\mathbf{A}} \quad \Delta \bar{\mathbf{B}}] &= \bar{\mathbf{G}} \bar{\mathbf{F}} [\bar{\mathbf{E}}_a \quad \bar{\mathbf{E}}_b] \end{aligned} \quad (19)$$

with

$$\begin{aligned} \bar{\mathbf{A}} &= \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix}, \\ \bar{\mathbf{B}} &= \begin{bmatrix} \mathbf{B} \\ -\mathbf{D} \end{bmatrix}, \\ \bar{\mathbf{G}} &= \begin{bmatrix} \mathbf{G}_a & 0 & \mathbf{G}_b & 0 \\ 0 & \mathbf{G}_c & 0 & \mathbf{G}_d \end{bmatrix}, \\ \bar{\mathbf{E}}_a &= \begin{bmatrix} \mathbf{E}_a & \mathbf{E}_c & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^T, \\ \bar{\mathbf{F}} &= \text{diag}(\mathbf{F}_a, -\mathbf{F}_c, \mathbf{F}_b, -\mathbf{F}_d), \\ \bar{\mathbf{E}}_b &= [0 \quad 0 \quad \mathbf{E}_b \quad \mathbf{E}_d]^T. \end{aligned} \quad (20)$$

The matrices $\bar{\mathcal{A}}$, $\bar{\mathcal{B}}$, $\bar{\mathbf{G}}$, $\bar{\mathbf{E}}_a$, and $\bar{\mathbf{E}}_b$ can be calculated based on (13) and (14), but they are not listed here due to their high dimensions.

3.2. Stability Theory under Delay/Dropout Condition. In the following, the state feedback controller $\delta u = \mathbf{K} \cdot \delta \bar{x}$ will be designed. Before we start, a frequently used lemma is introduced below which eliminates the uncertain term in LMIs effectively without introducing conservatism. Thus, by this lemma, existing theorems can be extended for uncertain systems straightforwardly.

Lemma 2 [36]. *With the given matrices \mathbf{M} , \mathbf{N} with appropriate dimensions, and a symmetric matrix \mathbf{W} , the matrix inequality*

$$\mathbf{W} + \mathbf{N}^T \mathbf{F}^T \mathbf{M}^T + \mathbf{M} \mathbf{F} \mathbf{N} < 0, \quad (21)$$

holds for all \mathbf{F} satisfying $\mathbf{F}^T \mathbf{F} \leq \mathbf{I}$ if and only if there exists a scalar $\varepsilon > 0$ such that

$$\mathbf{W} + \varepsilon \mathbf{M} \mathbf{M}^T + \varepsilon^{-1} \mathbf{N}^T \mathbf{N} < 0. \quad (22)$$

By Lemma 2, the theorem introduced below guarantees the robustness against delay/dropout for the closed-loop system (18).

Theorem 1 *For the given scalars $\eta \geq (i_{k+1} - i_k)h + \tau_{k+1}$ and $\rho_j (j=2, 3)$, the closed-loop system (18) is exponentially asymptotically stable with $\mathbf{K} = \mathbf{Y} \mathbf{X}^{-1}$ if there exist a scalar $\varepsilon > 0$ and matrices $\mathbf{P} > 0$, $\mathbf{T} > 0$, $\mathbf{X} > 0$, \mathbf{Y} , and $\mathbf{N}_i (i=1, 2, 3)$ with appropriate dimensions such that (23) holds.*

$$\begin{aligned} &\Theta(\eta, \rho_2, \rho_3, \mathbf{P}, \mathbf{T}, \mathbf{X}, \mathbf{Y}, \varepsilon) \\ &\triangleq \begin{bmatrix} \Omega_{11} & \Omega_{12} & \Omega_{13} & \eta \mathbf{N}_1 & (\bar{\mathbf{E}}_a \mathbf{X})^T \\ * & \Omega_{22} & \Omega_{23} & \eta \mathbf{N}_2 & (\bar{\mathbf{E}}_b \mathbf{Y})^T \\ * & * & \Omega_{33} & \eta \mathbf{N}_3 & 0 \\ * & * & * & -\eta \mathbf{T} & 0 \\ * & * & * & * & -\varepsilon \mathbf{I} \end{bmatrix} < 0, \end{aligned} \quad (23)$$

where

$$\begin{aligned} \Omega_{11} &= \text{sym}(\mathbf{N}_1) - \text{sym}(\bar{\mathbf{A}} \mathbf{X}) + \varepsilon \bar{\mathbf{G}} \bar{\mathbf{G}}^T, \\ \Omega_{12} &= \mathbf{N}_2^T - \mathbf{N}_1 - \bar{\mathbf{B}} \mathbf{Y} - \rho_2 \mathbf{X} \bar{\mathbf{A}}^T + \varepsilon \bar{\mathbf{G}} \bar{\mathbf{G}}^T \rho_2, \\ \Omega_{13} &= \mathbf{P} + \mathbf{N}_3^T + \mathbf{X} - \rho_3 \mathbf{X} \bar{\mathbf{A}}^T + \varepsilon \bar{\mathbf{G}} \bar{\mathbf{G}}^T \rho_3, \\ \Omega_{22} &= -\text{sym}(\mathbf{N}_2) - \text{sym}(\rho_2 \bar{\mathbf{B}} \mathbf{Y}) + \varepsilon \rho_2 \bar{\mathbf{G}} \bar{\mathbf{G}}^T \rho_2, \\ \Omega_{23} &= -\mathbf{N}_3^T + \rho_2 \mathbf{X} - \rho_3 \mathbf{Y}^T \bar{\mathbf{B}}^T + \varepsilon \rho_2 \bar{\mathbf{G}} \bar{\mathbf{G}}^T \rho_3, \\ \Omega_{33} &= \eta \mathbf{T} + 2\rho_3 \mathbf{X} + \varepsilon \rho_3 \bar{\mathbf{G}} \bar{\mathbf{G}}^T \rho_3. \end{aligned} \quad (24)$$

Proof. It can be known from [14] that for the given scalars η and $\rho_j (j=2, 3)$, the closed-loop system (18) is exponentially asymptotically stable with $\mathbf{K} = \mathbf{Y} \mathbf{X}^{-1}$ if there exist matrices

$\mathbf{P} > 0$, $\mathbf{T} > 0$, $\mathbf{X} > 0$, \mathbf{Y} , and $\mathbf{N}_i (i = 1, 2, 3)$ with appropriate dimensions such that (25) holds.

$$\begin{bmatrix} \text{sym}(\mathbf{N}_1) - \text{sym}(\bar{\mathcal{A}}\mathbf{X}) & \mathbf{N}_2^T - \mathbf{N}_1 - \bar{\mathcal{B}}\mathbf{Y} - \rho_2\mathbf{X}\bar{\mathcal{A}}^T & \mathbf{P} + \mathbf{N}_3^T + \mathbf{X} - \rho_3\mathbf{X}\bar{\mathcal{A}}^T & \eta\mathbf{N}_1 \\ * & -\text{sym}(\mathbf{N}_2) - \text{sym}(\rho_2\bar{\mathcal{B}}\mathbf{Y}) & -\mathbf{N}_3^T + \rho_2\mathbf{X} - \rho_3\mathbf{Y}^T\bar{\mathcal{B}}^T & \eta\mathbf{N}_2 \\ * & * & \eta\mathbf{T} + 2\rho_3\mathbf{X} & \eta\mathbf{N}_3 \\ * & * & * & -\eta\mathbf{T} \end{bmatrix} < 0. \quad (25)$$

Then by substituting $[\bar{\mathcal{A}}; \bar{\mathcal{B}}] = [\bar{\mathbf{A}}; \bar{\mathbf{B}}] + \bar{\mathbf{G}}\bar{\mathbf{F}}[\bar{\mathbf{E}}_a; \bar{\mathbf{E}}_b]$ into (25) and using Lemma 2 to matrix $\bar{\mathbf{F}}$, it can be found that (25) holds if there exists a scalar $\varepsilon > 0$ satisfying (26).

$$\begin{bmatrix} \tilde{\Omega}_{11} & \tilde{\Omega}_{12} & \tilde{\Omega}_{13} & \eta\mathbf{N}_1 \\ * & \tilde{\Omega}_{22} & \tilde{\Omega}_{23} & \eta\mathbf{N}_2 \\ * & * & \tilde{\Omega}_{33} & \eta\mathbf{N}_3 \\ * & * & * & -\eta\mathbf{T} \end{bmatrix} + \varepsilon \begin{bmatrix} -\bar{\mathbf{G}} \\ -\rho_2\bar{\mathbf{G}} \\ -\rho_3\bar{\mathbf{G}} \\ 0 \end{bmatrix} \begin{bmatrix} -\bar{\mathbf{G}} \\ -\rho_2\bar{\mathbf{G}} \\ -\rho_3\bar{\mathbf{G}} \\ 0 \end{bmatrix}^T \quad (26)$$

$$+ \varepsilon^{-1} \begin{bmatrix} (\bar{\mathbf{E}}_a\mathbf{X})^T \\ (\bar{\mathbf{E}}_b\mathbf{Y})^T \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} (\bar{\mathbf{E}}_a\mathbf{X})^T \\ (\bar{\mathbf{E}}_b\mathbf{Y})^T \\ 0 \\ 0 \end{bmatrix}^T < 0,$$

where

$$\begin{aligned} \tilde{\Omega}_{11} &= \text{sym}(\mathbf{N}_1) - \text{sym}(\bar{\mathbf{A}}\mathbf{X}), \\ \tilde{\Omega}_{12} &= \mathbf{N}_2^T - \mathbf{N}_1 - \bar{\mathbf{B}}\mathbf{Y} - \rho_2\mathbf{X}\bar{\mathbf{A}}^T, \\ \tilde{\Omega}_{13} &= \mathbf{P} + \mathbf{N}_3^T + \mathbf{X} - \rho_3\mathbf{X}\bar{\mathbf{A}}^T, \\ \tilde{\Omega}_{22} &= -\text{sym}(\mathbf{N}_2) - \text{sym}(\rho_2\bar{\mathbf{B}}\mathbf{Y}), \\ \tilde{\Omega}_{23} &= -\mathbf{N}_3^T + \rho_2\mathbf{X} - \rho_3\mathbf{Y}^T\bar{\mathbf{B}}^T, \\ \tilde{\Omega}_{33} &= \eta\mathbf{T} + 2\rho_3\mathbf{X}. \end{aligned} \quad (27)$$

Then by Schur complements, (26) is equivalent to (23). The theorem is proved.

Remark 2. In Theorem 1, inequality $\eta \geq (i_{k+1} - i_k)h + \tau_{k+1}$ describes the effect of the transmission delay and packet dropout simultaneously; hence, η can be viewed as a delay/dropout margin.

- (i) If no packet drops in the transmission ($i_{k+1} = i_k + 1$), inequality $\tau_{k+1} \leq \eta_{\max} - h$ holds. The maximum allowable transmission delay τ_{\max} is calculated as

$$\tau_{\max} = \eta_{\max} - h \quad (28)$$

- (ii) Assuming a constant delay τ in the transmission, inequality $(i_{k+1} - i_k) \leq (\eta_{\max} - \tau)/h$ holds. Thus, the maximum allowable number of consecutive packet dropout n_{\max} is calculated as

$$n_{\max} = \left\lfloor \frac{\eta_{\max} - \tau}{h} \right\rfloor - 1 \quad (29)$$

3.3. D-Stability Theory. Though Theorem 1 guarantees that the closed-loop system is exponentially stable, the dynamic performance can still be poor since no constraints are imposed on the rate of decay. To obtain a satisfactory dynamic performance, a simple way is to assign the closed-loop poles of the augmented system (18), namely, the eigenvalues of $\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathbf{K}$, to some suitable regions \mathcal{D} on the left-half complex plane. To achieve this, the below is introduced first which is usually referred as the \mathcal{D} -stability theory.

Lemma 3 [37]. *With the given LMI region $\mathcal{D} \triangleq \{s \in \mathbb{C} : \mathbf{L} + s\mathbf{M} + \bar{s}\mathbf{M}^T < 0\}$ and matrix \mathbf{A} , the eigenvalues of \mathbf{A} lie in \mathcal{D} if and only if there exists a symmetric matrix $\mathbf{X} > 0$ such that*

$$\mathbf{M}_{\mathcal{D}}(\mathbf{A}, \mathbf{X}) \triangleq \mathbf{L} \otimes \mathbf{X} + \mathbf{M} \otimes (\mathbf{A}\mathbf{X}) + \mathbf{M}^T \otimes (\mathbf{A}\mathbf{X})^T < 0. \quad (30)$$

Different types of regions, including the half-open plane, vertical strip area, and conic sector, can be covered by the above LMI description. Other complex regions can also be covered as long as they are convex. To guarantee a minimum decay rate and a minimum damping ratio, in this paper, the disk region $\mathcal{C}(-q, r)$ is chosen as the expected eigenvalue region of which the center is $-q$ and the radius is r . This disk region is described by taking

$$\begin{aligned} \mathbf{L} &= \begin{bmatrix} -r & q \\ q & -r \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}. \end{aligned} \quad (31)$$

Then based on Lemma 3, the theorem below guarantees all the eigenvalues of $\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathbf{K}$ in the disk region $\mathcal{C}(-q, r)$

Theorem 2. Given a disk region $\mathcal{C}(-q, r)$ defined by (31), all eigenvalues of $\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathbf{K}$ lie in $\mathcal{C}(-q, r)$ if and only if there exist a symmetric matrix $\mathbf{X} > 0$, a matrix \mathbf{Y} with appropriate dimension, and a scalar $\varepsilon > 0$ such that

$$\Xi(q, r, \mathbf{X}, \mathbf{Y}, \varepsilon) \triangleq \begin{bmatrix} -r\mathbf{X} + \varepsilon\bar{\mathbf{G}}\bar{\mathbf{G}}^T & q\mathbf{X} + \bar{\mathbf{A}}\mathbf{X} + \bar{\mathbf{B}}\mathbf{Y} & 0 \\ * & -r\mathbf{X} & (\bar{\mathbf{E}}_a\mathbf{X} + \bar{\mathbf{E}}_b\mathbf{Y})^T \\ * & * & -\varepsilon\mathbf{I} \end{bmatrix} < 0, \quad (32)$$

with the control gain matrix being calculated as $\mathbf{K} = \mathbf{Y}\mathbf{X}^{-1}$.

Proof. Being similar to the proof of Theorem 1, by substituting $[\bar{\mathcal{A}}; \bar{\mathcal{B}}] = [\bar{\mathbf{A}}; \bar{\mathbf{B}}] + \bar{\mathbf{G}}\bar{\mathbf{F}}[\bar{\mathbf{E}}_a; \bar{\mathbf{E}}_b]$ into $M_{\mathcal{D}}(\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathbf{K}, \mathbf{X}) < 0$ and using Lemma 2 to eliminate the uncertain matrix $\bar{\mathbf{F}}$, $M_{\mathcal{D}}(\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathbf{K}, \mathbf{X}) < 0$ is equivalent to

$$M_{\mathcal{D}}(\bar{\mathbf{A}} + \bar{\mathbf{B}}\mathbf{K}, \mathbf{X}) + \varepsilon \begin{bmatrix} \bar{\mathbf{G}}^T & 0 \end{bmatrix}^T \begin{bmatrix} \bar{\mathbf{G}}^T & 0 \\ 0 & \bar{\mathbf{E}}_a\mathbf{X} + \bar{\mathbf{E}}_b\mathbf{K}\mathbf{X} \end{bmatrix} < 0. \quad (33)$$

Then by Schur complements and taking $\mathbf{K} = \mathbf{Y}\mathbf{X}^{-1}$, (33) is equivalent to (32). The theorem is proved.

3.4. Controller Design Synthesis. Based on the aforementioned work, the controller with strong delay/dropout robustness and satisfactory dynamic performance can be obtained through solving LMIs in Theorem 1 and Theorem 2 simultaneously. These LMIs are written together as

$$\begin{cases} \Theta(\eta, \rho_2, \rho_3, \mathbf{P}, \mathbf{T}, \mathbf{X}, \mathbf{Y}, \varepsilon) < 0, \\ \Xi(q, r, \mathbf{X}, \mathbf{Y}, \varepsilon_1) < 0, \\ \mathbf{P} > 0, \\ \mathbf{T} > 0, \\ \mathbf{X} > 0, \\ \varepsilon > 0, \\ \varepsilon_1 > 0, \end{cases} \quad (34)$$

in which Θ and Ξ are matrices defined in (23) and (32), respectively, and q, r, η , and $\rho_j (j=2, 3)$ are predesignated scalars.

Then for the given scalars q, r , and $\rho_j (j=2, 3)$, the maximum delay/dropout margin η_{\max} can be calculated by a linear search. Correspondingly, the control gain matrix is calculated as $\mathbf{K} = \mathbf{Y}\mathbf{X}^{-1}$.

4. Application on the Turbofan Aeroengine Simulator

In this section, the controller design proposed above is applied on the turbofan aeroengine simulator by selecting

appropriate closed-loop poles region. The controller is designed based on the uncertain model established in Section 2 and then verified on the simulator itself to validate the applicability of the proposed method.

4.1. Controller Design. Once the uncertain state-space representation is ready, the center $-q$ and the radii r of the disk pole region $\mathcal{C}(-q, r)$ are determined to calculate the control gain matrix. In practical turbofan applications, the percentage overshoot (PO) and the setting time T_{setting} of the engine control system are usually required to satisfy

$$\begin{aligned} \text{PO} &\leq 2\%, \\ T_{\text{setting}} &\leq 2 \text{ s}. \end{aligned} \quad (35)$$

By taking the damping ratio $\xi = 0.8$ (damping angle $\theta = \arccos 0.8$) and the center $q = 6$, PO and T_{setting} are estimated by the formulae of the second-order system as

$$\begin{aligned} \text{PO} &= e^{-\pi\xi/\sqrt{1-\xi^2}} \times 100\% = 1.52\%, \\ T_{\text{setting}} &= -\frac{\ln\left(\Delta \times \sqrt{1-\xi^2}\right)}{q-r} \Bigg|_{\Delta=0.02} = 1.84\text{s}, \end{aligned} \quad (36)$$

in which $r = q \sin \theta$. This fact indicates that the dynamic performance requirements above can be met if $q \geq 6$ and $r \leq 0.6q$. Based on this criterion, by taking $q = 8$ and $r = 4.8$ and searching $\rho_j (j=2, 3)$ between $[-10, 10]$, the control gain matrix \mathbf{K} is calculated for (18) by solving LMIs in (34) as

$$\mathbf{K} = \begin{bmatrix} -8.829 & -3.244 & 41.298 & -4.678 \\ 1.378 & -1.625 & -3.259 & 6.965 \end{bmatrix}. \quad (37)$$

The corresponding delay/dropout margin is calculated as $\eta_{\max} = 0.104$. Then according to (18), the control gain matrix \mathbf{K} is implemented in the form of

$$\begin{aligned} \mathbf{K}_x &= \begin{bmatrix} -8.829 & -3.244 \\ 1.378 & -1.625 \end{bmatrix}, \\ \mathbf{K}_z &= \begin{bmatrix} 41.298 & -4.678 \\ -3.259 & 6.965 \end{bmatrix}. \end{aligned} \quad (38)$$

In the NEC system, the transmission delay τ is bounded in one sampling period h if a time-triggered data bus is utilized [23]. Therefore, taking τ as its worst value $h = 0.02\text{s}$, $\eta_{\max} = 0.104$ indicates a maximum allowable number of consecutive packet dropout $n_{\max} = 3$ based on Remark 2, namely, the NEC system is stable under a consecutive packet dropout being not severer than 3 in every 4 sampling cycles when the time delay $\tau = 0.02 \text{ s}$.

4.2. Simulation. For comparison, another controller $\mathbf{K}_{\text{without}\Delta}$ is calculated under the same condition but without

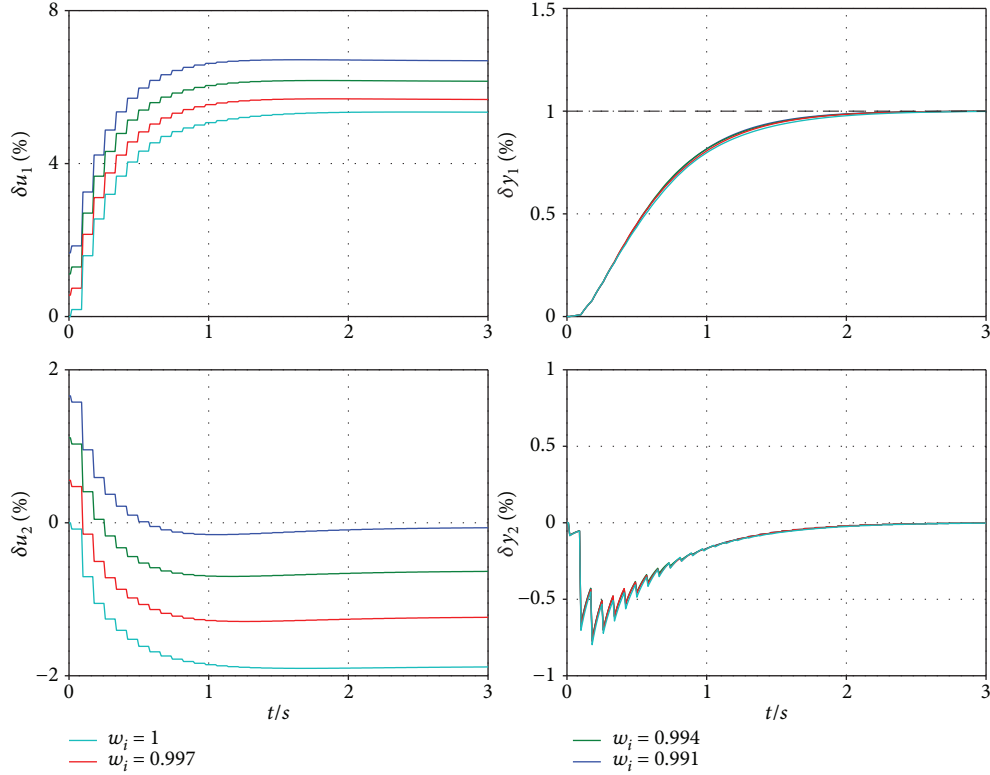


FIGURE 4: Step response under various deteriorations with \mathbf{K} .

considering the uncertainties. Comparison is performed between these two controllers to verify the effectiveness of our design.

By applying them on the component-level aeroengine simulator and varying the degree of deterioration w , unit step response simulations are conducted on the output y_1 as shown in Figures 4 and 5. In each simulation, the health parameter $w_i (i=1, \dots, 8)$ is taken as $\{1, 0.997, 0.994, 0.991\}$, while the network condition is taken as the extreme situation $\tau = 0.02s$ and $n_{\max} = 3$ as discussed above. The control input u and the output y in these figures have already been normalized by their designed values u_{designed} and y_{designed} .

Comparing Figure 4 (with uncertainties) with Figure 5 (without uncertainties), one can see that the settling time in Figure 4 is smaller than that in Figure 5. This shows that the controller design that is taking uncertainties into consideration is more robust and has a smaller settling time than the one without considering the uncertainties. Besides, it can be observed from Figure 4 that

- (1) The NEC system remains stable under the transmission delay and packet dropout described by the delay/dropout margin $\eta_{\max, \mathcal{E}(-8, 4, 8)} = 0.104$
- (2) The output y_1 presents a satisfactory as well as a consistent dynamic performance under different deteriorations. Besides, the response curves do not overshoot significantly since the disk region has a small imaginary part (damping angle $\theta = \arcsin 0.6$)

- (3) The output could track the reference command without the steady-state error

From the points listed above, the robust networked controller designed in this paper has achieved the expected goal, namely, the three control requirements proposed in Introduction.

4.3. Discussions. In this section, discussions are extended on the relationship between the delay/dropout margin and the disk region $\mathcal{E}(-q, r)$. Besides, the conservatism which is introduced when obtaining the norm-bounded uncertainties is also discussed.

4.3.1. Relationship between the Delay/Dropout Margin and the Disk Region. As mentioned above, the expected disk region $\mathcal{E}(-q, r)$ should satisfy $q \geq 6$ and $r \leq 0.6q$. Therefore, by varying the center $q \in [6, 20]$ and taking the radii r as $r = 0.6q$ for a constant damping ratio, different gain matrices $K_{\mathcal{E}(-q, r)}$ are and their delay/dropout margins η_{\max} are calculated in Table 3.

From Table 3, it also can be observed that the delay/dropout margins decrease rapidly while the pole region is being far away from the imaginary axis. Therefore, there should be a trade-off between the robustness against delay/dropout and the dynamical performance when designing the controller.

4.3.2. Conservatism Introduced in the Modeling Process. In this paper, a data-based method is applied to obtain the

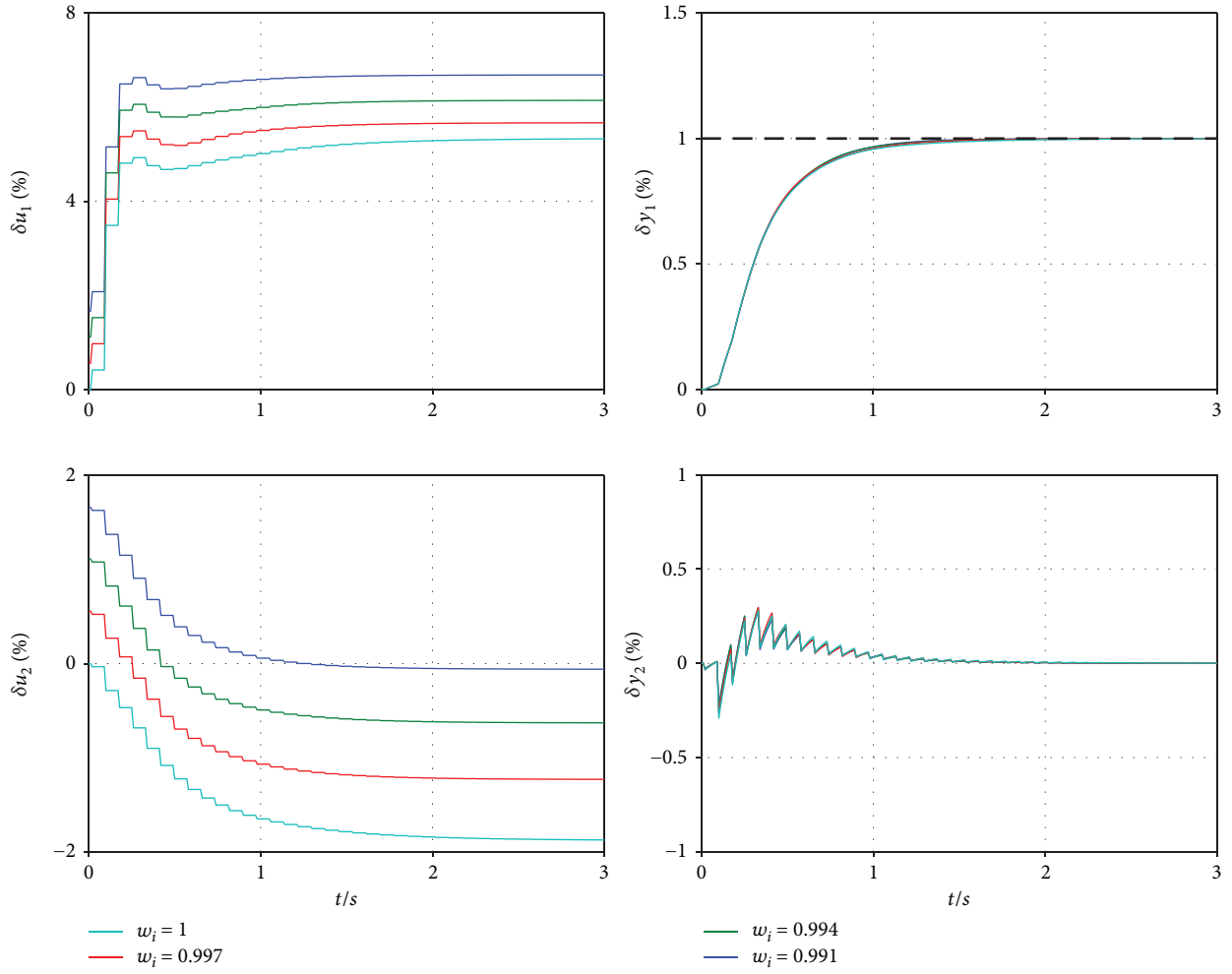


FIGURE 5: Step response under various deteriorations with $K_{\text{without}\Delta}$

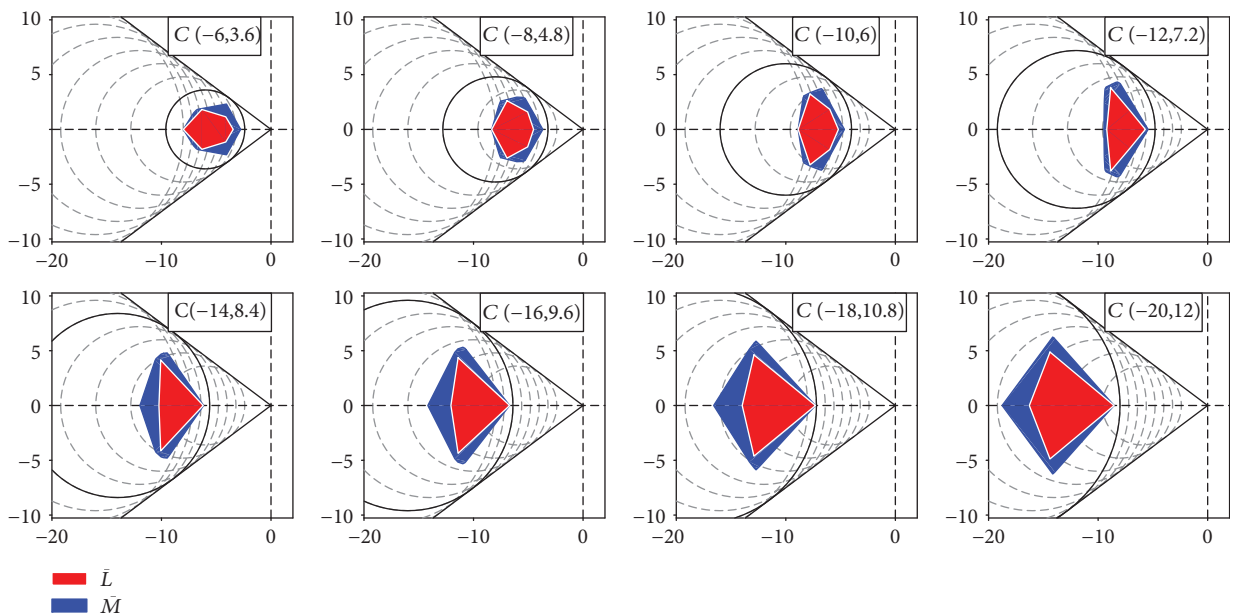


FIGURE 6: Convex hull of the closed-loop poles under different control gain matrices $K_{\mathcal{C}(-q,r)}$

norm-bounded uncertainties in Section 2.3. By this method, a group of system matrices (\mathcal{L} in (6)) is redescribed as a nominal matrix with norm-bounded uncertainties (\mathcal{M} in (9)). However, conservatism is introduced in this modeling process since \mathcal{L} is only a subset of \mathcal{M} . To demonstrate the conservatism graphically, integrators are incorporated as mentioned in Section 3.1 to augment \mathcal{L} and \mathcal{M} to $\bar{\mathcal{L}}$ and $\bar{\mathcal{M}}$ defined below as

$$\begin{aligned} \bar{\mathcal{L}} &\triangleq \{[\bar{A}_1; \bar{B}_1], \dots, [\bar{A}_{10}; \bar{B}_{11}]\} \text{ with} \\ \bar{A}_1 &\triangleq \begin{bmatrix} A_1 & \mathbf{0} \\ -C_1 & \mathbf{0} \end{bmatrix}, \\ \bar{B}_l &\triangleq \begin{bmatrix} B_l \\ -D_l \end{bmatrix} (l = 1, 2, \dots, 11) \end{aligned} \quad (39)$$

and

$$\begin{aligned} \bar{\mathcal{M}} &\triangleq \{[\bar{A}; \bar{B}] \mid [\bar{A}; \bar{B}] = [\bar{A}; \bar{B}] \\ &\quad + \bar{\mathbf{G}}\bar{\mathbf{F}}[\bar{\mathbf{E}}_a; \bar{\mathbf{E}}_b], \bar{\mathbf{F}}^T\bar{\mathbf{F}} \leq \mathbf{I}\} \end{aligned} \quad (40)$$

in which $\bar{\mathcal{M}}$ is exactly the augmented system matrices in (18).

Based on the gain matrices of various disk regions obtained in the previous section, the closed-loop poles (namely, the eigenvalues of $\bar{\mathcal{A}} + \bar{\mathcal{B}}\mathbf{K}_{\mathcal{G}(-q,r)}$) are calculated for $\bar{\mathcal{L}}$ and $\bar{\mathcal{M}}$. As a result, the convex hulls of these poles are drawn in Figure 6, in which the pole region of $\bar{\mathcal{L}}$ is denoted in red while the counterpart of $\bar{\mathcal{M}}$ is denoted in blue. It can be seen that certain conservatism is introduced but is still at an acceptable level.

4.3.3. Conservatism Introduced in the Controller Design Process. The conservatism is not just introduced by the modeling but also introduced by the controller. In this subsection, we combine the networked controller design theory [13] with the \mathcal{D} -stability theorem to design a controller for the uncertain system (18) as a comparison.

By using the same disk region selected in this paper and taking the lower bound of time delay τ_m as 0, the maximum allowable value of η_{\max} is calculated as 0.112 based on [13]. By contrast, η_{\max} calculated in this paper is 0.104. Considering a dropout-free transmission, the gap means a larger maximum allowable delay τ_{\max} by 8 ms which is small enough. Therefore, only a little conservatism is introduced in the controller design process.

However, it should be pointed out that the theorem in [13] is in the form of bilinear matrix inequality (BMI), which has to be solved by a complex iterative algorithm. In contrast, the method proposed in this paper is numerically tractable. This makes the controller much easier to be calculated with existing LMI toolbox.

5. Conclusion

In this paper, efforts are made to establish the uncertain description for aeroengines with aging and deterioration. Based on this model, the controller with both strong delay/

dropout robustness and satisfactory dynamic performance is designed through solving linear matrix inequalities (LMIs). The validity and applicability of this approach are elaborated through simulations under the deteriorated and networked conditions. The corresponding delay/dropout margin is also calculated, which provides a reference for future development of the aeroengine-distributed control system. Admittedly, with some degree of conservatism, the controller design approach presented in this paper offered a feasible and practical mean to obtain a robust networked controller

Data Availability

Data can be provided if needed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Assoc. Prof. Jiwei Wen in Jiangnan University and Mister Qian Feng in The University of Auckland for their beneficial suggestions.

References

- [1] D. E. Culley, R. Thomas, and J. Saus, "Concepts for distributed engine control," in *43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 5709, Cincinnati, OH, USA, July 2007.
- [2] J. A. DeCastro, C. A. Palmer, and A. Behbahani, "Meeting the requirements of distributed engine control via decentralized, modular smart sensing," in *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 6748, Nashville, TN, USA, July 2010.
- [3] O. A. Watts, L. Frediani, and M. W. Usrey, "Technical opportunities for high temperature smart p3 sensors and electronics for distributed engine control," in *52nd AIAA/SAE/ASEE Joint Propulsion Conference*, p. 4512, Salt Lake City, UT, USA, 2016.
- [4] A. Nasiri, S. K. Nguang, and A. Swain, "Adaptive sliding mode control for a class of mimo nonlinear systems with uncertainties," *Journal of the Franklin Institute*, vol. 351, no. 4, pp. 2048–2061, 2014.
- [5] L. Ruichao, G. Yingqing, S. K. Nguang, and C. Yifeng, "Takagi-sugeno fuzzy model identification for turbofan aero-engines with guaranteed stability," *Chinese Journal of Aeronautics*, vol. 31, no. 6, pp. 1206–1214, 2018.
- [6] T. M. Seitz, O. B. Macmann, A. R. Behbahani, and F. Khoury, "Development of distributed control systems for aircraft turbofan engines," in *52nd AIAA/SAE/ASEE Joint Propulsion Conference*, p. 4805, Salt Lake City, UT, USA, 2016.
- [7] G. L. Thomas, D. E. Culley, and A. Brand, "The application of hardware in the loop testing for distributed engine control," in *52nd AIAA/SAE/ASEE Joint Propulsion Conference*, p. 4654, Salt Lake City, UT, USA, 2016.
- [8] Y. Wei, J. Qiu, and H.-K. Lam, "A novel approach to reliable output feedback control of fuzzy-affine systems with time delays and sensor faults," *IEEE transactions on fuzzy systems*, vol. 25, no. 6, pp. 1808–1823, 2017.

- [9] Y. Wei, J. Qiu, P. Shi, and M. Chadli, "Fixed-order piecewise-affine output feedback controller for fuzzy-affine-model-based nonlinear systems with time-varying delay," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 4, pp. 945–958, 2017.
- [10] J. Zhang, P. Shi, J. Qiu, and S. K. Nguang, "A novel observer-based output feedback controller design for discrete-time fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 223–229, 2015.
- [11] A. Behbahani, M. Usrey, and Y. Liu, "High-temperature sensor and electronics packaging technologies for distributed engine controls," in *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, p. 3533, Cleveland, OH, USA, 2014.
- [12] M. Pakmehr, M. Dhingra, N. Fitzgerald et al., "Distributed architectures integrated with high-temperature electronics for engine monitoring and control," in *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 6148, San Diego, CA, USA, July–August 2011.
- [13] X. Jiang, Q. L. Han, S. Liu, and A. Xue, "A new H_∞ stabilization criterion for networked control systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 4, pp. 1025–1032, 2008.
- [14] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," in *Proceedings of the 2004 IEEE International Conference on Control Applications*, 2004, pp. 242–247, Taipei, Taiwan, September 2004.
- [15] M. Donkers, W. Heemels, N. Van de Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2101–2115, 2011.
- [16] A. Kruszewski, W.-J. Jiang, E. Fridman, J. P. Richard, and A. Toguyeni, "A switched system approach to exponential stabilization through communication network," *IEEE transactions on control systems technology*, vol. 20, no. 4, pp. 887–900, 2012.
- [17] W.-A. Zhang and L. Yu, "Modelling and control of networked control systems with both network-induced delay and packet-dropout," *Automatica*, vol. 44, no. 12, pp. 3206–3210, 2008.
- [18] W.-A. Zhang and L. Yu, "A robust control approach to stabilization of networked control systems with time-varying delays," *Automatica*, vol. 45, no. 10, pp. 2440–2445, 2009.
- [19] D. Huang and S. K. Nguang, "Robust control for uncertain networked control systems with random delays," in *Lecture Notes in Control and Information Sciences*, vol. 386, Springer, London, 2009.
- [20] J. Wen, L. Peng, and S. K. Nguang, "Finite-time control for discrete-time markovian jump systems with deterministic switching and time-delay," *International journal of control, automation and systems*, vol. 12, no. 3, pp. 473–485, 2014.
- [21] R. K. Belapurkar, *Stability and Performance of Propulsion Control Systems with Distributed Control Architectures and Failures*, [Ph. D. thesis], The Ohio State University, 2012.
- [22] R. Li, Y. Guo, Y. Chen, and Z. Zhao, "Networked guaranteed cost control for turbofan aero-engines," in *2017 36th Chinese Control Conference (CCC)*, pp. 251–256, Dalian, China, July 2017.
- [23] R. K. Yedavalli, R. K. Belapurkar, and A. Behbahani, "Design of distributed engine control systems for stability under communication packet dropouts," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 5, pp. 1544–1549, 2009.
- [24] J. Kratz and R. K. Yedavalli, "A riccati equation based robust control design with application to a gas turbine engine leading toward distributed control," in *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, p. 3639, Cleveland, OH, USA, 2014.
- [25] T. Seitz, R. Yedavalli, O. Macmann, and A. Behbahani, "Robust control of turbine engines using distributed networks with multiple time delays," *Journal of guidance, control, and dynamics*, vol. 40, no. 11, pp. 2771–2781, 2017.
- [26] S. K. Nguang and W. Assawinchaichote, " H_∞ filtering for fuzzy dynamical systems with D stability constraints," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 11, pp. 1503–1508, 2003.
- [27] S. K. Nguang and P. Shi, "Robust H_∞ output feedback control design for fuzzy dynamic systems with quadratic D stability constraints: an LMI approach," *Information Sciences*, vol. 176, no. 15, pp. 2161–2191, 2006.
- [28] S.-G. Zhang, Y.-Q. Guo, and J. Lu, "Research on aircraft engine component-level models based on gas Turb/MATLAB," *Journal of aerospace power*, vol. 27, no. 12, pp. 2850–2856, 2012.
- [29] J. Kurzke, "How to create a performance model of a gas turbine from a limited amount of information," in *ASME Turbo Expo 2005: Power for Land, Sea, and Air*, pp. 145–153, Reno, Nevada, USA, June 2005.
- [30] J. Kurzke, "About simplifications in gas turbine performance calculations," in *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pp. 493–501, Montreal, Canada, May 2007.
- [31] L. C. Jaw and J. D. Mattingly, *Aircraft Engine Controls: Design, System Analysis, and Health Monitoring*, American Institute of Aeronautics and Astronautics, 2009.
- [32] J. Lu, Y.-q. GUO, and X.-l. CHEN, "Establishment of aero-engine state variable model based on linear fitting method," *Journal of aerospace power*, vol. 26, no. 5, pp. 1172–1177, 2011.
- [33] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, Society for Industrial and Applied Mathematics, Siam, 2009.
- [34] F. Wu, Z. Shi, and G. Dai, "On robust stability of dynamic interval systems," *Control Theory & Applications*, vol. 18, no. 1, pp. 113–115, 2001.
- [35] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*, vol. 240, Prentice Hall, Upper Saddle River, NJ, USA, 2001.
- [36] L. Xie, M. Fu, and C. E. de Souza, " H_∞ control and quadratic stabilization of systems with parameter uncertainty via output feedback," *IEEE Transactions on Automatic Control*, vol. 37, no. 8, pp. 1253–1256, 1992.
- [37] M. Chilali and P. Gahinet, " H_∞ design with pole placement constraints: an LMI approach," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 358–367, 1996.

Research Article

Delay-Dependent Stability in Uncalibrated Image-Based Dynamic Visual Servoing Robotic System

Tao Li,¹ Hui Zhao ,^{1,2} and Yu Chang³

¹School of Electrical Engineering and Automation, Tianjin University, Tianjin 300072, China

²Tianjin Agricultural University, 300384, China

³School of Design, Tianjin University of Commerce, Tianjin 300134, China

Correspondence should be addressed to Hui Zhao; zhaohui3379@163.com

Received 13 May 2018; Revised 15 July 2018; Accepted 30 July 2018; Published 8 October 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Tao 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.

This paper addresses the stability problem of uncalibrated image-based visual servoing robotic systems. Both the visual feedback delay and the uncalibrated visual parameters can be the sources of instability for visual servoing robotic systems. To eliminate the negative effects caused by kinematic uncertainties and delays, we propose an adaptive controller including the delay-affected Jacobian matrix and design an adaptive law accordingly. Besides, the delay-dependent stability conditions are provided to show the relationship between the system stability and the delayed time in order to obtain less conservative results. A Lyapunov-Krasovskii functional is constructed, and a rigorously mathematic proof is given. Finally, the simulation results are presented to show the effectiveness of the proposed control scheme.

1. Introduction

For human beings, vision is an important sensory channel. Through visual sensors, robots also can monitor the circumstance and perform the tasks. Nowadays, the advanced visual processing techniques and high-speed image processors make vision-based robot systems capable of handling dynamical tasks, and the vision-based control has been applied to many industrial robot systems. It has become the mainstream of robot control.

Vision-based control can be traced back to 1980s [1]. Look-and-move is one of the early vision-based technologies [1–3]. In this approach, two nested loops run simultaneously: the visual loop is the external loop and the joint-space loop is the internal loop. Due to the sensitivities to disturbances and errors, the look-and-move architecture is not suitable for high-performance control tasks [4]. As an alternative, the visual servo (VS) technique is proposed [5]. This control architecture directly generates the control inputs using the visual information. Such a simple and direct structure is favorable for high-speed servoing tasks. Considerable visual servoing approaches have been investigated for various robot

systems and from many different aspects. Figure 1 shows two typical structures of the visual servoing control.

In the existing literature, there are two challenges in the field of visual servoing control: (a) the difficulties of calibration and (b) the image feedback signals of inferior quality.

The calibration of visual servoing systems includes the camera calibration, kinematic calibration, and dynamic calibration. For the sake of identifying unknown or uncertain system parameters, periodical and high-accurate calibration work usually is required, which is tedious and demanding. Without such calibration work, the system models cannot be accurately characterized and the closed-loop visual servoing systems could be unstable. To avoid such calibration work, the uncalibrated control approaches are proposed [6–10]. Some work [6–8] investigates the approaches with robust controllers for eliminating the negative effects of calibration errors of the system model, and the uncertain system parameters in the above work are replaced with the approximated ones. As for the case of unknown parameters or time-varying parameters, adaptive control techniques are proposed [11, 12]. To handle such parameters, in these methods, adaptive laws are designed to update them online.

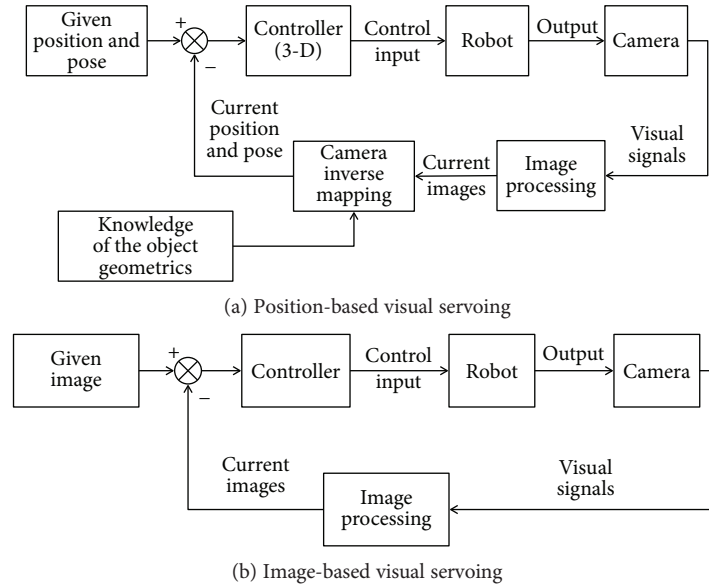


FIGURE 1: Visual servoing control scheme structure.

As another cause of system instability, the visual signals of inferior quality are also nonnegligible. Generally speaking, noise and delays in the visual signals are the main inducements. In this paper, we consider delays as the main reason for inferior image signals. As we know, the visual signal flows are expected to be synchronized with other system signals. However, asynchronization could happen due to many reasons including the limitation of image processing [13–17] and restriction of visual signal transmission [18]. Some early research studies the instability problem caused by image-processing (or image-sampling) delays [1, 2, 13]. These early efforts focused on reducing the image sampling time through a parallel or pipelined approach [2, 14–17]. With the development of the advanced image processor chips, such a problem has been resolved to a large extent. With the wide application of visual servo, its patterns are becoming various. The connections between visual sensors and controllers can be wireless or Internet, which also means the visual feedback path can be a source of delays due to the transmission block in the intercomponent information exchanges [1, 2, 13, 18]. Improving the speed and the reliability of communication links is a straightforward way to address the issue. But it inevitably leads to the increase of the cost. Consequently, designing proper control schemes to handle with delays is an alternative. In the IBVS control scheme design, the delay problem is studied by [19–23]. Using average joint angle values of the past and present moment to replace the present joint angle, [19] obtains predicted image feature position values by the Jacobian matrix to cope with the delay problem. One common flaw of the aforementioned methods is that they require the accurate knowledge of system parameters, and the acquirement of such information is based on calibration.

The two challenges make a visual servoing robotic system become typical complex industrial systems. This is because the mainstream noncalibration techniques usually require accurate image signals to compensate for the parametric

errors or to update the unknown parameters. Under the delayed image feedback loop, there is no accurate synchronized visual feedback available. In this context, the control of such systems is of high nonlinearity and complexity. Consequently, it is worthwhile and challenging to be investigated. This paper therefore will concentrate on the influence of visual transmission delays upon the uncalibrated visual servoing robotic systems.

In the literature of this area, [21] presents an online calibration method to overcome the time delay problem. Inoue and Hirai [22] design a two-layer controller called STP to compensate for the delays and the concept of virtual trajectory is introduced. Gao and Su [23] employ local fitting Jacobian matrix based on polynomial fitting to obtain more accurate Jacobian estimation and image precompensation for uncalibrated IBVS robotic system. Unfortunately, the controller design in the above literature is based on kinematics and fails to consider the dynamics of robots. It is well known that the dynamics of robot systems plays an important role in the stability, especially in the case of high speed. Much progress has been made in the aspect of the uncalibrated dynamic-based visual servoing systems control without delay effects [24–31]. As for the uncalibrated dynamic-based visual servoing systems with the delay effects, the relevant work focuses on the area of the distributed cooperative control [32–35]. Liu and Chopra [33] study an adaptive control algorithm to guarantee task-space synchronization of networked robotic manipulators in the presence of dynamic uncertainties and time-varying communication delays. Wang [34] investigates the problem of synchronization of networked robotic systems with kinematic and dynamic uncertainties in the case of nonuniform constant communication delays. Liang et al. [35] address cooperative tracking control problem of networked robotic manipulators in the presence of communication delays under strongly connected directed graphs. However, the above work considers the delays in the interagent information exchanges

rather than delays existing in the visual feedback of single dynamic-based visual servoing robotic system. To the best of the authors' knowledge, there is little literature considering the time delay problem in an uncalibrated dynamic-based visual servoing robotic system without using image-space velocity measurements. To address the aforementioned issues, the following problems are expected to be addressed. First, the modeling of the system. Delays, noncalibration, and velocity measurements, these contributing factors, need to simultaneously be included in the modeling of the system. Second, the handling of the time-varying parameters and how to avoid using image velocity measurements. Third, the delay-dependent stability conditions are expected to be given for obtaining less conservatism. The contributions of this paper can be summarized as follows. (a) An uncalibrated dynamic-based visual servoing model is developed to visual track a feature point whose image depth is time varying without using the image velocity and in the presence of unknown constant delayed visual feedback. (b) To handle overlapped effects of uncalibrated parameter uncertainties and the visual feedback delay, the novel Jacobian matrix called delay-affected Jacobian matrix is first proposed in this paper. (c) Lyapunov-Krasovskii stability theory is employed to analyze the stability of the delay-affected dynamic-based visual servoing system, and delay-dependent (d.d) stability conditions are given to obtain less conservative stability results.

The paper is organized as follows. Section 2 gives some preliminary knowledge used throughout the paper. In Section 3, the kinematic and dynamic models of dynamic-based visual servoing robotics systems are formulated. In Section 4, the main results of this paper, the controller design, and the adaptive laws are proposed to address the stability problem of the uncalibrated dynamic-based visual servoing robotic system with visual feedback delays. In Section 5, rigorous stability analyses are provided. Section 6 presents simulation results to show the effectiveness of the proposed control scheme. Section 7 concludes the paper.

2. Preliminaries

Lemma 1. *Let X , Y , and F be real matrices with proper dimensions, where $F^T F \leq I$. For any constant $\varepsilon > 0$, the following holds.*

$$XFY + Y^T F^T X^T \leq \varepsilon XX^T + \frac{1}{\varepsilon} Y^T Y. \quad (1)$$

Lemma 2. *Let $\phi : R \rightarrow R$ be a uniformly continuous function on $[0, \infty)$. Suppose that $\lim_{t \rightarrow \infty} \int_0^t \phi(\tau) d\tau$ exists and is finite. Then,*

$$\phi(t) \rightarrow 0 \text{ as } t \rightarrow \infty. \quad (2)$$

Lemma 3. *Consider functional differential equation*

$$\dot{x}(t) = f(t, x_t). \quad (3)$$

Let $f : R \times C \rightarrow R^n$ be a mapping from $R \times$ (a bounded subset of C) to a bounded subset of R^n , $u, v, w : \bar{R}_+ \rightarrow \bar{R}_+$ are continuous nondecreasing functions, for any $s > 0$, $u(s), v(s)$ are positive and $u(0) = v(0) = 0$. If there exists a continuous differentiable functional $V : R \times C \rightarrow R$, such that

$$u(\|\phi(0)\|) \leq V(t, \phi) \leq v(\|\phi\|_c), \quad (4)$$

and

$$\dot{V}(t, \phi) \leq -\omega(\|\phi(0)\|), \quad (5)$$

the zero solution of system (8) is uniformly stable. If zero solution of the system is uniformly stable and for any $s > 0$, $\omega(s) > 0$ holds, then zero solution of system (8) is uniformly asymptotically stable. If zero solution of the system is uniformly asymptotically stable and $\lim_{s \rightarrow \infty} \omega(s) = \infty$, then zero solution of system (8) is globally uniformly asymptotically stable.

3. Kinematics and Dynamics

In this section, we present the mathematical modeling of delayed visual servoing robotic systems with the eye-in-hand configuration. In the modeling process, both kinematics and dynamics are considered. To illustrate the kinematics of the system, Figure 2 shows the transformation among different frames.

Let $y(t) \in \mathbb{R}^2$ be the coordinates of a feature point's projection on the camera image plane and $r \in \mathbb{R}^3$ be the Cartesian coordinates of the feature point w.r.t the robot base frame. Based on the model developed in [7], the mapping between image position $y(t)$ and the Cartesian position r can be formulated as

$$y(t) = \frac{1}{z(t)} \begin{pmatrix} \Omega_1 \\ \Omega_2 \end{pmatrix} T_b^c \begin{pmatrix} r \\ 1 \end{pmatrix}, \quad (6)$$

where $z(t) \in \mathbb{R}$ denotes the depth of feature to the camera frame; $T_b^c \in \mathbb{R}^{4 \times 4}$ denotes the homogeneous transformation matrix from the camera frame to the base frame; $\Omega_i \in \mathbb{R}^{1 \times 4}$ denotes the i th row of the camera intrinsic parameter matrix Ω (Ω is an intrinsic parameter matrix which is derived from the typical model introduced in [36]). In reality, the feature point is stationary with respect to the robot base and it results in a constant column vector r .

The relationship between $z(t)$ and r can be formulated by

$$z(t) = \Omega_3 T_b^c \begin{pmatrix} r \\ 1 \end{pmatrix}, \quad (7)$$

where Ω_3 is the third row of the perspective projection matrix Ω .

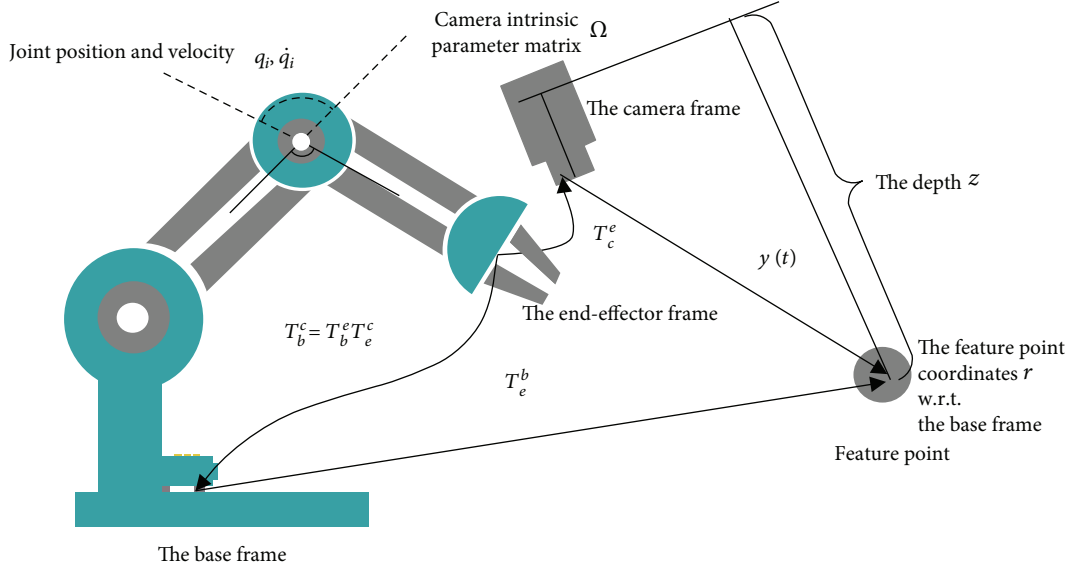


FIGURE 2: Robotic system with the eye-in-hand configuration.

Combining with (7), the derivative of (6) w.r. time t satisfies

$$\dot{y}(t) = \frac{1}{z(t)} (\Omega - y\Omega_3) \frac{\partial T_b^c}{\partial t} \begin{pmatrix} r \\ 1 \end{pmatrix}, \quad (8)$$

where y is short for $y(t)$. It should be noticed that T_b^c can be divided into two parts: T_b^e (the forward robot kinematics) and T_e^c (homogeneous matrix from the camera to the end effector). Due to the eye-in-hand configuration, T_e^c is a constant matrix. Then, one has

$$\frac{\partial T_b^c}{\partial t} \begin{pmatrix} r \\ 1 \end{pmatrix} = T_e^c \frac{\partial}{\partial q} (Rr + P) \dot{q}, \quad (9)$$

where R denotes the rotation matrix, P denotes the translation vector, and q denotes the joint position. For more details, see [7]. By letting the matrix L be the left 3×3 submatrix of ΩT_e^c , $L_0 \in \mathbb{R}^{2 \times 3}$ be the 1st and the 2nd rows of L and $L_3 \in \mathbb{R}^{1 \times 3}$ be the 3rd row of L , one can derive the mapping from joint velocities to image velocities as follows:

$$\dot{y}(t) = \underbrace{\frac{1}{z(t)} (L_0 - yL_3)}_{J(q)} \frac{\partial}{\partial q} (Rr + P) \dot{q}. \quad (10)$$

The nonlinear mapping $J(q)$ introduced in (10) is an important matrix in IBVS, which is known as Jacobian matrix [37, 38]. The differential of (7) w.r.t. time t satisfies

$$\dot{z}(t) = L_3 \frac{\partial}{\partial q} (Rr + P) \dot{q}. \quad (11)$$

The dynamics of robots can be given with Euler-Lagrange equation as follows [39]:

$$H(q(t))\ddot{q}(t) + \left(\frac{1}{2} \dot{H}(q(t)) + C(q(t), \dot{q}(t)) \right) \dot{q} + g(q(t)) = u, \quad (12)$$

where u is the $n \times 1$ vector formed by joint input of the manipulator; $H(q(t))$ is the $n \times n$ positive-definite and symmetric inertia matrix; $C(q(t), \dot{q}(t)) \in \mathbb{R}^{n \times n}$ is a skew-symmetric matrix such that for any proper dimensional vector ψ ,

$$\psi^T C(q(t), \dot{q}(t)) \psi = 0. \quad (13)$$

On the left side of (12), the first term is inertia force, the second term is the Coriolis and centrifugal forces, and the last term $g(q)$ is the gravitational force.

Remark 1. From Figure 2, it intuitively can be seen that the estimation of Jacobian matrices determined by homogeneous transformation matrices (T_b^e , T_b^c , and T_e^c) is directly affected by the delayed visual feedbacks. The complexity of the system mainly lies in the highly nonlinear relationship between delayed image states and joint states.

To facilitate analysis, we present Figure 3 to show the closed-loop structure of a typically delayed VS robotic system.

4. The Adaptive Controller Design

In this section, we will investigate the uncalibrated dynamic-based visual servoing robotic system with visual feedback delays and kinematic uncertainties. In our study, the formulation of the uncalibrated VS robotic system is partly based upon the *depth-independent Jacobian* model developed by

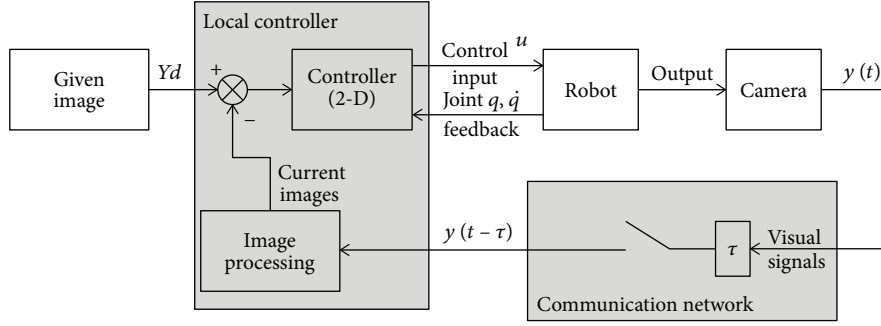


FIGURE 3: The structure of a delayed visual servoing robotic system.

[27]. This model allows depth to be time varying so that the visual servoing system can still be stabilized even in the presence of the fast-changing feature image depth.

From (10), we can easily split $1/z(t)$ from $J(q)$ and thereby obtain the *depth-independent* Jacobian matrix D which is given by

$$D = [L_0 - yL_3] \frac{\partial}{\partial q} (Rr + P). \quad (14)$$

Additionally, from (11), we define such a vector as follows:

$$D_3 = L_3 \frac{\partial}{\partial q} (Rr + P). \quad (15)$$

Therefore (10) and (11) can be rewritten as

$$\begin{aligned} \dot{z}(t) &= D_3 \dot{q}, \\ \dot{y}(t) &= \frac{1}{z(t)} D \dot{q}. \end{aligned} \quad (16)$$

In the uncalibrated dynamic-based visual servoing system, the estimate of Jacobian matrix is usually used as the replacement of unknown exact Jacobian matrix. It can be easily seen from (14) and (15) that components of the *depth-independent* Jacobian matrix D and the matrix D_3 can be classified as two categories: the known and the unknown. Known components are R , P , and unknown components are L , L_3 , and r . The estimate of Jacobian matrix can be analytically derived through the linear parameterization [40]. From (14), it can be seen that the known and the unknown are coupled. And the coupling of the known and the unknown hinders the linear parameterization of these matrices. The following property is proposed to decouple them.

Property 1. For a vector $\dot{q} \in \mathbb{R}^{n \times 1}$, the product $D\dot{q}$ can be linearly parameterized as follows:

$$D\dot{q} = Y_D(y, q, \dot{q})\theta_k, \quad (17)$$

$$D_3\dot{q} = Y_{D_3}(q, \dot{q})\theta_k, \quad (18)$$

where $Y_D(y, q, \dot{q}) \in \mathbb{R}^{2 \times p_1}$ and $Y_{D_3}(q, \dot{q}) \in \mathbb{R}^{1 \times p_1}$ are regressor matrices which consist of known parameters; $\theta_k \in \mathbb{R}^{p_1 \times 1}$ is a vector which consists of unknown parameters; and p_1 denotes the number of unknown parameters, which satisfies $p_1 \leq 36$.

Proof 1. Due to the limitation of pages, see proof in Appendix A.

By Property 1, the Jacobian matrix D can be expressed in a linear form: a known matrix (regressor matrix) multiplies an unknown vector. From (12), it can be clearly seen that the regressor matrix $Y_D(y, q, \dot{q})$ includes the current image position $y(t)$. Unfortunately, the feedback visual signals are delayed as we consider. We may use $y(t-h)$ to denote the coordinates of delayed feature image position, where h denotes the constant delayed time. In this case, the matrix $Y_D(y, q, \dot{q})$ cannot be obtained. Instead, we can only obtain $Y_D(y(t-h), q, \dot{q})$. After substituting this regressor matrix including the delayed visual feedback matrix into (14) and (12), we have

$$Q\dot{q} = Y_D(y(t-h), q, \dot{q})\theta_k = [L_0 - y(t-h)L_3] \frac{\partial}{\partial q} (Rr + P)\dot{q}, \quad (19)$$

where Q is named as delay-affected depth-independent Jacobian matrix. For simplification, we call it delay-affected Jacobian matrix hereafter. The relationship between Q and D is given by

$$Q = D + [y - y(t-h)]D_3. \quad (20)$$

Using the *delay-affected* Jacobian matrix Q and D_3 , we define a new composite Jacobian matrix as

$$J_d = Q + \frac{1}{2}[y(t-h) - y_d]D_3 + [y(t-h) - y_d]\dot{q}^+, \quad (21)$$

where \dot{q}^+ denotes the vector which satisfies $\dot{q}^+ \dot{q} = 1$.

Based on all above analyses, we now propose the controller for delay-affected uncalibrated VS robotic systems as follows:

$$u(t, t-h) = g(q) - \hat{J}_d^T(q)K_1[y(t-h) - y_d] - K_2\dot{q}, \quad (22)$$

where $K_1 \in \mathbb{R}^{2 \times 2}$ and $K_2 \in \mathbb{R}^{n \times n}$ are positive definite symmetric matrices and $\hat{J}_d(q)$ denotes the estimate of $J_d(q)$. Note that the estimate $\hat{J}_d(q)$ for the new Jacobian matrix is able to obtain from (21) by respectively replacing unknown matrices Q and D_3 with their estimates \hat{Q} and \hat{D}_3 , and it yields

$$\hat{J}_d\dot{q} = \hat{Q}\dot{q} + \frac{1}{2}[y(t-h) - y_d]\hat{D}_3\dot{q} + [y(t-h) - y_d]. \quad (23)$$

Additionally, recalling (12) and (18) in Property 1, we can easily derive the following linear parameterization form

$$\hat{J}_d\dot{q} = \underbrace{\left\{ Y_D(y(t-h), q, \dot{q}) + \frac{1}{2}[y(t-h) - y_d]Y_{D_3}(q, \dot{q}) \right\}}_{\bar{Y}(y(t-h), y(t), q, \dot{q})} \hat{\theta}_k + y(t-h) - y_d, \quad (24)$$

where $\bar{Y}(y(t-h), y(t), q, \dot{q})$ is the new regressor matrix including the delayed image state. To obtain $\hat{\theta}_k$, we proposed the following adaptive law:

$$\dot{\hat{\theta}}_k = \Gamma_k^{-1} \bar{Y}^T K_1 [y(t-h) - y_d], \quad (25)$$

where Γ_k is a positive definite symmetric matrix with proper dimensions and \bar{Y} is short for $\bar{Y}(y(t-h), y(t), q, \dot{q})$. Besides, it is not hard to derive \bar{Y}^m and \bar{Y}^M accordingly. Please refer to Notation in Introduction for the explanation. Additionally, it is also not hard to roughly give the bound of the unknown parameter vector θ_k according to the T_e^c and feature Cartesian coordinates [27]. Thereby, we assume that both $\hat{\theta}_k^m$ and $\hat{\theta}_k^M$ are known, i.e., $\hat{\theta}_k \in \mathcal{L}_\infty$. Basing on the above analyses, we can effortlessly know the bound of \hat{J}_d from (24), i.e., \hat{J}_d^m and \hat{J}_d^M can be regarded as known ones. We define

$$\begin{aligned} \hat{J}_{d0} &= \frac{1}{2} (\hat{J}_d^m + \hat{J}_d^M), \\ \Delta \hat{J}_d &= \frac{1}{2} (\hat{J}_d^M - \hat{J}_d^m). \end{aligned} \quad (26)$$

Consequently, \hat{J}_d can be expressed in the interval matrix form [41] as follows:

$$\hat{J}_d = \hat{J}_{d0} + \sum_{i,j=1}^{i=2,j=n} e_i g_{ij} e_j^T, \quad |g_{ij}| < \Delta G_{ij}, \quad (27)$$

where ΔG_{ij} denotes the element at the i th row and the j th column of $\Delta \hat{J}_d$. Likewise, $J(q)$ is also bounded. J^m and J^M are given by

$$\begin{aligned} J_0 &= \frac{1}{2} (J^m + J^M), \\ \Delta J &= \frac{1}{2} (J^M - J^m). \end{aligned} \quad (28)$$

Hence, $J(q)$ can be expressed in the interval matrix form

$$J = J_0 + \sum_{i,j=1}^{i=2,j=n} e_i f_{ij} e_j^T, \quad |f_{ij}| < \Delta J_{ij}, \quad (29)$$

where $e_i \in \mathbb{R}^2$ denotes the column vector whose i th element is 1 and the other element is 0; $e_j \in \mathbb{R}^n$ denotes the row vector whose j th element is 1 and others are 0; ΔJ_{ij} denotes the element at the i th row and j th column of ΔJ .

Remark 2. From (24), one of the key points in deriving \hat{J}_d^m and \hat{J}_d^M is the obtaining of $\hat{\theta}_k^m$ and $\hat{\theta}_k^M$. From the practical experience, the range scales of $\hat{\theta}_k$ actually depend on the (1) the initial value of $\hat{\theta}_k$, which is set artificially and (2) the real value θ_k which is unknown. Even if θ_k is unknown, we can easily give some estimates of its elements according to some other rough estimates. For more details, please refer to Appendix A.

5. Stability Analysis

Theorem 1. Consider the uncalibrated delayed visual servoing system described by (8), (11), (14), and (15) and the controller (22). For a given constant $h > 0$, if there exist symmetric matrices $K_1 > 0$, $K_2 > 0$, positive constants $\varepsilon_{ij} > 0$, $i = 1, 2$, $j = 1, 2, \dots, n$, such that the following nonlinear matrix inequalities hold

$$-K_2 + hJ_0^T (\Gamma_y + \Gamma_y^T \Gamma_y) J_0^T + \frac{h}{2} \Delta J_{ij}^2 [\mathfrak{F}_1 + \mathfrak{F}_2] \quad (30)$$

$$+ \hat{J}_{d0}^T \hat{J}_{d0} + \Delta G_{ij}^2 (\mathfrak{G}_1 + \mathfrak{G}_2) \leq 0,$$

$$-\Gamma_y + \frac{1}{2} \left(\frac{\dot{z}_M}{2} - 1 \right) + K_1^T K_1 \leq 0, \quad (31)$$

$$-K_1 + \frac{K_1^T K_1}{2} \leq 0, \quad (32)$$

where

$$\mathfrak{F}_1 = \sum_{i,j=1}^{i=2,j=n} \varepsilon_{ij} (e_j e_j^T),$$

$$\mathfrak{F}_2 = \sum_{i,j=1}^{i=2,j=n} \frac{1}{\varepsilon_{ij}} \left[e_j e_i^T (\Gamma_y + I)^2 e_i e_j^T \right],$$

$$\mathfrak{G}_1 = \frac{1}{4} \sum_{i,j=1}^{i=2,j=n} \zeta_{ij} e_j e_j^T,$$

$$\mathfrak{G}_2 = \sum_{\substack{i=2, j=n \\ i,j=1}} \frac{1}{\zeta_{ij}} \left[e_j e_i^T e_i e_j^T \right]. \quad (33)$$

Each ε_{ij} and ζ_{ij} denotes any positive constant separately. Then, the system is asymptotically stable, i.e., the image error of the feature point is convergent to zero, $\lim_{t \rightarrow \infty} \Delta y = 0$.

Proof 2. Combining (14), (19), and (21), we have

$$\begin{aligned} \widehat{J}_d \dot{q} &= J_q \dot{q} + (\widehat{J}_d(q) - J_d(q)) \dot{q} \\ &= Q \dot{q} + \frac{1}{2} [y(t-h) - y_d(t)] D_3 \dot{q} + [y(t-h) - y_d(t)] \dot{q}^+ \dot{q} + \bar{Y} \Delta \theta_k \\ &= z \dot{y} + \frac{1}{2} \dot{z} \Delta y + \frac{1}{2} [y(t) - y(t-h)] \dot{z} + [y(t-h) - y_d(t)] + \bar{Y} \Delta \theta_k, \end{aligned} \quad (34)$$

where $\Delta \theta_k = \widehat{\theta}_k - \theta_k$.

Substituting controller (22) into (12), we have the following closed-loop system,

$$\begin{aligned} H(q(t)) \ddot{q}(t) + \left(\frac{1}{2} \dot{H}(q(t)) + C(q(t), \dot{q}(t)) \right) \dot{q} \\ = -\widehat{J}_d^T(q) K_1 [y(t-h) - y_d] - K_2 \dot{q}. \end{aligned} \quad (35)$$

As aforementioned, the fact that \bar{Y} and $\Delta \theta_k$ are all bounded yields the result that

$$\int_0^t \Delta \theta_k^T \bar{Y}^T \bar{Y} \Delta \theta_k dr \leq B_M, \forall t \geq 0 \quad (36)$$

for some positive constants B_M .

Let us consider the following nonnegative Lyapunov-Krasovskii functional candidate,

$$\begin{aligned} V(x) &= \frac{1}{2} \dot{q}^T H(q) \dot{q} + \frac{1}{2} \Delta y^T K_1 z \Delta y \\ &+ \int_{-h}^0 \int_{t+\theta}^t \dot{y}^T(s) \Gamma_y \dot{y}(s) ds d\theta + \frac{1}{2} \Delta \theta_k^T \Gamma_k \Delta \theta_k \\ &+ B_M - \underbrace{\frac{1}{2} \int_0^t \Delta \theta_k^T \bar{Y}^T \bar{Y} \Delta \theta_k dr}_{\nabla}, \end{aligned} \quad (37)$$

where the employment of the term ∇ follows the typical practice (refer to [42], p118).

The time derivative of V along the trajectory of system is given by

$$\begin{aligned} \dot{V}(x) &= \dot{q}^T H(q) \ddot{q} + \dot{q}^T \frac{1}{2} \dot{H}(q) \dot{q} + \Delta y^T K_1 z \Delta \dot{y} \\ &+ \Delta y^T \frac{1}{2} K_1 \dot{z} \Delta y + \Delta \theta_k^T \Gamma_k \Delta \dot{\theta}_k + h \dot{y}^T(t) \Gamma_y \dot{y}(t) \\ &- \int_{t-h}^t \dot{y}^T(s) \Gamma_y \dot{y}(s) ds - \frac{1}{2} \Delta \theta_k^T \bar{Y}^T \bar{Y} \Delta \theta_k. \end{aligned} \quad (38)$$

Multiplying \dot{q}^T from left side to both sides of (35) yields

$$\begin{aligned} \dot{q}^T H(q) \ddot{q} + \dot{q}^T \frac{1}{2} \dot{H}(q) \dot{q} \\ = -\dot{q}^T \widehat{J}_d^T(q) K_1 \Delta y + \dot{q}^T \widehat{J}_d^T(q) K_1 [y(t) - y(t-h)] \\ - \dot{q}^T K_2 \dot{q}. \end{aligned} \quad (39)$$

Rewriting the (34) and then multiplying $\Delta y^T K_1$ from the left side of $z \dot{y} + (1/2) \dot{z} \Delta y$, we have

$$\begin{aligned} \Delta y^T K_1 z \dot{y} + \Delta y^T \frac{1}{2} K_1 \dot{z} \Delta y \\ = \Delta y^T K_1 \widehat{J}_d \dot{q} - \Delta y^T \left(\frac{\dot{z}}{2} - 1 \right) K_1 [y(t) - y(t-h)] \\ - \Delta y^T K_1 \Delta y - \Delta y^T K_1 \bar{Y} \Delta \theta_k. \end{aligned} \quad (40)$$

After taking differential of $(1/2) \Delta \theta_k^T \Gamma_k \Delta \theta_k$ and invoking (25), it yields

$$\Delta \theta_k^T \Gamma_k \Delta \dot{\theta}_k = \Delta \theta_k^T \bar{Y}^T K_1 [\Delta y - y(t) + y(t-h)]. \quad (41)$$

Substituting (39), (40), and (41) into (38), we obtain

$$\begin{aligned} \dot{V}(x) &= \dot{q}^T \widehat{J}_d^T(q) K_1 [y(t) - y(t-h)] - \dot{q}^T K_2 \dot{q} \\ &- \Delta y^T \left(\frac{\dot{z}}{2} - 1 \right) K_1 [y(t) - y(t-h)] - \Delta y^T K_1 \Delta y \\ &- \Delta \theta_k^T \bar{Y}^T K_1 [y(t) - y(t-h)] - \frac{1}{2} \Delta \theta_k^T \bar{Y}^T \bar{Y} \Delta \theta_k \\ &+ h \dot{y}^T(t) \Gamma_y \dot{y}(t) - \int_{t-h}^t \dot{y}^T(s) \Gamma_y \dot{y}(s) ds. \end{aligned} \quad (42)$$

Likewise, with Lemma 1, the below cross terms yield

$$\begin{cases} \dot{q}^T \widehat{J}_d^T(q) K_1 [y(t) - y(t-h)] \leq \frac{1}{2} \dot{q}^T \widehat{J}_d^T \widehat{J}_d \dot{q} + \frac{1}{2} \int_{t-h}^t \dot{y}^T(s) K_1^T K_1 \dot{y}(s) ds, \\ -\Delta y^T \left(\frac{\dot{z}}{2} - 1 \right) K_1 [y(t) - y(t-h)] \leq \frac{1}{2} \Delta y^T K_1 K_1^T \Delta y + \frac{1}{2} \left(\frac{\dot{z}}{2} - 1 \right)^2 \int_{t-h}^t \dot{y}^T(s) K_1^T K_1 \dot{y}(s) ds, \\ -\Delta \theta_k^T \bar{Y}^T K_1 [y(t) - y(t-h)] \leq \frac{1}{2} \Delta \theta_k^T \bar{Y}^T \bar{Y} \Delta \theta_k + \frac{1}{2} \int_{t-h}^t \dot{y}^T(s) K_1^T K_1 \dot{y}(s) ds. \end{cases} \quad (43)$$

Besides, from (10), $h\dot{y}^T(t)\Gamma_y\dot{y}(t)$ can be rewritten as $h\dot{q}^T J^T(q)\Gamma_y J(q)\dot{q}$.

Having obtained the results in (43), substituting them to (43) and we have the following inequality:

$$\begin{aligned} \dot{V}(x) \leq & \underbrace{\dot{q} \left(-K_2 + \frac{1}{2} \hat{J}_d^T \hat{J}_d + hJ^T \Gamma_y J \right)}_I \dot{q} \\ & + \underbrace{\left(-\Gamma_y + \frac{1}{2} \left(\frac{\dot{z}}{2} - 1 \right)^2 + K_1^T K_1 \right)}_{II} \int_{t-h}^t \dot{y}^T(s) K_1^T K_1 \dot{y}(s) ds + \Delta y^T \left(-K_1 + \frac{K_1^T K_1}{2} \right) \Delta y. \end{aligned} \quad (44)$$

We will analyze the term I and the term II one by one. Firstly, we consider the term I . In this term, both \hat{J}_d and J are time-varying matrices. It should be noted that we assume J^m and J^M being unknown ones as aforementioned. Using Lemma 1 and (29), we can easily derive

$$\begin{aligned} hJ^T \Gamma_y J \leq & hJ_0^T \left(\Gamma_y + \Gamma_y^T \Gamma_y \right) J_0^T + \sum_{i,j=1}^{i=2,j=n} \frac{h\varepsilon_{ij}}{2} \Delta J_{ij}^2 e_j e_j^T \\ & + \sum_{i,j=1}^{i=2,j=n} \frac{h}{2\varepsilon_{ij}} \Delta J_{ij}^2 \left[e_j e_i^T \left(\Gamma_y + I \right)^2 e_i e_j^T \right], \end{aligned} \quad (45)$$

where ε_{ij} denotes any positive constant.

With Lemma 1 and (27), we can effortlessly extend (1/2) $\hat{J}_d^T \hat{J}_d$ as

$$\begin{aligned} \frac{1}{2} \hat{J}_d^T \hat{J}_d \leq & \hat{J}_{d0}^T \hat{J}_{d0} + \sum_{i,j=1}^{i=2,j=n} \frac{\zeta_{ij}}{4} \Delta G_{ij}^2 e_j e_j^T, \\ & + \sum_{i,j=1}^{i=2,j=n} \frac{1}{\zeta_{ij}} \Delta G_{ij}^2 \left[e_j e_i^T e_i e_j^T \right], \end{aligned} \quad (46)$$

where ζ_{ij} denotes any positive constant.

Substituting (45) and (46) into term I , invoking (30), it yields

$$\begin{aligned} & -K_2 + \frac{1}{2} \hat{J}_d^T \hat{J}_d + hJ^T \Gamma_y J \\ & \leq -K_2 + hJ_0^T \left(\Gamma_y + \Gamma_y^T \Gamma_y \right) J_0^T \\ & \quad + \frac{h}{2} \Delta J_{ij}^2 [\mathfrak{F}_1 + \mathfrak{F}_2] + \hat{J}_{d0}^T \hat{J}_{d0} \\ & \quad + \Delta G_{ij}^2 (\mathfrak{G}_1 + \mathfrak{G}_2) \leq 0, \end{aligned} \quad (47)$$

where \mathfrak{F} and \mathfrak{G} are defined in (33).

Then, we consider the term II . In actual visual servoing robotic system, the depth changing velocity is actually

bounded. Here, we can reasonably assume that the \dot{z} is bounded, $\dot{z} < \dot{z}_M$. Invoking (31), we have

$$-\Gamma_y + \frac{1}{2} \left(\frac{\dot{z}}{2} - 1 \right)^2 + K_1^T K_1 \leq -\Gamma_y + \frac{1}{2} \left(\frac{\dot{z}_M}{2} - 1 \right)^2 + K_1^T K_1 \leq 0. \quad (48)$$

Combining (47), (48), and (32), we can finally have $\dot{V} \leq 0$ in (44), which means that the Lyapunov-Krasovskii functional V never increases its value so that it is upper bounded. From (37), bounded V directly implies that the joint velocity $\dot{q} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, $[y - y(t-h)] \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, the errors of $\theta_k \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, and image error $\Delta y \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. Then the joint acceleration $\ddot{q} \in \mathcal{L}_\infty$ can be concluded from the closed-loop dynamics (35). Therefore, the joint velocity \dot{q} is uniformly continuous. Note that it is not hard to derive $\dot{y} \in \mathcal{L}_\infty$ from (10) with bounded $J^T(q)$ and $\dot{q} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. Thereby, we can also conclude that Δy is uniformly continuous. $\dot{y} \in \mathcal{L}_\infty$ and $\dot{y}(t-h) \in \mathcal{L}_\infty$ yield $[\dot{y} - \dot{y}(t-h)] \in \mathcal{L}_\infty$ and hence we can derive that the image delay error $y - y(t-h)$ is uniformly continuous. And from (25), it can be derived that $\dot{\theta}_k \in \mathcal{L}_\infty$. Thereby, $\dot{\theta}_k$ is uniformly continuous. Invoking Lemma 2 and Lemma 3, we have $\lim_{t \rightarrow \infty} \Delta y = 0$, $\lim_{t \rightarrow \infty} \Delta \theta_k = 0$, and $\lim_{t \rightarrow \infty} [y - y(t-h)] = 0$. This completes the proof.

Remark 3. It can be clearly seen that the delay-dependent stability condition is presented in Theorem 1. Stability analyses given by [33–35] are delay-independent results, which means the stability conditions impose no constraint on system delays. Hence, their stability results hold with any magnitude of delays. However, in reality, the delays are usually bounded and the delay-dependent results are conservative. To obtain less conservative results, we should consider magnitude of delays. It is significant to the delay stability research due to less conservativeness.

Remark 4. In order to fully control 6-DOFs or more degree robots, we need more noncollinear feature points. For instance, three noncollinear feature points should be considered for a 6-DOF manipulator. The scheme proposed in this paper can be effortlessly generalized to the case of multiple feature points by the similar method described in [28]. Considering the page limitation, we only present the single feature point case in this paper.

6. Simulation Results

To show the effectiveness of the control scheme described in (22) and Theorem 1, we conduct the following simulations.

The actual visual parameters are set as follows: $f = 0.035$ m, $u_0 = 280$ pixels, $v_0 = 250$ pixels, $k_u = 1800$ pixels/m, $k_v = 1800$ pixels/m, and $\vartheta = \pi/2$ rad, where f is focal length; u_0 and v_0 are coordinates of camera principal point in the image frame; k_u and k_v denote scale factors along axis x and y , respectively; and ϑ denotes intersection angle between axis u and axis v . The intrinsic matrix Ω therefore can be derived as

$$\Omega = \begin{pmatrix} 63 & 0 & 280 & 0 \\ 0 & 63 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (49)$$

For the setting of the camera's position and pose, the T_e^c is set as follows:

$$T_e^c = \begin{pmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (50)$$

The gravitational acceleration is set as $g = 10 \text{ m/s}^2$. T_b^e is time varying and determined by forward kinematics of the manipulator whose parameters are given in Table 1.

From Property 1 and according to the ranges of q_i , z , L_0 , L_3 , $(\partial/\partial q)(Rr + P)$, in this paper, we may set J_d^M and J_d^m as

$$\begin{aligned} \hat{J}_d^m &= \begin{pmatrix} -602 & -120 & -85 \\ -150 & -120 & -30 \end{pmatrix}, \\ \hat{J}_d^M &= \begin{pmatrix} 602 & 120 & 85 \\ 150 & 120 & 30 \end{pmatrix}, \end{aligned} \quad (51)$$

and set \bar{Y}^M and \bar{Y}^m as

$$\begin{aligned} \bar{Y}^m &= \begin{pmatrix} -\Upsilon & 0_{1 \times 12} & -280 \times \Upsilon \\ -\Upsilon & 0_{1 \times 12} & -250 \times \Upsilon \end{pmatrix}, \\ \bar{Y}^M &= \begin{pmatrix} \Upsilon & 0_{1 \times 12} & 280 \times \Upsilon \\ \Upsilon & 0_{1 \times 12} & 250 \times \Upsilon \end{pmatrix}, \end{aligned} \quad (52)$$

where

$$\Upsilon = \begin{bmatrix} \underbrace{500 \dots 500}_{9} \underbrace{200 \dots 200}_{3} \end{bmatrix}. \quad (53)$$

The feature point's coordinates w.r.t. the base frame are $(150, 20)^T \text{m}$. The initial position coordinates on the image plane are $(140, 81.44)^T$, and the desired position coordinates on image plane are $(160.7, 120.6)^T$.

Besides, we set $\dot{z}_M = 0.3 \text{ m/s}$ here and K_1 and K_2 are obtained by solving the feasibility problem of (30), (31), and (32) with the solver *feasp*. In this simulation, we use $K_1 = \text{diag}\{3.250, 1.025\}$, $K_2 = \text{diag}\{50.689, 125.336\}$, $\Psi_k = 50I_{18}$, $\Gamma_k = 8I_{36}$.

Based on all above settings, two simulations are conducted. In the first simulation, the proposed control scheme is used to track the desired position under two different constant delays: $h_1 = 98 \text{ ms}$ and $h_2 = 198 \text{ ms}$. Figures 4(a), 5(a), 6(a), and 7(a) demonstrate the position errors, the position, the velocity, and the trajectory of the feature point on the image plane, respectively. It can be observed that the

TABLE 1: Parameters of manipulator in the simulation.

Link	l_i	α_i	d_i	q_i	$m_i(\text{kg})$	l_{ci}
1	1	0	0	$q1$	1	0.5
2	1	180	0	$q2$	1	0.5
3	0	0	1	$q3$	1	0.5

Notes: l_i denotes link length; α_i denotes link twist; d_i denotes link offset; q_i denotes joint angle; m_i denotes link mass; l_{ci} denotes the length between barycenter and its prior joint.

performance is almost identical even under the different delays, 198 ms and 98 ms. It verifies that the convergence of the system will be achieved once as long as the conditions given in Theorem 1 hold. Besides, Figure 7 also shows better position tracking performance with 98 ms delays than that of 198 ms. To show the convergence of estimates $\hat{\theta}_k$ to real values, we partly choose some elements in the vector $\hat{\theta}_k$. Figure 8 shows the profile of estimated parameters from $\hat{\theta}_{k4}$ to $\hat{\theta}_{k12}$. It should be noted that the kinematic parameters $\hat{\theta}_k$ converge only when the persistent excitation (P.E.) condition is satisfied. In our simulation, we choose the $\hat{\theta}_k(0)$ close to their real values such that these estimated parameters can converge to them. In most cases, the estimated parameters only converge to the true values up to a scale. However, it will not affect the convergence of image errors.

To demonstrate the superiority of the proposed control scheme, we make a comparison between the two control schemes: the scheme 1 and the scheme 2. The scheme 1 is the method proposed in this paper, and the scheme 2 originating from [30] is modified accordingly in this simulation as follows.

$$\begin{cases} \hat{J} = \hat{D} + \frac{1}{2} \Delta y \hat{D}_3, \\ u(t) = g(q) - \hat{J} K_1 \Delta y(t) - K_2 \dot{q}. \end{cases} \quad (54)$$

It should be noted that the Jacobian matrix in the scheme 2 does not consider the delay effects. Then, we conduct the second simulation. In this simulation, we use $\dot{z}_M = 0.05 \text{ m/s}$, $K_1 = \text{diag}\{50.250, 50.025\}$, $K_2 = \text{diag}\{60.168, 125.336\}$, $\Psi_k = 100I_{18}$, $\Gamma_k = 0.2I_{36}$, $h = 98 \text{ ms}$ for scheme 2.

From Figures 4(b)–7(b), it can be clearly seen that the performance of scheme 2 in the presence of delay time $h = 98 \text{ ms}$ is unsatisfying. Abnormal oscillations can be observed, which is caused by the delays. In contrast, the proposed scheme can still guarantee very satisfying control performance, which is not affected too much by delayed signals. In conclusion, the second simulation result shows the superiority of the proposed scheme over existing schemes that can eliminate the negative effect caused by delays to a great extent.

7. Conclusions

In this paper, we have proposed a control method for uncalibrated dynamic-based visual servoing robotic systems to

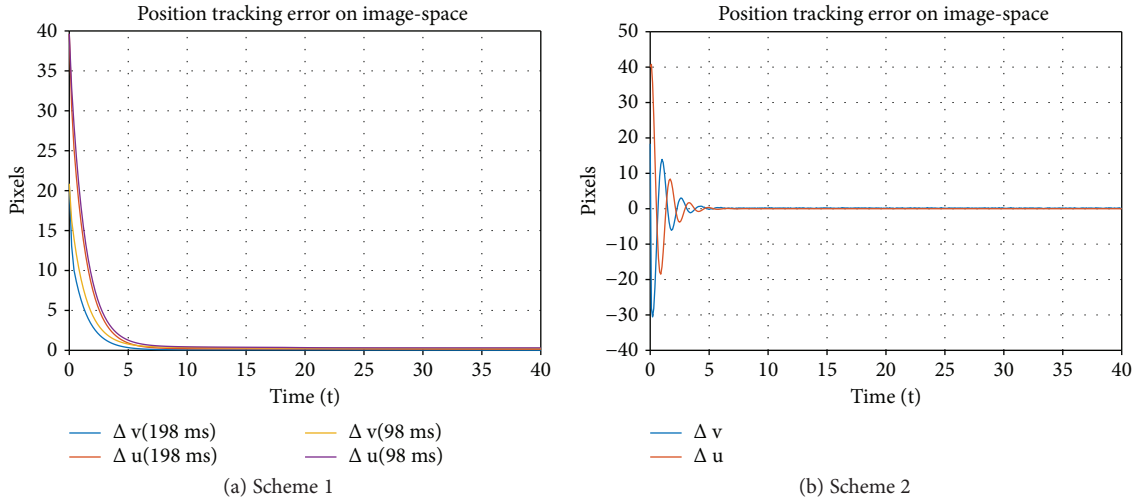


FIGURE 4: Position tracking errors.

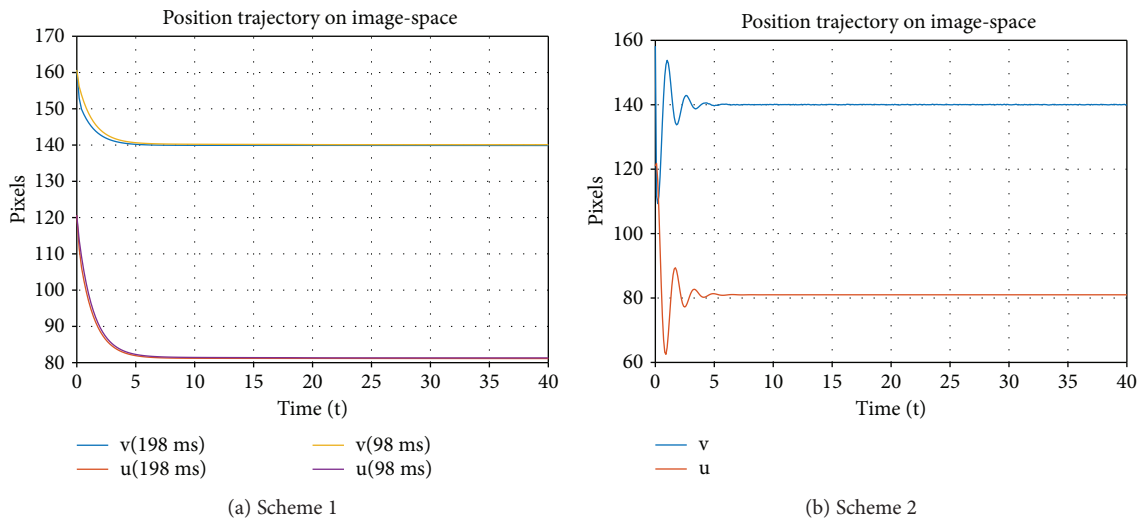


FIGURE 5: Position trajectory of the feature point on the image plane.

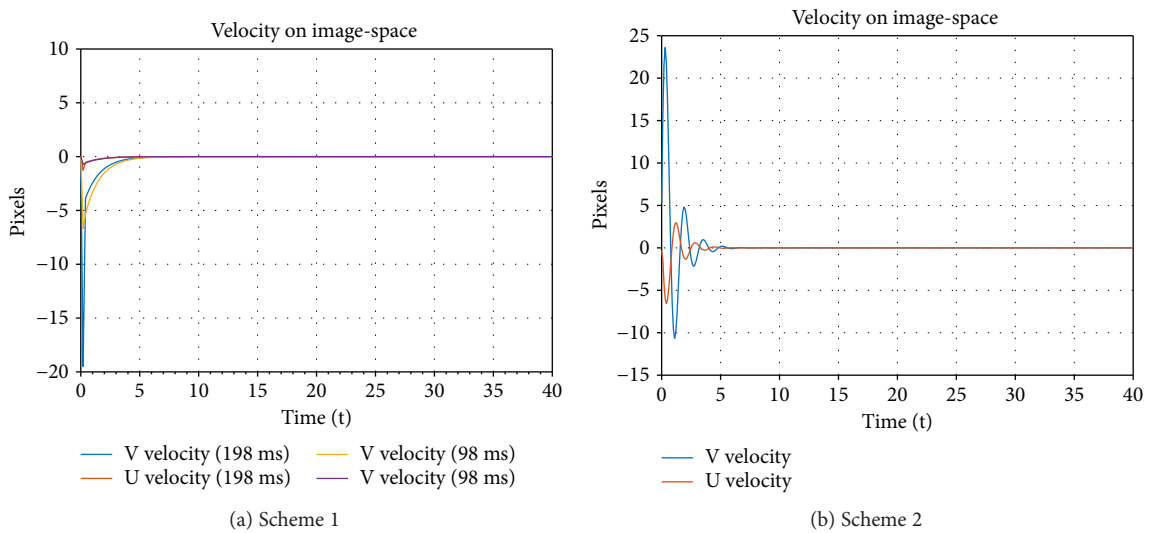


FIGURE 6: Velocity of the feature point on the image plane.

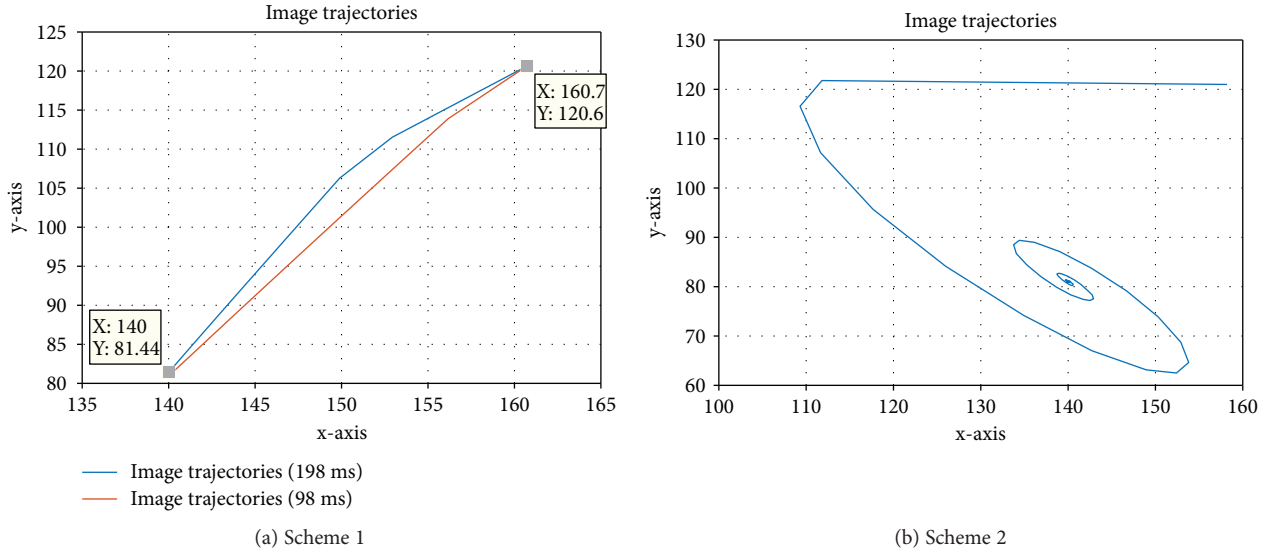
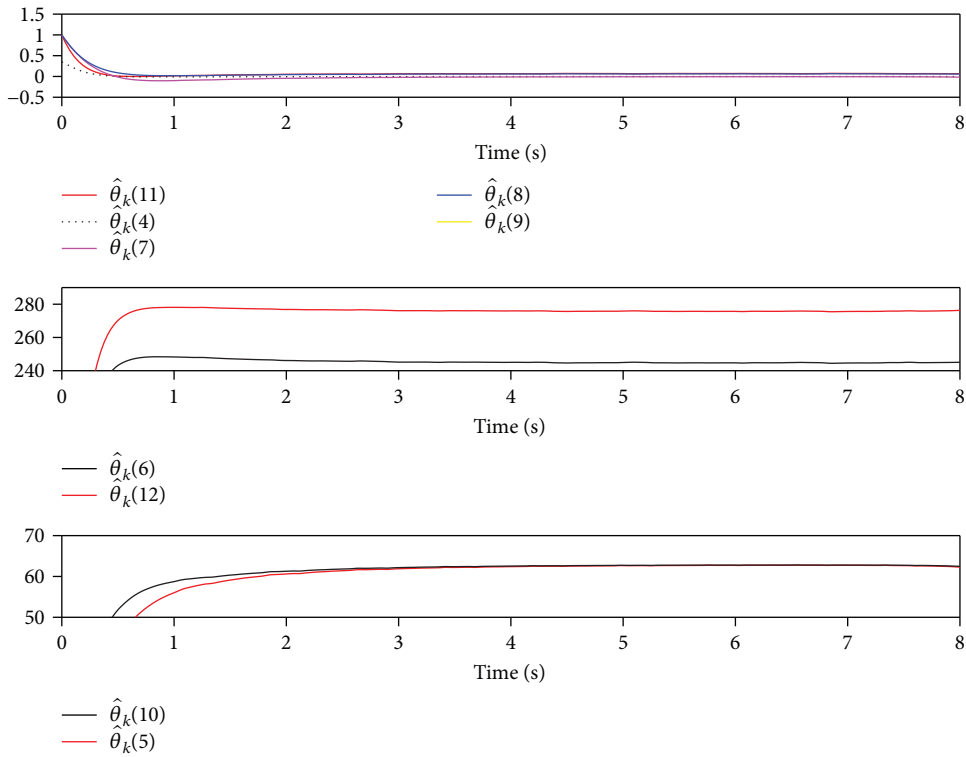


FIGURE 7: Position trajectory of the feature point on the image plane.

FIGURE 8: Estimated parameters from $\hat{\theta}_{k4}$ to $\hat{\theta}_{k12}$.

cope with the delay problem existing in the visual feedback loops. To handle the unknown camera intrinsic and extrinsic parameters, we introduced the *depth-independent* Jacobian matrix and used the linear parameterization to adaptively identify these uncertainties. Then, we took the delays into consideration and constructed a novel matrix called *delay-affected* Jacobian matrix. Based on the *delay-affected* Jacobian matrix, we proposed the adaptive controller. To prove the stability of the closed-loop system, the

Lyapunov-Krasovskii functional is constructed and delay-dependent stability conditions are also provided to obtain less conservative results. Simulation results of the proposed control scheme were presented to show the effectiveness. To further validate the performance of the proposed scheme, experimental tests on real networked visual servoing robotics systems are expected to be the most appropriate choice and this is also one of our main objectives in the future research.

Appendix

A. Proof of Property 1

Proof 3. Let L_i denote the i th row of L . Recalling (14), we can expand $L_i^T(\partial/\partial q)(Rx + P)$ as follows:

$$\begin{aligned} L_i^T \frac{\partial}{\partial q} (Rx + P) &= \frac{\partial}{\partial q} (L_i^T Rx + L_i^T P) \\ &= \frac{\partial}{\partial q} (r_{11}L_{i1}x_1 + r_{21}L_{i2}x_1 + r_{31}L_{i3}x_1 \\ &\quad + r_{12}L_{i1}x_2 + r_{22}L_{i2}x_2 + r_{32}L_{i3}x_2 \\ &\quad + r_{13}L_{i1}x_3 + r_{23}L_{i2}x_3 + r_{33}L_{i3}x_3 \\ &\quad + \rho_1L_{i1} + \rho_2L_{i2} + \rho_3L_{i3}), \end{aligned} \quad (\text{A.1})$$

where $r_{hl}(h = 1, 2, 3, l = 1, 2, 3)$ denotes the (h, l) element of matrix R , $\rho_k(k = 1, 2, 3)$ denotes the k th elements of P , $L_{ij}(i = 1, 2, 3, j = 1, 2, 3)$ denotes the (i, j) element of matrix L , and x_p denotes p th element of x . Let q_i and \dot{q}_i be the i th element of q and \dot{q} , respectively. When $p_1 = 36$, i.e., none of the elements r_{hl} and ρ_k equals to zero, vector $D\dot{q}$ will linearly depend on 36 unknown parameters, and $Y_D(y, q, \dot{q})$ can be derived. Define $\varrho_{hl} = \sum_{i=1}^n \dot{q}_i \partial r_{hl} / \partial q_i (h = 1, 2, 3, l = 1, 2, 3)$, $\varrho_k = \sum_{i=1}^n \dot{q}_i \partial \rho_k / \partial q_i (k = 1, 2, 3)$, $\varrho^T = (\varrho_{11}, \varrho_{12}, \varrho_{13}, \varrho_{21}, \varrho_{22}, \varrho_{23}, \varrho_{31}, \varrho_{32}, \varrho_{33}, \varrho_1, \varrho_2, \varrho_3)$, and $\theta_k^T = (L_{i1}x_1, L_{i1}x_2, L_{i1}x_3, L_{i2}x_1, L_{i2}x_2, L_{i2}x_3, L_{i3}x_1, L_{i3}x_2, L_{i3}x_3, L_{i1}, L_{i2}, L_{i3})$, $i = 1, 2, 3$. Specifically, we have $\theta_k = (\theta_{k1}^T, \theta_{k2}^T, \theta_{k3}^T)$ and

$$Y_D(y, q, \dot{q}) = \begin{pmatrix} \varrho^T & 0_{1 \times 12} & -u\varrho^T \\ 0_{1 \times 12} & \varrho^T & -v\varrho^T \end{pmatrix}. \quad (\text{A.2})$$

When r_{hl} is independent of q , ϱ_{hl} equals zero. $Y_D(y, q, \dot{q})$ can be obtained by removing elements which equal zero, accordingly, θ_k can be obtained by removing corresponding elements. When ρ_k is independent of q , $Y_D(y, q, \dot{q})$ and θ_k can be obtained by similarly removing. In this way, we can derive the expression of θ_k and $Y_D(y, q, \dot{q})$ for every $p_1 < 36$.

Besides, because d^T is a subset of D , the linearization of $d^T \dot{q}$ can be a direct result of the Property. When $p_1 = 36$, $Y_{D_3}(q, \dot{q})$ can be expressed as

$$Y_{D_3}(q, \dot{q}) = (0_{1 \times 24} \quad \varrho^T). \quad (\text{A.3})$$

When $p_1 < 36$, i.e., ρ_k is independent of q , $Y_{D_3}(q, \dot{q})$ and θ_k can be obtained by removing corresponding elements. Then, we can derive expression of θ_k and $Y_{D_3}(q, \dot{q})$ for every $p_1 < 36$. This completes the proof.

B. List of Notations & Symbols

TABLE 2: List of notations and symbols.

Notations	
Ω	Perspective projection matrix
T_b^e	Homogeneous transformation matrix from the end effector to the base
R	Rotation matrix included in T_b^e
T	Translation vector included in T_b^e
T_e^c	Homogeneous transformation matrix from the camera to the end effector
r	The Cartesian coordinates of the feature point w.r.t. the robot base frame
L	The matrix consists of the left 3×3 part of ΩT_e^c
L_0	The 1st and 2nd rows of L
L_3	The 3rd row of L
$J(q)$	Jacobian matrix
$H(q)$	The inertia matrix of manipulator dynamics
$C(q, \dot{q})$	The Coriolis and centrifugal forces
$g(q)$	The gravitational force
u	The control input
D	The depth-independent Jacobian matrix
D_3	The vector derived from $L_3 \partial / \partial q (Rr + P)$
Q	The delay-affected depth-independent Jacobian matrix
J_d	A novel composite Jacobian matrix
K_1, K_2	The control gain matrices
$Y_D(*), Y_{D_3}(*), *$	The regressor matrices
θ_k	The unknown parameter vector
e_i	$e_i \in \mathbb{R}^2$, the column vector whose i th element is 1 and the other element is 0
e_j	$e_j \in \mathbb{R}^n$, the row vector whose j th element is 1 and other elements are 0
Symbols	
$\ \cdot \ $	The standard Euclidean norm
A^T	The transposition of matrix A
A^m	The matrix consists of the minimum elements of A
A^M	The matrix consists of the maximum elements of A
A_{ij}	The element in the i th row and the j th column of matrix A
$y(t-h)$	The visual signal delayed unknown constant duration of h

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financed by Science and Technology Program of Tianjin, China under Grant 15ZXZNGX00290.

References

- [1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [2] M. Vincze, "Dynamics and system performance of visual servoing," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, pp. 644–649, San Francisco, CA, USA, 2000.
- [3] S. Benhimane and E. Malis, "A new approach to vision-based robot control with omni-directional cameras," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, pp. 526–531, Orlando, FL, USA, 2006.
- [4] R. Dahmouche, N. Andreff, Y. Mezouar, O. Ait-Aider, and P. Martinet, "Dynamic visual servoing from sequential regions of interest acquisition," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 520–537, 2012.
- [5] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 404–417, 1987.
- [6] M. Jägersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proceedings of International Conference on Robotics and Automation*, pp. 2874–2880, Albuquerque, NM, USA, 1997.
- [7] H. Hashimoto, T. Kubota, M. Sato, and F. Harashima, "Visual control of robotic manipulator based on neural networks," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 6, pp. 490–496, 1992.
- [8] E. Zergeroglu, D. M. Dawson, M. S. de Querioz, and A. Behal, "Vision-based nonlinear tracking controllers with uncertain robot-camera parameters," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 322–337, 2001.
- [9] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 41–50, 2017.
- [10] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2017.
- [11] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, "Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 5, pp. 814–825, 2005.
- [12] A. C. Leite and F. Lizarralde, "Passivity-based adaptive 3d visual servoing without depth and image velocity measurements for uncertain robot manipulators," *International Journal of Adaptive Control and Signal Processing*, vol. 30, no. 8–10, pp. 1269–1297, 2016.
- [13] P. I. Corke, *High-performance visual closed-loop robot control*, [Ph.D. thesis], Mechanical and Manufacturing Engineering, 1994.
- [14] M. Vincze, M. Ayromlou, S. Chroust, M. Zillich, W. Ponweiser, and D. Legenstein, "Dynamic aspects of visual servoing and a framework for real-time 3D vision for robotics," in *Sensor Based Intelligent Robots*, pp. 101–121, Springer, Berlin, Heidelberg, 2002.
- [15] J. A. Gangloff and M. F. de Mathelin, "High speed visual servoing of a 6 DOF manipulator using MIMO predictive control," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, pp. 3751–3756, San Francisco, CA, USA, 2000.
- [16] J. A. Gangloff and M. F. de Mathelin, "High-speed visual servoing of a 6-d.o.f. manipulator using multivariable predictive control," *Advanced Robotics*, vol. 17, no. 10, pp. 993–1021, 2003.
- [17] L. Cuvillon, E. Laroche, J. Gangloff, and M. de Mathelin, "GPC versus H ∞ control for fast visual servoing of a medical manipulator including flexibilities," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4044–4049, Barcelona, Spain, 2006.
- [18] H. Wu, L. Lou, C. C. Chen, S. Hirche, and K. Kuhnlenz, "Cloud-based networked visual servo control," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 554–566, 2013.
- [19] M. Nakadokoro, S. Komada, and T. Hori, "Stereo visual servo of robot manipulators by estimated image features without 3d reconstruction," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, pp. 571–576, Tokyo, Japan, 1999.
- [20] N. Dai, M. Nakamura, S. Komada, and J. Hirai, "Tracking of moving object by manipulator using estimated image feature and its error correction on image planes," in *The 8th IEEE International Workshop on Advanced Motion Control, 2004. AMC '04*, pp. 653–657, Kawasaki, Japan, 2004.
- [21] I. Kinbara, S. Komada, and J. Hirai, "Visual servo of active cameras and manipulators by time delay compensation of image features with simple on-line calibration," in *2006 SICE-ICASE International Joint Conference*, pp. 5317–5322, Busan, Republic of Korea, 2007.
- [22] T. Inoue and S. Hirai, "Robotic manipulation with large time delay on visual feedback systems," in *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1111–1115, Montreal, ON, Canada, 2010.
- [23] Z. Gao and J. Su, "Estimation of image Jacobian matrix with time-delay compensation for uncalibrated visual servoing," *Control Theory and Applications*, vol. 26, no. 1, pp. 218–234, 2009.
- [24] Y. H. Liu, H. Wang, and K. Lam, "Dynamic visual servoing of robots in uncalibrated environments," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3131–3136, Edmonton, Canada, 2006.
- [25] H. Wang and Y. H. Liu, "Uncalibrated visual tracking control without visual velocity," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, pp. 2738–2743, Orlando, FL, USA, 2006.
- [26] Y. Shen, D. Sun, Y.-H. Liu, and K. Li, "Asymptotic trajectory tracking of manipulators using uncalibrated visual feedback,"

- IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 1, pp. 87–98, 2003.
- [27] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, “Uncalibrated visual servoing of robots using a depth-independent interaction matrix,” *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 804–817, 2006.
- [28] H. Wang, Y.-H. Liu, and D. Zhou, “Dynamic visual tracking for manipulators using an uncalibrated fixed camera,” *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 610–617, 2007.
- [29] F. Lizarralde, A. C. Leite, L. Hsu, and R. R. Costa, “Adaptive visual servoing scheme free of image velocity measurement for uncertain robot manipulators,” *Automatica*, vol. 49, no. 5, pp. 1304–1309, 2013.
- [30] X. Liang, H. Wang, Y.-H. Liu, W. Chen, and J. Zhao, “A unified design method for adaptive visual tracking control of robots with eye-in-hand/fixed camera configuration,” *Automatica*, vol. 59, pp. 97–105, 2015.
- [31] T. Li and H. Zhao, “Global finite-time adaptive control for uncalibrated robot manipulator based on visual servoing,” *ISA Transactions*, vol. 68, pp. 402–411, 2017.
- [32] W. Qiao and R. Sipahi, “Consensus control under communication delay in a three-robot system: design and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 687–694, 2016.
- [33] Y.-C. Liu and N. Chopra, “Controlled synchronization of heterogeneous robotic manipulators in the task space,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 268–275, 2012.
- [34] H. Wang, “Passivity based synchronization for networked robotic systems with uncertain kinematics and dynamics,” *Automatica*, vol. 49, no. 3, pp. 755–761, 2013.
- [35] X. Liang, H. Wang, Y. H. Liu, W. Chen, G. Hu, and J. Zhao, “Adaptive task-space cooperative tracking control of networked robotic manipulators without task-space velocity measurements,” *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2386–2398, 2016.
- [36] K. Hashimoto, K. Nagahama, and T. Noritsugu, “A mode switching estimator for visual servoing,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pp. 1610–1615, Washington, DC, USA, 2002.
- [37] J. M. Sebastián, L. Pari, L. Angel, and A. Traslosheros, “Uncalibrated visual servoing using the fundamental matrix,” *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 1–10, 2009.
- [38] A. Shademan, A.-m. Farahmand, and M. Jägersand, “Robust jacobian estimation for uncalibrated visual servoing,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 5564–5569, Anchorage, AK, USA, 2010.
- [39] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3rd edition, 2002.
- [40] J.-J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.
- [41] F. Garofalo, G. Celentano, and L. Glielmo, “Stability robustness of interval matrices via Lyapunov quadratic forms,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 281–284, 1993.
- [42] R. Lozano, B. Brogliato, and O. E. an Maschke, “Dissipative systems analysis and control. Theory and applications,” *Measurement Science and Technology*, vol. 12, no. 12, p. 2211, 2001.

Research Article

Distributed Control of a Manufacturing System with One-Dimensional Cellular Automata

Irving Barragan-Vite , Juan C. Seck-Tuoh-Mora, Norberto Hernandez-Romero ,
Joselito Medina-Marin , and Eva S. Hernandez-Gress

Engineering Department, Autonomous University of Hidalgo State, Carr. Pachuca-Tulancingo, Col. Carboneras, Mineral de la Reforma, Hidalgo 42184, Mexico

Correspondence should be addressed to Irving Barragan-Vite; irving.vite@gmail.com

Received 27 April 2018; Revised 12 July 2018; Accepted 29 July 2018; Published 4 October 2018

Academic Editor: Zhiwei Gao

Copyright © 2018 Irving Barragan-Vite 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.

We present a distributed control modeling approach for an automated manufacturing system based on the dynamics of one-dimensional cellular automata. This is inspired by the fact that both cellular automata and manufacturing systems are discrete dynamical systems where local interactions given among their elements (resources) can lead to complex dynamics, despite the simple rules governing such interactions. The cellular automaton model developed in this study focuses on two states of the resources of a manufacturing system, namely, busy or idle. However, the interaction among the resources such as whether they are shared at different stages of the manufacturing process determines the global dynamics of the system. A procedure is shown to obtain the local evolution rule of the automaton based on the relationships among the resources and the material flow through the manufacturing process. The resulting distributed control of the manufacturing system appears to be heterarchical, and the evolution of the cellular automaton exhibits a Class II behavior for some given disordered initial conditions.

1. Introduction

The demand for top-quality products, as well as a variety of them, has led most manufacturing industries to automate their production processes [1]. By doing this, enterprises achieve an increase in their productivity as well as flexibility to rapidly respond to customers' demand. However, many issues arise at modeling automated manufacturing systems due to the rapid changes in global markets and the difficulty to foresee events such as machine failures and backlogging among other problems during a manufacturing system operation. Thus, decisions are required as soon as an undesired event occurs, and this involves having a control capable to react and prevent the manufacturing system to deviate from the optimal operating conditions.

Controlling a manufacturing system means to manage the activities of the system with the aim at operating according to the production plans and schedules, monitoring at

every moment the flow of the parts being processed [2]. In a manufacturing scheme, the distributed control consists in dividing the control task into several entities capable to make decisions [3]. This allows making decisions rapidly and offers robustness against unforeseen failures as well as flexibility, scalability, modularity, and reconfigurability which are features of today manufacturing systems and the manufacturing paradigms [4].

Although not new, agent-based and holonic manufacturing models meet most or all the features described above and are suitable for distributed control as it is shown in [5–7]. However, they are not easy to implement, and thus, other approaches have been developed. In [8], a distributed control method is proposed based on fuzzy logic and the decomposition of the system into several subsystems. The fuzzy logic is also applied in [9] to control the flow in a manufacturing system together with a multicriteria decision-making method. In [10], a supervisory control is developed in order to decide

when to increase the production capacity based on local controllers and the cumulative demand.

Among the tools for modeling automated manufacturing systems, Petri nets are one of the most used, due to their graphical representation and mathematical formalism [11–13]. However, Petri nets are difficult to manage when there is a large number of components in the system leading to a large number of possible states in the net. Computer simulation models have been also one of the most used tools to model automated manufacturing systems [14] as a way to overcome the difficulties resulted from analytical models [15]. A disadvantage of computer-based models is that they represent the logic of the systems and measure their performance for specific scenarios by simulation. Most of the graphical, analytical and simulation models rely on assumptions that are unrealistic, and hence, there is a need of modeling methods easy to analyze and that match with the real operation of the increasingly complex manufacturing systems.

This paper proposes a procedure to model an automated manufacturing system with cellular automata, inspired by the local interactions among the cells and the global dynamics they produce. Cellular automata are dynamical systems composed of identical elements called cells. Every cell can take a state over a finite set of states, and each cell interacts locally with its nearest neighboring cells to change its state. Once a local evolution rule has been determined, all the cells change their state synchronously over discrete time steps. When the cells are arranged in a linear way, the cellular automaton is one-dimensional.

One of the major contributions to the one-dimensional cellular automata study has been their dynamic classification [16]. No matter which initial state is chosen for a cellular automaton, their evolution exhibits properties of one of four classes. On account of the dynamical features of cellular automata, they have been used to model systems in biology [17, 18], ecology [19, 20], physics [21], and chemistry [22]. Urban systems, as well as land use, have been also application fields of cellular automata as it is shown in [23–26].

Cellular automata were proposed in [27, 28] to model automated manufacturing systems considering the resource activities; however, the dynamics of the proposed cellular automaton model are performed asynchronously. Assembly simulation and assembly sequence identification were addressed in [29] with cellular automata. In [30], cellular automaton models are proposed for flexible manufacturing systems. These models are obtained by means of Petri net representations which are reduced in order to obtain subnets which form cells; however, the reduction of the Petri nets seems to be difficult to perform for large models.

In the cellular automaton model proposed in this manuscript, the cells represent the resources of the system and they are arranged according to the sequence of operations that a part follows throughout the system. The contribution of this study relies on the construction of the local evolution rule, which considers the constraints of the flow of parts due to resource sharing and capacity, and how this rule is used to resolve and prevent conflicts in the manufacturing system based on deterministic decisions,

yielding a way to control the flow. The evolution of the cellular automaton shows a Class II behavior, and the control of the manufacturing system seems to be heterarchical; that is, it does not depend on centralized decisions.

The rest of the paper is structured as follows: Section 2 details the characteristics of automated manufacturing systems as well as that of cellular automata. Section 3 presents some preliminaries and considerations about the proposed modeling method. The modeling method is explained in Section 4. Section 5 shows the dynamics of the automated manufacturing system example by the cellular automaton evolutions. Section 6 presents a number of cases where the performance of the automated manufacturing system example under the control of the cellular automaton evolution rule is compared with the simulation of the system with a Petri net model. Finally, a discussion on the method is given in Section 7 and conclusions are provided in Section 8.

2. Basic Concepts

2.1. Automated Manufacturing Systems. Automated manufacturing systems (AMS) are discrete dynamical systems where events occur asynchronously [31, 32] such as the starting and the ending of an operation or the arrival or departure of parts [33]. Typically, an AMS is composed of numerically controlled machines which are connected by automated material handling devices like robots and automated guided vehicles. All the machines and material handling devices are controlled by a centralized computer system [34, 35]. The elements of the AMS interact each other to produce several parts which follow a determined sequence of operations carried out by machines, operators, or both of them.

An automated manufacturing system is a system capable to process a wide variety of parts simultaneously due to its high degree of automation [36]. An AMS is composed of workstations where the parts are automatically processed, an automated material transport system that interconnects the entire AMS and a computer control system that integrates all the manufacturing tasks. The operation of an AMS consists in processing a number of different parts. The parts are considered discrete entities which flow through a determined number of operations. Resources are required to perform the operations such as numerically controlled machines and material handling devices such as robots to load or unload the machines. Storage areas are also common components of an AMS, especially those which decouple operations referred to as buffers. Generally, the number of resources is limited and they need to be shared in order to carry out the operations. Resource sharing may originate well-known problems like conflicts and deadlocks [37] and diminish the system performance.

Figure 1 shows a drawing of an AMS. The manufacturing process carried out by the system consists in machining raw parts which follow a fixed sequence of operations. Firstly, the parts go to an operation performed by Machine 1, then, they go to a second operation carried out by Machine 2, and finally, the parts finish their processing at Machine 3. The parts are transported by a conveyor throughout the

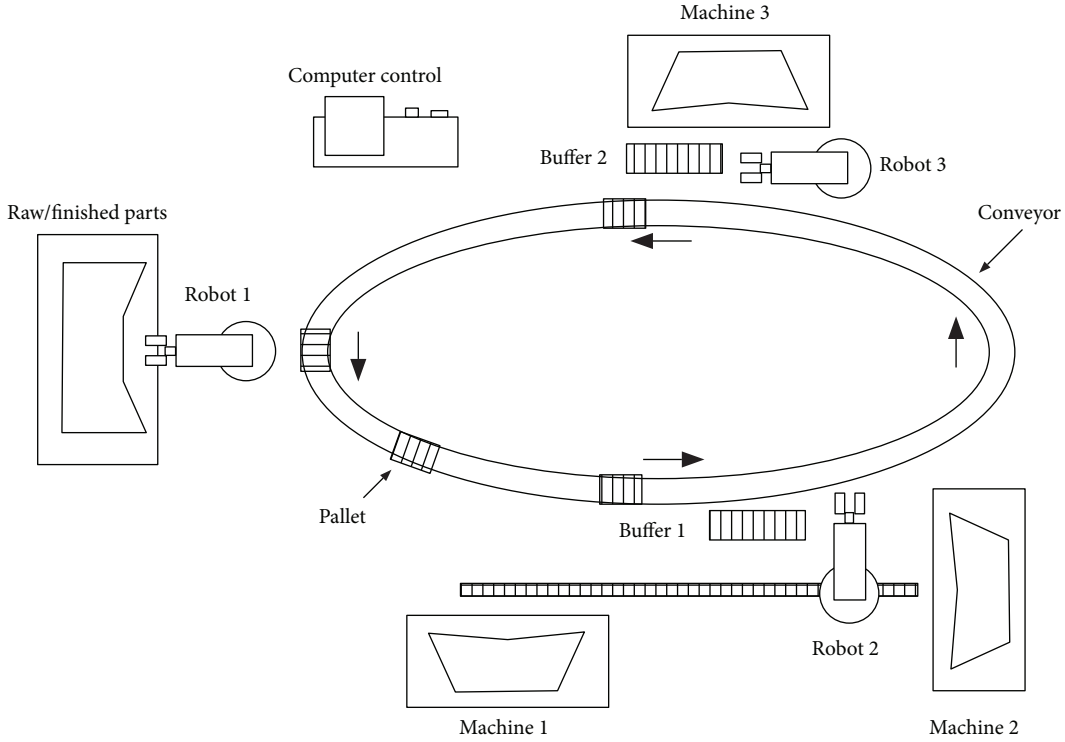


FIGURE 1: An automated manufacturing system (the arrows in bold shows the movement of the conveyor).

operations while robots load and unload the machines, the buffers, and the conveyor. The process begins when Robot 1 takes a raw part from a storage and puts it on pallets which are moved by the conveyor. Then, Robot 2 takes the part from the pallet, once the conveyor has stopped, and puts the part in Buffer 1. When the operation at Machine 1 has finished, Robot 2 withdraws the part and puts it in Buffer 1. If Machine 2 is idle, Robot 2 takes the part from Buffer 1 to Machine 2. As it can be seen in the figure, Robot 2 moves the parts from the conveyor belt to Buffer 1, from Buffer 1 to Machine 1 or Machine 2, from the Machines 1 and 2 to Buffer 1, and from Buffer 1 to the conveyor belt. All these movements are performed once Robot 2 is free and a part is ready to be moved.

When a part has been processed by Machine 2, it goes to the third and final operation. Once the part has reached the corresponding station, Robot 3 takes it to Buffer 2, and if Machine 3 is idle, then, the part is moved to the machine by Robot 3. When the third operation has finished, the part is taken to the storage for finished parts. Robot 1 takes the finished part from the conveyor belt and puts it in the storage area.

2.2. One-Dimensional Cellular Automata. A cellular automaton is a dynamical system whose evolution is discrete in time and space [38]. Cellular automata consist of identical elements called cells, and every cell can assume a state from a finite set of states. Each cell interacts with the nearest ones forming a neighborhood in order to change its state in a local way, causing a global state change of the automaton. Cells

evolve individually and simultaneously to the next time step according to an evolution rule that determines which will be the next state of each cell based on its current state and the ones of its neighbors. This is referred to as a local evolution rule. The following explanations are focused on cellular automata in one dimension.

Formally, the local evolution rule is defined by $\varphi : S^n \rightarrow S$, where S is the finite set of states and n is the size of the neighborhoods. We define the size of the neighborhoods as $n = l + r + 1$ such that r and l are the numbers of neighbors to the right and to the left, respectively, of every cell. r and l are fixed values and constant for every cell c_i ; the index i represents the position of the cell in the one-dimensional array of cells. In this way, let $c_i^t = a$ be the state of cell c_i at the instant t such that $a \in S$. Thus, $c_i^{t+1} = \varphi(c_{i-l}^t, \dots, c_{i-1}^t, c_i^t, c_{i+1}^t, \dots, c_{i+r}^t)$. A configuration C consists in an assignment of states to every cell in the linear arrangement. In particular, C^0 is the initial configuration, at the instant $t = 0$. Figure 2 shows the application of the local evolution rule in a one-dimensional cellular automaton of two states and neighborhoods of size $n = 3$, with $r = 1$ and $l = 1$.

Although theoretically one-dimensional cellular automata have an infinite number of cells, in practice and due to computational constraints, they consist of a limited number of cells. However, the cell that is at the rightmost side of the linear array and the one that is at the leftmost side lack from the right or the left neighbors, accordingly. A common practice to resolve this problem and to maintain a homogeneous neighborhood size for every cell is to consider periodic boundaries where the rightmost cell in the array takes as right

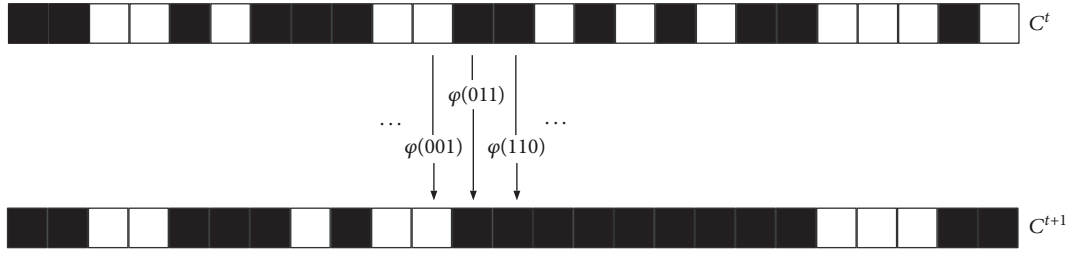


FIGURE 2: Example of the evolution of a one-dimensional cellular automaton with $n = 3$ and states 0 and 1 depicted as white and black squares, respectively.

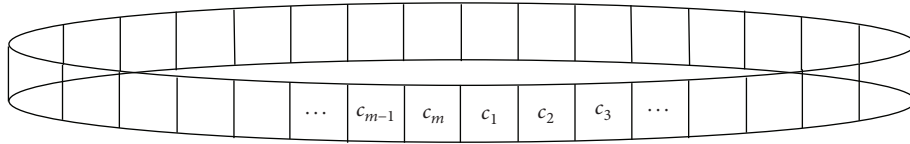


FIGURE 3: Ring formation of the linear array of cells, where m is the number of cells.

neighbors the first r cells of the left side of the array and likewise the leftmost cell takes as left neighbors the last l cells of the right side of the array. This can be thought of as forming a ring with the linear array as it is shown in Figure 3.

According to the long-term evolution of an automaton and Stephen Wolfram's classification, its global dynamics can be organized into four classes [39, 40]. Class I evolution tends to a fixed or stable state; Class II shows periodic and stable evolutions. For Class III, evolutions present a chaotic behavior and Class IV is considered the most interesting since evolutions produce complex behaviors over a stable or periodic background.

3. Modeling Method Assumptions

In this section, we give the basics of our proposed methodology to model an AMS with one-dimensional cellular automata. The modeling method is based on the assumptions that only one type of part is processed in the AMS and that the parts flow through the manufacturing process following the same sequence of operations. One way to model manufacturing systems is by means of fluid models where the parts go through a sequence of operations until they depart from the system. Thus, one-dimensional cellular automata are used in this paper taking into account such type of models, providing a way to represent not only the sequence of activities but the interaction among the resources. Additionally, many studies have been conducted to determine the properties of one-dimensional cellular automata [41–43], and a De Bruijn diagram is one of the classical tools to perform preliminary analyses. We take advantage of this tool to define the evolution rule by which the manufacturing system will be controlled.

The modeling approach does not take into account the time an operation takes to be executed, but only considers if a part is at an operation or not. If a part is at an operation, the resource that performs such an operation is busy and, no

matter how much time the operation takes, the next state of the part is to be in another operation according to the sequence of operations, and the resource may become idle.

In most AMS, some resources must be shared by the operations and this causes conflicts. A conflict arises when such operations request the shared resource at the same time. In the proposed modeling approach, the shared resources are not modeled by the cells of the automaton, but they are taken into account in determining the evolution rule. Only resources that are not shared are represented by each cell of the automaton. The possible states of the cells are 1 or 0, depending on whether the corresponding operation is being executed or not, respectively. A cell on state 1 not only means the corresponding operation is being executed but also means that a part is ready to be moved to the next operation such that a shared resource like a robot may be required to carry out the movement.

The movement of a part from one operation to another is represented by the evolution of the automaton, but the movement is subject to constraints in the flow of parts. Two types of constraints are considered: the first one is related to the request of a shared resource at the same time, and the second one is about the capacity of the resources to process or hold only one part at a time.

The sequence of operations is modeled in such a way that the first cell at the left of the array of cells represents the first operation; the second cell, the second operation; and so on. A cell changes from 0 to 1 whenever the previous cell is on state 1 which means that the part moves from one operation to the following one whenever there is no flow constraint such as the ones mentioned above.

Due to that a resource may be used at different stages of the manufacturing process, different cells may represent the same resource at different positions in the linear array of cells like a buffer that can be modeled by a cell before a machining operation and then another cell can model the same buffer after finishing the operation, meaning that the buffer holds

TABLE 1: Operations of the AMS shown in Figure 1 and the resources involved.

Resource	Activities	Resources requested to be available
Storage area	Hold raw parts and finished parts	Robot 1
Robot 1	Load and unload the conveyor	Conveyor and storage area
Robot 2	Load and unload Buffer 1, Machine 1, and Machine 2	Buffer 1, Machine 1, and Machine 2
Robot 3	Load and unload Buffer 2, Machine 3, and conveyor	Buffer 2, Machine 3, and conveyor
Machine 1	Operation 1	Robot 2
Machine 2	Operation 2	Robot 2
Machine 3	Operation 3	Robot 3
Conveyor	Transport parts	Robot 1, Robot 2, and Robot 3
Buffer 1	Temporally hold parts	Robot 2
Buffer 2	Temporally hold parts	Robot 3

unprocessed and processed parts, correspondingly. Then, another operation can follow and the same buffer can be used to hold the parts in the same way as the previous operation.

3.1. General Considerations to Obtain the Local Evolution Rule. The dynamics of the AMS rely on the construction of the local evolution rule of the cellular automaton. Resource sharing and capacity constraints are key to define the local evolution rule in such a way that the possible interactions between a resource and its nearest neighbors in the AMS operation are represented by the neighborhoods.

The size of the neighborhoods is determined by the cells representing an operation requesting the use of the same resource. For example, consider operations i and j which are modeled by cells c_i and c_j , with $i < j$ due to the sequence of activities, and assume that such activities request the use of the same resource such that c_j is the rightmost cell in the array requesting such a resource. Then, cell c_i is affected by the state of cell c_j in such a way that if $c_i = 1$ and $c_j = 1$, they are said to be in conflict, and therefore, c_j must be included into the neighborhood of c_i . Moreover, all of the cells between c_i and c_j , if any, will be also members of the neighborhood of c_i whether or not the middle cells request the same resource.

On the other hand, the way the flow of parts is modeled indicates that a cell changes its state from 0 to 1 whenever the previous cell is on state 1. Under this consideration, cell c_i is affected not only by cell c_j but by the state of cell c_{i-1} . Note that cell c_{i-1} does not request the same shared resource as cells c_i and c_j do. In this way, cells $c_{i-1}c_i c_{i+1}, \dots, c_{j-1}c_j$ form a neighborhood, such that $c_i^{t+1} = \varphi(c_{i-1}c_i c_{i+1}, \dots, c_{j-1}c_j)$. If other neighborhoods can be constructed in the same way, but not for operations that share a different resource, the neighborhood with the maximum number of cells from corresponding cells c_{i-1} to c_j defines the size of all of the neighborhoods. Additionally, cells $c_{i-1}c_i c_{i+1}, \dots, c_{j-1}c_j$, which define the size of the neighborhoods, become the *standard neighborhood* in order to determine the evolution rule. That is, each time c_i^{t+1} is going to be determined for each neighborhood formed by every cell in the linear array, the flow constraints among the operations represented in the standard neighborhood $c_{i-1}c_i c_{i+1}, \dots, c_{j-1}c_j$ are taken into account as

if every neighborhood has the same cells, namely, those of the standard neighborhood.

In order to resolve the flow constraints, such as the conflicts and the capacity of the resources described before, a policy must be defined since such situations are unavoidably represented by some of the neighborhoods according to the relations given among the cells of the standard neighborhood. Suppose $c_i^t = 1$ and $c_j^t = 1$ such that they request the same resource at the same time to move the corresponding part from the current operation to the next one. To provide a solution to this situation, a priority is given to activity j ; that is to say, the part at operation c_j is allowed to be moved first, whether or not such a movement can be done. Meanwhile, c_i remains unchanged at the next time step. This way to solve the conflicting situation is called the *priority policy* in the proposed modeling method. The priority policy is also applied to the capacity constraint when $c_i^t = 1$ and $c_{i+1}^t = 1$ such that the flow from c_i to c_{i+1} cannot be performed until the part at c_{i+1} releases the corresponding resource.

4. Modeling Method

4.1. AMS Description. The first step of the proposed method is to have a description of the operations of the AMS and the resources used to perform them. Likewise, it is needed to have an understanding of the whole operation of the AMS in order to identify the constraints related to carrying out the activities. As with any modeling method, these considerations are general and they can be applied to different manufacturing systems to obtain the corresponding cellular automaton model. The explanations given in this section of how to obtain the evolution rule as well as the arrangement of the cells are intended to be generalized to similar manufacturing systems since these types of systems do not vary significantly in their structure and operation, and it could be better for systems with more resources than the one exemplified here. Then, let us refer to the AMS that is shown in Figure 1. Table 1 provides the activities performed by all of the resources in the AMS as well as the resources or conditions required to carry out such activities. It is considered for this AMS that the conveyor is always available, there

of $2^7 = 128$ neighborhoods with the form $c_{i-1}c_i c_{i+1}, \dots, c_{m-1}c_m$, where c_i is the cell designated to evolve.

The constraints indicated for cells c_2, c_3, c_4, c_5, c_6 , and c_7 with regard to sharing Robot 2 are used to define the evolution rule. Although such constraints may not apply to all the cells and their corresponding neighborhoods, they are used to set policies to the flow of parts to model the AMS dynamics. Therefore, all of the 128 neighborhoods are considered to be possible states of cells $c_1, c_2, c_3, c_4, c_5, c_6$, and c_7 . For example, the neighborhood 0100000 means that Robot 2 is required to move the part from the conveyor (c_2) to Buffer 1 (c_3). Thus, for this neighborhood, $\varphi(0100000) = 0$ since the remainder of the cells after c_2 is not requesting the shared resource and the buffer is free to receive the part, leaving empty the conveyor after the movement since there is not any other part arriving at the buffer. The evolution of the automaton corresponding to this situation is partially shown in Figure 6.

Consider now the neighborhood 0010010 where Buffer 1 (c_3) and Machine 2 (c_6) request Robot 2 to move the corresponding part to the following activities: namely, to Machine 1 and Buffer 1 (c_4 and c_7 , correspondingly). In this case, c_3 and c_6 are in conflict and both of the movements cannot occur simultaneously since they are mutually exclusive. By the priority policy applied to the activity at the downstream of the manufacturing process, the movement from Machine 2 (c_6) to Buffer 1 (c_7) is allowed to occur first. Thus, $\varphi(0100100) = 1$ since cell c_2 must remain unchanged allowing the occurrence of the other activity. The evolution of the cellular automaton for this case is shown in Figure 7. It can be observed that in the following neighborhood formed by cell c_3 (1000100), such a cell must stay on state 0 at the next time step because cell c_2 does not change its state. That is, there is no movement from the conveyor (c_2) to Buffer 1 (c_3) since Robot 2 is used to move a part from Machine 2 (c_6) to Buffer 1 (c_7), and thus, $c_3^{t+1} = 0$. This way to determine the states of the cells at the next time step is the guideline to determine the evolution rule for all of the 128 neighborhoods. That is, the thing considered is not only the current state of the cells in the neighborhood but also the previous movements in the flow of the parts.

The general process of the evolution rule can be seen as going through the paths of the De Bruijn diagram of the cellular automaton obtained from the AMS of Figure 1, forming all the possible neighborhoods while traversing the paths. The De Bruijn diagram consists of 2^{n-1} nodes, and each node is a partial neighborhood of $n-1$ cells. To form a complete neighborhood, two nodes u and v are connected by a directed link from u to v as long as the rightmost $n-2$ cells of node u overlap with the leftmost $n-2$ cells of node v . At most, each node has $|S|$ leaving links. Therefore, to construct the evolution rule, a pair of nodes u and v must be chosen such that u overlaps with v , and then state c_i^{t+1} is determined as it was explained above. Next, another node w is chosen such that v overlaps with w , and one more time state c_i^{t+1} is determined. The process continues forming a path $u \rightarrow v \rightarrow w \rightarrow \dots$, and eventually, the path will either form a cycle or fall into a cycle.

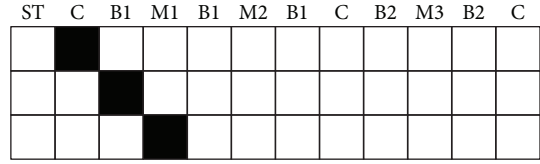


FIGURE 6: Evolution representing Robot 2 moving a part from the conveyor to Buffer 1.

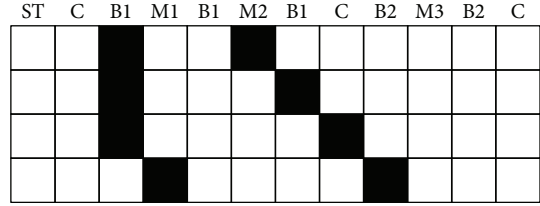


FIGURE 7: Robot 2 moving a part from Machine 2 to Buffer 1 and then to the conveyor, while another part in Buffer 1 waits until Robot 2 finishes the movements.

To illustrate the process, let us consider neighborhood 0100010 formed by nodes 010001 and 100010, as it is shown in Figure 8. For this neighborhood, $\varphi(0100010) = 1$ since cells c_2 and c_6 request the same resource at the same time and, by the priority policy, the operation at the downstream of the manufacturing process is allowed to use first the shared resource. After node 100010, one can go to either node 000100 or node 000101. Let us take the latter, that is, 000101, and form neighborhood 1000101. According to the decision made in the previous neighborhood, c_1 remain unchanged at $t+1$ in the current neighborhood, then c_2 do not receive any part from the previous operation, and thus, $\varphi(1000101) = 0$. As it can be observed, each time a node is chosen to continue the path, the partial neighborhood can end in 0 or 1. In order to give a brief description of the process to determine the evolution rule, the neighborhoods which end in 0 are chosen deliberately. In this way, after 000101, node 001010 is chosen such that neighborhood 0001010 is obtained. In this case, there is not any part to move from cell c_1 to cell c_2 and this last cell is not requesting the shared resource; therefore, $\varphi(0001010) = 0$. The same happens with neighborhood 0010100 formed by nodes 001010 and 010100, such that $\varphi(0010100) = 0$.

After 010100, the process continues with node 101000 and the resulting neighborhood is 0101000. In this situation, c_2 and c_4 are requesting the same resource, and by the priority policy, $\varphi(0101000) = 1$. Observe that according to the first neighborhood that originated this process, namely, 0100010, c_6 should change to 0 by applying the priority policy; however, due to the conditions of the new neighborhood 0101000, such a cell (now c_2) must remain unchanged at the next time step. As it can be noticed from the last node 101000, this one overlaps with node 010001 that started the path. Neighborhood 1010001 results from overlapping nodes 101000 and 010001, and for this case, $\varphi(1010001) = 0$; that is c_1 does not send any part to c_2 according to the previous

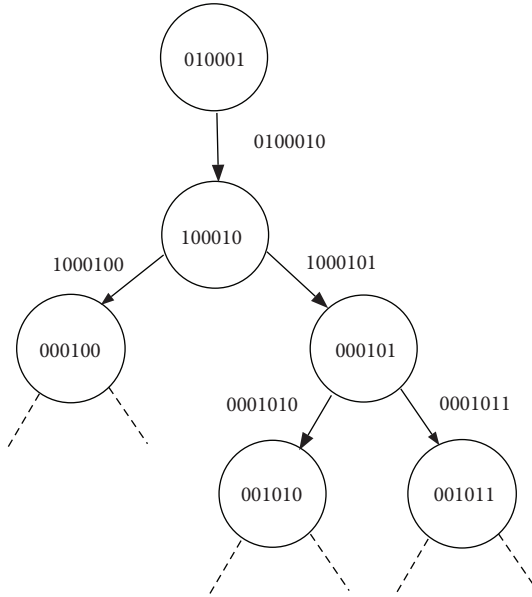


FIGURE 8: Process to determine the evolution rule following a path in the De Bruijn diagram.

neighborhood. Up to here, the process to determine the evolution rule is finished since the path followed has formed a cycle. The states determined for each neighborhood must be consistent with the flow of parts as it is shown in the De Bruijn subdiagram in Figure 9.

The process is applied to all of the remaining neighborhoods in the paths that have not been explored. Eventually, two or more paths will have common nodes, and for the neighborhoods formed by such nodes, the evolution rule definition must be consistent with the flow of parts considered in each path. Figure 10 presents some paths that share nodes with the cycle shown in Figure 9. Observe that the states determined for the evolution rule in each path is consistent with the flow of parts and the constraints considered for such a flow. Table 2 shows the complete local evolution rule definition, obtained as indicated in this section.

5. Cellular Automata Evolution Representing the Dynamics of the AMS

Once the evolution rule has been determined, the evolution of the cellular automaton can be carried out for different initial conditions of the AMS, which are represented by the initial configuration. In order to represent the dynamics of the AMS, a modification to the classical periodic boundary conditions is made, particularly to the right boundary. Since each cell of a configuration forms its neighborhood with $l = 1$ cell to the left and $r = 5$ cells to the right, the last five cells could not do it. Instead of completing the neighborhood of the last five cells with the ones at the beginning of the configuration, the missing cells are given state 0 as if the configuration was infinite to the right side, but with the cells with state 0. In fact, only the missing cell which is next to the last cell of the configuration will take the state of the first cell of the configuration. Figure 11 shows an evolution with the

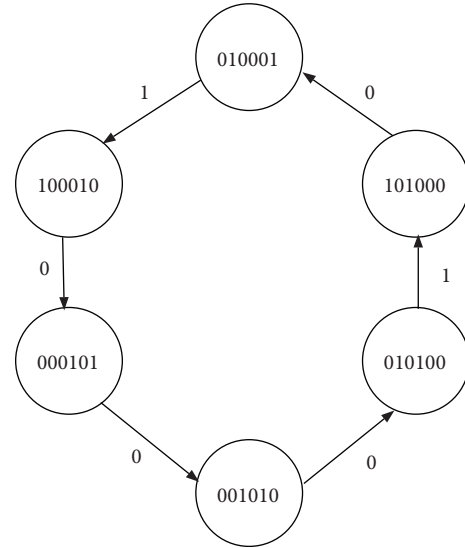


FIGURE 9: Path forming a cycle in the De Bruijn diagram starting and ending at node 010001. The result of the evolution is indicated by a number (0 or 1) next to the arrows.

resulting modification of the boundary condition. In that figure, the cell representing the third machine (M3) in the initial configuration is missing three of the five cells to the right to complete its neighborhood, namely, the third, the fourth, and the fifth. By the modification to the right boundary condition, the third cell takes the state of the first cell of the initial configuration, that is, 1. The fourth and the fifth cells take state 0 forming neighborhood 0100100.

Figure 12(a) shows the evolution where only one part is in the AMS at the storage area (c_1). According to the AMS, the part goes through the whole manufacturing process from one operation to another since there is no constraint in its flow towards the end of the process. In Figure 12(b), a conflict situation is shown when initial state 100100100011 is chosen. In this initial condition, the storage area has a part ready to enter to the manufacturing process (c_1) where Robot 1 is requested to move the part to the conveyor. Likewise, a part is at Machine 1 (c_4) such that it requests Robot 2 to move such a part to Buffer 1 once the machining operation has finished. At the same time, Buffer 1 (c_7) requests Robot 2 to move the part in the buffer to the conveyor in order that it can continue with the final operation at Machine 3. Finally, there is a part in the conveyor (c_{12}) ready to be put in the storage by Robot 1. All of these conditions lead to a conflict since according to the evolution rule definition and by the priority policy, there is a circular request among the involved cells or operations.

Figure 12(c) shows a part in the storage area and one in Buffer 1 (c_3). According to the evolution rule, the part at the storage area is not moved to the conveyor until the part at Buffer 1 has been moved forward in the manufacturing process such that it is out of the neighborhood of c_1 . The initial state in Figure 12(d) shows a part in the storage area and two parts in Buffer 1. One of the parts in Buffer 1 is ready to go to Machine 1, namely, the one at c_3 , and the other one (c_5) to Machine 2. Since both movements cannot occur

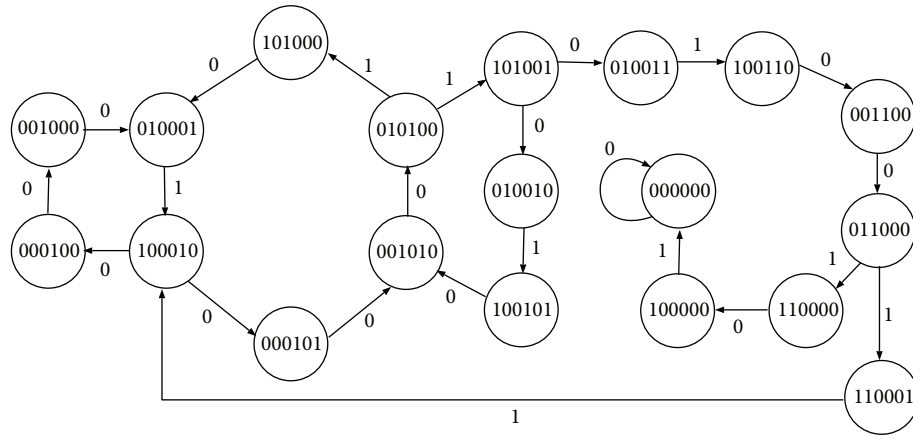


FIGURE 10: Some paths of the De Bruijn diagram showing the states determined for the evolution rule.

TABLE 2: Local evolution rule where $c_i^{t+1} = \varphi(c_{i-1}^t, c_i^t, c_{i+1}^t, \dots, c_{j-1}^t, c_j^t)$.

Neighborhood	c_i^{t+1}	Neighborhood	c_i^{t+1}	Neighborhood	c_i^{t+1}	Neighborhood	c_i^{t+1}
000000	0	010001	1	100010	0	110011	1
000001	0	010010	1	100011	0	1100100	1
000010	0	010011	1	1000100	0	1100101	1
000011	0	0100100	1	1000101	0	1100110	1
0000100	0	0100101	1	1000110	0	1100111	1
0000101	0	0100110	1	1000111	0	1101000	1
0000110	0	0100111	1	1001000	0	1101001	1
0000111	0	0101000	1	1001001	0	1101010	1
0001000	0	0101001	1	1001010	0	1101011	1
0001001	0	0101010	1	1001011	0	1101100	1
0001010	0	0101011	1	1001100	0	1101101	1
0001011	0	0101100	1	1001101	0	1101110	1
0001100	0	0101101	1	1001110	0	1101111	1
0001101	0	0101110	1	1001111	0	1110000	1
0001110	0	0101111	1	1010000	0	1110001	1
0001111	0	0110000	1	1010001	0	1110010	1
0010000	0	0110001	1	1010010	0	1110011	1
0010001	0	0110010	1	1010011	0	1110100	1
0010010	0	0110011	1	1010100	0	1110101	1
0010011	0	0110100	1	1010101	0	1110110	1
0010100	0	0110101	1	1010110	0	1110111	1
0010101	0	0110110	1	1010111	0	1111000	1
0010110	0	0110111	1	1011000	0	1111001	1
0010111	0	0111000	1	1011001	0	1111010	1
0011000	0	0111001	1	1011010	0	1111011	1
0011001	0	0111010	1	1011011	0	1111100	1
0011010	0	0111011	1	1011100	0	1111101	1
0011011	0	0111100	1	1011101	0	1111110	1
0011100	0	0111101	1	1011110	0	1111111	1
0011101	0	0111110	1	1011111	0		
0011110	0	0111111	1	1100000	0		
0011111	0	1000000	1	1100001	1		
0100000	0	1000001	1	1100010	1		

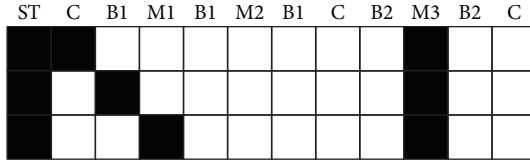


FIGURE 11: Evolution of the cellular automaton considering the boundary condition modification to the last five cells to the right side of each configuration.

simultaneously, the part at c_5 is chosen to be moved first by the priority policy at $t = 1$. At the next two time steps, the priority policy is applied again such that the part is moved from c_6 to c_7 at $t = 2$ and from c_7 to c_8 at $t = 3$. At $t = 4$, any part can be moved according to the evolution rule definition.

Some evolutions of the cellular automaton are provided in Figure 13 for disordered initial configurations of different widths. For a long number of generations depicted in Figures 13(c)–13(e), the cellular automaton exhibits periodic configurations in determined regions as well as the preservation of data, matching the properties of Class II of Wolfram classification [44]. The evolutions show the tendency of the parts to move towards the end of the manufacturing process. For an initial configuration with large blank spaces (strings of 0s), such a feature is easily noticed (Figures 13(a) and 13(d)). On the contrary, when there are fewer blank spaces in the initial configurations, the system tends to be blocked (Figures 13(c) and 13(d)) and the expected movement of the parts is not recovered for a while. In regard to the latter, when large blank spaces are present in the initial configurations, the evolutions rapidly reach a periodic behavior, but once a conflict arises, a halt in the movement of parts is transmitted towards the beginning of the process, just as it was modeled by the evolution rule. However, the movement of parts can be recovered once the conflicts are resolved at the final stages of the process and this is also transmitted to the beginning after some time steps in the evolution. The time required to resolve the conflicts makes the system to be in conflict in a periodic fashion.

6. Experiments and Results

In this section, the operation of the system shown in Figure 1 is performed using the proposed control method and a Petri net model simulating the normal operation of the system without a predefined control of the parts flow. The comparison is made in terms of the time required to produce a certain number of parts and in terms of the resource utilization, namely, machines and robots. A Petri net (PN) is a well-known tool to model manufacturing systems, and a generalized Petri net model is appropriate to compare the operation of the system. Generalized Petri nets are not timed nor stochastic, and the evolution of markings can be easily matched with the configurations of the cellular automaton such that discreteness in space and time is satisfied for both types of simulations. However, some considerations are needed to make a fair comparison between the two models (cellular automaton and Petri net). Broadly speaking, in a PN, the change of states occurs asynchronously; that is, if

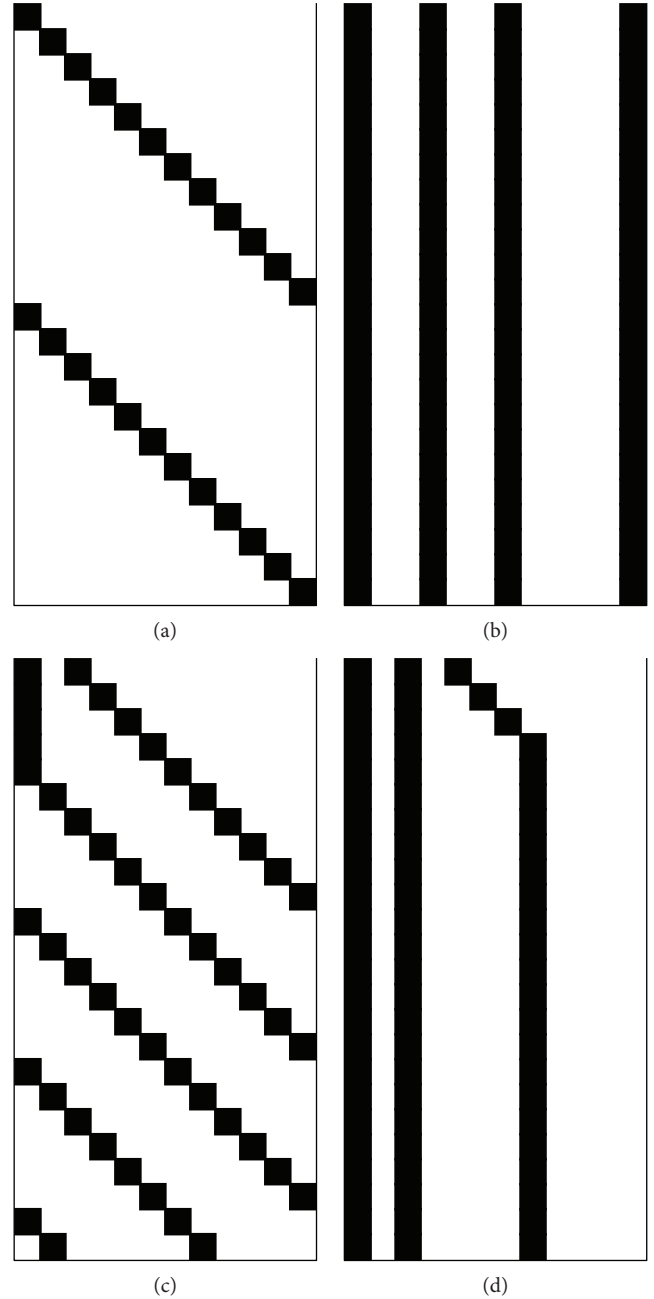


FIGURE 12: Evolutions of the cellular automaton representing different initial states of the AMS of Figure 1.

two movements can be done at the same time in the natural operation of a system, in the PN simulation, one of the movements is carried out first and then the other one, but not necessarily right one after the other. This is because of a matter of analysis which leads to the reachable marking tree, but if the two movements are parallel such that the occurrence of one them does not disable the occurrence of the other one at the next time step, then in the PN, the two simultaneous movements can be represented by firing the corresponding transitions one immediately after the other, reaching the same marking as if the two movements were done simultaneously. Another consideration is that for each

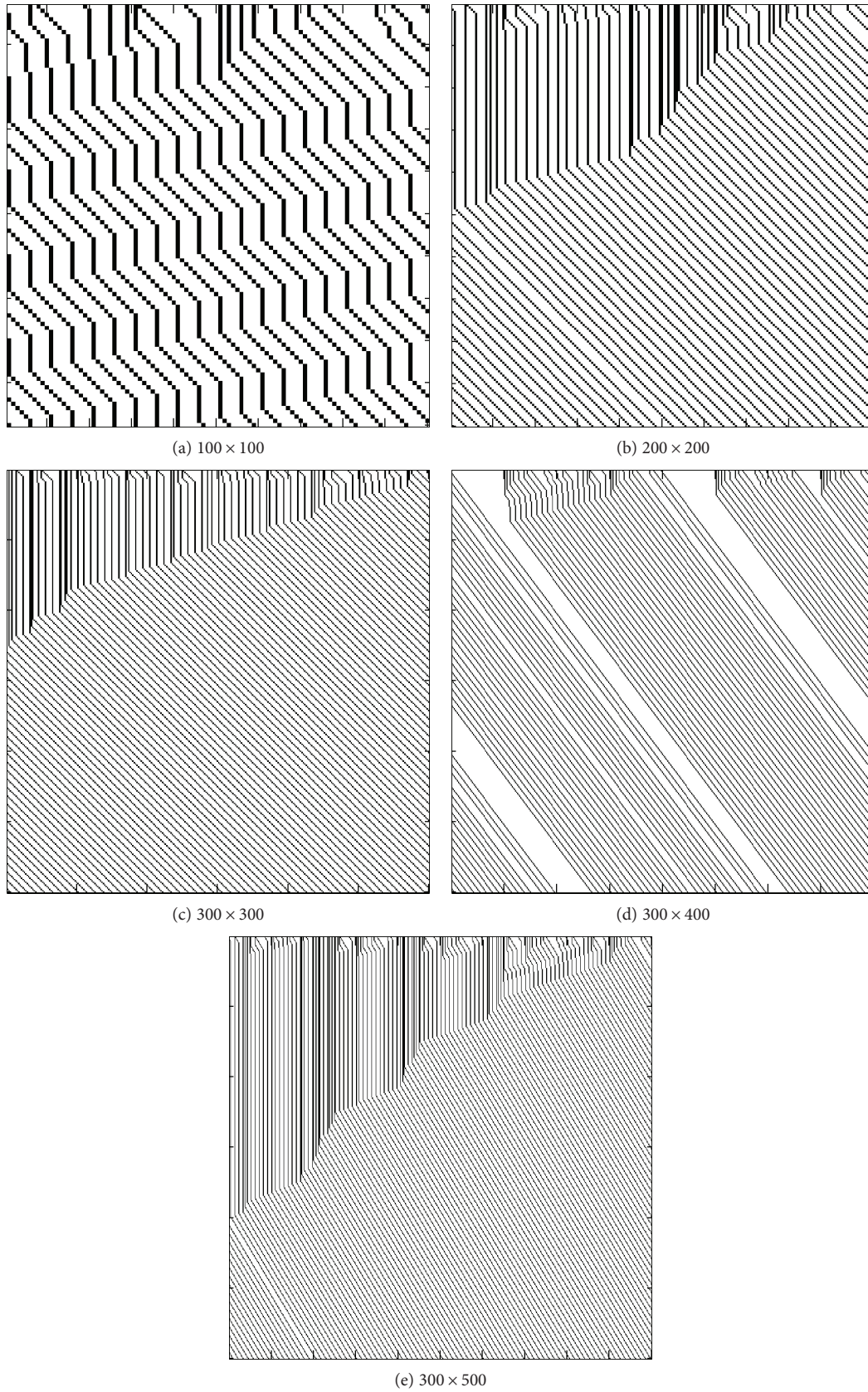


FIGURE 13: Evolutions of the cellular automaton for disordered initial configurations of different sizes (number of generations \times number of cells).

TABLE 3: Simulation results for the PN model and the cellular automaton.

Case	Parts	Machine utilization (%)						Robot utilization (%)						Time steps	
		M1		M2		M3		R1		R2		R3			
		PN	CA	PN	CA	PN	CA	PN	CA	PN	CA	PN	CA	PN	CA
0	1	8.33	8.33	8.33	8.33	8.33	8.33	50	50	33.33	33.33	16.67	16.67	12	12
1	2	16.67	12.5	11.11	12.5	11.11	12.5	61.11	68.75	44.44	50	16.67	18.75	18	16
2	2	7.69	7.69	15.39	15.39	15.39	15.39	61.54	61.54	61.54	61.54	15.39	15.39	13	13
3	3	6.67	5.88	13.33	35.29	20	17.65	46.67	41.18	80	70.59	20	17.65	15	17
4	3	8.33	6.67	8.33	6.67	25	20	41.67	33.33	91.67	73.33	25	20	12	15
5	4	6.25	55.56	12.5	11.11	18.75	16.67	43.75	38.89	75	66.67	25	22.22	16	18
6	4	5.26	4.55	10.53	50	21.05	18.18	42.11	36.36	78.95	68.18	21.05	18.18	19	22
7	5	8.33	7.69	12.5	11.54	25	19.23	58.33	53.85	75	69.23	20.83	19.23	24	26
8	5	4.76	7.69	14.27	42.31	42.86	19.23	47.62	53.85	85.71	69.23	23.81	23.01	22	26

time step in the cellular automaton evolution, two firings are needed in the PN simulation to reach a marking equivalent to the configuration reached in the automaton. These two firings represent the robots' movements which are not modeled in the cellular automaton.

Nine cases are used to compare the proposed control method with the normal operation of the system. Each case consists of a number of parts distributed at different stages in the system; this number of parts ranges from one to five. The PN model was constructed and simulated on a free application (<http://pipe2.sourceforge.net/>). For each case, the shortest path of firings was calculated for the PN. The shortest path is the minimum number of firings from the initial marking to the final one. The final marking is where all the parts have left the system, and they are stored in the storage area at the end of the system. For example, we developed the Case 0 to show the equivalence between the evolution of markings of the PN and the evolution of the cellular automaton. Case 0 considers only one part located at the storage area as an unprocessed part. The shortest path for this case is [T0 T1 T4 T5 T2 T3 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18 T19 T20 T21 T22 T23]. Since there are twelve stages where this part can be located through the system, at the conveyor, at Buffer 1, at Machine 1 being processed, and so on, the total number of time steps to reach the storage area for finished parts is twelve. This number of time steps is obtained from the shortest path by dividing each into two transitions from T0 to T23. In this way, the shortest path calculated for each of the eight cases represents the minimum time to process all the parts in the system. The shortest paths were recalculated from those obtained with the aforementioned PN application since the original ones do not consider simultaneous movements. The results are shown in Table 3.

7. Discussion

The evolution rule representing the dynamics of the AMS shown in Figure 1 is based on some considerations taken in regard to the number of operations that share resources and to the capacity constraint. Therefore, one can observe

in the evolutions shown in Figure 12 that some part movements could have been done, but they are not carried out due to constraints imposed to the evolution rule. For example, in Figure 12(c), Robot 1 is needed to move the part from c_1 to c_2 (storage area to conveyor) and Robot 2 is needed to move the part from c_3 to c_4 (Buffer 1 to Machine 1). However, by the priority policy, only the movement from c_3 to c_4 is allowed, although both of the movements are carried out by different resources. A similar situation occurs in the evolution shown in Figure 12(d), where the part at c_1 can be moved to c_2 since Robot 1 is needed to carry out the movement while Robot 2 is required to move the part from c_3 to c_4 . Despite this, the procedure to obtain the evolution rule can be seen as a general policy to avoid conflicts in the flow of parts. For example, in Figure 12(c), the priority policy avoids a conflict situation if both parts were moved simultaneously, arriving at the operations at Machine 1, Machine 2, and Buffer 1 where Robot 2 is shared. In a real situation, the decision to give the priority to some operation can be made under the consideration of, for example, the cost related to the operation and such a consideration can be taken into account in obtaining the evolution rule.

In regard to the results obtained from the experiments, only in Case 1, the cellular automaton produces the parts in a shorter time than in the PN simulation, with 16 time steps of the cellular automaton against 18 time steps of the PN. Moreover, the utilization of the resources is greater than in the PN simulation result. However, from Case 2 to Case 8, the utilization of the machines and robots with the cellular automaton is smaller than in the PN; however, the time needed to produce the parts is greater than in the PN simulation. Figure 14 shows the plots for the number of parts in the system of Case 0, Case 1, Case 3, Case 5, and Case 7. For Case 0, where there is only one part, the cellular automaton and the PN have the same plot. Such an equivalence is confirmed by the results shown in Table 3. For Cases 3, 5, and 7 where the cellular automaton produces the parts in slightly more time steps than the PN, it is observed that from the beginning of the cellular automaton evolution to the middle of such an evolution, the parts are produced faster than in the PN simulation. This leads to a low utilization of the resources.

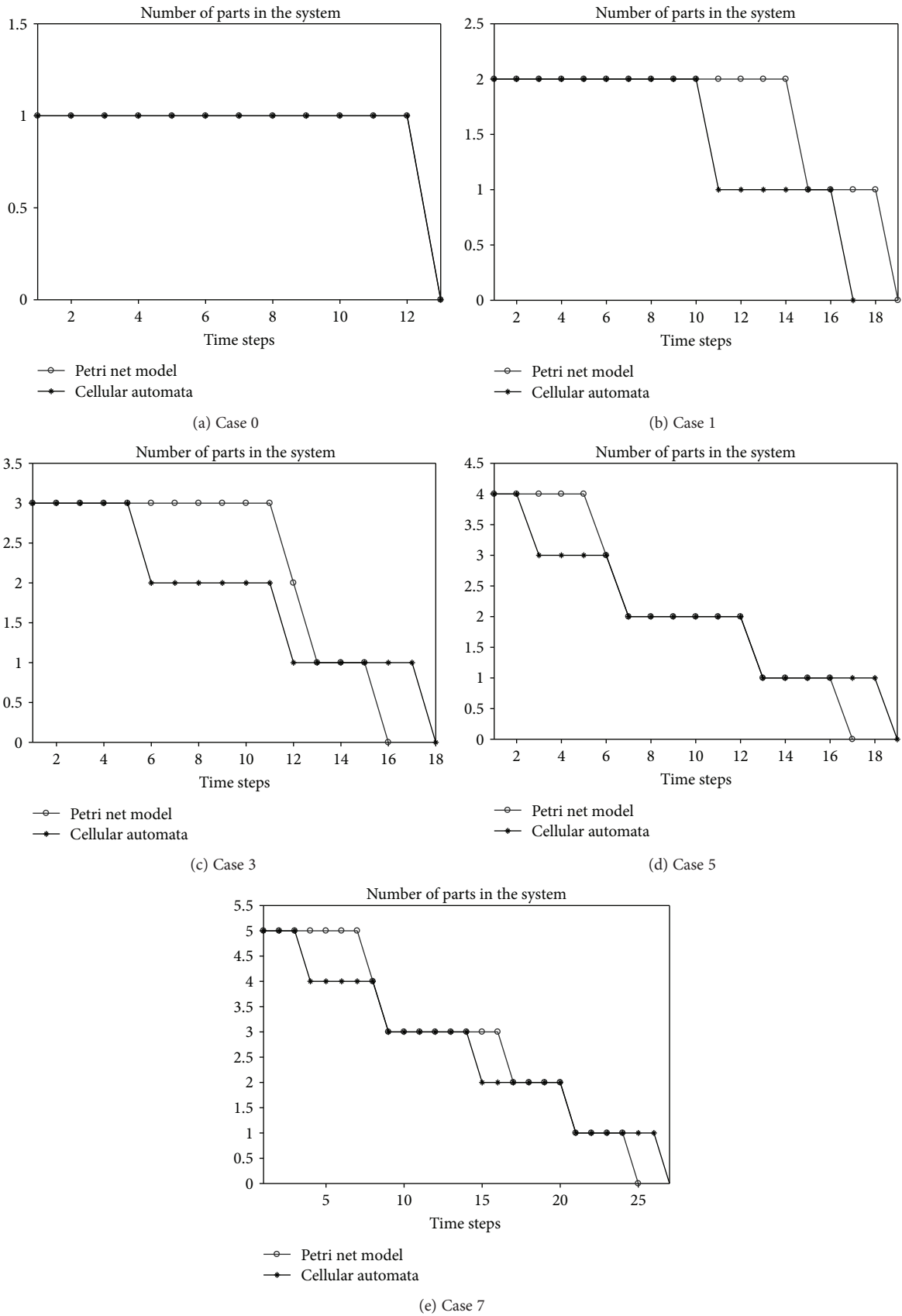


FIGURE 14: Plots showing the reduction of parts as they depart from the system.

Furthermore, a pattern in the production of parts is observed in the plots of Case 3, Case 5, and Case 7, which matches with the evolutions shown in Figure 13.

8. Conclusions

A method to model an automated manufacturing system with one-dimensional cellular automata has been presented, providing a new perspective in modeling the dynamics of such type of systems. Some considerations have been taken in order to represent the dynamics of an AMS such as that only one type of part is processed in the system and with a fixed sequence of operations. Likewise, the resources like the machines have been considered to have the capacity to process only one part at a time. As it can be seen in this paper, the decisions to control the flow of parts are taken locally, at each operation performed by the resources, which are modeled with the cells of the automaton and therefore obtaining a distributed control of the system. Moreover, due to the locality of the decisions, the control is not centralized on any entity but divided along to some of the entities of the system, namely, those modeled by the cells of the automaton.

The evolution of the cellular automaton shows the flow of the parts as it is expected in the real system. The local evolution rule delays the advance of the parts whenever a potential conflict can occur downstream in the process, and once the situation is resolved by a priority policy, the flow continues. However, it can be noticed from the evolutions that when there is an excess of parts in the system (initial configurations), this is blocked and therefore diminishing the efficiency of the system. Nevertheless, the comparative experiments show that the utilization of the resources is low for almost all the experiment cases. In a further work, different modeling strategies could be considered such as more states for the cells like failures and repairs; likewise, different local evolutions rule to manage different types of decisions. Indeed, the discreteness of the rules can be used instead of fuzzy rules, as it is done in many studies.

Data Availability

All the numerical results obtained to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by PRODEP grant F-PROMEP-38/Rev 03 and by the National Council for Science and Technology of Mexico (CONACYT) with project number CB-2014-237323.

References

- [1] H.-O. Guenther and T. E. Lee, "Scheduling and control of automated manufacturing systems," *OR Spectrum*, vol. 29, no. 3, pp. 373-374, 2007.
- [2] P. Leitão, "Agent-based distributed manufacturing control: a state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979-991, 2009.
- [3] D. Trentesaux, "Distributed control of production systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 971-978, 2009.
- [4] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: the ADACOR evolution," *Computers in Industry*, vol. 66, pp. 99-111, 2015.
- [5] R. V. Barenji, A. V. Barenji, and M. Hashemipour, "A multi-agent rfid-enabled distributed control system for a flexible manufacturing shop," *The International Journal of Advanced Manufacturing Technology*, vol. 71, no. 9-12, pp. 1773-1791, 2014.
- [6] T. Borangiu, S. Raileanu, D. Trentesaux, T. Berger, and I. Iacob, "Distributed manufacturing control with extended cnp interaction of intelligent products," *Journal of Intelligent Manufacturing*, vol. 25, no. 5, pp. 1065-1075, 2014.
- [7] J. A. Arauzo, R. del Olmo-Martínez, J. J. Laviós, and J. J. de Benito-Martín, "Programación y control de sistemas de fabricación flexibles: un enfoque holónico," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 12, no. 1, pp. 58-68, 2015.
- [8] K. Tamani, R. Boukezzoula, and G. Habchi, "Intelligent distributed and supervised flow control methodology for production systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 1104-1116, 2009.
- [9] K. Rudnik and D. Kacprzak, "Fuzzy topsis method with ordered fuzzy numbers for flow control in a manufacturing system," *Applied Soft Computing*, vol. 52, pp. 1020-1041, 2017.
- [10] K. Tamani, R. Boukezzoula, and G. Habchi, "Supervisory-based capacity allocation control for manufacturing systems," *International Journal of Manufacturing Technology and Management*, vol. 20, no. 1-4, pp. 259-285, 2010.
- [11] M. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*, World Scientific, New York, NY, USA, 1998.
- [12] Z. Li, N. Wu, and M. Zhou, "Deadlock control of automated manufacturing systems based on petri nets—a literature review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 437-462, 2011.
- [13] B. K. Choi, H. Kim, D. Kang, A. A. Jamjoom, and M. A. Abdullah, "Parameterized acd modeling of flexible manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 637-642, 2014.
- [14] J. Novak-Marcincin, "Computer modelling and simulation of automated manufacturing systems," *Annals of the Faculty of Engineering Hunedoara-International Journal of Engineering*, vol. 11, no. 2, pp. 23-26, 2013.
- [15] Q. Wang and C. R. Chatwin, "Key issues and developments in modelling and simulation-based methodologies for manufacturing systems analysis, design and performance evaluation," *International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11-12, pp. 1254-1265, 2005.
- [16] S. Wolfram, "Universality and complexity in cellular automata," *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 1-35, 1984.
- [17] S. Athithan, V. P. Shukla, and S. R. Biradar, "Dynamic cellular automata based epidemic spread model for population in

- patches with movement,” *Journal of Computational Environmental Sciences*, vol. 2014, Article ID 518053, 8 pages, 2014.
- [18] P. Eosina, T. Djatna, and H. Khusun, “A cellular automata modeling for visualizing and predicting spreading patterns of dengue fever,” *Telkomnika*, vol. 14, no. 1, pp. 228–237, 2016.
- [19] M. Martnez Molina, M. A. Moreno-Armendriz, and J. C. Seck Tuoh Mora, “On the spatial dynamics and oscillatory behavior of a predator-prey model based on cellular automata and local particle swarm optimization,” *Journal of Theoretical Biology*, vol. 336, pp. 173–184, 2013.
- [20] X. Ke, F. Wu, and C. Ma, “Scenario analysis on climate change impacts of urban land expansion under different urbanization patterns: a case study of wuhan metropolitan,” *Advances in Meteorology*, vol. 2013, Article ID 293636, 12 pages, 2013.
- [21] Z. Wang, S. Luo, H. Song, W. Deng, and W. Li, “Simulation of microstructure during laser rapid forming solidification based on cellular automaton,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 627528, 9 pages, 2014.
- [22] M. Eibinger, T. Zahel, T. Ganner, H. Plank, and B. Nidetzky, “Cellular automata modeling depicts degradation of cellulose material by a cellulase system with single-molecule resolution,” *Biotechnology for Biofuels*, vol. 9, no. 1, p. 56, 2016.
- [23] X. Li, X. Yan, X. Li, and J. Wang, “Using cellular automata to investigate pedestrian conflicts with vehicles in crosswalk at signalized intersection,” *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 287502, 16 pages, 2012.
- [24] M. E. Larraga and L. Alvarez-Icaza, “Cellular automata model for traffic flow with safe driving conditions,” *Chinese Physics B*, vol. 23, no. 5, article 050701, 2014.
- [25] M. Bukacek and P. Hrabak, “Boundary induced phase transition in cellular automata models of pedestrian flow,” *Journal of Cellular Automata*, vol. 11, pp. 327–338, 2016.
- [26] H. Li, C. Shao, H. Wu, J. Tian, and Y. Zhang, “Cellular automata approach for modeling lane changing execution,” *Journal of Cellular Automata*, vol. 11, pp. 339–350, 2016.
- [27] H. C. Shen and W. P. Yan, “Modelling autonomous assembly systems and fms using cellular automata,” *The International Journal of Advanced Manufacturing Technology*, vol. 7, pp. 333–338, 1992.
- [28] H. C. Shen, H. L. Chau, and K. K. Wong, “An extended cellular automaton model for flexible manufacturing systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 11, no. 4, pp. 258–266, 1996.
- [29] S. Minami, K. F. Pahng, M. J. Jakiela, and A. Srivastava, “A cellular automata representation for assembly simulation and sequence generation,” in *Proceedings. IEEE International Symposium on Assembly and Task Planning*, pp. 56–65, Pittsburgh, PA, USA, August 1995.
- [30] I. Barragán, J. C. Seck Tuoh, and J. Medina, “Relationship between Petri nets and cellular automata for the analysis of flexible manufacturing systems,” in *Advances in Computational Intelligence. MICAI 2012*, I. Baturshin and M. Gonzalez, Eds., vol. 7630 of Lecture Notes in Computer Science, pp. 338–349, Springer, Berlin, Heidelberg, 2013.
- [31] Z. Li and M. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*, Springer-Verlag, London, 2009.
- [32] W. Nai Qui and M. Zhou, Eds., *Systems Modeling and Control with Resource-Oriented Petri Nets*, CRC Press, Boca Raton, FL, USA, 2010.
- [33] M. P. Cabasino, M. Dotoli, and C. Seatzu, “Modelling manufacturing systems with place/transition nets and timed Petri nets,” in *Formal Methods in Manufacturing*, J. Campos, C. Seatzu, and X. Xie, Eds., pp. 3–27, CRC Press, 2014.
- [34] P. J. O’Grady, Ed., *Controlling Automated Manufacturing Systems*, Springer, Dordrecht, 1986.
- [35] H. K. Shivanand, *Flexible Manufacturing System*, New Age International, 2006.
- [36] P. M. Swamidass, Ed. P. M. Swamidass, Ed., “Automated Manufacturing System,” in *Encyclopedia of Production and Manufacturing Management*, pp. 50–51, Springer, Boston, MA, USA, 2000.
- [37] W. C. Yeh, “Real-time deadlock detection and recovery for automated manufacturing systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 20, no. 10, pp. 780–786, 2002.
- [38] A. Mitra and H. N. Teodorescu, “Detailed analysis of equal length cellular automata with fixed boundaries,” *Journal of Cellular Automata*, vol. 11, no. 5–6, pp. 425–448, 2016.
- [39] S. Wolfram, “Computation theory of cellular automata,” *Communications in Mathematical Physics*, vol. 96, no. 1, pp. 15–57, 1984.
- [40] L. D’Alotto, “A classification of one-dimensional cellular automata using infinite computations,” *Applied Mathematics and Computation*, vol. 255, pp. 15–24, 2015.
- [41] H. V. McIntosh, *One-Dimensional Cellular Automata*, Luniver Press, London, 2009.
- [42] W. Jin and F. Chen, “Topological chaos of universal elementary cellular automata rule,” *Nonlinear Dynamics*, vol. 63, no. 1–2, pp. 217–222, 2011.
- [43] Z. Bie, Q. Han, C. Liu, J. Huang, L. Song, and Y. Pei, “Chaotic behavior of one-dimensional cellular automata rule 24,” *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 304297, 8 pages, 2014.
- [44] S. Wolfram, *Theory and Applications of Cellular Automata, Volume 1 of Advanced Series on Complex Systems*, World Scientific, Philadelphia, 1986.

Research Article

Couple-Group Consensus: A Class of Delayed Heterogeneous Multiagent Systems in Competitive Networks

Lianghao Ji , Yue Zhang, and Yiliu Jiang

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Lianghao Ji; lianghao.ji@gmail.com

Received 19 June 2018; Accepted 6 August 2018; Published 3 October 2018

Academic Editor: Sing Kiong Nguang

Copyright © 2018 Lianghao Ji 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.

This paper discusses the couple-group consensus issues of a class of heterogeneous multiagent systems containing first-order and second-order dynamic agents under the influence of both input and communication delays. In distinction to the existing works, a novel distributed coordination control protocol is proposed which is not only on the foundation of the competitive interaction between the agents but also has no virtual velocity estimation in the first-order dynamics. Furthermore, without the restrictive assumptions existing commonly in the related works, several sufficient algebraic criteria are established for the heterogeneous systems to realize couple-group consensus asymptotically. The obtained conclusions show that the achievement of the systems' couple-group consensus intimately relates to the coupling weights between the agents, the systems control parameters, and the input time delays of the agents, while communication time delays between the agents are irrelevant to it. Finally, several simulations are illustrated to verify the effectiveness of the obtained theoretical results.

1. Introduction

As a fundamental issue of the coordinated control of multiagent systems (MASs), consensus problem of MASs has attracted wide attention in recent years due to its wide applications in various fields, including the mobile robot systems, distributed target tracking, and group decision-making. So far, lots of research works on various consensus problems have been constantly reported, such as in [1–5] and the refs. therein.

Group consensus, as an extension of consensus issue, implies that distinctive subconsensus states can be achieved in complex systems while separate subgroups cannot reach agreement. In the last few years, lots of research works on group consensus have widely emerged, such as in [6–14].

1.1. Related Works. Note that all the works mentioned above mainly focus on the MASs which constructed by multiple agents with homogeneous dynamics. Namely, all agents of the entire complex systems featured the identical dynamics. In fact, this case is very restricted because in the real world,

the difference existed in the dynamics among agents is unavoidable. Meanwhile, we usually have such applications using multiple agents with different dynamics to achieve the desired goals for the case of reducing control cost. Therefore, more and more concerns are devoted by the researchers to heterogeneous MASs' consensus problems, such as the [15], in which the necessary and sufficient criteria were proposed for the synchronization of the heterogeneous networks. Meanwhile, Wang et al. [16] also investigated the heterogeneous systems and addressed certain sufficient conditions to ensure the achievement of consensus. With regard of the time delays, Cui et al. discussed the consensus problem for the heterogeneous chaotic systems in [17]. Based on undirected topology, Goldin and Raisch addressed the consensus of the heterogeneous networks in [18]. In [19], Wang et al. studied the globally limited consensus of the heterogeneous systems and put forward certain criteria to guarantee the consensus. The consensus issues for the heterogeneous systems with time delays were also studied in [20, 21]. As for the discrete-time heterogeneous systems, Kim et al. [22] and Li et al. [23], respectively, discussed the consensus

problems for the systems with accidental connection failures. The second-order consensus for the systems with Euler-Lagrange networks was investigated in [24], where the multiple consensus states were achieved. In [25], the effective protocols for achieving consensus of the heterogeneous systems were designed and certain conditions based on the state transformation method were obtained. Based on certain hypothesis, Liu et al. also investigated the heterogeneous MASs and its group consensus problems in [26]. Considering the influence of input delays or not, Wen et al. studied the group consensus of the systems with heterogeneous dynamics [27, 28], respectively. By utilizing the Lyapunov method, in [29], Qin et al. studied both linear and nonlinear heterogeneous systems as well as their group consensus issues.

1.2. Primary Motivations. Note that most of the research works mentioned above have been paid attention to consensus problems of the heterogeneous MASs. As we know, group consensus owns an important practical significance in the coordinated control of large scale and complex tasks. Hence, many further works for group consensus of heterogeneous MASs need to be concerned with. Meanwhile, it is known that communication and input time delays existing in the complex systems can usually affect and even destroy the stability of the system. Therefore, one motivation of this paper is to discuss the group consensus problems of a class of heterogeneous MASs with time delays.

The main contributions of this paper are summarized in the following three aspects. First, to our knowledge, there are rare works on the group consensus issue of heterogeneous MASs which considered both input and communication delays. Second, from a different point of view, we propose an original couple-group consensus protocols which established on the agents' competitive interaction. It is distinguished with the aforementioned works which mainly modeled by the agents' cooperative relationship [15–28]. It is known that competitive interaction is also an important relationship in complex systems, such as in ecology, the problems of predator-prey on food chain. Meanwhile, in our control protocol, the first-order agents' dynamics possess no virtual velocity, which been included in many related works (e.g., in [20, 26–28]) for simplifying the process of analysis. Third, our results relax the following two conservative prerequisites existed in [20, 25–29]: in-degree balance and the geometric multiplicity of the zero eigenvalue of the systems' Laplacian matrix have no less than 2. Both of them limit the communication of the agents and the topology of the system. Based on matrix theory and frequency domain analysis, some criteria for ensuring the achievement of couple-group consensus are effectively proposed. The combination of the above innovative points makes the application scope of our results more general.

Notation. Throughout this paper, \mathbb{R} and \mathbb{C} indicate the real and complex numbers sets, respectively. $\forall z \in \mathbb{C}$, then its real part and modulus are defined as $\text{Re}(z)$ and $|z|$. I_N stands for the identity matrix with N -dimension. $\det(A)$ and $\lambda_i(A)$ mean the determinant and the i th eigenvalue of the matrix A , individually.

2. Preliminaries and Problem Statements

2.1. Graph Theory. Using the graph theory, the agents and their information exchange in a MAS with N agents can be represented by a digraph $G = (V, E, A)$, where $V = \{v_1, v_2, \dots, v_N\}$, $E \subseteq V \times V$, and $A = (a_{ij})_{N \times N} \in \mathbb{R}^{N \times N}$ denote the node set, the edge set, and the adjacency matrix, respectively. Noting that an undirected graph can be seen as a special digraph, we assume $a_{ij} > 0$ if $e_{ij} \in E$ throughout this paper. Namely, $a_{ij} > 0$ if and only if the node (agent) v_i can receive the information from the node (agent) v_j ; otherwise, $a_{ij} = 0$. Meanwhile, the neighbour sets and the in-degree of node i are defined as $N_i = \{j \in V : e_{ij} \in E\}$ and $D_i = \text{deg}_{\text{in}}(i) = \sum_{j=1}^N a_{ij}$, individually. Design in-degree matrix as $D = \text{diag}\{d_1, d_2, \dots, d_N\}$, so $L = D - A$ is defined to be the Laplacian matrix.

2.2. Problem Statement. In this paper, we suppose a heterogeneous multiagent system consisting of $n + m$ agents which contained first-order and second-order dynamics. For convenience, assume the first n and the remaining m agents own second-order and first-order dynamics, respectively, then the system can be described as follows:

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = u_i(t), \\ \quad i \in g_1, \\ \dot{x}_i(t) = u_i(t), \quad i \in g_2, \end{cases} \quad (1)$$

where $g_1 = \{1, 2, \dots, n\}$, $g_2 = \{n + 1, n + 2, \dots, n + m\}$, and $g = g_1 \cup g_2$. $x_i(t)$, $v_i(t)$, and $u_i(t) \in \mathbb{R}$ represent position, velocity, and input control of agent i , individually.

Regard a heterogeneous MAS, to each agent, its neighbor may has second-order and first-order dynamics, which may be distinguishingly denoted as $N_{i,s}$ and $N_{i,f}$. Thus, the neighbors set of agent i can be distinguished as $N_i = N_{i,f} \cup N_{i,s}$. With regard that the agents' dynamics in the systems are heterogeneous, then adjacency matrix A can be denoted as

$$A = \begin{bmatrix} A_s & A_{sf} \\ A_{fs} & A_f \end{bmatrix}, \quad (2)$$

where $A_s \in \mathbb{R}^{n \times n}$ and $A_f \in \mathbb{R}^{m \times m}$ represent the adjacency matrix consisting of all second-order or first-order agents, A_{fs} composes of coupling weights from first-order agents to second-order ones, and A_{sf} is the opposite of A_{fs} . The systems' Laplacian matrix can be modified as

$$L = D - A = \begin{bmatrix} L_s + D_{sf} & -A_{sf} \\ -A_{fs} & L_f + D_{fs} \end{bmatrix}, \quad (3)$$

in the Laplacian matrix L_s and L_f , the interactions of only second-order agents or first-order agents are included; $D_{sf} = \text{diag}\{\sum_{j \in N_{i,f}} a_{ij}, i \in g_1\}$ and $D_{fs} = \text{diag}\{\sum_{j \in N_{i,s}} a_{ij}, i \in g_2\}$

are in-degree matrix of agent i , which includes information received from the neighbors of different orders.

Firstly, some fundamental definitions and lemma are introduced as below.

Definition 1. To a heterogeneous MASs, such as (1), which can be said to achieve couple-group consensus asymptotically when and only when the following two conditions are satisfied:

$$\begin{aligned} \lim_{t \rightarrow +\infty} \|x_i(t) - x_j(t)\| &= 0, \quad \text{if } i, j \in g_k, k = 1, 2, \\ \lim_{t \rightarrow +\infty} \|v_i(t) - v_j(t)\| &= 0, \quad \text{if } i, j \in g_k, k = 1. \end{aligned} \quad (4)$$

Definition 2. For a bipartite graph $G = (V, E)$, where E and V denote its edge and vertex sets, it has the following

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = \sum_{j \in g_1} a_{ij} [x_j(t - \tau) - x_i(t - \tau)] + \sum_{j \in g_2} a_{ij} x_j(t - \tau) + \sum_{j \in g_1} a_{ij} [v_j(t - \tau) - v_i(t - \tau)] + \sum_{j \in g_2} a_{ij} v_j(t - \tau), \end{cases} \quad i \in g_1, \quad (5)$$

$$\begin{cases} \dot{x}_i(t) = v_i(t - \tau) + \sum_{j \in g_2} a_{ij} [x_j(t - \tau) - x_i(t - \tau)] + \sum_{j \in g_1} a_{ij} x_j(t - \tau), \\ \dot{v}_i(t) = \sum_{j \in g_2} a_{ij} [x_j(t) - x_i(t)] + \sum_{j \in g_1} a_{ij} x_j(t), \end{cases} \quad i \in g_2. \quad (6)$$

For the sake of analysis, the velocity estimation is added to the first-order agents in (6). Meanwhile, the systems (5) and (6) are modeled by the cooperative relationship among the agents.

To realize group consensus and be distinguished to (5) and (6), we design an original distributed control protocol which utilizes the agents' competitive interactions. It is showed as follows:

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = -\alpha \left[\sum_{j \in N_i} a_{ij} [x_j(t - \tau_{ij}) + x_i(t - \tau)] \right] - \beta v_i(t - \tau), \\ \dot{x}_i(t) = -\gamma \left[\sum_{j \in N_i} a_{ij} [x_j(t - \tau_{ij}) + x_i(t - \tau)] \right], \end{cases} \quad i \in g_1, \quad (7)$$

where τ_{ij} indicates communication delay between agent j and agent i and τ represents the identical input delay of the agents. Meanwhile, we suppose the control parameters α , β , and $\gamma > 0$.

two properties: (1) $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$ and (2) $\forall e = (p, q) \in E$, where $p \in V_1$ and $q \in V_2$.

Lemma 1 [10]. Regarding a directed bipartite graph which contains a directed spanning tree, it then possesses the following two properties: $\text{rank}(L) = n - 1$ and $\text{Re}(\lambda_i(L)) > 0$ when $\lambda_i(L) \neq 0$. Meanwhile, for a undirected bipartite graph, $\lambda_i(L) \in \mathbb{R}$. Where n is the number of agents of systems, matrix $L = D + A$.

3. Main Results

In [18], the authors investigated the group consensus of the heterogeneous systems with identical input time delay. The systems are described as follows:

Theorem 1. Consider the MASs (7) and suppose the systems' topology is an undirected bipartite graph, the systems' couple-group consensus can be realized asymptotically if these conditions hold: $\beta^2 > 2\alpha D_i$ and $\tau \in [1/2\beta, 1/2\gamma \max\{\tilde{D}_i\}]$, where $D_i = \sum_{j \in N_i} a_{ij}$, $i \in g_1$ and $\tilde{D}_i = \sum_{j \in N_i} a_{ij}$, $i \in g_2$.

Proof 1. Do Laplace transforms to (7) firstly, we get

$$\begin{cases} sx_i(s) = v_i(s), \\ sv_i(s) = -\alpha \left[\sum_{j \in N_i} a_{ij} [e^{-\tau_{ij}s} x_j(s) + e^{-\tau s} x_i(s)] \right] - \beta e^{-\tau s} v_i(s), \end{cases} \quad i \in g_1, \quad (8)$$

$$sx_i(s) = -\gamma \left[\sum_{j \in N_i} a_{ij} [e^{-\tau_{ij}s} x_j(s) + e^{-\tau s} x_i(s)] \right], \quad i \in g_2, \quad (9)$$

where $x_i(s)$ and $v_i(s)$ represent the Laplace transform of $x_i(t)$ and $v_i(t)$, respectively.

From the (8), we have

$$s^2 x_i(s) = -\alpha \left[\sum_{j \in N_i} a_{ij} [e^{-\tau_{ij}s} x_j(s) + e^{-\tau s} x_i(s)] \right] - \beta s e^{-\tau s} x_i(s), \quad i \in g_1. \quad (10)$$

After a simple manipulation, it yields that

$$s x_i(s) = \frac{-s^2 x_i(s) - \alpha [\sum_{j \in N_i} a_{ij} [e^{-\tau_{ij}s} x_j(s) + e^{-\tau s} x_i(s)]]}{\beta e^{-\tau s}}, \quad i \in g_1. \quad (11)$$

Define $x_s(s) = [x_1(s), x_2(s), \dots, x_n(s)]^T$, $x_f(s) = [x_{n+1}(s), x_{n+2}(s), \dots, x_{n+m}(s)]^T$, and

$$\hat{L} = (\hat{l}_{ij})_{(n+m) \times (n+m)} = \begin{cases} e^{-\tau_{ij}s} a_{ij}, & i \neq j, \\ \sum_{j \in N_i} a_{ij} e^{-\tau s}, & i = j, \end{cases} \quad (12)$$

then from the (9) and (11), we have

$$s x_s(s) = \frac{[-s^2 x_s(s) - \alpha (\hat{L}_s + \hat{D}_{sf}) x_s(s) - \alpha \hat{A}_{sf} x_f(s)]}{\beta e^{-\tau s}}, \quad (13)$$

$$s x_f(s) = -\gamma \hat{A}_{fs} x_s(s) - \gamma (\hat{L}_f + \hat{D}_{fs}) x_f(s).$$

Define $y(s) = [x_s^T(s), x_f^T(s)]^T$, then (13) can be rewritten as

$$s y(s) = \tilde{\Psi}(s) y(s), \quad (14)$$

where

$$\tilde{\Psi}(s) = \begin{bmatrix} \frac{-s^2 I_n - \alpha (\hat{L}_s + \hat{D}_{sf})}{\beta e^{-\tau s}} & \frac{-\alpha \hat{A}_{sf}}{\beta e^{-\tau s}} \\ -\gamma \hat{A}_{fs} & -\gamma (\hat{L}_f + \hat{D}_{fs}) \end{bmatrix}. \quad (15)$$

Define $\tilde{\Gamma}(s) = \det(sI - \tilde{\Psi}(s))$. Based on the Lyapunov stability criterion, the systems can reach group consensus if the roots of $\tilde{\Gamma}(s)$ are at $s=0$ or $\text{Re}(\lambda_i(\tilde{\Gamma}(s))) < 0$. Subsequently, we will discuss these two cases, respectively, according to the general Nyquist criterion.

When $s=0$, $\tilde{\Gamma}(0) = \det(D+A)(-\alpha/\beta)^n (-\gamma)^m$. According to Lemma 1, one can know that zero is the simple eigenvalue of the matrix $(D+A)$. Thus, the roots of $\tilde{\Gamma}(0)$ are at the point $s=0$.

When $s \neq 0$, set $\tilde{\Gamma}(s) = \det(\Phi(s) + I)$ and

$$\Phi(s) = \begin{bmatrix} \frac{s^2 I_n + \alpha (\hat{L}_s + \hat{D}_{sf})}{s \beta e^{-\tau s}} & \frac{\alpha \hat{A}_{sf}}{s \beta e^{-\tau s}} \\ \frac{\gamma \hat{A}_{fs}}{s} & \frac{\gamma (\hat{L}_f + \hat{D}_{fs})}{s} \end{bmatrix}. \quad (16)$$

Set $s = j\omega$, according to the general Nyquist criterion, when and only when the point $(-1, j0)$ is not enclosed by the Nyquist curve of $\Phi(j\omega)$; $\tilde{\Gamma}(s)$'s roots are located in the open left-half plane of the complex field. In other words, the group consensus can be reached in this situation. By the Gerschgorin disk theorem, one can obtain

$$\lambda(\Phi(j\omega)) \in \{\Phi_i, i \in g_1\} \cup \{\Phi_i, i \in g_2\}. \quad (17)$$

When $i \in g_1$, it follows

$$\Phi_i = \left\{ x : x \in \left\| x - \frac{\alpha}{j\omega\beta} \sum_{j \in N_i} a_{ij} - \frac{j\omega}{\beta} e^{j\omega\tau} \right\| \leq \sum_{j \in N_i} \left| \frac{\alpha a_{ij}}{j\omega\beta} e^{-j\omega(\tau_{ij}-\tau)} \right| \right\}. \quad (18)$$

Let $\sum_{j \in N_i} a_{ij} = D_i$, $i \in g_1$. Since the point $(-a, j0)$, $a \geq 1$ cannot be encircled in Φ_i , $i \in g_1$, then the following inequality is obtained:

$$\left| -a - \frac{\alpha D_i}{j\omega\beta} - \frac{j\omega}{\beta} e^{j\omega\tau} \right| > \sum_{j \in N_i} \left| \frac{\alpha a_{ij}}{j\omega\beta} e^{-j\omega(\tau_{ij}-\tau)} \right|. \quad (19)$$

Based on the Euler formula and from (19), we have

$$\left| -a + \frac{\alpha D_i}{\omega\beta} j - \frac{j\omega}{\beta} (\cos \omega\tau + j \sin \omega\tau) \right| > \left| \frac{\alpha D_i}{j\omega\beta} (\cos(\tau_{ij} - \tau) - j \sin(\tau_{ij} - \tau)) \right|. \quad (20)$$

Thus, we can get

$$a^2 - \frac{2a\omega}{\beta} \sin \omega\tau + \frac{\omega^2}{\beta^2} - \frac{2\alpha D_i}{\beta^2} \cos \omega\tau > 0. \quad (21)$$

It is easy to know that $a^2 - (2a\omega/\beta) \sin \omega\tau$ is monotonically increasing for $a \geq 1$. Thus, we obtain

$$1 - \frac{2\omega}{\beta} \sin \omega\tau + \frac{\omega^2}{\beta^2} - \frac{2\alpha D_i}{\beta^2} \cos \omega\tau > 0. \quad (22)$$

Here, as the control parameter β is positive, the following inequation can be derived:

$$\beta^2 - 2\omega\beta \sin \omega\tau + \omega^2 - 2\alpha D_i \cos \omega\tau > 0. \quad (23)$$

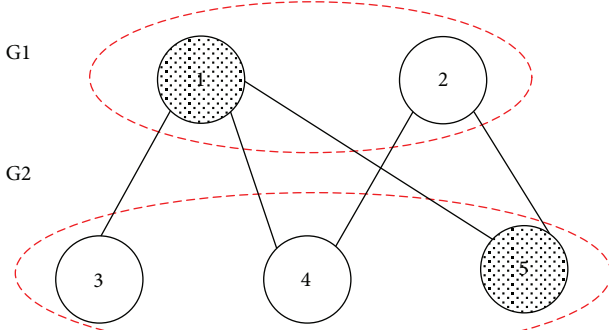


FIGURE 1: The undirected bipartite topology.

conditions hold: $\beta^2 > 2\alpha D_i$ and if $i \in g_1$, $\tau_i \in [0, (1/2\beta)]$; otherwise, $\tau_i \in [0, 1/2\gamma\tilde{D}_i]$, $i \in g_2$, where $D_i = \sum_{j \in N_i} a_{ij}$, $i \in g_1$ and $\tilde{D}_i = \sum_{j \in N_i} a_{ij}$, $i \in g_2$.

The proof is omitted here due to the limitation of space.

Remark 4. The results in Theorem 2 show that the upper bound of the input time delay will vary with the different dynamics of the agents and decided by the control parameter and coupling weights between the agents with same dynamic. Similarly, communication delays between agents have no effect on the achievement of systems' couple-group consensus.

Remark 5. The system topologies we considered in this paper are either an undirected bipartite graph or a directed bipartite graph containing a directed spanning tree. They seem to be a specific topology. But as a matter of fact, the systems' group consensus can usually be reached with the help of some stronger conditions. For instance, in [20, 25–29], the systems' topology is also an undirected graph or a digraph containing a spanning tree. At the same time, for the purpose of realizing the group consensus, some extra assumptions are needed, which have been mentioned in Remarks 1 and 2. It will be a challenge work to investigate the group consensus problem of a class of heterogeneous MASs in a more general condition.

4. Simulation

In this section, the correctness of the obtained results will be illustrated by several simulations.

Regard the heterogeneous systems with bipartite topology as Figure 1. Hereinto, agents 1 and 2 and 3, 4, and 5 belong to the subgroups G_1 and G_2 , individually. For the generality, agents 1 and 5 are represented as second-order agents and the first-order agents include the remaining agents 2, 3, and 4. Therefore, each subgroup in Figure 1 is designed as heterogeneous.

Remark 6. Distinguish from the cases in [24–26, 28], the dynamics of the agents in this paper are not restricted to be homogeneous within the same subgroup. Obviously, that is the special case of ours.

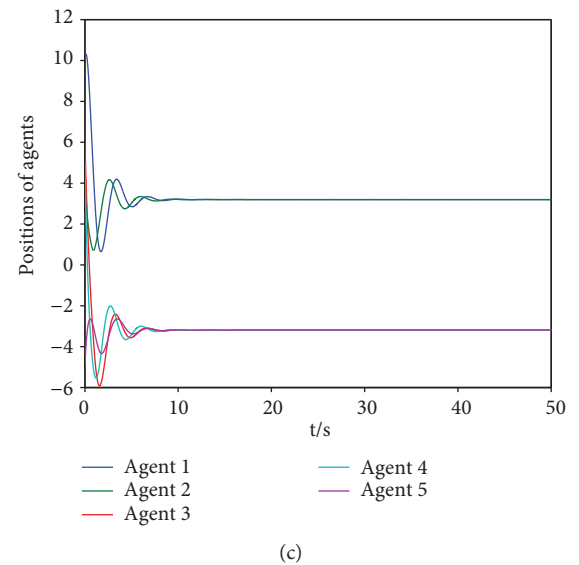
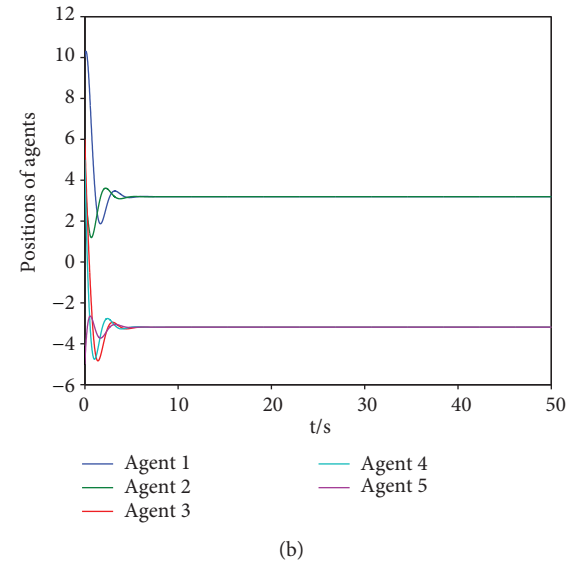
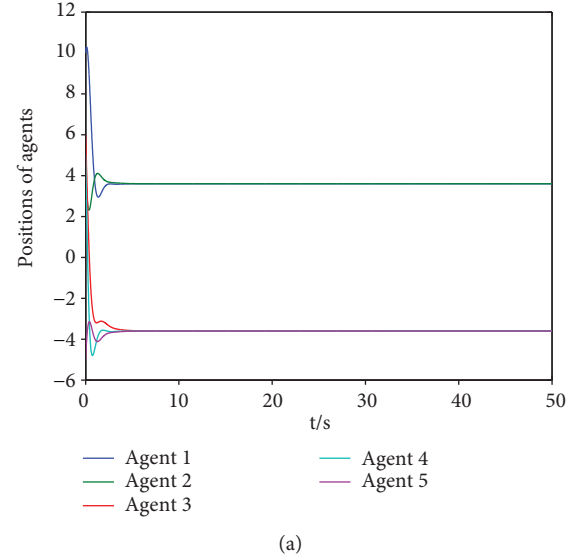
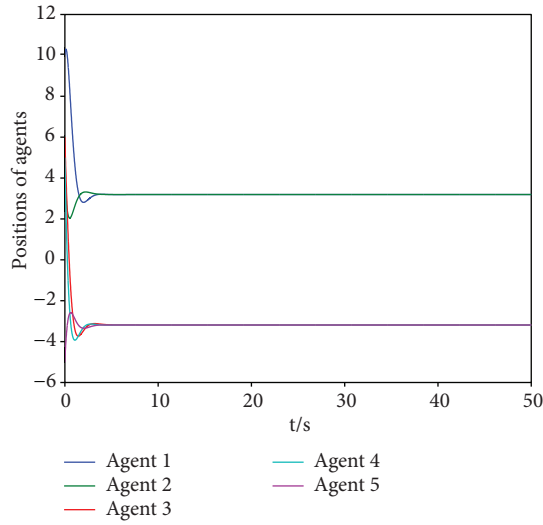
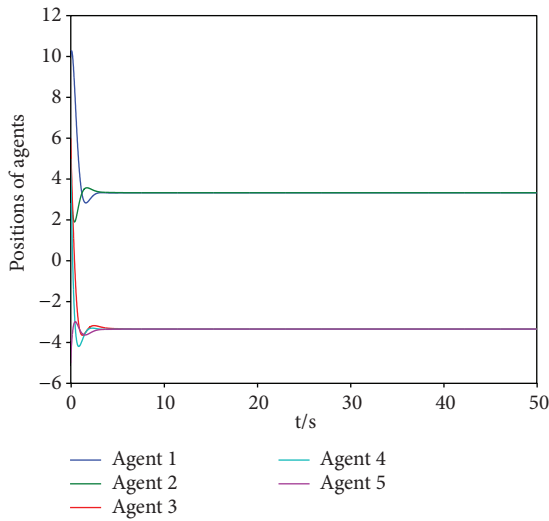


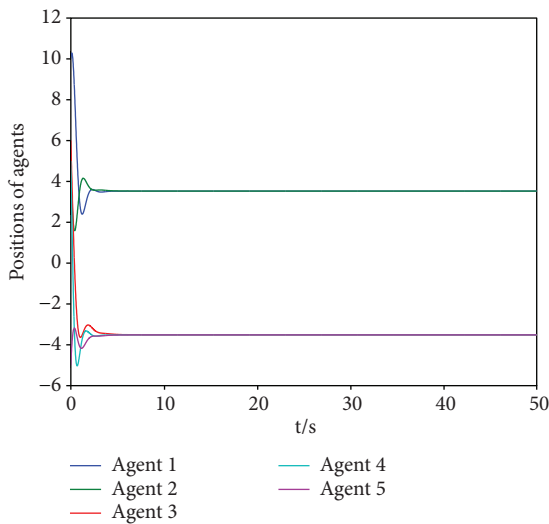
FIGURE 2: The agents' position trajectories, where $\tau = 0.15$. (a) $\tau_{ij} = 0$, (b) $\tau_{ij} = 0.5$, and (c) $\tau_{ij} = 0.8$.



(a)

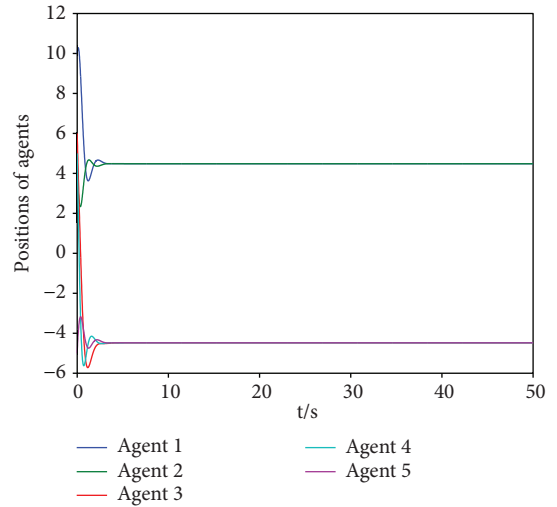


(b)

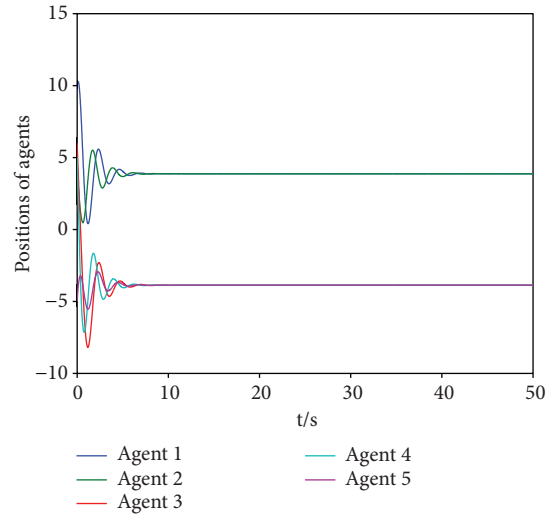


(c)

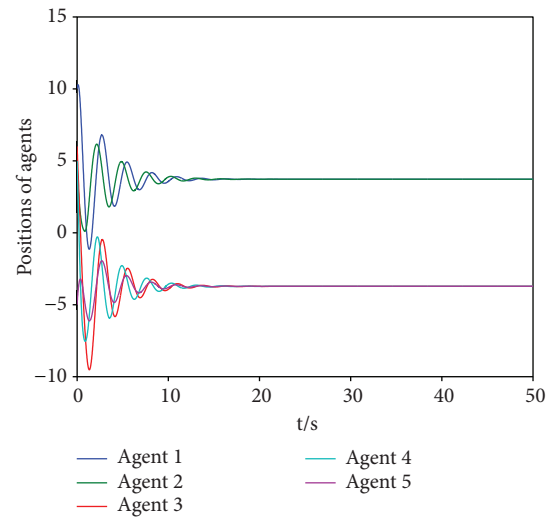
FIGURE 3: The agents' position trajectories, where $\tau_{ij} = 0.15$. (a) $\tau = 0$, (b) $\tau = 0.1$, and (c) $\tau = 0.15$.



(a)



(b)



(c)

FIGURE 4: The agents' position trajectories, where $\tau_1 = 0.15$, $\tau_2 = 0.2$, $\tau_3 = 0.5$, $\tau_4 = 0.25$, and $\tau_5 = 0.15$. (a) $\tau_{ij} = 0$, (b) $\tau_{ij} = 0.5$, and (c) $\tau_{ij} = 0.8$.

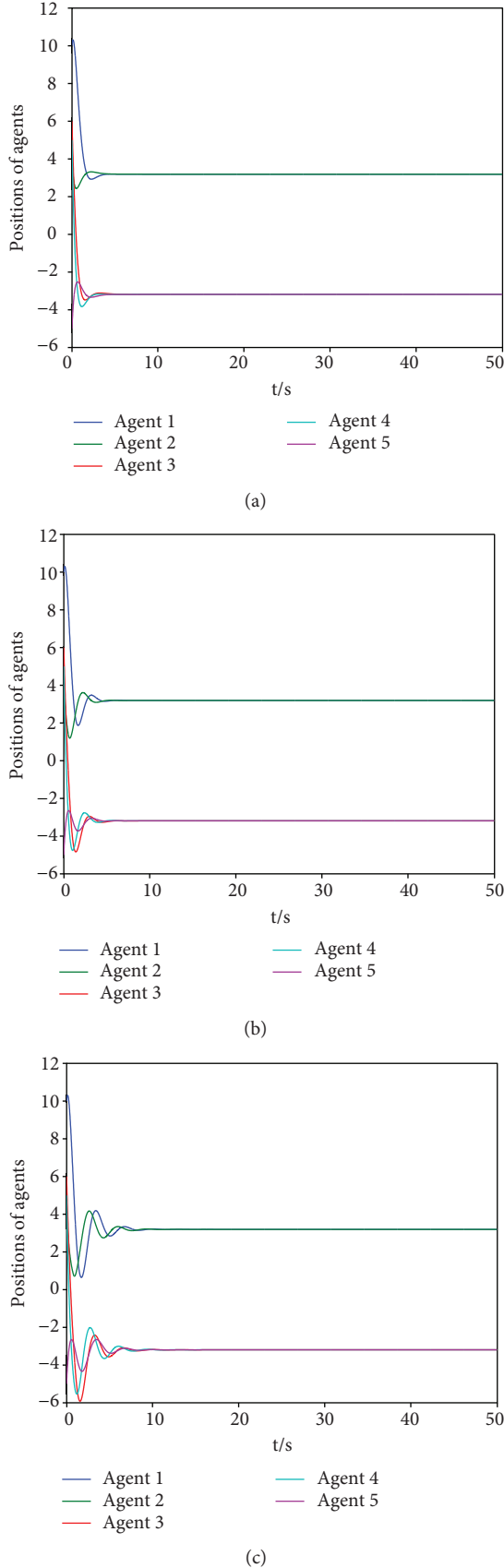


FIGURE 5: The agents' position trajectories, where $\tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5 = 0$. (a) $\tau_{ij} = 0$, (b) $\tau_{ij} = 0.5$, and (c) $\tau_{ij} = 0.8$.

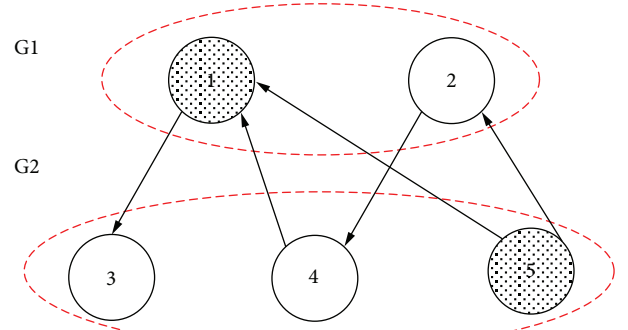


FIGURE 6: The directed bipartite topology.

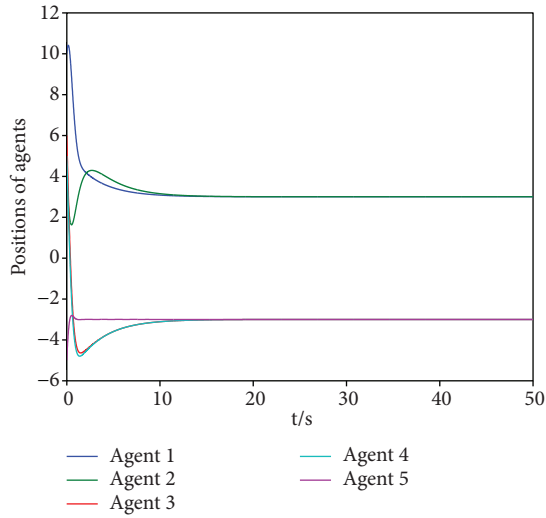
For simplicity, we set $a_{ij} = 1, i, j \in [1, 5]$ and select $\alpha = 1$, $\beta = 3$, and $\gamma = 1$. From the systems' topology shown in Figure 1, it is easy to know that $d_1 = 3$, $d_2 = 2$, $d_3 = 1$, $d_4 = 2$, and $d_5 = 2$.

Example 1. According to the group consensus criteria proposed in Theorem 1, the range of input time delay is calculated as $\tau \in [0, \min \{1/6, 1/4\}]$. In this case, we select the identical input time delay as $\tau = 0.15$ s. Here, the conditions of Theorem 1 are now all satisfied. From the results in Theorem 1, we know that communication delay does not affect the systems' couple-group consensus. Hence, for convenience, their values are set the same. Meanwhile, several different cases are considered to discover the impact of the delays including the communication and input delays on the systems' convergence rate. Figures 2 and 3 show the trajectories of the agents in the system (7). They indicate that all agents of the heterogeneous system converge to two reverse subgroups, that is, the couple-group consensus is realized.

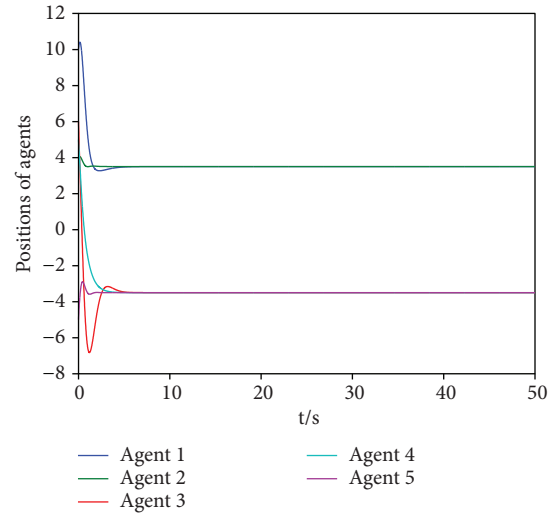
Remark 7. The results in Figures 2 and 3 reveal that the communication and input time delays are both capable to affect the trajectories of the agents. To illustrate, the shorter the delays, the faster the system converges. Hence, we can improve the convergence rate by reducing the time delays including communication time delays, the input ones, or both of them.

Example 2. From Theorem 2 and according to the given parameters, it has $\tau_i \leq 1/6, i \in \sigma_1, \tau_3 \leq 0.5, \tau_4 \leq 0.25$, and $\tau_5 \leq 0.25$. In the simulation, we choose $\tau_1 = 0.15$ s, $\tau_2 = 0.2$ s, $\tau_3 = 0.5$ s, $\tau_4 = 0.25$ s, and $\tau_5 = 0.15$ s. Clearly, the conditions in Theorem 2 are all hold. Similar to the Example 1, we also set several sets of time delays, then the agents' trajectories in the systems (30) under the influence of different input and communication time delays are illustrated as Figures 4 and 5. From them, we know that the couple-group consensus of each case is realized, respectively.

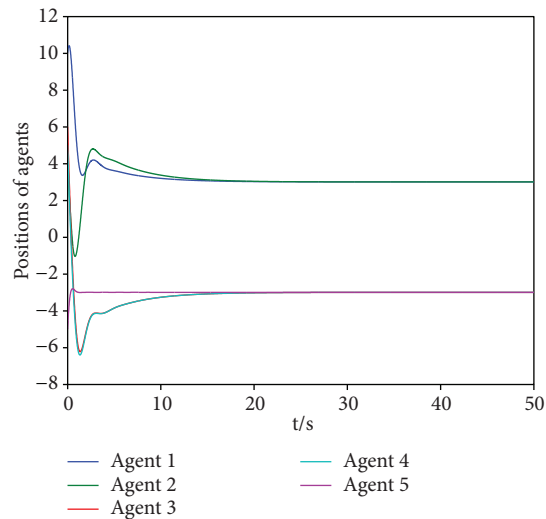
Example 3. Suppose the topology of the heterogeneous systems (7) is shown in Figure 6, which contains a directed spanning tree. We set the input time delays as $\tau = 0.15$ s that satisfies all the conditions proposed in Corollary 1.



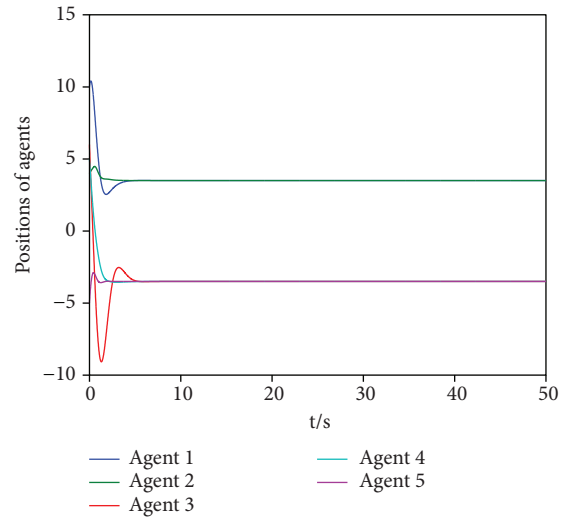
(a)



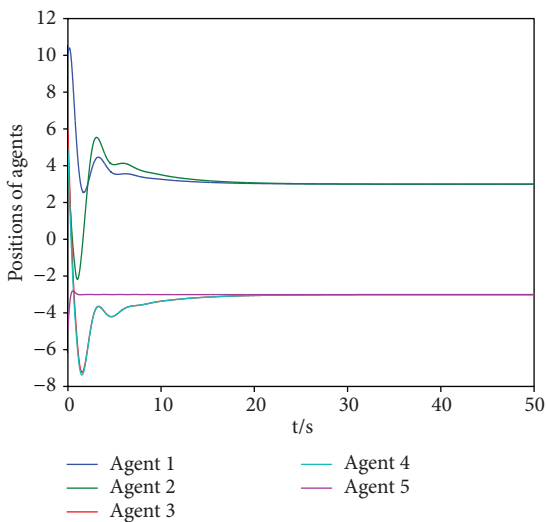
(a)



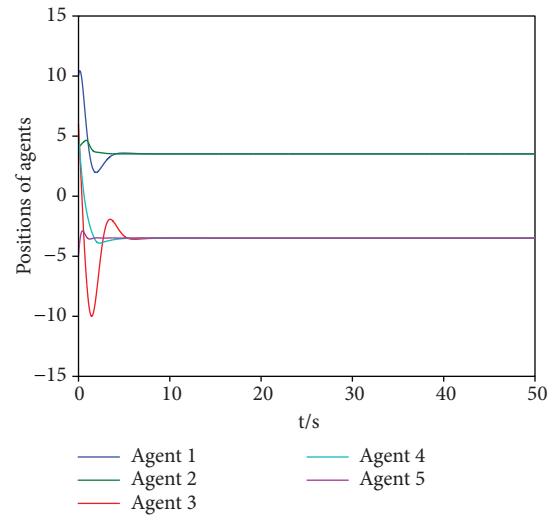
(b)



(b)



(c)



(c)

FIGURE 7: The agents' position trajectories, where $\tau = 0.15$. (a) $\tau_{ij} = 0$, (b) $\tau_{ij} = 0.5$, and (c) $\tau_{ij} = 0.8$ s.

FIGURE 8: The agents' position trajectories, where $\tau_1 = 0.15$, $\tau_2 = 0.2$, $\tau_3 = 0.5$, $\tau_4 = 0.25$, and $\tau_5 = 0.15$. (a) $\tau_{ij} = 0$, (b) $\tau_{ij} = 0.5$, and (c) $\tau_{ij} = 0.8$.

According to the trajectories of the agents shown in Figure 7, one can know that the couple-group consensus is achieved asymptotically. Similarly, regard Figure 6 as the topology of the systems (30). And set $\tau_1 = 0.15$ s, $\tau_2 = 0.2$ s, $\tau_3 = 0.5$ s, $\tau_4 = 0.25$ s, and $\tau_5 = 0.15$ s that follow the conditions in Corollary 2. From the trajectories of the agents shown in Figure 8, the couple-group consensus with the influence of different input time delays is also achieved.

5. Conclusion

Considering the influence of both communication and input delays, an original distributed coordination control protocol based on agents' competitive interaction is designed for realizing the consensus of heterogeneous systems. By utilizing matrix theory and general Nyquist criteria, some sufficient criteria as well as the input delays' upper bound are theoretically presented. From the results, we find that the control parameters of systems, the coupling weights between the agents, and the input delay play key roles on the achievement of couple-group consensus of the heterogeneous systems, while communication delays are independent with it. Furthermore, the simulation results show that the systems' convergence rate can be improved by reducing the time delays including communication, input, or both of them. In the future, this issues for the heterogeneous and/or hybrid MASs with switching interaction topology can be extended.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61572091, 61673080, and 61876200, the Natural Science Foundation Project of Chongqing Science and Technology Commission under Grant no. cstc2018jcyjAX0112, the Key Theme Special Project of Chongqing Science and Technology Commission under Grant nos. cstc2017zdcy-zdyfx0091 and cstc2017rgzn-zdyfx0022, and the Doctoral Start-up Foundation of Chongqing University of Posts and Telecommunications under Grant A2014-20.

References

- [1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [2] F. Xiao, T. Chen, and H. Gao, "Synchronous hybrid event- and time-driven consensus in multiagent networks with time delays," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1165–1174, 2016.
- [3] H. Li, G. Chen, T. Huang, Z. Dong, W. Zhu, and L. Gao, "Event-triggered distributed average consensus over directed digital networks with limited communication bandwidth," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3098–3110, 2016.
- [4] Y. Chen, H. Dong, J. Lu, X. Sun, and K. Liu, "Robust consensus of nonlinear multiagent systems with switching topology and bounded noises," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1276–1285, 2016.
- [5] X. Wu, Y. Tang, J. Cao, and W. Zhang, "Distributed consensus of stochastic delayed multi-agent systems under asynchronous switching," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1817–1827, 2016.
- [6] J. Yu and L. Wang, "Group consensus in multi-agent systems with switching topologies and communication delays," *Systems & Control Letters*, vol. 59, no. 6, pp. 340–348, 2010.
- [7] Y. Han, W. Lu, and T. Chen, "Cluster consensus in discrete-time networks of multiagents with inter-cluster nonidentical inputs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 566–578, 2013.
- [8] L. Ji, Q. Liu, and X. Liao, "On reaching group consensus for linearly coupled multi-agent networks," *Information Sciences*, vol. 287, pp. 1–12, 2014.
- [9] X. Liao and L. Ji, "On pinning group consensus for dynamical multi-agent networks with general connected topology," *Neurocomputing*, vol. 135, pp. 262–267, 2014.
- [10] Q. Wang and Y. Wang, "Cluster synchronization of a class of multi-agent systems with a bipartite graph topology," *Science China Information Sciences*, vol. 57, no. 1, pp. 1–11, 2014.
- [11] Y. Han, W. Lu, and T. Chen, "Achieving cluster consensus in continuous-time networks of multi-agents with inter-cluster non-identical inputs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 793–798, 2015.
- [12] Y. Shang, "Group consensus of multi-agent systems in directed networks with noises and time delays," *International Journal of Systems Science*, vol. 46, no. 14, pp. 2481–2492, 2015.
- [13] W. Xu and D. W. C. Ho, "Clustered event-triggered consensus analysis: an impulsive framework," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7133–7143, 2016.
- [14] H.-x. Hu, W. Yu, G. Wen, Q. Xuan, and J. Cao, "Reverse group consensus of multi-agent systems in the cooperation-competition network," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2036–2047, 2016.
- [15] J. Lunze, "Synchronization of heterogeneous agents," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2885–2890, 2012.
- [16] L. Wang, M. Z. Q. Chen, and Q.-G. Wang, "Bounded synchronization of a heterogeneous complex switched network," *Automatica*, vol. 56, pp. 19–24, 2015.
- [17] B. Cui, T. Ma, Y. Song, C. Feng, F. L. Lewis, and C. Zhao, "Distributed adaptive consensus control of heterogeneous multi-agent chaotic systems with unknown time delays," *IET Control Theory & Applications*, vol. 9, no. 16, pp. 2414–2422, 2015.
- [18] D. Goldin and J. Raisch, "Consensus for agents with double integrator dynamics in heterogeneous networks," *Asian Journal of Control*, vol. 16, no. 1, pp. 30–39, 2014.
- [19] L. Wang, M. Z. Q. Chen, W.-j. Feng, and Q.-g. Wang, "Global bounded consensus in heterogeneous multi-agent systems with directed communication graph," *IET Control Theory & Applications*, vol. 9, no. 1, pp. 147–152, 2015.

- [20] C.-L. Liu and F. Liu, "Dynamical consensus seeking of heterogeneous multi-agent systems under input delays," *International Journal of Communication Systems*, vol. 26, no. 10, pp. 1243–1258, 2013.
- [21] Y.-J. Sun, G.-l. Zhang, and J. Zeng, "Consensus analysis for a class of heterogeneous multiagent systems with time delay based on frequency domain method," *Mathematical Problems in Engineering*, vol. 2014, Article ID 248684, 7 pages, 2014.
- [22] J. M. Kim, Y. H. Choi, and J. B. Park, "Leaderless and leader-following consensus for heterogeneous multi-agent systems with random link failures," *IET Control Theory & Applications*, vol. 8, no. 1, pp. 51–60, 2014.
- [23] S. Li, G. Feng, X. Luo, and X. Guan, "Output consensus of heterogeneous linear discrete-time multiagent systems with structural uncertainties," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2868–2879, 2015.
- [24] H. Hu, W. Yu, Q. Xuan, C. Zhang, and G. Xie, "Group consensus for heterogeneous multi-agent systems with parametric uncertainties," *Neurocomputing*, vol. 142, pp. 383–392, 2014.
- [25] Y. Zheng and L. Wang, "A novel group consensus protocol for heterogeneous multi-agent systems," *International Journal of Control*, vol. 88, no. 11, pp. 2347–2353, 2015.
- [26] C. Liu, Q. Zhou, and X. Hu, "Group consensus of heterogeneous multi-agent systems with fixed topologies," *International Journal of Intelligent Computing and Cybernetics*, vol. 8, no. 4, pp. 294–311, 2015.
- [27] G. Wen, J. Huang, C. Wang, Z. Chen, and Z. Peng, "Group consensus control for heterogeneous multi-agent systems with fixed and switching topologies," *International Journal of Control*, vol. 89, no. 2, pp. 259–269, 2016.
- [28] G. Wen, Y. Yu, Z. Peng, and H. Wang, "Dynamical group consensus of heterogeneous multi-agent systems with input time delays," *Neurocomputing*, vol. 175, pp. 278–286, 2016.
- [29] J. Qin, Q. Ma, H. Gao, Y. Shi, and Y. Kang, "On group synchronization for interacting clusters of heterogeneous systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4122–4133, 2017.

Research Article

Optimal Allocation Method of Discrete Manufacturing Resources for Demand Coordination between Suppliers and Customers in a Fuzzy Environment

Wei Xu  and Yinyun Yu 

School of Management, Shenyang University of Technology, No. 111, Shenliao West Road, Economic & Technological Development Zone, Shenyang 110870, China

Correspondence should be addressed to Yinyun Yu; 3035905183@qq.com

Received 27 April 2018; Accepted 24 June 2018; Published 29 August 2018

Academic Editor: Sing Kiong Nguang

Copyright © 2018 Wei Xu and Yinyun Yu. 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.

Discrete manufacturing products are often assembled from multiple parts through a series of discrete processes. How to effectively configure resources in a discrete manufacturing environment is an important research topic worthy of attention. Based on an in-depth analysis of the discrete manufacturing operation model and the manufacturing resource allocation process, this paper fully considers the uncertainty factors of the manufacturing resource customers and the interests of the manufacturing resource suppliers and proposes a bilevel planning model under a fuzzy environment that comprehensively considers the customers' expectation bias and the suppliers' profit maximization. The method firstly uses a language phrase to collect the language evaluation of the customers and suppliers for manufacturing tasks and uses a trapezoidal fuzzy number to convert the language evaluation phrase into a value that can be calculated. Then, we use the prospect theory to optimize the constraint indicators based on the language evaluation of customers and suppliers. Next, the bilevel planning model for optimal configuration of manufacturing resources in discrete manufacturing environment is established under the consideration of the respective interests of both the customers and the suppliers, and the fast nondominated sorting genetic algorithm (NSGA-II) is used to solve the model. Finally, an example is given to verify the validity and feasibility of the model.

1. Introduction

With the rapid economic growth, customer demand for products has become more diverse. How to effectively grasp customer demand, shorten lead time, lower production cost, and increase product quality are key factors for companies to achieve sustainable development [1]. Due to unpredictable market changes, this requires the manufacturing systems to be able to rapidly reconstruct in response to rapid market changes. Discrete manufacturing has gradually become the mainstream model of manufacturing industry because of its advantages such as noncontinuity and reconfigurability [2]. With lots of manufacturing units that provide the same functionality but have different parameters in a discrete system, effective resource configuration which can reflect customer needs is often considered as a key technology [3].

Manufacturing resource configuration (MRC) plays a very important role in discrete systems, especially when manufacturing systems have to cope with shorter product life cycles [4]. In order to fulfill the dynamic customer needs, it always needs discrete systems to invoke several manufacturing units in sequence and combine them together fast. Due to complexity and diversity of manufacturing resources, resource optimal configuration has become a key issue in discrete systems and has been widely studied in both industrial community and academia. Despite of significant progress achieved by the researchers in manufacturing resource configuration, grey relational analysis [5], manufacturing grids [6], the idea of Pareto [7], graph theoretic methods [8], artificial intelligence-based methods [9], and other methods have been proposed successively. Most of current MRC algorithms where the data of MRC are in the form of

real number are not suitable for discrete systems environment because the MRC of discrete systems is often fuzzy and uncertain. For example, when MRD describe the quality of products, they can better express their customers' perception by using language evaluation words of "good," "bad," and so on. Therefore, this paper proposes a resource optimization configuration that considers the customers' expectation bias and considers the suppliers' profit maximization under fuzzy environment.

The remainder of this paper is organized as follows. After reviewing the related literature in Literature Reviews, some basic theories such as prospect theory and bilevel programming model are introduced in Preliminary Knowledge. Problem Description and Symbol Introduction presents the issues of the thesis research and some basic symbols. In The Proposed Method, it proposes a novel method for discrete manufacturing resource optimization configuration under fuzzy environment. An illustrative example is given to demonstrate the use of the proposed method in Illustrative Example. Conclusions and discussions are drawn with brief comments in Conclusions and Discussion.

2. Literature Reviews

MRC has emerged because of the need for manufacturing organizations to cope with shorter product life cycles, time-to-market, and a shift to respond to demands for MRD [10]. In the past years, many approaches, models, and methodologies have been proposed for solving manufacturing resource configuration problems. Among them, the main research algorithms are as follows:

- (i) Grey relational analysis
- (ii) Manufacturing grid
- (iii) The idea of Pareto
- (iv) Graph theoretic methods
- (v) Artificial intelligence-based methods

Zhang used grey relational analysis to further study manufacturing machine and manufacturing cell (MC) of multigranularity resource configuration process. During resource modeling, advanced information and sensor technologies are adopted to construct the information models of resources, which make the traditional production process more transparent, traceable, and on-line controllable [5]. Based on the quantum evolution theory, Zhang and Hu proposed a hybrid chaotic quantum evolutionary algorithm (CQEA) for resource combinatorial optimization (RCO) problems. Using an example to prove the proposed CQEA is effective, efficient, and scalable for the RCO problem in manufacturing grid system [6]. Li et al. gave a resource configuration method based on binary decision diagram (BDD) which is a directed acyclic graph (DAG) based on Shannon's decomposition. This method extends the scale of the reliability system. Through the results of three case studies, it is found that the decision graph expansion method is more computationally efficient than the traditional BDD

[7]. Xiang et al. introduce a new multiobjective optimization algorithm based on the combination of the idea of Pareto solution and group leader algorithm (GLA), which study of quality of service (QoS) and energy consumption assessment (EnCon) [8]. Tao et al. gave a parallel intelligent algorithm of resource configuration, which can minimize implementation time and cost and maximize the reliability of MGrid resource service composition paths [9].

Although there are many research results on the optimal configuration of resources, there are two obvious shortcomings in the current research results. First of all, the existing research results can promote and facilitate quicker and smarter decisions for service composition, but it cannot play any role in the fuzzy problem in the discrete system. That is, most of the existing research methods use real number, but the evaluation of manufacturing resources in discrete systems is often fuzzy and uncertain. Secondly, most of the existing research results are single objective, which can only consider the interests of one of the MRD or the MRP, and there are a few multiobjective research results, which consider the interests of both the MRD and the MRP, but when modeling, the multiobjective is converted into a single objective by using the weighted operator, and it is not truly multiobjective.

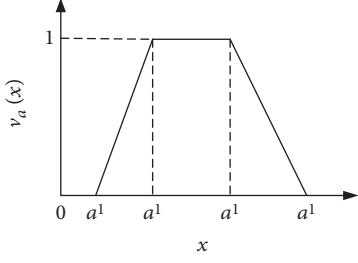
This paper proposes a bilevel programming model considering customer expectation under a fuzzy environment. Based on the prospect theory, we calculate the expected deviations of cost, quality, time, and green indicators of manufacturing resource customers and establish the objective function, then considering the efficiency, coordination, agility indicators of the manufacturing resource suppliers, we establish the objective function. Based on bilevel programming theory, the objective function and constraint conditions of MRD and MRP are established, respectively, and the fast nondominated sorting genetic algorithm (NSGA-II) is used to solve the model.

3. Preliminary Knowledge

3.1. Trapezoidal Fuzzy Number and Language Evaluation Phrases

3.1.1. Trapezoidal Fuzzy Numbers. Let \tilde{A} be a fuzzy set, and a fuzzy subset a of \tilde{A} is defined with a membership function $v_a(x)$ that maps each element x in a to a real number in the interval $[0, 1]$. The function value of $v_a(x)$ signifies the grade of membership of x in a [11]. A trapezoidal fuzzy number a represented with four points as follows: $a = (a^1, a^2, a^3, a^4)$, $a^1 \leq a^2 \leq a^3 \leq a^4$ (see Figure 1). Its membership function $v_a(x)$ is defined as

$$v_a(x) = \begin{cases} 0, & x < a^1, \\ \frac{x - a^1}{a^2 - a^1}, & a^1 \leq x \leq a^2, \\ 1, & a^2 \leq x \leq a^3, \\ \frac{x - a^4}{a^3 - a^4}, & a^3 \leq x \leq a^4, \\ 0, & a^4 \leq x, \end{cases} \quad (1)$$

FIGURE 1: A triangular fuzzy number a .

where a^1, a^2, a^3 , and a^4 are real numbers and these constants reflect the fuzziness of the evaluation data [12]. As shown in Figure 1, the trapezoidal fuzzy numbers can be denoted by (a^1, a^2, a^3, a^4) . The x in interval $[a^2, a^3]$ gives the maximal grade of $v_a(x)$, that is, $v_a(x) = 1$, and it is the most probable value of the evaluation data. a^1 and a^4 are the lower and upper limits of the available area for the evaluation data, and they reflect the fuzziness of the evaluation data. If $a^2 = a^3$, then $\tilde{a} = (a^1, a^2, a^3, a^4)$ is reduced to a triangular fuzzy number $\tilde{a} = (a^1, a^M, a^4)$, where $a^M = a^2 = a^3$.

Let $a = (a^1, a^2, a^3, a^4)$ and $b = (b^1, b^2, b^3, b^4)$. When $a > 0$ and $b > 0$, the basic arithmetic operations are as follows [13]:

- (1) $a + b = (a^1 + b^1, a^2 + b^2, a^3 + b^3, a^4 + b^4)$.
- (2) $a - b = (a^1 - b^1, a^2 - b^2, a^3 - b^3, a^4 - b^4)$.
- (3) $a \times b = (a^1 \times b^1, a^2 \times b^2, a^3 \times b^3, a^4 \times b^4)$.
- (4) $a/b = ((a^1/b^4), (a^2/b^3), (a^3/b^2), (a^4/b^1))$.

Definition 1 (see [14]). There are two trapezoidal fuzzy numbers $a = (a^1, a^2, a^3, a^4)$ and $b = (b^1, b^2, b^3, b^4)$. When $a^1 \geq b^4$, it can be said that a is greater than b and recorded as $a > b$; when $a^1 = b^1, a^2 = b^2, a^3 = b^3$, and $a^4 = b^4$, it can be said that a is equal to b and recorded as $a = b$; when $b^1 \geq a^4$, it can be said that a is less than b and recorded as $a < b$.

Definition 2 (see [14]). There are two trapezoidal fuzzy numbers $a = (a^1, a^2, a^3, a^4)$ and $b = (b^1, b^2, b^3, b^4)$. The distance between the trapezoidal fuzzy numbers $a = (a^1, a^2, a^3, a^4)$ and $b = (b^1, b^2, b^3, b^4)$ is defined as $d(a, b)$.

$$d(a, b) = \|a - b\| = \frac{1}{3} \sqrt{(a^1 - b^1)^2 + (a^2 - b^2)^2 + (a^3 - b^3)^2 + (a^4 - b^4)^2}. \quad (2)$$

3.1.2. Language Evaluation Phrases. Let $G = \{g_0, g_1, \dots, g_l\}$ be the preestablished finite and totally ordered linguistic term set with odd cardinalities, where g_i denotes the i th linguistic term of set G , and $l + 1$ is the cardinality of G . The middle term in linguistic term set G is thought to be represented an assessment of ‘‘approximately 0.5’’ and the remaining terms of G are thought to be placed around it symmetrically [15].

TABLE 1: Linguistic scales.

Linguistic variables and semantics	Trapezoidal fuzzy numbers
$g_0 =$ very unsatisfied (VU)	(0, 0, 0.077, 0.154)
$g_1 =$ unsatisfied (U)	(0.077, 0.154, 0.231, 0.308)
$g_2 =$ slightly unsatisfied (SU)	(0.231, 0.308, 0.385, 0.462)
$g_3 =$ middle (M)	(0.385, 0.462, 0.538, 0.615)
$g_4 =$ slightly satisfied (SS)	(0.538, 0.615, 0.692, 0.769)
$g_5 =$ satisfied (S)	(0.692, 0.769, 0.846, 0.923)
$g_6 =$ very satisfied (VS)	(0.846, 0.923, 1, 1)

For example, a linguistic term set with seven terms can be expressed as follows.

Let $G = \{g_0, g_1, g_2, g_3, g_4, g_5, g_6\}$ be a linguistic term set, where $g_0 =$ very unsatisfied (VU); $g_1 =$ unsatisfied (U); $g_2 =$ slightly unsatisfied (SU); $g_3 =$ middle (M); $g_4 =$ slightly satisfied (SS); $g_5 =$ satisfied (S); $g_6 =$ very satisfied (VS).

Therefore, g_i can be approximately expressed as a triangular fuzzy number $a = (a^1, a^2, a^3, a^4)$ using the following formula [16]:

$$a = (a^1, a^2, a^3, a^4) = \left(\max \left\{ \frac{2i-1}{2l+1}, 0 \right\}, \frac{2i}{2l+1}, \frac{2i+1}{2l+1}, \min \left\{ \frac{2i+2}{2l+1}, 1 \right\} \right). \quad (3)$$

For example, by (3), each linguistic term in a linguistic term set with seven terms, $G = \{g_0, g_1, g_2, g_3, g_4, g_5, g_6\}$, can be expressed as the corresponding trapezoidal fuzzy numbers listed in Table 1, where $l = 6$ and $i = 0, 1, 2, \dots, 6$.

3.2. Prospect Theory. Tversky and Kahneman believe that the actual decision-making behavior of an individual under uncertainty is deviated from the basic principle of expected utility. The uncertainty foreground f is a function from the natural state set S to the result set X , that is, $f : S \rightarrow X$. The foreground f is a sequence of (x_i, S_i) sequences, where S_i is a division of S which is called S_i event. When event S_i occurs, it produces a result of x_i . If $i > j$, then $x_i > x_j$. The result x_i of each foreground is arranged in ascending order, that is, $x_{-m} \leq x_{-m+1} \leq \dots \leq x_0 \leq x_1 \leq \dots \leq x_n$. If you choose x_0 as the reference point and its value is ‘‘0,’’ the profit is $x_i > 0$ and the loss is $x_i < 0$.

In 1992, Tversky and Kahneman proposed the cumulative prospect theory [17] based on the prospect theory [18]. Cumulative prospect theory has introduced capacity theory, which can better solve the problem of dominant advantage and deal with problems with multiple results. The capacity value can be expressed as w^+ and w^- , then the value of the foreground $f = (x_i, S_i)$, $(-m \leq i \leq n)$ is expressed as

$$V(f) = V(f^+) + V(f^-), \quad (4)$$

where $V(f^+)$ and $V(f^-)$ can be obtained by (5) and (6), respectively.

$$V(f^+) = \sum_{i=0}^n w_i^+ v(x_i), \quad (5)$$

$$V(f^-) = \sum_{i=0}^m w_i^- v(x_i). \quad (6)$$

Here, $w_i^+ = w^+(S_i \cup \dots \cup S_n) - w^+(S_{i+1} \cup \dots \cup S_n)$, $0 \leq i \leq n-1$; $w_i^- = w^-(S_{-m} \cup \dots \cup S_i) - w^-(S_{-m} \cup \dots \cup S_{i-1})$, $1-m \leq i \leq 0$.

If the result of event S_i is probabilistic, the decision problem can be seen as a probabilistic prospect, that is, $f = (x_i, S_i) \rightarrow f = (x_i, p_i)$. Here, $p(S_i) = p_i$ represents the probability of occurrence of event S_i . Under these circumstances, $w_i^+ = w^+(p_i \cup \dots \cup p_n) - w^+(p_{i+1} \cup \dots \cup p_n)$, $0 \leq i \leq n-1$; $w_i^- = w^-(p_{-m} \cup \dots \cup p_i) - w^-(p_{-m} \cup \dots \cup p_{i-1})$, $1-m \leq i \leq 0$.

The core content of the cumulative prospect theory is the value function v and the weight function p , which can be expressed as

$$V(f) = \sum_{i=-m}^n v(x_i) p_i. \quad (7)$$

3.2.1. The Weight Function. The weight function converts the probability into the decision weight, so the calculation formulas for the probability weight of the profit and loss are

$$p_1^+ = \frac{p_1^\gamma}{(p_1^\gamma + (1-p_1)^\gamma)^{1/\gamma}}, \quad (8)$$

$$p_2^- = \frac{p_2^\delta}{(p_2^\delta + (1-p_2)^\delta)^{1/\delta}}, \quad (9)$$

where p is the probability; γ and δ are that parameters that indicate the degree of curvature of the probability weight function.

3.2.2. The Value Function. A great breakthrough in the expectation theory is to replace the traditional utility function with a value function, so that the carriers concerned can be implemented in the value change rather than the final amount [19]. The value function is to convert surface value into decision value. The specific form of the value function of Tversky and Kahneman is [17]

$$v(x) = \begin{cases} x^\alpha, & x \geq 0, \\ -\lambda(-x)^\beta, & x < 0. \end{cases} \quad (10)$$

Here, when $x \geq 0$, $v(x)$ indicates profit; when $x < 0$, $v(x)$ indicates loss. α and β indicate the degree of roughness of the value function in the region of profit and loss, that is, the rate of decline in the sensitivity of decision-makers, $0 < \alpha < 1$ and $0 < \beta < 1$ [20]. λ indicates that the loss area of the value function is steeper than the income area, that is, it reflects the degree of loss avoidance of decision-makers. If $\lambda > 1$, it denotes that the decision-maker is more sensitive to the loss. People tend to risk gambling when faced with conditions of considerable loss but tend to accept certainty

profits when faced with fairly favorable earnings. The happiness caused by profit is not equal to that caused by the same amount of loss. The latter is greater than the former [21].

When people evaluate a thing or make a choice, they always intentionally or unintentionally compare it with a certain reference, which is called the reference point from the definition of mathematics. The reference point is a very important feature of the value function, because the profit or loss is always compared with a certain reference point. The value of the profit or loss from the reference point is located on the right side of the profit or loss to indicate positive evaluation, while the left side of the profit or loss indicates negative evaluation [22]. The reference point is used as an evaluation criterion. It is subjectively determined by an individual and will change due to different evaluation topics, environmental time, and the like.

This paper selects the expectations of MRD as a reference point for the value function. At the same time, we divide the decision indicators into benefit type and cost type. The profitability index means that the larger the index value, the better; and the cost index means that the smaller the index value, the better. According to (7) and Definition 2, the value function of the profitability index and cost index is, respectively,

$$V_{ijk}^{r_1} = \begin{cases} (h_{ijk}^r - \theta_{ik}^r)^{\alpha^k}, & h_{ijk}^r - \theta_{ik}^r \geq 0, \\ -\lambda^k (\theta_{ik}^r - h_{ijk}^r)^{\beta^k}, & h_{ijk}^r - \theta_{ik}^r < 0, \end{cases} \quad (11)$$

$$V_{ijk}^{r_2} = \begin{cases} (\theta_{ik} - h_{ijk})^{\alpha^k}, & h_{ijk} - \theta_{ik} \leq 0, \\ -\lambda^k (h_{ijk} - \theta_{ik})^{\beta^k}, & h_{ijk} - \theta_{ik} > 0, \end{cases} \quad (12)$$

where θ_{ik}^r is the reference point (i.e., expectation) given by the MRD B_k for the r index of the manufacturing task TS_i ; h_{ijk}^r indicates that the r index of the selected manufacturing unit MS_{ij} gives the manufacturing resource customers' perception after completing the manufacturing task. When $h_{ijk}^r - \theta_{ik}^r \geq 0$, $V_{ijk}^{r_1}$ and $V_{ijk}^{r_2}$ are referred to as the value of the profit generated by the MAD relative to the reference point θ_{ik}^r ; on the contrary, when $h_{ijk}^r - \theta_{ik}^r < 0$, $V_{ijk}^{r_1}$ and $V_{ijk}^{r_2}$ are referred to as the value of the loss generated by the MAD relative to the reference point θ_{ik}^r ; α^k and β^k are the degree of roughness of the profit area and the loss area of the value function, and λ^k is the loss aversion coefficient [20, 21]. In actual decision analysis, these parameters are usually obtained by nonlinear regression of experimental data [23].

3.3. The Bilevel Programming. Manufacturing unit configuration is the main form of discrete manufacturing tasks. The process of optimization selection belongs to the typical multi-objective optimization problems (MOP). The traditional MOP solution is to convert MOP into single-object problem solving. Common methods include the main target method, linear weighting method, and hierarchical optimization method. But the discrete manufacturing unit configuration

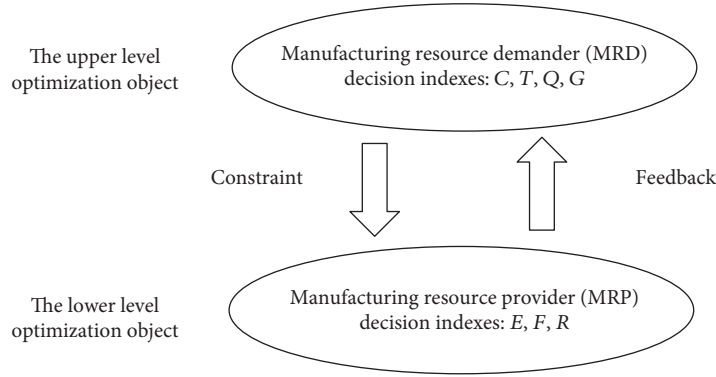


FIGURE 2: Schematic diagram of the bilevel programming.

optimization process involves the interests of the customers and suppliers, because the customers and suppliers each have a part of the optimization variables, each represents their own interests, and each variable exists between influences and constraints, and it cannot be solved using traditional methods.

The bilevel programming [24] model is a hierarchical model with a master-slave hierarchical structure. In the bilevel programming model, the upper and lower decision-makers have their own objective functions and constraints.

The upper decision-maker makes decisions firstly. According to their objective functions and constraints, the lower decision-maker obtains the optimal solutions within the possible range and sends back their own optimal solutions to the upper decision-maker. Then, the upper decision-maker obtains the global optimal solution of the problem within the possible scope based on the optimal solutions of the lower decision-maker. The mathematical description of the bilevel programming model is as follows:

$$\begin{aligned}
 & \text{(U) } \min_y Z(x, y) \\
 & \quad \text{s.t. } G(x, y) \leq 0, \\
 & \text{(L) } \min_y f(x, y) \\
 & \quad \text{s.t. } g(x, y) \leq 0,
 \end{aligned} \tag{13}$$

where (U) is the upper plan and (L) is the lower plan; Z is the objective function of the upper plan, x is the determinate variable of the upper plan, and G is the constraint condition of the decision variable x ; z is the objective function of the lower plan, y is the decision variable of the lower plan, g is the constraint condition of the decision variable y , and the lower decision variable y is the function of the upper level decision variable x , that is, $y = y(x)$.

The upper and lower optimization problems are relatively independent, and their optimization processes are dependent on each other, so the bilevel programming problems cannot usually be independently solved layer by layer. Not only the interests of MRD but also the interests of MRP must be taken into account in a discrete manufacturing environment. The idea of bilevel programming is used to solve the problem of optimal configuration of manufacturing

unit in discrete manufacturing environments, as shown in Figure 2.

4. Problem Description and Symbol Introduction

The optimal configuration of manufacturing resources in discrete environments refers to the process of configuring reasonable manufacturing resources or manufacturing resource combinations according to different manufacturing tasks [5]. After a discrete manufacturing task is issued, the MRP decomposes the total manufacturing task into a set of manufacturing subtasks according to the decomposition preferences. According to the matching principle of manufacturing resources, we find all manufacturing units that can meet the manufacturing subtasks and form a set of manufacturing units. Considering the profit of MRD and MRP, we select the optimal set of manufacturing units to form the optimal manufacturing resource configuration, as shown in Figure 3. With the emergence and development of networked manufacturing technologies, manufacturing units are not only confined to the internal structure of a single company but are also composed of multiple companies in different geographic locations, so the manufacturing resource optimization configurations are distribution, heterogeneity, and dynamics.

Compared with the existing method for optimal configuration of manufacturing resources, this paper proposes an optimization algorithm for manufacturing resources in a fuzzy environment that considers the interests of both MRD and MRP. Firstly, the MRP decomposes the MRD's product demand for the MRP (i.e., the total manufacturing task) according to design preferences and obtains a set of manufacturing subtasks. Secondly, we use the existing infrastructure to match the manufacturing subtasks with the existing manufacturing units. A set of manufacturing units for each subtask is obtained. Next, a questionnaire is used to obtain the MRP and MRD's linguistic evaluation of each manufacturing unit, and a trapezoidal fuzzy number is used to translate the language phrase into numerical values. Then, we obtain the customer's expected deviation value for each manufacturing unit based on the prospect theory. On this basis, a bilevel planning model that considers the

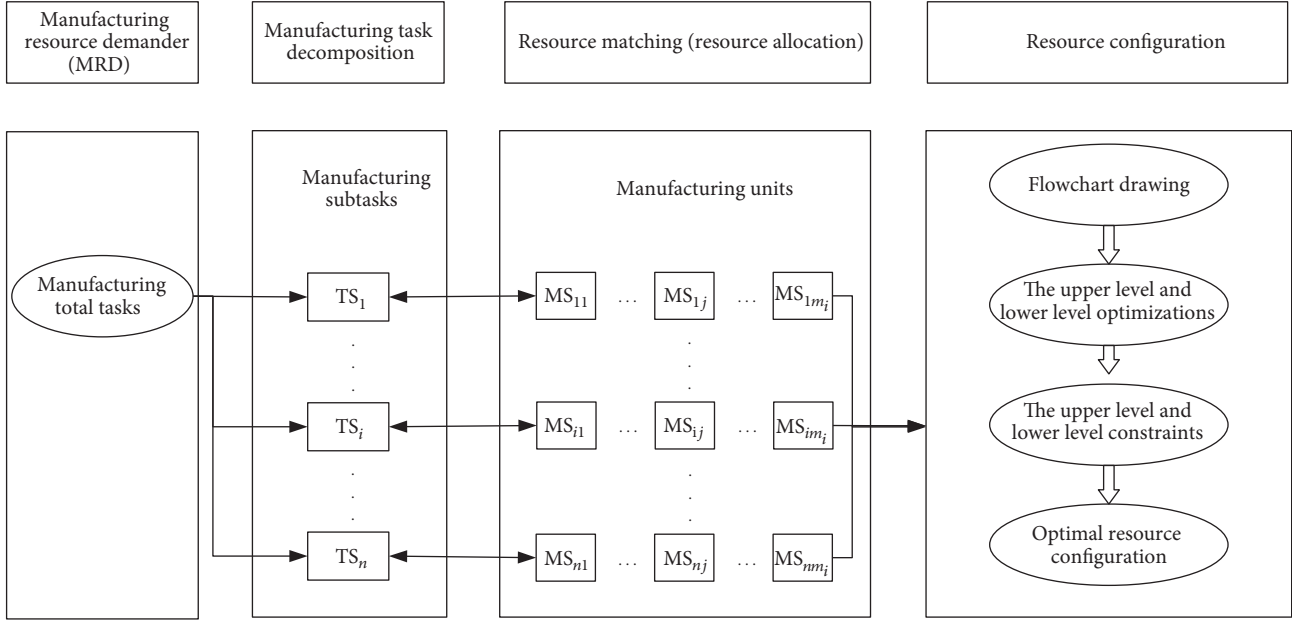


FIGURE 3: Manufacturing resource configuration diagram.

interests of both MRD and MRP was established and solved using NSGA-II algorithms. Finally, the resource configuration of aircraft-bearing processing was taken as an example to demonstrate the feasibility of the method.

- (i) $B = \{B_1, B_2, \dots, B_b\}$ is a set of manufacturing resource customers (MRD), where B_k denotes the k th manufacturing resource customers, $k = 1, \dots, b$.
- (ii) $TS = \{TS_1, TS_2, \dots, TS_n\}$ is a set of manufacturing subtasks, where TS_i denotes the i th manufacturing subtasks, $i = 1, \dots, n$.
- (iii) $MS_i = \{MS_{i1}, MS_{i2}, \dots, MS_{im_i}\}$ is a candidate manufacturing unit set of the i th manufacturing subtasks, which can provide a similar function. Where MS_{ij} denotes the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.
- (iv) $C_i = \{C_{i1}, C_{i2}, \dots, C_{im_i}\}$ is a cost set of the i th manufacturing subtasks, where C_{ij} denotes the cost of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.
- (v) $T_i = \{T_{i1}, T_{i2}, \dots, T_{im_i}\}$ is a time set of the i th manufacturing subtasks, where T_{ij} denotes the time of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.
- (vi) $Q_i = \{Q_{i1}, Q_{i2}, \dots, Q_{im_i}\}$ is a quality set of the i th manufacturing subtasks, where Q_{ij} denotes

quality of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.

- (vii) $G_i = \{G_{i1}, G_{i2}, \dots, G_{im_i}\}$ is a green set of the i th manufacturing subtasks, which is the set of environmental protection, where G_{ij} denotes green of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.
- (viii) $E_i = \{E_{i1}, E_{i2}, \dots, E_{im_i}\}$ is an efficiency set of the i th manufacturing subtasks, where E_{ij} denotes efficiency of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.
- (ix) $F_i = \{F_{i1}, F_{i2}, \dots, F_{im_i}\}$ is an agility set of the i th manufacturing subtasks, where F_{ij} denotes agility of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.
- (x) $R_i = \{R_{i1}, R_{i2}, \dots, R_{im_i}\}$ is a coordination set of the i th manufacturing subtasks, where R_{ij} denotes coordination of the j th manufacturing units of the i th manufacturing subtasks, $i = 1, \dots, n$ and $j = 1, \dots, m_i$.

5. The Proposed Method

5.1. Discrete Manufacturing Index Optimization Based on Prospect Theory. Prospect theory thinks that people are bounded rational. For example, when a person gains or loses the same item, the loss caused by the loss is much

greater than the joy after harvest. Therefore, the prospect theory can be used to indicate people's satisfaction with products or equipment. The prospect theory focuses on the difference in customer psychology, not just the customer's final value. In this paper, MRD expectation is used as a reference point to optimize the MRD indexes and MRP indexes.

5.1.1. Index Constraint Optimization of MRD

(1) *Cost Constraint.* In a discrete manufacturing system, the cost indicators of each manufacturing unit include not only the fixed costs of manufacturing units but also the logistics costs between manufacturing units. The optimization of the

cost constraint index is based on the prospect value of the MRD for the cost index. It can be calculated as

$$\begin{aligned} V(C) &= \sum_{i=1}^n V(C_i) \\ &= \sum_{i=1}^n \sum_{j=1}^{m_i} \left\| C_{ij} - \theta_{ij}^c \right\| H_{ij}, \quad i = 1, \dots, n, j = 1, \dots, m_i. \end{aligned} \quad (14)$$

Here, $V(C)$ denotes the total cost prospect value of manufacturing units selected by MRD; H_{ij} can be expressed as

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (15)$$

The traditional cost constraint is that the cost of completing the task for the selected manufacturing units' combination cannot be greater than the maximum cost required by the MRD. On the basis of the prospect theory, the traditional cost constraint conditions are transformed into that the combined cost prospect value of the selected manufacturing units cannot be greater than the maximum cost prospect value given by the MRD, which can be expressed as

$$V(C) \leq V(C)_{\max}. \quad (16)$$

(2) *Time Constraint.* In discrete manufacturing systems, the time of each manufacturing unit mainly refers to the total time from which the raw material to workshop to finished product leaving workshop. Each manufacturing unit time involves the time of the unit running and the

product twisting time between the manufacturing units. The optimization of the time constraint index is based on the prospect value of the MRD for the time index. It can be calculated as

$$\begin{aligned} V(T) &= \sum_{i=1}^n V(T_i) \\ &= \sum_{i=1}^n \sum_{j=1}^{m_i} \left\| T_{ij} - \theta_{ij}^T \right\| H_{ij}, \quad i = 1, \dots, n, j = 1, \dots, m_i. \end{aligned} \quad (17)$$

Here, $V(T)$ denotes the total time prospect value of manufacturing units selected by MRD; H_{ij} can be expressed as

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (18)$$

The traditional time constraint is that the time required for the selected manufacturing unit combination to complete the task cannot be greater than the longest delivery time required by the MRD. On the basis of the prospect theory, the traditional time constraint conditions are transformed into that the combined time prospect value of the selected manufacturing units cannot be greater than the maximum time prospect value given by the MRD.

$$V(T) \leq V(T)_{\max}. \quad (19)$$

(3) *Quality Constraint.* In the discrete manufacturing system, the quality index of each manufacturing unit refers to the

quality of each manufacturing unit to complete the relevant manufacturing tasks, that is, the quality qualification rate for each manufacturing unit to complete the relevant manufacturing tasks. The optimization of the quality constraint index is based on the prospect value of the MRD for the quality index. It can be calculated as

$$\begin{aligned} V(Q) &= \sum_{i=1}^n V(Q_i) \\ &= \sum_{i=1}^n \sum_{j=1}^{m_i} \left\| Q_{ij} - \theta_{ij}^Q \right\| H_{ij}, \quad i = 1, \dots, n, j = 1, \dots, m_i, \end{aligned} \quad (20)$$

where $V(T)$ denotes the total quality prospect value of manufacturing units selected by MRD; H_{ij} can be expressed as

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (21)$$

The traditional quality constraint is that the quality qualification rate of any manufacturing unit in the selected manufacturing units' combination should not be less than the minimum quality qualification rate required by the MDR. On the basis of the prospect theory, the traditional quality constraint conditions are transformed into that the quality prospect value of any manufacturing unit in the selected manufacturing unit combination should not be less than the minimum quality prospect value required by the MRD, which can be expressed as

$$V(Q_{ij}) \geq V(Q)_{\min}, \quad i = 1, \dots, n, j = 1, \dots, m_i. \quad (22)$$

(4) *Green Constraint.* Green is based on the requirements of the current low carbon environmental protection. The evaluation of the environmental indicators includes carbon emissions and processing material losses in the manufacturing process of manufacturing units. In discrete manufacturing systems, the green index of each

manufacturing unit is the degree of pollution to the environment by which the selected manufacturing units are combined to complete the related manufacturing tasks. The optimization of the green constraint index is based on the prospect value of the MRD for the green index. It can be calculated as

$$\begin{aligned} V(G) &= \sum_{i=1}^n V(G_i) \\ &= \sum_{i=1}^n \sum_{j=1}^{m_i} \left\| G_{ij} - \theta_{ij}^G \right\| H_{ij}, \quad i = 1, \dots, n, j = 1, \dots, m_i, \end{aligned} \quad (23)$$

where $V(G)$ denotes the total green prospect value of manufacturing units selected by MRD; H_{ij} can be expressed as

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (24)$$

The traditional environmental protection constraint is that the pollution rate of any manufacturing unit in the selected manufacturing unit combination cannot be greater than the highest manufacturing pollution rate required by the MRD. On the basis of the prospect theory, the traditional environmental constraints are translated into that the prospect value of pollution of any manufacturing unit in the selected combination of manufacturing units cannot be greater than the maximum prospect value of pollution required by the MRD, which can be expressed as

$$V(G_{ij}) \leq V(G)_{\max}, \quad i = 1, \dots, n, j = 1, \dots, m_i. \quad (25)$$

5.1.2. *Index Constraint Optimization of MRP.* Manufacturing tasks in discrete manufacturing systems are susceptible to changes in manufacturing tasks and changes in manufacturing resources, making it impossible to complete manufacturing tasks with high quality and efficiency. Therefore, it is very important for manufacturing tasks to consider the factors in the process of optimizing the configuration of manufacturing resources.

(1) *The Efficiency of the Manufacturing Units' Configuration.* The efficiency of the configuration of manufacturing units in a discrete manufacturing system refers to the number of qualified products that the manufacturing unit produces within a unit time after the manufacturing task is reached. The efficiency index of manufacturing units' configuration mainly includes the functional efficiency of each manufacturing unit E_{ij}^n and the decomposition capability of the manufacturing unit E_{ij}^f , which can be expressed as

$$\begin{aligned} \max E &= \max \sum_{i=1}^n \sum_{j=1}^{m_i} (E_{ij} H_{ij}) \\ &= \max \left\{ \sum_{i=1}^n \sum_{j=1}^{m_i} (E_{ij}^n H_{ij}) + \sum_{i=1}^n \sum_{j=1}^{m_i} (E_{ij}^f H_{ij}) \right\}, \\ & \quad i = 1, \dots, n, j = 1, \dots, m_i. \end{aligned} \quad (26)$$

Here,

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (27)$$

The functional efficiency E_{ij}^n of any manufacturing units cannot be less than the minimum functional efficiency E_{\min}^n required by MRP; and the decomposition capability of the manufacturing unit E_{ij}^f of any MRP in the manufacturing unit's configuration cannot be less than the minimum decomposition capability of the manufacturing unit E_{\min}^f required by MRP. They can be expressed as

$$\begin{aligned} E_{ij}^n &\geq E_{\min}^n, & i = 1, \dots, n, j = 1, \dots, m_i, \\ E_{ij}^f &\geq E_{\min}^f, & i = 1, \dots, n, j = 1, \dots, m_i. \end{aligned} \quad (28)$$

(2) *The Agility of the Manufacturing Units' Configuration.* The agility of the manufacturing unit configuration in a discrete manufacturing environment refers to the manufacturing units' configuration ability to react quickly and successfully complete the manufacturing task when the content of the manufacturing task changes or a manufacturing resource withdraws for some reason. So that enterprises

can cope with the rapidly changing and unpredictable market demand, thus obtaining long-term economic benefits of enterprises. The agility index of manufacturing units' configuration mainly includes the functional diversity F_{ij}^d and the manufacturing resource types F_{ij}^z of each manufacturing unit, which can be expressed as

$$\begin{aligned} \max F &= \max \sum_{i=1}^n \sum_{j=1}^{m_i} (F_{ij} H_{ij}) \\ &= \max \left\{ \sum_{i=1}^n \sum_{j=1}^{m_i} (F_{ij}^d H_{ij}) + \sum_{i=1}^n \sum_{j=1}^{m_i} (F_{ij}^z H_{ij}) \right\}, \\ & \quad i = 1, \dots, n, j = 1, \dots, m_i, \end{aligned} \quad (29)$$

where

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (30)$$

The functional diversity F_{ij}^d of any manufacturing unit cannot be less than the minimum functional diversity F_{\min}^d required by the MRP; the type of manufacturing resources F_{ij}^z provided by any MRP in the manufacturing units' configuration should not be less than the minimum type of manufacturing resource F_{\min}^z required by the MRP. They can be expressed as

$$\begin{aligned} F_{ij}^d &\geq F_{\min}^d, & i = 1, \dots, n, j = 1, \dots, m_i, \\ F_{ij}^z &\geq F_{\min}^z, & i = 1, \dots, n, j = 1, \dots, m_i. \end{aligned} \quad (31)$$

(3) *The Coordination of the Manufacturing Units' Configuration.* The coordination of the manufacturing units' configuration in discrete manufacturing environment refers to manufacturing units that can coordinate and efficiently

complete manufacturing task when the manufacturing task comes down. The coordination index of manufacturing units' configuration mainly includes the reliability of the manufacturing unit R_{ij}^k and the compatibility of the manufacturing unit R_{ij}^x , which can be expressed as

$$\begin{aligned} \max R &= \max \sum_{i=1}^n \sum_{j=1}^{m_i} (R_{ij} H_{ij}) \\ &= \max \left\{ \sum_{i=1}^n \sum_{j=1}^{m_i} (R_{ij}^k H_{ij}) + \sum_{i=1}^n \sum_{j=1}^{m_i} (R_{ij}^x H_{ij}) \right\}, \\ & \quad i = 1, \dots, n, j = 1, \dots, m_i, \end{aligned} \quad (32)$$

where

$$H_{ij} = \begin{cases} 1, & \text{the } i\text{th manufacturing subtask is completed by the manufacturing unit MS}_{ij}, \\ 0, & \text{the } i\text{th manufacturing subtask is not completed by the manufacturing unit MS}_{ij}. \end{cases} \quad (33)$$

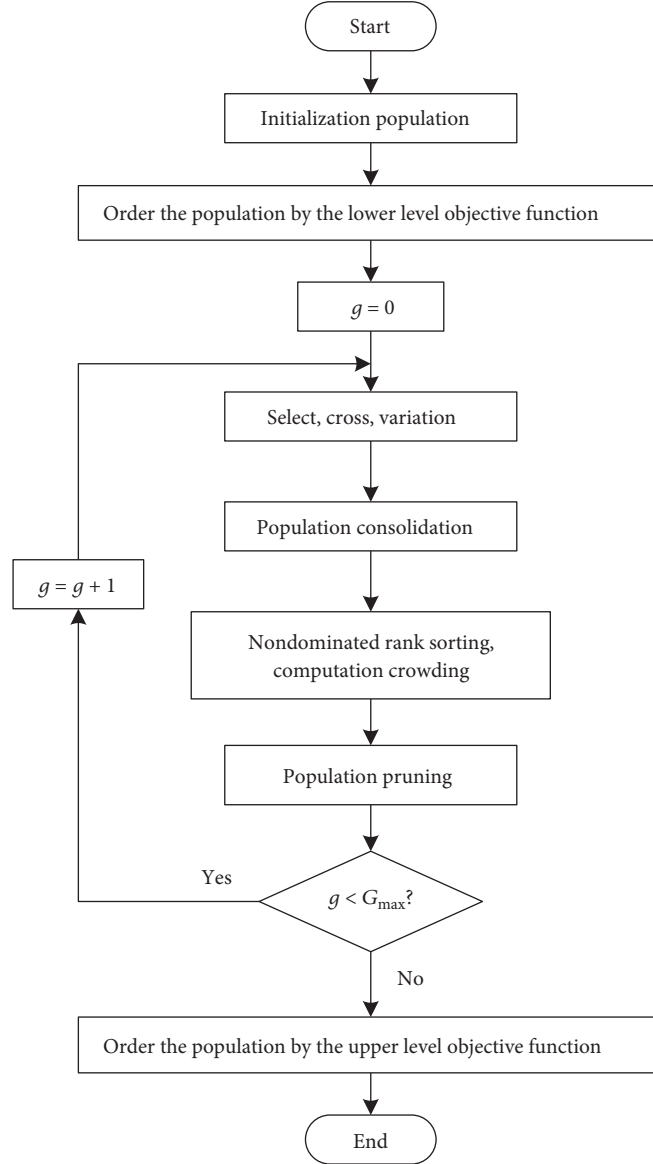


FIGURE 4: The NSGA-II algorithm for bilevel programming.

TABLE 2: Set of candidate manufacturing units.

Manufacturing subtasks	TS ₁	TS ₂	TS ₃	TS ₄	TS ₅	TS ₆	TS ₇	TS ₈
	MS ₁₁	MS ₂₁	MS ₃₁	MS ₄₁	MS ₅₁	MS ₆₁	MS ₇₁	MS ₈₁
	MS ₁₂	MS ₂₂	MS ₃₂	MS ₄₂	MS ₅₂	MS ₆₂	MS ₇₂	MS ₈₂
Manufacturing units	MS ₁₃	MS ₂₃	MS ₃₃	MS ₄₃	MS ₅₃	MS ₆₃	MS ₇₃	MS ₈₃
		MS ₂₄	MS ₃₄		MS ₅₄	MS ₆₄		MS ₈₄
		MS ₂₅				MS ₆₅		

The reliability of any manufacturing unit R_{ij}^k in the manufacturing units' configuration cannot be less than the minimum reliability R_{\min}^k required by the MRP. The coordination of any manufacturing unit R_{ij}^x in the manufacturing unit combination cannot be less than the minimum

coordination R_{\min}^x required by the MRP. They can be expressed as

$$\begin{aligned}
 R_{ij}^k &\geq R_{\min}^k, & i = 1, \dots, n, j = 1, \dots, m_i, \\
 R_{ij}^x &\geq R_{\min}^x, & i = 1, \dots, n, j = 1, \dots, m_i.
 \end{aligned} \tag{34}$$

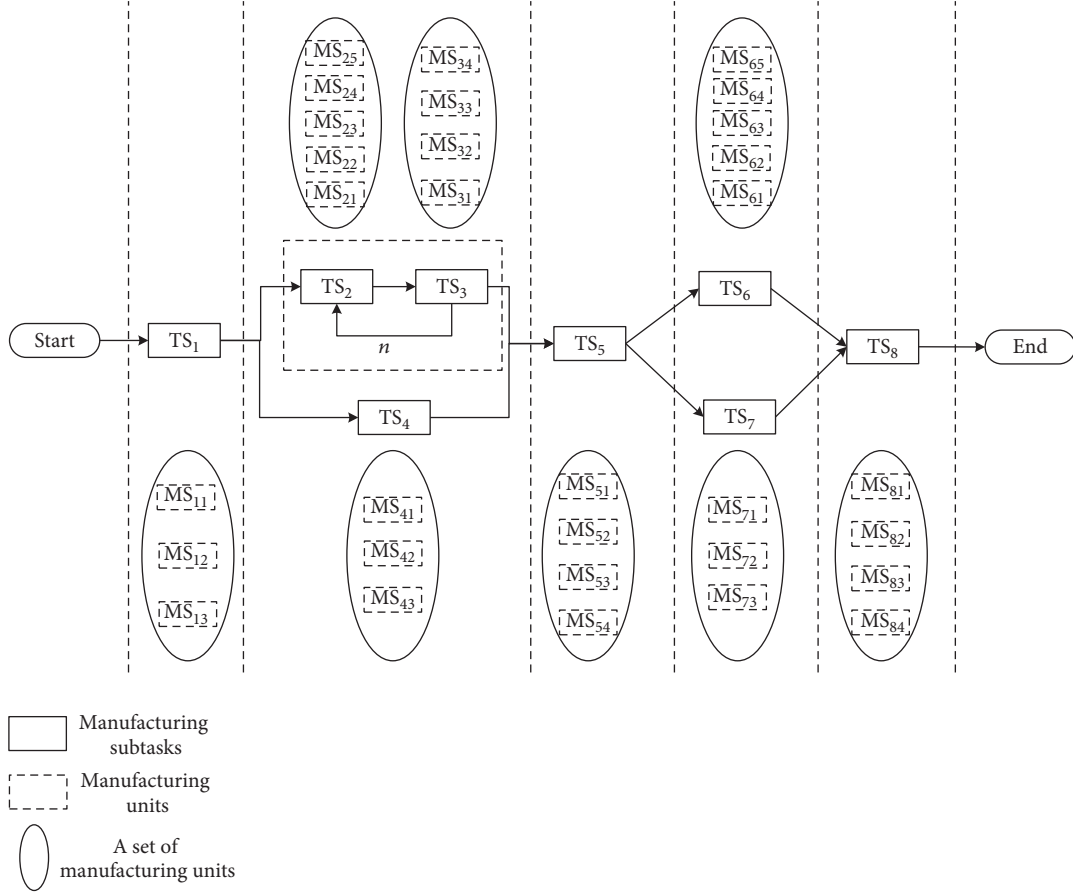


FIGURE 5: Manufacturing task flow chart.

5.2. Discrete Manufacturing Optimization Model Based on Bilevel Programming. In discrete manufacturing, it is difficult for MRD and MRP to express their preference for a certain characteristic with an explicit or implicit “utility function,” but it is generally easy to determine what level of expectation a certain target achieves and according to the actual situation to adjust the level of the object expected to reach. In these elements that influence multiobjective decision-making in discrete manufacturing environment, they may contain both integer variables and fractional variables and may be continuous or discontinuous [25].

To solve this kind of decision-making problem, this paper proposes a bilevel programming decision method based on the object expectation: selecting index cost (C), time (T), quality (Q), and green (G) as the upper level optimization objectives that affect the MRD of manufacturing resources and then, selecting the index efficiency (E), agility (F), and coordination (R) as the lower level optimization objectives that affect the MRP of manufacturing resources. The bilevel optimization model under the discrete manufacturing environment is established as follows:

$$(U) \min \quad Z = w_c \times \frac{V(C)}{V(C)_{\max}} + w_T \times \frac{V(T)}{V(T)_{\max}} + w_Q \times \frac{V(Q)}{V(Q)_{\max}} + w_G \times \frac{V(G)}{V(G)_{\max}} \quad (35)$$

$$\text{s.t.} \quad V(C_{ij}) \leq V(C)_{\max}, \quad (36)$$

$$V(T_{ij}) \leq V(T)_{\max}, \quad (37)$$

$$V(Q_{ij}) \geq V(Q)_{\min}, \quad (38)$$

$$V(G_{ij}) \geq V(G)_{\min}, \quad (39)$$

$$(L) \max \quad z = (E, F, R)^T \quad (40)$$

$$\text{s.t.} \quad E_{ij}^n \geq E_{\min}^n, \quad (41)$$

$$E_{ij}^f \geq E_{\min}^f, \quad (42)$$

$$F_{ij}^d \geq F_{\min}^d, \quad (43)$$

$$F_{ij}^z \geq F_{\min}^z, \quad (44)$$

$$R_{ij}^k \geq R_{\min}^k, \quad (45)$$

$$R_{ij}^x \geq R_{\min}^x. \quad (46)$$

Here, (35) denotes the upper-level optimization object given by the MRD, and (36) and (39) denote the quality, cost, time, and green indicator constraints of the upper-level optimization objectives; (40) denotes the lower-level optimization object given by the MRP, and (41) and (46) denote the

TABLE 3: Language evaluation of the MRD (one MRD's example).

Manufacturing units	C	T	Q	G
MS ₁₁	Between S and VS	Between M and VS	Between SU and M	Between M and SS
MS ₁₂	Greater than M	Between U and M	Greater than M	Between M and VS
MS ₁₃	Greater than M	Lower than M	Greater than M	Between VU and S
MS ₂₁	Between U and M	Greater than S	Between U and M	Lower than M
MS ₂₂	Between M and SS	Greater than M	Between S and VS	Between SS and S
MS ₂₃	Between M and S	Between U and M	Greater than M	Greater than SS
MS ₂₄	Between M and S	Between VU and SU	Greater than M	Lower than M
MS ₂₅	Greater than M	Between M and S	Between U and M	Between U and SS
MS ₃₁	Greater than S	Lower than M	Between M and SS	Lower than M
MS ₃₂	Between U and M	Between S and VS	Between M and S	Between VU and M
MS ₃₃	Greater than M	Greater than M	Between M and S	Greater than M
MS ₃₄	Greater than U	Greater than M	Between U and SS	Greater than M
MS ₄₁	Between VU and U	Between U and M	Between M and VS	Lower than M
MS ₄₂	Between U and M	Between M and SS	Greater than M	Greater than S
MS ₄₃	Greater than S	Between M and S	Greater than S	Greater than M
MS ₅₁	Greater than S	Between M and S	Between SU and SS	Between U and M
MS ₅₂	Between SS and S	Greater than M	Between U and VS	Between VU and SU
MS ₅₃	Between VU and U	Between S and VS	Between M and VS	Between M and S
MS ₅₄	Between M and S	Greater than S	Between VU and M	Lower than M
MS ₆₁	Between U and S	Between S and VS	Lower than M	Between S and VS
MS ₆₂	Lower than M	Lower than M	Greater than S	Greater than M
MS ₆₃	between SS and S	Between U and M	Greater than M	Greater than M
MS ₆₄	Greater than M	Greater than M	Between U and M	Between U and M
MS ₆₅	Between VU and M	Between S and VS	Between VU and SU	Between M and SS
MS ₇₁	Greater than M	Greater than S	Between M and S	Between M and S
MS ₇₂	Lower than M	Lower than M	Greater than M	Between M and S
MS ₇₃	Greater than S	Greater than SS	Greater than M	Between S and VS
MS ₈₁	Greater than M	Greater than S	Between U and M	Between M and S
MS ₈₂	Between U and M	Greater than S	Between U and SS	Greater than M
MS ₈₃	Between VU and SU	Between SS and S	Between M and VS	Lower than M
MS ₈₄	Between M and S	Between VU and U	Between U and S	Between U and M
Expectation value	M	SS	S	M

efficiency, coordination, and agility indicator constraints of the lower-level optimization object.

5.3. Model Solving. There are more famous algorithms for resource configuration optimization, such as genetic algorithm, ant colony algorithm, particle swarm optimization, and simulated annealing algorithm, but these algorithms have certain constraints when solving the bilevel programming model of manufacturing resource optimization configuration in this paper. The bilevel planning is a NP-hard (nondeterministic polynomial, NP) problem. The process of solving such problems is very complicated. The fast

nondominated sorting genetic algorithm (NSGA-II) is an improvement of the NSGA algorithm and is one of the best evolutionary multiobjective optimization algorithms [26, 27].

This paper uses the NSGA-II algorithm to solve the bilevel planning model under the discrete manufacturing environment. The general flow is shown in Figure 4.

6. Illustrative Example

6.1. Model Establishment of Discrete Resource Configuration. The MDR submits the manufacturing task to the MDP according to the market demand. On the basis of certain

TABLE 4: The transformation of trapezoid fuzzy numbers for language evaluation (one MRD's example).

Manufacturing units	C	T	Q	G
MS ₁₁	(0.692, 0.769, 1, 1)	(0.385, 0.462, 1, 1)	(0.231, 0.308, 0.538, 0.615)	(0.385, 0.462, 0.692, 0.769)
MS ₁₂	(0.385, 0.462, 1, 1)	(0.077, 0.154, 0.538, 0.615)	(0.385, 0.462, 1, 1)	(0.385, 0.462, 1, 1)
MS ₁₃	(0.385, 0.462, 1, 1)	(0, 0, 0.538, 0.615)	(0.385, 0.462, 1, 1)	(0, 0, 0.846, 0.923)
MS ₂₁	(0.077, 0.154, 0.538, 0.615)	(0.692, 0.769, 1, 1)	(0.077, 0.154, 0.538, 0.615)	(0, 0, 0.538, 0.615)
MS ₂₂	(0.385, 0.462, 0.692, 0.769)	(0.385, 0.462, 1, 1)	(0.692, 0.769, 1, 1)	(0.077, 0.154, 0.846, 0.923)
MS ₂₃	(0.385, 0.462, 0.846, 0.923)	(0.077, 0.154, 0.538, 0.615)	(0.385, 0.462, 1, 1)	(0.538, 0.615, 0.846, 0.923)
MS ₂₄	(0.385, 0.462, 0.846, 0.923)	(0, 0, 0.385, 0.462)	(0.385, 0.462, 1, 1)	(0, 0, 0.538, 0.615)
MS ₂₅	(0.385, 0.462, 1, 1)	(0.385, 0.462, 0.846, 0.923)	(0.077, 0.154, 0.538, 0.615)	(0.077, 0.154, 0.692, 0.769)
MS ₃₁	(0.692, 0.769, 1, 1)	(0, 0, 0.538, 0.615)	(0.385, 0.462, 0.692, 0.769)	(0, 0, 0.538, 0.615)
MS ₃₂	(0.077, 0.154, 0.538, 0.615)	(0.692, 0.769, 1, 1)	(0.385, 0.462, 0.846, 0.923)	(0, 0, 0.538, 0.615)
MS ₃₃	(0.385, 0.462, 1, 1)	(0.385, 0.462, 1, 1)	(0.385, 0.462, 0.846, 0.923)	(0.385, 0.462, 1, 1)
MS ₃₄	(0.077, 0.154, 1, 1)	(0.385, 0.462, 1, 1)	(0.077, 0.154, 0.692, 0.769)	(0.385, 0.462, 1, 1)
MS ₄₁	(0, 0, 0.231, 0.308)	(0.077, 0.154, 0.538, 0.615)	(0.385, 0.462, 1, 1)	(0, 0, 0.538, 0.615)
MS ₄₂	(0.077, 0.154, 0.538, 0.615)	(0.385, 0.462, 0.692, 0.769)	(0.385, 0.462, 1, 1)	(0.692, 0.769, 1, 1)
MS ₄₃	(0.692, 0.769, 1, 1)	(0.385, 0.462, 0.846, 0.923)	(0.692, 0.769, 1, 1)	(0.385, 0.462, 1, 1)
MS ₅₁	(0.692, 0.769, 1, 1)	(0.385, 0.462, 0.846, 0.923)	(0.231, 0.308, 0.692, 0.769)	(0.077, 0.154, 0.538, 0.615)
MS ₅₂	(0.538, 0.615, 0.846, 0.923)	(0.385, 0.462, 1, 1)	(0.077, 0.154, 1, 1)	(0, 0, 0.385, 0.462)
MS ₅₃	(0, 0, 0.231, 0.308)	(0.692, 0.769, 1, 1)	(0.385, 0.462, 1, 1)	(0.385, 0.462, 0.846, 0.923)
MS ₅₄	(0.385, 0.462, 0.846, 0.923)	(0.692, 0.769, 1, 1)	(0, 0, 0.538, 0.615)	(0, 0, 0.538, 0.615)
MS ₆₁	(0.077, 0.154, 0.849, 0.923)	(0.692, 0.769, 1, 1)	(0, 0, 0.538, 0.615)	(0.692, 0.769, 1, 1)
MS ₆₂	(0, 0, 0.538, 0.615)	(0, 0, 0.538, 0.615)	(0.692, 0.769, 1, 1)	(0.385, 0.462, 1, 1)
MS ₆₃	(0.538, 0.615, 0.846, 0.923)	(0.077, 0.154, 0.538, 0.615)	(0.385, 0.462, 1, 1)	(0.385, 0.462, 1, 1)
MS ₆₄	(0.385, 0.462, 1, 1)	(0.385, 0.462, 1, 1)	(0.077, 0.154, 0.538, 0.615)	(0.077, 0.154, 0.538, 0.615)
MS ₆₅	(0, 0, 0.538, 0.615)	(0.692, 0.769, 1, 1)	(0, 0, 0.385, 0.462)	(0.385, 0.462, 0.692, 0.769)
MS ₇₁	(0.385, 0.462, 1, 1)	(0.692, 0.769, 1, 1)	(0.385, 0.462, 0.846, 0.923)	(0.385, 0.462, 0.846, 0.923)
MS ₇₂	(0, 0, 0.538, 0.615)	(0, 0, 0.538, 0.615)	(0.385, 0.462, 1, 1)	(0.385, 0.462, 0.846, 0.923)
MS ₇₃	(0.692, 0.769, 1, 1)	(0.538, 0.615, 0.846, 0.923)	(0.385, 0.462, 1, 1)	(0.692, 0.769, 1, 1)
MS ₈₁	(0.385, 0.462, 1, 1)	(0.692, 0.769, 1, 1)	(0.077, 0.154, 0.538, 0.615)	(0.385, 0.462, 0.846, 0.923)
MS ₈₂	(0.077, 0.154, 0.538, 0.615)	(0.692, 0.769, 1, 1)	(0.077, 0.154, 0.692, 0.769)	(0.385, 0.462, 1, 1)
MS ₈₃	(0, 0, 0.385, 0.462)	(0.538, 0.615, 0.846, 0.923)	(0.385, 0.462, 1, 1)	(0, 0, 0.538, 0.615)
MS ₈₄	(0.385, 0.462, 0.846, 0.923)	(0, 0, 0.231, 0.308)	(0.077, 0.154, 0.846, 0.923)	(0.077, 0.154, 0.538, 0.615)
Expectation value	(0.385, 0.462, 0.538, 0.615)	(0.538, 0.615, 0.692, 0.769)	(0.692, 0.769, 0.846, 0.923)	(0.385, 0.462, 0.538, 0.615)

principles, the MDP decomposes the manufacturing task and gets the set of manufacturing subtasks including 8 manufacturing subtasks (TS_1, TS_2, \dots, TS_8). The MDP uses the function object matching principle to form a set of manufacturing units that complete each subtask, as shown in Table 2. The sequence of tasks is successive, as shown in Figure 5.

The questionnaire method to collect the language evaluation of the MRD is given by the acceptable range and expectation value for the four indexes ($C/T/Q/G$) of each manufacturing unit, as shown in Table 3.

According to (3) and Table 4, the acceptable range and expectation value of the MRD are converted into trapezoidal fuzzy numbers, as shown in Table 4.

Selecting the MRD expectation value as the reference point and referencing literature [28] obtains $\alpha = \beta = 0.88$, $\lambda = 2.25$, and $\delta = 0.56$. Next, we calculate the expected deviation (i.e., value function) of each manufacturing unit, according to Definition 2 and (11) and (12) and then based on (8) and (9) to calculate the decision weights of each manufacturing unit. Finally, according to (7), we can calculate the prospect value of MRD for each manufacturing, which is shown in Table 5.

In the same way, first of all, we get the language evaluation of three indexes (E/F/R) given by the MRP. Then, according to (3) and Table 2, we used formula $a = (a_1 + 2a_2 + 2a_3 + a_4)/6$ to defuzzify the trapezoidal fuzzy numbers, as shown in Table 6.

Assume that the parameters of the bilevel programming model for optimal configuration of manufacturing resources are $w_c = 0.272$, $w_T = 0.231$, $w_Q = 0.301$, $w_G = 0.196$, $V(C)_{\max} = 0.541$, $V(T)_{\max} = 0.425$, $V(Q)_{\min} = 0.786$, $V(G)_{\min} = 0.775$, $E_{\min}^n = 0.128$, $E_{\min}^f = 0.128$, $F_{\min}^d = 0.205$, $F_{\min}^z = 0.205$, $R_{\min}^k = 0.195$, and $R_{\min}^x = 0.180$. According to (35) and (40), a bilevel optimization model for resource optimization configuration is as follows:

$$\begin{aligned}
 \text{(U) min} \quad & Z = 0.506V(C_{ij}) + 0.544V(T_{ij}) \\
 & + 0.382V(Q_{ij}) + 0.253V(G_{ij}) \\
 \text{s.t.} \quad & V(C_{ij}) \leq 0.541, \\
 & V(T_{ij}) \leq 0.425, \\
 & V(Q_{ij}) \geq 0.214, \\
 & V(G_{ij}) \geq 0.225,
 \end{aligned} \tag{47}$$

$$\begin{aligned}
 \text{(L) max} \quad & z = (E, F, R)^T \\
 \text{s.t.} \quad & E_{ij}^n \geq 0.128, \\
 & E_{ij}^f \geq 0.128, \\
 & F_{ij}^d \geq 0.205, \\
 & F_{ij}^z \geq 0.205, \\
 & R_{ij}^k \geq 0.195, \\
 & R_{ij}^x \geq 180.
 \end{aligned} \tag{48}$$

6.2. Model Solving. NSGA-II algorithm was used to solve the model study (48). The initial population of the algorithm is 50; the crossover probability is 0.6; the mutation probability is 0.03; and the largest genetic algebra is 200. In the MATLAB 2016a calculation environment, the average fitness of each population under different fitness functions is calculated [29], as shown in Figure 6. In the 200-generation evolution process, the average fitness value after 50 generations tends to be stable. Therefore, after 50 generations of evolution, the Pareto optimal solution of the lower level optimization object of the manufacturing resource optimization configuration model is obtained. Calculate the Pareto frontier of optimal solution sets consisting of 50 solutions, as shown in Figure 7. Each point in the graph represents a Pareto optimal solution. The entire Pareto optimal solution set is located on the first-level Pareto frontier and distributed uniformly. That is, the ideal Pareto optimal solution set is obtained.

The Pareto optimal solution of the lower-level optimization of the manufacturing unit optimization selection model is regarded as the feasible solution of the upper optimization goal, the corresponding target value of the upper optimization goal is calculated, and the global optimal solution of the optimal selection bilevel programming model is obtained, according to the advantages and disadvantages. The degree is ranked and the

TABLE 5: The prospect value of MRD (one MRD's example).

Manufacturing units	C	T	Q	G
MS ₁₁	0.327	0.307	0.625	0.658
MS ₁₂	0.272	0.255	0.520	0.547
MS ₁₃	0.272	0.255	0.520	0.547
MS ₂₁	0.205	0.192	0.391	0.412
MS ₂₂	0.112	0.105	0.214	0.225
MS ₂₃	0.205	0.193	0.392	0.413
MS ₂₄	0.205	0.193	0.392	0.413
MS ₂₅	0.273	0.255	0.520	0.547
MS ₃₁	0.327	0.307	0.625	0.658
MS ₃₂	0.205	0.192	0.391	0.412
MS ₃₃	0.272	0.255	0.520	0.547
MS ₃₄	0.328	0.307	0.626	0.659
MS ₄₁	0.327	0.307	0.625	0.658
MS ₄₂	0.205	0.192	0.391	0.412
MS ₄₃	0.327	0.307	0.625	0.658
MS ₅₁	0.327	0.307	0.625	0.658
MS ₅₂	0.226	0.212	0.431	0.454
MS ₅₃	0.327	0.307	0.625	0.658
MS ₅₄	0.205	0.192	0.391	0.412
MS ₆₁	0.279	0.261	0.532	0.560
MS ₆₂	0.272	0.255	0.520	0.547
MS ₆₃	0.226	0.212	0.431	0.454
MS ₆₄	0.272	0.255	0.520	0.547
MS ₆₅	0.272	0.255	0.520	0.547
MS ₇₁	0.272	0.255	0.520	0.547
MS ₇₂	0.272	0.255	0.520	0.547
MS ₇₃	0.327	0.307	0.625	0.658
MS ₈₁	0.272	0.255	0.520	0.547
MS ₈₂	0.207	0.194	0.395	0.416
MS ₈₃	0.287	0.269	0.548	0.577
MS ₈₄	0.205	0.192	0.391	0.412

first five groups of solutions are listed. The group with the smallest target value is the global optimal solution that satisfies the optimal choice between the customer and the manufacturer.

The Pareto optimal solution of the lower level optimization object of the manufacturing unit optimization model is the feasible solution of the upper level optimization object; then, the corresponding the upper level optimization object values are calculated. Further, we get the global optimal solution of the bilevel programming model and sort them according to their pros and cons and take the first five solutions listed, as shown in Table 7. The group with the smallest object value is the global optimal solution of the optimization problem.

TABLE 6: Evaluation information of MRP.

Manufacturing units	The functional efficiency E_{ij}^n	The decomposition capability E_{ij}^f	The functional diversity F_{ij}^d	The manufacturing resource types F_{ij}^z	The reliability R_{ij}^k	The compatibility R_{ij}^x
MS ₁₁	0.872	0.718	0.423	0.577	0.682	0.572
MS ₁₂	0.718	0.346	0.718	0.718	0.682	0.417
MS ₁₃	0.718	0.282	0.718	0.419	0.329	0.286
MS ₂₁	0.346	0.872	0.346	0.282	0.818	0.561
MS ₂₂	0.573	0.718	0.861	0.483	0.682	0.542
MS ₂₃	0.650	0.346	0.718	0.714	0.682	0.415
MS ₂₄	0.650	0.205	0.718	0.282	0.329	0.201
MS ₂₅	0.714	0.654	0.346	0.423	0.548	0.489
MS ₃₁	0.872	0.282	0.577	0.282	0.621	0.242
MS ₃₂	0.346	0.872	0.654	0.282	0.621	0.561
MS ₃₃	0.718	0.718	0.654	0.718	0.402	0.618
MS ₃₄	0.564	0.718	0.423	0.718	0.682	0.618
MS ₄₁	0.128	0.346	0.718	0.282	0.682	0.277
MS ₄₂	0.346	0.577	0.718	0.872	0.828	0.591
MS ₄₃	0.872	0.654	0.872	0.718	0.475	0.583
MS ₅₁	0.872	0.654	0.500	0.346	0.536	0.464
MS ₅₂	0.731	0.718	0.564	0.205	0.682	0.454
MS ₅₃	0.128	0.872	0.718	0.654	0.268	0.680
MS ₅₄	0.654	0.872	0.282	0.282	0.268	0.561
MS ₆₁	0.501	0.872	0.282	0.872	0.828	0.750
MS ₆₂	0.282	0.282	0.872	0.718	0.682	0.382
MS ₆₃	0.731	0.346	0.718	0.718	0.329	0.417
MS ₆₄	0.718	0.718	0.346	0.346	0.195	0.499
MS ₆₅	0.282	0.872	0.205	0.577	0.621	0.655
MS ₇₁	0.718	0.872	0.654	0.654	0.682	0.680
MS ₇₂	0.282	0.282	0.718	0.654	0.682	0.361
MS ₇₃	0.872	0.731	0.718	0.872	0.329	0.673
MS ₈₁	0.718	0.872	0.346	0.654	0.389	0.680
MS ₈₂	0.345	0.872	0.410	0.718	0.682	0.701
MS ₈₃	0.205	0.731	0.718	0.282	0.475	0.485
MS ₈₄	0.654	0.128	0.500	0.346	0.276	0.180

6.3. *Analysis of Algorithm Results.* The results of the solution of the example are compared with the results of the traditional algorithm to prove the validity of the model example.

The upper-level optimization objective of the bilevel programming model, presents the objective of the traditional manufacturing resource configuration model. The exhaustive method is used to calculate the functional target decision-making scheme, and all the obtained data are ranked according to the degree of pros and cons. The top 10 groups are listed, as shown in Table 8.

From Tables 7 and 8, the five optimal solutions (groups 1, 3, 4, 7, and 9 in Table 7) of the model study in this paper all

contain the optimal top 10 optimal solutions. Therefore, the results of the model calculation in this paper not only satisfy the upper level optimization object but also satisfy the lower level optimization object of the model.

7. Discussion

In this paper, when establishing a bilevel programming of the discrete manufacturing resource optimization configuration model, the upper level optimization object is to select the smallest comprehensive value of the MRD and the lower level optimization object is to consider the maximum demand of the MRP.

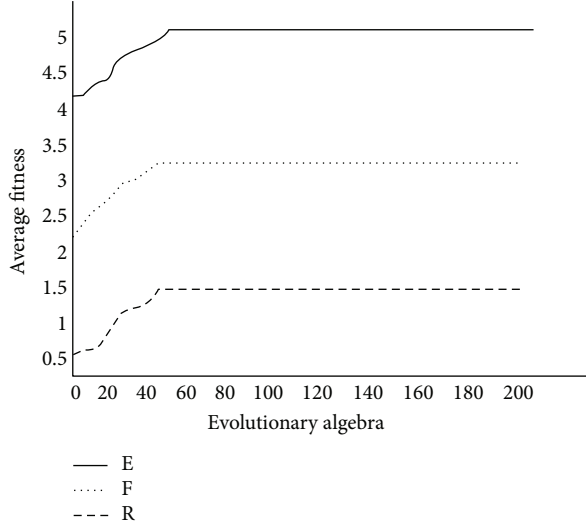


FIGURE 6: Average fitness curve of population.

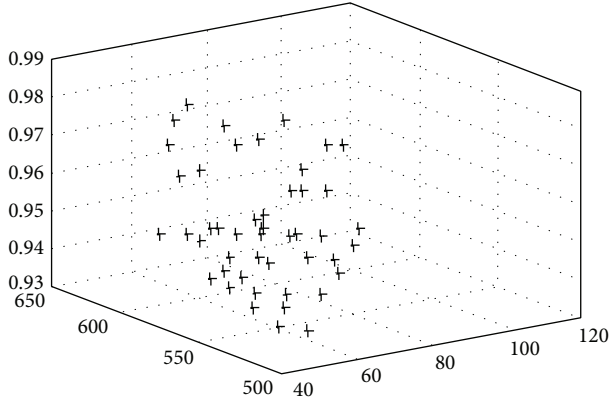


FIGURE 7: The Pareto frontier of optimal solution sets.

The upper level optimization goal is to select the upper level to consider the resource suppliers' comprehensive foreground value to be the smallest and the lower layer to consider the resource supply-side demand maximization, while the upper and lower layer constraints are. There are no interactions between the constraints imposed by the resource suppliers and the resource customers. The upper level and lower level give their own constraints, but the constraints are given by the MRD and the MRP, respectively, so there is no interaction between the constraints. We add a constraint condition which is strongly related to the upper level optimization object to the lower-level. A new bilevel programming model is established as follows:

$$(U) \min \quad Z = w_c \times \frac{V(C)}{V(C)_{\max}} + w_T \times \frac{V(T)}{V(T)_{\max}} \\ + w_Q \times \frac{V(Q)}{V(Q)_{\max}} + w_G \times \frac{V(G)}{(VG)_{\max}}$$

$$\begin{aligned} \text{s.t.} \quad & V(C_{ij}) \leq V(C)_{\max}, \\ & V(T_{ij}) \leq V(T)_{\max}, \\ & V(Q_{ij}) \geq V(Q)_{\min}, \\ & V(G_{ij}) \geq V(G)_{\min}, \end{aligned} \quad (49)$$

$$(L) \max \quad z = (E, F, R)^T \quad (50)$$

$$\text{s.t.} \quad V_{ij} \leq V_{\max}, \quad (51)$$

$$E_{ij}^n \geq E_{\min}^n, \quad (52)$$

$$E_{ij}^f \geq E_{\min}^f, \quad (53)$$

$$F_{ij}^d \geq F_{\min}^d, \quad (54)$$

$$F_{ij}^z \geq F_{\min}^z, \quad (55)$$

$$R_{ij}^k \geq R_{\min}^k, \quad (56)$$

$$R_{ij}^x \geq R_{\min}^x. \quad (57)$$

In the lower constraint, add (51) which represent the foreground value constraint of the MRD for each manufacturing unit. By adding (51) to the lower constraint conditions, when the MDP is optimizing the manufacturing resource configuration, it is a priority to discharge the prospect value of the MRD, thus simplifying the complexity of computing, speeding up the calculation speed, and further improving the interactivity between the upper level and the lower level.

8. Conclusions

The traditional discrete manufacturing object decision-making method only considers the requirements of the MRD and ignores the practical difficulties of the MRP or only from the perspective of the MRP to provide self-perceived the MRD satisfaction and then through the linear weighting method to convert the multiobjective optimization problem into a single-objective optimization problem to solve. This paper analyzes the importance and current deficiencies of manufacturing resource configuration in a discrete manufacturing environment and considers uncertainties such as manufacturing resource changes and manufacturing task changes. On this basis, according to the process and characteristics of manufacturing resource configuration in a discrete manufacturing environment, and from the interests of different participants, a bilevel programming mathematical model for manufacturing resource optimization configuration was built. It not only ensures the interests of different participants but also ensures the smooth progress of the manufacturing service. Finally, it uses NSGA-II to solve the model. The method presented in this paper has the characteristics of clear concept and simple calculation process and has strong operability and practicality. It provides a new way to solve the problem of multiobjective resource optimization configuration.

TABLE 7: Discrete manufacturing optimal solution sorting of NSGA-II.

Sequence	Manufacturing resource configuration plan	U	L
1	MS ₁₁ , MS ₂₅ , MS ₃₄ , MS ₄₂ , MS ₅₈ , MS ₆₆ , MS ₈₇	1.5938	1.4936
2	MS ₁₅ , MS ₂₈ , MS ₃₁ , MS ₄₁ , MS ₅₅ , MS ₇₄ , MS ₈₃	1.5940	1.4867
3	MS ₁₃ , MS ₂₃ , MS ₃₄ , MS ₄₂ , MS ₅₅ , MS ₇₃ , MS ₈₃	1.5939	1.4902
4	MS ₁₅ , MS ₂₂ , MS ₃₁ , MS ₄₄ , MS ₅₈ , MS ₆₆ , MS ₈₁	1.5940	1.4897
5	MS ₁₃ , MS ₂₅ , MS ₃₄ , MS ₄₂ , MS ₅₅ , MS ₆₅ , MS ₈₁	1.5937	1.4899

TABLE 8: The order of optimal solutions obtained by the exhaustion method.

Sequence	Manufacturing resource configuration plan	U	L
1	MS ₁₁ , MS ₂₅ , MS ₃₄ , MS ₄₂ , MS ₅₈ , MS ₆₆ , MS ₈₇	1.5938	1.4936
2	MS ₁₅ , MS ₂₂ , MS ₃₁ , MS ₄₄ , MS ₅₈ , MS ₆₆ , MS ₈₁	1.5938	1.4891
3	MS ₁₅ , MS ₂₈ , MS ₃₁ , MS ₄₁ , MS ₅₅ , MS ₇₄ , MS ₈₃	1.5940	1.4867
4	MS ₁₃ , MS ₂₃ , MS ₃₄ , MS ₄₂ , MS ₅₅ , MS ₇₃ , MS ₈₃	1.5939	1.4902
5	MS ₁₅ , MS ₂₅ , MS ₃₁ , MS ₄₁ , MS ₅₅ , MS ₇₄ , MS ₈₃	1.5937	1.4871
6	MS ₁₃ , MS ₂₃ , MS ₃₄ , MS ₄₂ , MS ₅₆ , MS ₇₃ , MS ₈₇	1.5931	1.4894
7	MS ₁₅ , MS ₂₂ , MS ₃₁ , MS ₄₄ , MS ₅₈ , MS ₆₆ , MS ₈₁	1.5940	1.4897
8	MS ₁₁ , MS ₂₈ , MS ₃₄ , MS ₄₂ , MS ₅₅ , MS ₆₆ , MS ₈₇	1.5934	1.4937
9	MS ₁₃ , MS ₂₅ , MS ₃₄ , MS ₄₂ , MS ₅₅ , MS ₆₅ , MS ₈₁	1.5937	1.4899
10	MS ₁₃ , MS ₂₂ , MS ₃₄ , MS ₄₂ , MS ₅₈ , MS ₇₃ , MS ₈₇	1.5936	1.4890

Data Availability

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

Conflicts of Interest

The authors of this study state that there are no conflicts of interest to disclose.

Acknowledgments

This work was financially supported by Liaoning Province Planning Office of Philosophy and Social Science (L18BJY029), Shenyang Planning Office of Philosophy and Social Science (18ZX019), the Liaoning Province Department of Education Project (WGD2016002), Shenyang Science and Technology Innovation Knowledge Base (SYKJ201807), and Shenyang Federation Social Science Circles (Grant no. SYSK2018-05-04). The authors wish to acknowledge the contribution of Liaoning Key Lab of Equipment Manufacturing Engineering Management, Liaoning Research Base of Equipment Manufacturing Development, Liaoning Key Research Base of Humanities and Social Sciences, Research Center of Micromanagement Theory, and Shenyang University of Technology.

References

- [1] H. Zheng, Y. Feng, and J. Tan, "A fuzzy QoS-aware resource service selection considering design preference in cloud manufacturing system," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1–4, pp. 371–379, 2016.
- [2] Y. C. Li and S. H. Hong, "Real-time demand bidding for energy management in discrete manufacturing facilities," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 739–749, 2017.
- [3] A. Sharif Ahmadian, M. Hosseingholi, and A. Ejlali, "Discrete feedback-based dynamic voltage scaling for safety critical real-time systems," *Scientia Iranica*, vol. 20, no. 3, pp. 647–656, 2013.
- [4] K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, "Social cloud computing: a vision for socially motivated resource sharing," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 551–563, 2012.
- [5] Y. Zhang, D. Xi, R. Li, and S. Sun, "Task-driven manufacturing cloud service proactive discovery and optimal configuration method," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1–4, pp. 29–45, 2016.
- [6] H. Zhang and Y. Hu, "A hybrid chaotic quantum evolutionary algorithm for resource combinatorial optimization in manufacturing grid system," *International Journal of Advanced Manufacturing Technology*, vol. 52, no. 5–8, pp. 821–831, 2011.
- [7] S. Li, S. Si, H. Dui, Z. Cai, and S. Sun, "A novel decision diagrams extension method," *Reliability Engineering and System Safety*, vol. 126, no. 126, pp. 107–115, 2014.
- [8] F. Xiang, Y. Hu, Y. Yu, and H. Wu, "QoS and energy consumption aware service composition and optimal-selection based on Pareto group leader algorithm in cloud manufacturing system," *Central European Journal of Operations Research*, vol. 22, no. 4, pp. 663–685, 2014.

- [9] F. Tao, Y. Laili, L. Xu, and L. Zhang, "FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2023–2033, 2013.
- [10] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: key to future manufacturing," *Journal of Intelligent Manufacturing*, vol. 11, no. 4, pp. 403–419, 2000.
- [11] M. M. Gupta, "Fuzzy set theory and its applications," *Fuzzy Sets and Systems*, vol. 47, no. 3, pp. 396–397, 1992.
- [12] S. K. Das, T. Mandal, and S. A. Edalatpanah, "A mathematical model for solving fully fuzzy linear programming problem with trapezoidal fuzzy numbers," *Applied Intelligence*, vol. 46, no. 3, pp. 509–519, 2017.
- [13] D. Takači, A. Takači, and A. Takači, "On the operational solutions of fuzzy fractional differential equations," *Fractional Calculus and Applied Analysis*, vol. 17, no. 4, pp. 99–107, 2014.
- [14] L. Jorba and R. Adillon, "A generalization of trapezoidal fuzzy numbers based on modal interval theory," *Symmetry*, vol. 9, no. 10, p. 198, 2017.
- [15] B. Farhadinia, "Multiple criteria decision-making methods with completely unknown weights in hesitant fuzzy linguistic term setting," *Knowledge-Based Systems*, vol. 93, pp. 135–144, 2016.
- [16] Y. Xu, A. Xu, J. M. Merigó, and H. Wang, "Hesitant fuzzy linguistic ordered weighted distance operators for group decision making," *Journal of Applied Mathematics and Computing*, vol. 49, no. 1-2, pp. 285–308, 2015.
- [17] A. Tversky and D. Kahneman, "Advances in prospect theory: cumulative representation of uncertainty," *Journal of Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.
- [18] D. Kahneman and A. Tversky, "Prospect theory: an analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–291, 1979.
- [19] Y. Wu, C. Xu, and T. Zhang, "Evaluation of renewable power sources using a fuzzy MCDM based on cumulative prospect theory: a case in China," *Energy*, vol. 147, pp. 1227–1239, 2018.
- [20] P. Bromiley, "A Prospect theory model of resource allocation," *Decision Analysis*, vol. 6, no. 3, pp. 124–138, 2009.
- [21] A. S. Booi, B. M. S. van Praag, and G. van de Kuilen, "A parametric analysis of prospect theory's functionals for the general population," *Theory and Decision*, vol. 68, no. 1-2, pp. 115–148, 2010.
- [22] M. Zhao, J. Gong, L. Wang, DO. Mathematics, and DM. University, "Stochastic multiple attribute decision-making model of grey interval linguistic variable based on prospect theory," *Statistics and Decision*, vol. 23, pp. 45–48, 2017.
- [23] C. Zhou, W. Tang, and R. Zhao, "Optimal consumer search with prospect utility in hybrid uncertain environment," *Journal of Uncertainty Analysis and Applications*, vol. 3, no. 1, p. 6, 2015.
- [24] Y. Kristianto, P. Helo, and R. J. Jiao, "Mass customization design of engineer-to-order products using benders' decomposition and bi-level stochastic programming," *Journal of Intelligent Manufacturing*, vol. 24, no. 5, pp. 961–975, 2013.
- [25] S. H. Wu, Z. Z. Yang, and X. D. Dong, "Optimization method for location and distribution of manufacturing industry based on the bi-level programming model," *Systems Engineering-Theory and Practice*, vol. 35, no. 11, pp. 2840–2848, 2015.
- [26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGAII," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [27] M. G. Gong, L. C. Jiao, D. D. Yang, and W. P. Ma, "Research on evolutionary multi-objective optimization algorithms," *Journal of Software*, vol. 20, no. 2, pp. 271–289, 2009.
- [28] L. Weining, L. Bo, and S. Dihua, "Multi-task oriented service composition in cloud manufacturing," *Computer Integrated Manufacturing Systems*, vol. 19, no. 1, pp. 199–209, 2013.
- [29] S. Kaikai, X. Wensheng, and L. Jiangyong, "Manufacturing resource allocation method based on bi-level programming in cloud manufacturing," *Computer Integrated Manufacturing Systems*, vol. 21, no. 7, pp. 1941–1951, 2015.

Research Article

Towards Reduced-Order Models of Solid Oxide Fuel Cell

Maciej Ławryńczuk 

*Institute of Control and Computation Engineering, Faculty of Electronics and Information Technology,
Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland*

Correspondence should be addressed to Maciej Ławryńczuk; m.lawrynczuk@ia.pw.edu.pl

Received 24 January 2018; Accepted 6 February 2018; Published 4 July 2018

Academic Editor: Sing Kiong Nguang

Copyright © 2018 Maciej Ławryńczuk. 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 objective of this work is to find precise reduced-order discrete-time models of a solid oxide fuel cell, which is a multiple-input multiple-output dynamic process. At first, the full-order discrete-time model is found from the continuous-time first-principle description. Next, the discrete-time submodels of hydrogen, oxygen, and water pressures (intermediate variables) are reduced. Two model reduction methods based on observability and controllability Grammians are compared: the state truncation method and reduction by residualisation. In all comparisons, the second method gives better results in terms of dynamic and steady-state errors as well as Nyquist plots. Next, the influence of the order of the pressure models on the errors of the process outputs (the voltage and the pressure difference) is studied. It is found that the number of pressure model parameters may be reduced from 25 to 19 without any deterioration of model accuracy. Two suboptimal reduced models are also discussed with only 14 and 11 pressure parameters, which give dynamic trajectories and steady-state characteristics that are very similar to those obtained from the full-order structure.

1. Introduction

There are three important reasons why renewable energy sources are becoming more and more popular. Firstly, burning of fossil fuels leads to air pollution and significant climate changes. Secondly, both mentioned phenomena badly affect public health, which has a very negative effect on the economy. Thirdly, fossil fuels are located in some countries only, whereas sources of renewable energy actually exist practically in all countries (of course not all of them are possible in all locations). Access to energy sources naturally increases national security. The countries that do not have fossil fuels may switch to renewable energy and become independent of other countries. As a result, the role of renewable energy is important and it is expected to grow fast in the future [1, 2]. Usually, renewable energy is obtained from wind turbines [3], geothermal systems [4], solar collectors [5], marine systems [6], and biofuels [7]. Additionally, solid oxide fuel cells (SOFCs) are very promising sources of energy. SOFCs are electrochemical devices that are able to directly convert the chemical energy stored in hydrocarbon fuels into electrical energy [8, 9]. They have many advantages, namely, high electrical efficiency, fuel flexibility, low emission,

quiet operation, and relatively low cost. That is why SOFCs are expected to become sound alternatives to conventional power generation schemes not only for domestic but also for commercial and industrial sectors.

Economically efficient and technologically safe operation of SOFCs requires well-designed control algorithms. Control of SOFCs is a challenging task, since they are nonlinear dynamic systems and it is essential to precisely satisfy some technological constraints that must be imposed on process variables [10]. Hence, for controlling SOFCs, advanced Model Predictive Control (MPC) algorithms are preferred rather than classical Proportional-Integral-Derivative (PID) controllers. An important feature of MPC is the fact that a mathematical model of the controlled SOFC is used online to successively calculate the best possible sequence of manipulated variable(s). In the simplest case, for prediction in MPC, linear models of the process may be used [11, 12] and the resulting control quality is better than that of the classical PID. However, in order to obtain very good control quality, a nonlinear dynamic model of the SOFC must be used in MPC. Different variants of nonlinear MPC algorithms for the SOFC are discussed in [13–16] (different model structures and optimisation algorithms are possible). Important applications

of the mathematical model of the SOFC also include process optimisation [17], fault tolerant control [15], and estimation [16, 18].

In MPC, optimisation, fault tolerant control, and estimation, different model structures may be used. Firstly, the first-principle model based on technological laws may be used [13, 16]. Secondly, empirical (black-box) models are possible, for example, neural networks [14, 19] or fuzzy systems [20, 21]. Although it may be easier to use empirical models than rigorous first-principle ones, it is necessary to point out three important disadvantages of black-box structures. Empirical structures make it possible to predict the sequence of the output variable(s) for a given sequence of the input and disturbance variable(s), but some intermediate process variables are usually not modelled. Moreover, frequently used black-box models typically have numerous parameters, much more than the fundamental ones. Finally, accuracy of black-box models may be good in typical operating conditions, but for other ones they are likely to generate output value(s) far from those calculated by the fundamental models (and the real process).

This work is concerned with the fundamental model of the SOFC, which is a multiple-input multiple-output nonlinear dynamic process. The objective is to find precise reduced-order discrete-time models. To achieve this goal, the discrete-time submodels of hydrogen, oxygen, and water pressures (intermediate variables) are reduced by means of two methods. In the first approach, the balanced Grammian of the state-space realisation is found and the state variables corresponding to small entries of the Grammian are removed. In the second reduction method, the model parameters are additionally adjusted in such a way that the steady-state gain of the reduced model is equal to that of the full-order one. The reduced models are compared in terms of dynamic and steady-state errors as well as Nyquist plots. Next, the influence of the order of the pressure models on the errors of the process outputs (the voltage and the pressure difference) is studied. The ideal reduced model and suboptimal ones, which give good compromise between accuracy and complexity, are discussed and compared with the full-order structure.

This work is structured as follows. The SOFC is shortly described in Section 2 and its full-order continuous-time model is detailed in Section 3. Section 4 derives the full-order discrete-time model. Section 5 discusses two model reduction methods. The main part of the paper, presented in Section 6, at first details reduction of the hydrogen, oxygen, and water pressure models and next studies the influence of the reduced pressure models on modelling accuracy of two process outputs. Finally, Section 7 concludes the paper.

2. SOFC System Description

The literature concerned with first-principle modelling of SOFCs is rich, for example, [9, 22–24]. The fundamental model of the SOFC introduced in [25] and next discussed in [12, 26] is considered here. The following assumptions are made:

- (1) The gases are ideal.

- (2) The stack is fed with hydrogen and air.
- (3) The channels that transport gases along the electrodes have a fixed volume, but their lengths are small. Hence, it is only necessary to define one single pressure value in their interior.
- (4) The exhaust of each channel is via a single orifice. The ratio of pressures between the interior and exterior of the channel is large enough to consider that the orifice is choked.
- (5) The temperature is stable at all times.
- (6) Because the working conditions are not close to the upper and lower extremes of current, the only source of losses is ohmic losses.
- (7) The Nernst equation can be applied.

The considered SOFC has two manipulated variables (the inputs of the process): the input gas flow rate q_f^{in} (mol s⁻¹) and the input oxygen flow rate $q_{\text{O}_2}^{\text{in}}$ (mol s⁻¹); one disturbance (the uncontrolled input) I_{in} which is the external current load (A) and four controlled variables (the outputs of the process): the stack output voltage V_r (V), fuel utilisation U_f (–), the fuel cell pressure difference p_{dif} (atm) between the hydrogen and oxygen passing through the anode and cathode gas compartments, and the ratio $R_{\text{H}_2\text{O}}$ (–) between hydrogen and oxygen flow rates. The partial pressures of hydrogen, oxygen, and water are denoted by p_{H_2} , p_{O_2} , and $p_{\text{H}_2\text{O}}$, respectively (atm). The input hydrogen flow and the hydrogen reacted flow are denoted by $q_{\text{H}_2}^{\text{in}}$ and $q_{\text{H}_2}^r$, respectively (mol s⁻¹).

3. Continuous-Time Model

Figure 1 depicts the structure of the fundamental continuous-time model of the SOFC system. The pressure of hydrogen is

$$p_{\text{H}_2} = \frac{1/K_{\text{H}_2}}{\tau_{\text{H}_2}s + 1} (q_{\text{H}_2}^{\text{in}} - 2K_r I_r), \quad (1)$$

where the input hydrogen flow is

$$q_{\text{H}_2}^{\text{in}} = \frac{1}{\tau_f s + 1} q_f^{\text{in}}. \quad (2)$$

Hence, the pressure of hydrogen is

$$p_{\text{H}_2} = \frac{1/K_{\text{H}_2}}{\tau_{\text{H}_2}s + 1} \left(\frac{1}{\tau_f s + 1} q_f^{\text{in}} - 2K_r \frac{1}{\tau_e s + 1} I_{\text{in}} \right), \quad (3)$$

where K_{H_2} , τ_{H_2} , τ_f , and τ_e denote the valve molar constant for hydrogen, the response time of hydrogen flow, the fuel processor time constant, and the electrical time constant, respectively. The pressure of oxygen is

$$\begin{aligned} p_{\text{O}_2} &= \frac{1/K_{\text{O}_2}}{\tau_{\text{O}_2}s + 1} (q_{\text{O}_2}^{\text{in}} - K_r I_r) \\ &= \frac{1/K_{\text{O}_2}}{\tau_{\text{O}_2}s + 1} \left(q_{\text{O}_2}^{\text{in}} - K_r \frac{1}{\tau_e s + 1} I_{\text{in}} \right), \end{aligned} \quad (4)$$

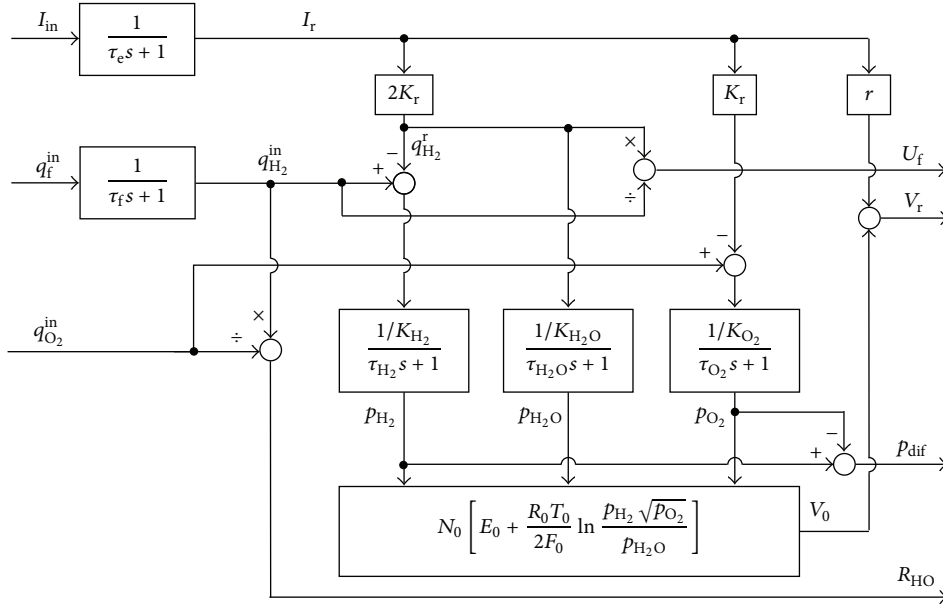


FIGURE 1: The structure of the fundamental continuous-time model of the SOFC system.

where K_{O_2} and τ_{O_2} denote the valve molar constant for oxygen and the response time of oxygen flow, respectively. The pressure of water is

$$P_{H_2O} = \frac{1/K_{H_2O}}{\tau_{H_2O}s + 1} q_{H_2}^r, \quad (5)$$

where the hydrogen flow that reacts is

$$q_{H_2}^r = 2K_r I_r = \frac{2K_r}{\tau_e s + 1} I_{in}. \quad (6)$$

Hence, the pressure of water is

$$P_{H_2O} = \frac{2K_r/K_{H_2O}}{(\tau_e s + 1)(\tau_{H_2O}s + 1)} I_{in}, \quad (7)$$

where K_{H_2O} and τ_{H_2O} denote the valve molar constant for water and the response time of water flow, respectively. Finally, outputs of the process are defined. Applying Nernst's equation and taking into account ohmic losses, the stack output voltage is

$$V_r = N_0 \left[E_0 + \frac{R_0 T_0}{2F_0} \ln \frac{P_{H_2} \sqrt{P_{O_2}}}{P_{H_2O}} \right] - I_r r, \quad (8)$$

where N_0 , E_0 , R_0 , T_0 , and F_0 denote the number of cells in series in the stack, the ideal standard potential, the universal gas constant, the absolute temperature, and Faraday's constant, respectively. Fuel utilisation is defined as the ratio between the hydrogen flow that reacts and the input hydrogen flow.

$$U_f = \frac{q_{H_2}^r}{q_{H_2}^{in}}. \quad (9)$$

The ratio between hydrogen and oxygen flow rates is

$$R_{HO} = \frac{q_{H_2}^{in}}{q_{O_2}^{in}}. \quad (10)$$

The pressure difference between the hydrogen and oxygen passing through the anode and cathode gas compartments is

$$P_{dif} = P_{H_2} - P_{O_2}. \quad (11)$$

All things considered, the continuous-time fundamental model consists of (2), (3), (4), (6), (7), (8), (9), (10), and (11). The values of the model parameters are given in Table 1. Table 2 gives values of process variables for the initial operating point. The values of process inputs are constrained: $0 \text{ mol s}^{-1} \leq q_f^{in} \leq 1.7023 \text{ mol s}^{-1}$ and $0 \text{ mol s}^{-1} \leq q_{O_2}^{in} \leq 1.6134 \text{ mol s}^{-1}$; the value of the disturbance is also limited: $200 \text{ A} \leq I_{in} \leq 300 \text{ A}$.

4. Model Discretisation

Discrete-time versions of the continuous-time transfer functions (3), (4), and (7) must be found. For this purpose, the continuous-time polynomials of the proper order of the variable s must be determined. From (3), one has

$$P_{H_2} = \frac{1/K_{H_2}}{\tau_{H_2} \tau_f s^2 + (\tau_{H_2} + \tau_f) s + 1} q_f^{in} - \frac{2K_r/K_{H_2}}{\tau_{H_2} \tau_e s^2 + (\tau_{H_2} + \tau_e) s + 1} I_{in} \quad (12)$$

From (4),

$$P_{O_2} = \frac{1/K_{O_2}}{\tau_{O_2} s + 1} q_{O_2}^{in} - \frac{K_r/K_{O_2}}{\tau_{O_2} \tau_e s^2 + (\tau_{O_2} + \tau_e) s + 1} I_{in} \quad (13)$$

TABLE 1: Parameters of the fundamental continuous-time model of the SOFC system.

Parameter	Value	Unit	Description
E_0	1.18	V	Ideal standard potential
F_0	96.485	C mol ⁻¹	Faraday's constant
K_{H_2}	8.43×10^{-1}	mol s ⁻¹ atm ⁻¹	Valve molar constant for hydrogen
K_{H_2O}	2.81×10^{-1}	mol s ⁻¹ atm ⁻¹	Valve molar constant for water
K_{O_2}	2.52	mol s ⁻¹ atm ⁻¹	Valve molar constant for oxygen
K_r	0.996×10^{-3}	mol s ⁻¹ A ⁻¹	Constant, $K_r = N_0/4F_0$
N_0	384	–	Number of cells in series in the stack
R_0	8.314	J mol ⁻¹ K ⁻¹	Universal gas constant
T_0	1273	K	Absolute temperature
r	0.126	Ω	Ohmic loss
τ_e	0.8	s	Electrical time constant
τ_f	5	s	The fuel processor time constant
τ_{H_2}	26.1	s	Response time of hydrogen flow
τ_{H_2O}	78.3	s	Response time of water flow
τ_{O_2}	2.91	s	Response time of oxygen flow

TABLE 2: The values of process variables for the initial operating point.

Variable	Value	Unit
q_f^{in}	0.7023	mol s ⁻¹
$q_{O_2}^{\text{in}}$	0.6134	mol s ⁻¹
I_{in}	300	A
$q_{H_2}^{\text{in}}$	0.7023	mol s ⁻¹
$q_{H_2}^r$	5.976000×10^{-1}	mol s ⁻¹
p_{H_2}	1.241993×10^{-1}	atm
p_{O_2}	1.248413×10^{-1}	atm
p_{H_2O}	2.126690	atm
V_r	3.335865×10^2	V
U_f	8.509184×10^{-1}	–
R_{HO}	1.144930	–
p_{dif}	-6.419816×10^{-4}	atm

From (7),

$$p_{H_2O} = \frac{2K_r/K_{H_2O}}{\tau_e \tau_{H_2O} s^2 + (\tau_e + \tau_{H_2O}) s + 1} I_{\text{in}}. \quad (14)$$

Next, model equations are discretised using the first-order holder with the sampling period equal to 1 second. The discrete-time version of (12) is

$$p_{H_2}(k) = \frac{\beta_1^1 q^{-1} + \beta_2^1 q^{-2}}{1 + \alpha_1^1 q^{-1} + \alpha_2^1 q^{-2}} q_f^{\text{in}}(k) + \frac{\beta_1^2 q^{-1} + \beta_2^2 q^{-2}}{1 + \alpha_1^2 q^{-1} + \alpha_2^2 q^{-2}} I_{\text{in}}(k), \quad (15)$$

where k denotes the discrete sampling instant and q^{-1} is a unit delay operator. The discrete-time version of (13) is

$$p_{O_2}(k) = \frac{\beta_1^3 q^{-1}}{1 + \alpha_1^3 q^{-1}} q_{O_2}^{\text{in}}(k) + \frac{\beta_1^4 q^{-1} + \beta_2^4 q^{-2}}{1 + \alpha_1^4 q^{-1} + \alpha_2^4 q^{-2}} I_{\text{in}}(k). \quad (16)$$

The discrete-time version of (14) is

$$p_{H_2O}(k) = \frac{\beta_1^5 q^{-1} + \beta_2^5 q^{-2}}{1 + \alpha_1^5 q^{-1} + \alpha_2^5 q^{-2}} I_{\text{in}}(k). \quad (17)$$

Additionally, the discrete-time version of (2)

$$q_{H_2}^{\text{in}}(k) = \frac{\beta_1^6 q^{-1}}{1 + \alpha_1^6 q^{-1}} q_f^{\text{in}}(k) \quad (18)$$

and the discrete-time version of (6)

$$q_{H_2}^r(k) = \frac{\beta_1^7 q^{-1}}{1 + \alpha_1^7 q^{-1}} I_{\text{in}}(k) \quad (19)$$

must be taken into account. The final form of the discrete-time fundamental model is as follows. From the discrete transfer functions (15), (16), (17), (18), and (19), direct dependence of current values of process variables (i.e., for the current sampling instant k) as functions of corresponding variables in the previous instants is found. From (15), one has

$$p_{H_2}(k) = \sum_{i=1}^{n_{p_{H_2}}} b_i^{1,1} q_f^{\text{in}}(k-i) + \sum_{i=1}^{n_{p_{H_2}}} b_i^{1,2} I_{\text{in}}(k-i) - \sum_{i=1}^{n_{p_{H_2}}} a_i^1 p_{H_2}(k-i), \quad (20)$$

where $b_1^{1,1} = \beta_1^1$, $b_2^{1,1} = \beta_2^1 + \alpha_1^1 \beta_1^1$, $b_3^{1,1} = \alpha_1^2 \beta_2^1 + \alpha_2^2 \beta_1^1$, $b_4^{1,1} = \alpha_2^2 \beta_2^1$, $b_1^{1,2} = \beta_1^2$, $b_2^{1,2} = \beta_2^2 + \alpha_1^1 \beta_1^2$, $b_3^{1,2} = \alpha_1^1 \beta_2^2 + \alpha_2^1 \beta_1^2$, $b_4^{1,2} =$

TABLE 3: Parameters of the full-order discrete-time fundamental model.

Equation	Parameters		
(20)	$a_1^1 = -3.030057$ $a_2^1 = 3.288185$ $a_3^1 = -1.475212$ $a_4^1 = 2.172670 \times 10^{-1}$	$b_1^{1,1} = 4.201819 \times 10^{-3}$ $b_2^{1,1} = -1.366717 \times 10^{-3}$ $b_3^{1,1} = -3.688449$ $b_4^{1,1} = 1.070128 \times 10^{-3}$	$b_1^{1,2} = -3.832050 \times 10^{-5}$ $b_2^{1,2} = 4.319957 \times 10^{-5}$ $b_3^{1,2} = 1.443104 \times 10^{-5}$ $b_4^{1,2} = -1.974194 \times 10^{-5}$
(21)	$a_1^2 = -1.704879$ $a_2^2 = 9.093077 \times 10^{-1}$ $a_3^2 = -1.440946 \times 10^{-1}$	$b_1^{2,1} = 1.154038 \times 10^{-1}$ $b_2^{2,1} = -1.149061 \times 10^{-1}$ $b_3^{2,1} = 2.344823 \times 10^{-2}$	$b_1^{2,2} = -5.160259 \times 10^{-5}$ $b_2^{2,2} = 6.187518 \times 10^{-6}$ $b_3^{2,2} = 2.156491 \times 10^{-5}$
(22)	$a_1^3 = -1.273815$ $a_2^3 = 2.828690 \times 10^{-1}$	$b_1^3 = 3.867789 \times 10^{-5}$ $b_2^3 = 2.550836 \times 10^{-5}$	
(23)	$a_1^4 = -8.187307 \times 10^{-1}$	$b_1^4 = 1.812692 \times 10^{-1}$	
(24)	$a_1^5 = -2.865048 \times 10^{-1}$	$b_1^5 = 1.421282 \times 10^{-3}$	

$\alpha_2^1 \beta_2^2, a_1^1 = \alpha_1^1 + \alpha_1^2, a_2^1 = \alpha_2^1 + \alpha_2^2 + \alpha_1^1 \alpha_1^2, a_3^1 = \alpha_1^1 \alpha_2^2 + \alpha_2^1 \alpha_1^2,$
and $a_4^1 = \alpha_2^1 \alpha_2^2$. From (16), one has

$$p_{O_2}(k) = \sum_{i=1}^{n_{p_{O_2}}} b_i^{2,1} q_{O_2}^{in}(k-i) + \sum_{i=1}^{n_{p_{O_2}}} b_i^{2,2} I_{in}(k-i) - \sum_{i=1}^{n_{p_{O_2}}} a_i^2 p_{O_2}(k-i), \quad (21)$$

where $b_1^{2,1} = \beta_1^3, b_2^{2,1} = \alpha_1^4 \beta_2^3, b_3^{2,1} = \alpha_2^4 \beta_1^3, b_1^{2,2} = \beta_1^4, b_2^{2,2} = \beta_2^4 + \alpha_1^3 \beta_1^4, b_3^{2,2} = \alpha_1^3 \beta_2^4, a_1^2 = \alpha_1^3 + \alpha_1^4, a_2^2 = \alpha_2^2 + \alpha_1^3 \alpha_1^4,$ and $a_3^2 = \alpha_1^3 \alpha_2^4$. From (17), one has

$$p_{H_2O}(k) = \sum_{i=1}^{n_{p_{H_2O}}} b_i^3 I_{in}(k-i) - \sum_{i=1}^{n_{p_{H_2O}}} a_i^3 p_{H_2O}(k-i), \quad (22)$$

where $b_1^3 = \beta_1^5, b_2^3 = \beta_2^5, a_1^3 = \alpha_1^5,$ and $a_2^3 = \alpha_2^5$. From (18), one has

$$q_{H_2}^{in}(k) = b_1^4 q_f^{in}(k-i) - a_1^4 q_{H_2}^{in}(k-i), \quad (23)$$

where $b_1^4 = \beta_1^6$ and $a_1^4 = \alpha_1^6$. From (19), one has

$$q_{H_2}^r(k) = b_1^5 I_{in}(k-i) - a_1^5 q_{H_2}^r(k-i), \quad (24)$$

where $b_1^5 = \beta_1^7$ and $a_1^5 = \alpha_1^7$. The discrete-time version of the Nernst equation (8) is

$$V_r(k) = N_0 \left[E_0 + \frac{R_0 T_0}{2F_0} \ln \frac{p_{H_2}(k) \sqrt{p_{O_2}(k)}}{p_{H_2O}(k)} \right] - I_r(k) r. \quad (25)$$

In the discrete-time domain fuel utilisation (9) is

$$U_f(k) = \frac{q_{H_2}^r(k)}{q_{H_2}^{in}(k)} \quad (26)$$

and the ratio between hydrogen and oxygen flow rates (10) is

$$R_{HO}(k) = \frac{q_{H_2}^{in}(k)}{q_{O_2}^{in}(k)} \quad (27)$$

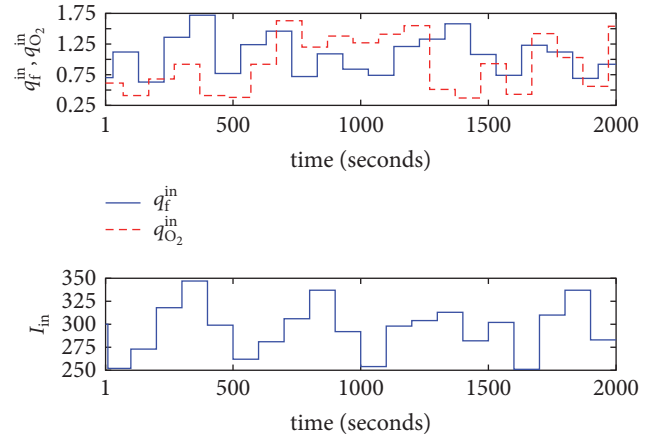


FIGURE 2: The sequences of the manipulated variables and the disturbance used for dynamic simulations.

and the pressure difference (11) is

$$p_{dif}(k) = p_{H_2}(k) - p_{O_2}(k). \quad (28)$$

All things considered, the full-order discrete-time model is defined by (20), (21), (22), (23), (24), (25), (26), (27), and (28). The order of model dynamics is defined by the integer numbers: $n_{p_{H_2}} = 4, n_{p_{O_2}} = 3,$ and $n_{p_{H_2O}} = 2$. Parameters of the full-order fundamental discrete-time model are given in Table 3.

To demonstrate effectiveness of the considered model discretisation method, two full-order dynamic models are simulated: the continuous-time one and the discrete-time one. The continuous-time model is simulated in Simulink; the differential equations are solved by the Runge-Kutta method of order 45 with a variable step size. The discrete-time model is implemented in MATLAB; the differential equations are solved with the constant step (equal to the sampling time). Both models are excited by the same series of step changes of the manipulated and disturbance variables depicted in Figure 2. Obtained trajectories of both dynamic systems (i.e., the sequences of three pressures and four process outputs) are compared in Figure 3. The discrete-time model gives

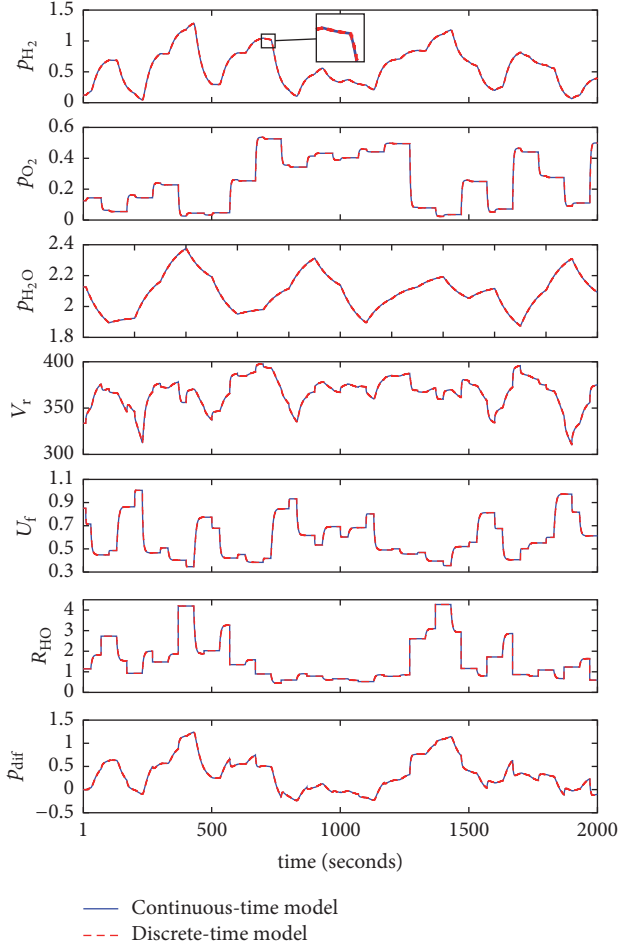


FIGURE 3: The comparison of the pressures and the outputs calculated from continuous-time and discrete-time models of full order.

practically the same responses as the continuous-time one; no important differences are present.

5. Model Reduction Methods

The state-space representation of a discrete-time dynamic system is considered:

$$\begin{aligned} x(k+1) &= \mathbf{A}x(k) + \mathbf{B}u(k), \\ y(k) &= \mathbf{C}x(k). \end{aligned} \quad (29)$$

It is assumed that matrix \mathbf{A} is asymptotically stable. The controllability, \mathbf{W}_c , and observability, \mathbf{W}_o , Grammians are [27]

$$\begin{aligned} \mathbf{W}_c &= \sum_{i=0}^{\infty} \mathbf{A}^i \mathbf{B} \mathbf{B}^T (\mathbf{A}^T)^i, \\ \mathbf{W}_o &= \sum_{i=0}^{\infty} (\mathbf{A}^T)^i \mathbf{C}^T \mathbf{C} \mathbf{A}^i. \end{aligned} \quad (30)$$

To reduce the order of the dynamic system (29), a balanced state-space representation must be used. For such a system, both Grammians are equal and diagonal:

$$\mathbf{W}_c = \mathbf{W}_o = \mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}. \quad (31)$$

The balanced representation of system (29) is found by setting matrix \mathbf{A} to $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$, setting matrix \mathbf{B} to $\mathbf{T}^{-1}\mathbf{B}$, and setting matrix \mathbf{C} to $\mathbf{C}\mathbf{T}$, where the transformation matrix \mathbf{T} may be found by the procedure described in [28, 29].

In this work, two model reduction algorithms [30] are considered: model reduction by state truncation (algorithm 1) and model reduction by residualisation (algorithm 2). In algorithm 1, the states for which the corresponding coefficients σ_i are relatively “small” are simply removed. Model (29) is reformulated:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} u(k), \quad (32)$$

$$y(k) = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix},$$

where the states x_1 and x_2 are associated with “big” and “small” values of σ_i , respectively. In the reduced model, the first state variables are only considered:

$$\begin{aligned} x_1(k+1) &= \mathbf{A}_{11}x_1(k) + \mathbf{B}_1u(k), \\ y(k) &= \mathbf{C}_1x_1(k). \end{aligned} \quad (33)$$

When some of state variables are simply removed from the model, the steady-state properties of the resulting model may significantly differ from those of the original one. To solve the problem, in algorithm 2, the states of model (32) for which the corresponding coefficients σ_i are “small” are also removed, but afterwards the model matrices are modified to guarantee that the steady-state gains of the full-order and reduced models are the same. Since state variables x_2 are assumed to be significantly faster than x_1 ones, it follows that $x_2(k+1) = x_2(k)$ and

$$x_2(k) = (\mathbf{I} - \mathbf{A}_{22})^{-1} (\mathbf{A}_{21}x_1(k) + \mathbf{B}_2u(k)). \quad (34)$$

Hence, the reduced model becomes

$$\begin{aligned} x_1(k+1) &= (\mathbf{A}_{11} + \mathbf{A}_{12}(\mathbf{I} - \mathbf{A}_{22})^{-1}\mathbf{A}_{21})x_1(k) \\ &\quad + (\mathbf{B}_1 + \mathbf{A}_{12}(\mathbf{I} - \mathbf{A}_{22})^{-1}\mathbf{B}_2)u(k), \\ y(k) &= (\mathbf{C}_1 + \mathbf{C}_2(\mathbf{I} - \mathbf{A}_{22})^{-1}\mathbf{A}_{21})x_1(k) \\ &\quad + \mathbf{C}_2(\mathbf{I} - \mathbf{A}_{22})^{-1}\mathbf{B}_2u(k). \end{aligned} \quad (35)$$

In the second algorithm in the output equation, there is a possible direct influence of the process input(s) on the

output(s), which is not necessary in the first algorithm. Therefore, the second algorithm leads to the difference equation for hydrogen pressure.

$$p_{\text{H}_2}(k) = \sum_{i=0}^{n_{\text{PH}_2}} b_i^{1,1} q_f^{\text{in}}(k-i) + \sum_{i=0}^{n_{\text{PH}_2}} b_i^{1,2} I_{\text{in}}(k-i) - \sum_{i=1}^{n_{\text{PH}_2}} a_i^1 p_{\text{H}_2}(k-i). \quad (36)$$

In contrast to the original full-order model and the reduced one obtained in the first algorithm (see (20)), the coefficients $b_0^{1,1}$ and $b_0^{1,2}$ may be necessary. Similarly, the second algorithm leads to the difference equation for oxygen pressure.

$$p_{\text{O}_2}(k) = \sum_{i=0}^{n_{\text{PO}_2}} b_i^{2,1} q_{\text{O}_2}^{\text{in}}(k-i) + \sum_{i=0}^{n_{\text{PO}_2}} b_i^{2,2} I_{\text{in}}(k-i) - \sum_{i=1}^{n_{\text{PO}_2}} a_i^2 p_{\text{O}_2}(k-i). \quad (37)$$

In contrast to the original full-order model and the reduced one obtained in the first algorithm (see (21)), the coefficients

$$g = [6.7211 \times 10^{-1} \quad 7.8946 \times 10^{-2} \quad 3.7688 \times 10^{-5} \quad 1.3656 \times 10^{-17}]. \quad (39)$$

The last two diagonal entries of the Gramian are small in comparison with the first two ones, which suggests that the last two state variables may be reduced. However, for a thorough analysis, the reduced models of the first, the second, and the third orders are considered; in each case, two reduction algorithms are used. The parameters of the reduced-order hydrogen pressure model of order $n_{\text{PH}_2} = 1, 2, 3$ calculated by the first algorithm are given in Table 4, whereas the results of the second algorithm are given in Table 5. The general form of the reduced models calculated by the first algorithm is the same as that of the full-order model (see (20)), whereas the reduced model determined by the second method is characterised by (36). In the latter case, the coefficients $b_0^{1,1}$ and $b_0^{1,2}$ may be necessary, which manifest a direct influence of the model input q_f^{in} and the disturbance I_{in} on the model output p_{H_2} . When the reduced model has the third order, the additional coefficients are not necessary to guarantee that the steady-state gain of the reduced model is the same as that of the full-order one. When the reduced model has the second order, only one of them is used, but in the case of the first-order model two of them are necessary.

To compare full-order and reduced models of the hydrogen pressure, it is convenient to consider their Nyquist plots. It is because Nyquist plots show both steady-state

$b_0^{2,1}$ and $b_0^{2,2}$ may be necessary. Finally, the second algorithm leads to the difference equation for water pressure.

$$p_{\text{H}_2\text{O}}(k) = \sum_{i=0}^{n_{\text{PH}_2\text{O}}} b_i^3 I_{\text{in}}(k-i) - \sum_{i=1}^{n_{\text{PH}_2\text{O}}} a_i^3 p_{\text{H}_2\text{O}}(k-i) \quad (38)$$

in which the coefficient b_0^3 may be necessary in contrast to the original full-order model and the reduced one generated by the first algorithm (see (22)).

6. Reduction of the Solid Oxide Fuel Cell Model

At first, separate reduction of hydrogen, oxygen, and water pressure models is considered. Next, the problem of finding the most appropriate orders of dynamics of these models is discussed taking into account two process outputs (the voltage and the pressure difference) whose modelling accuracy depends on the pressure models. Models of the remaining two process outputs, that is, fuel utilisation and the ratio between hydrogen and oxygen flow rates, are not reduced, since they are calculated from very simple equations, (26) and (27).

6.1. Reduction of the Hydrogen Pressure Model. The diagonal of the balanced Gramian of the state-space realisation corresponding to the discrete-time difference equation (20) describing hydrogen pressure is

and dynamic behaviours of models. Figure 4 depicts the Nyquist plots of reduced-order and full-order models. As expected, the first algorithm does not guarantee that the steady-state gain of the reduced model is exactly the same as that of the original one (the gain may be read for $\omega = 0$). Furthermore, for other frequencies, the Nyquist plot of the reduced models is different from that of the full-order one. This observation is true for the simplest first-order model and, for the disturbance I_{in} to output channel for the second-order structure, increasing the model order makes it possible to obtain precise steady-state gain. The second algorithm is much more efficient, since all reduced models, even the first-order one, have the same gain as the original one. Furthermore, the Nyquist plots of the reduced models are very similar to those of the full-order ones.

The models are also compared in a quantitative way. For this purpose, the dynamic model error is defined:

$$E_{p_{\text{H}_2}}^{\text{dyn}} = \frac{1}{n_{\text{dyn}}} \sum_{k=1}^{n_{\text{dyn}}} (p_{\text{H}_2}^{\text{full}}(k) - p_{\text{H}_2}^{\text{reduced}}(k))^2. \quad (40)$$

The outputs of the full-order and reduced models for the sampling instant k are denoted by $p_{\text{H}_2}^{\text{full}}(k)$ and $p_{\text{H}_2}^{\text{reduced}}(k)$, respectively. Both models are excited by the same series of

TABLE 4: Parameters of the reduced-order hydrogen pressure model of order $n_{p_{H_2}}$ calculated by the first algorithm.

$n_{p_{H_2}}$	Parameters		
1	$a_1^1 = -9.768754 \times 10^{-1}$	$b_1^{1,1} = 3.054446 \times 10^{-2}$	$b_1^{1,2} = -6.812056 \times 10^{-5}$
2	$a_1^1 = -1.781141$ $a_2^1 = 7.879551 \times 10^{-1}$	$b_1^{1,1} = 4.201816 \times 10^{-3}$ $b_2^{1,1} = 3.881002 \times 10^{-3}$	$b_1^{1,2} = -6.557908 \times 10^{-5}$ $b_2^{1,2} = 4.914773 \times 10^{-5}$
3	$a_1^1 = -2.067646$ $a_2^1 = 1.298261$ $a_3^1 = -2.257529 \times 10^{-1}$	$b_1^{1,1} = 4.201819 \times 10^{-3}$ $b_2^{1,1} = 2.677158 \times 10^{-3}$ $b_3^{1,1} = -1.111925 \times 10^{-3}$	$b_1^{1,2} = -3.832050 \times 10^{-5}$ $b_2^{1,2} = 6.319519 \times 10^{-6}$ $b_3^{1,2} = 2.051302 \times 10^{-5}$

TABLE 5: Parameters of the reduced-order hydrogen pressure model of order $n_{p_{H_2}}$ calculated by the second algorithm.

$n_{p_{H_2}}$	Parameters		
1	$a_1^1 = -9.651884 \times 10^{-1}$	$b_0^{1,1} = -1.345838 \times 10^{-1}$ $b_1^{1,1} = 1.758786 \times 10^{-1}$	$b_0^{1,2} = -3.381314 \times 10^{-5}$ $b_1^{1,2} = -4.844625 \times 10^{-5}$
2	$a_1^1 = -1.781141$ $a_2^1 = 7.879551 \times 10^{-1}$	$b_1^{1,1} = 4.201816 \times 10^{-3}$ $b_2^{1,1} = 3.881002 \times 10^{-3}$	$b_0^{1,2} = 3.820428 \times 10^{-5}$ $b_1^{1,2} = -1.519383 \times 10^{-4}$ $b_2^{1,2} = 9.763310 \times 10^{-5}$
3	$a_1^1 = -2.067646$ $a_2^1 = 1.298261$ $a_3^1 = -2.257529 \times 10^{-1}$	$b_1^{1,1} = 4.201819 \times 10^{-3}$ $b_2^{1,1} = 2.677158 \times 10^{-3}$ $b_3^{1,1} = -1.111925 \times 10^{-3}$	$b_1^{1,2} = -3.832050 \times 10^{-5}$ $b_2^{1,2} = 6.319519 \times 10^{-6}$ $b_3^{1,2} = 2.051302 \times 10^{-5}$

step changes of the manipulated variables and steps of the disturbance depicted in Figure 2; that is, $n_{\text{dyn}} = 2000$. Additionally, the steady-state model error is defined:

$$E_{p_{H_2}}^{\text{ss}} = \frac{1}{n_{\text{ss}}} \sum_{p=1}^{n_{\text{ss}}} \left(p_{H_2}^{\text{full}}(p) - p_{H_2}^{\text{reduced}}(p) \right)^2. \quad (41)$$

In this case, $p_{H_2}^{\text{full}}(p)$ and $p_{H_2}^{\text{reduced}}(p)$ denote outputs of the full-order and reduced steady-state models for the data point p , respectively. They are derived easily from the dynamic models. For the second algorithm, using (36), the dynamic model for hydrogen pressure is

$$p_{H_2} = \left(\sum_{i=0}^{n_{p_{H_2}}} b_i^{1,1} q_f^{\text{in}} + \sum_{i=0}^{n_{p_{H_2}}} b_i^{1,2} I_{\text{in}} \right) \left(1 + \sum_{i=1}^{n_{p_{H_2}}} a_i^1 \right)^{-1}. \quad (42)$$

In the case of the reduced model obtained by the first algorithm (see (20)), $b_0^{1,1} = b_0^{1,2} = 0$. For the steady-state models, $n_{\text{ss}} = 4116$. Dynamic and steady-state errors of the reduced models of hydrogen pressure are given in Table 6. As observed from the Nyquist plots, the second algorithm gives much better results (lower model errors). In particular, always $E_{p_{H_2}}^{\text{ss}} = 0$ for the second algorithm, since it always guarantees that the steady-state properties of the original and reduced models are the same. When the errors are lower than the machine accuracy (2.2204×10^{-16}), they are treated as 0. Both reduction algorithms give perfect models of order three.

Finally, Figure 5 compares the hydrogen pressure trajectories (model outputs) calculated by reduced-order ($n_{p_{H_2}} = 1, 2, 3$) and full-order ($n_{p_{H_2}} = 4$) models determined by both algorithms; all models are excited by the same series of step changes of the manipulated variables and steps of the disturbance depicted in Figure 2. As expected, taking

into account the Nyquist plots and model errors, the second algorithm gives much better results for the same model order. For the sampling instants $k = 230$ and $k = 231$, the simplest first-order model obtained by the second algorithm gives negative pressure (of order 10^{-2}). Both reduction algorithms give perfect models of order three; that is, there is no difference between their outputs and the output of the full-order model. Reduced models of order one and two calculated by algorithm 2 give very precise trajectories, whereas the corresponding models found by algorithm 1 have bigger errors but they are still acceptable.

6.2. Reduction of the Oxygen Pressure Model. The diagonal of the balanced Grammian of the state-space realisation corresponding to the discrete-time difference equation (21) describing oxygen pressure is

$$g = \left[2.3217 \times 10^{-1} \quad 6.1788 \times 10^{-5} \quad 1.2064 \times 10^{-20} \right]^T. \quad (43)$$

Although the last two diagonal entries of the Grammian are small in comparison with the first ones, which suggests that the last two state variables may be reduced, the reduced models of the first and the second orders are considered. The parameters of the reduced-order oxygen pressure model of order $n_{p_{H_2}} = 1, 2$ calculated by the first algorithm are given in Table 7, whereas the results of the second algorithm are given in Table 8. The first algorithm finds the models of a general structure the same as the full-order model (see (21)), whereas the reduced model of the first order determined by the second method is characterised by (37). In this case, the coefficient $b_0^{2,2}$ may be necessary, which manifests a direct influence of the disturbance I_{in} on the model output p_{O_2} . For the second-order reduced model, the additional coefficient

TABLE 6: The dynamic error $E_{p_{H_2}}^{\text{dyn}}$ and the steady-state error $E_{p_{H_2}}^{\text{ss}}$ of the reduced hydrogen pressure models ($n_{p_{H_2}} = 1, 2, 3$).

$n_{p_{H_2}}$	$E_{p_{H_2}}^{\text{dyn}}$		$E_{p_{H_2}}^{\text{ss}}$	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
1	2.5687×10^{-3}	1.1652×10^{-4}	1.7068×10^{-2}	0.0
2	2.0740×10^{-4}	3.5651×10^{-8}	4.1544×10^{-4}	0.0
3	0.0	0.0	0.0	0.0

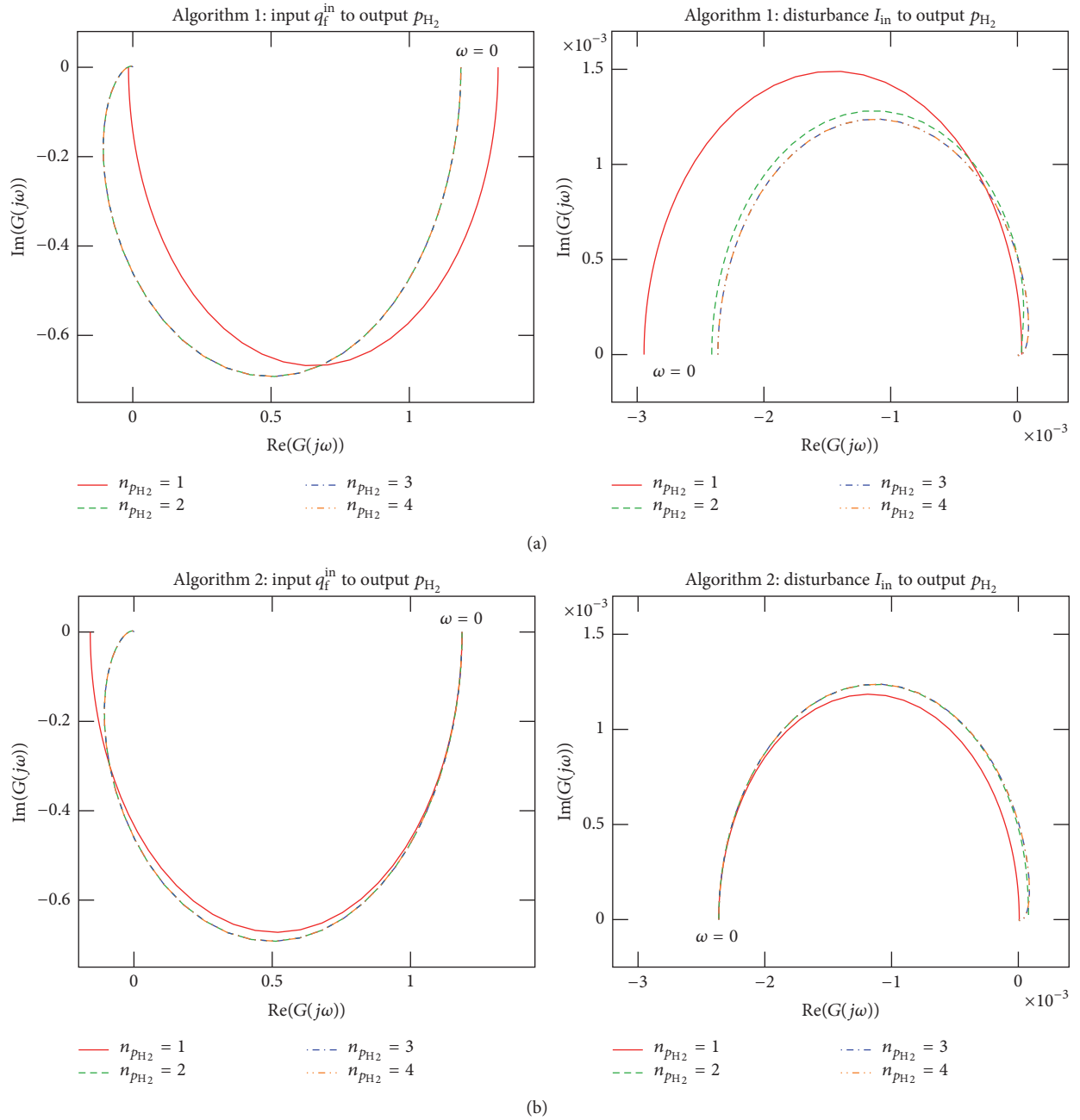
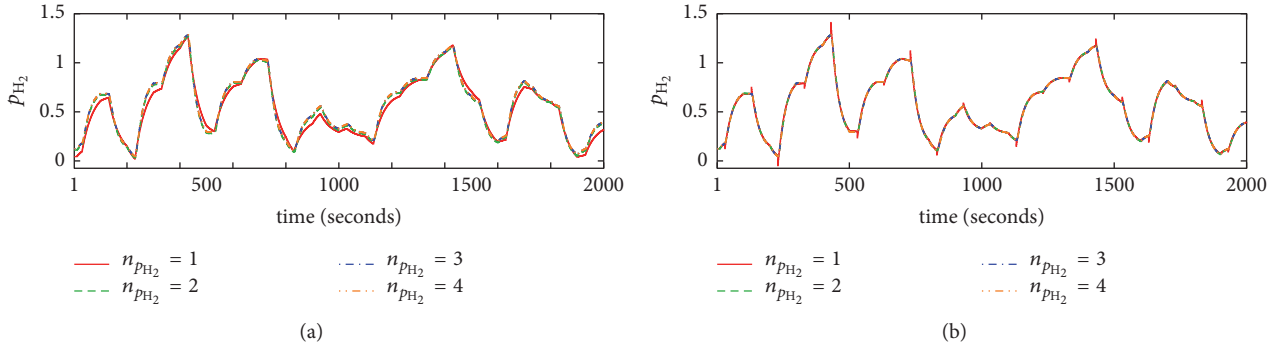
FIGURE 4: The Nyquist plots for the hydrogen pressure models of reduced order ($n_{p_{H_2}} = 1, 2, 3$) and full order ($n_{p_{H_2}} = 4$): reduction algorithm 1 (a) and reduction algorithm 2 (b).

TABLE 7: Parameters of the reduced-order oxygen pressure model of order $n_{p_{O_2}}$ calculated by the first algorithm.

$n_{p_{O_2}}$	Parameters		
1	$a_1^2 = -7.091825 \times 10^{-1}$	$b_1^{2,1} = 1.154038 \times 10^{-1}$	$b_1^{2,2} = -9.182485 \times 10^{-5}$
2	$a_1^2 = -9.956871 \times 10^{-1}$ $a_2^2 = 2.031841 \times 10^{-1}$	$b_1^{2,1} = 1.154038 \times 10^{-1}$ $b_2^{2,1} = -3.306375 \times 10^{-2}$	$b_1^{2,2} = -5.160259 \times 10^{-5}$ $b_2^{2,2} = -3.040813 \times 10^{-5}$

TABLE 8: Parameters of the reduced-order oxygen pressure model of order $n_{p_{O_2}}$ calculated by the second algorithm.

$n_{p_{O_2}}$	Parameters		
1	$a_1^2 = -7.091825 \times 10^{-1}$	$b_1^{2,1} = 1.154038 \times 10^{-1}$	$b_0^{2,2} = 5.637353 \times 10^{-5}$ $b_1^{2,2} = -1.713157 \times 10^{-4}$
2	$a_1^2 = -9.956871 \times 10^{-1}$ $a_2^2 = 2.031841 \times 10^{-1}$	$b_1^{2,1} = 1.154038 \times 10^{-1}$ $b_2^{2,1} = -3.306375 \times 10^{-2}$	$b_1^{2,2} = -5.160259 \times 10^{-5}$ $b_2^{2,2} = -3.040813 \times 10^{-5}$

FIGURE 5: The hydrogen pressure trajectories calculated by reduced order ($n_{p_{H_2}} = 1, 2, 3$) and full order ($n_{p_{H_2}} = 4$) models determined by algorithm 1 (a) and algorithm 2 (b).

is not used, but it is necessary for the first-order model to modify its steady-state gain.

Figure 6 depicts the Nyquist plots of reduced-order and full-order models of oxygen pressure. Both reduction methods give very precise models for the input $q_{O_2}^{\text{in}}$ to output channel; for all examined model orders, Nyquist plots are practically the same. Very important differences are present for the disturbance I_{in} to output channel. For the first-order model obtained by the first algorithm, the steady-state gain is different from that of the full-order model, but this discrepancy is not present for the second-order structure. For the second algorithm, modelling of the steady-state gain is excellent, but the first-order model gives wrong results as frequency increases. Similar to the first algorithm, this discrepancy is not present for the second-order structure.

For the oxygen pressure approximator, the dynamic model error, $E_{p_{O_2}}^{\text{dyn}}$, and steady-state model error, $E_{p_{O_2}}^{\text{ss}}$, are defined similar to what is done in the case of hydrogen pressure (see (40) and (41), resp.). Errors of the reduced models of hydrogen pressure are given in Table 9. As suggested by the Nyquist plots, the second algorithm gives much better results (lower dynamic and steady-state errors). For the second algorithm, the steady-state error is 0. Both reduction algorithms give perfect models of order two.

Finally, Figure 7 compares the hydrogen pressure trajectories calculated by reduced-order ($n_{p_{O_2}} = 1, 2$) and full-order ($n_{p_{O_2}} = 3$) models determined by both algorithms. As expected, taking into account the Nyquist plots and model errors, the second algorithm gives much better results for the first-order model; that is, the model obtained by the first algorithm does not have the correct steady-state properties, but this error is not significant. All models of the second order are perfect.

6.3. *Reduction of the Water Pressure Model.* The diagonal of the balanced Grammian of the state-space realisation corresponding to the discrete-time difference equation (22) describing water pressure is

$$g = [3.6014 \times 10^{-3} \quad 5.4358 \times 10^{-5}]^T. \quad (44)$$

Although there is some difference in the order of the diagonal elements of the balanced Grammian, it is not so huge as in the case of the two previously considered pressure models. For the water pressure model, the only option is to reduce the second state variable. The parameters of the reduced-order hydrogen pressure model of order $n_{p_{H_2O}} = 1$ calculated by the first algorithm are given in Table 10, whereas the

TABLE 9: The dynamic error $E_{p_{O_2}}^{\text{dyn}}$ and the steady-state error $E_{p_{O_2}}^{\text{ss}}$ of the reduced oxygen pressure models ($n_{p_{O_2}} = 1, 2$).

$n_{p_{O_2}}$	$E_{p_{O_2}}^{\text{dyn}}$		$E_{p_{O_2}}^{\text{ss}}$	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
1	5.5587×10^{-4}	9.5827×10^{-8}	1.1166×10^{-3}	0.0
2	0.0	0.0	0.0	0.0

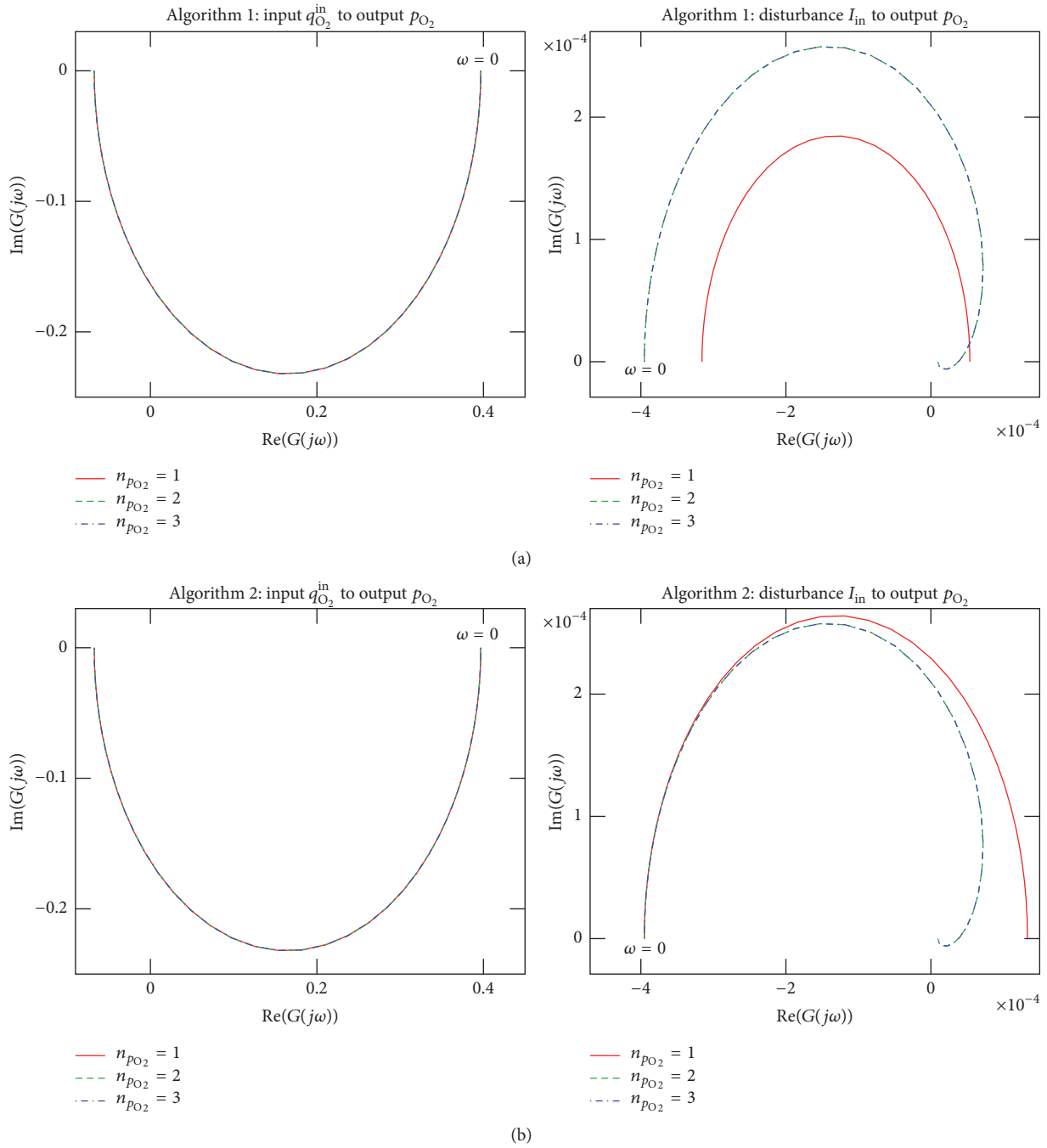
FIGURE 6: The Nyquist plots for the oxygen pressure models of reduced order ($n_{p_{O_2}} = 1, 2$) and full order ($n_{p_{O_2}} = 3$): reduction algorithm 1 (a) and reduction algorithm 2 (b).

TABLE 10: Parameters of the reduced-order water pressure model of order $n_{p_{H_2O}} = 1$ calculated by the first algorithm.

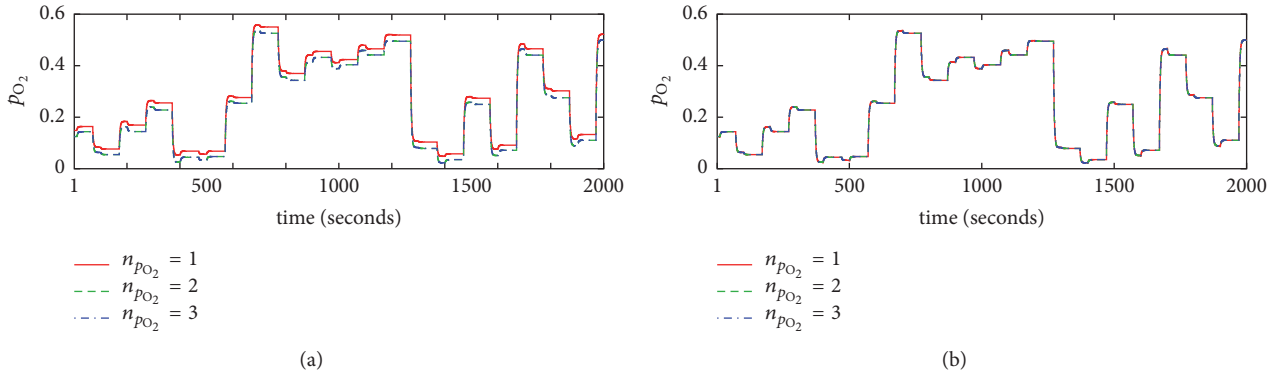
$n_{p_{H_2O}}$	Parameters	
1	$a_1^3 = -9.877933 \times 10^{-1}$	$b_1^3 = 8.736752 \times 10^{-5}$

TABLE 11: Parameters of the reduced-order water pressure model of order $n_{p_{H_2O}}$ calculated by the second algorithm.

$n_{p_{H_2O}}$	Parameters		
1	$a_1^3 = -9.877933 \times 10^{-1}$	$b_1^3 = 1.580941 \times 10^{-4}$	$b_0^3 = -6.819480 \times 10^{-5}$

TABLE 12: The dynamic error $E_{p_{H_2O}}^{\text{dyn}}$ and the steady-state error $E_{p_{H_2O}}^{\text{ss}}$ of the reduced water pressure model ($n_{p_{H_2O}} = 1$).

$n_{p_{H_2O}}$	$E_{p_{H_2O}}^{\text{dyn}}$		$E_{p_{H_2O}}^{\text{ss}}$	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
1	4.0891×10^{-4}	7.4056×10^{-8}	8.2667×10^{-4}	0.0

FIGURE 7: The oxygen pressure trajectories calculated by reduced order ($n_{p_{O_2}} = 1, 2$) and full order ($n_{p_{O_2}} = 3$) models determined by algorithm 1 (a) and algorithm 2 (b).

results of the second algorithm are given in Table 11. The general form of the reduced models calculated by the first algorithm is the same as that of the model of the full order (see (22)), whereas the reduced model determined by the second method is characterised by (38). In the latter case, the coefficient b_0^3 is necessary, which manifests a direct influence of the disturbance I_{in} on the model output p_{H_2O} .

Figure 8 depicts the Nyquist plots of reduced-order and full-order models. As it is observed in the case of hydrogen and oxygen models, the first algorithm gives slightly worse results for all frequencies. In particular, it does not guarantee that the steady-state gain of the reduced model is exactly the same as that of the original one. The second algorithm leads to a very good model.

For the water pressure approximator, the dynamic model error, $E_{p_{H_2O}}^{\text{dyn}}$, and the steady-state model error, $E_{p_{H_2O}}^{\text{ss}}$, are defined similar to what is done in the case of hydrogen pressure (see (40) and (41), resp.). Errors of the reduced model of water pressure are given in Table 12. As observed from the Nyquist plots, the model obtained by the second algorithm is characterised by much lower errors. For the second algorithm, the steady-state error is 0, but due to a

relatively low dynamic order of the original model (only two), the dynamic error is greater than 0, although it is very small.

Finally, Figure 9 compares the water pressure trajectories calculated by reduced-order ($n_{p_{H_2O}} = 1$) and full-order ($n_{p_{H_2O}} = 2$) models determined by both algorithms. As expected, taking into account the Nyquist plots and model errors, the second algorithm gives excellent results; the model generated by the first one does not have correct steady-state properties, but the error is relatively small.

6.4. Reduction of the Voltage and the Pressure Difference Models. In the previous sections, hydrogen, oxygen, and water pressure models have been reduced separately. The pressures are intermediate model variables; the actual outputs variables are the voltage V_r and the pressure difference p_{dif} . Hence, it is necessary to select order of the pressure models in such a way that not only pressures but also outputs are approximated precisely. For this purpose, the dynamic and steady-state errors of both outputs are defined similar to what is done in the case of hydrogen pressure (see (40) and (41), resp.). For the output V_r , the dynamic and steady-state errors are denoted by $E_{V_r}^{\text{dyn}}$ and $E_{V_r}^{\text{ss}}$, respectively. Next, for the

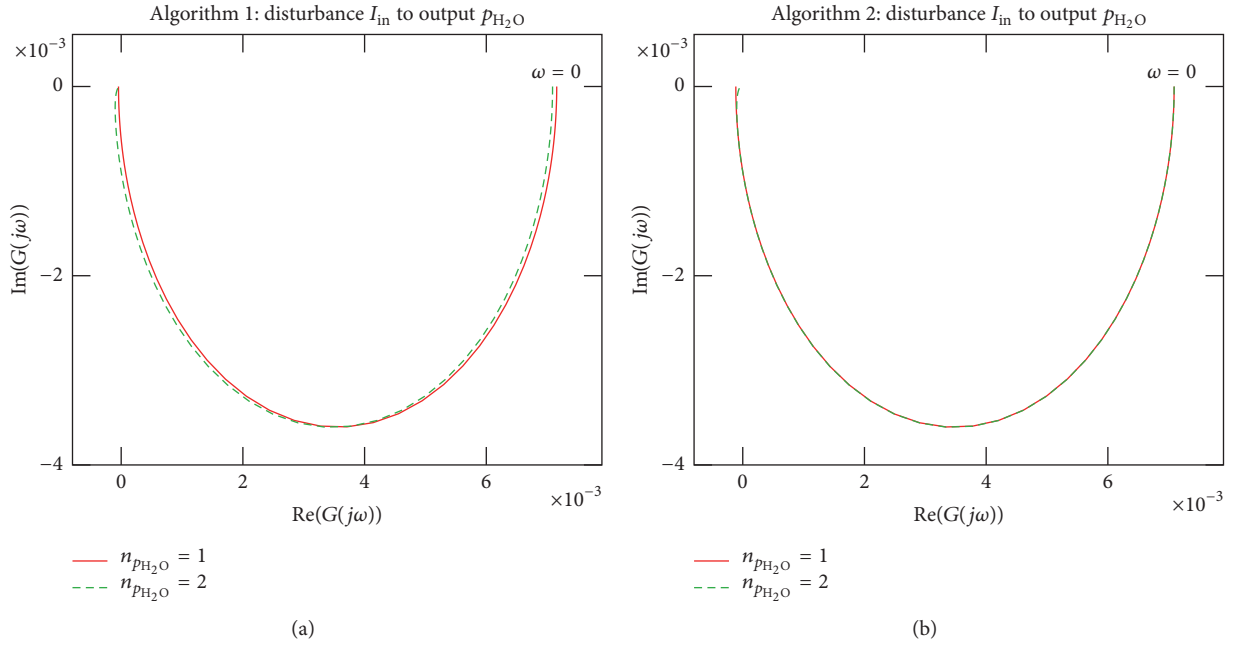


FIGURE 8: The Nyquist plots for the water pressure model of reduced order ($n_{p_{H_2O}} = 1$) and full order ($n_{p_{H_2O}} = 2$): reduction algorithm 1 (a) and reduction algorithm 2 (b).

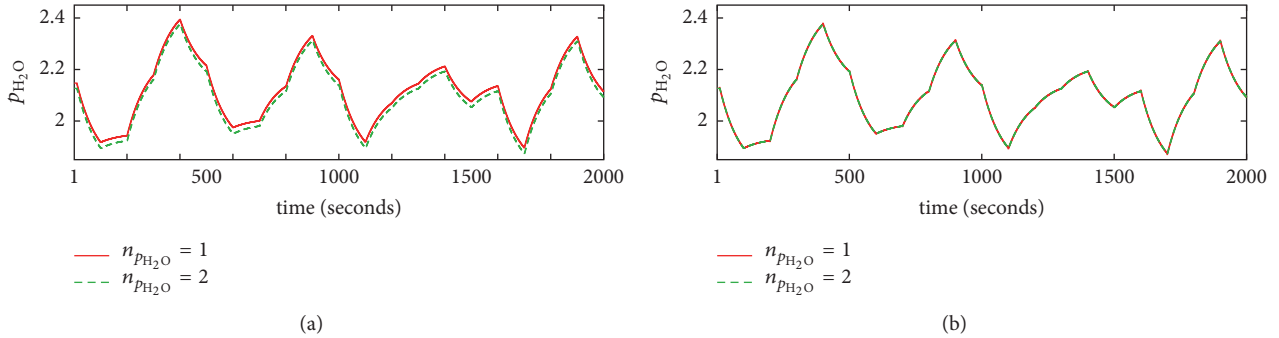


FIGURE 9: The water pressure trajectories calculated by reduced order ($n_{p_{H_2O}} = 1$) and full order ($n_{p_{H_2O}} = 2$) models determined by algorithm 1 (a) and algorithm 2 (b).

output p_{dif} , the errors are denoted by $E_{p_{\text{dif}}}^{\text{dyn}}$ and $E_{p_{\text{dif}}}^{\text{ss}}$. Table 13 gives values of the dynamic and steady-state errors of the voltage for all possible combinations of orders of hydrogen, oxygen, and water pressure models ($n_{p_{H_2}}$, $n_{p_{O_2}}$, and $n_{p_{H_2O}}$). As suggested by the partial results obtained for reduced models of the pressures, also when the errors of the output variable V_r are taken into account, the second algorithm gives much precise models. In practice, all models obtained by the second algorithm have the steady-state error equal to 0. Considering the dynamic error, three models are selected for further analysis. The perfect model (all errors are 0) that has the lowest number of parameters is characterised by the orders $n_{p_{H_2}} = 3$, $n_{p_{O_2}} = 2$, and $n_{p_{H_2O}} = 2$. Furthermore, two suboptimal models (their errors are greater than 0) are chosen: in the first case, $n_{p_{H_2}} = 2$, $n_{p_{O_2}} = 1$, and $n_{p_{H_2O}} = 1$; in the second case, $n_{p_{H_2}} = n_{p_{O_2}} = n_{p_{H_2O}} = 1$. The first

suboptimal model offers a reasonable compromise between accuracy complexities; the second one has the minimal order. Figure 10 compares dynamic trajectories of the voltage V_r calculated by three reduced-order models and the full-order one in case of two compared algorithms. In general, all models work correctly. The perfect reduced model gives no errors, whereas the suboptimal structures lead to small errors. From the comparison of the voltage trajectories, similar to the observations made so far for pressure models, it is clear that the second algorithm gives much more precise models.

Next, the second output variable is taken into account. Table 14 gives values of the dynamic and steady-state errors of the pressure difference for all possible combinations of orders of hydrogen and pressure models ($n_{p_{H_2}}$ and $n_{p_{O_2}}$); the order of the water pressure model does not influence the pressure difference. All the errors are very small, since modelling the pressure difference is much easier than modelling of the

TABLE 13: The influence of the order of hydrogen, oxygen, and water pressure models ($n_{p_{\text{H}_2}}$, $n_{p_{\text{O}_2}}$, and $n_{p_{\text{H}_2\text{O}}}$) on the dynamic error $E_{V_r}^{\text{dyn}}$ and the steady-state error $E_{V_r}^{\text{ss}}$ of the output variable V_r .

$n_{p_{\text{H}_2}}$	$n_{p_{\text{O}_2}}$	$n_{p_{\text{H}_2\text{O}}}$	$E_{V_r}^{\text{dyn}}$		$E_{V_r}^{\text{ss}}$	
			Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
1	1	1	1.7031×10^1	1.3307	1.2006×10^2	6.8080×10^{-10}
1	1	2	1.7026×10^1	1.3307	1.198117×10^2	6.8080×10^{-10}
1	2	1	1.6021×10^1	1.3247	1.084092×10^2	0.0
1	2	2	1.5278×10^2	1.3248	1.073773×10^2	0.0
1	3	1	1.6021×10^1	1.3247	1.084092×10^2	0.0
1	3	2	1.5278×10^2	1.3248	1.073773×10^2	0.0
2	1	1	4.4605	4.5855×10^{-3}	6.2967×10^1	6.8380×10^{-10}
2	1	2	4.8173	4.3832×10^{-3}	6.3211×10^1	6.8380×10^{-10}
2	2	1	1.9113	5.7115×10^{-4}	4.5223×10^1	0.0
2	2	2	1.5297	5.7115×10^{-4}	4.4634×10^1	0.0
2	3	1	1.5297	6.4325×10^{-4}	4.5223×10^1	0.0
2	3	2	1.5297	5.7115×10^{-4}	4.4634×10^1	0.0
3	1	1	5.6368	2.8609×10^{-3}	2.3427×10^1	6.8080×10^{-10}
3	1	2	6.3333	2.7232×10^{-3}	2.4225×10^1	6.8080×10^{-10}
3	2	1	4.1917×10^{-2}	7.5057×10^{-6}	4.0895×10^{-2}	0.0
3	2	2	0.0	0.0	0.0	0.0
3	3	1	4.1917×10^{-2}	7.5057×10^{-6}	4.0895×10^{-2}	0.0
3	3	2	0.0	0.0	0.0	0.0
4	1	1	5.6368	2.8609×10^{-3}	2.3427×10^1	6.8080×10^{-10}
4	1	2	6.3333	2.7232×10^{-3}	2.4225×10^1	6.8080×10^{-10}
4	2	1	4.1917×10^{-2}	7.5057×10^{-6}	4.0895×10^{-2}	0.0
4	2	2	0.0	0.0	0.0	0.0
4	3	1	4.1917×10^{-2}	7.5057×10^{-6}	4.0895×10^{-2}	0.0

voltage. It is because the Nernst equation (25) is nonlinear, whereas the pressure difference depends linearly only on hydrogen and oxygen pressures (see (28)). The second observation is that the second algorithm in comparison with the first one gives smaller errors. In particular, all steady-state errors of the model generated by the second method are equal to 0. The same three reduced-order models are selected taking into account the errors of the pressure difference: the perfect reduced structure (with all errors equal to 0) and two suboptimal ones. Figure 11 compares dynamic trajectories of the pressure difference p_{dif} calculated by three reduced-order models and the full-order one in case of two compared algorithms. As in the case of the voltage model, also all pressure difference models work correctly. The perfect reduced model gives no errors, whereas the suboptimal reduced structures lead to small errors. As always, in this study, the second algorithm gives more precise models.

To show good steady-state properties of the simplest reduced model ($n_{p_{\text{H}_2}} = n_{p_{\text{O}_2}} = n_{p_{\text{H}_2\text{O}}} = 1$), its steady-state characteristics obtained by two considered algorithms are compared with the ideal characteristics of the full-order model in Figure 12. When the voltage is considered, there are very small differences between the ideal characteristic and that of the reduced model generated by algorithm 1. The

characteristic of the model produced by the second algorithm is very precise, as demonstrated numerically in Table 13. When the pressure difference is considered, the differences between the model obtained by the first algorithm and the full-order one are practically not visible; the second algorithm gives perfect characteristic, as demonstrated numerically in Table 14.

Finally, Table 15 gives a comparison of dynamic and steady-state model errors for the ideal full-order model and the chosen three reduced ones. The errors for two output variables are given and only the second algorithm is considered, since it always gives better results than the first one. The number of pressure model parameters is given. Although the perfect reduced-order model gives all errors equal to 0, it only reduces the overall number of parameters from 25 to 19. That is why in practical applications it may be more attractive to use suboptimal structures. The first suboptimal reduced model makes it possible to reduce the number of parameters to 14 but at the same time its dynamic and steady-state behaviours are practically the same as that of the full-order model. It is interesting to point out that even the simplest first-order reduced model, which makes it possible to reduce the number of parameters to 11, gives quite good results. Of course, selection of the models depends on specifications of the considered application.

TABLE 14: The influence of the order of hydrogen and oxygen pressure models ($n_{p_{H_2}}$ and $n_{p_{O_2}}$) on the dynamic error $E_{p_{dif}}^{dyn}$ and the steady-state error $E_{p_{dif}}^{ss}$ of the output variable p_{dif} .

$n_{p_{H_2}}$	$n_{p_{O_2}}$	$E_{p_{dif}}^{dyn}$		$E_{p_{dif}}^{ss}$	
		Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
1	1	4.4275×10^{-3}	1.1629×10^{-4}	2.3936×10^{-2}	0.0
1	2	2.5494×10^{-3}	1.1652×10^{-4}	1.7068×10^{-2}	0.0
1	3	2.5494×10^{-3}	1.1652×10^{-4}	1.7068×10^{-2}	0.0
2	1	1.4349×10^{-3}	2.1408×10^{-8}	2.8942×10^{-3}	0.0
2	2	2.0677×10^{-4}	3.5651×10^{-8}	4.1543×10^{-4}	0.0
2	3	2.0677×10^{-4}	3.5651×10^{-8}	4.1543×10^{-4}	0.0
3	1	5.5417×10^{-4}	9.5827×10^{-8}	1.1166×10^{-3}	0.0
3	2	0.0	0.0	0.0	0.0
3	3	0.0	0.0	0.0	0.0
4	1	5.5417×10^{-4}	9.5827×10^{-8}	1.1166×10^{-3}	0.0
4	2	0.0	0.0	0.0	0.0

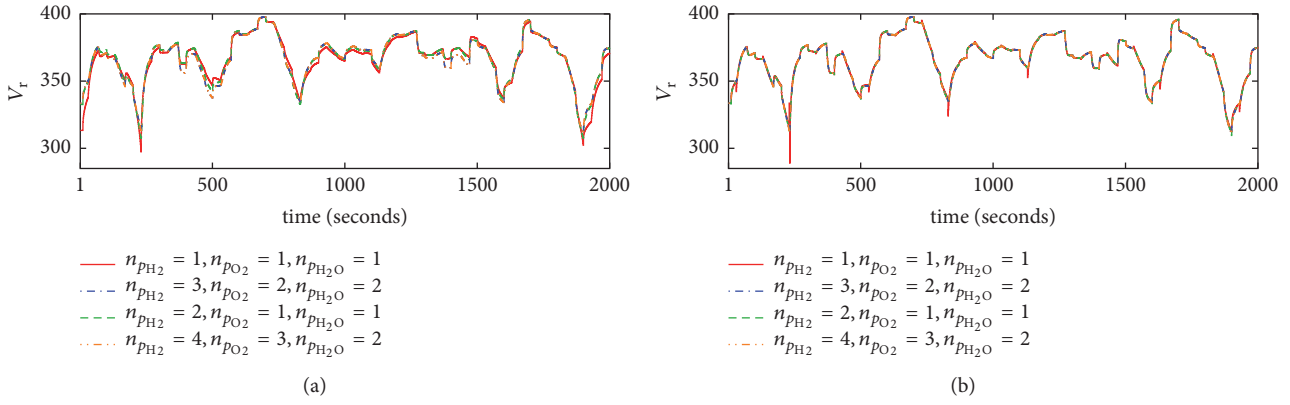


FIGURE 10: The voltage V_r trajectories calculated by three reduced-order and full-order ($n_{p_{H_2}} = 4$, $n_{p_{O_2}} = 3$, and $n_{p_{H_2O}} = 2$) models determined by algorithm 1 (a) and algorithm 2 (b).

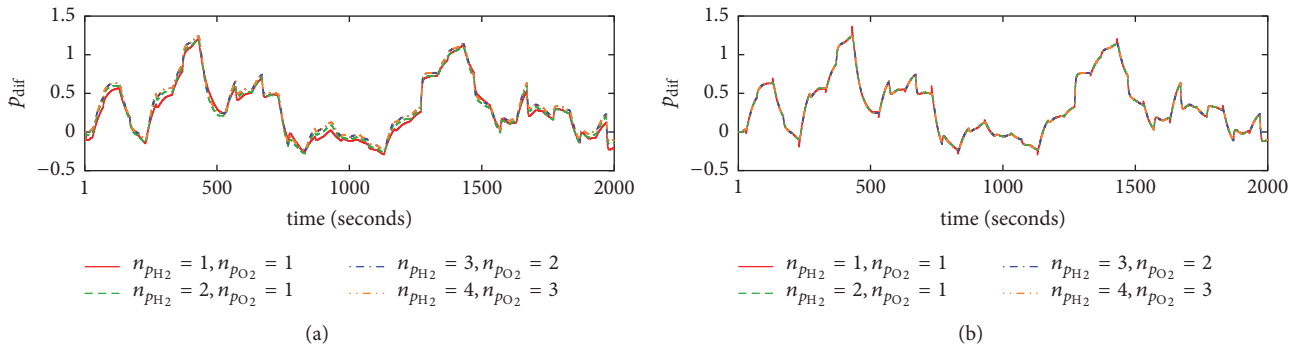


FIGURE 11: The pressure difference p_{dif} trajectories calculated by three reduced-order and full-order ($n_{p_{H_2}} = 4$ and $n_{p_{O_2}} = 3$) models determined by algorithm 1 (a) and algorithm 2 (b).

7. Conclusions

This work discusses reduction of the dynamic model of the solid oxide fuel cell. Firstly, the discrete-time representation of the continuous-time model is found. It is shown that the discrete-time model gives process trajectories practically the

same as the original continuous-time one. Next, the discrete-time submodels of hydrogen, oxygen, and water pressures (intermediate variables) are reduced by means of state truncation and residualisation methods. The reduced pressure models are assessed taking into account both dynamic and steady-state errors as well as Nyquist plots. Next, the influence

TABLE 15: Comparison of the ideal full-order model ($n_{p_{H_2}} = 4$, $n_{p_{O_2}} = 3$, and $n_{p_{H_2O}} = 2$) and the chosen reduced ones calculated by the second algorithms in terms of the number of pressure parameters as well as the dynamic and steady-state errors for two process output variables (algorithm 2).

$n_{p_{H_2}}$	$n_{p_{O_2}}$	$n_{p_{H_2O}}$	Parameters	$E_{V_r}^{dyn}$	$E_{V_r}^{ss}$	$E_{p_{dif}}^{dyn}$	$E_{p_{dif}}^{ss}$
4	3	2	25	0.0	0.0	0.0	0.0
3	2	2	19	0.0	0.0	0.0	0.0
2	1	1	14	4.3832×10^{-3}	6.8380×10^{-10}	2.1408×10^{-8}	0.0
1	1	1	11	1.3307	6.8080×10^{-10}	1.1629×10^{-4}	0.0

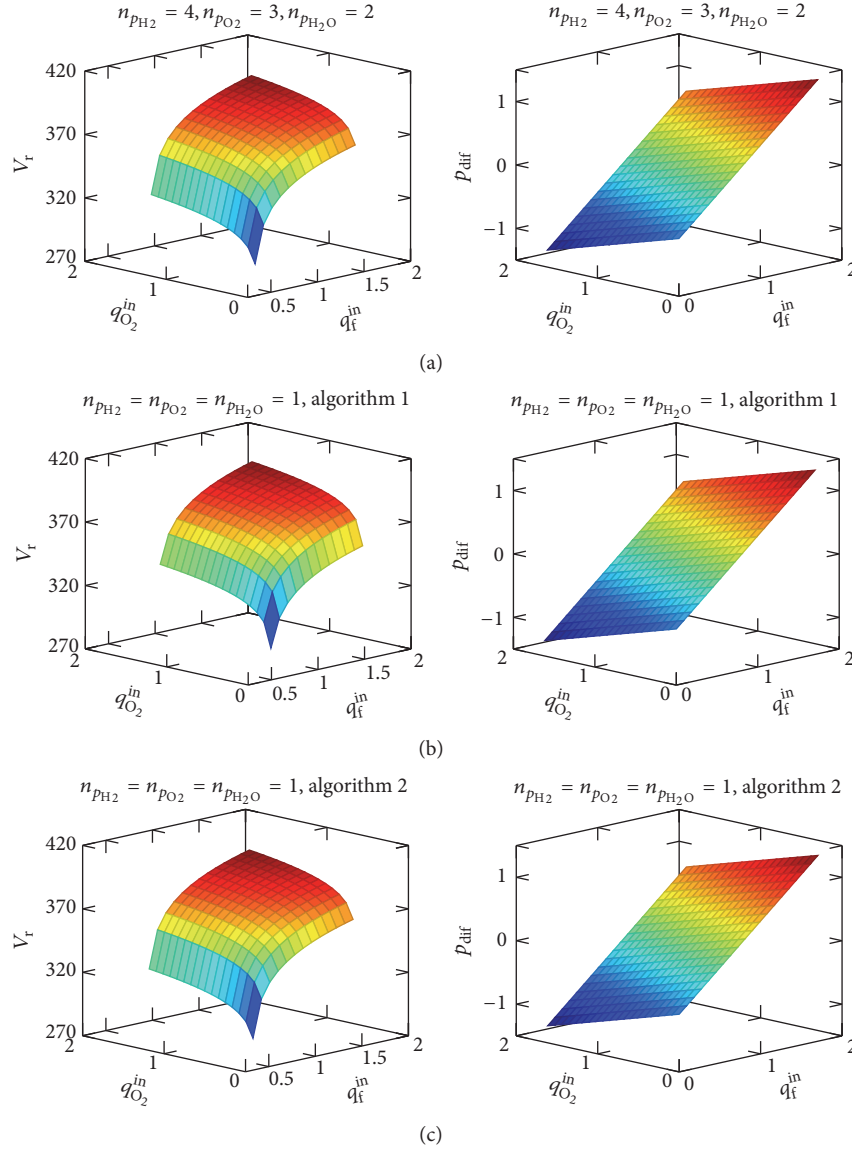


FIGURE 12: The steady-state voltage V_r and partial difference p_{dif} steady-state characteristics calculated by the full-order ($n_{p_{H_2}} = 4$, $n_{p_{O_2}} = 3$, and $n_{p_{H_2O}} = 2$) model (a) and the simplest reduced model ($n_{p_{H_2}} = n_{p_{O_2}} = n_{p_{H_2O}} = 1$) (b, c).

of the order of the pressure models on the dynamic and steady-state errors of the process outputs (the voltage and the pressure difference) is studied. Three reduced models are chosen: with 19, 14, and 11 pressure parameters (the original one has 25 of them). In general, in all cases, the second reduction method is more precise.

It is important to summarise the advantages of the described reduced models:

- (1) The reduced models calculate not only the output variables (as the black-box models) but also the intermediate variables (the pressures).

- (2) The reduced models are selected taking into account dynamic and static errors as well as Nyquist plots.
- (3) The first reduced model (with 19 pressure parameters) does not have any steady-state and dynamic errors; it is able to calculate pressures and process outputs exactly in the same way it is done by the full-order model.
- (4) Although the second and the third reduced models (with 14 and 11 pressure parameters, resp.) have some errors, they are quite small and the reduced models give dynamic trajectories and steady-state characteristics that are very similar to those obtained by the full-order structure.

The reduced-order model of the solid oxide fuel cell may be further used in all model-based methods. Firstly, it may be used for design and implementation of MPC algorithms [31], in particular computationally efficient MPC schemes [32]. In addition to classical MPC algorithms, robust versions with guaranteed stability may be considered [33]. Secondly, it may be also used in online process optimisation [31] cooperating with MPC. Thirdly, the model may be used in fault diagnosis of the process and fault-tolerant control [34]. The presented model reduction approach may be also used for different variants of dynamic models of fuel cells [9, 22–24, 35, 36].

Data Availability

The dynamic and steady-state data sets used in this work may be obtained directly from the author on request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research has not received any specific grant from funding agencies in the public, commercial, or not-for-profit sectors, but it has been performed as part of the employment of the author at the Institute of Control and Computation Engineering, Warsaw University of Technology.

References

- [1] M. Deissenroth, M. Klein, K. Nienhaus, and M. Reeg, "Assessing the plurality of actors and policy interactions: agent-based modelling of renewable energy market integration," *Complexity*, vol. 2017, Article ID 7494313, pp. 1–24, 2017.
- [2] P. Singh and B. Khan, "Smart microgrid energy management using a novel artificial shark optimization," *Complexity*, vol. 2017, Article ID 2158926, pp. 1–22, 2017.
- [3] L. Wang, M. Cai, H. Zhang, F. Alsaadi, and L. Chen, "Active fault-tolerant control for wind turbine with simultaneous actuator and sensor faults," *Complexity*, vol. 2017, Article ID 6164841, pp. 1–11, 2017.
- [4] A. Daniilidis, T. Scholten, J. Hooghiem, C. De Persis, and R. Herber, "Geochemical implications of production and storage control by coupling a direct-use geothermal system with heat networks," *Applied Energy*, vol. 204, pp. 254–270, 2017.
- [5] A. V. Arasu and T. Sornakumar, "Design, development and performance studies of embedded electronic controlled one axis solar," *Asian Journal of Control*, vol. 9, no. 2, pp. 163–169, 2007.
- [6] S. Zou, O. Abdelkhalik, R. Robinett et al., "Model Predictive Control of parametric excited pitch-surge modes in wave energy converters," *International Journal of Marine Energy*, vol. 19, pp. 32–46, 2017.
- [7] G. Di Blasio, M. Viscardi, and C. Beatrice, "DoE method for operating parameter optimization of a dual-fuel bioethanol/diesel light duty engine," *Journal of Fuels*, vol. 2015, Article ID 674705, 14 pages, 2015.
- [8] J. Larminie and A. Dicks, *Fuel Cell Systems Explained*, John Wiley and Sons, Chichester, UK, 2003.
- [9] J. Milewski, K. Świrski, M. Santarelli, and P. Leone, *Advanced Methods of Solid Oxide Fuel Cell Modelling*, Springer, London, UK, 2011.
- [10] V. Knyazkin, L. Söder, and C. Canizares, "Control challenges of fuel cell-driven distributed generation," in *Proceedings of the IEEE Bologna PowerTech Conference*, pp. 564–569, Bologna, Italy, June 2003.
- [11] K. C. Bhuyan, S. K. Sao, and K. Mahapatra, "An FPGA based controller for a SOFC DC-DC power system," in *Advances in Power Electronics*, vol. 2013, pp. 1–12, 2013.
- [12] X. Wang, B. Huang, and T. Chen, "Data-driven predictive control for solid oxide fuel cells," *Journal of Process Control*, vol. 17, no. 2, pp. 103–114, 2007.
- [13] L. T. Jacobsen, B. J. Spivey, and J. D. Hedengren, "Model predictive control with a rigorous model of a solid oxide fuel cell," in *Proceedings of the 1st American Control Conference (ACC '13)*, pp. 3741–3746, USA, June 2013.
- [14] N. Ji, D. Xu, and F. Liu, "A novel adaptive neural network constrained control for solid oxide fuel cells via dynamic anti-windup," *Neurocomputing*, vol. 214, pp. 134–142, 2016.
- [15] X. Wu and D. Gao, "Fault tolerance control of SOFC systems based on nonlinear model predictive control," *International Journal of Hydrogen Energy*, vol. 42, no. 4, pp. 2288–2308, 2017.
- [16] X. W. Zhang, S. H. Chan, H. K. Ho, J. Li, G. Li, and Z. Feng, "Nonlinear model predictive control based on the moving horizon state estimation for the solid oxide fuel cell," *International Journal of Hydrogen Energy*, vol. 33, no. 9, pp. 2355–2366, 2008.
- [17] L. Khani, A. S. Mehr, M. Yari, and S. M. S. Mahmoudi, "Multi-objective optimization of an indirectly integrated solid oxide fuel cell-gas turbine cogeneration system," *International Journal of Hydrogen Energy*, vol. 41, no. 46, pp. 21470–21488, 2016.
- [18] H. Cheng, X. Li, J. Jiang, Z. Deng, J. Yang, and J. Li, "A nonlinear sliding mode observer for the estimation of temperature distribution in a planar solid oxide fuel cell," *International Journal of Hydrogen Energy*, vol. 40, no. 1, pp. 593–606, 2015.
- [19] J. Milewski and K. Świrski, "Modelling the SOFC behaviours by artificial neural network," *International Journal of Hydrogen Energy*, vol. 34, no. 13, pp. 5546–5553, 2009.
- [20] J. Jiang, X. Li, Z. Deng, J. Yang, Y. Zhang, and J. Li, "Thermal management of an independent steam reformer for a solid oxide fuel cell with constrained generalized predictive control," *International Journal of Hydrogen Energy*, vol. 37, no. 17, pp. 12317–12331, 2012.
- [21] T. Zhang and G. Feng, "Rapid load following of an SOFC power system via stable fuzzy predictive tracking controller," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 2, pp. 357–371, 2009.

- [22] R. Bove and S. Ubertini, Eds., *Modelling Solid Oxide Fuel Cell*, Springer, London, UK, 2008.
- [23] S. A. Hajimolana, M. A. Hussain, W. M. A. W. Daud, M. Soroush, and A. Shamiri, "Mathematical modeling of solid oxide fuel cells: a review," *Renewable & Sustainable Energy Reviews*, vol. 15, no. 4, pp. 1893–1917, 2011.
- [24] B. Huang, Y. Qi, and M. Murshed, "Solid oxide fuel cell: Perspective of dynamic modeling and control," *Journal of Process Control*, vol. 21, pp. 1426–1437, 2007.
- [25] J. Padullés, G. W. Ault, and J. R. McDonald, "Integrated SOFC plant dynamic model for power systems simulation," *Journal of Power Sources*, vol. 86, no. 1, pp. 495–500, 2000.
- [26] Y. Zhu and K. Tomsovic, "Development of models for analyzing the load-following performance of microturbines and fuel cells," *Electric Power Systems Research*, vol. 62, no. 1, pp. 1–11, 2002.
- [27] C.-T. Chen, *Linear System Theory and Design*, Oxford University Press, New York, NY, USA, 1999.
- [28] A. Laub, M. Heath, C. Paige, and R. Ward, "Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms," *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 115–122, 1987.
- [29] B. C. Moore, "Principal component analysis in linear systems: controllability, observability, and model reduction," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.
- [30] B. Roffel and B. Betlem, *Process Dynamics and Control: Modeling for Control and Predictions*, Wiley, Chichester, UK, 2006.
- [31] P. Tatjewski, *Advanced Control of Industrial Processes, Structures and Algorithms*, Springer, London, UK, 2007.
- [32] M. Ławryńczuk, "Computationally efficient model predictive control algorithms: a neural network approach," in *Studies in Systems, Decision and Control*, vol. 3, Springer, Cham, Switzerland, 2014.
- [33] F. Zhou, H. Peng, X. Zeng, X. Tian, and X. Peng, "RBF-ARX model-based robust MPC for nonlinear systems with unknown and bounded disturbance," *Journal of The Franklin Institute*, vol. 354, no. 18, pp. 8072–8093, 2017.
- [34] M. Witczak, *Lecture Notes in Electrical Engineering*, vol. 266, Springer, Cham, Switzerland, 2014.
- [35] V. De Marco, G. Florio, and P. Fragiaco, "Optimal Operation Conditions for a Methane Fuelled SOFC and Microturbine Hybrid System," *Journal of Renewable Energy*, vol. 2015, Article ID 508138, pp. 1–13, 2015.
- [36] J. D. J. Rubio and A. G. Bravo, "Optimal control of a PEM fuel cell for the inputs minimization," *Mathematical Problems in Engineering*, vol. 2014, Article ID 698250, pp. 1–7, 2014.

Research Article

Estimation and Synthesis of Reachable Set for Singular Markovian Jump Systems

Yucai Ding  and Hui Liu

School of Science, Southwest University of Science and Technology, Mianyang 621010, China

Correspondence should be addressed to Yucai Ding; dycyer@163.com

Received 31 January 2018; Accepted 28 March 2018; Published 26 June 2018

Academic Editor: Sing Kiong Nguang

Copyright © 2018 Yucai Ding and Hui Liu. 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 problems of reachable set estimation and state-feedback controller design are investigated for singular Markovian jump systems with bounded input disturbances. Based on the Lyapunov approach, several new sufficient conditions on state reachable set and output reachable set are derived to ensure the existence of ellipsoids that bound the system states and output, respectively. Moreover, a state-feedback controller is also designed based on the estimated reachable set. The derived sufficient conditions are expressed in terms of linear matrix inequalities. The effectiveness of the proposed results is illustrated by numerical examples.

1. Introduction

The research on singular systems has attracted significant attention in the past years due to the fact that singular systems can better describe a larger class of physical systems such as robotic systems, electric circuits, and mechanical systems. When singular systems experience abrupt changes in their structures, it is natural to model them as singular Markovian jump systems [1, 2]. The analysis and synthesis of such class of systems have gained considerable attention because of their importance in applications (see, e.g., the literature [3–10] and the references therein).

Reachable set is one of the important techniques for parameter estimation or state estimation problems [11]. Reachable set for a dynamic system is the set containing all the system states starting from the origin under bounded input disturbances. However, the exact shape of reachable sets of a dynamic system is very complex and hard to obtain; for this reason a number of researchers began to turn their attention to the reachable set estimation problem. The common strategies for reachable set estimation are ellipsoidal method [12] and polyhedron method [13]. The main idea of these methods is to detect simple convex shapes like ellipsoid or polyhedron, which contains all the system states. Compared with polyhedron method, the primary advantage of ellipsoidal method is that the ellipsoid structure is simple

and directly related to quadratic Lyapunov functions. As a result, linear matrix inequalities (LMIs) techniques can be used to determine bounding ellipsoids. In the framework of bounding ellipsoid, the reachable set estimation problem for linear time delay systems has received significant research attention in recent years. In [14] sufficient conditions for the existence of bounding ellipsoids containing the reachable set of continuous-time linear systems with time-varying delays were derived by using the Lyapunov-Razumikhin function. In [15], by using Lyapunov-Krasovskii functional method, the author derived some less conservative conditions than those in [14]. In [16] the reachable set of delayed systems with polytopic uncertainties was investigated by using the maximal Lyapunov-Krasovskii functional approach, and some new conditions bounding the set of reachable states are derived. Interesting results on reachable set of delayed systems with polytopic uncertainties can also be found in [17–20]. In addition, some other strategies without using Lyapunov-Krasovskii functional have been provided to estimate the reachable set of continuous-time linear time-varying systems [21] and nonlinear time delay systems [22, 23]. The authors in [24] extended the ideas of reachable set estimation of continuous-time systems to discrete-time systems, wherein a fundamental result (Lemma 2.1 [24]) for the reachable set estimation of discrete-time systems was proposed. The authors in [25] improved the fundamental result obtained

in [24] and provided a basic tool (Lemma 4 [25]) for the reachable set estimation of discrete-time systems. On the basis of the general ideas proposed in [25], the reachable set estimation problem was also extended to some classes of complicated systems, such as singular systems [26], Markovian jump systems [27], switched linear systems [28], and T-S fuzzy systems [29, 30]. For the reachable set estimation of discrete-time systems, the other important contributions can be found in [31, 32]. On the other hand, the problem of controller design for specifications involved with the reachable set of a control system is also a very important issue [33]. The controller design problems concerning reachable set were studied in [34] and [35] by using ellipsoidal method and polyhedron method, respectively. Two issues were raised in [34]: the first one is to design a controller such that the reachable set of the closed-loop system is contained in an ellipsoid, and the admissible ellipsoid should be as small as possible; the second one is to design a controller such that the reachable set of the closed-loop system is contained in a given ellipsoid. By constructing suitable Lyapunov-Krasovskii functional, LMI-based sufficient conditions for the existence of controller guaranteeing the ellipsoid bounds as small as possible have been derived for continuous-time delay systems [34] and discrete-time periodic systems [36]. It is obvious that the LMI-based controller design is quite simple and numerically tractable. However, it should be pointed out that the reachable set estimation and synthesis problems of singular Markovian jump systems are much more difficult and challenging than that for nonsingular Markovian jump systems since the ellipsoid containing the reachable set is not directly related to quadratic Lyapunov functions. To the best of the authors' knowledge, no related results have been established for reachable set estimation and synthesis of singular Markovian jump systems, which has motivated this paper.

In this paper, we consider the problems of reachable set estimation and synthesis of singular Markovian jump systems. By using the Lyapunov approach, the estimation conditions on state reachable set and output reachable set are derived, respectively. Moreover, the desired state-feedback controller is designed based on the estimated reachable set.

Notation. Throughout this paper, \mathbb{R}^n denotes the n -dimensional Euclidean space; A^T represents the transpose of A ; $\text{Sym}(M)$ stands for $M + M^T$; $X > 0$ (< 0) means X is a symmetric positive (negative) definite matrix; $\mathbb{E}\{\cdot\}$ refers to the expectation; $X_{l \times m}$ denotes the matrix composed of elements of first l rows and m columns of matrix X ; $\|\cdot\|$ refers to the Euclidean vector norm; the symbol “ $*$ ” in LMIs denotes the symmetric term of the matrix; I is the unit matrix with appropriate dimensions.

2. Problem Formulation

Consider the following singular Markovian jump system:

$$\begin{aligned} E\dot{x}(t) &= A(r_t)x(t) + B(r_t)u(t) + D(r_t)\omega(t) \\ y(t) &= C(r_t)x(t), \end{aligned} \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^p$ is the measured output, and $\omega(t) \in \mathbb{R}^q$ is the exogenous disturbance which satisfies

$$\omega^T(t)\omega(t) \leq \bar{\omega}^2. \quad (2)$$

$E, A(r_t), B(r_t), C(r_t)$, and $D(r_t)$ are real constant matrices with appropriate dimensions and $\text{rank}(E) = l < n$. $\{r_t, t \geq 0\}$ is a continuous-time Markovian process with transition rate matrix $\Pi = [\pi_{ij}]$ ($i \in \mathcal{S} = \{1, 2, \dots, N\}$) and the evolution of Markovian process is governed by the following transition rate:

$$\Pr\{r_{t+\Delta} = j \mid r_t = i\} = \begin{cases} \pi_{ij}\Delta + o(\Delta), & j \neq i \\ 1 + \pi_{ii}\Delta + o(\Delta), & j = i, \end{cases} \quad (3)$$

where $\Delta > 0$ and $\lim_{\Delta \rightarrow 0} (o(\Delta)/\Delta) = 0$; $\pi_{ij} \geq 0$ for $i \neq j$ is the transition rate from mode i to mode j and $\pi_{ii} = -\sum_{j=1, j \neq i}^N \pi_{ij}$.

For notational simplicity, in the sequel, for each possible $r_t = i, i \in \mathcal{S}$, matrices $A(r_t), B(r_t), C(r_t)$, and $D(r_t)$ will be denoted by A_i, B_i, C_i , and D_i , respectively. When $u(t) = 0$, the system $E\dot{x}(t) = A_i x(t) + D_i \omega(t)$ is referred to as a free system.

In this paper we are interested in determining ellipsoids that contain, respectively, the state reachable set and output reachable set. In the reachable set analysis, it is required that systems should be asymptotically stable. When this requirement is not met, we will further design a state-feedback controller such that the reachable set of the closed-loop system is contained in the smallest ellipsoid.

The state reachable set of the free system in (1) is defined by

$$\begin{aligned} \mathcal{R}_x &= \{x(t) \in \mathbb{R}^n \mid x(t), t \\ &\geq 0, \text{ is a solution of (1) for } x(0) = 0\}. \end{aligned} \quad (4)$$

An ellipsoid $\mathcal{E}(X)$ bounding the reachable set can be always represented as follows:

$$\mathcal{E}(X) = \{x \in \mathbb{R}^n \mid x^T X x \leq 1, X > 0\}. \quad (5)$$

Particularly, when $X = \kappa I$ for $\forall \kappa > 0$, the ellipsoid $\mathcal{E}(X)$ will become a ball which is denoted by $\mathcal{B}(X)$.

Since $\text{rank}(E) = l < n$, there exist two nonsingular matrices M and N such that

$$\begin{aligned} MEN &= \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \\ MA_i N &= \begin{bmatrix} A_{11i} & A_{12i} \\ A_{21i} & A_{22i} \end{bmatrix}, \\ MD_i &= \begin{bmatrix} D_{1i} \\ D_{2i} \end{bmatrix}. \end{aligned} \quad (6)$$

Let $N^{-1}x(t) = \tilde{x}(t) = [\tilde{x}_1^T(t) \ \tilde{x}_2^T(t)]^T$. Then the free system can be rewritten as the following differential-algebraic form:

$$\dot{\tilde{x}}_1(t) = A_{11i}\tilde{x}_1(t) + A_{12i}\tilde{x}_2(t) + D_{1i}\omega(t) \quad (7)$$

$$0 = A_{21i}\tilde{x}_1(t) + A_{22i}\tilde{x}_2(t) + D_{2i}\omega(t). \quad (8)$$

The following definition and lemma are also useful in deriving the main results.

Definition 1 (see [2]). (I) The free system is said to be regular if $\det(sE - A_i)$ is not identically zero for each $i \in \mathcal{S}$.

(II) The free system is said to be impulse-free if $\deg(\det(sE - A_i)) = \text{rank}(E)$ for each $i \in \mathcal{S}$.

Lemma 2 (see [2]). *For any matrices U and $V \in \mathbb{R}^{n \times n}$ with $V > 0$, one has $UV^{-1}U^T \geq U + U^T - V$.*

Lemma 3 (see [12]). *Let $V(x(t))$ be a Lyapunov function and $V(x(0)) = 0$. If*

$$\dot{V}(x(t)) + \alpha V(x(t)) - \frac{\alpha}{\bar{\omega}^2} \omega^T(t) \omega(t) \leq 0, \quad \alpha > 0, \quad (9)$$

then $V(x(t)) \leq 1, \forall t \geq 0$.

3. Main Results

3.1. State Reachable Set Estimation. In this subsection, we will focus our attention on determining a ball which contains the state reachable set of the free system.

Theorem 4. *If there exist nonsingular matrices $X_i \in \mathbb{R}^{n \times n}$ and a scalar $\alpha > 0$ such that the following LMIs hold for each $i \in \mathcal{S}$,*

$$E^T X_i = X_i^T E \geq 0, \quad (10)$$

$$\mathcal{H} = \begin{bmatrix} \text{Sym}(A_i^T X_i) + \sum_{j=1}^N \pi_{ij} E^T X_j + \alpha E^T X_i & X_i^T D_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I \end{bmatrix} < 0, \quad (11)$$

then the state reachable set of free system starting from the origin is mean-square bounded within the following set:

$$\bigcap_{i=1}^N \mathcal{E}(\widehat{X}_i) = \bigcap_{i=1}^N \{x \in \mathbb{R}^n \mid x^T \widehat{X}_i x \leq 1\}, \quad (12)$$

where

$$\begin{aligned} \widehat{X}_i &= \frac{1}{r_i^2} N^{-T} N^{-1} \\ r_i &= \frac{1}{\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\widetilde{X}_i)}} \\ &+ \|A_{22i}^{-1}\| \left(\frac{\|A_{21i}\|}{\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\widetilde{X}_i)}} + \|D_{2i}\| \bar{\omega} \right) \\ \widetilde{X}_i &= (M^{-T} X_i N)_{l \times l}. \end{aligned} \quad (13)$$

Proof. We first prove the regularity and nonimpulsiveness of the free system. Let $M^{-T} X_i N = \begin{bmatrix} X_{11i} & X_{12i} \\ X_{21i} & X_{22i} \end{bmatrix}$. Then, by (10), we obtain that $X_{12i} = 0$. From (11), it is easy to show that

$$\text{Sym}(A_i^T X_i) + \sum_{j=1}^N \pi_{ij} E^T X_j + \alpha E^T X_i < 0. \quad (14)$$

Pre- and postmultiplying (14) by N^T and N , respectively, we get

$$\begin{bmatrix} * & * \\ * & A_{22i}^T X_{22i} + X_{22i}^T A_{22i} \end{bmatrix} < 0, \quad (15)$$

where $*$ will be irrelevant to the results of the following discussion; thus the real expressions of these two variables are omitted. It follows from (15) that

$$A_{22i}^T X_{22i} + X_{22i}^T A_{22i} < 0 \quad (16)$$

which implies that A_{22i} is nonsingular for each $i \in \mathcal{S}$. Therefore, by Definition 1, we have that the free system is regular and nonimpulsive.

Next, we will show the state reachable set of free system is mean-square bounded within the set $\mathcal{E}(\widehat{X})$. Consider the following Lyapunov function:

$$V(x(t), r_t) = x^T(t) E^T X_i x(t). \quad (17)$$

Let \mathcal{L} be the weak infinitesimal generator of the random process $\{x(t), r_t\}$. Calculating the difference of $V(x(t), r_t)$ along the trajectories of the free system, we get

$$\begin{aligned} \mathcal{L}V(x(t), r_t) &= x^T(t) E^T X_i x(t) + x^T(t) X_i^T E \dot{x}(t) \\ &+ x^T(t) \sum_{j=1}^N \pi_{ij} E^T X_j x(t). \end{aligned} \quad (18)$$

Defining the augmented system variable as $\xi(t) = [x^T(t) \omega^T(t)]^T$ and using conditions (10) and (11), we have

$$\begin{aligned} \mathcal{L}V(x(t), r_t) + \alpha V(x(t), r_t) - \frac{\alpha}{\bar{\omega}^2} \omega^T(t) \omega(t) \\ = \xi^T(t) \mathcal{H} \xi(t) < 0. \end{aligned} \quad (19)$$

Then we can deduce from Lemma 3 that $\mathbb{E}\{x^T(t) E^T X_i x(t)\} \leq 1$, which infers that

$$\mathbb{E}\{\bar{x}^T(t) N^T E^T M^T M^{-T} X_i N \bar{x}(t)\} \leq 1. \quad (20)$$

Recalling that $X_{12i} = 0$, it follows from (20) that

$$\mathbb{E}\{\bar{x}_1^T(t) \widetilde{X}_i \bar{x}_1(t)\} \leq 1. \quad (21)$$

From (21) we have $\min_{i \in \mathcal{S}} \lambda_{\min}(\widetilde{X}_i) \mathbb{E}\{\|\bar{x}_1(t)\|^2\} \leq \mathbb{E}\{\bar{x}_1^T(t) \widetilde{X}_i \bar{x}_1(t)\} \leq 1$, which implies that $\mathbb{E}\{\|\bar{x}_1(t)\|\} \leq 1/\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\widetilde{X}_i)}$.

Since A_{22i} is nonsingular for each $i \in \mathcal{S}$, (8) can be rewritten as

$$\tilde{x}_2(t) = -A_{22i}^{-1}(A_{21i}\tilde{x}_1(t) + D_{2i}\omega(t)). \quad (22)$$

Then we can deduce that

$$\begin{aligned} & \mathbb{E} \{\|\tilde{x}_2(t)\|\} \\ & \leq \|A_{22i}^{-1}\| \left(\frac{\|A_{21i}\|}{\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\tilde{X}_i)}} + \|D_{2i}\|\bar{\omega} \right). \end{aligned} \quad (23)$$

It follows from the fact $\tilde{x}(t) = [\tilde{x}_1^T(t) \ 0]^T + [0 \ \tilde{x}_2^T(t)]^T$ that

$$\mathbb{E} \{\|\tilde{x}(t)\|\} \leq \mathbb{E} \{\|\tilde{x}_1(t)\| + \|\tilde{x}_2(t)\|\} \leq r_i, \quad (24)$$

where $r_i = 1/\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\tilde{X}_i)} + \|A_{22i}^{-1}\|(\|A_{21i}\|/\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\tilde{X}_i)} + \|D_{2i}\|\bar{\omega})$.

By (24), it can be seen that

$$\frac{1}{r_i^2} \mathbb{E} \{\tilde{x}^T(t) \tilde{x}(t)\} \leq 1 \quad (25)$$

which implies that the trajectories of (7)-(8) are mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{B}((1/r_i^2)I)$. Moreover, notice that $\tilde{x}(t) = N^{-1}x(t)$, and (25) can be rewritten as $(1/r_i^2)\mathbb{E}\{x^T(t)(N^{-T}N^{-1})x(t)\} \leq 1$. By denoting $(1/r_i^2)N^{-T}N^{-1} = \tilde{X}_i$, the state reachable set of free system is mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{E}(\tilde{X}_i)$. \square

It should be noted that inequality (10) represents a nonstrict LMI. This may lead to numerical problems since equality constraints are usually not satisfied perfectly. Below, we will develop a numerically tractable and nonconservative LMI condition.

Theorem 5. *If there exist symmetric positive definite matrices $P_i \in \mathbb{R}^{n \times n}$, nonsingular matrices Q_i , and a scalar $\alpha > 0$ such that the following LMI holds for each $i \in \mathcal{S}$,*

$$\begin{bmatrix} \Psi_i (P_i E + R^T Q_i S^T)^T D_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I \end{bmatrix} < 0, \quad (26)$$

then the state reachable set of free system is mean-square bounded within the following set:

$$\bigcap_{i=1}^N \mathcal{E}(\hat{P}_i) = \bigcap_{i=1}^N \{x \in \mathbb{R}^n \mid x^T \hat{P}_i x \leq 1\}, \quad (27)$$

where

$$\begin{aligned} \Psi_i &= \text{Sym} \left(A_i^T (P_i E + R^T Q_i S^T) \right) + \sum_{j=1}^N \pi_{ij} E^T P_j E \\ &+ \alpha E^T P_i E \\ \hat{P}_i &= \frac{1}{\tilde{r}_i^2} N^{-T} N^{-1} \\ \tilde{r}_i &= \frac{1}{\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\tilde{P}_i)}} \\ &+ \|A_{22i}^{-1}\| \left(\frac{\|A_{21i}\|}{\sqrt{\min_{i \in \mathcal{S}} \lambda_{\min}(\tilde{P}_i)}} + \|D_{2i}\|\bar{\omega} \right) \\ \tilde{P}_i &= (M^{-T} P_i M^{-1})_{l \times l}, \end{aligned} \quad (28)$$

$R \in \mathbb{R}^{(n-1) \times n}$ is any matrix with full row rank and satisfies $RE = 0$; $S \in \mathbb{R}^{n \times (n-1)}$ is any matrix with full column rank and satisfies $ES = 0$.

Proof. Let $X_i = P_i E + R^T Q_i S^T$ in (26); it is easy to obtain (10) and (11). In this case, inequality (20) will be replaced by $\mathbb{E}\{\tilde{x}^T(t) N^T E^T M^T M^{-T} P_i M^{-1} M E N \tilde{x}(t)\} \leq 1$, which infers that $\mathbb{E}\{\tilde{x}_1^T(t) \tilde{P}_i \tilde{x}_1(t)\} \leq 1$ with $\tilde{P}_i = (M^{-T} P_i M^{-1})_{l \times l}$. Then, following the same lines as (22)–(25) in Theorem 4, we can get $\mathbb{E}\{x^T(t) \hat{P}_i x(t)\} \leq 1$ for any $i \in \mathcal{S}$. Therefore, the state reachable set of free system is mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{E}(\hat{P}_i)$. \square

Remark 6. In order to make the ellipsoid $\mathcal{E}(\hat{P}_i)$ as small as possible, we require $\text{trace}(\hat{P}_i) \rightarrow \max$. For this purpose, we can add the additional requirement $\tilde{P}_i > \epsilon I$ and then maximize a positive scalar ϵ , which is equivalent to the following minimization problem:

$$\begin{aligned} & \min \quad \bar{\epsilon} \\ & \text{s.t.} \quad (26), \\ & \begin{bmatrix} -\bar{\epsilon} I & I \\ I & -\tilde{P}_i \end{bmatrix} < 0, \end{aligned} \quad (29)$$

where $\bar{\epsilon} = 1/\epsilon$.

3.2. Output Reachable Set Estimation

Theorem 7. *If there exist symmetric positive definite matrices $P_i \in \mathbb{R}^{n \times n}$ and $Y_i \in \mathbb{R}^{n \times n}$, nonsingular matrices Q_i , and a scalar $\alpha > 0$ such that the following LMIs hold for each $i \in \mathcal{S}$,*

$$\begin{bmatrix} \Psi_i (P_i E + R^T Q_i S^T)^T D_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I \end{bmatrix} < 0 \quad (30)$$

$$C_i^T Y_i C_i \leq \widehat{P}_i, \quad (31)$$

then the output reachable set of free system is mean-square bounded within the following set:

$$\bigcap_{i=1}^N \mathcal{E}(Y_i) = \bigcap_{i=1}^N \{y \in \mathbb{R}^p \mid y^T Y_i y \leq 1\}, \quad (32)$$

where Ψ_i , R , S , and \widehat{P}_i are defined in Theorem 5.

Proof. By Theorem 5, LMI (30) ensures

$$\mathbb{E} \{x^T(t) \widehat{P}_i x(t)\} \leq 1, \quad \forall i \in \mathcal{S}. \quad (33)$$

With this and (31), we obtain that

$$\mathbb{E} \{x^T(t) C_i^T Y_i C_i x(t)\} \leq \mathbb{E} \{x^T(t) \widehat{P}_i x(t)\} \leq 1. \quad (34)$$

Due to the fact that $y(t) = C_i x(t)$, (34) can be rewritten as $\mathbb{E} \{y^T(t) Y_i y(t)\} \leq 1$. Thus, the output reachable set of free system is mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{E}(Y_i)$. \square

Remark 8. The output reachable set is also expected to be as small as possible. To achieve this goal, we first solve LMI (30) and get \widehat{P}_i satisfying $\text{trace}(\widehat{P}_i) \rightarrow \max$, which can be implemented by using (29). Then we add the additional requirement $C_i^T Y_i C_i > \delta I$ and maximize a positive scalar δ , which is equivalent to the following minimization problem:

$$\begin{aligned} \min \quad & \bar{\delta} \\ \text{s.t.} \quad & (31), \\ & \begin{bmatrix} -\bar{\delta}I & I \\ I & -C_i^T Y_i C_i \end{bmatrix} < 0, \end{aligned} \quad (35)$$

where $\bar{\delta} = 1/\delta$.

3.3. State-Feedback Controller Design. In this section, we turn our attention to the state-feedback control problem. Our goal here is to find a state-feedback controller, which not only stabilizes the closed-loop system, but also makes the ellipsoid bound on the reachable set of closed-loop system as small as possible.

Now, consider the state-feedback controller $u(t) = K_i x(t)$, where K_i is a gain matrix to be determined later. By using this controller, the closed-loop system can be obtained as

$$\begin{aligned} E\dot{x}(t) &= \widetilde{A}_i x(t) + D_i \omega(t) \\ y(t) &= C_i x(t), \end{aligned} \quad (36)$$

where $\widetilde{A}_i = A_i + B_i K_i$.

Theorem 9. Consider singular Markov jump system (1). If there exist nonsingular matrices $P_i \in \mathbb{R}^{n \times n}$, matrices S_i , and

scalars $\alpha > 0$, $\delta_i > 0$ such that the following LMIs hold for each $i \in \mathcal{S}$,

$$P_i^T E^T = E P_i \geq 0 \quad (37)$$

$$P_i^T E^T \leq \delta_i I \quad (38)$$

$$\begin{bmatrix} \Phi_i & D_i & \mathcal{W}_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I & 0 \\ * & * & -\mathcal{F}_i \end{bmatrix} < 0, \quad (39)$$

then the reachable set of system (1) is mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{E}(\bar{P}_i) = \bigcap_{i=1}^N \{x \in \mathbb{R}^n \mid x^T \bar{P}_i x \leq 1\}$, and the desired controller gain matrix is given by $K_i = S_i P_i^{-1}$, where

$$\begin{aligned} \Phi_i &= \text{Sym}(P_i^T A_i^T + S_i^T B_i^T) + (\alpha + \pi_{ii}) P_i^T E^T \\ \mathcal{W}_i &= [\sqrt{\pi_{i1}} P_i^T \cdots \sqrt{\pi_{i(i-1)}} P_i^T \sqrt{\pi_{i(i+1)}} P_i^T \cdots \sqrt{\pi_{iN}} P_i^T] \\ \mathcal{F}_i &= \text{diag}(P_1 + P_1^T - \delta_1 I, \dots, P_{i-1} + P_{i-1}^T - \delta_{i-1} I, P_{i+1} \\ &\quad + P_{i+1}^T - \delta_{i+1} I, \dots, P_N + P_N^T - \delta_N I) \\ \bar{P}_i &= \frac{1}{\bar{r}_i^2} N^{-T} N^{-1} \\ \bar{r}_i &= \sqrt{\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i) + \|A_{22i}^{-1}\| \left(\|A_{21i}\| \sqrt{\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i)} \right.} \\ &\quad \left. + \|D_{2i}\| \bar{\omega} \right)} \\ \check{P}_i &= (N^{-1} P_i M^T)_{1 \times 1}. \end{aligned} \quad (40)$$

Proof. Denote $X_i = P_i^{-1}$ and $S_i = K_i P_i$ for each $i \in \mathcal{S}$. Then, pre- and postmultiplying (39) by $\text{diag}(X_i^T, I, I)$ and its transpose, respectively, we obtain

$$\begin{bmatrix} \widetilde{\Phi}_i & X_i^T D_i & \widetilde{\mathcal{W}}_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I & 0 \\ * & * & -\widetilde{\mathcal{F}}_i \end{bmatrix} < 0, \quad (41)$$

where $\widetilde{\Phi}_i = \text{Sym}(\widetilde{A}_i^T X_i) + (\alpha + \pi_{ii}) E^T X_i$, $\widetilde{\mathcal{W}}_i = [\sqrt{\pi_{i1}} I \cdots \sqrt{\pi_{i(i-1)}} I \sqrt{\pi_{i(i+1)}} I \cdots \sqrt{\pi_{iN}} I]$.

Using Lemma 2, we have

$$\begin{aligned} \delta_i^{-1} P_j P_j^T &\geq P_j + P_j^T - \delta_i I, \\ j &= 1, 2, \dots, i-1, i+1, \dots, N. \end{aligned} \quad (42)$$

From (41) and (42), it is easy to obtain that

$$\begin{bmatrix} \widetilde{\Phi}_i & X_i^T D_i & \widetilde{\mathcal{W}}_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I & 0 \\ * & * & -\widetilde{\mathcal{F}}_i \end{bmatrix} < 0, \quad (43)$$

where $\tilde{\mathcal{F}}_i = \text{diag}(\delta_1^{-1}P_1P_1^T, \dots, \delta_{i-1}^{-1}P_{i-1}P_{i-1}^T, \delta_{i+1}^{-1}P_{i+1}P_{i+1}^T, \dots, \delta_N^{-1}P_NP_N^T)$.

By Schur complement, the previous matrix inequality becomes

$$\begin{bmatrix} \widehat{\Phi}_i & X_i^T D_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I \end{bmatrix} < 0, \quad (44)$$

where $\widehat{\Phi}_i = \text{Sym}(\widetilde{A}_i^T X_i) + (\alpha + \pi_{ii})E^T X_i + \sum_{j=1, j \neq i}^N \delta_j \pi_{ij} X_j^T X_j$.

Since $j \in \mathcal{S}$, (38) infers that $P_j^T E^T \leq \delta_j I$. Pre- and postmultiplying the previous matrix inequality by X_j^T and X_j , respectively, we have $E^T X_j \leq \delta_j X_j^T X_j$. This together with (44) implies

$$\begin{bmatrix} \overline{\Phi}_i & X_i^T D_i \\ * & -\frac{\alpha}{\bar{\omega}^2} I \end{bmatrix} < 0, \quad (45)$$

where $\overline{\Phi}_i = \text{Sym}(\widetilde{A}_i^T X_i) + \sum_{j=1}^N \pi_{ij} E^T X_j + \alpha E^T X_i$.

Pre- and postmultiplying (37) by X_i^T and X_i , respectively, we obtain

$$E^T X_i = X_i^T E \geq 0. \quad (46)$$

From the above discussion, we show that if (37)–(39) hold, then (45) and (46) hold. Thus, it follows from Theorem 4 that the closed-loop system can be stabilized by the designed state-feedback controller.

Next, we show that the reachable set of the closed-loop system (36) is mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{E}(\overline{P}_i)$. From (45) and (46), it is easy to show that there exists a Lyapunov function $V(x(t), r_t) = x^T(t)E^T X_i x(t)$ such that $\mathcal{L}V(x(t), r_t) + \alpha V(x(t), r_t) - (\alpha/\bar{\omega}^2)\omega^T(t)\omega(t) < 0$, where $\mathcal{L}V(x(t), r_t)$ denotes the difference of $V(x(t), r_t)$ along the trajectories of (36). It follows from Lemma 3 that $\mathbb{E}\{x^T(t)E^T X_i x(t)\} \leq 1$, which implies

$$\mathbb{E}\{\tilde{x}^T(t)N^T E^T M^T M^{-T} X_i N \tilde{x}(t)\} \leq 1. \quad (47)$$

Noting that $X_i = P_i^{-1}$, then (47) can be rewritten as

$$\mathbb{E}\{\tilde{x}^T(t)N^T E^T M^T M^{-T} P_i^{-1} N \tilde{x}(t)\} \leq 1. \quad (48)$$

Recalling that $M^{-T}P_i^{-1}N$ is a lower triangular matrix, (48) infers that

$$\mathbb{E}\{\tilde{x}_1^T(t)\check{P}_i^{-1}\tilde{x}_1(t)\} \leq 1, \quad (49)$$

where $\check{P}_i^{-1} = (M^{-T}P_i^{-1}N)_{\times 1}$.

By (49), it can be seen that

$$\begin{aligned} \min_{i \in \mathcal{S}} \lambda_{\min}(\check{P}_i^{-1}) \mathbb{E}\{\|\tilde{x}_1(t)\|^2\} &\leq \mathbb{E}\{\tilde{x}_1^T(t)\check{P}_i^{-1}\tilde{x}_1(t)\} \\ &\leq 1. \end{aligned} \quad (50)$$

Using the fact that $\min_{i \in \mathcal{S}} \lambda_{\min}(\check{P}_i^{-1}) = 1/\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i)$, we get

$$\mathbb{E}\{\|\tilde{x}_1(t)\|\} \leq \sqrt{\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i)}. \quad (51)$$

This together with (24) yields

$$\begin{aligned} \mathbb{E}\{\|\tilde{x}_2(t)\|\} \\ \leq \|A_{22i}^{-1}\| \left(\|A_{21i}\| \sqrt{\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i)} + \|D_{2i}\| \bar{\omega} \right). \end{aligned} \quad (52)$$

From (24), (51), and (52), we have that

$$\mathbb{E}\{\|\tilde{x}(t)\|\} \leq \hat{r}_i, \quad (53)$$

where $\hat{r}_i = \sqrt{\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i)} + \|A_{22i}^{-1}\| (\|A_{21i}\| \sqrt{\max_{i \in \mathcal{S}} \lambda_{\max}(\check{P}_i)} + \|D_{2i}\| \bar{\omega})$. This implies that $(1/\hat{r}_i^2) \mathbb{E}\{\tilde{x}^T(t)\tilde{x}(t)\} \leq 1$. Recalling that $\tilde{x}(t) = N^{-1}x(t)$, we have

$$\frac{1}{\hat{r}_i^2} \mathbb{E}\{x^T(t)(N^{-T}N^{-1})x(t)\} \leq 1. \quad (54)$$

By denoting $(1/\hat{r}_i^2)N^{-T}N^{-1} = \overline{P}_i$, the reachable set of closed-loop system (36) is mean-square bounded within the set $\bigcap_{i=1}^N \mathcal{E}(\overline{P}_i)$. \square

Remark 10. In order to make the ellipsoid $\mathcal{E}(\overline{P}_i)$ as small as possible, we shall carry out the following minimization problem:

$$\begin{aligned} \min \quad & \epsilon \\ \text{s.t.} \quad & (37), (38), (39), \\ & \check{P}_i \leq \epsilon I. \end{aligned} \quad (55)$$

4. Numerical Examples

In this section, two numerical simulation examples are given to show the effectiveness of the main results derived above.

Example 11. Consider the free system in (1) with the following parameters:

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} -0.8695 & -1.5760 \\ -0.2389 & 1.8258 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -0.5043 & 0.1206 \\ -1.1634 & -1.4435 \end{bmatrix},$$

$$D_1 = D_2 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix},$$

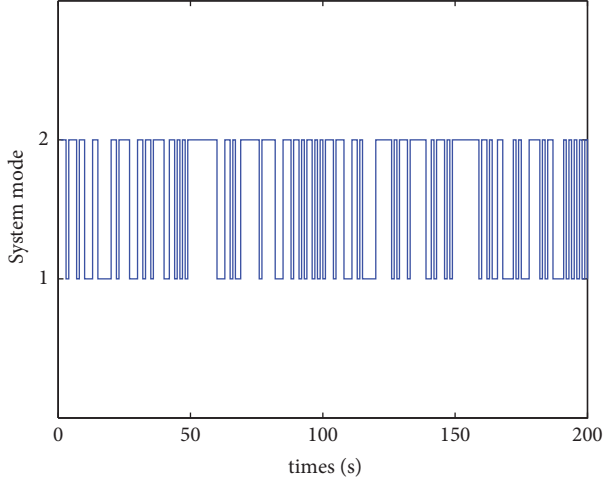


FIGURE 1: The switching between two modes.

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$
(56)

The switching between two modes is described by the following transition rate matrix:

$$\Pi = \begin{bmatrix} -3 & 3 \\ 6 & -6 \end{bmatrix}.$$
(57)

In this example, we choose $M = I$, $N = I$. By solving optimization problem (29) with the aid of `fminsearch`, the minimal $\bar{\epsilon}$ and the corresponding α are 0.1312 and 1.7419, respectively. Using the above parameter values, we can obtain $\hat{P}_1 = (1/\bar{r}_1^2)I = 2.1414I$ and $\hat{P}_2 = (1/\bar{r}_2^2)I = 0.9992I$ by solving (26). Owing to Theorem 5, the state reachable set \mathcal{R}_x is mean-square bounded within the set $\mathcal{B}(\hat{P}_1) \cap \mathcal{B}(\hat{P}_2) = \mathcal{B}(\hat{P}_1)$.

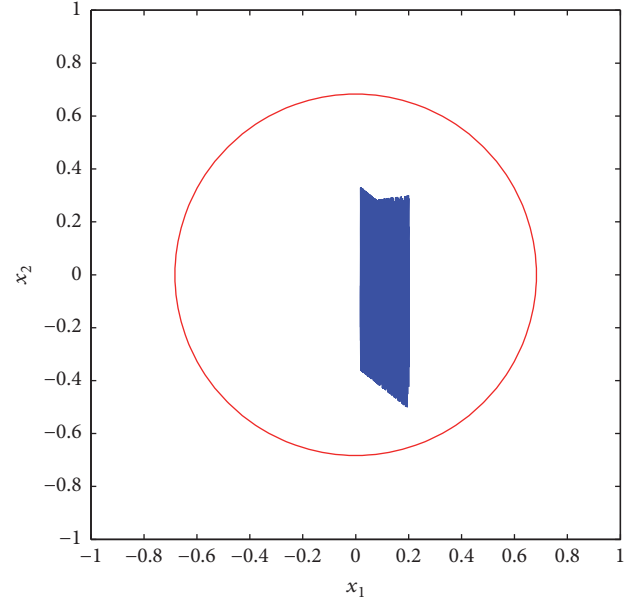
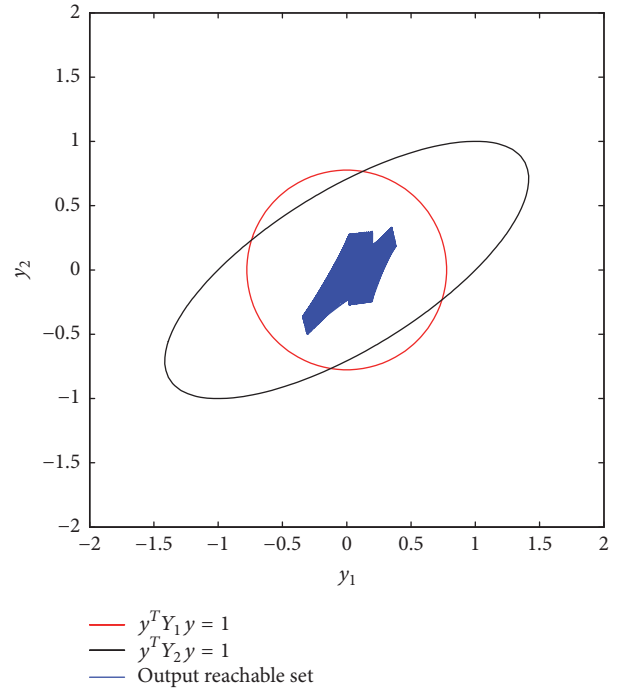
By applying Theorem 7, we have the following results:

$$Y_1 = \begin{bmatrix} 1.6561 & 0 \\ 0 & 1.6561 \end{bmatrix},$$

$$Y_2 = \begin{bmatrix} 0.9975 & -0.9975 \\ -0.9975 & 1.9950 \end{bmatrix}.$$
(58)

Therefore, the output reachable set of free system is mean-square bounded within the set $\mathcal{B}(Y_1) \cap \mathcal{E}(Y_2)$.

For simulation we assume that $x_0 = [0.2 \ 0.1303]^T$ and the disturbance is chosen as $\omega(t) = \sin(t)$. A case for stochastic variation with transition rate matrix Π is shown in Figure 1. The state reachable set \mathcal{R}_x and the ball $\mathcal{B}(\hat{P}_2)$ are depicted in Figure 2. Figure 2 shows that the trajectory of

FIGURE 2: The state reachable set \mathcal{R}_x and the bounding ball $\mathcal{B}(\hat{P}_1)$.FIGURE 3: The output reachable set \mathcal{R}_y and the bounding ellipsoids.

the system is mean-square bounded within the region $\mathcal{B}(\hat{P}_2)$. The output reachable set \mathcal{R}_y , the ball $\mathcal{B}(Y_1)$, and the ellipsoid $\mathcal{E}(Y_2)$ are depicted in Figure 3. Figure 3 shows that the output reachable set \mathcal{R}_y is mean-square bounded within the region $\mathcal{B}(Y_1) \cap \mathcal{E}(Y_2)$.

Example 12. Consider system (1) with the following parameters:

$$\begin{aligned}
 E &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \\
 A_1 &= \begin{bmatrix} 0.25 & 1.97 \\ 0.44 & 2.31 \end{bmatrix}, \\
 A_2 &= \begin{bmatrix} 1.90 & -1.72 \\ 1.87 & 0.64 \end{bmatrix}, \\
 B_1 &= \begin{bmatrix} 0.14 \\ 0.79 \end{bmatrix}, \\
 B_2 &= \begin{bmatrix} 0.34 \\ -0.50 \end{bmatrix}, \\
 D_1 &= \begin{bmatrix} 0.50 \\ 0.50 \end{bmatrix}, \\
 D_2 &= \begin{bmatrix} 0.70 \\ 0.70 \end{bmatrix}.
 \end{aligned} \tag{59}$$

The switching between two modes is described by the following transition rate matrix:

$$\Pi = \begin{bmatrix} -3 & 3 \\ 6 & -6 \end{bmatrix}. \tag{60}$$

By Theorem 9, we get the following results:

$$\begin{aligned}
 \alpha &= 0.1997, \\
 \delta_1 &= 22.1356, \\
 \delta_2 &= 18.0362, \\
 P_1 &= \begin{bmatrix} 18.0211 & 0 \\ 3.8045 & 102.2184 \end{bmatrix}, \\
 P_2 &= \begin{bmatrix} 14.7029 & 0 \\ -8.6240 & 67.2193 \end{bmatrix}.
 \end{aligned} \tag{61}$$

Therefore, the gain matrices of state-feedback controller can be obtained as

$$\begin{aligned}
 K_1 &= [-12.7027 \quad -5.3634], \\
 K_2 &= [-7.1957 \quad 4.6404].
 \end{aligned} \tag{62}$$

The corresponding parameter values \hat{r}_1^2 and \hat{r}_2^2 are, respectively, 27.7747 and 314.7993, which imply $\bar{P}_1 = (1/\hat{r}_1^2)I = 0.0360I$ and $\bar{P}_2 = (1/\hat{r}_2^2)I = 0.0032I$. Applying this controller makes the state reachable set of closed-loop system (36) mean-square bounded within the region $\mathcal{B}(\bar{P}_1)$.

For the purpose of the simulation, we assume the initial condition $x_0 = [0.3 \quad 0.9763]^T$ and the disturbance is chosen as $\omega(t) = \sin(0.2t)$. Figure 4 shows one possible switching between two modes. Figure 5 depicts the state reachable set of closed-loop system (36).

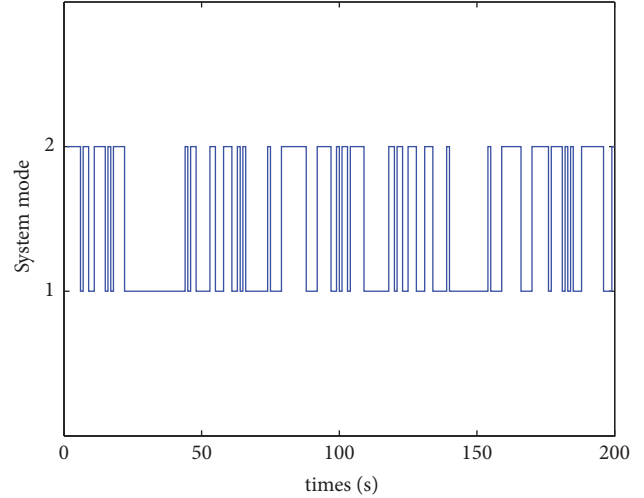


FIGURE 4: The switching between two modes.

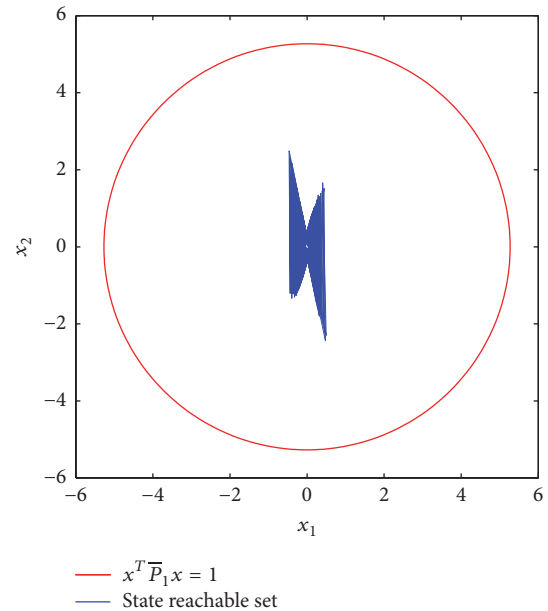


FIGURE 5: The state reachable set \mathcal{R}_x and the bounding ball $\mathcal{B}(\bar{P}_1)$.

5. Conclusions

This paper has dealt with the problems of reachable set estimation and state-feedback controller design for singular Markovian jump systems. New sufficient conditions for the state reachable set estimation and output reachable set estimation have been, respectively, derived in terms of linear matrix inequalities. Based on the estimated reachable set, the state-feedback controller has also been designed. Numerical examples and simulation results have been provided to demonstrate the effectiveness of the proposed methods.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Applied Basic Research Project of Science and Technology Department of Sichuan Province (no. 2017JY0336), a project supported by Scientific Research Fund of Sichuan Provincial Education Department (no. 16ZA0146), the National Natural Science Foundation of China (no. 11501474), the Longshan Academic Talents Research Support Program of the Southwest University of Science and Technology (no. 17LZX537 and no. 17LZXY11), and the Doctoral Research Foundation of Southwest University of Science and Technology (no. 13zx7141).

References

- [1] S. Xu and J. Lam, *Robust Control and Filtering of Singular Systems*, Springer, Berlin, Germany, 2006.
- [2] E.-K. Boukas, *Control of Singular Systems with Random Abrupt Changes*, Communications and Control Engineering, Springer, Berlin, Germany, 2008.
- [3] Y. Xia, E.-K. Boukas, P. Shi, and J. Zhang, “Stability and stabilization of continuous-time singular hybrid systems,” *Automatica*, vol. 45, no. 6, pp. 1504–1509, 2009.
- [4] L. Wu, P. Shi, and H. Gao, “State estimation and sliding-mode control of Markovian jump singular systems,” *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 55, no. 5, pp. 1213–1219, 2010.
- [5] Z.-G. Wu, J. H. Park, H. Su, and J. Chu, “Stochastic stability analysis for discrete-time singular Markov jump systems with time-varying delay and piecewise-constant transition probabilities,” *Journal of The Franklin Institute*, vol. 349, no. 9, pp. 2889–2902, 2012.
- [6] Y. Kao, J. Xie, and C. Wang, “Stabilization of singular Markovian jump systems with generally uncertain transition rates,” *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 59, no. 9, pp. 2604–2610, 2014.
- [7] Y. Ding, H. Liu, and J. Cheng, “H ∞ filtering for a class of discrete-time singular Markovian jump systems with time-varying delays,” *ISA Transactions*, vol. 53, no. 4, pp. 1054–1060, 2014.
- [8] Q. Zhang, L. Li, X.-G. Yan, and S. K. Spurgeon, “Sliding mode control for singular stochastic Markovian jump systems with uncertainties,” *Automatica*, vol. 79, pp. 27–34, 2017.
- [9] Y. Ding, S. Zhong, and S. Long, “Asymptotic stability in probability of singular stochastic systems with Markovian switchings,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 18, pp. 4312–4322, 2017.
- [10] Y. Ma, X. Jia, and D. Liu, “Finite-time dissipative control for singular discrete-time Markovian jump systems with actuator saturation and partly unknown transition rates,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 53, pp. 49–70, 2018.
- [11] B. T. Polyak, S. A. Nazin, C. Durieu, and E. Walter, “Ellipsoidal parameter or state estimation under model uncertainty,” *Automatica*, vol. 40, no. 7, pp. 1171–1179, 2004.
- [12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15 of *SIAM Studies in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.
- [13] E. K. Kostousova, “On tight polyhedral estimates for reachable sets of linear differential systems,” in *Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences, ICNPAA 2012*, pp. 579–586, Austria, July 2012.
- [14] E. Fridman and U. Shaked, “On reachable sets for linear systems with delay and bounded peak inputs,” *Automatica*, vol. 39, no. 11, pp. 2005–2010, 2003.
- [15] J.-H. Kim, “Improved ellipsoidal bound of reachable sets for time-delayed linear systems with disturbances,” *Automatica*, vol. 44, no. 11, pp. 2940–2943, 2008.
- [16] Z. Zuo, D. W. Ho, and Y. Wang, “Reachable set bounding for delayed systems with polytopic uncertainties: the maximal Lyapunov-Krasovskii functional approach,” *Automatica*, vol. 46, no. 5, pp. 949–952, 2010.
- [17] O. M. Kwon, S. M. Lee, and J. H. Park, “On the reachable set bounding of uncertain dynamic systems with time-varying delays and disturbances,” *Information Sciences*, vol. 181, no. 17, pp. 3735–3748, 2011.
- [18] Z. Zuo, Y. Chen, Y. Wang, D. W. Ho, M. Z. Chen, and H. Li, “A note on reachable set bounding for delayed systems with polytopic uncertainties,” *Journal of The Franklin Institute*, vol. 350, no. 7, pp. 1827–1835, 2013.
- [19] B. Zhang, J. Lam, and S. Xu, “Relaxed results on reachable set estimation of time-delay systems with bounded peak inputs,” *International Journal of Robust and Nonlinear Control*, vol. 26, no. 9, pp. 1994–2007, 2016.
- [20] Y. Sheng and Y. Shen, “Improved reachable set bounding for linear time-delay systems with disturbances,” *Journal of The Franklin Institute*, vol. 353, no. 12, pp. 2708–2721, 2016.
- [21] L. V. Hien and H. M. Trinh, “A new approach to state bounding for linear time-varying systems with delay and bounded disturbances,” *Automatica*, vol. 50, no. 6, pp. 1735–1738, 2014.
- [22] P. T. Nam, P. N. Pathirana, and H. Trinh, “Reachable set bounding for nonlinear perturbed time-delay systems: the smallest bound,” *Applied Mathematics Letters*, vol. 43, pp. 68–71, 2015.
- [23] Y. Sun and F. Meng, “Reachable set estimation for a class of nonlinear time-varying systems,” *Complexity*, Article ID 5876371, Art. ID 5876371, 6 pages, 2017.
- [24] N. D. That, P. T. Nam, and Q. P. Ha, “Reachable set bounding for linear discrete-time systems with delays and bounded disturbances,” *Journal of Optimization Theory and Applications*, vol. 157, no. 1, pp. 96–107, 2013.
- [25] J. Lam, B. Zhang, Y. Chen, and S. Xu, “Reachable set estimation for discrete-time linear systems with time delays,” *International Journal of Robust and Nonlinear Control*, vol. 25, no. 2, pp. 269–281, 2015.
- [26] Z. Feng and J. Lam, “On reachable set estimation of singular systems,” *Automatica*, vol. 52, pp. 146–153, 2015.
- [27] Z. Feng and W. X. Zheng, “On reachable set estimation of delay Markovian jump systems with partially known transition probabilities,” *Journal of The Franklin Institute*, vol. 353, no. 15, pp. 3835–3856, 2016.
- [28] Y. Chen, J. Lam, and B. Zhang, “Estimation and synthesis of reachable set for switched linear systems,” *Automatica*, vol. 63, pp. 122–132, 2016.
- [29] Z. Feng, W. X. Zheng, and L. Wu, “Reachable set estimation of T-S fuzzy systems with time-varying delay,” *IEEE Transactions on Fuzzy Systems*, 2016.

- [30] W. Xiang, H.-D. Tran, and T. T. Johnson, "Output reachable set estimation for switched linear systems and its application in safety verification," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 62, no. 10, pp. 5380–5387, 2017.
- [31] L. Van, N. Thanh, and H. Trinh, "New results on state bounding for discrete-time systems with interval time-varying delay and bounded disturbance inputs," *IET Control Theory & Applications*, vol. 8, no. 14, pp. 1405–1414, 2014.
- [32] L. V. Hien, N. Trung, and H. Binh, "A novel approach to state bounding for discrete-time Markovian jump systems with interval time-varying delay," *IMA Journal of Mathematical Control and Information*, vol. 33, no. 2, pp. 293–307, 2016.
- [33] H. Trinh and L. V. Hien, "On reachable set estimation of two-dimensional systems described by the Roesser model with time-varying delays," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 1, pp. 227–246, 2018.
- [34] B. Zhang, J. Lam, and S. Xu, "Reachable set estimation and controller design for distributed delay systems with bounded disturbances," *Journal of The Franklin Institute*, vol. 351, no. 6, pp. 3068–3088, 2014.
- [35] E. K. Kostousova, "On feedback target control for uncertain discrete-time bilinear systems with state constraints through polyhedral technique," in *Proceedings of the 9th International Conference for Promoting the Application of Mathematics in Technical and Natural Sciences, AMiTaNS 2017*, Bulgaria, June 2017.
- [36] Y. Chen and J. Lam, "Estimation and synthesis of reachable set for discrete-time periodic systems," *Optimal Control Applications and Methods*, vol. 37, no. 5, pp. 885–901, 2016.