

Complexity

Theory and Applications of Complex Cyber-Physical Interactions

Lead Guest Editor: Hang Su

Guest Editors: Ning Wang, Yanan Li, Yan Wu, and Juan Sandoval





Theory and Applications of Complex Cyber-Physical Interactions

Complexity

Theory and Applications of Complex Cyber-Physical Interactions

Lead Guest Editor: Hang Su


Guest Editors: Ning Wang, Yanan Li, Yan Wu, and
Juan Sandoval



Copyright © 2021 Hindawi Limited. 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.

Chief Editor

Hiroki Sayama , USA

Associate Editors

Albert Diaz-Guilera , Spain
Carlos Gershenson , Mexico
Sergio Gómez , Spain
Sing Kiong Nguang , New Zealand
Yongping Pan , Singapore
Dimitrios Stamovlasis , Greece
Christos Volos , Greece
Yong Xu , China
Xingang Yan , United Kingdom





Academic Editors

Andrew Adamatzky, United Kingdom
Marcus Aguiar , Brazil
Tarek Ahmed-Ali, France
Maia Angelova , Australia
David Arroyo, Spain
Tomaso Aste , United Kingdom
Shonak Bansal , India
George Bassel, United Kingdom
Mohamed Boutayeb, France
Dirk Brockmann, Germany
Seth Bullock, United Kingdom
Diyi Chen , China
Alan Dorin , Australia
Guilherme Ferraz de Arruda , Italy
Harish Garg , India
Sarangapani Jagannathan , USA
Mahdi Jalili, Australia
Jeffrey H. Johnson, United Kingdom
Jurgen Kurths, Germany
C. H. Lai , Singapore
Fredrik Liljeros, Sweden
Naoki Masuda, USA
Jose F. Mendes , Portugal
Christopher P. Monterola, Philippines
Marcin Mrugalski , Poland
Vincenzo Nicosia, United Kingdom
Nicola Perra , United Kingdom
Andrea Rapisarda, Italy
Céline Rozenblat, Switzerland
M. San Miguel, Spain
Enzo Pasquale Scilingo , Italy
Ana Teixeira de Melo, Portugal

Shahadat Uddin , Australia
Jose C. Valverde , Spain
Massimiliano Zanin , Spain


Contents

Research on Deviation Detection of Belt Conveyor Based on Inspection Robot and Deep Learning

Yi Liu , Changyun Miao , Xianguo Li , and Guowei Xu 


Research Article (15 pages), Article ID 3734560, Volume 2021 (2021)

Dynamic Characteristics of Microring Driven by the Symmetrically Distributed Electrostatic Force

Qingheng Meng, Yuanlin Zhang, Jin Wei, Yuh-Chung Hu, Yan Shi, and Tao Yu 


Review Article (12 pages), Article ID 1926052, Volume 2021 (2021)

Metal Roof Fault Diagnosis Method Based on RBF-SVM

Liman Yang, Lianming Su, Yixuan Wang , Haifeng Jiang, Xueyao Yang, Yunhua Li , Dongkai Shen, and Na Wang 

Research Article (12 pages), Article ID 9645817, Volume 2020 (2020)

Impedance Model-Based Optimal Regulation on Force and Position of Bimanual Robots to Hold an Object

Darong Huang, Hong Zhan, and Chenguang Yang 








Research Article (13 pages), Article ID 3561807, Volume 2020 (2020)

Universal Walking Control Framework of Biped Robot Based on Dynamic Model and Quadratic Programming

Xiaokun Leng, Songhao Piao , Lin Chang, Zhicheng He, and Zheng Zhu

Research Article (13 pages), Article ID 2789039, Volume 2020 (2020)

Multisensor-Based Autonomous Grasp Planning for Mobile Manipulator Navigation System with a Novel Soft Gripper

Heng Zhang , Yingbai Hu , Jianghua Duan , Qing Gao , Langcheng Huo , Qiwen Wang , and Yongquan Chen 


Research Article (18 pages), Article ID 5497241, Volume 2020 (2020)

A Novel Human-Like Control Framework for Mobile Medical Service Robot

Xin Zhang , Jiehao Li , Wen Qi , Xuanyi Zhou , Yingbai Hu , Hao Quan , and Zhen Wang 


Research Article (11 pages), Article ID 2905841, Volume 2020 (2020)

Simulation Framework for Cyber-Physical Production System: Applying Concept of LVC Interoperation

Byeong Soo Kim, Seunghoon Nam, Yooeui Jin, and Kyung-Min Seo 


Research Article (11 pages), Article ID 4321873, Volume 2020 (2020)

Position Tracking of a Pneumatic-Muscle-Driven Rehabilitation Robot by a Single Neuron Tuned PID Controller

Jun Zhong , Yue Zhu, Chun Zhao, Zhenfeng Han, and Xin Zhang



Research Article (9 pages), Article ID 1438391, Volume 2020 (2020)

Synchronous Analysis for Fuzzy Coupled Neural Networks with Column Pinning Controllers







Dawei Gong , Shijie Song, Michel Lopez, and Edgar N. Sanchez

Research Article (14 pages), Article ID 1397069, Volume 2020 (2020)




The Design of Compact Robotic-Assisted Needle Position System with MPC-Based Remote Control

Jing Guo, Yi Liu, Jin Wang, Chao Zeng , Jie Huang , and Chao Liu
Research Article (13 pages), Article ID 5406084, Volume 2020 (2020)


A Novel Method to Evaluate Patient-Ventilator Synchrony during Mechanical Ventilation

Liming Hao, Shuai Ren , Yan Shi, Na Wang , Yixuan Wang , Zujin Luo , Fei Xie , Meng Xu , Jian Zhang, and Maolin Cai
Research Article (15 pages), Article ID 4828420, Volume 2020 (2020)

Numerical Calibration Method for Vehicle Velocity Data from Electronic Registration Identification of Motor Vehicles Based on Mobile Edge Computing and Particle Swarm Optimization Neural Network

Jingfeng Yang , Zhiyong Luo, Nanfeng Zhang, Honggang Wang, Ming Li , and Jinchao Xiao 
Research Article (13 pages), Article ID 2413564, Volume 2020 (2020)




Research on Discriminative Skeleton-Based Action Recognition in Spatiotemporal Fusion and Human-Robot Interaction

Qiubo Zhong , Caiming Zheng, and Haoxiang Zhang
Research Article (10 pages), Article ID 8717942, Volume 2020 (2020)






Adaptive Fuzzy Control for Attitude Stabilization of Spacecraft with Deployable Composite Laminated Solar Array

Wei Zhang, Weibing Zhu , Shijie Zhang, and Xiangtian Zhao
Research Article (26 pages), Article ID 3098684, Volume 2020 (2020)

Optimization of Fuel Injection Control System of Two-Stroke Aeroengine of UAV

Yixuan Wang, Yan Shi , Maolin Cai, Weiqing Xu , Jian Zhang, Wei Zhong, and Na Wang 
Research Article (12 pages), Article ID 8921320, Volume 2020 (2020)

DOA Estimation without Source Number for Cyber-Physical Interactions

Jianzhong Li , Xiaobo Gu , Ruidian Zhan , Xiaoming Xiong , and Yuan Liu 
Research Article (5 pages), Article ID 9768361, Volume 2020 (2020)


Output Feedback Adaptive Dynamic Surface Sliding-Mode Control for Quadrotor UAVs with Tracking Error Constraints

Guoqiang Zhu, Sen Wang, Lingfang Sun , Weichun Ge, and Xiuyu Zhang 
Research Article (23 pages), Article ID 8537198, Volume 2020 (2020)

A Bi-Level Probabilistic Path Planning Algorithm for Multiple Robots with Motion Uncertainty

Jingchuan Wang , Ruochen Tai , and Jingwen Xu 
Research Article (16 pages), Article ID 9207324, Volume 2020 (2020)

Distributed Fuzzy Adaptive Control for Heterogeneous Nonlinear Multiagent Systems with Similar Composite Structure

Yongqing Fan , Tiantian Xiao, and Zhen Li
Research Article (10 pages), Article ID 4081904, Volume 2020 (2020)

Research Article

Research on Deviation Detection of Belt Conveyor Based on Inspection Robot and Deep Learning

Yi Liu ^{1,2}, Changyun Miao ³, Xianguo Li ³, and Guowei Xu ²

¹School of Mechanical Engineering, Tiangong University, Tianjin 300387, China

²Center for Engineering Internship and Training, Tiangong University, Tianjin 300387, China

³School of Electronics and Information Engineering, Tiangong University, Tianjin 300387, China

Correspondence should be addressed to Changyun Miao; miaochangyun@tiangong.edu.cn

Received 9 June 2020; Revised 30 November 2020; Accepted 6 February 2021; Published 25 February 2021

Academic Editor: Hang Su

Copyright © 2021 Yi 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.

The deviation of the conveyor belt is a common failure that affects the safe operation of the belt conveyor. In this paper, a deviation detection method of the belt conveyor based on inspection robot and deep learning is proposed to detect the deviation at its any position. Firstly, the inspection robot captures the image and the region of interest (ROI) containing the conveyor belt edge and the exposed idler is extracted by the optimized MobileNet SSD (OM-SSD). Secondly, Hough line transform algorithm is used to detect the conveyor belt edge, and an elliptical arc detection algorithm based on template matching is proposed to detect the idler outer edge. Finally, a geometric correction algorithm based on homography transformation is proposed to correct the coordinates of the detected edge points, and the deviation degree (DD) of the conveyor belt is estimated based on the corrected coordinates. The experimental results show that the proposed method can detect the deviation of the conveyor belt continuously with an RMSE of 3.7 mm, an MAE of 4.4 mm, and an average time consumption of 135.5 ms. It improves the monitoring range, detection accuracy, reliability, robustness, and real-time performance of the deviation detection of the belt conveyor.

1. Introduction

Belt conveyor is continuous transportation equipment in modern production with the advantages of large capacity, being suitable for long distance, low freight, high efficiency, stable operation, convenient loading and unloading, being suitable for bulk material transportation, etc. It has become one of the three main industrial conveyances together with automobile and train and has been widely used in coal, mines, ports, electric power, metallurgy, chemical industry, and other fields [1]. The operating condition of the belt conveyor is harsh, and the deviation fault occurs due to improper installation and adjustment, manufacturing errors, bearing idler failure, uneven distribution of materials, adhesion of slime to the driving pulley and the bend pulley, etc. [2]. The deviation fault of the belt conveyor can lead to the tearing of the conveyor belt, the material spilling, and the belt conveyor damage. To ensure safe operation, it is required to detect the deviation of the conveyor belt.

At present, the main detection method of the deviation fault is to install two sets of deviation switches on the racks on both sides of the conveyor belt. Once deviation occurs, the conveyor belt pushes the action arm of one set of the deviation switch to deflect, and the deviation switch is triggered and sends an alarm signal. If the conveyor belt continues to deviate to the set stop position, it triggers another set of deviation switch and it will control the belt conveyor to stop. This method can only detect two fixed belt deviation positions and cannot estimate the deviation degree (DD). Meanwhile, it often gives false alarms to cause downtime and affect production. In addition, the deviation switch is prone to be damaged by the conveyor belt running at high speed, and its reliability and accuracy are poor. Since the deviation fault cannot be effectively detected, the longitudinal tearing, the belt breaking, and the belt conveyor damage occur frequently.

In order to detect the deviation of the conveyor belt effectively, in recent years, researchers have paid considerable attention to the machine vision-based methods, which

are contactless and more reliable. Yang et al. [3] proposed a deviation detection method based on the image segmentation algorithm. The linear array CCD was used to collect the images of the lower surface of the running conveyor belt, and the image segmentation algorithm based on the column threshold was used to detect the conveyor belt edge, according to which the deviation of the conveyor belt can be detected qualitatively. This method is simple and efficient, but the lens of the camera is easy to be polluted by materials and dusts, which affects the imaging quality. Mei et al. [4] proposed a deviation detection method based on the image enhancement algorithm, which used the adaptive threshold and the image enhancement algorithm to detect the conveyor belt edge, and calculated the offset and the distortion of the conveyor belt, according to which the DD was estimated and the deviation fault was predicted. This method is efficient and accurate, but still with the defect that the lens is susceptible to contamination. The neural network (NN) is with prominent nonlinear mapping and feature extraction capability. It has been applied to minimally invasive surgery, the control of robotic manipulators, mechanical fault detection, human activities monitoring, robot tool dynamics identification, medical images [5–11], etc. and has demonstrated superior performance. Liu et al. [12] proposed a deep learning-based deviation detection method of the belt conveyor, which has reliable object detection and anti-jamming ability, but the error of the estimated DD is large. The cameras of the abovementioned deviation detection methods are all installed in fixed positions and can only detect the fixed positions. However, the deviation fault may occur at any position of the belt conveyor. Therefore, it is more suitable to use an inspection robot to detect the deviation along the belt conveyor. An inspection robot is a great substitute for human beings to perform periodic inspection tasks in dangerous scenes and has been widely used in the power system [13] and construction [14]. In recent years, it has been implemented in coal mines [15], but the deviation detection along the belt conveyor is still unsolved.

To address this issue, a deviation detection method of the belt conveyor based on inspection robot and deep learning is proposed in this paper, which provides a more intelligent solution for the monitoring of the belt conveyor. The main idea is to combine the deep learning algorithm and the digital image processing technology to detect the deviation of the conveyor belt on an inspection robot. To summarize, our contributions are listed as follows:

- (1) The novel deviation detection method is proposed. The inspection robot is applied to detect the deviation at any position along the belt conveyor. In order to determine the relative position of the conveyor belt, the idler outer edge is used as a reference, and the deep learning algorithm is introduced to enhance the ability of the inspection robot to deal with the complex environment. In addition, the digital image processing algorithms are combined to detect edges and correct distortion.
- (2) The ROI detector based on MobileNet SSD (M-SSD) is designed to detect the conveyor belt edge and the

exposed idler in real time. The prediction source layers of M-SSD are customized to improve the detection accuracy and speed.

- (3) A conveyor belt edge detection method based on the Hough line transform is designed to detect the conveyor belt edge, and it is more robust than other methods. An elliptical arc detection algorithm based on template matching is proposed to detect the idler outer edge. The template elliptical arcs are generated by the linear transformation of a semicircle, and this method can detect the idler outer edge efficiently.
- (4) A novel geometric correction method is proposed. The geometric correction is simplified from three-dimensional (3D) projection transformation into a combination of plane homography transformation and plane geometric transformation. The DD is estimated by the proposed DD estimation method, and it is effective in different DDs.

The rest of this paper is structured as follows. Section 2 details the proposed deviation detection method and its components. Section 3 presents the experimental platform and the result analysis. Section 4 concludes our work and gives the further research directions.

2. Materials and Methods

2.1. Schematic of the Belt Conveyor Deviation Detection Method. The inspection robot runs on the rack track beside the belt conveyor, and it moves parallel to the central axis l_c of the belt conveyor. A mobile camera is fixed on the metal frame of the inspection robot, and its optical axis is perpendicular to l_c . The mobile camera captures the image containing the conveyor belt and the exposed idlers, and the lateral axis of the image is parallel to l_c . The proposed method combines the deep learning algorithm and the image processing technology, as shown in Figure 1. It can be summarized in two phases. In phase 1, the ROI containing the conveyor belt edge and the exposed idler is extracted. In phase 2, the conveyor belt edge and the idler outer edge are detected, and the DD is estimated based on the corrected coordinates of them. This method combines the anti-jamming ability of deep learning and the precision of digital image processing and detects the deviation in complex and changeable background environments.

Phase 1 is composed of three steps: (1) the algorithm captures the image continuously and crops the image to two subimages with the same size, named *imgO* and *imgI*, respectively. (2) We input the two subimages to the well-trained ROI detector OM-SSD to extract the ROI containing the conveyor belt edge and the exposed idler, named *ROI_O* and *ROI_I*, respectively, and they may contain the outer idler or the inner idler (the idler on the outer side of the conveyor belt appears in the upper half of the image, and the inner side one appears in the lower half, which are called the outer idler and the inner idler, respectively). (3) If the ROIs are extracted successfully, we step into phase 2; otherwise, the procedure will be aborted and we go with the next image. Phase 2 consists of the following four steps: (1) the

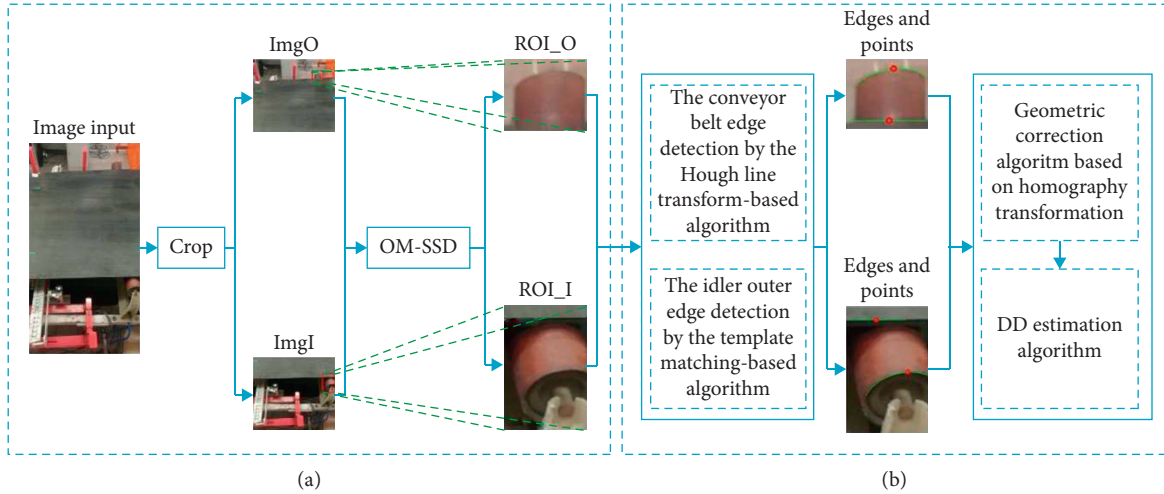


FIGURE 1: Schematic of the deep-learning-based belt conveyor deviation detection method. The inspection robot captures an image and divides it into $imgO$ and $imgI$; then, they are input into the ROI detector (OM-SSD) to extract the ROI_O and ROI_I , respectively. The ROIs are converted into gray images, and then, the conveyor belt edge detection algorithm and the idler outer edge detection algorithm are implemented on them to get the edges. The points are the midpoints of the detected lines representing the conveyor belt edges and the vertices of the idler outer edges, respectively. The coordinates of the points are corrected by the geometric correction algorithm and then used to estimate the DD. The distances between the two points in ROI_O and ROI_I represent the lengths of the exposed outer and inner idler, and the difference between the lengths indicates the DD. (a) Phase 1: ROI extraction. (b) Phase 2: DD estimation.

conveyor belt edge is detected by using the Hough line transform-based algorithm. (2) The idler outer edge is detected by the elliptical arc detection algorithm based on template matching. (3) The coordinates of the conveyor belt edge point and the idler outer edge point are corrected by implementing the geometric correction algorithm based on homography transformation; then, the length of the exposed idlers can be calculated. (4) The DD is estimated based on the length of the exposed idlers.

2.2. ROI Extraction Algorithm Based on M-SSD

2.2.1. The Framework of the ROI Extraction Algorithm.

ROI extraction can be regarded as object detection. According to the detection stage, the methods of deep-learning-based object detection can be divided into 2 categories: two-stage and one-stage detection methods. The two-stage detection method is represented by faster RCNN [16], which is comprised of two stages. In stage 1, the proposal bounding boxes and the objectness scores are predicted simultaneously, and in stage 2, the region proposals are used for detection and regression. Faster RCNN can be considered as a combination of fast RCNN [17] and region proposal network (RPN), and it performs well in the public datasets. However, fast RCNN and RPN have to be trained alternately to share the parameters of the convolution layers and the prediction layers, which makes it inconvenient for integration and implementation. The one-stage detection method is represented by SSD [18] and YOLO [19], and they do not have an explicit generation stage of the proposal bounding boxes. Compared with YOLO and faster RCNN, SSD adopts pyramid architecture, imitates RPN to generate default boxes at each position from multiple source layers, and predicts each classification score of each

default box, thus making full use of each feature maps and avoiding the alternative training of the classification network and RPN, which makes it more flexible and widely used in different applications [18, 20].

2.2.2. Object Detect Network Based on MobileNet.

The superior nonlinear mapping capability of the convolution neural network (CNN) is mostly derived from the depth and the capacity of the network, and the model goes deeper and larger for better performance in accuracy, such as AlexNet [21], VGGNet [22], GoogLeNet [23], and ResNet [24]. However, the training and deployment of them require higher hardware costs (memory and GPU), which limit their application in embedded and mobile devices. Therefore, streamlining the network has become an active research area in recent years [25–27]. The key point of the network pruning is to balance the tradeoff between accuracy and costs, that is, to reduce the parameters and computations dramatically with a slight degradation in accuracy. MobileNet [27] is an efficient CNN specified for mobile vision applications. The core is a streamlined architecture named depthwise separable convolution (DSC). Given an input feature map of size $D_W \times D_H \times M$, a convolution kernel of size $D_K \times D_K \times N$, the standard convolution assuming stride one and padding, has the computational cost of

$$MAdd_s = D_W D_H M D_K D_K N, \quad (1)$$

where $MAdd_s$ is the multiadds [27], D_W , D_H , and M are the width, height, and number of channels of the input feature map, respectively, the width and the height of the convolution kernel are D_K , and the number of channels is N . The DSC divides the standard convolution into two

parts: a depthwise convolution and a pointwise convolution. The former is applied as a single filter per each input channel and outputs a feature map of size $D_W \times D_H \times M$, and the latter creates a linear combination of the output by using a simple convolution of size $1 \times 1 \times N$. Then, the computational cost of the DSC with stride one and padding can be calculated as

$$\text{MAdds}_D = D_W D_H M D_K D_K + D_W D_H M N. \quad (2)$$

Then, the computational cost ratio (CCR) of the DSC and the standard convolution can be written as

$$\frac{\text{MAdds}_D}{\text{MAdds}_s} = \frac{1}{N} + \frac{1}{D_K D_K}. \quad (3)$$

The state-of-art networks usually use small kernels [22], i.e., $D_K = 3$. When the output channel N gets larger, the CCR can be close to $1/9$. In addition, the number of parameters P_s in the standard convolution is

$$P_s = D_W D_H M D_K D_K N, \quad (4)$$

while in the DSC, the number of parameters P_D is

$$P_D = M D_K D_K + M N, \quad (5)$$

and the ratio of them is as follows:

$$\frac{P_D}{P_s} = \frac{1}{D_W D_H N} + \frac{1}{D_K D_K D_W D_H}. \quad (6)$$

It can be proved that the larger the feature map and the number of the output channels get, the more the computations and the parameters decrease, and the experimental results show that the degradation in performance of the DSC is almost negligible [27]. With the lightweight architecture, deep CNNs and object detection frameworks can be deployed in the embedded devices successfully. The feature extraction network of M-SSD is shown in Table 1, with 22 layers, and all the convolution layers are followed by batchnorm layers and rectified linear unit (ReLU) layers, including the standard convolution (Conv) layers and the depthwise and pointwise convolution layers in the DSC.

2.2.3. ROI Detector. To detect objects of different sizes, M-SSD uses L11, L13, L15, L17, L19, and L21 as the prediction source layers in predicting the object bounding box. For the k -th of the 6 layers, the ratio S_k of the default box to the input image is determined as

$$S_k = S_{\min} + \frac{S_{\max} - S_{\min}}{m - 1} (k - 1), \quad k \in [1, m], \quad (7)$$

where S_{\min} and S_{\max} are the minimum and maximum of the ratio, which are 0.2 and 0.9, respectively. $m = 6$ is the number of the prediction source layers. To detect objects with different shapes, M-SSD sets different aspect ratios r_k for each default box at each position of the k -th layer, which can be determined as follows:

TABLE 1: The backbone of M-SSD.

Layer	Input size	Operator	Stride	Output channel
L0	$300 \times 300 \times 3$	Conv	2	32
L1	$150 \times 150 \times 32$	DSC	1	64
L2	$150 \times 150 \times 64$	DSC	2	128
L3	$75 \times 75 \times 128$	DSC	1	128
L4	$75 \times 75 \times 128$	DSC	2	256
L5	$38 \times 38 \times 256$	DSC	1	256
L6	$38 \times 38 \times 256$	DSC	2	512
L7	$19 \times 19 \times 512$	DSC	1	512
L8	$19 \times 19 \times 512$	DSC	1	512
L9	$19 \times 19 \times 512$	DSC	1	512
L10	$19 \times 19 \times 512$	DSC	1	512
L11	$19 \times 19 \times 512$	DSC	1	512
L12	$19 \times 19 \times 512$	DSC	2	1024
L13	$10 \times 10 \times 1024$	DSC	1	1024
L14	$10 \times 10 \times 1024$	Conv	1	256
L15	$10 \times 10 \times 256$	Conv	2	512
L16	$5 \times 5 \times 512$	Conv	1	128
L17	$5 \times 5 \times 128$	Conv	2	256
L18	$3 \times 3 \times 256$	Conv	1	128
L19	$3 \times 3 \times 128$	Conv	2	256
L20	$2 \times 2 \times 256$	Conv	1	64
L21	$1 \times 1 \times 64$	Conv	2	128

$$r_k = \begin{cases} \left\{ 1, 2, \frac{1}{2} \right\}, & k = 1, \\ \left\{ 1, 2, \frac{1}{2}, 3, \frac{1}{3} \right\}, & k \in [2, 6]. \end{cases} \quad (8)$$

In addition, a square default box with a scale of $\sqrt{S_k S_{k+1}}$ is added when $k \in [2, 6]$. Assuming an input image of size $300 \times 300 \times 3$, the number of default boxes is 1917.

In this paper, OM-SSD algorithm for extracting the ROI containing the conveyor belt edge and the exposed idler is proposed. On the basis of M-SSD, OM-SSD sets the aspect ratios of L11's default boxes to the same ratios as those of other layers, and L14–L21 are removed; L11 and L13 are the only source layers for prediction, and the reason is as follows. For the extraction of the ROI, it is the same thing as detecting the image of the exposed idler with an extended small patch of the conveyor belt. Since the distance between the inspection robot and the idler does not change very much, the image size of the ROI varies in a small range, and the default boxes with unrelated scale contribute almost nothing to the detection but slowing down the speed [18], so the prediction source layers L14–L21, which are used to generate large scale default boxes in M-SSD, are removed. Since the idler is elongated, the aspect ratio of the exposed idler varies greatly, so the aspect ratios of L11 are expanded. Since most feature extraction layers in OM-SSD are DSCs, the number of the parameters in the model is small, and then, OM-SSD has less trouble with overfitting [27]. The original loss function of SSD [18] is adopted for training:

$$L(x, c, l, g) = \frac{(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g))}{N_b}, \quad (9)$$

where N_b is the number of matched default boxes, x is the match indicator of the default box and the ground truth, c is the prediction confidence of multiple classes, and l and g are the parameters of the predicted box and the ground truth box, respectively. Confidence loss $L_{\text{conf}}(x, c)$ and localization loss $L_{\text{loc}}(x, l, g)$ are the main components of the loss function and are weighted by weight term α , which is set to 1.

In order to extract the ROI efficiently, the input image is divided into two subimages: imgO and imgI, which may contain the outer idler and the inner idler, respectively. Then, they are resized to $300 \times 300 \times 3$ and input into OM-SSD. The detector outputs the label indicating the classification of the idler and the bounding box of the ROI.

2.3. Conveyor Belt Edge Detection Algorithm Based on the Hough Line Transform. The conveyor belt edge in the ROI can be regarded as a straight line. The conventional algorithms of line detection can be divided into two categories: Hough transform-based and line segment-based algorithms [28]. The latter has the advantages of high efficiency and accuracy; however, this method is based on region grow, and if a long line is blocked or blurred partially, it is often detected as multiple lines. The former does not suffer from this drawback and is more robust. The Hough line transform maps the edge point (x, y) in the image space to a sine-like curve in the parameter space by

$$\rho = x \cos \theta + y \sin \theta, \quad (10)$$

where ρ is the distance of the line and the origin and θ is the angle between the x -axis and the line. The curves corresponding to the edge points on the same line will intersect at a point (ρ_0, θ_0) . The plausible lines can be obtained by substituting the coordinates of the intersection points of the curves into equation (10).

The conveyor belt edge detection algorithm based on the Hough line transform consists of six steps and is described in Algorithm 1.

To reduce noise and keep sharpness, the kernel size of the Gaussian filter is set to 3, and the standard deviation is 1.0. The low and high threshold of the Canny edge detector are 0.2 and 0.4 of the maximum gradient, respectively, and the accumulator threshold for the plausible lines extraction is set to 30 empirically. The prior knowledge of the conveyor belt edge is as follows. The angle between the conveyor belt edge and the x -axis is no more than 15° , and then, the plausible lines with $|\theta| > 15^\circ$ are discarded. In addition, ROI_I may contain the edges of the upper and lower surface of the conveyor belt, and generally, the upper one is more prominent, so the uppermost proposal line is considered as the conveyor belt edge. ROI_O only contains the upper surface edge; thus, the longer proposal line at the bottom of the ROI_O is extracted.

2.4. Idler Outer Edge Detection Based on Template Matching

2.4.1. Features of the Idler Outer Edge in the Image. The general structure of the idler is a cylinder, and the shape of its outer edge is a standard circle. However, the image of

the idler outer edge obtained by the inspection robot is an elliptical arc with large gaps, varying scales, and interference with similar arcs, as shown in Figure 2; this is caused by the following reasons: (1) due to the shooting position, the angle between the plane of the idler outer edge and the image plane is not 0° , which results in the standard circle being compressed vertically, and only the upper half of the outer idler outer edge can be obtained and the lower part of the inner idler outer edge is blocked by the bracket. (2) The idler outer edge may be sheared horizontally with variations in scale, which is caused by the movement of the inspection robot. (3) The protruding edge on the idler outer edge will induce a pseudoarc, and the shadow of the conveyor belt edge on the idler will also appear as an interference.

2.4.2. Idler Outer Edge Detection Algorithm. The general arc detection algorithms can be classified into two categories: Hough transform-based [30] and gradient region growth-based algorithms [31]. The former suffers from tremendous computation, while the latter is more suitable for the images with simple content. More importantly, they do not perform well in detecting an elliptical arc.

The elliptical arc of the idler outer edge can be generated by horizontal scaling, vertical scaling, and horizontal shearing of a semicircle, according to which an idler outer edge detection algorithm based on template matching is proposed. Firstly, the template elliptical arcs are generated based on the prior transformation parameters, and then, the corresponding template elliptical arcs and the edge points in I_e are traversed to find the one with the highest matching confidence. The proposed algorithm is composed of the following three steps: (1) given the radius r_i and the vertical scaling factor s_j , the coordinate (w, z) of an arbitrary point on the base arc is determined as

$$z = s_j \sqrt{r_i^2 - (w - r_i)^2}, \quad (11)$$

where $w \in [0, 2r_i]$ and r_i and s_j are used to scale the base arc horizontally and vertically. Given the horizontal shearing factor c_k , the coordinate (x_a, y_a) of an arbitrary point on the template elliptical arc can be determined as

$$\begin{cases} x_a = w + c_k z, \\ y_a = z. \end{cases} \quad (12)$$

The images of the representative template elliptical arcs with different parameters are shown in Figure 3. (2) We consider $I_e(x, y)$ in I_e as the origin coordinate, count the number of the edge pixels $n_e(x, y)$ on the template elliptical arc route and the number of the valid pixels $n_v(x, y)$, and traverse all the template elliptical arcs determined by the combinations of $r_i = R_{\text{min}} + iR_{\text{step}}$, $i \in [0, 1, 2, \dots, N_r - 1]$, $s_j = S_{\text{min}} + jS_{\text{step}}$, $j \in [0, 1, 2, \dots, N_s - 1]$, and $c_k = C_{\text{min}} + kC_{\text{step}}$, $k \in [0, 1, 2, \dots, N_c - 1]$ (R_{min} , S_{min} , and C_{min} are the minimum values of the radius, the vertical scaling factor, and the horizontal shearing factor, R_{step} , S_{step} , and C_{step} are their steps, and N_r , N_s , and N_c are their numbers, respectively), and the matching confidence $p(x, y, i, j, k)$ that

Input: ROI image ROI_O or ROI_I.

Output: Coordinates of the start and end points of the proposal line.

- (1) Convert ROI_O or ROI_I into gray image;
- (2) Denoise the gray image by using Gaussian filter;
- (3) Obtain the edge map I_e by using Canny edge detector [29];
- (4) Map the edge points in I_e to the parameter space by using Hough line transform, then set the accumulators;
- (5) Extract the plausible lines by setting a proper accumulator threshold;
- (6) Extract the proposal line from the plausible lines by the prior knowledge of the conveyor belt edge.

ALGORITHM 1: Conveyor belt edge detection algorithm based on the Hough line transform.

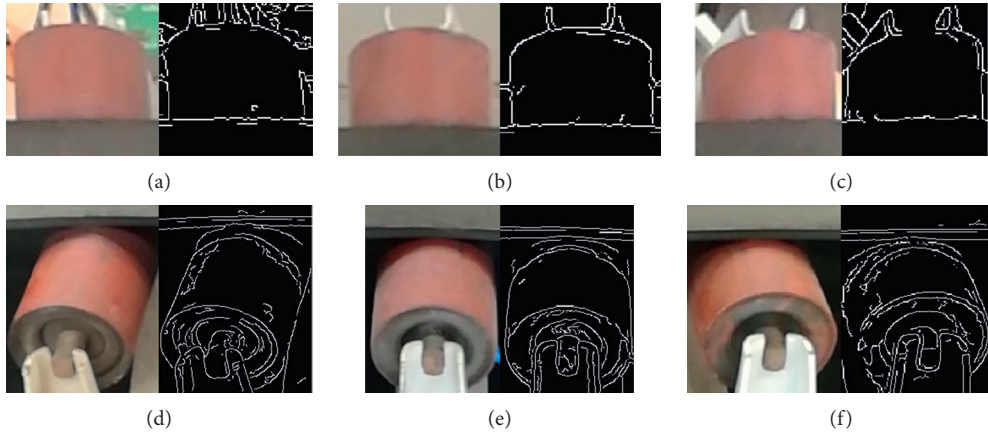


FIGURE 2: Typical images of the ROI and their edge maps. With the movement of the inspection robot, the position of the ROI in the image changes from left to right, the idler outer edge presents an elliptical arc, and the edge map of the ROI_I is disturbed by similar elliptical arcs. (a) ROI_O on the left, (b) ROI_O in the middle, (c) ROI_O on the right, (d) ROI_I on the left, (e) ROI_I in the middle, and (f) ROI_I on the right.

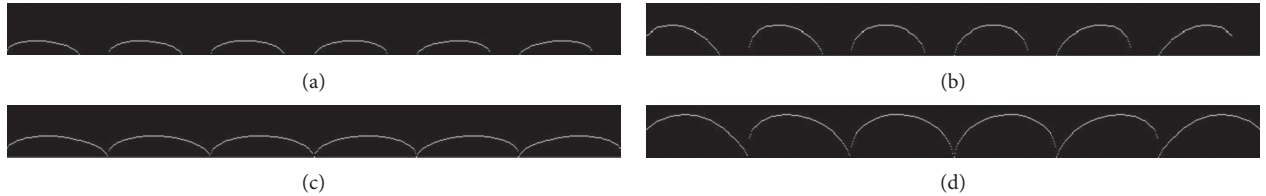


FIGURE 3: Typical template elliptical arcs with different parameters. (a) $r_i = 43$, $s_j = 0.4$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 . (b) $r_i = 43$, $s_j = 0.8$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 . (c) $r_i = 60$, $s_j = 0.4$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 . (d) $r_i = 60$, $s_j = 0.8$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 .

indicates the probability of a true idler outer edge can be defined as

$$p(x, y, i, j, k) = \max \frac{n_e(x, y)}{n_v(x, y)}. \quad (13)$$

C_{\min} varies with the horizontal location of the ROI in the image. Assuming that the abscissa value of the left boundary of ROI_O or ROI_I is x_l and its ratio to the width of imgO or imgI is r_x , C_{\min} can be determined as

$$C_{\min} = C_{\text{step}} \text{floor} \left(\frac{r_x (C_r - C_l) + C_l}{C_{\text{step}}} \right), \quad (14)$$

by which C_{\min} is aligned to an integral multiple of C_{step} within $[C_l, C_r]$. (3) We traverse all of the edge points in I_e , repeat step (2), calculate all the $p(x, y, i, j, k)$, and then, sort them in descending order and output the parameters (x, y, i, j, k) of top K. The origin coordinate of the idler outer edge is located in the left half of I_e ; thus, only the edge points on the left half are traversed to speed up. Generally, matching confidence of the true idler outer edge is higher than that of the pseudoarc; then, the parameters with the highest matching confidence are used to describe the idler outer edge.

We summarize the idler outer edge detection algorithm based on template matching in Algorithm 2.

Input: Edge map I_e , prior transformation parameters R_{\min} , R_{step} , N_r , S_{\min} , S_{step} , N_s , C_l , C_r , C_{step} and N_c .

Output: Idler outer edge parameters (x, y, i, j, k) .

- (1) Compute the template elliptical arcs by using equations (11) and (12);
- (2) Consider the edge point $I_e(x, y)$ in the left half of I_e as the origin coordinate, compute the matching confidence $p(x, y, i, j, k)$ by using equation (13);
- (3) Traverse all the edge points $I_e(x, y)$ in I_e and repeat step 2, then output the parameters (x, y, i, j, k) with the highest matching confidence.

ALGORITHM 2: Idler outer edge detection algorithm based on template matching.

2.5. Geometric Correction Based on Homography Transformation

2.5.1. Equivalence of Conveyor Belt Imaging. Geometric correction is required for the measurement with a monocular camera. The surface of the trough conveyor belt is not a plane, and distortion will occur when the surface is projected onto the image plane.

To correct this distortion, the imaging process is analyzed as follows. The deviation of the conveyor belt is defined in its cross-section plane, and the imaging model is shown in Figure 4. a'' and d'' are the vertices of the outer edges of the inner and outer idlers, b_1 and c_1 are the edge points of the conveyor belt upper surface, and their projection points on the image plane I are a , d , b , and c , respectively. The thickness of the conveyor belt should not be ignored and is marked as b_1b_2 or c_1c_2 , where b_2 and c_2 are the edge points of the lower surface of the conveyor belt. According to the geometry, the length of the exposed idlers $a''b_2$ and c_2d'' can be obtained by the internal and external parameters of the camera as well as the pixel distance of ab and cd , and they can be used to estimate the DD of the conveyor belt. However, to obtain these parameters directly, 3D perspective transformation is needed, which requires more reference points that are difficult to get in engineering.

To address this issue, the imaging process is equivalent to two projections. We suppose there is a horizontal auxiliary plane I' with a'' and d'' on it. b' and c' are the projection points of b_1 and c_1 on I' with O_c as the perspective center, and the first equivalent projection is that the image of the 3D objects such as the conveyor belt and the exposed idlers are projected to plane I' . The second one is that the image on I' is projected to I . The process of the measurement by a monocular camera can be regarded as the inverse imaging process described above.

2.5.2. Geometric Correction Algorithm. As the imaging process is equivalent to two projections, the geometric correction is simplified from 3D projection transformation

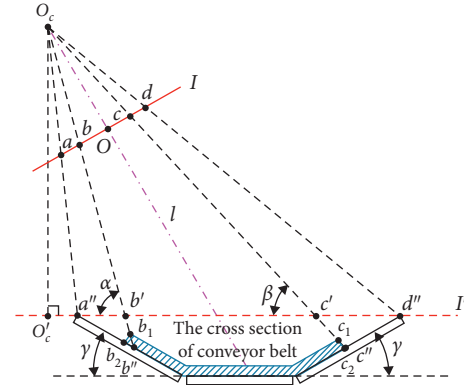


FIGURE 4: The imaging model of the conveyor belt in the cross-section plane.

into a combination of plane homography transformation and plane geometric transformation.

For any point (u, v) on the image plane I , there is a nonsingular matrix H of size 3×3 and a scale factor ρ that can map (u, v) to (x', y') on I' linearly. The homogeneous expression is as follows:

$$\begin{bmatrix} \frac{x'}{\rho} \\ \frac{y'}{\rho} \\ 1 \end{bmatrix} = \begin{bmatrix} x'_a \\ y'_a \\ 1 \end{bmatrix} = H \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (15)$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}.$$

There are eight degrees of freedom in H , and four pairs of calibration points, at least, are needed to solve equation (15). The calibration points can be obtained by a rectangular calibration board placed on I' . The width and aspect ratio of the calibration board are W_a and R_a , and the calibration points

(u_i, v_i) ($i = 1, 2, 3, 4$) are the coordinates of the upper left, upper right, lower left, and lower right corner of the calibration board on the image plane I , respectively. The corrected coordinates (u_{ai}, v_{ai}) ($i = 1, 2, 3, 4$) can be defined as

$$\left\{ \begin{array}{l} u_{a1} = u_1, \\ v_{a1} = v_1, \\ u_{a2} = u_1 + \sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}, \\ v_{a2} = v_1, \\ u_{a3} = u_1, \\ v_{a3} = v_1 + \frac{\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}}{R_a}, \\ u_{a4} = u_{a2}, \\ v_{a4} = v_{a3}, \end{array} \right. \quad (16)$$

and then, ρ is calculated by

$$\rho = \frac{W_a}{\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}}, \quad (17)$$

and the elements in H can be obtained by solving

$$\left\{ \begin{array}{l} u_{ai} = \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + 1}, \\ v_{ai} = \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + 1}, \end{array} \right. \quad i = 1, 2, 3, 4. \quad (18)$$

When the coordinates of a , d , b , and c are detected, they can be transformed to a'' , d'' , b' , and c' by equation (15). The vertical projection point of the camera optical center O_C in plane I' is O_C' , the vertical and horizontal distance between O_C and a'' in the cross section plane are $O_C O_C'$ and $O_C' a''$, and the angle between the idler center axis and the horizontal plane is γ ; then, the length of the exposed idlers can be calculated by the plane geometric transformation as follows:

$$\left\{ \begin{array}{l} a'' b_2 = a'' b' \cos \gamma + \frac{a'' b' \sin \gamma - b_1 b_2}{\tan(\arctan(O_C O_C' / O_C' a'' + a'' b') - \gamma)}, \\ c_2 d'' = c' d'' \cos \gamma - \frac{c' d'' \sin \gamma - b_1 b_2}{\tan(\arctan(O_C O_C' / O_C' a'' + a'' c') + \gamma)}. \end{array} \right. \quad (19)$$

2.6. DD Estimation. The slope of the conveyor belt edge is gentle, and the midpoints of the lines detected in Section 2.3 can be regarded as the edge points b and c in the image. Meanwhile, the vertices of the detected elliptical arcs in

Section 2.4 are used as the vertices a and d in the image. Then, $a'' b_2$ and $c_2 d''$ can be calculated by equation (15) and (19). According to the Chinese industry standard (GB/T 10595-2017), the deviation of the conveyor belt is defined as the deviation between the centerlines of the conveyor belt and the belt conveyor, and the conveyor belt is considered to be in normal operation when the deviation is within 5% of the belt width, and early warning or shutdown command should be issued when that exceeds 5% consistently [4].

The DD is defined as the percentage of the conveyor belt deviation and its width W_b . When both of $a'' b_2$ and $c_2 d''$ can be obtained, the DD is calculated as follows:

$$DD = \frac{a'' b_2 - c_2 d''}{2W_b} \times 100\%. \quad (20)$$

However, $a'' b_2$ and $c_2 d''$ are not always available. When there is only $a'' b_2$, the DD should be calculated by

$$DD = \frac{a'' b_2 - 0.5(W_i - W_b)}{W_b} \times 100\%, \quad (21)$$

where W_i is the groove length of the idler set, which is longer than $a'' d''$. $(W_i - W_b)$ is the groove length of the exposed idler set, which can also be obtained from the previous normal detection results:

$$W_i - W_b = a'' b_2 + c_2 d''. \quad (22)$$

When there is only $c_2 d''$, the DD should be calculated by

$$DD = \frac{0.5(W_i - W_b) - c_2 d''}{W_b} \times 100\%. \quad (23)$$

If $a'' b_2$ and $c_2 d''$ are both unavailable, this detection process will be aborted. The DD is a signed variable, a positive one indicates that the conveyor belt deviates to the outer side, and a negative one indicates that of the inner side.

3. Results and Discussion

An experimental setup was built to validate the proposed method. To illustrate the advantages of the proposed ROI detector, OM-SSD is compared with M-SSD in the self-built dataset. Then, the performance of the deviation detection algorithm is verified at different standard DDs and shooting heights, and the preferred shooting height is given through the comparative experiments. Furthermore, the time consumption of the proposed algorithms is examined on a NVIDIA Jetson TX2.

3.1. Experimental Setup. The inspection robot and belt conveyor experimental platform is shown in Figure 5. The inspection robot runs on the rack track beside the belt conveyor and is equipped with a source light to make up the poor illumination. The mobile camera (camera sensor: SONY IMX298, resolution: 1080×1920 , aperture: $f/2.0$) is fixed on the metal frame, which moves along with the inspection robot and is height adjustable. The belt conveyor is 7.7 m in length, 1 m in width, and 1.55 m in height (from the

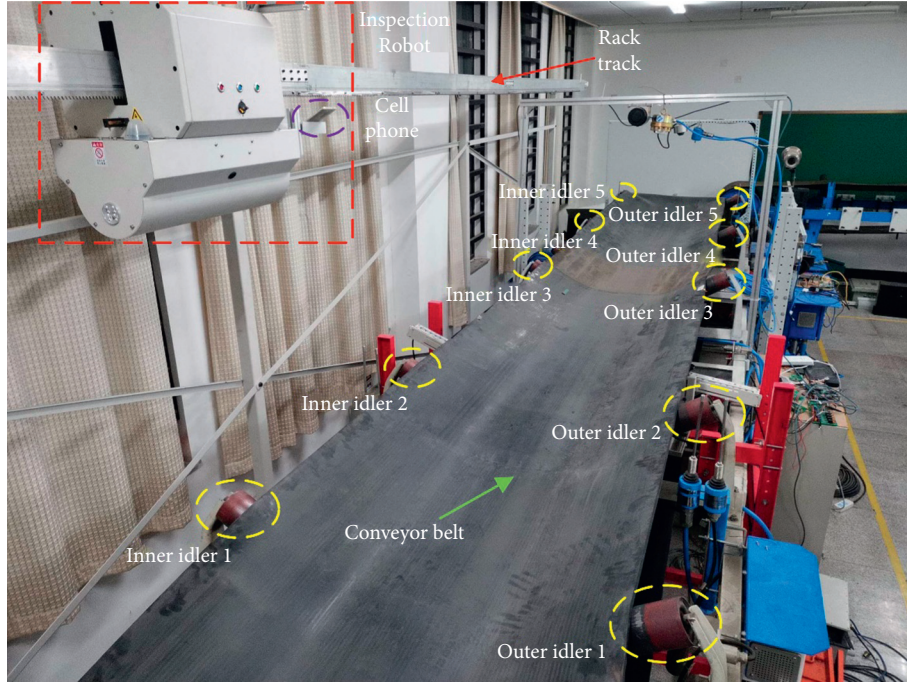


FIGURE 5: The inspection robot and belt conveyor experimental platform.

vertex of the side idler outer edge to the ground), and the belt is 11 mm in thickness. The upper belt is supported by 5 sets of idlers (diameter: 89 mm, γ : 30°) and only the side idlers of each set could be captured, which are named as inner idler 1–5 and outer idler 1–5, respectively.

The belt conveyor in mine roadway is close to the bearing wall, so the horizontal distance $O_c'a''$ cannot vary too much and is set to 0.35 m empirically. Due to the limitation of the tunnel height and the integrity of the imaging content, the shooting height from O_c to the ground varies between 2.4 m and 3.0 m, and O_cO_c' can be obtained by subtracting the belt conveyor height from the shooting height. To simulate the effect of the shooting height, it is set to 2.4 m, 2.6 m, 2.8 m, and 3.0 m, respectively. To simulate the deviation of the belt conveyor, the standard DD is adjusted to +5%, 0%, and -5%, respectively. It should be noted that only one set of the idlers and the corresponding conveyor belt edges can be used to indicate the standard DD due to the twist of the conveyor belt when it deviates. In these 12 combinations, the horizontal and vertical perspective angles are set to -60° and 0° , respectively, the inspection robot captures an image every 0.06 m along the conveyor belt from the head to tail, and then, 12 sets of images are obtained and the images without idlers are discarded. The validation dataset of the proposed deviation detection method is composed of 184 images containing the specific idlers, which are picked out from each set, when the standard DD is +5% or 0%, the images containing the idler 2 are picked out, and when the standard DD is -5%, the images containing the idler 4 are picked out; the typical images are shown in Figure 6. Each of the remaining images is cropped to two subimages of $1080 \times 960 \times 3$ pixel for the fine tuning of the ROI detector. The cropped subimages are randomly sorted and divided into

three sets for training, validation, and testing, and the number of them is 551, 237, and 388, respectively.

The desktop computer configured as Intel i7-7820X3.6 GHz CPU, 16 GB memory, NVIDIA RTX 2080Ti, and Ubuntu 18.04 is used for the fine tuning of the ROI detector, and the performance validation of the proposed deviation detection method is conducted on a NVIDIA Jetson TX2, which is configured as Ubuntu 16.04, Python 2.7, OpenCV 4.11, and Caffe. The algorithm proposed in Section 2.4 is written in C++ and embedded in the source code of OpenCV 4.11 to compile. The integration of all the proposed algorithms is based on Python 2.7.

3.2. Training Result of the ROI Detector. The backbone network parameters of OM-SSD adopt the parameters of M-SSD (<https://github.com/chuanqi305/MobileNet-SSD>) with a mean average precision (mAP) of 0.727, which is pretrained on PASCAL VOC0712. OM-SSD is fine tuned on the dataset mentioned in Section 3.1. The optimization algorithm is the root mean square prop (RMSProp) with a base learning rate of 0.0002, which is reduced by half at the 5000th and 20000th iteration. The fine tuning ends after 30000 iterations. Before deploying the trained model, the batchnorm layers and the scale layers are merged with the previous convolution layers, and the compute unified device architecture (CUDA) units are used in the convolution computation to further speed up the forward inference. To investigate the influence of the different prediction source layers on the model, a comparative experiment of OM-SSD with different prediction source layers was implemented, and the configuration is shown in Table 2, where the selected layers are used as the prediction source layers, and all the feature extraction layers behind the last selected one are

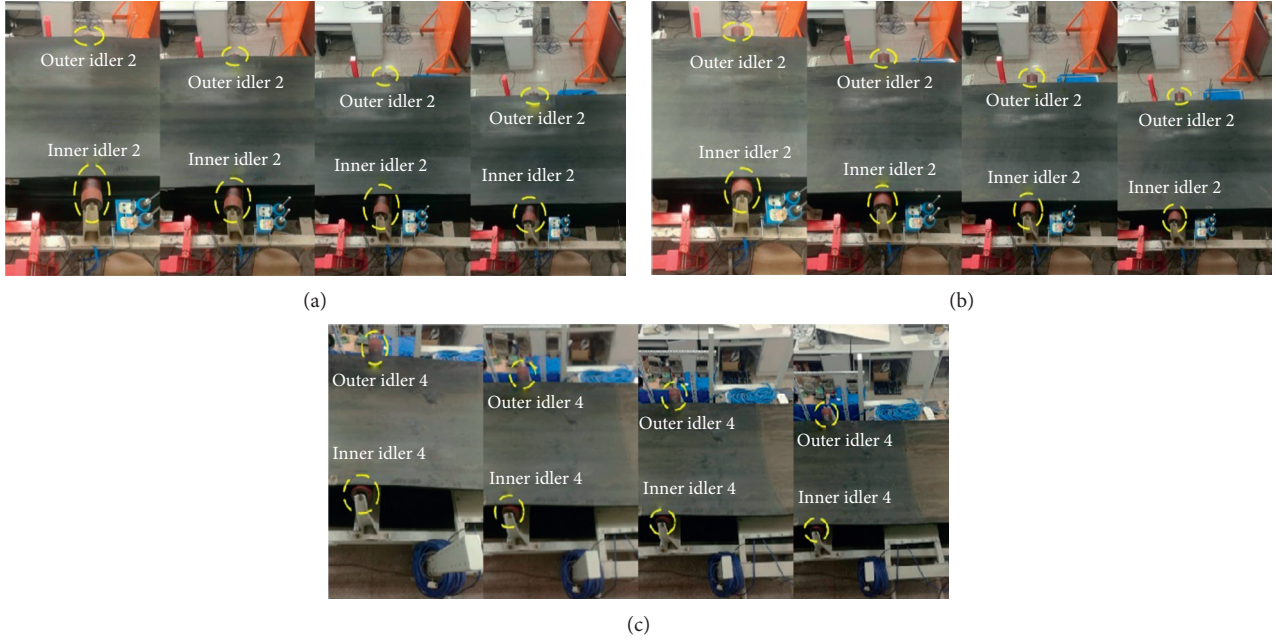


FIGURE 6: Conveyor belt images captured at different DDs and heights. (a) DD = +5%, shooting height from left to right: 2.4 m, 2.6 m, 2.8 m, and 3.0 m. (b) DD = 0%, shooting height from left to right: 2.4 m, 2.6 m, 2.8 m, and 3.0 m. (c) DD = -5%, shooting height from left to right: 2.4 m, 2.6 m, 2.8 m, and 3.0 m.

TABLE 2: Configuration of OM-SSD with different prediction source layers.

Layer model	L11	L13	L15	L17	L19	L21
OM-SSD-1	✓					
OM-SSD-2	✓	✓				
OM-SSD-3	✓	✓	✓			
OM-SSD-4	✓	✓	✓	✓		
OM-SSD-5	✓	✓	✓	✓	✓	
OM-SSD-6	✓	✓	✓	✓	✓	✓

removed. The training loss and the mAP of each OM-SSD- x ($x = 1, 2, \dots, 6$) and M-SSD are shown in Figures 7 and 8. The results show that the final training loss of each model is within 2.5, and the mAP is over 0.98, but the subtle difference may affect the detection accuracy. Among the models, the training loss of OM-SSD-2 is less than 2.0 in the late stage and finally converges to within 1.0, while that of other models is more than 1.0. The mAP of OM-SSD-2 is over 0.995 in the late stage and ends up with 0.9995, which is better than that of others, and this shows that the features used to detect the ROI are mainly derived from L11 and L13. The mAP curve of M-SSD shows a negative spike in the late stage, indicating that the model is not robust. Therefore, it can be concluded that the proposed OM-SSD-2 is more suitable for the detection of the ROI containing the conveyor belt edge and the exposed idler.

3.3. Results of the Deviation Detection

3.3.1. Accuracy of the ROI Detector. The well-trained OM-SSD-2 is used to extract ROI from the images in the validation dataset of the proposed deviation detection method. According to the shooting sequence, the images in each set

are numbered as 0, 1, 2, \dots , respectively. Due to the different shooting heights, the number of the images in each set varies from 11 to 20, and the higher the shooting height, the larger the number of the images.

The detection confidences of ROI_Is and ROI_Os are depicted in Figure 9. When the ROI is detected, the confidence varies in $[0.2, 1.0]$, and a higher confidence indicates a more accurate result. The confidence is 0 when the ROI is missed. From the perspective of the ROI, the number of the missed ROI_I and ROI_O is 7 and 19, respectively, that is, the missed detection rate of ROI_I (3.8%) is lower than that of ROI_O (10.9%), and the former shows a higher average confidence (77.6%) than the latter (44.9%). The reason is that more textures and details of the inner idler can be captured, and the features are more distinguishable. From the perspective of the shooting height, the missed detection rates at the four shooting heights are 5.7%, 6.0%, 15.3%, and 2.6%, respectively, and the lower ones are at 2.4 m and 3.0 m. When the shooting height is 2.4 m, all the ROI_Is are detected, and the confidences are close to 1.0, while the ROI_Os are missed 4 times. When the shooting height is

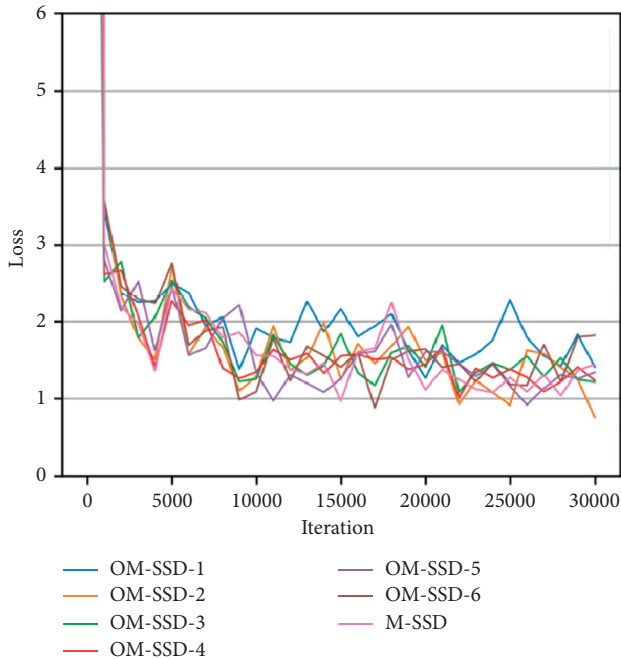


FIGURE 7: The training losses of different models.

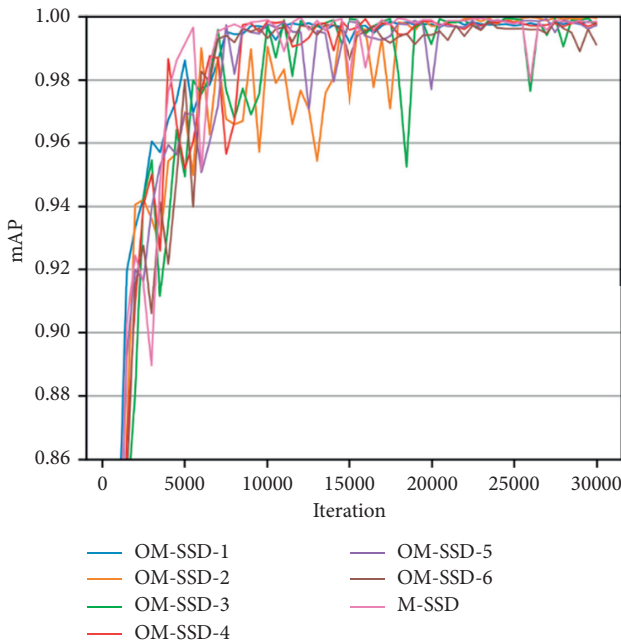


FIGURE 8: The mAPs of different models during training.

3.0 m, the number of the missed ROI_Is increases to 2, while that of ROI_Os decreases to 1. Part of the reason is that when the shooting height is lower, the camera is closer to the inner idler and more textures and details can be captured, and when the shooting height is higher, more content of the outer idler can be captured, while some details of the inner idler are lost. It can be concluded that, to achieve better ROI detection accuracy, the shooting height of 2.4 m and 3.0 m is preferred.

3.3.2. Accuracy of the DD Estimation. When ROIs are detected, the DD can be estimated by performing the conveyor belt edge detection, the idler outer edge detection, the geometric correction, and the DD estimation algorithm.

The empirical values of the prior transformation parameters for the idler outer edge detection are shown in Tables 3 and 4. The typical results of the conveyor belt edge detection and the idler outer edge detection are shown in Figure 10. It can be seen that the conveyor belt edges and the idler outer edges are detected correctly. Meanwhile, the conveyor belt edge points and the vertices of the idler outer edges are marked with circles.

The estimated DDs at each standard DD and shooting height are shown in Figure 11. It can be revealed that the fluctuation of the detection results increased along with the decrease of the standard DD from +5% to -5%. This can be explained by the fact that a larger size of ROI can lead to higher detection accuracy of the edges, and the image of the inner idler is bigger than that of the outer idler. As the standard DD decreases, so does the length of the inner exposed idler, and the detection accuracy degrades. Another reason is that when only one of ROI_O and ROI_I is detected, errors will be introduced by using the previous normal detection results. When a false conveyor belt edge or a false idler outer edge is detected, the estimated DD deviates from the ground truth. It can also be revealed that the image number affects the result of the DD estimation irregularly; that is to say, the horizontal shearing of the idler image is not the major factor that affects the detection accuracy, which verifies the validation of the proposed idler outer edge detection algorithm indirectly.

For the estimated DDs at each standard DD and shooting height, the mean and standard deviation (SD) are used to assess the accuracy and precision, as shown in Table 5. It shows that the estimated DDs at 2.4 m are closer to the ground truth than that at other shooting heights. When the standard DDs are +5% and -5%, the means of the estimated DDs are 5.04% and -5.03% at 2.4 m, which are the best in all shooting heights, and the SDs are within 0.20%. When the standard DD is 0%, the mean of the estimated DDs at each shooting height is within 0.3%. In addition, RMSE and MAE [32] are used to assess the average accuracy at each shooting height. As is shown in Table 5, the values of RMSE and MAE at 2.4 m are 0.37% and 0.44%, respectively, which are the smallest. Considering the width of the conveyor belt in the experiment is 1 m, the RMSE and MAE are 3.7 mm and 4.4 mm, respectively, which is much more accurate than the error of 16 mm in [12]. Meanwhile, 2.4 m is also the preferred shooting height for the ROI detection. Therefore, it can be concluded that 2.4 m is the preferred shooting height for the deviation detection of the belt conveyor.

3.3.3. Time Consumption. The time consumption of the proposed algorithm is shown in Table 6. The average time consumption of the deviation detection algorithm is 135.5 ms and is faster than that of 260 ms in [12]. The time

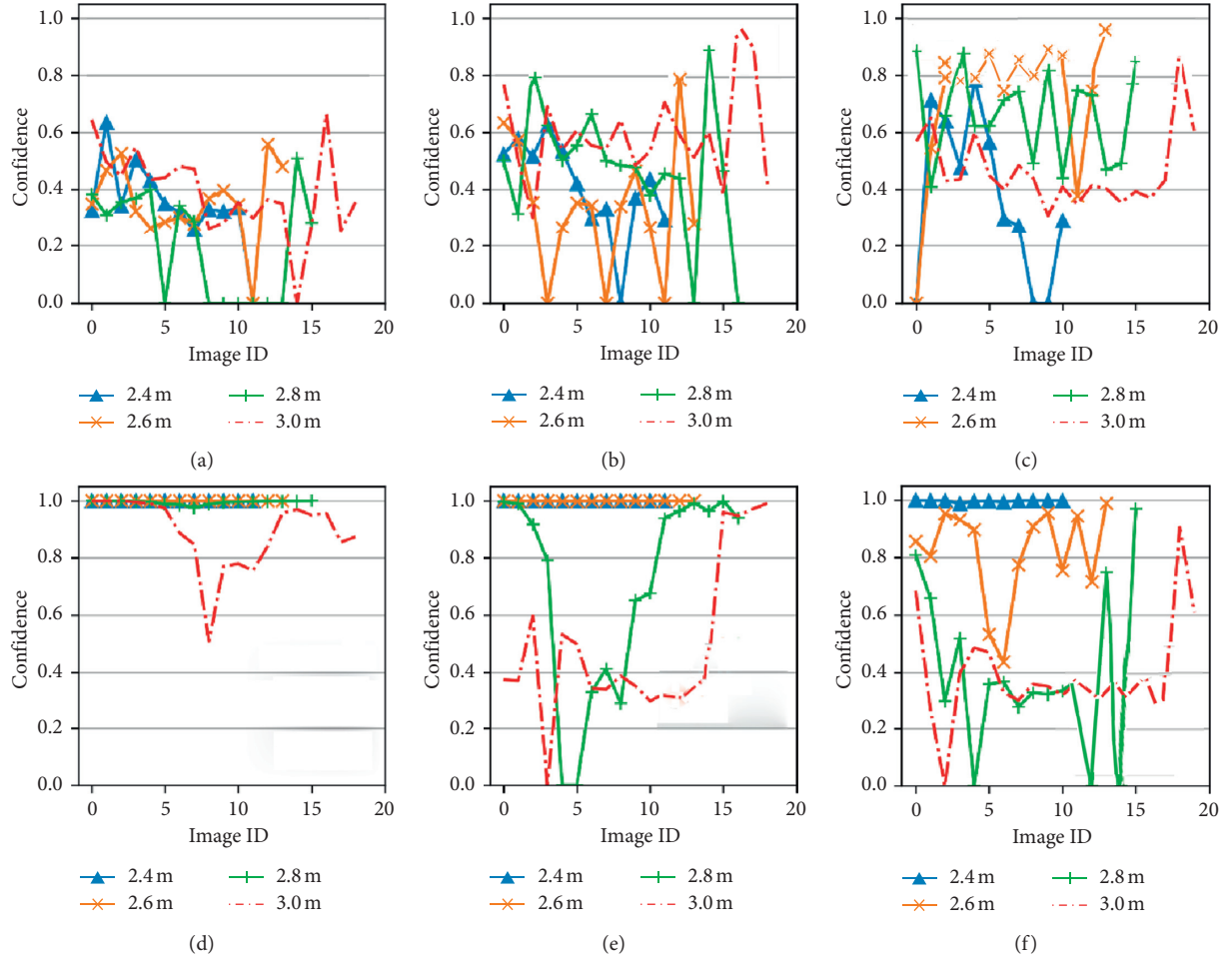


FIGURE 9: Detection confidences at different standard DDs and heights. (a) ROI_O, DD: +5%. (b) ROI_O, DD: 0%. (c) ROI_O, DD: -5%. (d) ROI_I, DD: +5%. (e) ROI_I, DD: 0%. (f) ROI_I, DD: -5%.

TABLE 3: The prior transformation parameters for the inner idler outer edge detection.

Height (m)	Parameter									
	R_{\min} (pixel)	R_{step} (pixel)	N_r	S_{\min}	S_{step}	N_s	C_l	C_r	C_{step}	N_c
2.4	65	1	8	0.6	0.05	4	-0.6	0.6	0.05	4
2.6	52	1	8	0.6	0.05	4	-0.6	0.6	0.05	4
2.8	47	1	8	0.5	0.05	4	-0.6	0.6	0.05	4
3.0	43	1	8	0.5	0.05	4	-0.6	0.6	0.05	4

TABLE 4: The prior transformation parameters for the outer idler outer edge detection.

Height (m)	Parameter									
	R_{\min} (pixel)	R_{step} (pixel)	N_r	S_{\min}	S_{step}	N_s	C_l	C_r	C_{step}	N_c
2.4	41	1	8	0.35	0.05	4	-0.5	0.5	0.05	4
2.6	37	1	8	0.25	0.05	4	-0.5	0.5	0.05	4
2.8	36	1	8	0.2	0.05	4	-0.5	0.5	0.05	4
3.0	34	1	8	0.2	0.05	4	-0.5	0.5	0.05	4

consumption is mainly composed of three components: the time consumption of OM-SSD-2, the conveyor belt edge detection, and the idler outer edge detection algorithm,

and the time consumption of the geometric correction and the DD estimation can be ignored. During each deviation detection process, OM-SSD-2 is called twice; if

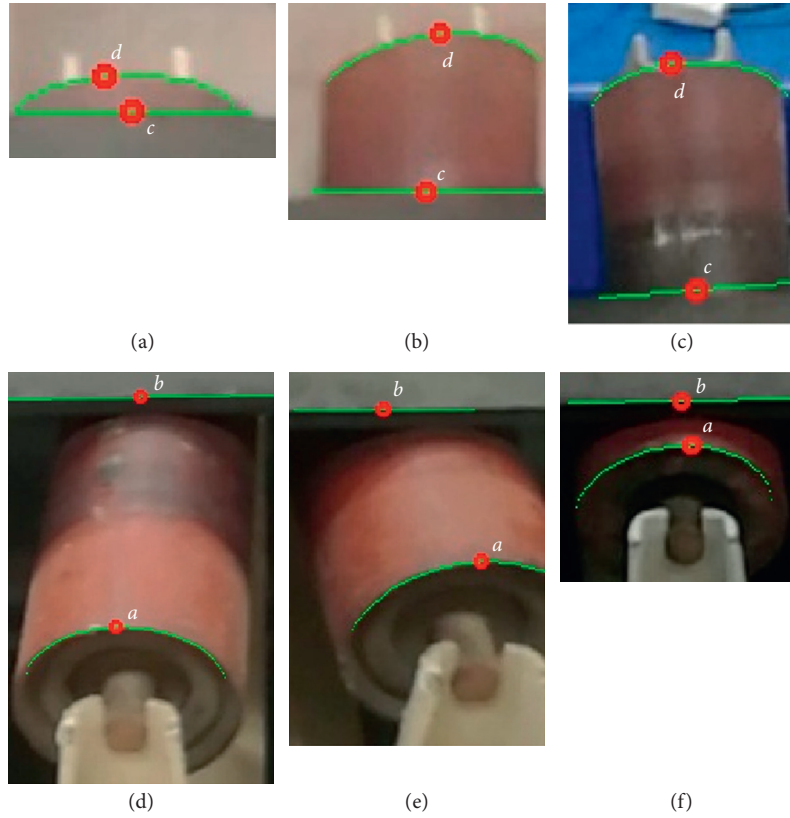


FIGURE 10: Detection results of the idlers, lines, and arcs. (a) Outer idler, DD: +5%. (b) Outer idler, DD: 0%. (c) Outer idler, DD: -5%. (d) Inner idler, DD: +5%. (e) Inner idler, DD: 0%. (f) Inner idler, DD: -5%.

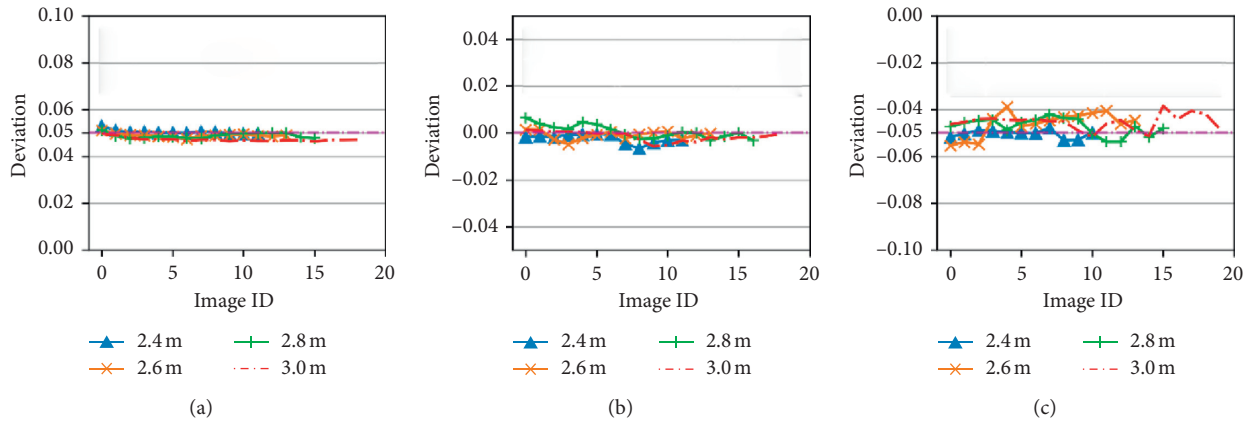


FIGURE 11: Estimated DDs at each standard DD and shooting height. (a) DD: +5%. (b) DD: 0%. (c) DD: -5%.

TABLE 5: Deviation detection results of the belt conveyor.

Height (m)	DD: +5%		DD: 0%		DD: -5%		RMSE (%)	MAE (%)
	Mean (%)	SD	Mean (%)	SD	Mean (%)	SD (%)		
2.4	5.04	0.08	-0.24	0.19	-5.03	0.16	0.37	0.44
2.6	4.90	0.07	-0.10	0.17	-4.60	0.50	0.69	0.87
2.8	4.89	0.10	-0.06	0.28	-4.72	0.35	0.56	0.76
3.0	4.72	0.08	-0.15	0.20	-4.54	0.32	0.67	0.93

TABLE 6: Time consumption of the proposed algorithm.

Algorithm	Time consumption (ms)
OM-SSD-2	41.6
Conveyor belt edge detection	4.7
Idler outer edge detection	23
Geometric correction and DD estimation	0.1
Belt conveyor deviation detection	135.5

one of the inner and outer idler is blocked by the conveyor belt and not detected, the remaining components will only be called once.

It is time consuming to perform the homography transformation on a high-resolution color image (i.e., $1080 \times 1920 \times 3$). In this paper, all the detection algorithms are performed on the original image, and the geometric correction and the DD estimation are implemented on only four points, and it improves the real-time performance significantly.

4. Conclusions

In this paper, a deviation detection method of the belt conveyor based on inspection robot and deep learning is proposed to detect the deviation at its any position. Firstly, the image of the belt conveyor is captured by the inspection robot continuously, and the ROI is extracted by the proposed ROI detector based on M-SSD. Secondly, the Hough line transform algorithm is used to detect the conveyor belt edge, and the elliptical arc detection algorithm based on template matching is proposed to detect the idler outer edge. Finally, a geometric correction algorithm based on homography transformation is proposed to correct the coordinates of the conveyor belt edge point and the idler outer edge point, based on which the DD of the conveyor belt is estimated. From the experimental results, the following concluding remarks can be drawn:

- (1) The proposed method can detect the deviation at any position along the belt conveyor, and with the lens down, it overcomes the drawback that the lens is susceptible to contamination in the existing machine vision-based method. In addition, this method provides a new intelligent solution for the monitoring of the belt conveyor.
- (2) The accuracy of the proposed OM-SSD-2 on the self-built dataset is 0.9995, and the conveyor belt edge detection method and the idler outer edge detection algorithm can detect the edges efficiently.
- (3) The experimental results show that the proposed method can detect the deviation of the conveyor belt continuously with an RMSE of 3.7 mm, an MAE of 4.4 mm, and average time consumption of 135.5 ms, which is better than that of the existing method, and the preferred shooting height is 2.4 m.

In order to further improve the performance of the deviation detection method of belt conveyor, the following research can be studied in the future:

- (1) The self-built dataset should be expanded to enhance the adaptability of the proposed method.
- (2) To improve the accuracy of the edge detection and object detection, complementary information should be fused into the image, and the thermal infrared information may be the best choice in mechanical fault detection.
- (3) Due to the powerful nonlinear mapping ability of deep learning and the convenient access to multi-spectral data, the deviation detection method based on an end-to-end deep learning framework and multispectral information may be the next research direction, and the CNN-based image registration, multispectral information fusion, object detection, and deviation estimator should be investigated, improved, and integrated.

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 regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (grant no. 51274150), the Key Projects of Science and Technology Support of Tianjin, China (grant no. 18YFZCGX00930), and the Key R&D Projects Supported by Achievements Transformation Relay of Tianjin, China (grant no. 18YFJLCG00060).


References

- [1] J. Li and C. Miao, "The conveyor belt longitudinal tear on-line detection based on improved SSR algorithm," *Optik*, vol. 127, no. 19, pp. 7395–8202, 2016.
- [2] L. Zhao and Y. Lin, "Typical failure analysis and processing of belt conveyor," *Procedia Engineering*, vol. 26, pp. 942–946, 2011.
- [3] Y. Yang, C. Miao, X. Li, and X. Mei, "On-line conveyor belts inspection based on machine vision," *Optik*, vol. 125, no. 19, pp. 5803–5807, 2014.
- [4] X. Mei, C. Miao, Y. Yang, and X. Li, "Rapid inspection technique for conveyor belt deviation," *Journal of Mechanical Engineering Research and Developments*, vol. 39, no. 3, pp. 653–662, 2016.

- [5] H. Su, C. Yang, G. Ferrigno, and E. De Momi, "Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
- [6] Q. Zhou, S. Zhao, H. Li, R. Lu, and C. Wu, "Adaptive neural network tracking control for robotic manipulators with dead zone," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3611–3620, 2019.
- [7] H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno, and E. De Momi, "Improved recurrent neural network-based manipulator control with remote center of motion constraints: experimental results," *Neural Networks*, vol. 131, pp. 291–299, 2020.
- [8] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: a review," *IEEE Access*, vol. 7, pp. 122644–122662, 2019.
- [9] W. Qi, H. Su, and A. Andrea, "A smartphone-based adaptive recognition and real-time monitoring system for human activities," *IEEE Transactions on Human-Machine Systems*, vol. 2020, 2020.
- [10] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2943–2949, 2020.
- [11] J. Li, Y. Luo, L. Shi et al., "Automatic fetal brain extraction from 2D in utero fetal MRI slices using deep neural network," *Neurocomputing*, vol. 378, pp. 335–349, 2020.
- [12] Y. Liu, Y. Wang, C. Zeng, W. Zhang, and J. Li, "Edge detection for conveyor belt based on the deep convolutional network," 2018.
- [13] A. B. Alhassan, X. Zhang, H. Shen, and H. Xu, "Power transmission line inspection robots: a review, trends and challenges for future research," *International Journal of Electrical Power & Energy Systems*, vol. 118, 2020.
- [14] S.-N. Yu, J.-H. Jang, and C.-S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," *Automation in Construction*, vol. 16, no. 3, pp. 255–261, 2007.
- [15] W. Yang, X. Zhang, and H. Ma, "An inspection robot using infrared thermography for belt conveyor," 2016.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.
- [17] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, London, UK, 2015.
- [18] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," *Computer Vision-ECCV 2016*, vol. 9905, pp. 21–37, 2016.
- [19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," 2016.
- [20] V. Osin, A. Cichocki, and E. Burnaev, "Fast multispectral deep fusion networks," *Bulletin of the Polish Academy of Sciences-Technical Sciences*, vol. 66, no. 6, pp. 875–889, 2018.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
- [23] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, London, UK, 2015.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
- [25] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," 2015.
- [26] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," 2018.
- [27] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017.
- [28] H. Li, X. Zhang, and Z. Chen, "Automatic corner detection of checkerboard based on LSD," *Optics and Precision Engineering*, vol. 23, no. 12, pp. 3480–3489, 2015.
- [29] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [30] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
- [31] J. Cai, P. Huang, and B. Zhang, "Fast circle detector based on region-growing of gradient and histogram of euclidean distance," *Acta Optica Sinica*, vol. 35, no. 3, pp. 193–202, 2015.
- [32] Y. Gao, T. Qiao, H. Zhang, Y. Yang, Y. Pang, and H. Wei, "A contactless measuring speed system of belt conveyor based on machine vision and machine learning," *Measurement*, vol. 139, pp. 127–133, 2019.

Review Article

Dynamic Characteristics of Microring Driven by the Symmetrically Distributed Electrostatic Force

Qingheng Meng,¹ Yuanlin Zhang,¹ Jin Wei,¹ Yuh-Chung Hu,² Yan Shi,³ and Tao Yu ¹

¹School of Mechatronics and Automobile Engineering, Yantai University, Yantai 264005, China

²Department of Mechanical and Electromechanical Engineering, National Ilan University, Yilan, Yilan County 260, Taiwan

³School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

Correspondence should be addressed to Tao Yu; yutao@ytu.edu.cn

Received 15 July 2020; Revised 12 October 2020; Accepted 21 January 2021; Published 4 February 2021

Academic Editor: Juan Sandoval

Copyright © 2021 Qingheng Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims at investigating the dynamic characteristics of a microring driven by dual arch electrodes because they are basic elements of microelectrostatic motors. The dual arch electrodes surround the periphery of the microring and are arranged symmetrically to the center of the ring. The electrodes are fixed while the microring is flexible. The electrostatic force will deform the microring, while the deflection of the microring changes the gap between the microring and the electrodes, thereby changing the electrostatic force. Therefore, this is an electromechanical coupling effect. The nonlinear partial-differential equation that governs the motion of the microring is derived based on thin shell theory. Then, based on the assumption of small deflection, the nonlinear governing equation is linearized by truncating the higher-order terms of the Taylor series expansion of the nonlinear electrostatic force. After that, the linearized governing equation is discretized into a set of ordinary differential equations using Galerkin method in which the mode shape functions of the ring are adopted. The influences of the structural damping of the microring and the span of the arch electrodes on the forced response and dynamical stabilities of the microring are investigated. The results show that the damping ratio has a great influence on the system instability during high-frequency excitation. The unstable region of the system can increase with the increase of the electrode span; the response amplitude can also be increased within a certain range.

1. Introduction

In microelectromechanical systems (MEMS), electrostatic principle is widely used in microsensors/actuators because of its advantages of easy control, fast response, low power consumption, and easy integration with IC processes [1]. Electrostatic forces are mainly used in gyroscopes [2], electrostatic motors [3], torsion micromirrors [4], accelerometers [5], and microswitches [6]. Their key microstructures include beams, plates, and rings. In the design of electrostatic driven microstructures, the energy domain coupling of electrical and mechanical energies is a very important issue. Vibration analysis of microstructures based on different composite materials is the focus of current research [7–10]. Vibration analysis of microstructures is

different from macrostructures because the force proportional to the area plays a leading role [11]. The interaction of energy domains makes the analysis of MEMS complicated. Electrostatically driven microstructures will bring many nonlinear characteristics, such as dynamic instability, jumping, bifurcation, and chaos [12, 13]. Stability is the basic condition for the system to work. An unstable system does not have the ability to adjust and cannot work normally [14, 15].

Since the 21st century, the research contents of electrostatic force in MEMS mainly include pull-in voltage, damping, stability, and amplitude frequency-response. For electrostatic driven devices, the driving voltage has a critical value, and the electrostatic force exhibits a strong nonlinearity at this value. When the driving voltage exceeds

this critical value, the equilibrium between the elastic restoring force and the electrostatic force is broken, the microstructure and electrode are attracted together, and the pull-in phenomenon occurs. The voltage limit for the pull-in phenomenon is the pull-in voltage, and the corresponding position is called the pull-in position [16, 17]. Farokhi et al. [18] studied the pull-in characteristics of electrostatically driven microarches and the effect of system parameters on the pull-in instability. The electrostatic switch needs to determine the pull-in voltage to prevent excessive voltage from causing adhesion failure of the structure [19]. The driving principle of the micromotor includes electrostatic type, electromagnetic type, and piezoelectric type [20–22]. In MEMS, since the existence of microscale effects, damping has a great impact on the function of microstructures, so damping cannot be ignored. Bao and Yang [23] surveyed kinds of literature on various damping models of MEMS microstructures and introduced related experiments. Belardinelli et al. [24] established a microbeam mechanical model considering both pressure film damping and thermoelastic damping and analyzed the effects of the two types of damping on the natural frequency of the beam. In the past, the research objects of electrostatically driven microstructures were mainly microplates and microbeams. Recently, researchers have begun to focus on other microstructures, such as the key components in gyroscopes and electrostatic motors: rings [25–28]. In the last century, researchers have established different theoretical models and systematically studied the inherent characteristics of the ring and the forced response under harmonic excitation [29]. Xu and Qin [30] theoretically studied the jump up and down phenomenon of electrostatically driven microring near the natural frequency and explored the effect of excitation voltage on the critical frequency of the jump phenomenon.

Dynamic pull-in instability needs to consider factors such as damping, AC voltage excitation, and inertial effects caused by sudden changes in DC voltage, which all affect the dynamic pull-in process [26]. In addition to studying the dynamic pull-in instability of the system, the stability of the system also needs to be considered. Hu et al. [31] studied the stability of electrostatic driven microbeam system theoretically. The authors of [32, 33] studied the stability and forced response of the microring system under the traveling electrostatic force, which is generated by an electrode. Yu et al. [34] established a theoretical model of a rotating ring under a uniform electric field to analyze the inherent characteristics and dynamic stability of the system. In the above study of the stationary ring, the electric field is generated by a constant voltage, without considering the effect of dynamic voltage. The microring is driven by pairs of electrodes. In the study of electrostatically driven microring, damping is mostly ignored, but damping has a great influence on the response and stability of the system. Therefore, in this paper, the physical model of a pair of electrostatically driven microrings under alternating voltage control is established, and the effect of damping on the system is considered.

The magnitude of the electrostatic force is proportional to the square of the voltage and inversely proportional to the square of the gap between the ring and the electrode. Based on the theory of thin shells and the electrostatic force model, a dynamic model of electrostatically driven microring is established [29, 35]. The electrostatic force is controlled by a pair of fixed electrodes applying an AC voltage to control the deformation of the microring. This model takes into account the structural deformation coupling. Based on the small deflection assumption, the electrostatic force can be expanded using Taylor series to obtain a linear periodic time-varying system equation. Discrete partial differential equations use Galerkin method to obtain ordinary differential matrix equations. First, the system performs modal analysis, then Floquet theory is used to judge the stability of the system, and the forced response of the system is obtained by the RungeKutta numerical integration method. This paper studies the stability and the amplitude-frequency response of the microring, which provides a reference for the design of the distributed electrostatic force-driven microring.

2. Model

The main components of an electrostatic motor consist of a ring-shaped rotor and several pairs of arch-shaped driving electrodes. The driving electrodes surround the periphery of the ring at equal intervals and symmetrically to the center of the ring. Each pair of the driving electrode consists of two driving electrodes symmetrical to the center of the ring. When the motor is working, the voltage is sequentially applied to each pair of the driving electrode, and the ring is elastically deformed at the corresponding position. The periodic voltage generates a periodic electric field, which generates a periodic electrostatic force. The periodic deformation caused by the electrostatic force changes the capacitance between the electrode and the ring, thereby driving the ring to move [30].

Based on the aforesaid working principle, it is worthwhile to investigate the dynamic characteristics of the ring under the action of a pair of electrodes. As shown in Figure 1, a dynamic voltage is applied between the microring and the electrode, which generates a time-varying electric field to drive the ring. The potential difference \hat{V} is the control voltage, which is set to $\hat{V} = \hat{V}_D + \hat{V}_A \sin(\hat{\Omega}\hat{t})$, where \hat{V}_D , \hat{V}_A , $\hat{\Omega}$, and \hat{t} are DC voltage, AC voltage, AC angular frequency, and time, respectively. The average radius and thickness of the ring are r and h , the span of the two electrodes is ϕ , and the gap between the ring and the electrode is g . There is a fixed coordinate system X - Y on the ring, and the radial displacement caused by electrostatic force at the angular coordinate θ is $\hat{u}(\theta, \hat{t})$.

2.1. Equation of Motion. According to the principle of parallel capacitor plates, the distributed force shown in Figure 1 can be expressed as

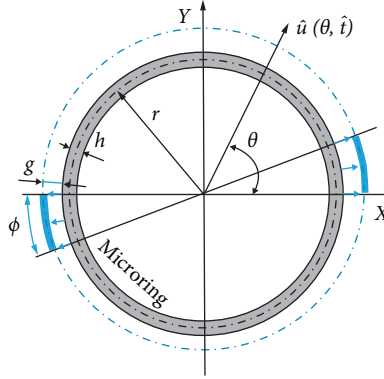


FIGURE 1: Schematic diagram of electrostatically driven microring.

$$[H(\theta) - H(\theta - \phi) + H(\theta - \pi) - H(\theta - \pi - \phi)] \frac{\varepsilon b \hat{V}^2}{2(g - \hat{u})^2}, \quad (1)$$

where ε is the permittivity of the medium between the ring and electrode, b is the width of the ring, and H represents the unit step function.

The microring of this system is mainly subjected to radial forces, so its circumferential inertial force is negligible. According to the thin shell theory [29], the dynamic equation of the ring subjected to the electrostatic force of Figure 1 is

$$\frac{EI}{r^4} \left(\frac{\partial^4 \hat{u}}{\partial \theta^4} + 2 \frac{\partial^2 \hat{u}}{\partial \theta^2} + \hat{u} \right) + \rho A \frac{\partial^2 \hat{u}}{\partial t^2} = [H(\theta) - H(\theta - \phi) + H(\theta - \pi) - H(\theta - \pi - \phi)] \frac{\varepsilon b \hat{V}^2}{2(g - \hat{u})^2}. \quad (2)$$

For easier essay writing, $H(\theta) - H(\theta - \phi) + H(\theta - \pi) - H(\theta - \pi - \phi)$ is set to $f(\theta, \phi)$. The authors introduce dimensionless variables as

$$u = \hat{u}/g, t = \hat{t}/T, \Omega = \hat{\Omega}T, V_D = \hat{V}_D \sqrt{\frac{r^3 \varepsilon b}{4\phi EI g^3}}, \quad (3)$$

$$V_A = \hat{V}_A \sqrt{\frac{r^3 \varepsilon b}{4\phi EI g^3}}, V = V_D + V_A \sin(\Omega t),$$

where $T = \sqrt{\rho A r^4 / EI}$. Using the dimensionless variables in Equation (3), the dimensionless motion equation is expressed as

$$\frac{\partial^4 u}{\partial \theta^4} + 2 \frac{\partial^2 u}{\partial \theta^2} + u + \frac{\partial^2 u}{\partial t^2} = f(\theta, \phi) \frac{V^2}{(1 - u)^2}. \quad (4)$$

Based on the assumption of small deformation, the nonlinear part of the electrostatic force term is expanded using the Taylor series concerning the initial equilibrium position ($u=0$), ignoring the higher-order terms and retaining the linear terms, that is

$$\frac{1}{(1 - u)^2} \approx 1 + 2u. \quad (5)$$

Finally, a linear time-varying motion equation is obtained as

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial^4 u}{\partial \theta^4} + 2 \frac{\partial^2 u}{\partial \theta^2} + (1 - 2f(\theta, \phi)V^2)u = f(\theta, \phi)V^2. \quad (6)$$

Due to the alternating voltage on the left side of the equation of motion (6), the microring system becomes a time-varying system, and the driving voltage V can change the dynamic characteristics of the microring.

2.2. Discretization. For the geometrical periodicity of a circular ring, the deflection function $u(\theta, t)$ can be represented by

$$u(\theta, t) = \sum_{k=2}^n [\alpha_k(t) \cos k\theta + \beta_k(t) \sin k\theta], \quad (7)$$

where k is the circumferential wave number and $\alpha_k(t)$ and $\beta_k(t)$ are generalized coordinate functions about time t . The term $k=1$ is a rigid body mode, so it can be eliminated [29, 34], such that $k=2, 3 \dots n$ form a complete basis space. Substituting equation (7) in equation (6) gives

$$\begin{aligned}
& \sum_{k=2}^n \left[\ddot{\alpha}_k(t) + k^4 \alpha_k(t) - 2k^2 \alpha_k(t) + [1 - 2f(\theta, \phi)V^2] \alpha_k(t) \right] \cos k\theta \\
& + \sum_{k=2}^n \left[\ddot{\beta}_k(t) + k^4 \alpha_k(t) - 2k^2 \beta_k(t) + [1 - 2f(\theta, \phi)V^2] \beta_k(t) \right] \sin k\theta \\
& = f(\theta, \phi)V^2.
\end{aligned} \tag{8}$$

Since the orthogonality of trigonometric functions, one can discretize the equation (8) by multiplying equation (8)

with $\cos(m\theta)$ and $\sin(m\theta)$, where $m = 2, 3, 4, \dots$, which gets the following two equations:

$$\begin{aligned}
& \sum_{k=2}^n \left[\ddot{\alpha}_k(t) + k^4 \alpha_k(t) - 2k^2 \alpha_k(t) + [1 - 2f(\theta, \phi)V^2] \alpha_k(t) \right] \cos k\theta \cos(m\theta) \\
& + \sum_{k=2}^n \left[\ddot{\beta}_k(t) + k^4 \alpha_k(t) - 2k^2 \beta_k(t) + [1 - 2f(\theta, \phi)V^2] \beta_k(t) \right] \sin k\theta \cos(m\theta) \\
& = f(\theta, \phi)V^2 \cos(m\theta),
\end{aligned} \tag{9}$$

$$\begin{aligned}
& \sum_{k=2}^n \left[\ddot{\alpha}_k(t) + k^4 \alpha_k(t) - 2k^2 \alpha_k(t) + [1 - 2f(\theta, \phi)V^2] \alpha_k(t) \right] \cos k\theta \sin(m\theta) \\
& + \sum_{k=2}^n \left[\ddot{\beta}_k(t) + k^4 \alpha_k(t) - 2k^2 \beta_k(t) + [1 - 2f(\theta, \phi)V^2] \beta_k(t) \right] \sin k\theta \sin(m\theta) \\
& = f(\theta, \phi)V^2 \sin(m\theta).
\end{aligned} \tag{10}$$

Integrating equations (9) and (10) over the microring's circumference gives

$$\begin{aligned}
& \sum_{k=2}^n \left[\ddot{\alpha}_k(t) + (k^2 - 1)^2 \alpha_k(t) \right] \int_0^{2\pi} \cos k\theta \cos(m\theta) d\theta - \sum_{k=2}^n 2V^2 \alpha_k(t) \int_0^{2\pi} f(\theta, \phi) \cos k\theta \cos(m\theta) d\theta \\
& + \sum_{k=2}^n \left[\ddot{\beta}_k(t) + (k^2 - 1)^2 \beta_k(t) \right] \int_0^{2\pi} \sin k\theta \cos(m\theta) d\theta - \sum_{k=2}^n 2V^2 \beta_k(t) \int_0^{2\pi} f(\theta, \phi) \sin k\theta \cos(m\theta) d\theta \\
& = V^2 \int_0^{2\pi} f(\theta, \phi) \cos(m\theta) d\theta,
\end{aligned} \tag{11}$$

$$\begin{aligned}
& \sum_{k=2}^n \left[\ddot{\alpha}_k(t) + (k^2 - 1)^2 \alpha_k(t) \right] \int_0^{2\pi} \cos k\theta \sin(m\theta) d\theta - \sum_{k=2}^n 2V^2 \alpha_k(t) \int_0^{2\pi} f(\theta, \phi) \cos k\theta \sin(m\theta) d\theta \\
& + \sum_{k=2}^n \left[\ddot{\beta}_k(t) + (k^2 - 1)^2 \beta_k(t) \right] \int_0^{2\pi} \sin k\theta \sin(m\theta) d\theta - \sum_{k=2}^n 2V^2 \beta_k(t) \int_0^{2\pi} f(\theta, \phi) \sin k\theta \sin(m\theta) d\theta \\
& = V^2 \int_0^{2\pi} f(\theta, \phi) \sin(m\theta) d\theta.
\end{aligned} \tag{12}$$

Due to the orthogonality of trigonometric functions, Equations (11) and (12) can be simplified and expressed in the following matrix expression:

$$\{\ddot{X}\}_{2(n-1) \times 1} + [\mathbf{K}]_{2(n-1) \times 2(n-1)} \{\mathbf{X}\}_{2(n-1) \times 1} = \{\mathbf{Q}\}_{2(n-1) \times 1}, \quad (13)$$

where $\{\mathbf{X}\}$, $[\mathbf{K}]$, and $\{\mathbf{Q}\}$ are generalized coordinates vector, stiffness matrix, and generalized force vector, respectively. They are specifically expressed as

$$\{\mathbf{X}\}^T = \{\alpha_2(t) \ \alpha_3(t) \ \alpha_4(t) \ \cdots \ \alpha_n(t) \ \beta_2(t) \ \beta_3(t) \ \beta_4(t) \ \cdots \ \beta_n(t)\}, \quad (14)$$

$$[\mathbf{K}] = [\mathbf{K}^k]_{2(n-1) \times 2(n-1)} + [\mathbf{K}^e]_{2(n-1) \times 2(n-1)}, \quad (15)$$

where the stiffness matrix $[\mathbf{K}]$ consists of $[\mathbf{K}^k]$ and $[\mathbf{K}^e]$. $[\mathbf{K}^k]$ is structural stiffness-matrix, which is a diagonal matrix. $[\mathbf{K}^e]$ is electrostatic stiffness-matrix due to the

dynamic voltage. The elements in the stiffness matrix are expressed as

$$\begin{cases} K_{ij}^k = (i^2 - 1)^2, & (i = j), \\ K_{ij}^k = 0, & (i \neq j), \end{cases} \quad (16)$$

$$[\mathbf{K}^e] = \begin{bmatrix} [\mathbf{K}^a]_{(n-1) \times (n-1)} & [\mathbf{K}^b]_{(n-1) \times (n-1)} \\ [\mathbf{K}^c]_{(n-1) \times (n-1)} & [\mathbf{K}^d]_{(n-1) \times (n-1)} \end{bmatrix}, \quad (17)$$

$$K_{ij}^a = \begin{cases} -\frac{V^2}{\pi} \left[\frac{\sin(i+j)\phi}{i+j} (1 + (-1)^{i+j}) + \frac{\sin(i-j)\phi}{i-j} (1 + (-1)^{i-j}) \right], \\ -\frac{2V^2}{\pi} \left[\phi + \frac{\sin(2i\phi)}{4i} + \frac{\sin(2i(\pi + \phi))}{4i} \right], & i = j, \end{cases} \quad (18)$$

$$K_{ij}^b = \begin{cases} \frac{V^2}{\pi} \left[\frac{\cos(i+i)\phi}{i+j} (1 + (-1)^{i+j}) + \frac{\cos(i-j)\phi}{i-j} (1 + (-1)^{i-j}) - \frac{2i}{i^2 - j^2} - \left[\frac{(-1)^{i+j}}{i+j} + \frac{(-1)^{i-j}}{i-j} \right] \right], \\ -\frac{2V^2}{\pi} \left[\frac{(\sin i\phi)^2}{2i} + \frac{(\sin i(\pi + \phi))^2}{2i} \right], & i = j, \end{cases} \quad (19)$$

$$K_{ij}^c = \begin{cases} \frac{V^2}{\pi} \left[\frac{\cos(i+j)\phi}{i+j} (1 + (-1)^{i+j}) - \frac{\cos(i-j)\phi}{i-j} (1 + (-1)^{i-j}) + \frac{2j}{i^2 - j^2} - \left[\frac{(-1)^{i+j}}{i+j} - \frac{(-1)^{i-j}}{i-j} \right] \right], \\ -\frac{2V^2}{\pi} \left[\frac{(\sin k\phi)^2}{2k} + \frac{(\sin k(\pi + \phi))^2}{2k} \right], & i = j. \end{cases} \quad (20)$$

$$K_{ij}^d = \begin{cases} \frac{V^2}{\pi} \left[\frac{\sin(i+j)\phi}{i+j} (1 + (-1)^{i+j}) - \frac{\sin(i-j)\phi}{i-j} (1 + (-1)^{i-j}) \right], \\ -\frac{2V^2}{\pi} \left[\phi - \frac{\sin 2k\phi}{4k} - \frac{\sin 2k(\pi + \phi)}{4k} \right], & i = j, \end{cases} \quad (21)$$

The generalized force vector is

$$\{\mathbf{Q}\} = \begin{cases} \{\mathbf{Q}^\alpha\}_{(n-1) \times 1} \\ \{\mathbf{Q}^\beta\}_{(n-1) \times 1} \end{cases},$$

$$\begin{cases} Q_i^\alpha = \frac{V^2}{\pi} \left(\frac{\sin(i\phi)}{i} + \frac{\sin(i(\pi + \phi))}{i} \right), \\ Q_i^\beta = \frac{V^2}{\pi} \left(-\frac{\cos(i\phi)}{i} + \frac{1}{i} - \frac{\cos(i(\pi + \phi))}{i} + \frac{\cos(i\pi)}{i} \right). \end{cases} \quad (22)$$

2.3. *Damping.* In this microring system, there is more or less a certain amount of damping, and it is generally obtained from experiments. The effect of damping on the forced

$$[\mathbf{C}] = \begin{bmatrix} [\mathbf{C}_1] & \\ & [\mathbf{C}_2] \end{bmatrix}, [\mathbf{C}_1] = [\mathbf{C}_2] = \begin{bmatrix} 2\xi\omega_1 & & \\ & \ddots & \\ & & 2\xi\omega_{n-1} \end{bmatrix}_{(n-1) \times (n-1)}, \quad (23)$$

where ξ is the damping ratio and ω_n is the natural frequencies of the ring. The damping matrix is a principal diagonal matrix. Finally, when the system has damping, the equation of motion expressed by the matrix is

$$\begin{aligned} \{\ddot{\mathbf{X}}\}_{2(n-1) \times 1} + [\mathbf{C}]_{2(n-1) \times 2(n-1)} \{\dot{\mathbf{X}}\}_{2(n-1) \times 1} \\ + [\mathbf{K}]_{2(n-1) \times 2(n-1)} \{\mathbf{X}\}_{2(n-1) \times 1} = \{\mathbf{Q}\}_{2(n-1) \times 1}. \end{aligned} \quad (24)$$

3. Natural Frequencies and Modes

In this section, the mode of the microring is derived. Table 1 shows the material parameters and geometric parameters of the microring.

Modal analysis is to find the inherent characteristics of the system when there is no external force, such as natural frequency and mode shape. When the system has no electrostatic force and the damping is neglected, equation (24) can be simplified to

$$\begin{Bmatrix} \ddot{\alpha}_k(t) \\ \ddot{\beta}_k(t) \end{Bmatrix} + \begin{bmatrix} (k^2 - 1)^2 & 0 \\ 0 & (k^2 - 1)^2 \end{bmatrix} \begin{Bmatrix} \alpha_k(t) \\ \beta_k(t) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}, \quad (25)$$

where $k = 2, 3, 4, \dots$. The generalized coordinate functions $\alpha_k(t)$ and $\beta_k(t)$ are usually set to

$$\begin{Bmatrix} \alpha_k(t) \\ \beta_k(t) \end{Bmatrix} = \begin{Bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \end{Bmatrix} e^{i\omega_k t}, \quad (26)$$

where $\bar{\alpha}_k$ and $\bar{\beta}_k$ are the amplitudes of $\alpha_k(t)$ and $\beta_k(t)$, respectively. Inserting equation (26) into equation (25) yields the following matrix expression:

TABLE 1: Material parameters and geometric parameters.

Properties	value
Density	$\rho = 2.7 \times 10^3 \text{ Kg/m}^3$
Young's modulus	$E = 6.8 \times 10^{10} \text{ Pa}$
Mean radius	$r = 0.02 \text{ m}$
Radial thickness	$h = 3 \times 10^{-4} \text{ m}$
Gap	$g = 6 \times 10^{-5} \text{ m}$
Width	$b = 5 \times 10^{-3} \text{ m}$

response is to eliminate the transient response and eventually reach the steady state. In practical applications, the steady-state response is the focus of research. Adding equivalent damping in the radial direction of the ring is to add the damping term $\lambda \dot{u}$ to equation (4). According to previous studies, the damping matrix can be set to [29, 36]

$$\begin{bmatrix} (k^2 - 1)^2 & 0 \\ 0 & (k^2 - 1)^2 \end{bmatrix} \begin{Bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \end{Bmatrix} = \omega_k^2 \begin{Bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \end{Bmatrix}. \quad (27)$$

Generally, the amplitudes $\bar{\alpha}_k$ and $\bar{\beta}_k$ are not zero, and the natural frequency can be obtained, namely, $\omega_{k-1}^2 = (k^2 - 1)^2$, ($k = 2, 3, 4, \dots$). The corresponding natural modes of the microring are $\cos k\theta + \sin k\theta$, ($k = 2, 3, 4, \dots$). Figure 2 shows the first three natural modes of the microring.

4. Dynamic Analysis

4.1. *Forced Response.* The differential equations (24) are numerically solved using the Runge-Kutta method. It is necessary to reduce the order of the second-order differential equations and transform them into the state space. The space state vector is defined as

$$\{\mathbf{Y}\}_{4(n-1) \times 1} = \begin{Bmatrix} \{\mathbf{X}\} \\ \{\dot{\mathbf{X}}\} \end{Bmatrix}. \quad (28)$$

Equation (24) is transformed into the state space and expressed as

$$\{\dot{\mathbf{Y}}\} = [\mathbf{A}]\{\mathbf{Y}\} + [\mathbf{B}]\{\mathbf{F}\}, \quad (29)$$

where

$$[\mathbf{A}]_{4(n-1) \times 4(n-1)} = \begin{bmatrix} [0]_{2(n-1) \times 2(n-1)} & [\mathbf{I}]_{2(n-1) \times 2(n-1)} \\ -[\mathbf{K}]_{2(n-1) \times 2(n-1)} & -[\mathbf{C}]_{2(n-1) \times 2(n-1)} \end{bmatrix}, \quad (30)$$

$$[\mathbf{B}]_{4(n-1) \times 4(n-1)} = \begin{bmatrix} [0]_{2(n-1) \times 2(n-1)} & [0]_{2(n-1) \times 2(n-1)} \\ [0]_{2(n-1) \times 2(n-1)} & [\mathbf{I}]_{2(n-1) \times 2(n-1)} \end{bmatrix}, \quad (31)$$

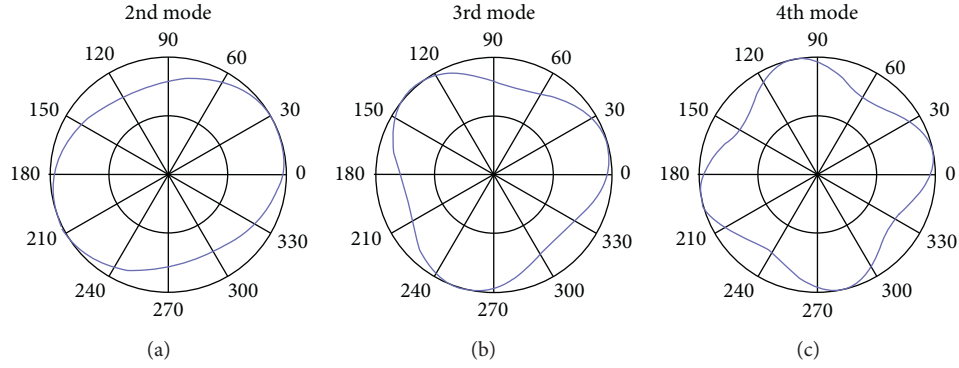


FIGURE 2: The first three natural modes.

$$\{\mathbf{F}\}_{4(n-1) \times 1} = \begin{Bmatrix} \{\mathbf{Q}\}_{2(n-1) \times 1} \\ \{\mathbf{Q}\}_{2(n-1) \times 1} \end{Bmatrix}, \quad (32)$$

where $[0]$ and $[\mathbf{I}]$ are the zero matrix and the unit matrix, respectively. Since the $[\mathbf{A}]$ matrix contains $[\mathbf{K}]$, the matrix is periodically time-varying.

4.2. Stability. According to the static stiffness terms (17)–(21), the stiffness term of this system is related to the voltage, and the voltage changes periodically with time, so the system is a periodic time-varying system. The stability of

periodic time-varying systems is an important subject. According to the characteristics of the system, Floquet theory is used to analyze its stability.

To analyze the stability of the system, the homogeneous part of equation (29) needs to be considered, that is,

$$\{\dot{\mathbf{Y}}\} = [\mathbf{A}]\{\mathbf{Y}\}. \quad (33)$$

The parameter matrix $[\mathbf{A}]$ is a periodically time-varying matrix with the period of $\hat{T} = 2\pi/(2\Omega)$, and set $4(n-1)$ linearly independent initial conditions:

$$\{\mathbf{Y}(0)\}_1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}, \{\mathbf{Y}(0)\}_2 = \begin{Bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}, \dots, \{\mathbf{Y}(0)\}_{4(n-1)} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{Bmatrix}. \quad (34)$$

The solution of equation (33) in a period \hat{T} is solved by the RungeKutta numerical method, namely, $\{\mathbf{Y}(\hat{T})\}_1, \{\mathbf{Y}(\hat{T})\}_2, \dots, \{\mathbf{Y}(\hat{T})\}_{4(n-1)}$, and the solutions of these homogeneous equations are formed into a matrix $[\mathbf{D}]$:

$$[\mathbf{D}] = \begin{bmatrix} \{\mathbf{Y}(\hat{T})\}_1 & \{\mathbf{Y}(\hat{T})\}_2 & \dots & \{\mathbf{Y}(\hat{T})\}_{4(n-1)} \end{bmatrix}. \quad (35)$$

The stability of the system can be determined by the eigenvalues (μ_i) of matrix $[\mathbf{D}]$ [37]. The system is stable if all the eigenvalues have magnitudes less than unity, i.e., $|\mu_i| < 1$, unstable if at least one eigenvalue greater than unity, i.e., $|\mu_i| > 1$, and marginally stable if at least one eigenvalue with unit magnitude and multiplicity less than unity.

5. Results and Discussion

To investigate the dynamic characteristics of the electrostatically driven microring system, the stability of the system is first analyzed, and then the amplitude-frequency response curve of the system is solved. The following stability analysis and forced response analysis are developed using the first

three modes. Considering that the microring is driven by three pairs of electrodes, the maximum electrode span is $\pi/3$, so the electrode span range is $[0, \pi/3]$. The authors selected the angular coordinate $\theta = 0$ for dynamic characteristics analysis.

5.1. Unstable Region. Damping is a factor that cannot be ignored in the system. Here, we first consider the impact of damping on system stability. Using the stability analysis method in Section 4.2, the unstable region of the system under different damping is studied.

Figure 3 shows the unstable region of the system when the AC voltage $V_d = 0.5$ and the electrode span $\phi = \pi/3$. The four figures show the unstable regions of the system when the damping ratios $\xi = 0.0001, 0.001, 0.01$, and 0.1 , respectively. It can be seen that the unstable region mainly appears at the natural frequency and $2\omega_1, 3\omega_1, \omega_1/2$, and $2\omega_1/3$. When the damping ratio is less than 0.01 , the system is most likely to show instability at ω_1 and $2\omega_1$. When the damping ratio is equal to 0.1 , the system is most likely to show instability at

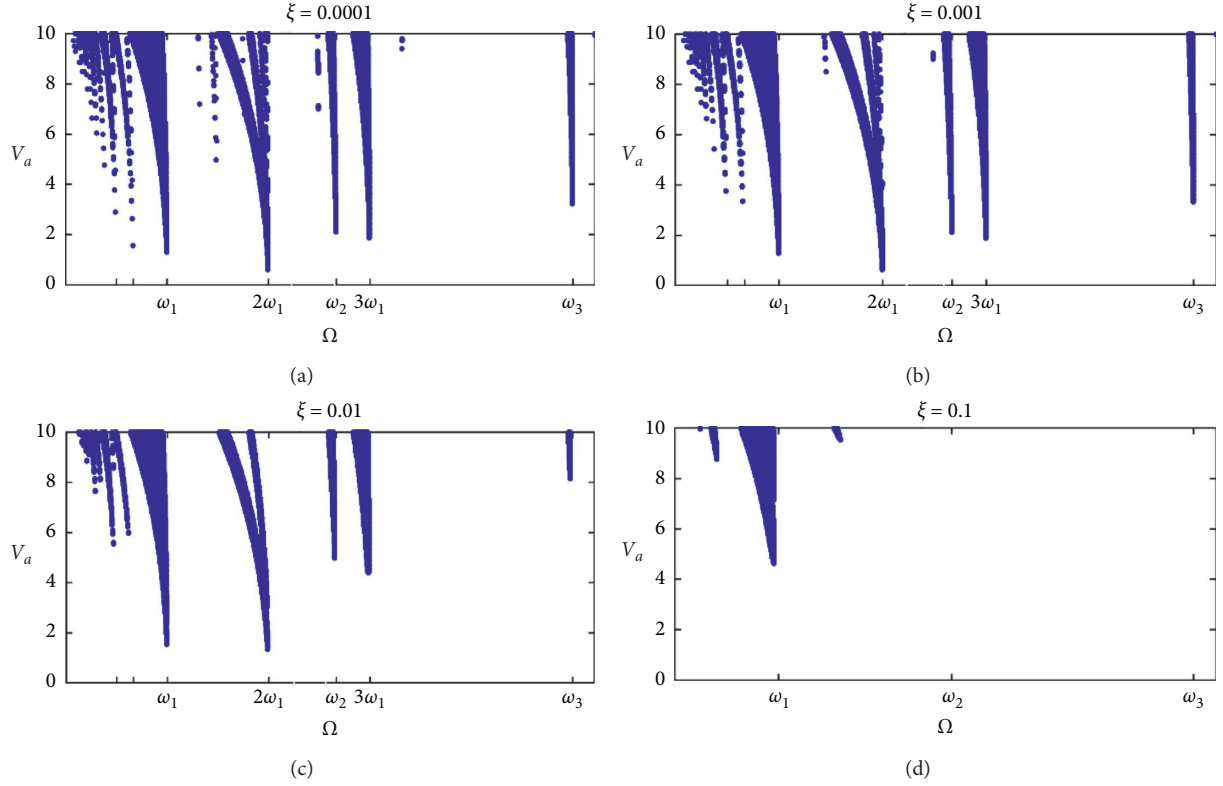


FIGURE 3: The stabilities of the circular-ring with respect to AC voltage and voltage frequency Ω for different damping ratio ξ .

$\Omega = \omega_1$. The system easily exhibits instability at the first-order natural frequency. When the damping ratio changes from 0.0001 to 0.001, the unstable region of the system changes little. When the damping is increased, the change in the unstable region at high frequencies is relatively large. When the damping ratio is equal to 0.1, the unstable region near the first-order natural frequency is most obvious.

The electrode span ϕ also affects the unstable region of the system. Figure 4 shows the unstable region of the system under different electrode span when the damping ratio $\xi = 0.001$. It can be seen that increasing the electrode span increases the width of the unstable region and also decreases the minimum voltage value of the unstable region.

From Figure 5, one noticed that the relationship between the minimum voltage value of the unstable region and the electrode angle when the excitation frequencies are ω_1 and $2\omega_1$. As shown in Figure 5, increasing the angle of the electrode decreases the minimum voltage value in the unstable region. Combined with Figure 4, it can be seen that increasing the electrode span increases the unstable region. When the electrode angle is small, the minimum voltage value V_{\min} will decrease rapidly as the electrode span ϕ increases. As the electrode span increases, the change rate of the voltage value becomes slower. When the electrode span is close to $\pi/3$, the change rate of the lowest voltage is close to zero.

5.2. Amplitude-frequency Response. The equation (29) is solved under zero initial conditions to obtain the forced response of the microring system.

The amplitude-frequency response is shown in Figure 6 when DC voltage $V_d = 0.5$, AC voltage amplitude $V_a = 1$ and damping ratio $\xi = 0.001$. The electrode span interval is $[0, \pi/3]$. Response amplitude changes with electrode span. In addition to the peaks near the natural frequency in the figure, there are also obvious peaks at $\omega_1/2$ and $2\omega_1$. The square of the potential difference can be written $V^2 = V_D^2 + 2V_D V_A \sin(\Omega t) + V_A^2 (1 - \cos(2\Omega t))/2$. When the voltage frequency is Ω , the electrostatic force has two frequency components Ω and 2Ω at the same time, so a peak appears at $\omega_1/2$. Due to the existence of electrostatic force, an electrostatic stiffness matrix $[\mathbf{K}^e]$ is introduced. The electrostatic stiffness couples the generalized coordinates $\alpha_k(t)$ and $\beta_k(t)$, so that the system has a peak at the non-natural frequency $2\omega_1$.

Figure 7 shows the relationship between the maximum system response and the electrode span. Other parameters are $V_d = 0.5$, $\xi = 0.001$, $\phi = \pi/3$, and $V_a = 1$. It can be seen that when the electrode span is $[0, \pi/4]$, the response amplitude increases as the electrode span increases. When the electrode span range is $[\pi/4, \pi/3]$, the response amplitude increases slowly, and finally the response amplitude decreases as the electrode span increases.

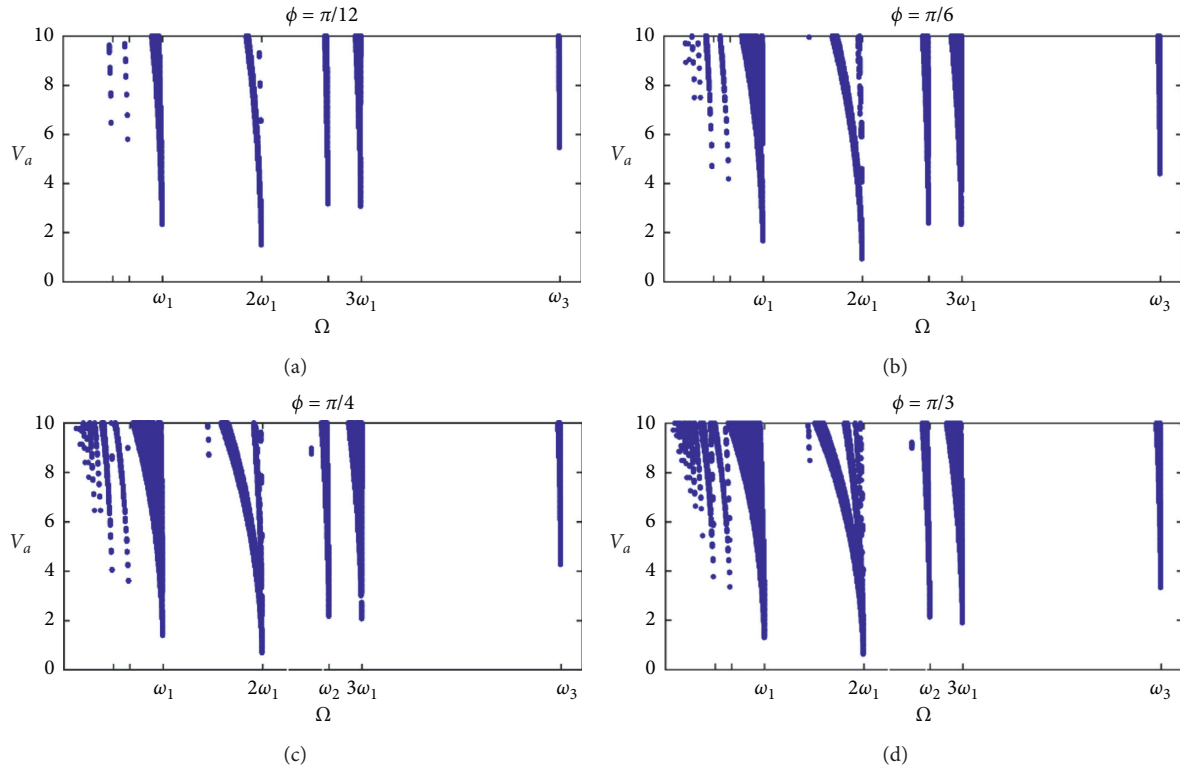


FIGURE 4: The unstable region of the system under different electrode spans.

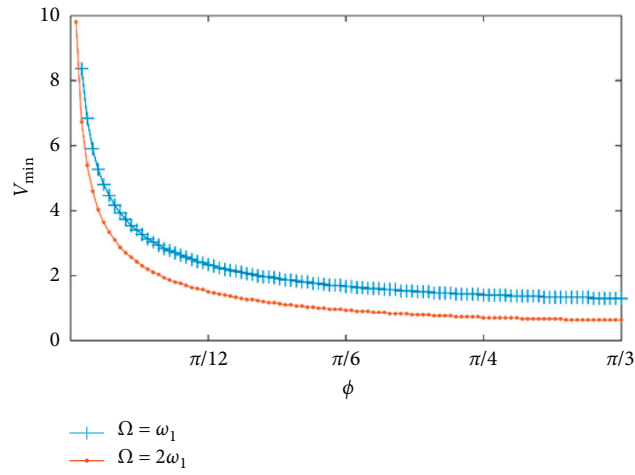


FIGURE 5: The minimum voltage value of unstable region changes with ϕ .

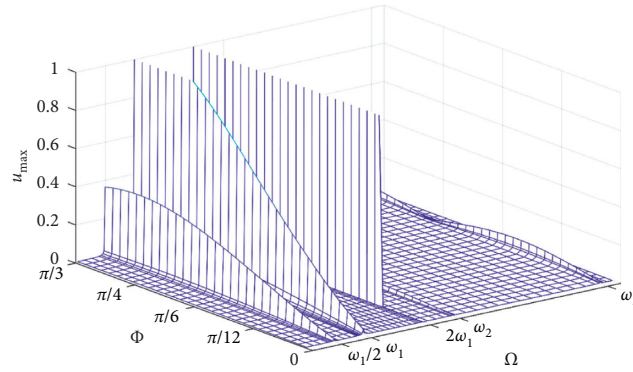


FIGURE 6: Amplitude-frequency response under different electrode spans.

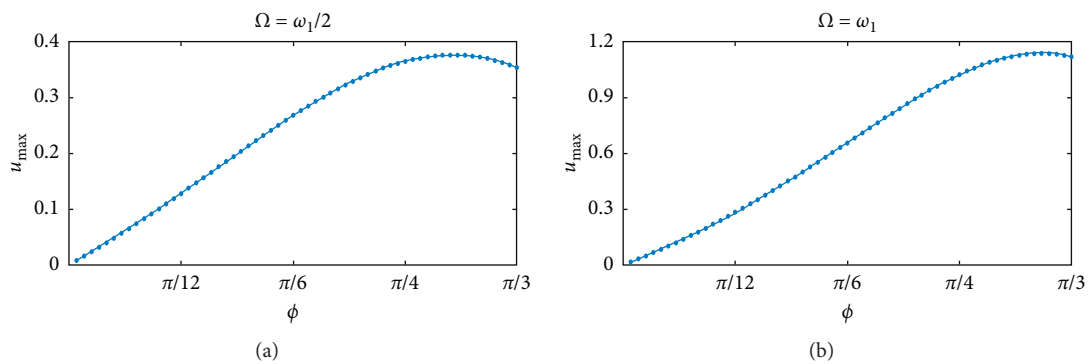


FIGURE 7: Response amplitude spectrum of the system at voltage excitation frequency $\omega_1/2$.

6. Conclusions

Based on the linearized model of the electrostatic force considering coupling with microring deformation, a dynamic model for a microring system has been established. Due to the coupling effect between the electrostatic force and the deformation of the microring, the stiffness of the microring is related to the electrostatic force parameters. Then, the time-varying voltage makes the stiffness matrix of the system have time-varying characteristics, which in turn causes the dynamic characteristics of the system to become complicated. Though the dynamic model obtained in this paper, the dynamical stability and response of the system with various parameters of the damping and electrode span are analyzed. Some conclusions are summarized as follows:

- (1) The unstable region of the system is mainly at the natural frequency and the frequency multiplication. Damping has a greater impact on unstable regions at high frequencies. When the damping is large, the unstable region of the system mainly appears near the first-order natural frequency.
- (2) As the electrode span increases, the unstable area of the system increases and the minimum value of the voltage in the unstable region and the rate of change of the minimum voltage value decrease. The response amplitude increases first and then

decreases as the increase of the electrode span. These are very helpful for the design of the electrostatically driven microring system.

Data Availability

All data included in this study are available upon request from the corresponding author.

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Shandong Province (ZR2018MEE021), Equipment Pre Research Fund Project (61402100501), and Innovation and Technology Program of Colleges in Shandong Province (2019KJB005).

References

- [1] Y. Huang, A. Sai Sarathi Vasani, R. Doraiswami, M. Osterman, and M. Pecht, "MEMS reliability review," *IEEE Transactions on Device and Materials Reliability*, vol. 12, no. 2, pp. 482–493, 2012.

- [2] B. J. Gallacher, J. S. Burdess, and K. M. Harish, "A control scheme for a MEMS electrostatic resonant gyroscope excited using combined parametric excitation and harmonic forcing," *Journal of Micromechanics and Microengineering*, vol. 16, no. 2, pp. 320–331, 2006.
- [3] F. T. Han, Q. P. Wu, and L. Wang, "Experimental study of a variable-capacitance micromotor with electrostatic suspension," *Journal of Micromechanics and Microengineering*, vol. 20, no. 11, p. 115034, 2010.
- [4] Y. Ma, S. Islam, and Y.-J. Pan, "Electrostatic torsional micromirror with enhanced tilting angle using active control methods," *IEEE/Asme Transactions on Mechatronics*, vol. 16, no. 6, pp. 994–1001, 2010.
- [5] C. Comi, A. Corigliano, A. Ghisi, and S. Zerbini, "A resonant micro accelerometer based on electrostatic stiffness variation," *Meccanica*, vol. 48, no. 8, pp. 1893–1900, 2013.
- [6] J.-E. Wong, J. H. Lang, and M. A. Schmidt, "An electrostatically-actuated MEMS switch for power applications," in *Proceedings of the IEEE Thirteenth Annual International Conference on Micro Electro Mechanical Systems (Cat. No. 00CH36308)*, pp. 633–638, IEEE, Miyazaki, Japan, January 2000.
- [7] Ö. Civalek and C. Demir, "A simple mathematical model of microtubules surrounded by an elastic matrix by nonlocal finite element method," *Applied Mathematics and Computation*, vol. 289, pp. 335–352, 2016.
- [8] B. Akgöz and Ö. Civalek, "Effects of thermal and shear deformation on vibration response of functionally graded thick composite microbeams," *Composites Part B: Engineering*, vol. 129, pp. 77–87, 2017.
- [9] F. Ebrahimi, M. R. Barati, and Ö. Civalek, "Application of Chebyshev-Ritz method for static stability and vibration analysis of nonlocal microstructure-dependent nanostructures," *Engineering with Computers*, vol. 36, no. 3, pp. 953–964, 2019.
- [10] Ö. Civalek, B. Uzun, M. Ö. Yaylı, and B. Akgöz, "Size-dependent transverse and longitudinal vibrations of embedded carbon and silica carbide nanotubes by nonlocal finite element method," *The European Physical Journal Plus*, vol. 135, no. 4, p. 381, 2020.
- [11] Ç. Demir and Ö. Civalek, "Torsional and longitudinal frequency and wave response of microtubules based on the nonlocal continuum and nonlocal discrete models," *Applied Mathematical Modelling*, vol. 37, no. 22, pp. 9355–9367, 2013.
- [12] F. M. Alsaleem, M. I. Younis, and H. M. Ouakad, "On the nonlinear resonances and dynamic pull-in of electrostatically actuated resonators," *Journal of Micromechanics and Microengineering*, vol. 19, no. 4, Article ID 045013, 2009.
- [13] J. Han, Q. Zhang, and W. Wang, "Static bifurcation and primary resonance analysis of a MEMS resonator actuated by two symmetrical electrodes," *Nonlinear Dynamics*, vol. 80, no. 3, pp. 1585–1599, 2015.
- [14] J. Machowski, J. Bialek, and J. Bumbypp. 3–5, John Wiley, Hoboken, NJ, USA, 2008.
- [15] J. A. Richards, *Analysis of Periodically Time-Varying Systems*, pp. 50–51, Springer Science & Business Media, Berlin, Germany, 2012.
- [16] F. M. Alsaleem, M. I. Younis, and L. Ruzziconi, "An experimental and theoretical investigation of dynamic pull-in in MEMS resonators actuated electrostatically," *Journal of Microelectromechanical Systems*, vol. 19, no. 4, pp. 794–806, 2010.
- [17] B. Wang, S. Zhou, J. Zhao, and X. Chen, "Size-dependent pull-in instability of electrostatically actuated microbeam-based MEMS," *Journal of Micromechanics and Microengineering*, vol. 21, no. 2, Article ID 027001, 2011.
- [18] H. Farokhi, M. H. Ghayesh, and S. Hussain, "Pull-in characteristics of electrically actuated MEMS arches," *Mechanism and Machine Theory*, vol. 98, pp. 133–150, 2016.
- [19] Y. Mafinejad, A. Kouzani, K. Mafinezhad, and R. Hosseini-zhad, "Low insertion loss and high isolation capacitive RF MEMS switch with low pull-in voltage," *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 1–4, pp. 661–670, 2017.
- [20] J. Francis and P. A. Michael, "Investigation of micro/nano motors based on renewable energy sources," *Materials Today: Proceedings*, vol. 27, pp. 150–157, 2020.
- [21] L. Yan, D. Liu, H. Lan, and Z. Jiao, "Compact traveling wave micromotor based on shear electromechanical coupling," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1572–1580, 2016.
- [22] L. Ran, "A novel three-dimensional contact model of piezoelectric traveling wave ultrasonic micromotor," *Smart Materials and Structures*, vol. 29, p. 7, 2020.
- [23] M. Bao and H. Yang, "Squeeze film air damping in MEMS," *Sensors and Actuators A: Physical*, vol. 136, no. 1, pp. 3–27, 2007.
- [24] P. Belardinelli, M. Brocchini, L. Demeio, and S. Lenci, "Dynamical characteristics of an electrically actuated microbeam under the effects of squeeze-film and thermoelastic damping," *International Journal of Engineering Science*, vol. 69, no. aug, pp. 16–32, 2013.
- [25] W.-C. Chuang, H.-L. Lee, P.-Z. Chang, and Y.-C. Hu, "Review on the modeling of electrostatic MEMS," *Sensors*, vol. 10, no. 6, pp. 6149–6171, 2010.
- [26] W.-M. Zhang, H. Yan, Z.-K. Peng, and G. Meng, "Electrostatic pull-in instability in MEMS/NEMS: a review," *Sensors and Actuators A: Physical*, vol. 214, pp. 187–218, 2014.
- [27] B. Chouvion, S. McWilliam, and A. A. Popov, "Effect of nonlinear electrostatic forces on the dynamic behaviour of a capacitive ring-based Coriolis Vibrating Gyroscope under severe shock," *Mechanical Systems and Signal Processing*, vol. 106, pp. 395–412, 2018.
- [28] S. Sieberer, S. McWilliam, and A. A. Popov, "Nonlinear electrostatic effects in MEMS ring-based rate sensors under shock excitation," *International Journal of Mechanical Sciences*, vol. 157–158, pp. 485–497, 2019.
- [29] W. Soedel and M. S. Qatu, pp. 64–69, Acoustical Society of America, 2005.
- [30] L. Xu and L. Qin, "Nonlinear analysis for a novel actuator," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 426–437, 2010.
- [31] Y. C. Hu, C. M. Chang, and S. C. Huang, "Some design considerations on the electrostatically actuated microstructures," *Sensors and Actuators A: Physical*, vol. 112, no. 1, pp. 155–161, 2004.
- [32] X. Ye, Y. Chen, D.-C. Chen, K.-Y. Huang, and Y.-C. Hu, "The electromechanical behavior of a microring driven by traveling electrostatic force," *Sensors*, vol. 12, no. 2, pp. 1170–1180, 2012.
- [33] Y. Li, T. Yu, and Y.-C. Hu, "The stiffness variation of a microring driven by a traveling piecewise-electrode," *Sensors*, vol. 14, no. 9, pp. 17256–17274, 2014.
- [34] T. Yu, J. Kou, and Y.-C. Hu, "Vibration of a rotating microring under electrical field based on inextensible approximation," *Sensors*, vol. 18, no. 7, p. 2044, 2018.
- [35] A. Fargas-Marques, J. Casals-Terré, and A. M. Shkel, "Resonant pull-in condition in parallel-plate electrostatic

- actuators,” *Journal of Microelectromechanical Systems*, vol. 16, no. 5, pp. 1044–1053, 2007.
- [36] S. C. Huang and W. Soedel, “Effects of coriolis acceleration on the free and forced in-plane vibrations of rotating rings on elastic foundation,” *Journal of Sound and Vibration*, vol. 115, no. 2, pp. 253–274, 1987.
- [37] K. Farhang and A. Midha, “Steady-state response of periodically time-varying linear systems, with application to an elastic mechanism,” *Journal of Mechanical Design*, vol. 117, no. 4, pp. 633–639, 1995.

Research Article

Metal Roof Fault Diagnosis Method Based on RBF-SVM

Liman Yang,¹ Lianming Su,¹ Yixuan Wang ,¹ Haifeng Jiang,² Xueyao Yang,¹ Yunhua Li ,³ Dongkai Shen,¹ and Na Wang ¹

¹School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

²Center Internatinal Group Co, Ltd., Beijing 100176, China

³Beijing University of Aeronautics & Astronautics, Beijing, China

Correspondence should be addressed to Yixuan Wang; magic_wyx@163.com and Na Wang; lion_na987@buaa.edu.cn

Received 16 June 2020; Revised 20 October 2020; Accepted 13 November 2020; Published 4 December 2020

Academic Editor: Yanan Li

Copyright © 2020 Liman Yang 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.

Metal roof enclosure system is an important part of steel structure construction. In recent years, it has been widely used in large-scale public or industrial buildings such as stadiums, airport terminals, and convention centers. Affected by bad weather, various types of accidents on metal roofs frequently occurred, causing huge property losses and adverse effects. Because of wide span, long service life and hidden fault of metal roof, the manual inspection of metal roof has low efficiency, poor real-time performance, and it is difficult to find hidden faults. On the basis of summarizing the working principle of metal roof and cause of accidents, this paper classifies the fault types of metal roofs in detail and establishes a metal roof monitoring and fault diagnosis system using distributed multisource heterogeneous sensors and Zigbee wireless sensor networks. Monitoring data from strain gauge, laser ranging sensor, and ultrasonic ranging sensor is utilized comprehensively. By extracting time domain feature, the data trend characteristics and correlation characteristics are analyzed and fused to eliminate erroneous data and find superficial faults such as sensor drift and network interruption. Aiming to the hidden faults including plastic deformation and bolt looseness, an SVM fault diagnosis algorithm based on RBF kernel function is designed and applied to diagnose metal roof faults. The experimental results show that the RBF-SVM algorithm can achieve high classification accuracy.

1. Introduction

Metal roof enclosure system has been rapidly developed in public buildings and industrial buildings due to its excellent structural performance, beautiful appearance, and environmental protection [1, 2]; its typical applications are shown in Figure 1. Metal roof enclosure system in service period just in major cities of China covers no less than 1 billion square meters since 2008 and grows at a rate of no less than 50,000 square meters per year; therefore, its safety and reliability have drawn extensive attention.

Metal roofs are susceptible to extreme weather and other factors during usage. In recent years, accidents of metal roof have occurred occasionally. In 2010, 2011, and 2013, the metal roof of Beijing Airport T3 terminal was lifted up by wind three times; these accidents caused more than a thousand flights to delay, which seriously affected the normal operation of Beijing Airport. In 2016, metal roof of the Guangzhou Baiyun Airport terminal building was

rippled off by a weak wind, and the speed of the wind did not exceed the design safety threshold of the metal roof. The No.3 typhoon “Tian Ge” in 2017 caused a large number of metal roofs in Zhuhai to be destroyed [3]. By far, the solutions of these problems are mainly to reinforce the safety check calculation in the design phase to improve the roof bearing capacity threshold or increase the wind-proof clamp and other reinforcement measures at local negative high-pressure area, but these methods mainly focus on the theoretical model and general regularity analysis; they do not have the support of real-time monitoring data, there is few research on the real-time monitoring system of metal roof during service period, and this is the bottleneck that needs to be broken in wind-resistance and disaster prevention engineering of large-span building metal roof [4]. The metal roof of airport rippled off by wind can be shown in Figure 2.

In view of the damage of metal roof enclosure system, this paper used structural health monitoring technology (SHM) [5] to replace traditional manual inspection; the real-

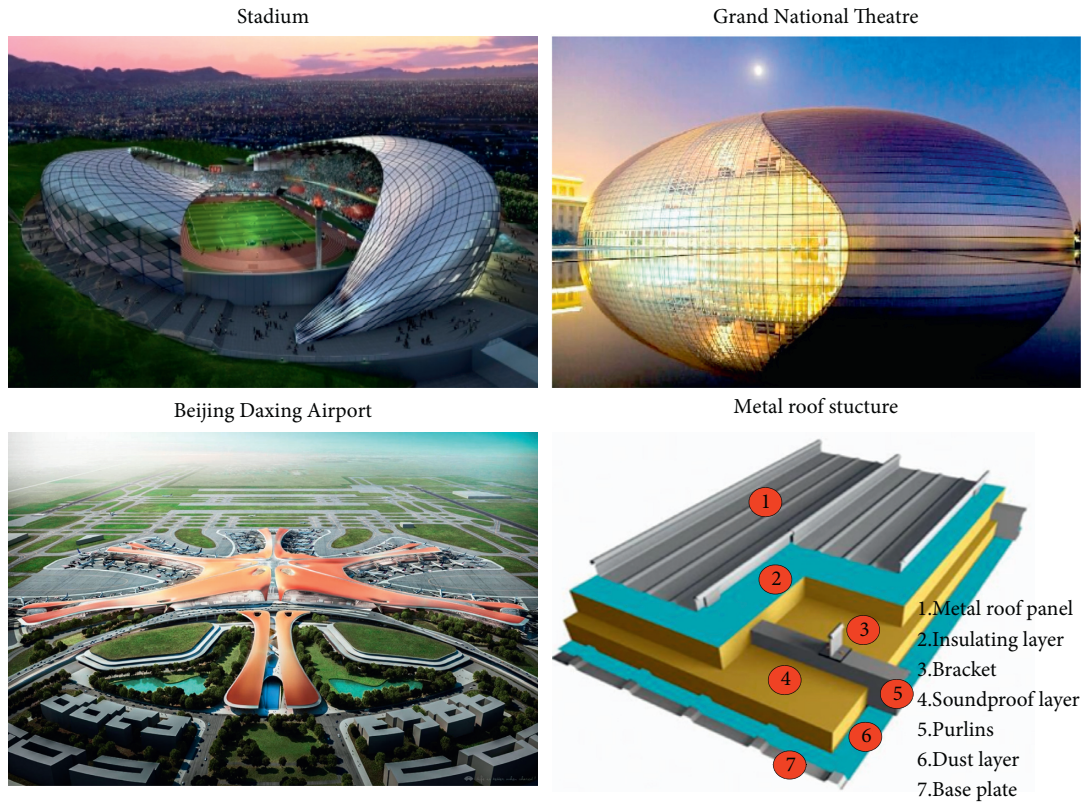


FIGURE 1: Widely used metal roof enclosure system.

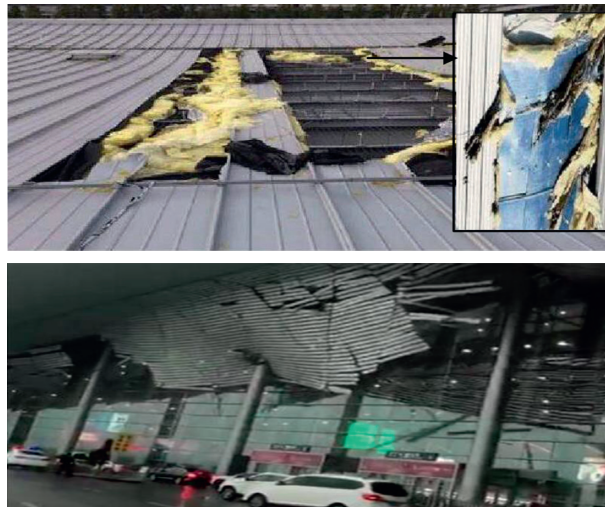


FIGURE 2: The metal roof of airport lifted off by wind.

time monitoring of metal roof can reduce the occurrence of accidents effectively. This kind of technology has become a global research hotspot and is widely used in large-scale civil infrastructure [6], such as airports, high-rise buildings [7], large-span buildings, stadiums, railways [8], and pipelines [9].

At present, the automatic monitoring methods of civil structures are mainly divided into two categories: intelligent robot monitoring and sensor network monitoring. Robot

monitoring is often used in harsh environments, and the monitoring area is limited. Because of the wide span, stable working environment and long period of performance changing of metal roof, distributed sensor network monitoring method should be considered in priority in metal roofs.

Sensor network monitoring obtains monitoring data of building structure by embedding or pasting sensors on the surface of the monitored structure, and based on this, fault

diagnosis and health evaluation of the building structure are carried out [10]. The construction of sensor network monitoring system is mainly divided into three parts, which are sensor selection, monitoring network construction, and data storage and analysis. In the sensor selection part, Yan et al. [11] in Southwest Jiaotong University designed a bridge structure health monitoring system for Hutong Yangtze River Bridge. In this paper, not only the large amount of data and wide coverage of sensor features were considered, the correlation between the collected data was also used. In this way, the accuracy of fault diagnosis was increased and, at the same time, some hidden faults can be discovered by data fusion algorithm [12]. In data transmission part, wireless sensors are widely used in the project, for example, in order to monitor the health of bridge structure, Alampalli et al. [13] realized the monitoring system using wireless network; they also used big data cloud computing technology to evaluate the health of the bridge. In Poland, SALA [14] in Polish Academy of Sciences established a health monitoring system for a 40-meter long steel truss bridge using wireless sensor network, and they also used solar energy to power the piezoelectric ceramic strain gauges in order to save the energy consumption. In this paper, we use Zigbee [15] wireless network combined with Profibus and Internet to form a star topology for data transmission; the master and slave PLC communicate structure can improve the data processing capacity significantly. In data analysis and fault diagnosis algorithm part, there are mainly three ways to process the data, the method based on modern signal processing technology [16–18], the method based on statistics [19], and the method based on machine learning and data mining techniques. In recent years, machine learning [20] and data mining techniques are widely used in SHM technology, such as Bayesian neural networks [21], wavelet packet analysis, posterior probability SVM, DS evidence theory, and OLSVM [22] algorithm. In this paper, in order to comprehensively consider the trend characteristics and correlation characteristics of large amount of monitoring data, the support vector machine (SVM) algorithm based on RBF kernel function is used for fault diagnosis of metal roof. This algorithm can multiclassify the monitoring data by using multidimensional feature value input, and it can also improve the classification accuracy through grid search method, so as to achieve accurate fault diagnosis of the metal roof.

This paper is divided into three sections. The first section analyzes the damage mechanism of metal roof failure and selects the appropriate sensors and transmission scheme to build the metal roof health management system. The second section summarizes the types of faults that need to be distinguished by algorithms and introduces the preprocesses of monitoring data. In the third section, the RBF-SVM algorithm to classify the monitoring data is realized, and we use grid search method to optimize the parameters of RBF-SVM. Finally, the classification results are evaluated in the last section of the paper.

2. Metal Roof Failure Causes Analysis and Fault Monitoring System Composition

Metal roofs are exposed to the natural environment for a long period; therefore, they are easily affected by external weather conditions. Wind destroying is the main cause of metal roof damage [23]. This section will first analyze the types and causes of metal roof failures, and then based on the failures, a metal roof online monitoring and health management system will be established.

2.1. Summary of Failure Reasons of Metal Roof. The research object of this paper is standing seam metal roof enclosure system, which is widely used on the roof of industry buildings. Standing seam metal roof is a metal roof enclosure system formed by roofing panels, fixed supports, purlins, and pressed bottom plates. When the roof is subjected to the upward wind suction, the wind force is transmitted to the fixed support through the interboard nip connection structure and then transmitted to the purlins by the self-tapping screw and finally transmitted to the main structure safely. The metal panel is fixed on the purlins by a T-shaped bracket, and the board rib of the roof panel is clamped on the plum head of the T-shaped support so that the panel is fixed on the purlin of the roof. The structure of the system is shown in Figure 3. The main damage form of metal roof is excessive deformation and displacement perpendicular to the longitudinal direction of the roof panel, which causes the metal roof panel to be plastically deformed or even rippled off. Since the metal roof is not fixed in the lateral direction, the main monitoring object is the longitudinal stress and displacement of the metal roof. The lateral displacement is only used as a reference factor for monitoring the fatigue damage of the roof panel without excessive analysis.

Combined with the research and analysis of metal roof accidents in recent years, the types and causes of metal roof failures are summarized as follows:

- (1) Plastic deformation: during the long-term service of metal panels, due to the influence of alternating loads, especially the wind load, metal panels are plastically deformed, resulting in a decrease in the wind-resistant performance of the metal panels, affecting the safety of the metal roof.
- (2) Lateral slip: standing seam metal roof is connected with the purlins by the T-shaped bracket. When the ambient temperature changes, the metal roof will deform due to the thermal expansion and contraction, resulting in the internal stress of the metal roof becoming larger. Repeated stress changes can cause fatigue damage to the metal roof, so attention should be paid to the lateral slip of the metal roof due to temperature.
- (3) Longitudinal deformation: under weather load such as wind and snow, longitudinal deformation will

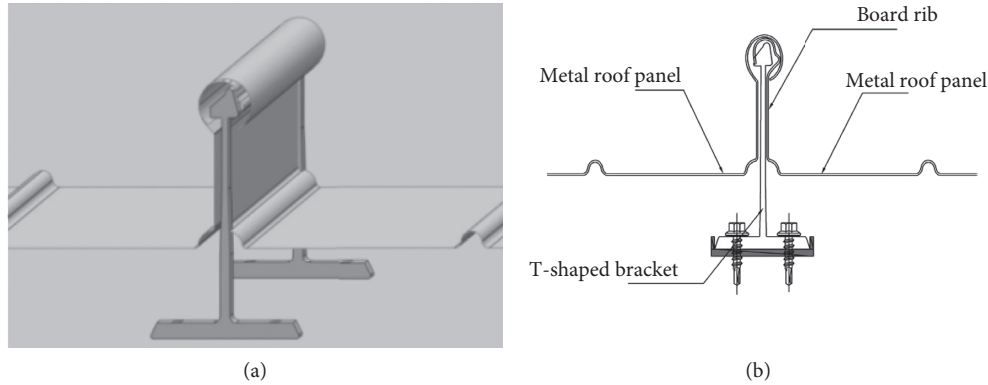


FIGURE 3: The structure of metal roof enclosure system. (a) The structure of standing seam metal roof. (b) The panel is fixed on the purlins with T-shaped bracket.

occur, which is the main reason affecting the health of metal roofing. Excessive longitudinal deformation easily leads to plastic deformation of metal roof and affects fatigue life and makes the roof more easily to be lifted up by wind.

- (4) Bolt looseness: standing seam metal roof is fixed on the purlins by bolts. The bolts may tend to loosen during long-term service, which leads to the safety hazard of the metal roof being damaged by the wind. Therefore, the bolt looseness also affects the metal roof, and this kind of fault is difficult to be monitored by one kind of sensor; we use two types of monitoring data to diagnose this fault.

The main fault forms and causes of standing seam metal roof are summarized in the following Table 1:

2.2. Real-Time Monitoring and Health Management System of Metal Roof Enclosure System. On the basis of summarizing the fault types and causes of metal roof, a distributed multisource heterogeneous sensors' scheme was selected to establish a wireless transmission network and a data backup storage system to monitor the health of metal roof in real time.

In this system, three kinds of sensors are used to implement distributed multisource heterogeneous monitoring on metal roof; distributed monitoring can obtain monitoring data in different areas of the roof, making the data more comprehensive. Multisource heterogeneous sensors can take advantage of the correlation of different types of data, which can not only ensure the accuracy of the data but also diagnose the hidden faults through data fusion. The sensors include a stress sensor, a laser ranging sensor, and an ultrasonic ranging sensor. The terminal node sensor is shown in Figure 4. The type and function of the sensor are shown in Table 2:

The metal roof real-time monitoring and health management system includes three subsystem modules: data

acquisition module, wireless network transmission module, data backup and interface module. The function and relationship among different modules are shown in Figure 5.

The data acquisition module consists of distributed sensors which layout in different areas of the roof. The data transfer to the STM32 SCM was combined by ADS1256 high-precision A/D acquisition chip. Then, the measurement data is transmitted to Zigbee wireless network transmission module; the transmission module uses Zigbee protocol to transfer data to the PLC master and slave station for data processing by wireless mode. PLC S7-1500 CPU is used for master station of the system; meanwhile, PLC S7-1200 CPU is used for slave station, and finally, the collected data is converted into decimal readable data by calculation in PLC and transfer to upper computer. The data is stored in database of the upper computer. The interface of the system is developed in upper computer by WinCC, and it realizes the real-time monitoring and fault diagnosis of the metal roof by analyzing the data stored in its database. Since mature solutions are mostly used in this monitoring system, its economic cost can be disregarded compared with metal roofing system. Meanwhile, it is more valuable to reduce the hidden danger caused by metal roof failure through real-time data analysis.

3. Metal Roof Fault Classification and Data Preprocessing Based on Monitoring Data

The data collected by the metal roof real-time monitoring system is saved in the database of the upper computer, and the fault of metal roof can be diagnosed by getting data from the database. Because of the large number of distributed sensors, it will be difficult to analyze all sensor data. Therefore, the stress and displacement thresholds should be set according to the physical properties of the metal roof. Then, we calculate the proportion of the sensors where the stress and displacement reach the threshold in the region

TABLE 1: Main fault and causes of metal roof.

Fault forms	Fault causes	Expression form
Plastic deformation	Alternating loads cause damage accumulation	Deformation and decrease of wind-resistant
Lateral slip	Temperature load	Inner stress and lateral deformation
Longitudinal deformation	Alternating loads, especially the wind	Longitudinal stress and deformation
Bolt looseness	Alternating wind loads	Increase of longitudinal displacement



FIGURE 4: Terminal node sensor layout.

TABLE 2: Type and function of the sensor.

Fault forms	Sensor	Function
Plastic deformation	Strain gauge, laser ranging sensor	Compare with past data to calculate the plastic deformation value
Lateral slip	Ultrasonic ranging sensor	Monitoring lateral slip distance
Longitudinal deformation	Strain gauge, laser ranging sensor	Cooperate with each other to monitor the longitudinal stress and displacement
Bolt looseness	Strain gauge, laser ranging sensor	Cooperate with each other to find the hidden fault

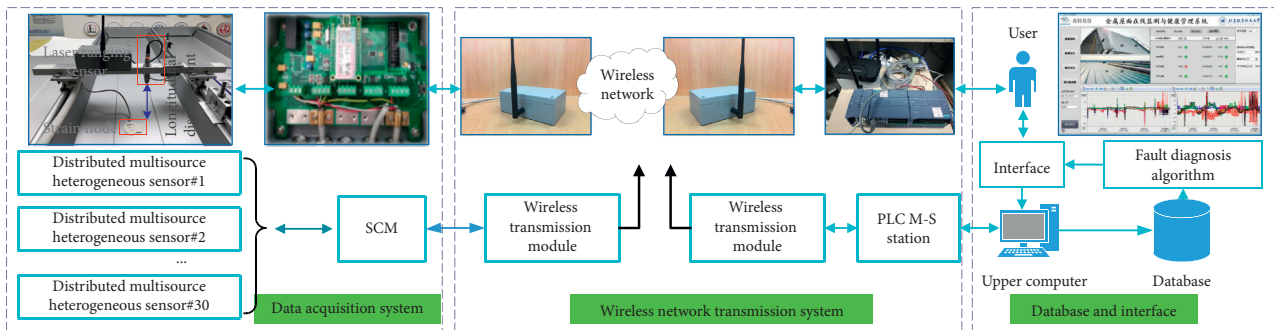


FIGURE 5: Structure diagram of whole system.

and select the sensor data from the region with high fault proportion for fault diagnosis.

3.1. Metal Roof Fault Classification Based on Monitoring Data. Considering the metal roof itself fault and network transmission fault, the fault types are mainly divided into three categories, metal roof fault, sensor fault, and network failure. Metal roof fault mainly includes three types, long-term plastic deformation of the panel, rain and snow

accumulation-induced or wind-induced longitudinal deformation, and bolt of T-shaped bracket looseness; sensor faults include zero drift of sensor acquisition data and short-term interference of the sensor, and network failure is the interruption of data transmission. The fault types of metal roof are summarized in Table 3.

The metal roof fault diagnosis process is divided into two levels. In the first level, the faults can be diagnosed by the real-time monitoring system, they are easy to be distinguished. However, some faults are difficult to distinguish or

TABLE 3: Fault types of metal roof enclosure system.

Fault types of metal roof enclosure system	
Metal roof fault	Panel plastic deformation Panel longitudinal deformation Bolt looseness of bracket
Sensor fault	Zero drift Short-term interference
Network failure	Transmission interruption

they need two kinds of data to diagnosis, so in the second level, machine learning algorithm is used to classify complex or hidden faults of metal roof enclosure system. The specific diagnosis process is shown in Figure 6.

In the second level, the support vector machine (SVM) algorithm based on RBF kernel function is used to classify the fault type. This part mainly focuses on the faults that are complex or hidden and diagnosed the fault by correlation of data measured by multisource heterogeneous sensors. Based on the built-up metal-roof real-time monitoring system, the monitoring data of metal roof in different fault states are obtained from experimental system, and the monitoring data are preprocessed by time domain analysis to extract the characteristic quantity under different conditions of metal roof, and then the training set of RBF-SVM is established by corresponding the characteristic quantity and fault. Finally, the training set is used for training the RBF-SVM model and realizing the fault diagnosis of the metal roof.

3.2. Data Preprocessing Method. In the data preprocessing process, the correlation characteristics of different kinds of data collected by the heterogeneous sensors can effectively avoid the error data from faulty sensor and ensure the accuracy of fault diagnosis and fault early warning. The SVM algorithm's training needs to select representative data to include various situations from a large number of detection and experimental data, thus these groups of data may come from different dates. The typical data selected are shown partly in Figure 7, in which the types of data are denoted. The red curve in Figure 7 is the strain data from strain gage, the blue curve is the displacement data collected by the laser ranging sensor, the state of the metal roof is judged by the trend information of the data collected by these two sensors, and the correlation information of the data collected by two heterogeneous sensors avoids the error data and makes fault diagnosis more accurate.

For convenience in comparing, the data shown in the abovementioned figure ignores the impact of the numerical dimension on result. The monitoring data are normalized to be distributed between 0 and 1, and the formula is as follows:

$$y' = \frac{y - y_{\min}}{y_{\max} - y_{\min}}. \quad (1)$$

Different causes of roof deformation will lead to different trends of detection data of different sensors, which is the advantage and necessity of using a variety of sensors for data acquisition. The dataset shown above contains four types of data, which are the monitoring data under normal roof condition, the monitoring data when the roof is longitudinally deformed, the monitoring data when the sensor fault occurs, and the monitoring data when the bolts are loosen. It can be seen from Figure 7 that under normal condition, the stress and displacement data of roof panel only have little changes and there is no obvious fluctuation. Two types of data from heterogeneous sensor have high correlation. In this condition, the states of sensor and roof are both normal. When the roof is longitudinally deformed, monitoring data from laser ranging sensor and strain gage have same trend changes, which means they have high correlation. In this condition, it can be judged that the roof has undergone longitudinal deformation. When in the sensor fault condition, one of the sensors experiences a zero drift fault, the data from two sensors are less of correlation and the data of one sensor is in rise trend. As for the bolt loosen fault, it is difficult to find by manual inspection; however, it can be diagnosed by heterogeneous sensor system. When the bolts fixed on purlins are loosen, the panel will be displaced in the longitudinal direction in a large extent but, at the same time, the strain value of the panel is less and it changes slower than the displacement, just like the dataset shown in Figure 7; the displacement data change in a large range, while the strain value changes less. In this situation, we can assume that the bolt of metal roof has loosened.

Considering the above information, five kinds of statistics are selected as input characteristics: average and standard deviation of metal roof longitudinal displacement data and roof panel strain data are selected to reflect trend information, and difference between displacement and the strain data variance and correlation coefficients of these two types of data are selected to reflect correlation information. The statistic calculation formula is shown in Table 4.

In order to avoid the influence of data value range on the fault diagnosis result, the input characteristics quantity needs to be normalized, and the formula is as follows:

$$y_{\text{in}} = \frac{y_i - \mu}{\sigma(\mu)}, \quad (2)$$

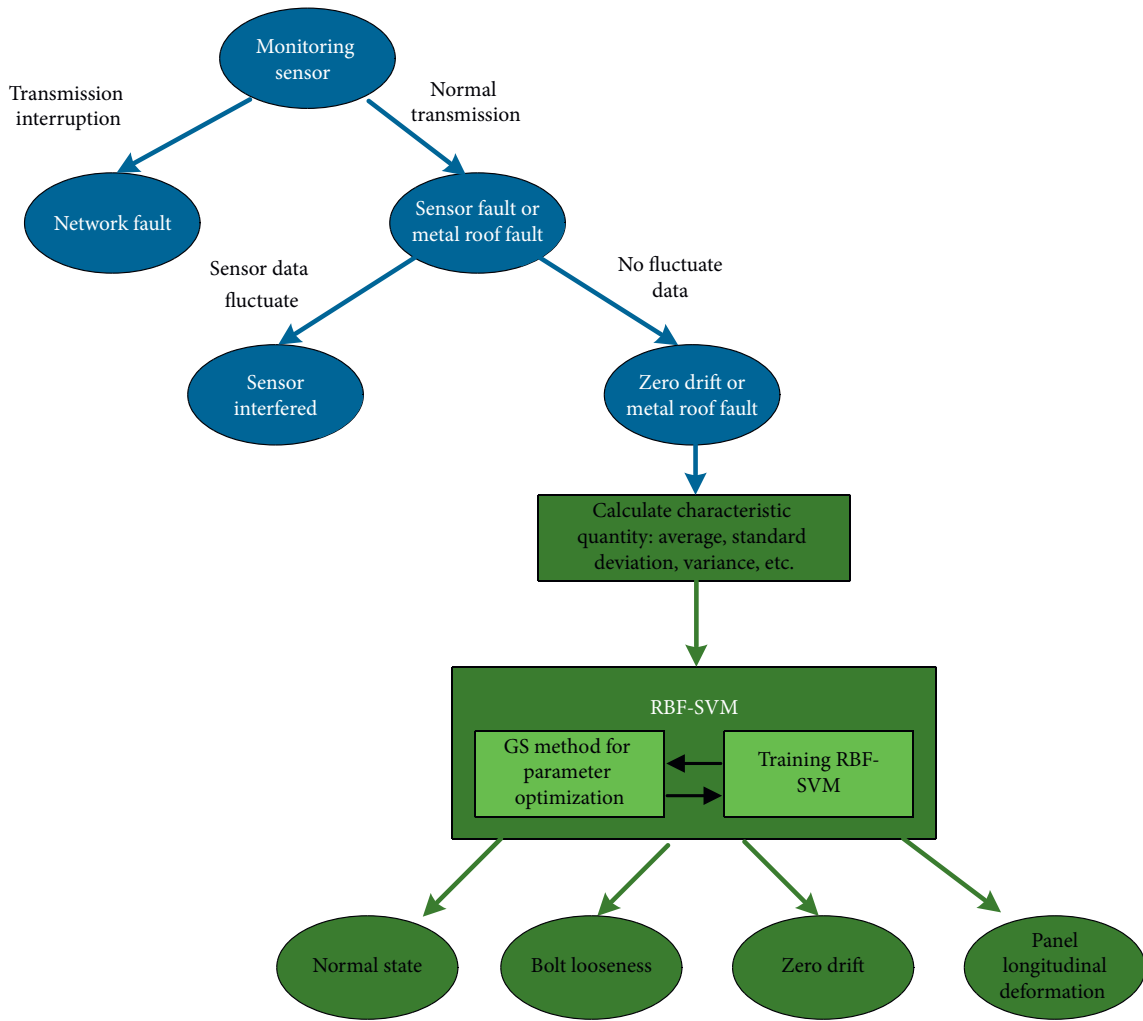


FIGURE 6: Fault diagnosis process.

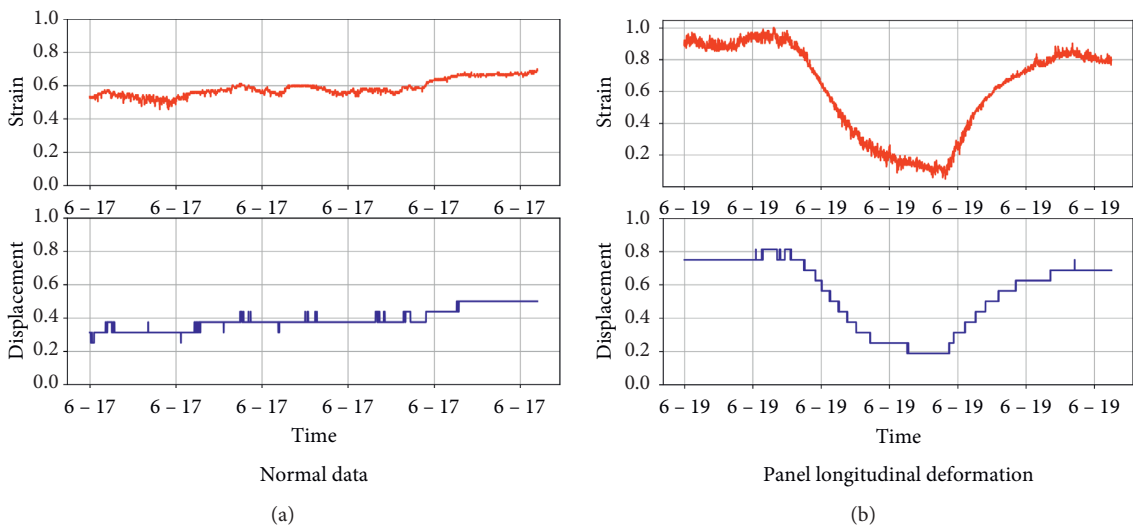


FIGURE 7: Continued.

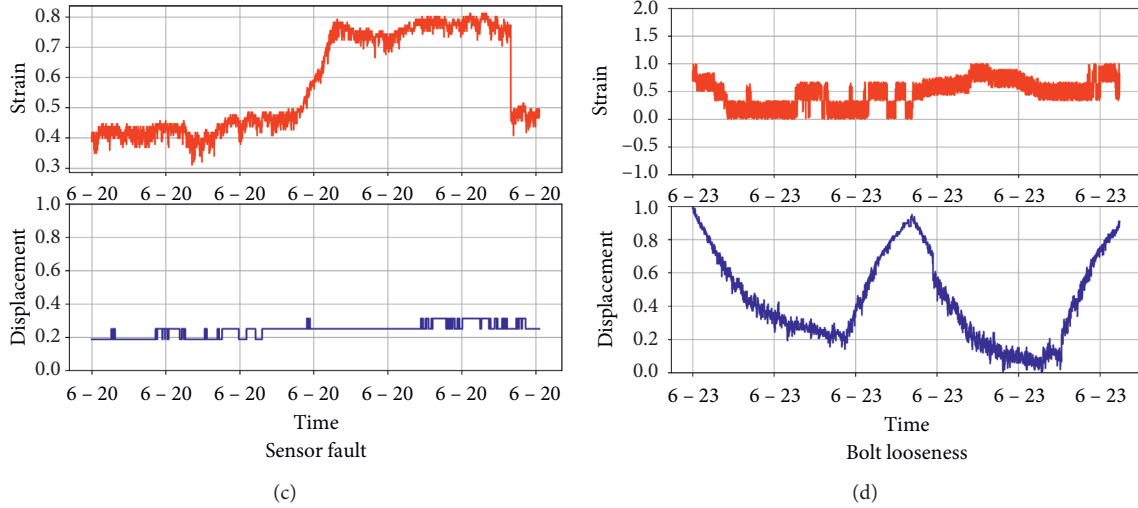


FIGURE 7: Sensor dataset.

TABLE 4: Statistics and formulas.

	Statistics	Formulas
Trend information	Average of strain	$\mu = \sum_{t=1}^n y_t/n$
	Average of displacement	
	Standard deviation of strain	
	Standard deviation of displacement	
Correlation information	Normalized difference between strain and displacement	$\sigma(\mu) = \sqrt{(1/N) \sum_{i=1}^N (y_i - \mu)^2}$
	Variance between strain and displacement	$Cov(x, y) = E[xy] - E[x]E[y]$
	Correlation coefficients between strain and displacement	$\rho_{xy} = (Cov(x, y)/\sigma(x)\sigma(y))$

where μ is the average of characteristics, $\sigma(\mu)$ is the stand deviation of the characteristics, and y_{in} is the normalized characteristics.

Finally, 593 sets of characteristics quantities are selected for training and testing including 70% of training set and 30% of testing set. The input characteristics quantity and labels are shown in Table 5; labels 0–4 respectively express the normal data of the metal roof, the strain sensor fault data, the laser ranging sensor fault data, the roof large longitudinal deformation data, and the bolt looseness data.

4. Metal Roof Fault Diagnosis Algorithm Based on RBF-SVM and Parameter Optimization Method

Through data preprocessing, the trend information and correlation information of monitoring data are obtained. By using these two types of information, multiclassification of monitoring data can be realized, and in this way, fault diagnosis of metal roof can be achieved. In the classification algorithm, SVM has automatic model selection, which can transform training into quadratic programming problem

and it also has good learning ability of small sample. RBF kernel function SVM algorithm can meet the distinction of different fault types of metal roof and reduce computational complexity to ensure the efficiency of the fault diagnosis algorithm.

4.1. SVM Algorithm. SVM is an efficient data classification algorithm. By finding support vectors in dataset, it establishes a hyperplane and classifies the data. For a simple two-dimensional linear separable problem, a straight line can be established to separate the data; while in fault diagnosis of metal roof, the monitoring data needs to be classified into multiple categories. Therefore, it is necessary to map the characteristic quantity of the monitoring data to a high-dimensional feature space H and then construct the (generalized) optimal hyperplane in H to realize multi-classification of monitoring data.

The simple SVM classifier is used in binary classification. Assume that input points set $\{x_i\} \in R^d$ consists of two types of points. If the points x_i belong to the first class, then $y_i = 1$, or if they belong to the second class, then $y_i = -1$. Assuming that training sample set is

TABLE 5: The training and testing dataset of RBF-SVM.

Strain avg	Strain std	Disp. avg	Disp. std	Diff.	Rolling corr.	Rolling cov.	Labels
17.4860	0.1268	0.7000	0.0000	0.2371	0.81	0.02	0
17.4510	0.1408	0.7000	0.0000	0.2329	0.81	0.02	0
17.6995	0.1513	0.7000	0.0000	0.2624	0.81	0.02	0
17.8022	0.1273	0.7500	0.0504	0.2686	0.81	0.02	0
...
22.1410	0.0452	4.1244	0.3018	0.3829	0.15	0.02	1
22.0745	0.0544	4.1063	0.2647	0.3771	0.04	0.01	1
22.0267	0.0494	4.1418	0.2892	0.3673	-0.09	-0.01	1
21.8738	0.0779	4.0773	0.3007	0.3568	-0.19	-0.03	1
...
22.8608	0.0787	1.8000	0.0000	0.7435	-0.66	-0.11	2
22.8713	0.1005	1.8000	0.0000	0.7447	-0.63	-0.10	2
22.9087	0.1093	1.8000	0.0000	0.7492	-0.59	-0.09	2
...
23.6892	0.1238	1.9000	0.0000	0.8298	0.90	0.01	3
23.5877	0.1108	1.9000	0.0000	0.8177	0.92	0.01	3
23.6752	0.1132	1.9000	0.0000	0.8281	0.93	0.01	3
...
21.1118	0.0458	-0.4192	0.0876	0.2560	0.05	0.00	4
21.1899	0.0339	-0.2800	0.0783	0.2434	0.39	0.01	4
21.2004	0.0303	-0.0523	0.1014	0.1989	0.58	0.03	4
21.2044	0.0314	0.1271	0.0638	0.1629	0.69	0.04	4
17.4860	0.1268	0.7000	0.0000	0.2371	0.81	0.02	0

$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, $y_i \in \{-1, +1\}$, the goal of the support vector machine is to find a hyperplane that satisfies the classification requirements according to the principle of minimum structural risk and linearly divide the dataset.

In the sample space, the divided hyperplanes can be described by the following linear equation:

$$\omega^T x + b = 0, \quad (3)$$

where $\omega = (\omega_1; \omega_2; \dots; \omega_d)$ is the normal vector, which determines the direction of the hyperplane; b is the displacement term, which determines the distance between the hyperplane and the origin. The distance from any point in the sample space to the hyperplane can be written as follows:

$$r = \frac{|\omega^T x + b|}{\|\omega\|}. \quad (4)$$

Assuming that hyperplane (ω, b) can classify training samples correctly, for $(x_i, y_i) \in D$, if $y_i = +1$, then $\omega^T x_i + b > 0$, or if $y_i = -1$, then $\omega^T x_i + b < 0$. In this way, the formula can be expressed as follows:

$$\begin{cases} \omega^T x_i + b \geq 1, & y_i = +1, \\ \omega^T x_i + b \leq -1, & y_i = -1. \end{cases} \quad (5)$$

The training sample pointing closest to the hyperplane makes the equal sign true. They are called as ‘‘support vectors,’’ and the sum of distances of two different classes of support vectors to the hyperplane can be expressed as follows:

$$\gamma = \frac{2}{\omega}. \quad (6)$$

It is called gap. The optimal hyperplane can not only separate the samples of the two classes without error but also maximize the gap of two classes. The classification problem can be transformed into a minimum value problem with constraints, and it can be expressed as follows:

$$\begin{aligned} \min_{\omega, b} & \frac{1}{2} \|\omega\|^2 \\ \text{s.t. } & y_i (\omega^T x_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned} \quad (7)$$

This is the basic formula of SVM algorithm.

In the basic principles, most are ideal hypotheses. However, in the process of using SVM for fault diagnosis, fault data cannot be linearly divided. It is necessary to allow the SVM algorithm to make errors in dividing certain monitoring data to avoid overfitting. In this situation, soft margin is used, and nonnegative slack vector ξ_i is added to the formula, so the target function of problem has changed as follows:

$$\begin{aligned} \min_{w,b} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t. } y_i(\omega^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0, \quad i = 1, 2, \dots, m, \end{aligned} \quad (8)$$

where $C \sum_{i=1}^m \xi_i$ is the penalty term; C is the penalty factor, which controls the degree of penalty for wrong samples. The higher the C value, the higher the penalty for the error, and the stricter the linear classification. It is important to choose the optimal C in SVM, and the optimization algorithm will be mentioned in the following part of the paper.

Meanwhile, the fault diagnosis of metal roof is a nonlinear multiclassification problem. In order to realize the nonlinear multiclassification, Kernel function can be used to map the original nonlinear multiclassification sample into the high-dimension space H , transform the nonlinear classification problem into the linear classification problem, and, at the same time, use penalty term to ensure the accuracy of classification. Kernel function $K(x, x_i)$ can be expressed as follows:

$$K(x, x_i) = \phi(x) \cdot \phi(x_i). \quad (9)$$

According to the classification sample, by selecting the appropriate kernel function $K(x, x_i)$ instead of the inner product operation in the high-dimensional space, the linear classification after a linear transformation can be realized, and the computational complexity is not increased. Similar to the optimization method without linear transformation, in order to solve the dual problem, the Lagrangian multiplier is used to construct the Lagrangian function. The final optimal classification function is as follows:

$$f(x) = \text{sgn} \left[\sum_{i=1}^n \alpha_i^* y_i K(x, x_i) + b^* \right]. \quad (10)$$

In this paper, we use RBF kernel function to optimize the SVM algorithm called RBF-SVM, the formula is as follows:

$$K(x, y) = e^{-\gamma \|x - y\|^2}. \quad (11)$$

4.2. Grid Search (GS) Method Parameter Optimization and Result Analysis. In RBF-SVM algorithm, in order to solve the nonlinear multiclassification problem, the penalty factor and RBF kernel function are used to reduce overfitting and computational complexity. The value of the penalty factor C and kernel parameter γ determines the classification accuracy. The smaller parameter C and γ make the decision surface smooth and the model simple, thus the accuracy is reduced. On the contrary, the larger parameter C and γ increase the accuracy of the model and make the model more complex. The common methods of parameter optimization in SVM are grid search method, genetic algorithm, and particle swarm optimization algorithm, the latter two of which are heuristic algorithms. The grid search method can find the optimal solution of a given range of parameters, but the speed is slow. The heuristic algorithm has too many parameters, and different parameters have a great impact on the results, which increases the difficulty of using the algorithm, and it may fall into the local optimal solution. Grid search method [24] provides different combinations of C and γ values for RBF-SVM in an exhaustive way and calculates the classification accuracy of each combination by cross-validation method. Finally, it selects the combination with the highest accuracy to ensure RBF-SVM has the highest accuracy. In the division of the grid, the parameter interval can be enlarged to make a tradeoff between the local optimization and the computational efficiency.

As Figure 8 shows, 5-fold cross-validation is used to find the best combination of values C and γ . The classification accuracy of each combination is shown in the figure by different colors, color red expresses the high accuracy and color blue expresses the low accuracy. The abscissa of the figure represents the value of γ and the ordinate represents the value of C . Finally, the combination with the highest classification accuracy is selected, the value of C is 10^6 , and the value of γ is 10^{-4} .

According to the classification result and confusion matrix, the RBF-SVM algorithms have a good accuracy and recall rate, and it can be used to diagnosis metal roof fault correctly. The fault of metal roof can be divided into longitudinal deformation, laser ranging sensor fault strain sensor fault, and bolt looseness. With this method, it can not only distinguish the metal roof fault accurately but also effectively distinguish the error data caused by sensor fault and hidden fault like bolt looseness. The metal roof real-time monitoring system and the fault diagnosis algorithm that were proposed in this paper make the metal roof monitoring more intelligent and reduce the security risk of metal roof effectively.

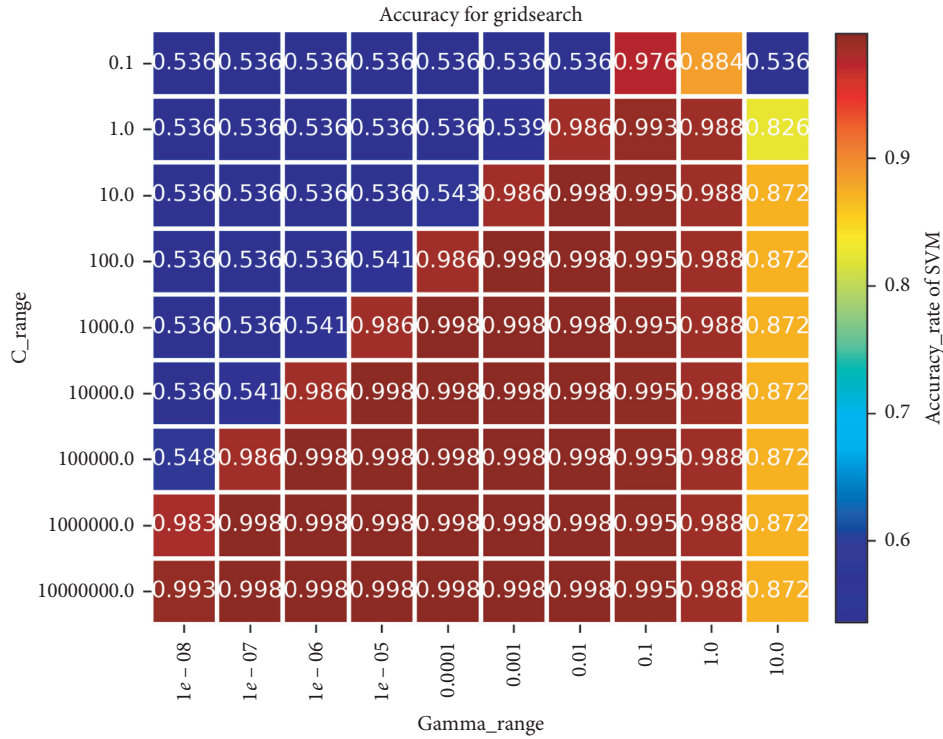


FIGURE 8: Grid search method parameter optimization. After parameter optimization, classification result and confusion matrix of RBF-SVM algorithm can be obtained as shown in Tables 6 and 7.

TABLE 6: Fault diagnosis result of RBF-SVM.

Classification	Accuracy P	Recall rate R	F1-score	Data number
Longitudinal deformation	1.0	1.0	1	24
Laser ranging sensor fault	1.0	1.0	1	16
Strain gage sensor fault	0.98	0.88	0.93	16
Normal data	1.0	1.0	0.93	99
Bolt looseness	1.0	1.0	1.0	23
Average/sum	0.99	0.99	0.99	178

TABLE 7: Confusion matrix of RBF-SVM.

Classification	Longitudinal deformation	Laser ranging sensor fault	Strain gage sensor fault	Normal data	Bolt looseness
Longitudinal deformation	24	0	0	0	0
Laser ranging sensor fault	0	16	0	0	0
Strain gage sensor fault	0	0	14	2	0
Normal data	0	0	0	99	0
Bolt looseness	0	0	0	0	23

5. Conclusion

In order to reduce the safety hazards of widely used metal roof enclosure system, this paper introduces a metal roof real-time monitoring system and fault diagnosis algorithm. Based on analysis of causes and types of faults in the metal roof, the distributed multisource heterogeneous sensors with wireless network are used to monitor the state of the metal roof. It makes the monitoring and maintenance of metal roof change from manual inspection to real-time monitoring, which improves accuracy and reduces security risks.

Compared with traditional monitoring system, this system uses the distributed network to collect data from metal roofs with wide span and large area; meanwhile, it uses multi-source heterogeneous sensors to ensure the accuracy of monitoring data and find the hidden fault of metal roof, and Zigbee wireless network can reduce the damage of roof and make it easy to install. In fault diagnosis algorithm, not only the trend information of data is considered but also the correlation information is used to improve the accuracy of fault diagnosis and find the hidden fault of metal roof. In the process of optimizing the RBF-SVM algorithm, grid search

method is used to improve the classification accuracy. After the verification by experimental data, the RBF-SVM algorithm we used can achieve the 99% of classification accuracy and recall rate. In the test dataset, 176 of 178 sets of data are correctly classified, which have a high classification accuracy. The research on metal roof real-time monitoring system and fault diagnosis algorithm will promote the development of intelligent monitoring and maintenance of metal roof and reduce the safety risk of it effectively.

Data Availability

The data used 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.

References

- [1] L. Yang, "Inspection and reconstruction of metal-roof deformation under wind pressure based on bend sensors," *Sensors*, vol. 17, no. 5, 2017.
- [2] J. Liu, Z. Cui, and J. Li, "Research status and prospect of wind resistance performance of metal roofing with standing seam," *Structure Science*, vol. 34, no. 5, pp. 118–124, 2018.
- [3] Y. Xu and Z. Xie, "Research progress on wind load characteristics and wind resistance capacity of long span metal roof," *Journal of Building Structure*, vol. 3, pp. 41–49, 2019.
- [4] W. Hu and Y. Wu, "The main problems and analysis of metal roof in engineering application in China," *Industrial Building*, vol. 43, no. 6, pp. 150–156, 2013.
- [5] X. Li and W. Zhang, "Design of prognostic and health management structure for UAV system," *International Conference on Systems Engineering*, vol. 1, pp. 12–16, 2011.
- [6] C. Xu, "Research on basic theory and key technology of wireless sensor network for health monitoring of large-scale engineering," Thesis, Wuhan University of Technology, Wuhan, China, 2007.
- [7] T. Zhang, D. Wang, J. Cao, Y. Q. Ni, L.-J. Chen, and D. Chen, "Elevator-assisted sensor data collection for structural health monitoring," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1555–1568, 2012.
- [8] V. J. Hodge, S. O'Keefe, M. Weeks, and A. Moulds, "Wireless sensor networks for condition monitoring in the railway industry: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1088–1106, 2015.
- [9] S. Choi, B. Song, R. Ha, and H. Cha, "Energy-aware pipeline monitoring system using piezoelectric sensor," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1695–1702, 2012.
- [10] S. S. Sani, H. N. Rad, and M. N. Rad, "A new inexpensive system for SHM of bridge decks using wireless sensor networks based on measurements of temperature and humidity," in *Proceedings of the International Conference on Knowledge-based Engineering & Innovation*, Tehran, Iran, November 2016.
- [11] Z. G. Yan, Q. Yue, and Z. Shi, "Design of structural health monitoring system for Hutong changjiang river bridge," *Bridge Construction*, vol. 47, no. 4, pp. 7–12, 2017.
- [12] Y. Sun, W. Meng, C. Li, N. Zhao, K. Zhao, and N. Zhang, "Human localization using multi-source heterogeneous data in indoor environments," *IEEE Access*, vol. 5, pp. 812–822, 2017.
- [13] S. Alampalli, S. Alampalli, and M. Ettouney, "Big data and high-performance analytics in structural health monitoring for bridge management," in *Proceedings of the SPIE Smart Structures/NDE Conference*, Las Vegas, NV, USA, March 2016.
- [14] K. Sekuła and P. Kołakowski, "Piezo-based weigh-in-motion system for the railway transport," *Structural Control & Health Monitoring*, vol. 19, no. 2, pp. 199–215, 2012.
- [15] J. S. Lee, Y. W. Su, and C. C. Shen, "A comparative study of wireless protocols: bluetooth, UWB, ZigBee, and wi-fi," in *Proceedings of the Conference of the IEEE Industrial Electronics Society*, Florence, Italy, October 2008.
- [16] A. R. M. Rao and K. Lakshmi, "Damage diagnostic technique combining POD with time-frequency analysis and dynamic quantum PSO," *Meccanica*, vol. 50, no. 6, pp. 1551–1578, 2015.
- [17] R. P. Bandara, T. H. Chan, and D. P. Thambiratnam, "Structural damage detection method using frequency response functions," *Structural Health Monitoring: An International Journal*, vol. 13, no. 4, pp. 418–429, 2014.
- [18] S. K. Panigrahi, S. Chakraverty, and B. K. Mishra, "Vibration based damage detection in a uniform strength beam using genetic algorithm," *Meccanica*, vol. 44, no. 6, pp. 697–710, 2009.
- [19] L. Yu and J. C. Lin, "Cloud computing-based time series analysis for structural damage detection," *Journal of Engineering Mechanics*, vol. 143, no. 1, 2017.
- [20] Q. Zhou, H. Zhou, Q. Zhou, F. Yang, L. Luo, and T. Li, "Structural damage detection based on posteriori probability support vector machine and Dempster-Shafer evidence theory," *Applied Soft Computing*, vol. 36, pp. 368–374, 2015.
- [21] S. Arangio and F. Bontempi, "Structural health monitoring of a cable-stayed bridge with Bayesian neural networks," *Structure and Infrastructure Engineering*, vol. 11, no. 4, pp. 575–587, 2015.
- [22] X. Li, W. Yu, and S. Villegas, "Structural health monitoring of building structures with online data mining methods," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1291–1300, 2016.
- [23] L. Li, "Research on wind resistance of metal roof system in large span stadium," Thesis, Tianjin University, Tianjin, China, 2014.
- [24] C. Liu, S. Q. Yin, M. Zhang, Y. Zeng, and J. Y. Liu, "An improved grid search algorithm for parameters optimization on SVM," *Applied Mechanics and Materials*, vol. 644–650, pp. 2216–2219, 2014.

Research Article

Impedance Model-Based Optimal Regulation on Force and Position of Bimanual Robots to Hold an Object

Darong Huang,¹ Hong Zhan,¹ and Chenguang Yang ²

¹Key Laboratory of Autonomous Systems and Networked Control, School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China

²Bristol Robotics Laboratory, University of the West of England, Bristol BS16 1QY, London, UK

Correspondence should be addressed to Chenguang Yang; cyang@ieee.org

Received 24 July 2020; Revised 3 September 2020; Accepted 24 October 2020; Published 17 November 2020

Academic Editor: Juan Sandoval

Copyright © 2020 Darong Huang 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.

Bimanual robots have been studied for decades and regulation on internal force of the being held object by two manipulators becomes a research interest in recent years. In this paper, based on impedance model, a method to obtain the optimal target position for bimanual robots to hold an object is proposed. We introduce a cost function combining the errors of the force and the position and manage to minimize its value to gain the optimal coordinates for the robot end effectors (EE). To implement this method, two necessary algorithms are presented, which are the closed-loop inverse kinematics (CLIK) method to work out joint positions from desired EE pose and the generalized-momentum-based external force observer to measure the subjected force acting on the EE so as to properly compensate for the joint torques. To verify the effectiveness, practicality, and adaptivity of the proposed scheme, in the simulation, a bimanual robot system with three degrees of freedom (DOF) in every manipulator was constructed and employed to hold an object, where the results are satisfactory.

1. Introduction

Robots were invented mainly to help human being to carry out tasks, reducing the intensity of human labour, for example, grasping, holding, or moving an object. In such a process, we need robot to keep the object fixed on the end effector (EE) in a proper manner and move along the desired trajectory. Researchers have made effort to perfect these movements in many specific aspects. For instance, in [1], the optimal grasp rectangle for determining the correct position to pick up an object is obtained. Precision of robotic grasping is considered in [2], using probabilistic postgrasp displacement estimation method. Inspired by the scheme of human grasping, [3] presents a controller with tactile sensing and the grasp force can be adjusted as needed; for example, the object slips or not. In [4], the object with dynamics parameters and deformable shape is considered and the grasping forces can be regulated based on a similar tactile framework. The application scenarios of these motions vary from daily life (cloth grasping or towel folding [5])

to industry (plate and belt assembling [6]), military [7], and so on. The literature also demonstrates the fact of a trend of research on holding a target object by robots.

The above-mentioned implementations are related to managing to make an object “stick” to EE of one manipulator. These methods are good, but not better than multirobot system (MRS) to some extent [8]. Compared to a single manipulator, MRS is more complex but competent due to its higher redundancy, larger loading capability, and stronger robustness [9]. It has also been employed in many fields, for example, welding [10], assembly [11], and source seeking [12]. Generally, thanks to the aforementioned advantages, MRS is more prominent to handle complicated manipulations, such as curving and screwing. In the domain of MRS, the bimanual robotic system [6, 13] or the humanoid (dual-arm) robotic system [14, 15] has drawn a lot of attention from scientists and specialists.

In the specific domain of the bimanual robot system, the collaboration of the two manipulators becomes one of the research hotspots in the past decades [16–18]. The regulation

of the internal force of a held object or the two robotic manipulators is a pronounced tendency and plenty of related methods appear. An adaptive global neural network control law for bimanual robots system is reported in [13] to simultaneously guarantee the internal forces to converge within a small neighbourhood of the desired values. Based on relative impedance models, in bimanual robots teleoperation system, [19] improves the internal force control performance by deducing the desired position; [20] implements the proposed method on the two manipulator systems to carry out hold-and-move task also in the teleoperation field and the authors separately handle the orientation, position, and force regulation problem. A symmetric strategy is developed in [21] for redundant dual-arm robots to regulate both arm's pose and force coordination. In [22], a hybrid force/position controller is designed based on a fuzzy observer and a two-arm robot is applied to demonstrate the effectiveness in simulation. Researchers have done a great deal of work on achieving either better force control or the minimum position/orientation tracking errors. However, when they are in pursuit of a nice control performance on one side (e.g., force), they may sacrifice the performance on the other side (e.g., position/orientation).

Sometimes, we need a compromise between the behaviours of the force and the position; for example, when two robots are commanded to manipulate a deformable object from one place to another, they are informed to hold the object either with a desired internal force or along a pre-designed trajectory (EE poses). However, few works take both of them into consideration in bimanual robot manipulation but just take one of the many items into account; for example, [23] focuses on only the position item. Hence, in this paper, we bend our effort for finding a suitable method to determine the optimal compromise when two robots cooperate in holding an object. Thanks are due to the linear-quadratic approaches presented in [24, 25], which, in a geometric way, explain the optimal intersection point of the force and the position in impedance model-based control. It gives us the inspiration to construct a similar cost function to estimate the control performance containing the force and the position tracking in the bimanual manipulation task. Methods to obtain the optimal solution to such kind of cost function in impedance model-based linear/nonlinear system can be found in the literature; for example, [26] seeks help from iterative adaptive dynamic programming algorithm, and [27, 28] solve it by constructing the Hamilton-Jacobi-Bellman equation (HJB) while a neural network-based computational method is provided in [29] to handle the algebraic Riccati equation (ARE) that usually accompanies the HJB. The techniques in [29] make the solution to the HJB accessible, which is confirmed by researchers who apply the method into robot-environment interaction [30]. Inspired by these papers, in this paper, we construct a suitable performance measure to fit the bimanual robots' collaboration situation in order to gain an optimal regulation for force and position.

To facilitate the proposed method, we still need two auxiliary techniques to transfer the Cartesian tasks into robot joint space control problems. First, environmental

forces surely affect the tracking performance of a robot; therefore, before we compensate the contact force acting on each EE, we need to measure how heavy the contact force is at the current time. In this respect, [31] introduces several ideas that we can turn to; for example, the technique regarding generalized momentum [32] can be an appropriate choice and is adopted by many researchers to design force observer without installing sensors on EE [33–36]. Second, a specific coordinate in the task space is the target pose we need to command our EE to move to, but we can only indirectly control the joint actuator, not directly the EE. Hence, position mapping from the task space to the joint space should be calculated. In this regard, the closed-loop inverse kinematics (CLIK), which has been studied for decades [37–40], is a sophisticated algorithm to obtain joint positions in real time and becomes our option in this paper. In the latter sections, we are going to introduce these two supplementary approaches to make our proposed bimanual robotic system complete.

The contributions of this paper lie in the following:

- (i) This paper constructs a relatively complete bimanual robot simulation system, which contains two 3-degree-of-freedom (DOF) robots (different from those simple 2-DOF robots employed in many papers) and provides basic and practical techniques to make this system function well when it carries out tasks. These techniques include inverse kinematics algorithm CLIK and force observer that plays the role of a force sensor installed on the EE. Such a basic platform, with multiple robots and basic control algorithms, provides possibilities for the researcher to conduct simulation within a larger range of application scenarios.
- (ii) We take both the force and the position regulation into consideration in the scenario that bimanual robots cooperate to hold an object. Despite the fact that we might be not able to minimize both tracking errors of the two factors (force and position), we make the sum of the two closed to the minimum values by modelling the object as an impedance model and constructing proper cost function to suit the solving procedure. And to work out the optimal solution, a model-free computational algorithm is introduced.

The structure of the paper is organized as follow: Section 2 (Methods) develops the proposed method, including the restatement on the purpose of this paper, some preliminary knowledge, and the introduction of the force observer and the CLIK; right after Methods comes Section 3 (Simulations and Analysis) that involves the simulation configuration, procedures, results, and analysis; the final section (Conclusions) summarizes the whole paper.

2. Methods

In this section, we are going to (1) explain the optimal target (namely, the significance) of minimizing the sum of the tracking errors of both the force and the position, (2)

develop the adaptive optimal method on regulation on force and position of bimanual robots in holding an object, and (3) present the preliminary techniques (force observer and CLIK) that contribute to the bimanual manipulating system. A brief introduction of the whole control system of the bimanual robots is depicted in Figure 1, which will be explained gradually in the following subsections.

2.1. Geometrical Interpretation on Minimizing the Sum of the Errors. When robots interact with physical objects in the form of an impedance model, it is usually commanded to “touch” them with its EE moving to a certain position or approaching the object’s centre point until it senses a desired counterforce. In these cases, a single error of either force or position is considered and can be properly minimized. However, in many other cases, we need the tracking of both the force and the position to reach a relatively good performance rather than just one of them. And the following are the explanation of what the relatively “good” point (the optimal target) is.

Let us consider a deformable object’s physical model to be

$$f = g(\ddot{x}, \dot{x}, x), \quad (1)$$

where f denotes the subjected force, x , \dot{x} , and \ddot{x} represent the deformation, deforming velocity, and deforming acceleration of the object, respectively, and $g(\cdot)$ is the mapping with regard to the object’s impedance model.

In implementation, we desire that the object subjects to a certain force f_d with a certain deformation state x_d , but we do not know the exact parameters in $g(\cdot)$. Hence, in most cases, $f_d \neq g(\ddot{x}, \dot{x}, x_d)$. Therefore, referring to [24, 25], we can obtain an optimal point $(f_{\text{opt}}, x_{\text{opt}})$ that is the nearest to the desired set (f_d, x_d) and satisfies the condition $f_{\text{opt}} = g(\ddot{x}, \dot{x}, x_{\text{opt}})$. It can be explained geometrically via Figure 2 [25].

Note 1. If we take \dot{x} or \ddot{x} into consideration, it will turn out to be a three- or four-dimensional optimization problem.

Only in such an optimal point $(f_{\text{opt}}, x_{\text{opt}})$, can the sum of tracking errors of the force and the deformation be the minimum; namely,

$$(f_{\text{opt}} - f_d)^2 + (x_{\text{opt}} - x_d)^2 = \min\{(f - f_d)^2 + (x - x_d)^2\}. \quad (2)$$

If we add weights W_f and W_x to the tracking errors in equation (2), it turns out to be

$$W_f(f_{\text{opt}} - f_d)^2 + W_x(x_{\text{opt}} - x_d)^2 = \min\{W_f(f - f_d)^2 + W_x(x - x_d)^2\}. \quad (3)$$

In the case of bimanual robots holding an object, f becomes the object’s internal force and x equals the distance between the EEs of the two robotic arms. We can adjust weights W_f and W_x to achieve different tracking performance. Particularly, if $W_f = 0$ and $W_x \neq 0$, that is, pure

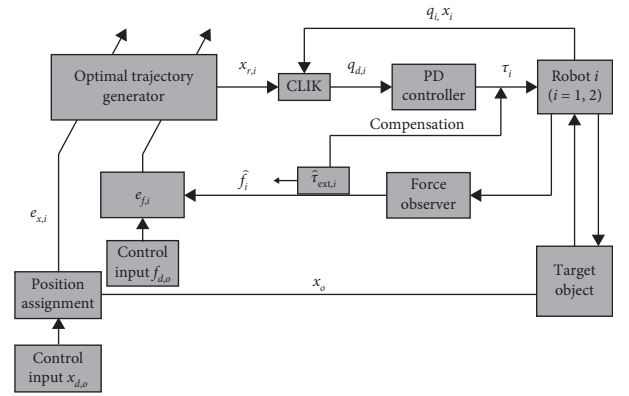


FIGURE 1: The whole system structure of the bimanual robot manipulation.

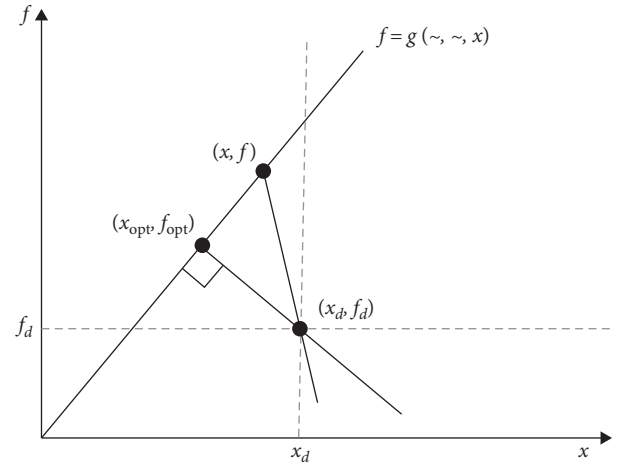


FIGURE 2: A 2D graph to explain the geometrical optimal point of the force and the deformation, which locates at the intersection between the line $f = g(\sim, \sim, x)$ and the line perpendicular to it and passing through (f_d, x_d) .

position control, while $W_f \neq 0$ with $W_x = 0$ represents pure force control.

Considering the 3D space of the task space, f and x denote generalize force (force and torque) and position (position and orientation) with 6 dimensions in each; the performance measure can be written in a quadratic form as

$$\Gamma = e_f^T W_f e_f + e_x^T W_x e_x, \quad (4)$$

where $e_f = f - f_d$ and $e_x = x - x_d$ are tracking errors of force and position.

If we need to guarantee the tracking performance within a certain time interval $[t_a, t_b]$, equation (4) needs to be modified into equation (5):

$$\Gamma = \int_{t_a}^{t_b} (e_f^T W_f e_f + e_x^T W_x e_x) dt. \quad (5)$$

By minimizing equation (5), we can obtain the optimal performance from t_a to t_b .

2.2. Adaptive Optimal Regulation on Force and Position. In object holding tasks, as we do not move the object, we ignore the acceleration information. Let us consider the object to be held as a damping-spring model whose expression is written as

$$-f_o = C_o \dot{x}_o + K_o x_o, \quad (6)$$

which is a specific situation of equation (1) with x_o and \dot{x}_o denoting the object's deformation and its deforming velocity and f_o standing for its internal force.

In such case with the object being a damping-spring model, considering the preliminary knowledge in the previous subsection, with reference to [30, 34], the cost function is set to be

$$\Gamma_m = \int_0^{t_f} \left[(x_o - x_{d,o})^T W_x (x_o - x_{d,o}) + (\hat{f}_o - f_{d,o})^T W_f (\hat{f}_o - f_{d,o}) \right] dt, \quad (7)$$

where t_f stands for the system running final time; $x_{d,o}$, \hat{f}_o , and $f_{d,o}$ are the user-desired deformation, the estimated subjected force (the real internal force is usually inaccessible and the estimated value is obtained through force observer), and the user-desired object subjected force, respectively; $W_x \geq 0$ and $W_f \geq 0$ denote the weights for trajectory tracking error and force tracking error, respectively. W_x and W_f can be defined according to the tracking demand; for example, if we require precision in force tracking, W_x can be set to closed 0 while $W_f \neq 0$, and likewise, the situation with $W_x \neq 0$ and W_f closed to 0 makes the position tracking more accurate.

The absolute value of subjected force on EE $f_{ext,i}$ (the subscript $i = 1, 2$ is used to distinguish between robot 1 and robot 2) equals the internal force of object f_o in a relatively static condition and the estimation of the subjected force on EE $\hat{f}_{ext,i}$ usually converges to the real value. In this paper and in the application scenarios, we have

$$f_o = |f_{ext,i}| = |\hat{f}_{ext,i}| = \hat{f}_o. \quad (8)$$

In this paper, we want the object internal force to be minimized when it is held by two robots, namely, using a minimum force to hold an object; therefore, we set the desired internal force:

$$f_{d,o} = 0. \quad (9)$$

Then, equation (7) in this paper can be rewritten as

$$\Gamma_m = \int_0^{t_f} \left[(x_o - x_{d,o})^T W_x (x_o - x_{d,o}) + (\hat{f}_{ext,i})^T W_f (\hat{f}_{ext,i}) \right] dt. \quad (10)$$

In order to employ the optimal control [29] in trajectory tracking problem, we define a linear system to determine the $x_{d,o}$ as

$$\begin{aligned} \dot{\eta}_o &= U \eta_o, \\ x_{d,o} &= V \eta_o, \end{aligned} \quad (11)$$

where η_o is treated as a system state; U and V are two matrices to be determined. Then equation (7) can be rewritten as

$$\begin{aligned} \Gamma_m &= \int_0^{t_f} \left\{ \begin{bmatrix} x_o^T & x_{d,o}^T \end{bmatrix} \begin{bmatrix} W_x & -W_x V \\ -V^T W_x & V^T W_x V \end{bmatrix} \begin{bmatrix} x_o \\ x_{d,o} \end{bmatrix} + e_f^T W_f e_f \right\} dt \\ &= \int_0^{t_f} \left\{ \xi^T Q \xi + e_f^T R e_f \right\} dt, \end{aligned} \quad (12)$$

where $\xi = \begin{bmatrix} x_o^T & \eta_o^T \end{bmatrix}^T$, $e_f = \hat{f}_{ext,i} - f_{d,o} = \hat{f}_{ext,i}$, $Q = \begin{bmatrix} W_x & -W_x V \\ -V^T W_x & V^T W_x V \end{bmatrix}$, and $R = W_f$. Let us view ξ as a state in another system, modified from equation (6), whose state-space form is

$$\dot{\xi} = A \xi + B e_f, \quad (13)$$

where $A = \begin{bmatrix} -C_o^{-1} K_o & 0 \\ 0 & U \end{bmatrix}$ and $B = \begin{bmatrix} -C_o^{-1} \\ 0 \end{bmatrix}$.

In this system, e_f is treated as the system input. Through linear optimal control theory [41] and the proven model-free adaptive dynamic programming (ADP) algorithm in [29], an optimal control law

$$e_f = -K_{opt} \xi, \quad (14)$$

can be acquired such that the performance index (equation (12)) is minimized without knowing A or B . The key point is to obtain the feedback gain matrix K ($\rightarrow K_{opt}$), which can be computed via the following steps (refined and depicted in Figure 3).

Step 1. Take $e_f = -K_0 \xi + \nu$ as the input on the time interval $[t_0, t_l]$, in which K_0 is stabilizing and ν is the exploration noise to meet the persistent excitation (PE) condition. Compute $\delta_{\xi,\xi}$, $I_{\xi,\xi}$, and I_{ξ,e_f} through

$$\begin{aligned} \delta_{\xi,\xi} &= \left[\bar{\xi}(t_1) - \bar{\xi}(t_0), \bar{\xi}(t_2) - \bar{\xi}(t_1), \dots, \bar{\xi}(t_l) - \bar{\xi}(t_{l-1}) \right]^T, \\ I_{\xi,\xi} &= \left[\int_{t_0}^{t_1} \xi \otimes \xi dt, \int_{t_1}^{t_2} \xi \otimes \xi dt, \dots, \int_{t_{l-1}}^{t_l} \xi \otimes \xi dt \right]^T, \\ I_{\xi,e_f} &= \left[\int_{t_0}^{t_1} \xi \otimes e_f dt, \int_{t_1}^{t_2} \xi \otimes e_f dt, \dots, \int_{t_{l-1}}^{t_l} \xi \otimes e_f dt \right]^T, \end{aligned} \quad (15)$$

with “ \otimes ” denotes the Kronecker product and

$$\bar{\xi} = \left[\xi_1^2, \xi_1 \xi_2, \dots, \xi_1 \xi_p, \xi_2^2, \xi_2 \xi_3, \dots, \xi_2 \xi_p, \dots, \xi_{p-1}^2, \xi_{p-1} \xi_p, \xi_p^2 \right]^T, \quad (16)$$

until the following condition is satisfied:

$$\text{rank} \left(\begin{bmatrix} I_{\xi,\xi} & I_{\xi,e_f} \end{bmatrix} \right) = \frac{p(p+1)}{2} + pr, \quad (17)$$

where p is the number of elements in ξ and r is the number of elements in e_f .

Step 2. Solve P_k and K_{k+1} according to

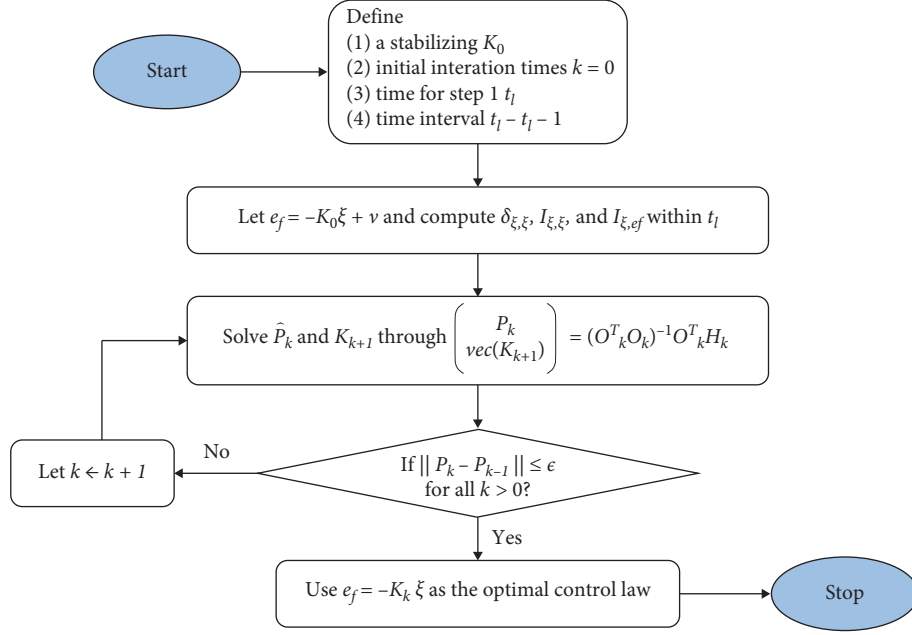


FIGURE 3: The flowchart of the procedures for obtaining the approximate optimal control law.

$$\begin{bmatrix} \hat{P}_k \\ \text{vec}(K_{k+1}) \end{bmatrix} = (\Theta_k^T \Theta_k)^{-1} \Theta_k^T \Xi_k, \quad (18)$$

where “vec” is the operator that “stretches” a matrix to a vector; that is, if $X = [c_1, c_2, c_3]$ with $c_1, c_2,$ and c_3 are column vectors, then $\text{vec}(X) = [c_1^T, c_2^T, c_3^T]^T$, and

$$\begin{aligned} \hat{P}_k &= [P_{1,1}, 2P_{1,2}, \dots, 2P_{1,p}, P_{2,2}, 2P_{2,3}, \dots, 2P_{2,p}, \\ &\quad \dots, P_{p-1,p-1}, 2P_{p-1,p}, P_{p,p}]^T, \\ \Theta_k &= [\delta_{\xi,\xi}, -2I_{\xi,\xi}(I_p \otimes K_k^T R) - 2I_{\xi,ef}(I_p \otimes R)], \\ \Xi_k &= -I_{\xi,\xi} \text{vec}(Q_k), \end{aligned} \quad (19)$$

where I_p is a unit matrix with p dimensions, $P_{i,j}$ ($i, j = 1, 2, 3, \dots, p$) are the elements of P_k , and

$$Q_k = Q + K_k^T R K_k. \quad (20)$$

Note 2. The solution of P_* is the key point to the well-known algebraic Riccati equation (ARE) before deriving the optimal feedback gain matrix K_{opt} [29].

Step 3. Let $k \leftarrow k + 1$ and repeat Step 2 until $\|P_k - P_{k-1}\| \leq \epsilon$, where $\epsilon > 0$ is a predefined constant.

Step 4. Use equation (21) as the approximate optimal control law.

$$e_f = -K_k \xi. \quad (21)$$

Remark 1. It is noticed that the above procedure does not rely on A or B in equation (13), which shows one of its favourable properties that the optimal solution is applicable to that system with unknown dynamics parameters, for

example, an object with unknown mass, damping, or spring parameters. In many application scenarios, these parameters are usually inaccessible.

2.3. Generalized-Momentum-Based Force Observer. As mentioned in equation (7), we need to observe the robot's subjected force to replace the object's internal force. Thus, in this subsection, the target is to construct a force observer to “sense” the force act on the EE without installing sensor hardware on it.

The two-robots dynamics model in the bimanual system can be expressed as

$$\begin{cases} M_1(q_1) \ddot{q}_1 + C_1(q_1, \dot{q}_1) \dot{q}_1 + G_1(q_1) = \tau_1 + \tau_{\text{ext},1}, \\ M_2(q_2) \ddot{q}_2 + C_2(q_2, \dot{q}_2) \dot{q}_2 + G_2(q_2) = \tau_2 + \tau_{\text{ext},2}, \end{cases} \quad (22)$$

where $q_i \in R^{n_i}$ stands for the joint position of robot i ($i = 1, 2$) (n_i is the joint number of robot i); $M_i(q_i) \in R^{n_i \times n_i}$ denotes the inertia matrix of the robot i ; $C_i(q_i, \dot{q}_i) \in R^{n_i \times n_i}$ is the centripetal and Coriolis coupling matrix; $G_i(q_i) \in R^{n_i}$ represents the gravitational force; $\tau_i \in R^{n_i}$ is the control input torque, and $\tau_{\text{ext},i} \in R^{n_i}$ is the torque act on joints caused by the subjected force $f_{\text{ext},i}$ on the EE.

Property 1. $M_i(q_i)$ is a symmetric and positive definite matrix [42].

Property 2. The result of $\dot{M}_i(q_i) - 2C_i(q_i, \dot{q}_i)$ is a skew-symmetric matrix [42].

As it is known [43], the relationship between $\tau_{\text{ext},i}$ and $f_{\text{ext},i}$ is

$$f_{\text{ext},i} = J_i \tau_{\text{ext},i}, \quad (23)$$

where J_i is the Jacobian matrix of robot i . Hence, we can estimate $f_{\text{ext},i}$ indirectly via observing $\tau_{\text{ext},i}$. Referring to the concept of generalized momentum [32] and its applications [33–36], we define p_i as the generalized momentum with

$$p_i = M_i \dot{q}_i. \quad (24)$$

Then, we have

$$\begin{aligned} \dot{p}_i &= M_i \ddot{q}_i + \dot{M}_i \dot{q}_i \\ &= \dot{M}_i \dot{q}_i + \tau_i + \tau_{\text{ext},i} - C_i \dot{q}_i - G_i, \end{aligned} \quad (25)$$

and with properties in [42], the derivative of inertia matrix M with respect to time is

$$\dot{M}_i = C_i + C_i^T. \quad (26)$$

Substituting equation (26) into equation (25), we obtain

$$\begin{aligned} \dot{p}_i &= C_i^T \dot{q}_i + \tau_i - G_i + \tau_{\text{ext},i} \\ &= u_i + \tau_{\text{ext},i}, \end{aligned} \quad (27)$$

where we define $u_i = C_i^T \dot{q}_i + \tau_i - G_i$.

According to [32, 36], the observer for the generalized momentum p_i can be written as

$$\dot{\hat{p}}_i = u_i + L_i (p_i - \hat{p}_i), \quad (28)$$

where

$$\hat{\tau}_{\text{ext},i} = L_i (p_i - \hat{p}_i) \quad (29)$$

is the estimation of $\tau_{\text{ext},i}$. Next, we obtain the estimated value of the subjected force $f_{\text{ext},i}$ with the help of equation (23). That is,

$$\hat{f}_{\text{ext},i} = J_i \hat{\tau}_{\text{ext},i}, \quad (30)$$

which will be used to compare with the desired contact force $f_{d,i} (= f_{d,o} = 0)$.

2.4. EE Pose Assignment and CLIK Algorithm. The basic problem of the bimanual robot system is to drive the robot EEs to first touch and then hold the object. In this aspect, we should have the desired trajectory and the way to convert the trajectory into desired Cartesian space to joint angles.

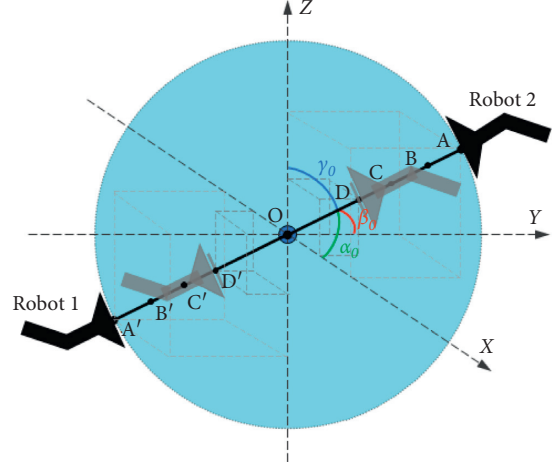
So far, the optimal control input e_f^{opt} is obtained according to equation (21), then, the optimal deformation is derived

$$x_o^{\text{opt}} = -\frac{1}{K_1} \left(e_f^{\text{opt}} + K_2 \eta_o \right), \quad (31)$$

where $[K_1 K_2] = K_k$.

Considering the object being held by two manipulators, we have to transform the optimal deformation x_o^{opt} of the object into certain reference positions $x_{r,i}$ for each EE. To this end, we apply a simple symmetric way (a more complicated way is introduced in [21]) for robots to hold the object. As depicted in Figure 4, the reference robots' EE position $x_{r,i}$ is obtained

$$x_{r,i} = x_o + T_i, \quad (32)$$



AA': two EEs just touch the object.
BB': a deformation (distance) for a desired force.
CC': an optimal deformation x_o^{opt} .
DD': a desired deformation (distance).
O: the target holding centre.

FIGURE 4: A symmetric way to hold the object by two robots, where two EEs approach the object centre symmetrically.

where $x_o = \begin{bmatrix} P_o \\ O_o \end{bmatrix}$ (different from object deformation x_o) is the target holding centre within the object with P_o and O_o denoting the object's position and orientation, respectively, and $T_i = \begin{bmatrix} T_{P_i} \\ T_{O_i} \end{bmatrix}$ is the transformation from the centre to the EE with T_{P_i} and T_{O_i} being the position and the orientation transformation. To be specific,

$$T_{P_i} = \pm \frac{1}{2} x_o^{\text{opt}} \begin{bmatrix} \cos \alpha_o \\ \cos \beta_o \\ \cos \gamma_o \end{bmatrix}, \quad (33)$$

$$T_{O_i} = \begin{cases} 0_{3 \times 1}, & i = 1, \\ -\pi I_{3 \times 1}, & i = 2, \end{cases}$$

where α_o , β_o , and γ_o , as shown in Figure 4, represent the desired orientation of the object and $i = 1$ and 2 represents the robot i on the left ($i = 1$) and the right ($i = 2$) of the object.

Finally, the Cartesian space coordinate $x_{r,i}$ can be reached through CLIK [37–40]. From Figure 1, we can see that we need to derive desired joint position $q_{d,i}$ based on the information of the robot's current state (q_i , and x_i) and the target EE position $x_{r,i}$.

In CLIK, the inverse kinematics problem of a class of serial-link manipulators is treated as a dynamical problem. As we usually use serial-link robots in the bimanual manipulator system, the computation from the joint space velocity to the task space velocity is described as

$$\dot{q}_{d,i} = J_i^\dagger (\dot{x}_{r,i} + k_E e_{x,r,i}), \quad (34)$$

where $e_{x_{r,i}} = x_i - x_{r,i}$ is the EE tracking error.

Then, the target joint position, which is employed as the input of the PD controller in Figure 1, is derived by

$$q_{d,i} = \int_0^{t_c} J_i^\dagger (\dot{x}_{r,i} + k_E e_{x_{r,i}}) dt, \quad (35)$$

where t_c represents the current time.

3. Simulations and Analysis

In this section, we conduct two groups of simulations on MATLAB platform, in which two 3-DOF robots are employed to hold an object. The first group aims to demonstrate the importance of the force observer and the CLIK. The second group mainly shows the significance of the synchronously optimal regulation on force and position in bimanual robots holding tasks. Some basic information on the bimanual robot system is listed in Table 1.

3.1. Simulation Group 1. In the first group of simulation, two cases are conducted. Case 1 is a demo, commanding two robot EEs to collaborate to approach an object before holding it. Case 2 is a comparison between the situations with and without using force observer.

Case 1. In this case, we suppose that there exists an object ($C_o = 0.01$, $K_o = 1$) that (randomly) locates within the intersection of the two robots' workspace, shown in Figure 5. From 0 s to 2 s, we generate two paths for both robots, and they move their EEs from the initial positions to the holding target positions through the derived desired joint position $q_{d,i}$ obtained via CLIK. From 2 s to 6.5 s, the two robots are commanded to hold the object according to the desired object desired deformation trajectory $x_o = 0.2(1 - e^{-(t-2)})$, where $U = -1$ and $V = 0.2$. Finally, from 6.5 s to 10 s, based on the aforementioned pose assignment method and according to the desired object poses, new EE trajectories are generated. And as robots move along the two paths, the object also turns along the desired poses. Some sample frames are posed in Figure 5, and some changing variables as time goes by are depicted in Figure 6.

Based on the results, it is not hard to tell that (1) the CLIK algorithm plays an important role in generating the $q_{d,i}$, enabling EEs to move precisely along the desired path so as to enable the object to deform and turn along the desired trajectory (Figures 6(e)–6(f)); (2) the force observer performs well in estimating the subjected forces acting on EEs such that the estimated object internal force is worked out (Figures 6(b) and 6(c)).

Case 2. In this case, we demonstrate the importance of the developed force observer by comparing the deformation tracking errors (namely, the tracking performance of EEs) in the conditions with and without using force observer. We place the object in the centre between the initial position of EEs, seen in Figure 7, and then command the two robots to hold the object for a while along the desired trajectory $x_{d,o} = 0.2(1 - e^{-t})$. The object's deformation information

TABLE 1: Partial parameters of the bimanual robot system.

Items	Values
Robot 1 link length	$l11, l12, l13 = 0.5, 0.3, 0.1$ (m)
Robot 2 link length	$l21, l22, l23 = 0.5, 0.3, 0.1$ (m)
Robot 1 link mass	$m11, m12, m13 = 1.5, 1.0, 0.5$ (kg)
Robot 2 link mass	$m21, m22, m23 = 1.5, 1.0, 0.5$ (kg)
Robot 1 initial joint position	$[(2/3)\pi, -(1/3)\pi, -(1/3)\pi]^T$ (rad)
Robot 2 initial joint position	$[(1/3)\pi, (1/3)\pi, (1/3)\pi]^T$ (rad)
Robot 1 base coordinate	$[0, 0]$ (m)
Robot 2 base coordinate	$[0.5, 0]$ (m)
Gravitational acceleration	$g = 0$
Sample time	10^{-4} (s)
PD controller gains	$Kp = 10^2 \text{diag} [40, 35, 28]$ $Kd = \text{diag} [10, 1, 0.01]$

collected from the two conditions is illustrated in Figure 8. It is apparent that, under the circumstance without employing force observer, the subjected force is not sensed, and thus, the compensation for the joint torque is 0 resulting in that the PD controller is always not able to address the EE position steady-state tracking error. We can also see larger steady-state errors as the external force goes up when using the same holding path but different object impedance parameters ($K_o = 1, 5, 10$) of the object. We can conclude that, without force observer, the steady-state errors stay larger as the external force becomes heavier.

Therefore, we can tell that, in the task of bimanual robots holding an object, the compensation for the contact force is of great importance. In other words, the force observer acts like a sensor installed on the EE and senses the contact force before informing the system to make up for the joint control torques so as to enable EEs to precisely move along the desired paths. In order to demonstrate the effectiveness of the force observer in estimating the subjected external torques and forces, some other pieces of information corresponding to the simulation of Figure 8(g) are displayed in Figure 9. Figures 9(a)–9(d) show how the estimated joint torques (derived from equation (29)) and the estimated external forces (calculated from equation (30)) are closed to the real joint torques and the real subjected external forces, respectively, and Figure 9(e) illustrates slight error between the object's internal force and the resultant force of $\hat{f}_{\text{ext},2,x}$ and $\hat{f}_{\text{ext},2,y}$ (namely, the estimated object's internal force).

3.2. Simulation Group 2. In the second group of simulation, we also have two cases to illustrate how the impedance model-based optimal regulation method works. Case 3 shows a comparison between the optimal solutions obtained via the paper-introduced algorithm without knowing model parameters (A and B in equation (13)) and the solutions by solving algebraic Riccati equation (ARE) based on known A and B . Case 4 demonstrates how effectively and adaptively the optimal method can regulate the force and position under various conditions.

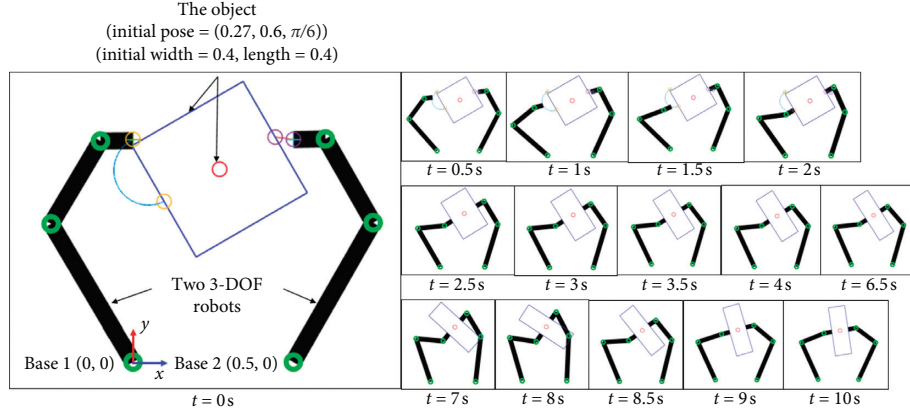


FIGURE 5: Some frames captured in the task described in Case 1.

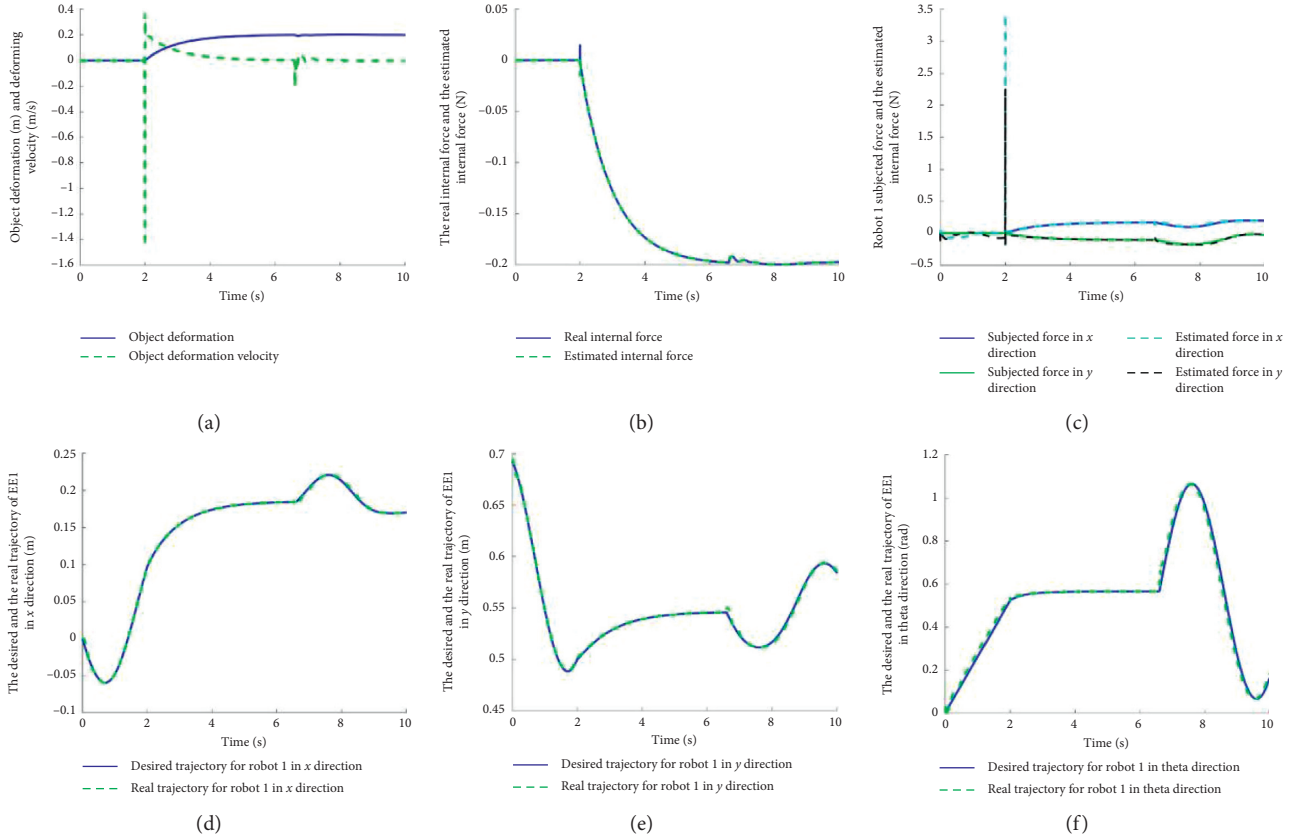


FIGURE 6: Some changing variables in Case 1, including object deforming information, internal force, estimated forces, and EE trajectories.

Case 3. In this case, the object impedance model is firstly set as $-f_o = 0.01\dot{x}_o + 1x_o$, and it is placed on the pose depicted in Figure 7; the other variables are $W_x = 1$, $W_f = 1$, $U = -1$, $V = 0.2$, $\varepsilon = 0.03$, $t_l - t_{l-1} = 0.03$, $K_0 = [-1, 1]$, and $\nu = \sum_{i=1}^8 (0.001/i)\sin(it)$. By exactly knowing these items, the optimal solution $K_{\text{opt}} = [-0.4142, 0.1404]$ is obtained through solving ARE, while the $K_k = [-0.4187, 0.1324]$ is also gained without knowing the information of A and B , through the aforementioned procedures (from Step 1 to Step 4). We can tell that K_k is very closed to K_{opt} , and the control

performance based on K_k is also satisfactory according to the cost function results depicted in Figure 10(b), despite a gap between them, which does not affect the effectiveness of K_k since the gap increment does not significantly tend to become larger after 2.5 s. This gap is mainly accumulated within 0 s–2.5 s, because (i) when executing Step 1 (0 s–0.15 s), a not very proper K_0 and the exploration noise ν are used; and, referring to Figure 10(a), (ii) to avoid a sudden big change between x_o (obtained through K_0 in Step 1) and x_o^{opt} (obtained through K_k via equation (31)), from 0.15 s on,

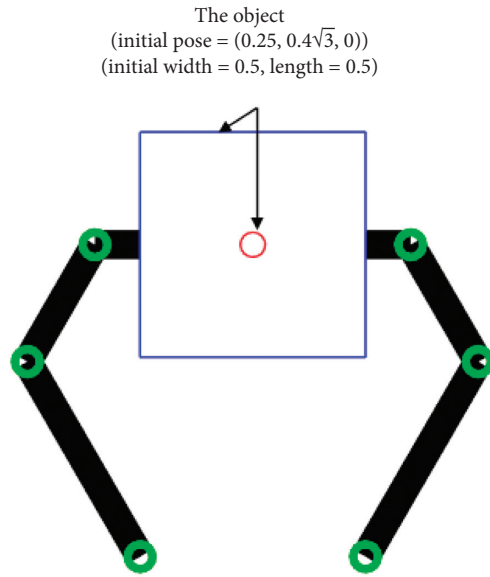


FIGURE 7: The initial state of the system in Cases 2–4.

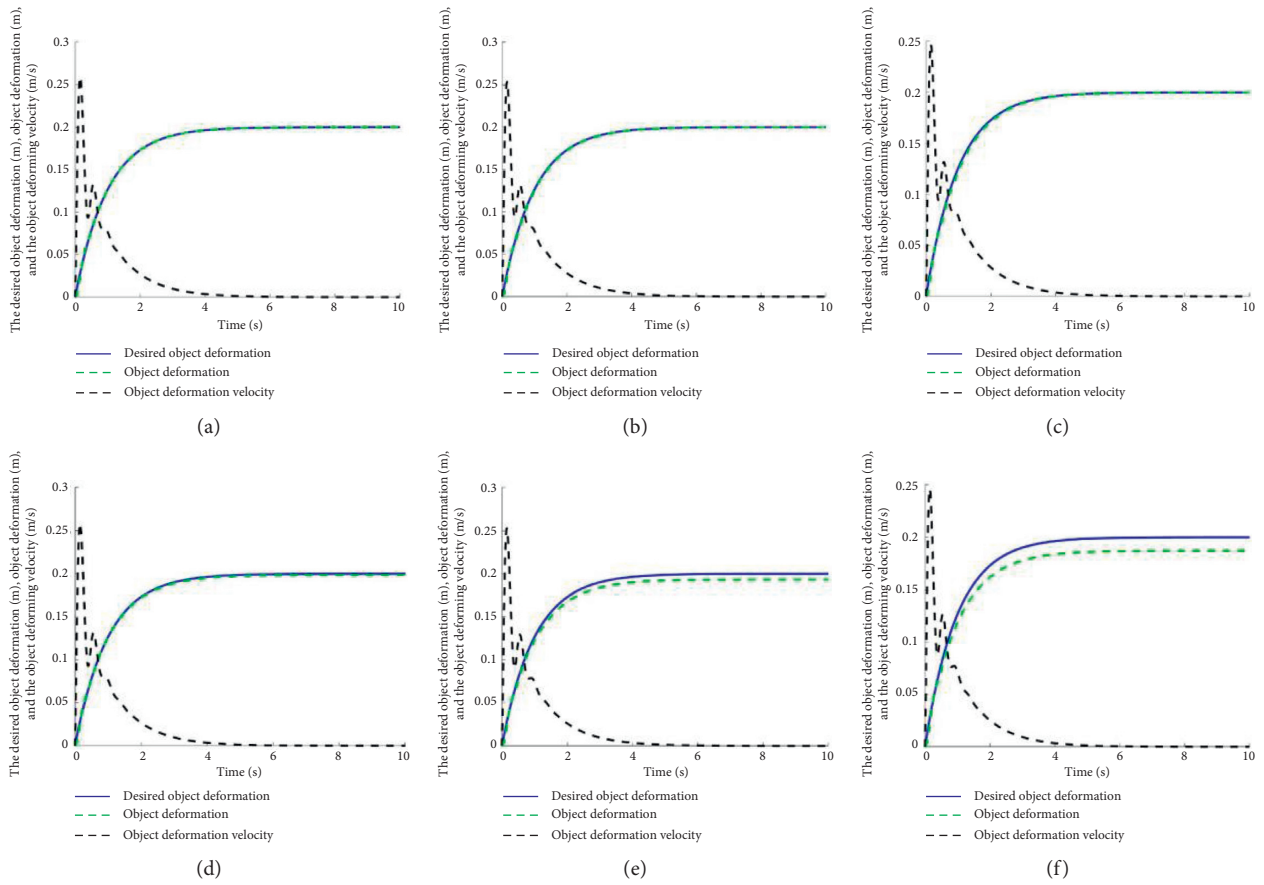


FIGURE 8: Comparison of deformation information of the object between the conditions with and without using force observer. (a) $K_o = 1$ with force observer, (b) $K_o = 5$ with force observer, (c) $K_o = 10$ with force observer, (e) $K_o = 1$ without force observer, (f) $K_o = 5$ without force observer, and (g) $K_o = 10$ without force observer.

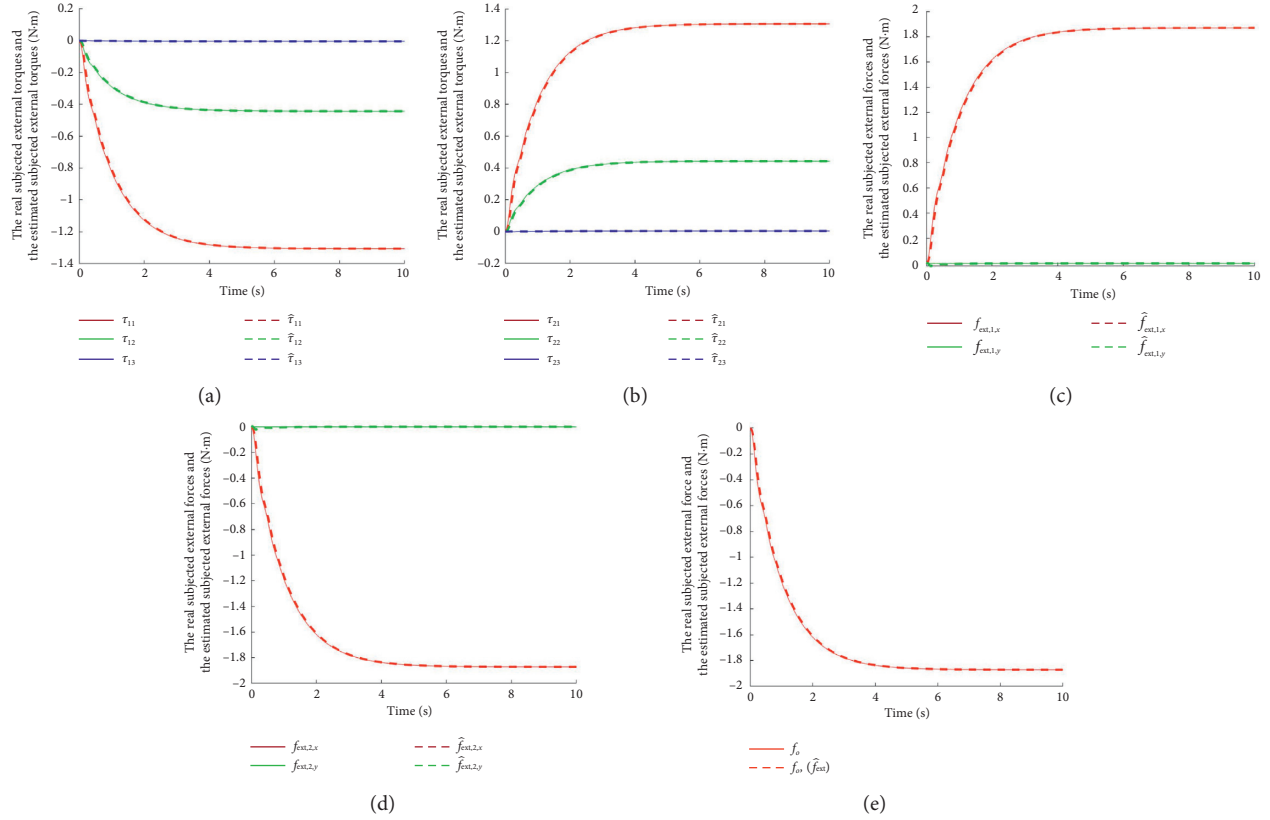


FIGURE 9: Some other system states corresponding to the simulation of Figure 8(g). (a) Subjected joint torques in robot 1. (b) Subjected joint torques in robot 2. (c) Subjected external force in EE 1. (d) Subjected external force in EE 2. (e) Object's internal force.

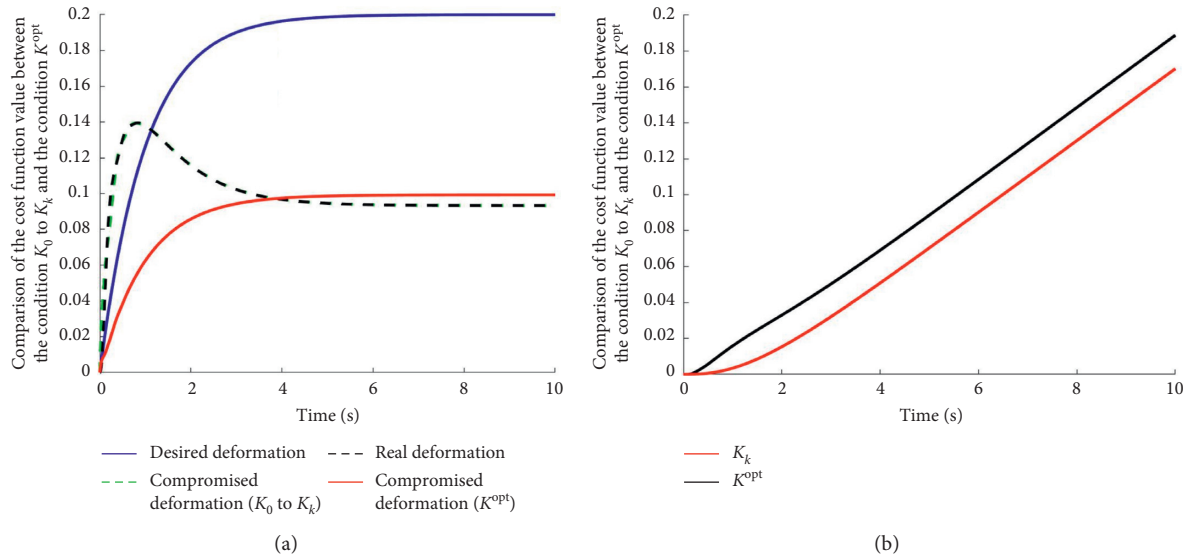
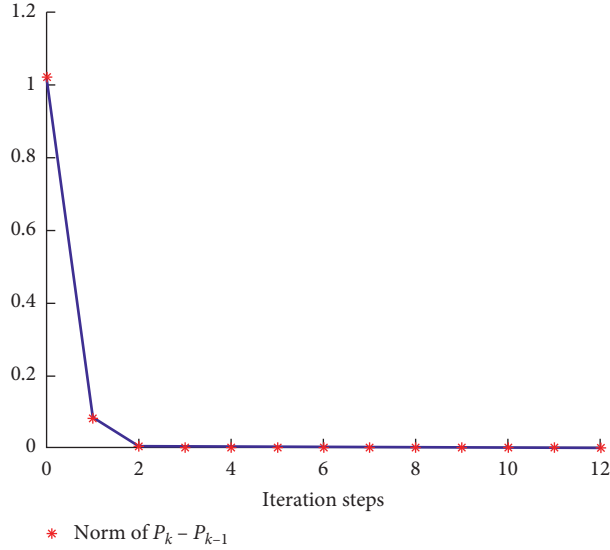


FIGURE 10: (a) The comparison of the object deformation information between different K^* values and (b) the comparison of the cost between different K^* values.

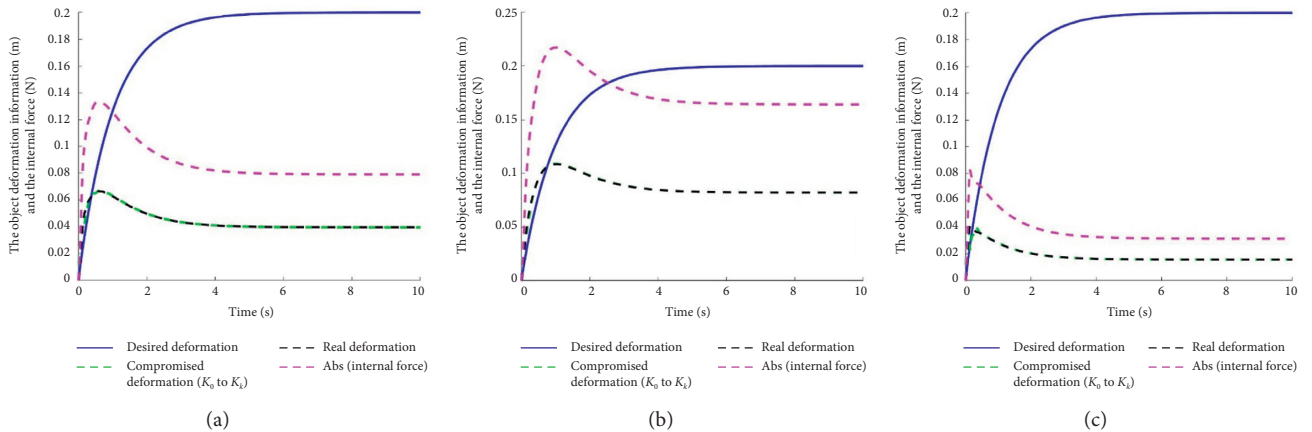
x_o^{opt} is not directly utilized. From 0.15 s on, the modified deformation trajectory is calculated through

$$x_o = \text{decay} * x_o + (1 - \text{decay}) * x_o^{\text{opt}}, \quad (36)$$

where $\text{decay} = e^{\text{rate} * (t - t')}$ with $\text{rate} = -1$ and t' being the time when equation (17) is satisfied for the first time, which results in that x_o keeps going up before it smoothly converges to x_o^{opt} .

FIGURE 11: Euclidean norm value of $P_k - P_{k-1}$.TABLE 2: K_k and K_{opt} values in the condition with different weights.

Weight $[W_x \ W_f]$	[1, 1]	[1, 2]	[2, 1]	[3, 1]	[1, 3]
K_k	[-0.2780, 0.0901]	[-0.1464, 0.0481]	[-0.5117, 0.1620]	[-0.7187, 0.2231]	[-0.0995, 0.0329]
K_{opt}	[-0.2361, 0.0890]	[-0.1213, 0.0469]	[-0.4495, 0.1626]	[-0.6458, 0.2259]	[-0.0817, 0.0319]

FIGURE 12: The deformation information and the internal force in the condition with different weights. (a) $[W_x \ W_f] = [1, 1]$, (b) $[W_x \ W_f] = [3, 1]$, and (c) $[W_x \ W_f] = [1, 3]$.

In Figure 11, we also see a fast converging rate of P_k in this case, with the value of $\|P_k - P_{k-1}\|$ decreasing less than ϵ within a few steps of iterations.

In a word, this model-free computational adaptive optimal method is feasible to seek a solution to minimize cost function in equation (12) in bimanual roots holding-damping-spring-model task, minimizing the sum of the tracking position errors and force errors.

Case 4. In this case, we applied the adaptive optimal method under different conditions to test its adaptation performance. The object impedance model is set as $-f_o = 0.01\dot{x}_o + 2x_o$ in this case.

Different weight values and the corresponding K_k and the K_{opt} values are shown in Table 2, and the corresponding simulation results are shown in Figure 12. Through these results, on the one hand, this method is proved to be compatible in different conditions. As we can see from Table 2, all K_k are closed to K_{opt} in different weight conditions. Besides, if we notice the difference between the object impedance model in Case 4 and that in Case 3, we can conclude that this algorithm is also compatible in dealing with different kinds of damping-spring models since the differences between K_k and K_{opt} among all the presented results are small. On the other hand, the practicality of this method is also validated. Figure 12 reveals how the weights

work: if we want to keep the position tracking errors smaller, we can increase the value of W_x ; vice versa, if we want to keep the force tracking to more precise, we can increase the value of W_f . For instance, comparing Figures 12(a) and 12(c), as the desired holding force is 0, the increase of W_f (from 1 to 3) decreases the force tracking errors, in spite of losing some precision in the position tracking.

4. Conclusions

This paper introduces an impedance model-based optimal approach to regulate force and position tracking in bimanual robots holding-object task. In the whole system, several algorithms are employed, for example, the CLIK technique to help derived joint positions for robots to move to the right poses according to the desired deformation of the object; the force observer scheme based on generalized momentum theory to help robots to sense the subjected force exerted by the deformed object, instead of installing sensors on EE, such that the robot joint control torques can be correctly compensated and the EE steady-state tracking errors are tackled; and the model-free computational adaptive optimal method to seek a solution to the optimal trade-off between the position and the force tracking errors. The simulation results demonstrate both the effectiveness and the practicality of this optimal scheme, in which the K_k value in equation (21) is acquired within a very short time without knowing the object's impedance model, which is also very closed to the exact optimal value K_{opt} . And through modifying the weight value in the cost function, different desired control performance can be easily achieved. In a word, by properly integrating these techniques, the bimanual robots can "feel" the force raised by the held object and exert the optimal regulation on their poses according to user-defined weights.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001913. Thanks are due to Dr. Yanan Li, Mr. Dianye Huang, Mr. Guangzhu Peng, and Dr. Jing Luo for their kind technical support.

References

- [1] J. Zhang, M. Li, Y. Feng, and C. Yang, "Robotic grasp detection based on image processing and random forest," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 2427-2446, 2020.
- [2] J. Zhao, J. Liang, and O. Kroemer, "Towards precise robotic grasping by probabilistic post-grasp displacement estimation," 2019, <https://arxiv.org/abs/1909.02129>.
- [3] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067-1079, 2011.
- [4] M. Kaboli, K. Yao, and G. Cheng, "Tactile-based manipulation of deformable objects with dynamic center of mass," in *Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 752-757, IEEE, Cancun, Mexico, November 2016.
- [5] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 2308-2315, IEEE, Anchorage, AK, USA, May 2010.
- [6] K. Chatzilygeroudis, B. Fichera, I. Lauzana et al., "Benchmark for bimanual robotic manipulation of semi-deformable objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2443-2450, 2020.
- [7] P. Sapaty, "Military robotics: latest trends and spatial grasp solutions," *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 4, pp. 9-18, 2015.
- [8] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in *Proceedings of the 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, pp. 1-5, IEEE, Chennai, India, August 2012.
- [9] L. Xiao, J. Gong, and J. Chen, "Industrial robot control systems: a review," in *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)*, pp. 1069-1082, Springer, Singapore, 2020.
- [10] Y. Gan, J. Duan, M. Chen, and X. Dai, "Multi-robot trajectory planning and position/force coordination control in complex welding tasks," *Applied Sciences*, vol. 9, no. 5, p. 924, 2019.
- [11] J. A. Marvel, R. Bostelman, and J. Falco, "Multi-robot assembly strategies and metrics," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1-32, 2018.
- [12] L. Briñón-Arranz, A. Renzaglia, and L. Schenato, "Multirobot symmetric formations for gradient and hessian estimation with application to source seeking," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 782-789, 2019.
- [13] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1162-1171, 2017.
- [14] J. Azeta, C. Bolu, A. A. Abioye et al., "A review on humanoid robotics in healthcare," in *Proceedings of the 4th International Conference on Mechatronics and Mechanical Engineering*, Kuala Lumpur, Malaysia, November 2017.
- [15] H. K. Kalra and R. Chadha, "A review study on humanoid robot SOPHIA based on artificial intelligence," *International Journal of Technology and Computing (IJTC). Techlive Solutions*, vol. 4, no. 3, 2018.
- [16] Z. G. Hou, L. Cheng, and M. Tan, "Multicriteria optimization for coordination of redundant robots using a dual neural network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1075-1087, 2009.
- [17] Z. G. Hou, L. Cheng, and M. Tan, "Decentralized robust adaptive control for the multiagent system consensus problem using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 3, pp. 636-647, 2009.

- [18] Y. Jiang, Z. Liu, C. Chen, and Y. Zhang, "Adaptive robust fuzzy control for dual arm robot with unknown input deadzone nonlinearity," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1301–1314, 2015.
- [19] Z. Lu, P. Huang, and Z. Liu, "Relative impedance-based internal force control for bimanual robot teleoperation with varying time delay," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 778–789, 2020.
- [20] D. Sun, Q. Liao, and A. Loutfi, "Single master bimanual teleoperation system with efficient regulation," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1022–1037, 2020.
- [21] F. Zhang, J. Qu, H. Liu, and Y. Fu, "A pose/force symmetric coordination method for a redundant dual-arm robot," *Assembly Automation*, vol. 38, no. 5, pp. 678–688, 2018.
- [22] Z. Lu, P. Huang, Z. Liu, and H. Chen, "Fuzzy-observer-based hybrid force/position control design for a multiple-sampling-rate bimanual teleoperation system," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 7, pp. 1383–1396, 2018.
- [23] H. Qiao, M. Wang, J. Su et al., "The concept of "attractive region in environment" and its application in high-precision tasks with low-precision systems," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2311–2327, 2014.
- [24] M. Matinfar and K. Hashtrudi-Zaad, "Optimization-based robot impedance controller design," in *Proceedings of the 2004 43rd IEEE Conference On Decision And Control (CDC) (IEEE cat. No. 04CH37601)*, pp. 1246–1251, IEEE, Nassau, Bahamas, December 2004.
- [25] M. Matinfar and K. Hashtrudi-Zaad, "Optimization-based robot compliance control: geometric and linear quadratic approaches," *The International Journal of Robotics Research*, vol. 24, no. 8, pp. 645–656, 2005.
- [26] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.
- [27] R. Johansson and M. W. Spong, "Quadratic optimization of impedance control," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 616–621, IEEE, San Diego, CA, USA, May 1994.
- [28] D. Liu, D. Wang, F.-Y. Wang, H. Li, and X. Yang, "Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2834–2847, 2014.
- [29] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.
- [30] S. S. Ge, Y. Li, and C. Wang, "Impedance adaptation for optimal robot-environment interaction," *International Journal of Control*, vol. 87, no. 2, pp. 249–263, 2014.
- [31] A. Stolt, "On robotic assembly using contact force control and estimation," Ph.D. thesis, Department of Automatic Control, Lund University, Lund, Sweden, 2015.
- [32] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, pp. 634–639, IEEE, Taipei, Taiwan, September 2003.
- [33] G. Peng, C. Yang, W. He, and C. L. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3138–3148, 2019.
- [34] C. Yang, G. Peng, Y. Li et al., "Neural networks enhanced adaptive admittance control of optimized robot–environment interaction," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2568–2579, 2019.
- [35] A. Wahrburg, B. Matthias, and H. Ding, "Cartesian contact force estimation for robotic manipulators—a fault isolation perspective," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 1232–1237, 2015.
- [36] A. Wahrburg, E. Morara, G. Cesari et al., "Cartesian contact force estimation for robotic manipulators using Kalman filters and the generalized momentum," in *Proceedings of the 2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1230–1235, IEEE, Gothenburg, Sweden, August 2015.
- [37] B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint-based robot control," *Robotica*, vol. 8, no. 3, pp. 231–243, 1990.
- [38] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 410–425, 1991.
- [39] P. Falco and C. Natale, "On the stability of closed-loop inverse kinematics algorithms for redundant robots," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 780–784, 2011.
- [40] E. Bjoerlykhaug, "A closed loop inverse kinematics solver intended for offline calculation optimized with GA," *Robotics*, vol. 7, no. 1, p. 7, 2018.
- [41] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, Wiley, Hoboken, NJ, USA, 1995.
- [42] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, Cham, Switzerland, 2016.
- [43] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson/Prentice Hall, London, UK, 2009.

Research Article

Universal Walking Control Framework of Biped Robot Based on Dynamic Model and Quadratic Programming

Xiaokun Leng,^{1,2} Songhao Piao ,¹ Lin Chang,^{1,2} Zhicheng He,^{1,2} and Zheng Zhu²

¹School of Computer Science, Harbin Institute of Technology, Harbin 150001, China

²Leju (Shenzhen) Robotics, Shenzhen 518057, China

Correspondence should be addressed to Songhao Piao; piaoshhit@163.com

Received 16 June 2020; Revised 29 August 2020; Accepted 1 October 2020; Published 31 October 2020

Academic Editor: Hang Su

Copyright © 2020 Xiaokun Leng 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.

Biped robot research has always been a research focus in the field of robot research. Among them, the motion control system, as the core content of the biped robot research, directly determines the stability of the robot walking. Traditional biped robot control methods suffer from low model accuracy, poor dynamic characteristics of motion controllers, and poor motion robustness. In order to improve the walking robustness of the biped robot, this paper solves the problem from three aspects: planning method, mathematical model, and control method, forming a robot motion control framework based on the whole-body dynamics model and quadratic planning. The robot uses divergent component of motion for trajectory planning and introduces the friction cone contact model into the control frame to improve the accuracy of the model. A complete constraint equation system can ensure that the solution of the controller meets the dynamic characteristics of the biped robot. An optimal controller is designed based on the control framework, and starting from the Lyapunov function, the convergence of the optimal controller is proved. Finally, the experimental results show that the method is robust and has certain anti-interference ability.

1. Introduction

Humanoid robots are expected to perform various tasks in the near future. Humanoid robots have the characteristics of multiple joints and multiple degrees of freedom, which makes them have the potential to satisfy multitasking. The humanoid robot's motion control often faces some difficult situations, such as the number of degrees of freedom required by the robot to perform a variety of expected tasks is higher than the number of degrees of freedom the robot has. For a legged robot with six underactuated degrees of freedom as a whole, the most important tasks of motion control are maintaining balance and completing the expected trajectory. Generally speaking, the robot's feet are used to complete these motion trajectories, and then the corresponding grasping tasks are performed by the arms. However, with the development of motion control technology, arm movements can also be used to enhance the overall balance performance of the robot. Arm movement was originally used for dynamic balance during walking,

but it can also be used to assist in tasks such as kicking [1]. Therefore, a full-body motion control framework needs to be established to find the best control output, so that the robot can complete various expected tasks under the constraint conditions such as maintaining balance [2, 3].

The selection of the control variables of the robot's whole-body motion control frame has a great influence on the control performance that the robot can achieve. Examples of well-known whole-body motion control include virtual model control [4, 5], passivity-based whole-body controllers [6–8], and inverse dynamics-based approaches such as those presented in [9, 10]. Most methods solve local minima, i.e., optimize only the state of the current control frame, and then use the centroid dynamics model for predictive control. The method of using the whole-body dynamics model for predictive control has a higher model complexity and a larger amount of calculation [11]. Thanks to the improvement of the computing power of the controller, it can be applied to the gait control of biped robots [12].

There are too many control algorithms for biped robots, and it is difficult to describe all control systems with one structure. The controller is usually divided into walking pattern generator and stabilizer. The former is used to generate a series of joint trajectories to act on the actual robot. When the robot is about to fall, the latter will modify the previously generated trajectory based on various sensor readings (contact force sensor, gyroscope, accelerometer, etc.). The ASIMO robot is the most outstanding representative of this method, usually based on the zero moment point (ZMP), the divergent motion component (DCM), or their deformation [13]. Although it is impossible to unify all methods, the structure of the controller can be divided into upper-level motion plan (advanced motion plan) and lower-level tracking (low-level tracking) [14]. The upper-level motion plan usually uses a simplified dynamic model to generate the center of gravity and foot trajectory, while the lower level below considers using a more complex model to generate the position or torque of each joint. The generation of high-level motion trajectories usually requires model-based prediction of the future. This is the basic idea of model predictive control (MPC). The mainstream method is still to form a trajectory optimization problem and then use MPC to solve the optimization problem and implement a real-time solution [15]. The simplified model is usually used because the calculation amount of the complex model is too large, and there is currently no method for performing real-time calculation. The specific degree of simplification is related to the movement to be achieved, the form of trajectory optimization, and the computing power of the existing controller. It should be noted that even in the case where the upper-layer trajectory is based on a simplified model, since the lower layer considers a more complex model, this makes the resulting joint trajectory conform to the dynamic characteristics of the actual robot and can follow the upper-layer trajectory very well. Usually, the lower level needs to solve inverse kinematics (IK) or inverse dynamics (ID) [16]. The mainstream method is to integrate IK or ID into one or more quadratic programming (QP) and then solve these QP problems in real time to generate the final joint trajectory [17, 18].

This paper proposes a whole-body motion control method based on inverse dynamics, integrating motion control into an optimization framework based on quadratic programming and using joint torque and linear contact force as optimization variables for quadratic programming problems. Considering the complexity of the algorithm and the real-time nature of control, this paper selects the existing technology and finally forms the most complex control framework. The second section introduces the DCM trajectory planning; the third section introduces the control framework used for trajectory tracking; and the fourth section introduces the construction of the secondary planning controller. Section 5 verifies the control framework through experiments.

2. Biped Gait Trajectory Planning Based on DCM

DCM points in three-dimensional space can be defined as

$$\xi = \mathbf{x} + b\dot{\mathbf{x}}. \quad (1)$$

$\xi = [\xi_x, \xi_y, \xi_z]^T$ is DCM point; $\mathbf{x} = [x, y, z]^T$ and $\dot{\mathbf{x}} = [\dot{x}, \dot{y}, \dot{z}]^T$ are the position and velocity of the center of mass (CoM) point, respectively. The formula above shows that the CoM point can naturally and steadily follow the movement of the DCM point [19, 20]. After differentiation, we obtain

$$\dot{\xi} = -\frac{1}{b}\mathbf{x} + \frac{1}{b}\dot{\xi} + \frac{b}{m}F. \quad (2)$$

It shows that the external force F can directly affect the dynamics of DCM. Through analysis, the combined force F received by the system is

$$\mathbf{F} = \frac{mg}{\Delta z_{\text{vrp}}}(\mathbf{x} - r_{\text{vrp}}). \quad (3)$$

In formula (3), m is the system mass, g is the gravity coefficient, \mathbf{x} is the robot coordinate, r is the VRP, which is the space vector, and Δz_{vrp} is the scalar, which represents the height of the robot's torso.

In order to make the actual DCM point run on the planned trajectory, the closed-loop control method is used to track the expected DCM trajectory:

$$\dot{\xi} - \dot{\xi}_d = -k_\xi \underbrace{(\xi - \xi_d)}_{e_\xi}. \quad (4)$$

Adopt proportional (P) control rate to achieve stable tracking of DCM points. Among them, when $k_\xi > 0$, the DCM error can converge gradually and steadily. For a given virtual repellent point (VRP) r_{vrp} and the corresponding centroidal moment pivot point (CMP), \mathbf{r}_{ecm} position, it can converge to the expected value. Expected DCM tracking control rate is

$$r_{\text{vrp}} = \xi + k_\xi \sqrt{\frac{\Delta z_{\text{vrp}}}{g}}(\xi - \xi_d) - \sqrt{\frac{\Delta z_{\text{vrp}}}{g}}\dot{\xi}_d. \quad (5)$$

Among them, formula (5) only stabilizes the unstable part of DCM and does not affect the natural and stable CoM dynamic changes. The closed-loop dynamic situation is

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \underbrace{\begin{bmatrix} -\sqrt{\frac{g}{\Delta z_{\text{vrp}}}}I & \sqrt{\frac{g}{\Delta z_{\text{vrp}}}}I \\ 0 & -k_\xi I \end{bmatrix}}_{\text{feedback}} \begin{bmatrix} x \\ \xi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ k_\xi I & I \end{bmatrix}}_{\text{feedforward}} \begin{bmatrix} \xi_d \\ \dot{\xi}_d \end{bmatrix}. \quad (6)$$

Among them, I and 0 represent the unit matrix and zero matrix of 3×3 . For $k_\xi > 0$, the eigenvectors of the system matrix are stable, and the feed-forward terms ξ_d and $\dot{\xi}_d$ are

both bounded and then can meet bounded input bounded output stable (BIBO) standards.

3. Dynamic Control Framework for Biped Robot

A universal control method for biped robots is to control rigid joint tracking position through IK. This method only requires kinematic models and is widely used in actual robots; the other is based on ID. The method can provide a compliant motion trajectory and is robust to external disturbances, but its performance depends largely on the quality of the dynamic model, which is difficult to obtain by measuring actual robots.

At present, the better method is to use the inverse kinematics IK as a complement to the controller based on the inverse dynamics ID to compensate the modeling error to improve the anti-interference ability. Many biped robot control methods can be decoupled into two layers: a planner that outputs the position of the center of mass and a low-level controller responsible for joint layer control. However, in order to expand the application space of the biped robot and make the robot more robust to external disturbances, the underlying controller also needs to consider full-body kinematics and dynamics. The controller architecture needs to solve for whole-body ID and IK in each control cycle to track higher-level targets. The control model is divided into two parts: IK and ID. Both are formulated as two independent QP problems. Problems have their own goals and constraints.

The servo control command of the joint layer is calculated by the following formula:

$$i = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_f(\tau_d - \tau). \quad (7)$$

In formula (7), q_d , \dot{q}_d , and τ_d are the expected joint position, speed, and torque, and q , \dot{q} , and τ are measured quantities. Use ID to calculate τ_d for force control and IK to calculate q_d and \dot{q}_d .

For robot gait, the robot trajectory is generated at the user level, such as the expected trajectory of feet, hands, and CoM in the Cartesian coordinate system. The full-body robot controller takes it as input and calculates the control amount of each individual joint, such as the joint position, speed, and torque, and then uses these control variables as reference inputs for the joint layer servo controller. The robot's joint position, speed, joint acceleration, and torque are calculated separately. IK and ID are described by QP problems:

$$\begin{aligned} \min_{\chi} \quad & 0.5\chi^T G\chi + g^T \chi \\ \text{s.t.} \quad & C_E\chi + c_E = 0, \\ & C_I\chi + c_I \geq 0. \end{aligned} \quad (8)$$

In the formula, unknown state variables χ and constraints C_E , c_E , C_I , and c_I are all related to specific control problems. Two QP problems are solved once in each control cycle. The cost function can be optimized as

$$0.5\|A\chi - b\|^2. \quad (9)$$

So, $G = A^T A$ and $g = -AT^b$. A and b can be broken down into

$$\begin{aligned} A &= [w_0 A_1 \ w_1 A_1 \ \cdots \ w_n A_1]^T, \\ b &= [w_0 b_1 \ w_1 b_1 \ \cdots \ w_n b_1]^T, \end{aligned} \quad (10)$$

w_i is the weight needed for the cost function.

3.1. Joint Force Control Based on Inverse Dynamics. The motion equation and constraint equation of a biped robot can be described as

$$\begin{aligned} M(q)\ddot{q} + h(q, \dot{q}) &= S\tau + J^T(q)F, \\ J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} &= \ddot{x}. \end{aligned} \quad (11)$$

In formula (11), (q, \dot{q}) is the complete state of the system, including 6 degrees of freedom on the underactuated torso, $M(q)$ is the inertia matrix, and $h(q, \dot{q})$ is the sum of gravity, centrifugal force, and Coriolis force, S is a selection matrix, where the first 6 rows of the 6 degree of freedom (DOF) corresponding to the underactuated torso are zero, and the rest is a unit matrix, τ is the joint moment vector, $J^T(q)$ is the Jacobian matrix of all contact points, F is the vector of all contact forces in the world coordinate system, and x is the vector of contact position and direction in the Cartesian space. The number of rows and columns of F and J^T depends on the number of contact points. In order to describe the contact force of the biped robot with the ground more clearly, the equation of motion (11) was rewritten as follows:

$$M(q)\ddot{q} + h(q, \dot{q}) = S_\tau + \sum_{k=0}^{N_c} J_k^T(q) \cdot f_k. \quad (12)$$

In formula (12), N_c is the number of foot contact friction cones, $J_k^T(q)$ is the Jacobian matrix of each friction cone, and f_k is the contact force.

3.1.1. Cartesian Space Acceleration. Using the acceleration deviation in the Cartesian space as the evaluation function, we obtain

$$\begin{aligned} \ddot{x} &= J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q}, \\ A_{\text{cart}} &= [J(q) \ 0 \ 0], \\ b_{\text{cart}} &= \ddot{x}^* - \dot{J}(q, \dot{q})\dot{q}, \\ \ddot{x}^* &= K_p(x_d^* - x) + K_d(\dot{x}_d^* - \dot{x}) + \ddot{x}_d^*. \end{aligned} \quad (13)$$

Calculate the input \ddot{x}^* by formula (13), where x_d^* , \dot{x}_d^* , and \ddot{x}_d^* are specified by the higher-level controller. Calculate the sum based on the current robot state through forward kinematics, and track the CoM, hand, foot, and torso directions. Acceleration can be calculated by equation (13). For different calculation targets, the Jacobian rows without constraints need to be omitted accordingly. In order to enhance the stability of calculations during contact, contact is not regarded as a hard constraint, but a penalty function

with a higher weight is used to represent the contact to speed up the solution of the quadratic programming problem. The cost function at the time of contact x_d^* , \dot{x}_d^* can be ignored and set $\ddot{x}^* = 0$.

At the same time, considering the influence of CoM trajectory, torso posture, foot center trajectory of swinging leg, and foot center trajectory of standing leg on robot control, the cost function formula (10) is extended to (14), where w is the weighting coefficient and A_{com} is the centroid tracking cost in the end acceleration tracking cost:

$$A = \begin{bmatrix} w_{\text{CoM}} A_{\text{CoM}} \\ w_{\text{Torsor}} A_{\text{Torsor}} \\ w_{\text{Swing}} A_{\text{Swing}} \\ w_{\text{Stance}} A_{\text{Stance}} \\ w_x A_x \end{bmatrix}, \quad (14)$$

$$b = \begin{bmatrix} w_0 b_0 \\ w_1 b_1 \\ \vdots \\ w_n b_n \\ w_x b_x \end{bmatrix}.$$

Constraint equation (15) corresponds to the CoM trajectory:

$$\begin{aligned} \ddot{x} &= J \cdot \ddot{q} + \dot{J} \cdot \dot{q}, \\ A_{\text{CoM}} &= [J_{\text{CoM}}(q)], \\ b_{\text{CoM}} &= \ddot{x}_{\text{CoM}}^* - \dot{J}_{\text{CoM}}(q) \cdot \dot{q}. \end{aligned} \quad (15)$$

Similarly, the swing leg and standing leg trajectory constraint equations are

$$\begin{aligned} A_{\text{Swing}} &= [J_{\text{Swing}}(q)], \\ b_{\text{Swing}} &= \ddot{x}_{\text{Swing}}^* - \dot{J}_{\text{Swing}}(q) \cdot \dot{q}, \\ A_{\text{Stance}} &= [J_{\text{Stance}}(q)], \\ b_{\text{Stance}} &= \ddot{x}_{\text{Stance}}^* - \dot{J}_{\text{Stance}}(q) \cdot \dot{q}. \end{aligned} \quad (16)$$

The body posture constraint equation is

$$\begin{aligned} A_{\text{Torsor}} &= [J_{\text{Torsor}} \omega], \\ b_{\text{Torsor}} &= \dot{\omega}_{\text{Torsor}}^* - J_{\text{Torsor}}(q) \cdot \dot{q}, \\ \dot{\omega}_{\text{Torsor}}^* &= K_p (\xi_d^* - \xi) + K_d (\omega_d^* - \omega) + \dot{\omega}_d. \end{aligned} \quad (17)$$

3.1.2. Center of Pressure (CoP). Given the contact force and contact moment ${}^b M$, ${}^b F$ in the foot coordinate system, the position of the pressure center in the foot coordinate system is

$$p = \begin{bmatrix} \left(\frac{{}^b M_y}{{}^b F_z} \right) \\ \left(\frac{{}^b M_x}{{}^b F_z} \right) \end{bmatrix}. \quad (18)$$

Cost function of pressure center deviation is

$$\begin{aligned} A_{\text{CoP}} &= \begin{bmatrix} 0 & 0 & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} & \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \end{bmatrix}, \\ b_{\text{CoP}} &= 0. \end{aligned} \quad (19)$$

In formula (19), p is the contact force amplitude vector, which is consistent with the general β_{ij} meaning of the constraint equation, $*$ represents the reference value, and p_x^* , p_y^* is the expected pressure center position in the foot coordinate system given by the high-level controller. In formula (20), R is the transformation matrix from the world coordinate system to the foot coordinate system.

3.1.3. Weight Distribution. In the two-leg support, specify the desired weight distribution $w^* = F_{zl} / (F_{zl} + F_{zr})$, and add this to the cost function:

$$\begin{aligned} A_{\text{weight}} &= \begin{bmatrix} 0 & 0 & S_{\text{weight}} \end{bmatrix}, \\ b_{\text{weight}} &= 0. \end{aligned} \quad (20)$$

In formula (20), S_{weight} is a row vector with zeros, except for $S_{\text{weight}}(3) = 1 - w^*$ and $S_{\text{weight}}(9) = -w^*$, the rest are zero.

3.1.4. Track State Variables and Regularization Directly. Use the expected value directly to evaluate χ ; then formula (21) is obtained, where A is the selection matrix, b is target vector, q is joint position, τ is torque, F is end contact force, and $*$ represents the reference value:

$$\begin{aligned} A_{\text{state}} &= I, \\ b_{\text{state}} &= [\dot{q}^* \quad \tau^* \quad F^*]^T. \end{aligned} \quad (21)$$

If no target value is specified in the formula, 0 is used. This item can be used to directly control specific joints or forces, which can directly adjust the value of the state variable χ , making QP problems easier to solve.

3.1.5. Torque Change. In order to avoid joint torque oscillation, the speed of change of τ needs to be considered:

$$\begin{aligned} A_{d\tau} &= \begin{bmatrix} 0 & I & 0 \end{bmatrix}, \\ b_{d\tau} &= \tau_{\text{prev}}. \end{aligned} \quad (22)$$

In formula (22), τ_{prev} is the output of the previous period.

3.2. Friction Cone Contact Model. In the control of a biped robot, it is difficult to solve the contact force under different contact conditions. In order to simplify the solution process, the contact surface of the foot is approximated by multiple contact points, and the contact force of these contact points is solved. For the feet of a robot, a friction cone constraint can be used to determine whether the point will slide. The point will not slide only if the contact force vector is inside the friction cone. As shown in Figure 1, the apex of the friction cone is located at the contact point, the normal line is perpendicular to the ground, and the slope is determined by the friction coefficient of the contact between the sole and the point.

However, this constraint method represented by a friction cone is nonlinear, although the solver can handle conic constraints. However, the calculation of the cone constraint is heavy. In order to speed up the solution, this paper uses a convex polygon as shown in Figure 1 to approximate the cone constraint of the contact point. In the contact force model represented by the convex polygon model, the feasible contact force and contact moment of the foot are a linear combination of the contact force f_i of each point. f_i is the product of a unit vector ${}^0u f_i$ representing the direction of the contact force in space and a scalar ρ_i representing the magnitude of the contact force, as shown in formula (25):

$$f_i = {}^0u f_i \rho_i. \quad (23)$$

Therefore, at a given contact point, the contact forces in all friction cones can be represented by the contact forces at a total of $n\rho$ contact points. The unit vector ${}^0u f_i$ of the feasible contact force direction is determined by the contact point. The amplitude scalar ρ_i of the contact force is limited to a positive value. All $n\rho$ contact force amplitude scalars are combined into a contact force amplitude vector ρ ; then,

$$\rho = [\rho_1 \ \cdots \ \rho_{n_p}]^T. \quad (24)$$

Contact force vector ρ can be mapped to six-dimensional space contact force and moment:

$$\begin{bmatrix} F_x & F_y & F_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T = {}^0A_{\rho,k} \rho. \quad (25)$$

In formula (25), k represents the k th part of the robot that is in contact with the external environment. Each ${}^0A_{\rho,k}$ maps the contact force amplitude vector ρ on the limb to the contact force and contact torque to the k limb.

Each foot of a biped robot is approximated by four contact points located on the edge of the foot, and the friction cone of each contact point is approximated by 4 contact force unit vectors, so the total number of contact force variables per leg $n_p = 16$. In formula (25), the force ρ is mapped to the foot contact torque through the matrix $A_{\rho,k}$.

It should be noted that the friction force obtained is reasonable only when the measured friction coefficient, the contact state of the foot, and the ground normal vector of the

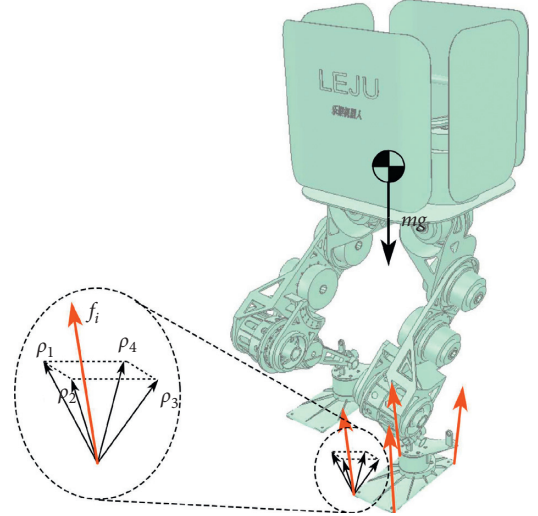


FIGURE 1: Friction cones and polyhedral cones.

contact point are approximately accurate; otherwise, the system will be seriously unstable.

3.3. Constraints. Considering the constraints such as the output torque of each joint of the actual robot, the motion equations need to be used as equation constraints, and various inequality constraints must be added to ensure that the solution results do not violate relevant physical conditions. Inequality constraints include positive contact forces, joint moments within a certain range, etc.; the optimization problem is formulated as

$$\begin{aligned} \min_{\ddot{q}, \beta, f} \quad & \omega_1 \omega_1 A_{\text{CoP}} f_i - b^2 + \omega_2 A_{\ddot{q}} \ddot{q} - b^2 + \varepsilon_1 \tau - \tau_{\text{prev}}^2 + \varepsilon_2 \sum_{ij} \beta_{ij}^2, \\ M(q) \ddot{q} + h(q, \dot{q}) = \quad & S_\tau + \sum_{j=1}^{N_c} J_j^T(q) \cdot f_j, \\ f_j = \sum_{i=1}^{N_c} \beta_{ij} v_{ij}, \quad & \beta_{ij} \geq 0, \forall i = \{1, \dots, 4\}, \\ \forall j = \{1, 2, \dots, N_c\}, \\ \tau < \tau_{\text{max}}. \end{aligned} \quad (26)$$

In formula (26), they are dynamic model constraint, friction cone constraint, contact force amplitude vector constraint, and joint torque constraint. The cost items are CoP tracking cost, end acceleration tracking cost, and torque change cost, τ is the joint torque, contact force vector cost, β_{ij} is the contact force amplitude vector, f_i represents the friction between the sole of the foot and the ground, and τ_{max} is the limit value of the output torque of each joint. The coefficients of each cost term are ω_1 , ω_2 , ε_1 , and ε_2 .

At present, the friction coefficient f is obtained through preliminary experiments, and the friction coefficient of different roads is different. The function of the friction cone is to limit the force of the robot on the ground when the friction coefficient is small, to prevent sliding friction during

walking, which will cause the robot to lose stability. In the future, under unknown walking conditions, sliding friction during walking will be detected and corrected.

Finally, the robot control framework is shown in Figure 2. There are many parameters in the control framework that need to be adjusted. The parameters that need to be adjusted include the PD controller feedback coefficient when calculating the end acceleration cost, the weight coefficient of each end point cost in the end acceleration cost, and the weight coefficient of each cost in the secondary programming problem, such as ω_1 , ω_2 , ε_1 , and ε_2 .

Parameter debugging process is as follows:

- (1) The weight coefficients w_1 and w_2 in the end acceleration cost are set to a larger value, about 1–100. Debug w_1 and w_2 until better CoP tracking and end acceleration tracking are achieved.
- (2) Debug the PD controller feedback coefficient until the robot achieves a better position tracking effect.
- (3) Given small values of ε_1 and ε_2 , about 10^{-6} – 10^{-2} , adjust the parameters until the robot joint torque and contact force fluctuations are minimal.

Although there are many total parameters, it is easier to determine the appropriate control parameters when debugging in stages.

4. Optimal Controller Based on PD Control and Quadratic Programming

The trajectory planning algorithm and the IK and ID algorithms can be used to obtain the desired angle of each joint of the robot, but the actual output torque of each joint is limited. Therefore, it is necessary to design an optimal controller to limit the joint torque of the biped robot. Through optimized force control, the robot's motion posture can better track the expected posture. The biped robot dynamics model is

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J(q)F_1. \quad (27)$$

Output dynamics are

$$\dot{y} = T_y \dot{q} - \frac{\partial y_d}{\partial t}. \quad (28)$$

Differentiating formula (28), we obtain

$$\ddot{y} = \dot{T}_y \dot{q} + J_y \ddot{q} - \frac{\partial^2 y_d}{\partial t^2} - \frac{\partial}{\partial q} \left(\frac{\partial y_d}{\partial t} \right) \dot{q}. \quad (29)$$

Substitute \ddot{q} into equation (29) to obtain equation (31), where

$$\ddot{q} = -D^{-1}C\dot{q} - D^{-1}G + D^{-1}BU + D^{-1}J(q)F, \quad (30)$$

$$\begin{aligned} \ddot{y} = & \dot{J}_y \dot{q} + J_y D^{-1}BU - J_y D^{-1}(G - JF_1 + c\dot{q}) \\ & - \frac{\partial^2 y_d}{\partial t^2} - \frac{\partial}{\partial q} \left(\frac{\partial y_d}{\partial t} \right) \dot{q}. \end{aligned} \quad (31)$$

Sorting formula (31), we obtain

$$\begin{aligned} \ddot{y} - \frac{1}{J_y} \dot{q} + \frac{\partial^2 y_d}{\partial t^2} + \frac{\partial}{\partial q} \left(\frac{\partial y_d}{\partial t} \right) \dot{q} + J_y D^{-1}(G - JF_1 + c\dot{q}) \\ = J_y D^{-1}Bu. \end{aligned} \quad (32)$$

Multiplying formula (32) left by $D_y = (J_y D^{-1} J_y^T)^{-1}$, we obtain

$$F = - \left(K_p y + K_d \dot{y} + \dot{D}_y \dot{y} - (C_y + J_{y_d}) \dot{q} - G_y - D_y \frac{\partial^2 y_d}{\partial t^2} \right). \quad (33)$$

Among them, $D_y J_y D^{-1} BU = F$, $D_y T_y D^{-1}(G - JF_1) = G_y$, $J_{y_d} = D_y (\partial/\partial q) (\partial y_d/\partial t)$, and $C_y = D_y (J_y D^{-1} C - \dot{J}_y)$. Corresponding form is

$$Bu = (\mathbf{1} - N[(\mathbf{1} - BB^T)N]^+) J_y^T F. \quad (34)$$

Among them,

$$N = (\mathbf{1} - J_y^T (J_y D^{-1} J_y^T)^{-1} J_y D^{-1}). \quad (35)$$

Formula (34) describes the relationship between the auxiliary variable, and when the controller F is used, the corresponding control law can be substituted into the equation to find. Consider the form of F :

$$F = - \left(K_p y + K_d \dot{y} + \dot{D}_y \dot{y} - (C_y + J_{y_d}) \dot{q} - G_y - D_y \frac{\partial^2 y_d}{\partial t^2} \right). \quad (36)$$

Put it into formula (27); then,

$$\begin{aligned} D_y \ddot{y} = & -k_p y - k_d \dot{y} - \dot{D}_y \dot{y}, \\ \ddot{y} = & -D_y^{-1} k_p y - D_y^{-1} k_d \dot{y} - D_y^{-1} \dot{D}_y \dot{y}. \end{aligned} \quad (37)$$

For the above system, consider the following Lyapunov function:

$$V = \frac{1}{2} y^T k_p y + \frac{1}{2} \dot{y}^T D_y \dot{y} + \alpha y^T D_y \dot{y}. \quad (38)$$

Deriving from the above formula, we obtain

$$\begin{aligned} \dot{V} = & \frac{1}{2} \dot{y}^T k_p y + \frac{1}{2} y^T k_p \dot{y} + \frac{1}{2} \dot{y}^T D_y \dot{y} + \frac{1}{2} \dot{y}^T D_y \ddot{y} \\ & + \frac{1}{2} \dot{y}^T D_y \dot{y} + \alpha y^T D_y \dot{y} + \alpha \dot{y}^T D_y \dot{y} + \alpha y^T D_y \ddot{y} + \alpha \dot{y}^T \dot{D}_y \dot{y} \\ = & y^T (-\alpha k_p) y + y^T \left(k_p - \frac{k_p D_y^T D_y}{2} - \alpha k_d + \alpha \dot{D}_y \right) \dot{y} \\ & + \dot{y}^T \left(-\frac{k_p}{2} \right) y + \dot{y}^T \left(\frac{k_d D_y^T D_y + \dot{D}_y D_y^T D_y + k_d - 2\alpha D_y}{2} \right) \dot{y}. \end{aligned} \quad (39)$$

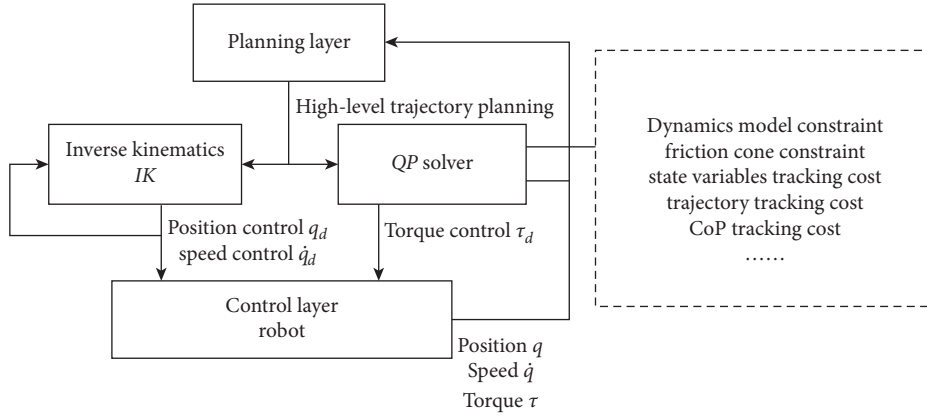


FIGURE 2: Robot control framework.

Converting the above formula into a matrix, we obtain

$$\dot{V} = - \begin{bmatrix} y^T & \dot{y}^T \end{bmatrix} \begin{bmatrix} \alpha k_p & -k_p + \frac{k_d D_y^{-T} D_y}{2} + \alpha k_d - \dot{\alpha} D_y \\ \frac{k_p}{2} & \frac{k_d D_y^{-T} D_y + \dot{D}_y^T D_y^{-T} D_y + k_d - 2\alpha D_y}{2} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix}. \quad (40)$$

In order to ensure the positive definiteness of the matrix in formula (40), $k_p > ((k_d D_y^T D_y)/2)$, a smaller value α is often selected, but it is necessary to ensure that formula (41) is a positive definite matrix:

$$\begin{bmatrix} \alpha k_p & -k_p + \frac{k_d D_y^{-T} D_y}{2} + \alpha k_d - \dot{\alpha} D_y \\ \frac{k_p}{2} & \frac{k_d D_y^{-T} D_y + \dot{D}_y^T D_y^{-T} D_y + k_d - 2\alpha D_y}{2} \end{bmatrix}. \quad (41)$$

For an actual biped robot system, the physical meaning of D in the actual system is an inertia matrix, which is positively definite and symmetrical. Therefore, D_y can be obtained as a symmetric matrix. The selection conditions for k_p can be modified to $k_p > ((k_d D_y^T D_y)/2)$; the corresponding matrix to be verified can also be simplified to

$$\begin{bmatrix} \alpha k_p & -k_p + \frac{k_d}{2} + \alpha k_d - \dot{\alpha} D_y \\ \frac{k_p}{2} & \frac{k_d + \dot{D}_y^T + k_d - 2\alpha D_y}{2} \end{bmatrix}. \quad (42)$$

The construction space of the biped robot system includes the joint angles $q = (q_1, q_2, \dots, q_{12})$ and the corresponding underactuated degrees of freedom. The mass data and the moment of inertia data of each part of the robot body are obtained from the CAD model. Considering trajectory following, its control problem is of the form:

$$\begin{aligned} \min_{F, u, \delta} \quad & (F - F_{\text{ref}}(t, q, \dot{q}))^T (F - F_{\text{ref}}(t, q, \dot{q})) + 1000 * \delta^2 \\ \text{s.t.} \quad & (\alpha(y)y^T + y^T) \left(K_p y + K_d \dot{y} + \frac{1}{2} \dot{D}_y \dot{y} - C_y \dot{q} - G_y - D_y \frac{\partial^2 y_d}{\partial t^2} + u \right) \leq \delta, \\ & Bu = (1 - N[(1 - BB^T)N]^+)^T J_y^T F, \\ & |u_i| \leq U_{\text{lim}} Nm. \end{aligned} \quad (43)$$

Among them, u is the actual output torque value of each joint, and U_{lim} is the maximum allowable torque of each joint, according to the actual torque limit of each joint. By reasonable selection of the coefficients K_p and K_d in the controller, the influence of system

nonlinearity on system stability can be avoided to the greatest extent. The matrix B represents the selection matrix in the underdrive system. The first 6 rows of the corresponding underdrive 6 DOFs are 0, and the rest are the unit matrix.

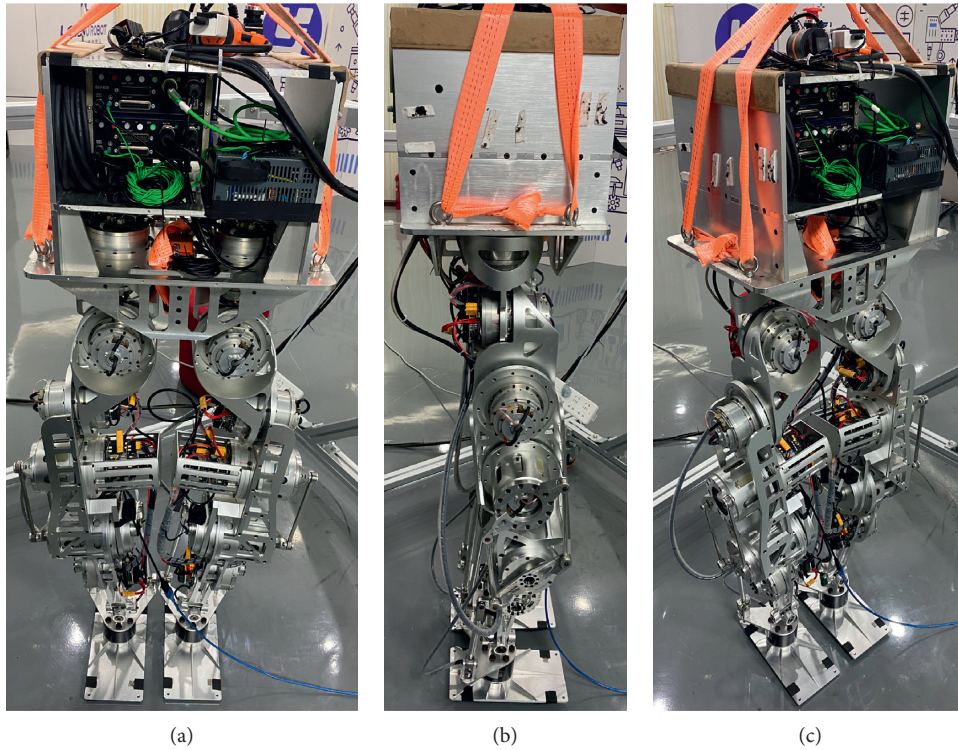


FIGURE 3: Biped robot for experiment. (a) Front view. (b) Side view. (c) Oblique view.

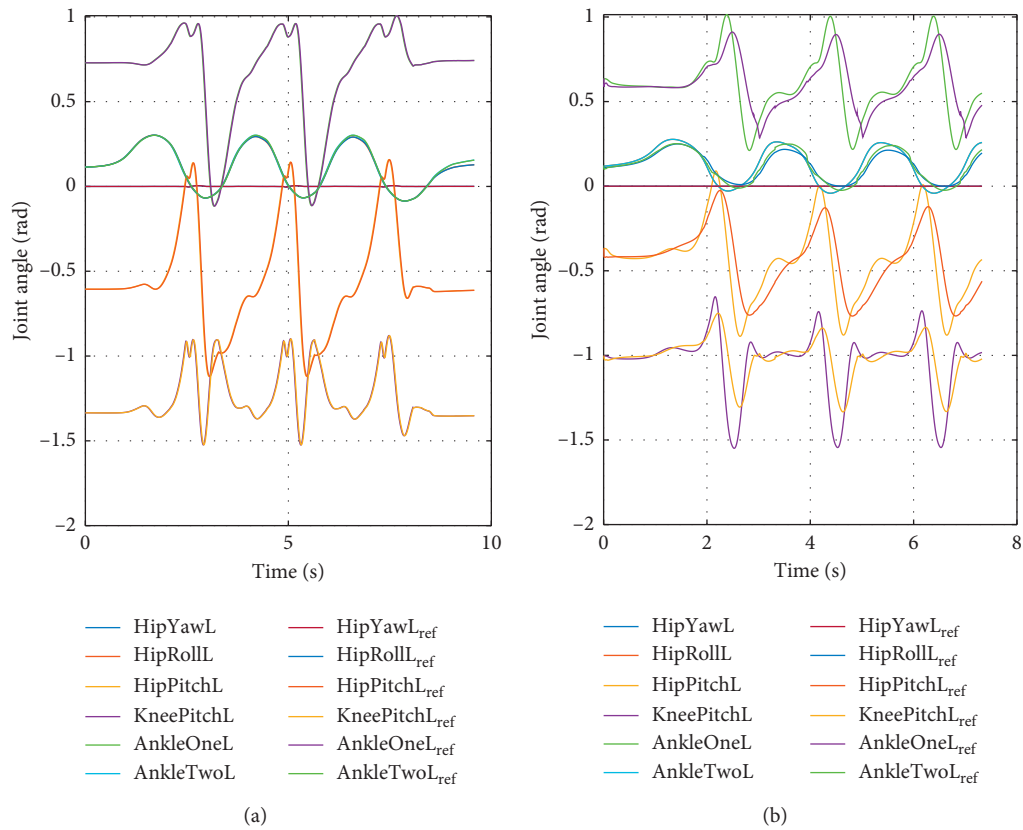


FIGURE 4: Angle of joints of biped robot. (a) Position control. (b) Force control.

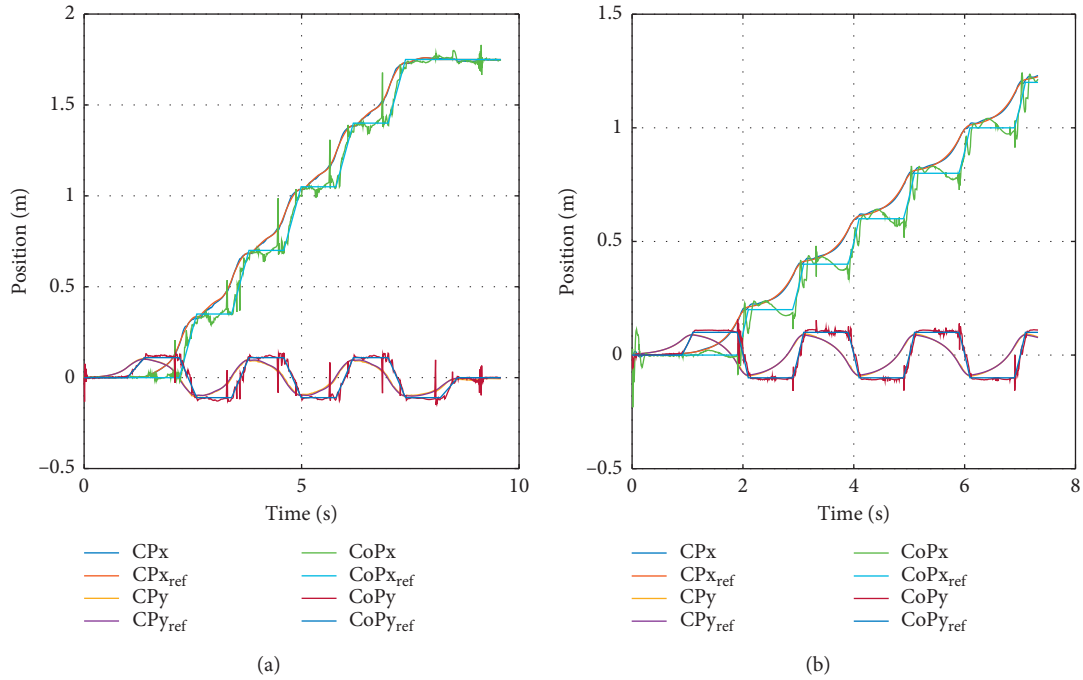


FIGURE 5: CoP trajectory of biped robot. (a) Position control. (b) Force control.

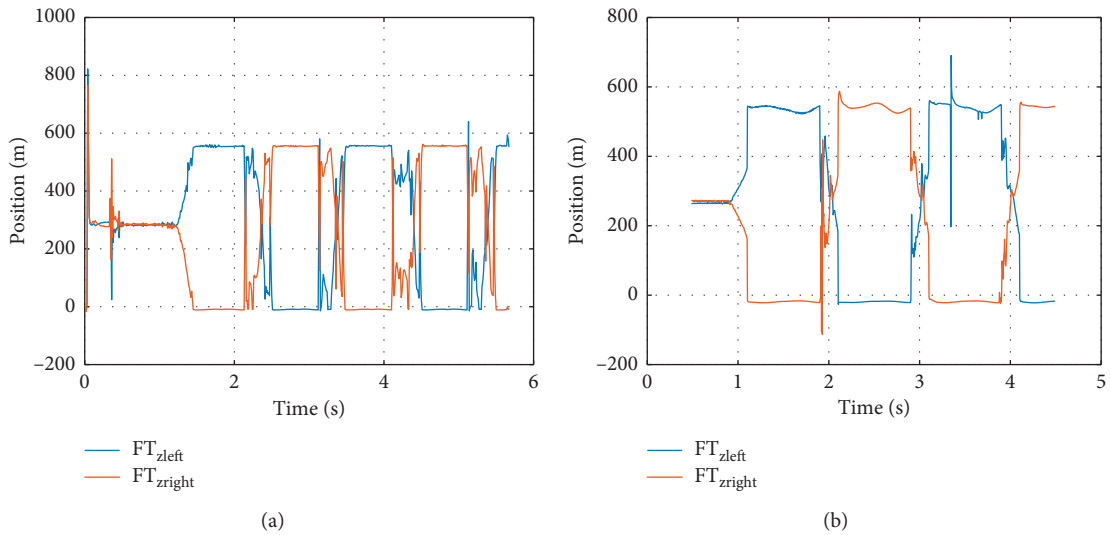


FIGURE 6: SDF data of biped robot. (a) Position control. (b) Force control.

In order to quickly solve the optimal control problem, the open-source OSQP solver is selected to solve the abovementioned quadratic programming problem. The solver uses an improved first-order alternating direction multiplier method (ADMM) to solve the quadratic programming problem [21]. By decomposing the matrix at the initial stage, the calculation amount in the subsequent process is greatly reduced. The optimal control algorithm is robust; its optimization results are not sensitive to the initial

value, and good solution results can be obtained by using different initial values.

The control framework integrates methods that can improve the robustness of the system. The framework uses DCM for trajectory planning, and DCM itself has anti-interference ability; uses friction cone and whole-body dynamics model to improve the accuracy of the model and reduce the control error caused by the rough model; considers responsibility for the actual situation to obtain a variety

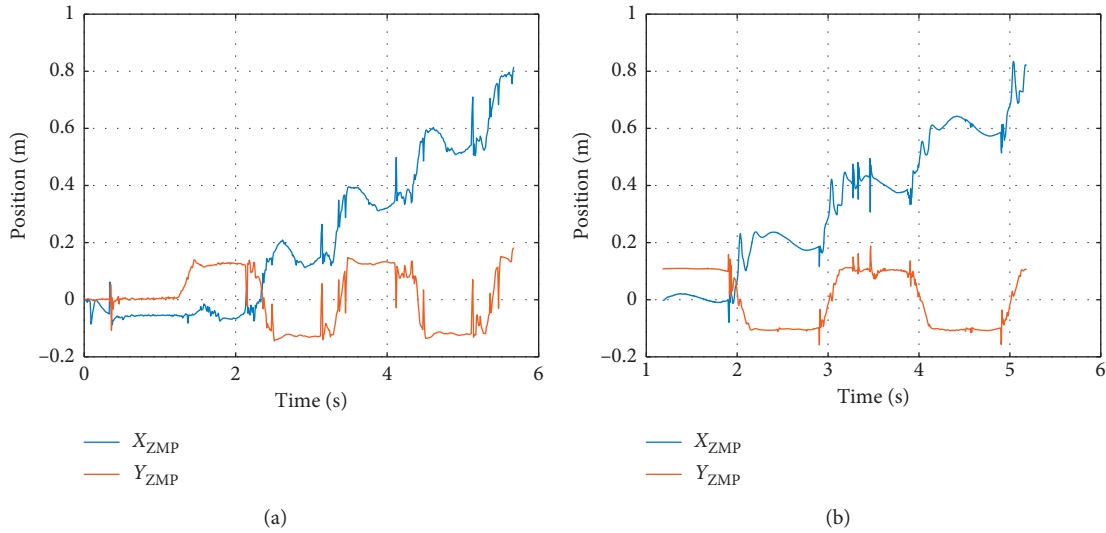


FIGURE 7: ZMP trajectory of biped robot. (a) Position control. (b) Force control.

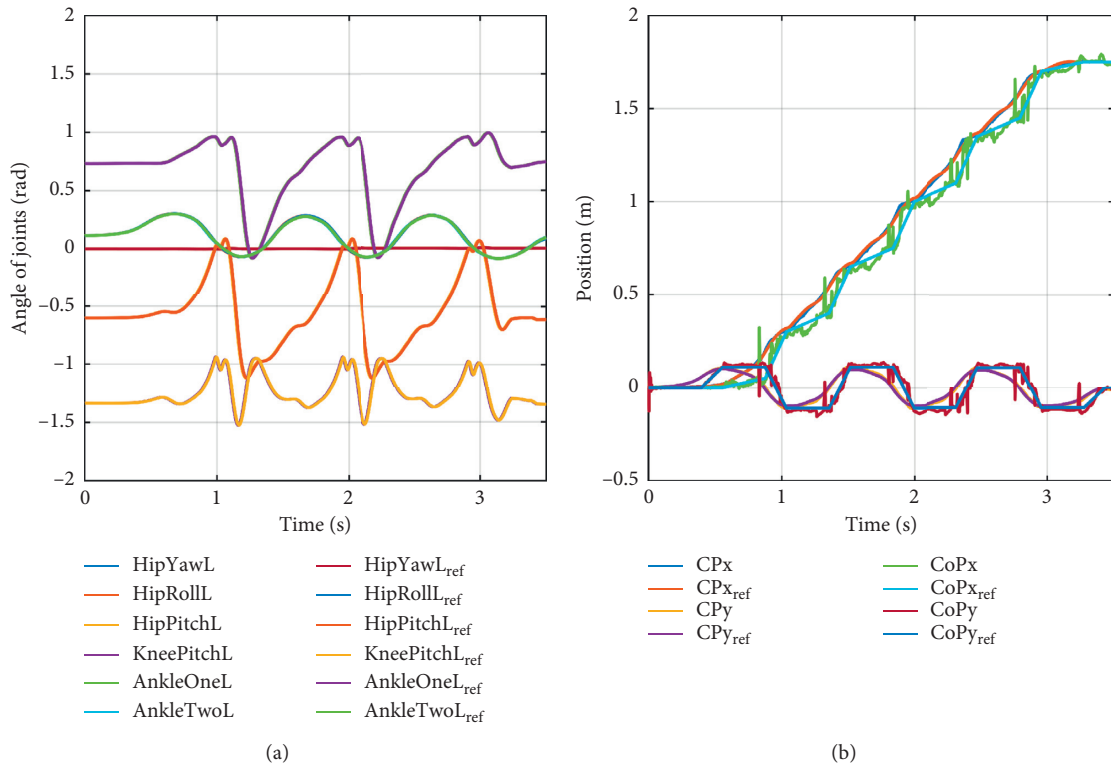


FIGURE 8: Biped robot increase weight experiment. (a) Angle of joints. (b) CoP trajectory.

of constraints and use QP; realizes the optimal controller; and further improves the robustness of robot walking.

5. Experimental Verification

Under the robot control framework, force control and position control are applied to the biped robot, respectively, and the comparison effect is examined. The experimental conditions are that the biped robot moves forward 5 steps,

the robot stride is 30 cm, and the period is 1.2 seconds. The weight of the physical robot is about 60 kg, as shown in Figure 3. The robot's inverted pendulum time constant ω is selected according to the interest rate difference.

Figure 4(a) is the joint angle plan and actual trajectory curve under position control, and Figure 4(b) is the joint angle plan and actual trajectory curve under force control. Figure 5(a) shows the CP and CoP trajectories under position control, and Figure 5(b) shows the CP and CoP

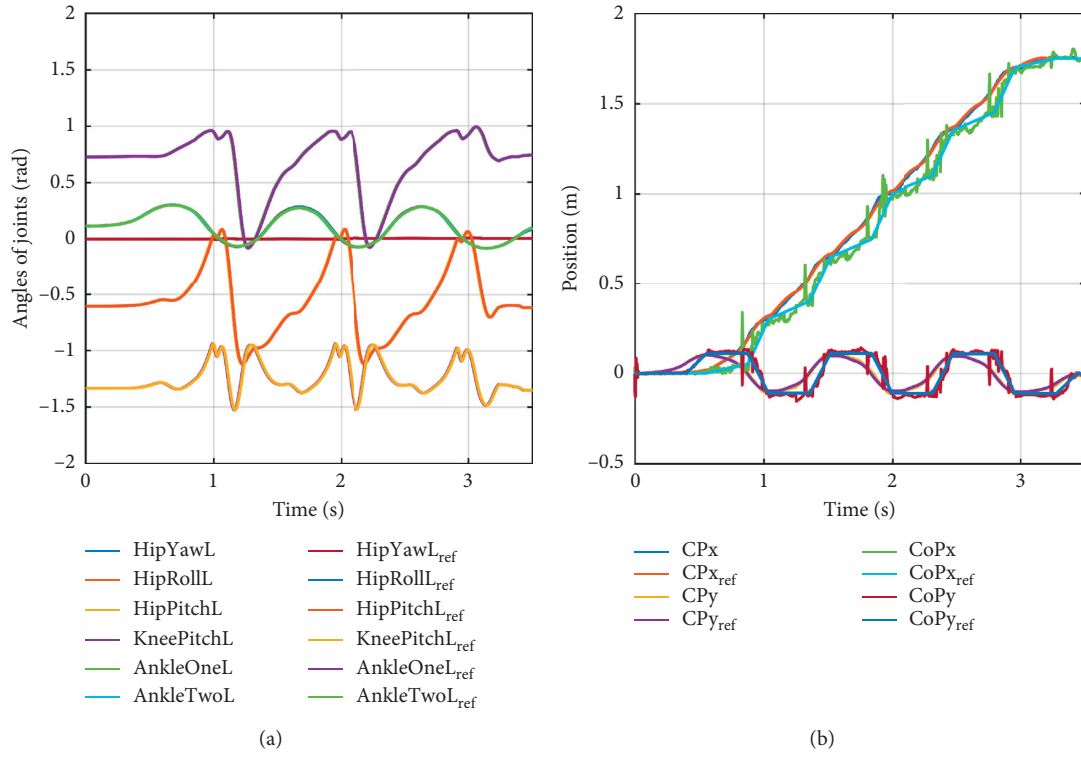


FIGURE 9: The centroid of the biped robot moves forward 4 cm experiment. (a) Angles of joints. (b) CoP trajectory.

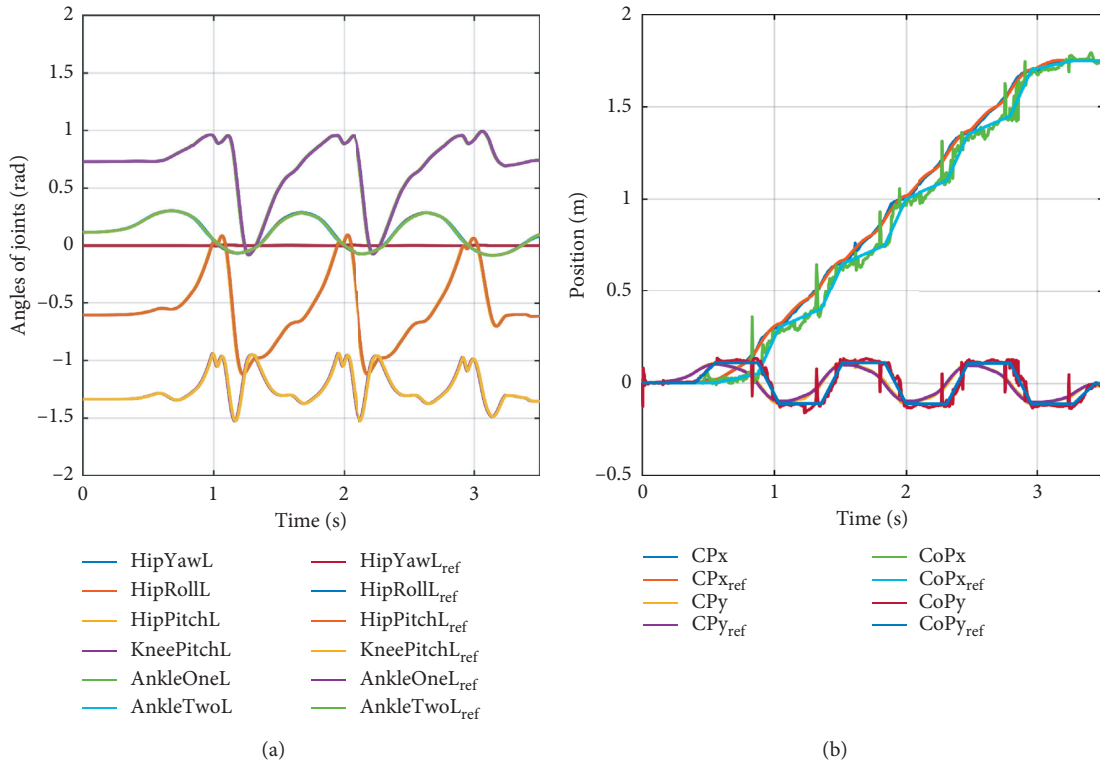


FIGURE 10: Biped robot antithrust experiment. (a) Angles of joints. (b) CoP trajectory.

trajectories under force control. It can be seen that based on the position control, each joint performs position control according to the trajectory plan calculated in advance, so the joint angle planning curve and the actual trajectory curve are basically coincident. In the case of force control, the joint output force is tracked and controlled, and the joint angle planning curve and actual curve are no longer consistent. Although the position control ensures the accuracy of position output, the control effect of the CoP trajectory is not as good as that of the force control method. Under the force control method, the CoP trajectory curve of the robot walking will be smoother, and the fluctuation is smaller than that of the position control method.

Under the position control and force control methods, the six-dimensional force (SDF) data of the ankle joint is shown in Figures 6(a) and 6(b). The force control method can effectively reduce the impact force of the robot's foot and the ground. Under position control, the robot's feet are in hard contact with the ground, and the ground reaction force is greater than the force control method, which affects the robot's walking stability. The robot goes through several walking cycles to balance the impact. Compared with the position control method, the force control method can better absorb the external impact and ground reaction force and restore the balance in a shorter time.

Figures 7(a) and 7(b) are ZMP trajectories under two control modes.

It can be seen that based on position control, ZMP trajectory control is not as good as force control. During walking, ZMP can only be proved to be stable if it is within the stable area of the foot, but from the perspective of ZMP trajectory, the actual ZMP is calculated and fluctuates greatly, so ZMP trajectory only shows the stability of walking to a certain extent.

Figures 8(a) and 8(b) show the trajectory of the joint position after the robot's mass increases by 2 kg and the trajectory effect of the CP point and the CoP. It can be seen that after the mass of the robot increases, its joint trajectory remains almost unchanged, and there is a certain deviation between the actual CP trajectory and the reference trajectory, but due to the robustness of the optimal controller, the tracking error of the robot is small. The CP trajectory will remain within a certain range and will not cause the robot to fall, and the tracking of the CoP trajectory will be better.

Figures 9(a) and 9(b) show the robot joint position trajectory after the robot's center of mass is moved forward by 4 cm and the CP point and CoP trajectory effects. There is a certain inherent deviation between the actual CP trajectory and the reference trajectory of the robot, but due to the robustness of the optimal controller, the deviation value of the CP trajectory will eventually tend to a more stable value, which will not cause the robot to be unstable. In other words, CoP trajectory tracking is better.

Figures 10(a) and 10(b) show that the trunk of the robot is subjected to a continuous thrust of 30 N at 1.5–1.6 s, and the thrust is along the positive direction of the x -axis. It can be seen that when the robot is subjected to thrust, the CP trajectory along the x -axis direction has a significant deviation, but it will not cause the CP tracking divergence. After

the thrust stops, due to the role of the optimal controller, the CP tracking error gradually decreases until converged to approximately 0; relatively speaking, the CoP tracking is good, and the corresponding joint running trajectory is less affected by thrust, which is almost the same as the effect when it is not subjected to thrust.

6. Conclusion

This paper presents the technique of the robot motion control based on the dynamic model and quadratic programming. In order to improve the walking robustness of the biped robot, this article starts with the construction of the robot motion control framework from three aspects: planning method, mathematical model, and control method. Firstly, the biped gait trajectory planning is introduced based on the divergent component of motion. Then, the dynamic control framework is discussed for biped robot, including the joint force control, friction cone contact model, and various constraints. This paper integrates these methods into the control framework and verifies the effectiveness of the control framework on physical robots. Finally, the experimental results show that the method is robust and has certain anti-interference ability.

Further research work will focus on improving the accuracy of the dynamics model, the robustness of the control method, and the verification experiments in more complex scenarios, for example, complex walking scenes (such as sand or outdoors) and greater walking disturbances, etc. The ultimate goal is for the robot to walk stably in a human environment.

Nomenclature

- CoM: Center of mass, the trajectory curve of the center of mass during the robot walking
- CoP: Center of pressure, the trajectory curve of the pressure center of the foot of the robot during the robot walking
- CP: Capture point, robot's walking trajectory based on DCM planning
- DCM: Divergent motion component, a walking strategy of biped robot
- ID: Inverse dynamics, calculate the torque required for the joints given the motion of the robot
- IK: Inverse kinematics, calculate the required joint angle for a given foot position
- QP: Quadratic programming, convex quadratic programming problem
- ZMP: Zero moment point, a criterion of walking stability of biped robot.

Data Availability

No data were used to support this study.

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 Leju (Shenzhen) Robotics Co., Ltd.

Supplementary Materials








The supplementary attachment is the description of the experimental platform and experimental data. (*Supplementary Materials*)

References

- [1] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: coupled planning, terrain mapping and whole-body control," 2020, <http://arxiv.org/abs/2003.05481>.
- [2] C. Yang, K. Huang, H. Cheng, Y. Li, and C.-Y. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2398–2409, 2017.
- [3] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2018.
- [4] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: an intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.
- [5] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart, "Scarleth: design and control of a planar running robot," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 562–567, IEEE, San Francisco, CA, USA, September 2011.
- [6] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Reactive whole-body control: dynamic mobile manipulation using a large number of actuated degrees of freedom," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 20–33, 2012.
- [7] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.
- [8] B. Henze, C. Ott, and M. A. Roa, "Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3253–3258, IEEE, Chicago, IL, USA, September 2014.
- [9] T. Koolen, J. Smith, G. Thomas et al., "Summary of team IHMC's virtual robotics challenge entry," in *Proceedings of the 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 307–314, IEEE, Atlanta, GA, USA, October 2013.
- [10] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.
- [11] M. Van, X. P. Do, and M. Mavrouniotis, "Self-tuning fuzzy pid-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *ISA Transactions*, vol. 96, pp. 60–68, 2020.
- [12] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2589–2594, IEEE, Hong Kong, China, May 2014.
- [13] S. Shigemi, A. Goswami, and P. Vadakkepat, "ASIMO and humanoid robot research at honda," in *Humanoid Robotics: A Reference*, pp. 55–90, Springer, New York City, NY, USA, 2019.
- [14] J. Park, Y. Lee, M. Kim, S. Hwang, J. Jung, and J. Kim, "Human-assisted humanoid robot control," in *Robotics Research*, pp. 85–90, Springer, New York City, NY, USA, 2020.
- [15] J. Nubert, J. Kohler, V. Berenz, F. Allgower, and S. Trimpe, "Safe and fast tracking on a robot manipulator: robust mpc and neural network control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.
- [16] S. Lyu and C. C. Cheah, "Human-robot interaction control based on a general energy shaping method," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2445–2460, 2019.
- [17] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.
- [18] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.-Y. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1759–1770, 2018.
- [19] J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4420–4427, IEEE, San Francisco, CA, USA, September 2011.
- [20] J. Engelsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.
- [21] T. V. Dang, K. V. Ling, and J. M. Maciejowski, "Embedded ADMM-based QP solver for MPC with polytopic constraints," in *Proceedings of the 2015 European Control Conference (ECC)*, pp. 3446–3451, IEEE, Linz, Austria, July 2015.

Research Article

Multisensor-Based Autonomous Grasp Planning for Mobile Manipulator Navigation System with a Novel Soft Gripper

Heng Zhang ^{1,2}, Yingbai Hu ³, Jianghua Duan ⁴, Qing Gao ^{1,2}, Langcheng Huo ^{1,2},
Qiwen Wang ^{1,2} and Yongquan Chen ^{1,2}

¹Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China

²Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, China

³Department of Informatics, Technical University of Munich, Munich 85748, Germany

⁴Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS), Shenzhen 518055, China

Correspondence should be addressed to Yongquan Chen; yqchen@cuhk.edu.cn

Received 21 June 2020; Revised 25 September 2020; Accepted 18 October 2020; Published 30 October 2020

Academic Editor: Yanan Li

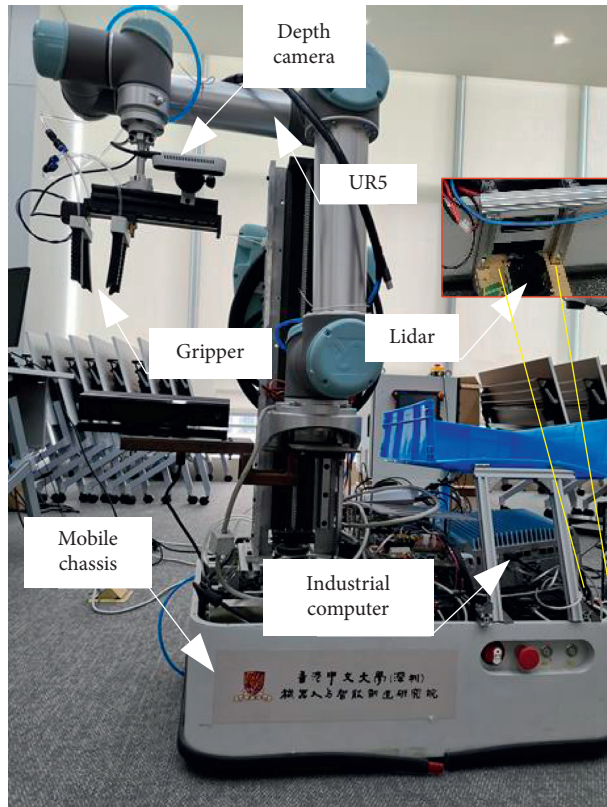
Copyright © 2020 Heng Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile manipulators are widely used in different fields for transferring and grasping tasks such as in medical assisting devices, industrial production, and hotel services. It is challenging to improve navigation accuracies and grasping success rates in complex environments. In this paper, we develop a multisensor-based mobile grasping system which is configured with a vision system and a novel gripper set in an UR5 manipulator. Additionally, an error term of a cost function based on DWA (dynamic window approach) is proposed to improve the navigation performance of the mobile platform through visual guidance. In the process of mobile grasping, the size and position of the object can be identified by a visual recognition algorithm, and then the finger space and chassis position can be automatically adjusted; thus, the object can be grasped by the UR5 manipulator and gripper. To demonstrate the proposed methods, comparison experiments are also conducted using our developed mobile grasping system. According to the analysis of the experimental results, the motion accuracy of the mobile chassis has been improved significantly, satisfying the requirements of navigation and grasping success rates, as well as achieving a high performance over a wide grasping size range from 1.7 mm to 200 mm.

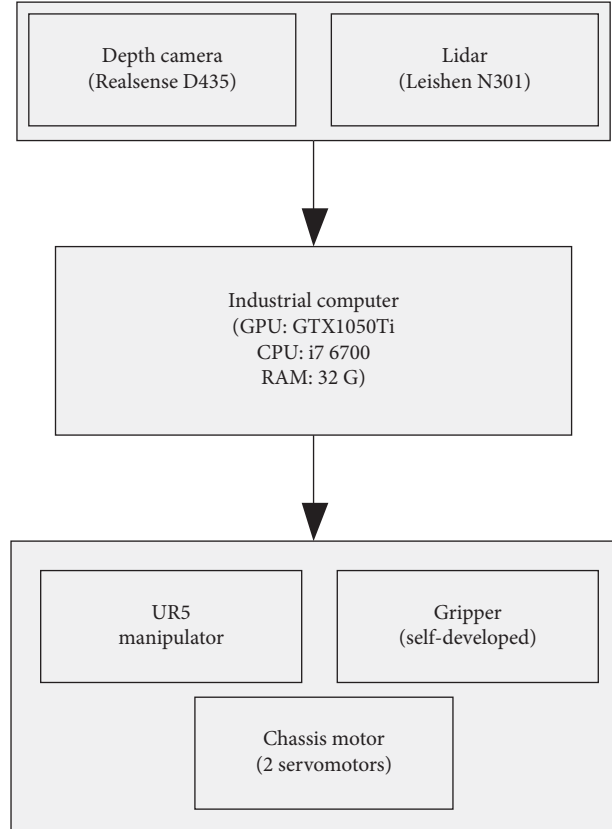
1. Introduction

The applications of mobile grasping robots are becoming more and more extensive. The application fields include hospitals [1], restaurants [2], supermarkets [3], factories [4], and hotels [5]. In life, people often need to transfer items. Currently, mobile grasping robots can help us grasp, transfer, and place these items. Amazon and Jingdong have held several mobile grasping robot competitions [6], which promoted the development of mobile grasping robots. However, there is still room for optimization of the control algorithm for mobile control and the grasping mechanism in the grasping process. Therefore, in this paper, a study of performance improvement is carried out, focusing on these two aspects. The overall diagram and system diagram of the robot are shown in Figure 1.

In terms of path planning, there are several popular local path planners, such as fractional-order PD (FOPD) controller [7], Timed-Elastic-Band [8], and DWA [9]. These methods help mobile robots run more efficiently to avoid obstacles and reach the destination via a global path. Especially, due to the high efficiency of DWA, it is widely used in ROS Navigation. In terms of the soft grippers, there are a variety of grippers that are suitable for different situations. Wang et al. published many papers on pneumatic soft grippers and introduced soft grippers in several different scenes [10–15]. Hao et al. proposed grasping characteristics under variable effective lengths [16]. Zhong et al. introduced a soft pneumatic dexterous gripper with convertible grading modes [17]. Batsuren and Yu proposed a soft robotic gripper with chambered fingers for performing in-hand



(a)



(b)

FIGURE 1: Mobile grasping robot. (a) Photo of the mobile grasping robot. (b) Sensor fusion-based mobile grasping system.

manipulation [18]. Fu and Zhang proposed a soft robot hand with a pin-array structure [19]. In a recent review, Yoon summarized the robust responsive materials, fabrication methods, and applications of soft grippers [20].

However, there are still some shortcomings when the robot is in a complex environment. (1) Because of its fixed parameters, it cannot adapt a complex environment well. (2) It runs poorly in a context with dynamic obstacles. As for the first case, adopt the fuzzy logic controller to change weight parameters to improve its performance [21]. With respect to the second one, predict the motion of obstacles in the future to enhance obstacle avoidance ability [22, 23]. Although there are all kinds of grasping structures, there is still much room to improve the size range of grasping [10, 16–18]. Objects with outer packages or smooth surfaces can be directly grasped by suction cups, but some objects are hard to grasp by suction cups, such as towels without outer packages, knitted gloves, cotton toys, smaller spoons, ear-picks, pens, chopsticks, and toothpicks. To solve these problems, in this study, a pneumatic flexible gripper is designed which can be opened or closed, and the grasping of the above items was performed using this gripper. In order to grasp objects in different positions, a mobile chassis is designed [24]. In order to ensure that the mobile robot runs as close to the global path as possible, an error term of cost function based on DWA is proposed and a comparative trial is carried out between our method and DWA running on

ROS [25]. The mobile grasping robotic system designed in this paper uses a combination of the soft gripper, the UR5 manipulator, and the mobile chassis to successfully complete the grasping task.

The innovations and contributions of this paper are as follows: (1) An error term of cost function based on DWA is proposed and a comparative experiment between our method and DWA running on ROS is carried out. (2) A kind of soft gripper for mobile robotic systems with adjustable finger spacing is proposed. The finger structure is optimized, soft grasping is realized, and the range of objects that can be grasped is improved compared with that of the traditional soft gripper [10, 16–18]. (3) The fingers with different shore hardness were analysed by finite element method; then the actual experiment was carried out after printing with 3D printer, and the finger with the largest fingertip force under the same air pressure was selected. (4) The grasping stability coefficient of grasping objects is modelled and the optimal solution is calculated by genetic algorithm. Furthermore, the relationships between the success rate of grasping and grasping stability coefficient, air pressure, and weight are analysed.

The remainder of this paper is arranged as follows. In Section 2, the proposed gripper is introduced. In Section 3, the finite element analysis of a single finger with three kinds of Shore hardness values is carried out. Then, after the fabrication of a single finger with three kinds of Shore

hardness values, the bending angle and fingertip force of the fingers under different air pressures are tested. Additionally, the finger end running track is calculated, and the last finger is selected according to the fingertip force. In Section 4, a new error term of cost function for local path planning is modelled and the grasping model is established. At last, the grasping stability coefficient is calculated. In Section 5, a comparative experiment is carried out to verify our method performance. After the gripper is assembled in the mobile grasping robot, the grasping experiment is carried out, and the results are compared with those of an existing gripper. The paper is concluded in Section 6 with suggestions for future work.

2. The Mobile Grasping System and the Design of the Gripper

2.1. Control of the Mobile Grasping Robotic System. During the process of automatic grasping, when the system receives the grasping command, the mobile chassis first moves to the position where the object is stored, in accordance with the path planning. Then the visual algorithm recognizes the target object to be grasped and sends the corresponding grasping strategy (the grasping strategy includes the chassis position, the grasping path, and the opening size of the grippers). According to the grasping strategy, the system sends the position control command to the microcontroller of the gripper and adjusts the chassis position to a position convenient for grasping. The stepping motor rotates forward or reverses according to the command. At this time, the fingers fixed on the stepper motor remain motionless. The finger fixed on the slider moves away from or near the stepping motor and moves to the designated position according to the number of pulses calculated by the single-chip microcomputer to realize the grasping of objects of different sizes. The UR5 manipulator makes the gripper move to the location of the object, and then the gripper grasps the object. The whole process can realize the grasping of objects of different sizes, as shown in Figure 2.

In terms of the finger spacing control, for example, when grasping bread, the visual algorithm recognizes that the object to be grasped is bread. After analysis, it is found that the optimal grasping size of the gripper is 100 mm. Suppose that the last object grasped is milk, and the opening size of the corresponding gripper is 50 mm. Then, the stepper motor rotates forward, and the fingers fixed on the slider move away from the stepper motor. After sending 5000 pulses, the gripper opens to 100 mm.

2.2. Design Idea of the Gripper. When individuals try to grab large objects, they usually use two hands to clamp the object, as shown in Figure 3. In this paper, according to grabbing habits of individuals, we propose a kind of soft gripper that can simulate human hand grabbing. If this kind of gripper can grab many kinds of goods in daily life, then it will be very helpful for grasping robots.

2.3. Structure Design of Fingers. The finger design structure of Wang et al. [12] is based on the wrinkle shape design idea of the pleated-type morphology of the fluidic elastomer robot. Based on this design structure, many convex points are designed on the cover surface of the finger in this paper. The purpose is to increase the friction with the contact object and improve the reliability of grasping. The design method of the pointed and flat fingertip at the end of the finger uses two opposite fingers to grasp small commodities, such as earpick. A single finger consists of 12 air chambers, each of which is 1.5 mm thick except for the air chambers at both ends. The air chamber width of the designed finger is 5 mm, and the total length of the finger is 88 mm. The single finger section is shown in Figure 4.

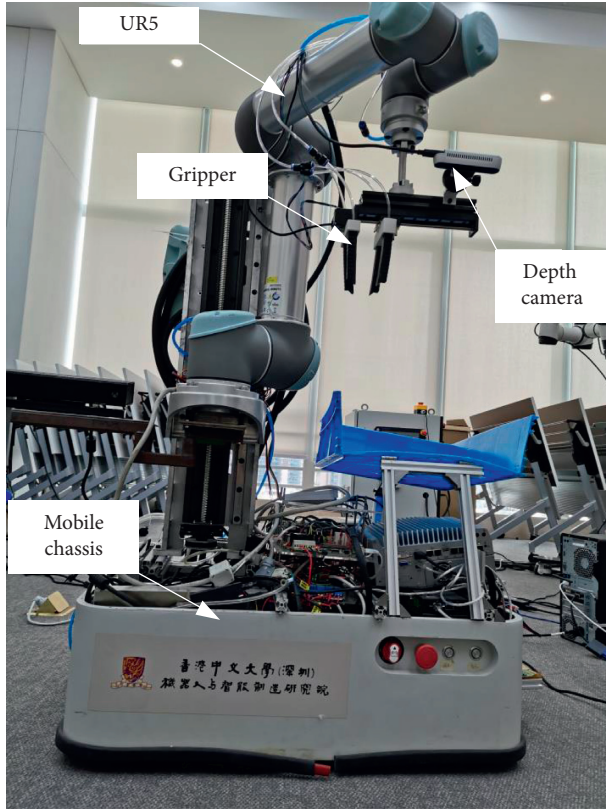
Figure 4(a) depicts a chamber, and Figure 4(b) illustrates a cover. The two parts are bonded together to form a pneumatic finger. The section view of the pneumatic finger is shown in Figure 4(c). When inflated to the pneumatic finger, each air chamber will expand like a balloon. Because of the expansion of each air chamber, the air chambers will repel each other, causing the finger to bend. When multiple fingers are combined, the function of grasping objects can be realized.

2.4. Structure Design of the Gripper. To grasp as many kinds of objects as possible, we choose a linear motor. The principle of the linear motor is to use a stepping motor to drive a screw rod to rotate and use the screw rod to drive a slider to slide. The whole gripper is mainly composed of three parts: a linear motor, four soft fingers, and a connecting bracket between the linear motor and soft fingers. Two of the four soft fingers are fixed on the slider, and the other two are fixed on the stepping motor. Furthermore, the sliding of the slider can change the size of the grasping space, and the linear motor of different lengths can be selected according to the different objects to be grasped. To facilitate the experiment, a 200 mm linear motor is selected in this paper. The overall design structure diagram of the gripper is shown in Figure 5.

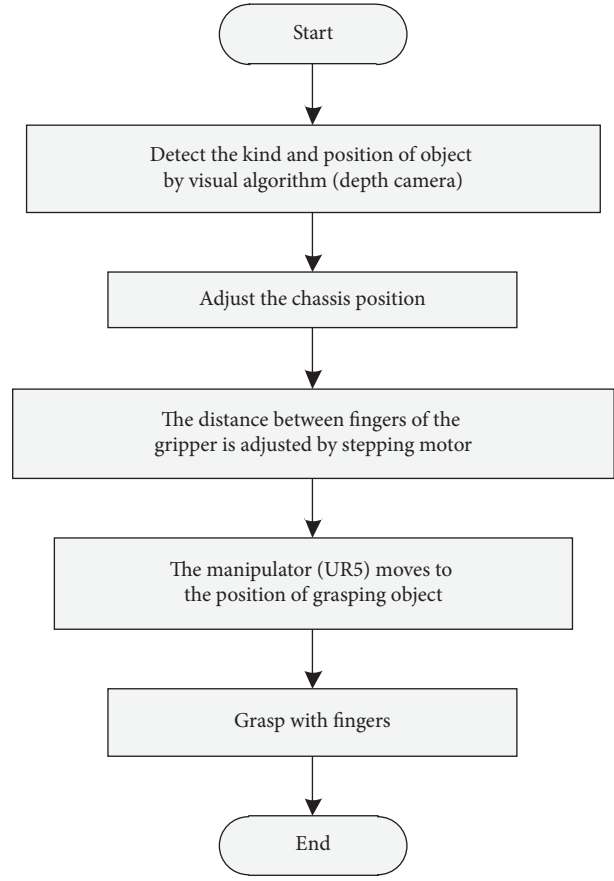
The two ends of the sider rail of the linear motor are equipped with limit switches, which can prevent the slider from running beyond the limit range. The control of the linear motor is controlled by a single-chip microcomputer (STM32F103C8T6). After the serial port of the single-chip microcomputer receives the instruction of running to a certain target position sent by the industrial computer, the single-chip microcomputer program counts the steps of the stepping motor to make the slider move to the target position. Combined with the advantages of the variable grasping space of the slide, the strategy of grasping objects of various sizes is realized.

3. Finger Performance Evaluation

3.1. Finite Element Analysis. To verify the feasibility of the finger design, finite element mechanical analysis is carried out in ABAQUS (Dassault Systèmes, MA). In the simulation, three kinds of 3D printing rubber-like materials are selected,



(a)



(b)

FIGURE 2: Grasping function of the mobile robot. (a) Mobile grasping robotic system. (b) Robot grasping flow chart.

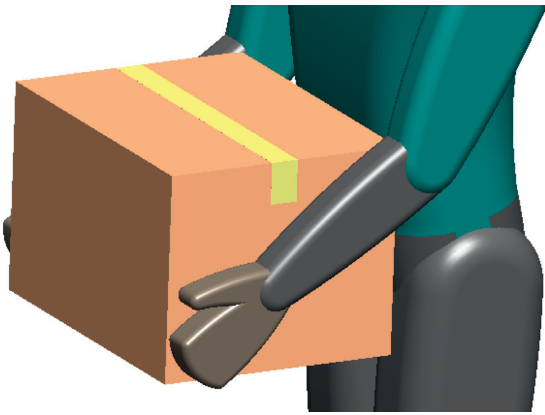


FIGURE 3: Man grabbing a box.

with Shore hardness values of 30, 50, and 70. The material properties in ABAQUS cannot be directly input into the Shore hardness, and the conversion equation from the Shore hardness to Young's modulus is as follows [26]:

$$E(\text{MPa}) = \frac{0.0981(56 + 7.66s)}{0.137505(254 - 2.54s)}, \quad (1)$$

where s denotes the Shore hardness and E denotes Young's modulus. The calculation results are shown in Table 1.

Because rubber-like materials are usually incompressible, we set Poisson's ratio to 0.49. Set the upper surface of the cover and the part bonded to the chamber as the nontensile fibre cloth material; the elastic module is 6.5 GPa, and Poisson's ratio is 0.2. The contact between the sidewalls of the chamber is set to be friction-free. A tetrahedron is used for mesh, and the mesh size is 2 mm. The finite element analysis is performed after all settings are completed.

The rubber fingers with Shore hardnesses of 30, 50, and 70 were analysed by the finite element method. The corresponding analysis results are shown in Figures 6(a)–6(c).

Under the action of gravity, the fingers will bend to a certain extent, and the smaller the Shore hardness is, the larger the bending angle is. The bending angle α defined in this paper is shown in Figure 6(b). The bending angle curves of the three kinds of Shore hardness fingers under different air pressures are shown in Figure 6(d). From the simulation experiment, it is found that, for fingers with the same Shore hardness, the bending angle is approximately proportional to the air pressure. It can also be seen from Figure 6(d) that the smaller the Shore hardness is, the larger the slope is.

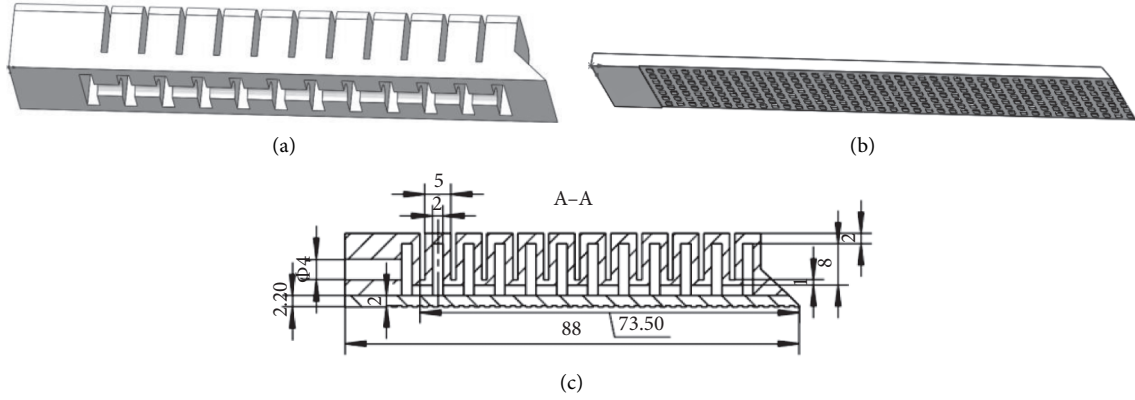


FIGURE 4: The design process of one finger. (a) The air chamber part of the finger. (b) The cover part of the finger. (c) The section view of the finger.

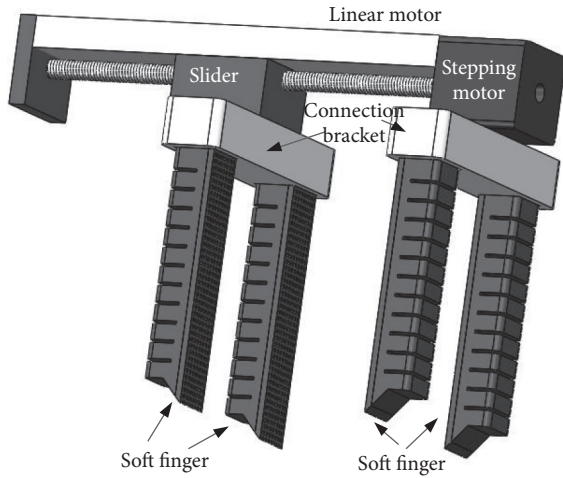


FIGURE 5: Overall design structure diagram of the gripper.

TABLE 1: Material properties.

Shore hardness	Young's modulus (MPa)
30	1.1468
50	2.4661
70	5.5445

By using the linear fitting in Origin software, the linear equations of the relationship between the bending angles α [°] of the fingers with Shore hardnesses of 30, 50, and 70 and the air pressure P [kPa] are fitted as equations (2)–(4), respectively:

$$\alpha = 1.87166P + 4.06845, \quad (2)$$

$$\alpha = 0.97143P + 1.67474, \quad (3)$$

$$\alpha = 0.51338P + 1.105. \quad (4)$$

The linear correlation coefficient R and standard deviation S of the finger fitting equation of the three kinds of Shore hardness values are shown in Table 2. The correlation

coefficient is a statistical index used to reflect the close relationship between variables. The calculation equation of the linear correlation coefficient R is as follows:

$$R(P, \alpha) = \frac{\text{Cov}(P, \alpha)}{\sqrt{\text{Var}[P]\text{Var}[\alpha]}}, \quad (5)$$

where $\text{Cov}(P, \alpha)$ is the covariance of P and α , $\text{Var}[P]$ is the variance of P , and $\text{Var}[\alpha]$ is the variance of α .

Table 2 and Figure 6(d) show that P is approximately linearly proportional to α . In the simulation, equations (2)–(4) can be used to predict the bending angle of fingers under different air pressures, and the hardness of fingers is 30, 50, and 70, respectively.

3.2. Fabrication of a Single Finger and Experiments.

Because of the complex cavity structure of soft grippers, machining is difficult. At present, the commonly used manufacturing processes are shape deposition manufacturing technology (SDM) [27, 28], soft lithography [29, 30], lost-wax casting [31, 32], and 3D printing of composite materials [33]. Because 3D printers have high resolution [10], the performance of each printed finger has better consistency and repeatability. The 3D printer (Objet500 Connex3 system, Stratasys, Minnesota, USA) used in this study can print rubber-like materials with the highest precision of 0.02 mm and a layer thickness of 0.036 mm.

After 3D printing, ERGO5881 glue made in Switzerland is used to tightly stick the cover to the chamber. The air pressure controller uses the active controller PCU-SMN produced by Rochu; its adjustable pressure range is 0 kPa to 120 kPa, and it can also output a nonadjustable negative pressure of -70 kPa. The rubber fingers with Shore hardness values of 30, 50, and 70 are tested under different air pressures. The corresponding experimental results are shown in Figures 7(a)–7(c).

Under the action of gravity, consistent with the conclusion of the simulation results, the lower the Shore hardness is, the greater the bending angle is. The relationships between the bending angle and air pressure for different soft fingers are shown in Figure 7(d). Because the

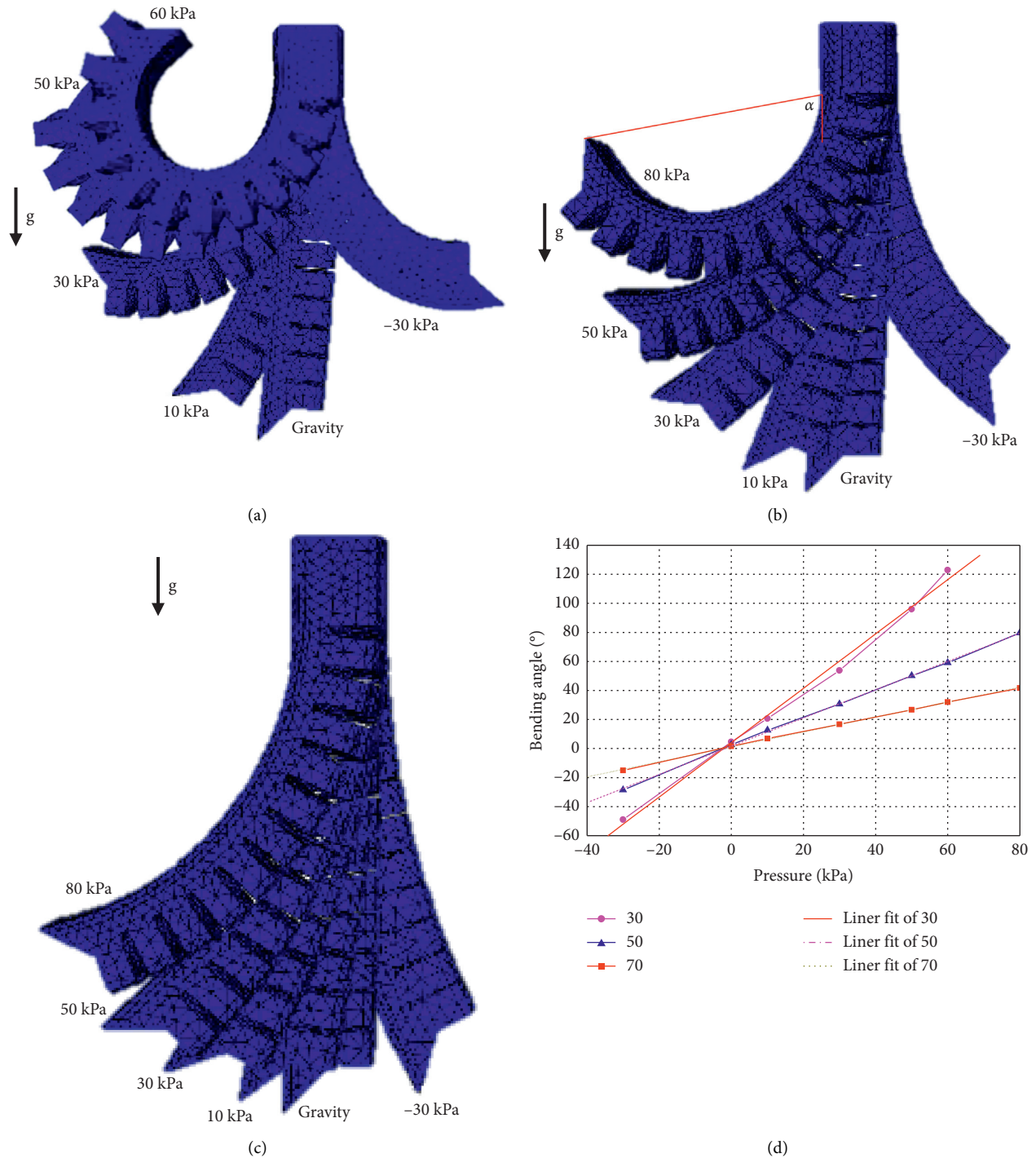


FIGURE 6: Results of the finite element analysis. (a) The deformation of the fingers with Shore hardness of 30 under different air pressures. (b) The deformation with Shore hardness of 50 under different air pressures. (c) The deformation with Shore hardness of 70 under different air pressures. (d) The relationships between the air pressure and bending angle for different soft fingers.

negative pressure is not adjustable in the experiment, the negative pressure in Figure 7(d) is only -70 kPa. There is a certain difference between the actual experimental and simulation results. This difference may be because the simulation model used in ABAQUS needs to be improved or because of some deviation in the hardness of the 3D printed materials. However, the trend of the simulation

results is consistent with the actual results. In the positive pressure part, the bending angle of the fingers with the same Shore hardness is approximately proportional to the air pressure, which is consistent with the conclusion of the simulation. In the actual experiment, the negative pressure is only -70 kPa, and the bending has almost reached the limit at -70 kPa, which does not indicate a difference with

TABLE 2: Correlation coefficient and standard deviation of the three kinds of finger bending angle fitting.

Shore hardness	R	S
30	0.99742	5.04313
50	0.99972	0.9667
70	0.99968	0.545

the simulation results, so only when the air pressure is greater than 0 kPa are the simulation and experimental results compared.

Using the linear fitting function in Origin software, when the air pressure is greater than 0 kPa, the linear equations of the relationship between the three finger bending angles α [°] with Shore hardness values of 30, 50, and 70 and the air pressure P [kPa] are fitted as equations (6)–(8), respectively:

$$\alpha = 3.13366P + 3.08763, \quad (6)$$

$$\alpha = 2.21799P - 5.29767, \quad (7)$$

$$\alpha = 1.82175P - 4.89558. \quad (8)$$

The correlation coefficient R and standard deviation S of the linear fitting equation of the three kinds of Shore hardness fingers are shown in Table 3.

In practical use, we can roughly predict the degree of curvature of the fingers that correspond to Shore hardness values of 30, 50, and 70 by equations (6)–(8), respectively.

To further understand the performance of fingers, the fingertip force is tested. The type of dynamometer used is SF-50, the measuring range is -50 N to 50 N, and the resolution is 0.01 N. The test diagram of the fingertip force is shown in Figure 8(a). The upper end of the finger is fixed to the bracket, and the contact surface of the fingertip and the dynamometer is fully contacted. Under the action of gravity, the indicator number of the dynamometer is 0 N. With an increase in air pressure, the indicator number of the dynamometer begins to increase gradually. The relationship between the fingertip force and air pressure is shown in Figure 8(b).

Figure 8(b) shows that the fingertip force of the same Shore hardness finger is approximately proportional to the air pressure, while the smaller the Shore hardness is, the greater the fingertip force is. Because of the error of the dynamometer or the error of air pressure adjustment, the curve of fingertip force and air pressure does not completely exhibit a straight line.

Using the linear fitting function in Origin, the equations of the relationship between the fingertip force F [N] and the air pressure P [kPa] with Shore hardnesses of 30, 50, and 70 are fitted as equations (9)–(11), respectively:

$$F = 0.00926P - 0.06067, \quad (9)$$

$$F = 0.00857P - 0.05857, \quad (10)$$

$$F = 0.00825P - 0.0875. \quad (11)$$

The correlation coefficient R and standard deviation S of the finger fitting equation of Shore hardnesses of 30, 50, and 70 are shown in Table 4.

In practical use, the fingertip force under different air pressures can be roughly predicted by equations (9)–(11).

3.3. Analysis of the Fingertip Movement Track. During the process of grasping the object, after the robot moves to the grasping position, the finger moves until it contacts the object [34, 35]. For soft fingers, if the influence of the gravity is ignored before touching the object, then the bending of the fingers caused by air pressure can be regarded as a circular arc, as shown in Figure 9.

After the finger is bent, the arc radius is R , the arc length s is the effective bending length of the finger, l is the chord length, and the centre angle of finger bending is φ . The plane coordinate system is established at the first air chamber of the finger root. The y -axis is tangent to the arc, and the x -axis is through the centre of the circle. According to the geometric relationship, we obtain the following:

$$\begin{aligned} w &= \frac{s}{R}, \\ l &= 2R \sin\left(\frac{w}{2}\right), \\ x &= -l \sin\left(\frac{w}{2}\right), \\ y &= -l \cos\left(\frac{w}{2}\right). \end{aligned} \quad (12)$$

The fingertip movement track of angle φ is drawn from 0 to π , as shown in Figure 10.

4. Navigation and Gripper Grasping Model

4.1. Local Path Planning. To improve the mobile robot running efficiency, we need the mobile robot to run as close to the global path as possible; therefore, an error term of the cost function based on DWA is defined, as seen in Figure 11.

In Figure 11, the green ellipse denotes a global path, and each red cross denotes a discrete point on the global path. The grey-dashed line and blue arc denote optional local paths generated by DWA. Taking the blue local path as an example, H_0, H_1, H_2, H_3 and $H_4 \in H$ denote each red cross of the global path. L_0, L_1, L_2, L_3 and $L_4 \in L$ denote each discrete point of a local path. L_0 denotes the current position of the robot and the first point of a local path. H_0 denotes the point that has the shortest distance from the robot along the global path. e_0 denotes the deviation between L_0 and H_0 . e_1, e_2, e_3 , and e_4 are the same. The deviation is exaggerated in the figure. Additionally, $|H_0H_1| = |L_0L_1|$. Five points on the local path are chosen to calculate the deviation relative to the counterpart on the global path. Errors e_1 to e_4 are summed as the error term. Thus, our robot can move along the global path better and improve its efficiency.

The error term of the cost function is

$$J_{\text{error}}(L_k, H) = \gamma(e_1 + e_2 + e_3 + e_4). \quad (13)$$

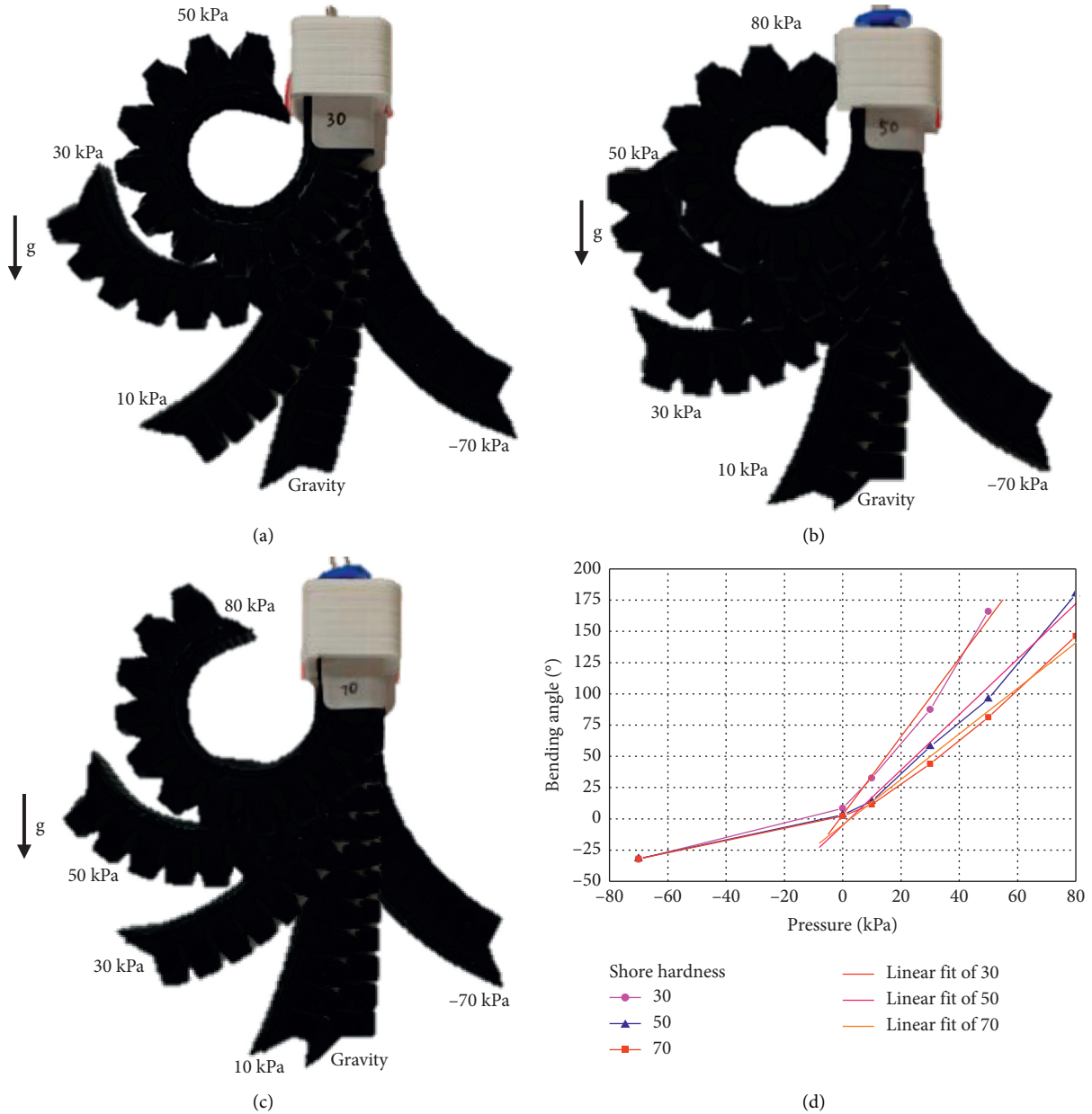


FIGURE 7: Experimental result. (a) The deformation of the fingers with Shore hardness of 30 under different air pressures. (b) The deformation with Shore hardness of 50 under different air pressures. (c) The deformation with Shore hardness of 70 under different air pressures. (d) The relationships between the air pressure and bending angle for different soft fingers.

L_k indicates the k th optional local path generated by DWA, H indicates the global path, and γ is a weight parameter.

4.2. Grasping Modelling. There are many different evaluation schemes for grasp stability. Ko and Chen [36] proposed an optimal grasping manipulation for multi-fingered robots using semismooth Newton method. Li and Sastry [37] established a generalized external force ellipsoid. The minimum singular value of the grasping matrix G is calculated, and the singular value is used to describe the distance of the position of the contact point

from the singular grasping position to quantitatively describe the grasping stability. Xiong and Xiong [38] introduced the grasping stability coefficient w to quantify the grasping stability. This performance index is related to the shape characteristics of the grasping object and the position of the contact point on the surface of the grasping object, that is, the grasping matrix G . The larger the stability coefficient w is, the higher the grasping stability is. w is $0 \leq w \leq 1$ after normalization, and its expression is as follows:

$$w = \sqrt{\det(GG^T)}. \quad (14)$$

TABLE 3: Correlation coefficient and standard deviation of the three kinds of finger bending angle fitting.

Shore hardness	R	S
30	0.99437	9.07033
50	0.99387	9.1398
70	0.99489	6.85057

As shown in Figure 12, the reference coordinate system $\{O\}$ is fixed on the object to be grasped, and the schematic diagram of the contact between the i th finger and the object is shown. The representation r_i is the vector representation of the i th contact point in the reference coordinate system, and $r_i = (x_i \ y_i \ z_i)^T$. $\{C_i\}$ is the contact coordinate system of the i th contact point, whose origin coordinate coincides with the contact point, and the z -axis is along the normal direction of the contact point. Force f_{ci} is the contact force at the i th contact point. The components f_{cix} and f_{ciy} are the tangential components of the contact force, and f_{ciz} is the normal component of the contact force. $f_c = (f_{c1} \ f_{c2} \ \dots \ f_{cm})^T$.

The grasping matrix G can be expressed as follows:

$$G = \begin{pmatrix} I & \dots & I \\ R_1 & \dots & R_m \end{pmatrix} \in R^{6 \times 3m}, \quad (15)$$

where I is the unit matrix and $I \in R^{3 \times 3}$. R_i is

$$R_i = \begin{pmatrix} 0 & -z_i & y_i \\ z_i & 0 & -x_i \\ -y_i & x_i & 0 \end{pmatrix}. \quad (16)$$

The combined external force and the combined external torque acting on the object are called the combined external force spiral F_a as follows:

$$F_a = \begin{pmatrix} F_x & F_y & F_z & M_x & M_y & M_z \end{pmatrix}^T, \quad (17)$$

where $F = (F_x \ F_y \ F_z)^T$ is the combined external force and $M = (M_x \ M_y \ M_z)^T$ is the external moment.

It is assumed that there is friction point contact between the fingers and the object when a multifingered hand grasps the object; that is, the rolling or sliding of the fingers on the object surface is not considered. According to the theory of helix, the contact between fingers and objects can be regarded as a mapping between the force exerted by fingers on the contact point and the resultant helix of a reference point on the object.

$$Gf_c = -F_a. \quad (18)$$

When a manipulator grasps an object, the finger can only "press" the object but not "pull" the object, so the normal component of the contact force exerted by the finger on the object must be positive; that is,

$$-f_{ciz} \leq 0, \quad i = 1, \dots, m. \quad (19)$$

The contact between the finger and object is the friction point contact. To prevent relative sliding between the finger

and object, each contact point must meet the friction constraint condition; that is, the contact force f_{ciz} of each contact point must be located in the friction cone (Figure 13). Specifically, the tangential component of the contact force must be less than or equal to the product of the normal component and friction factor μ ; that is,

$$\sqrt{f_{cix}^2 + f_{ciy}^2} - \mu f_{ciz} \leq 0. \quad (20)$$

where β is the friction angle and $\beta = \arctan(\mu)$.

Finally, the optimization problem of the grasp point based on the grasp stability index can be summarized as follows:

$$\begin{cases} \min & \rightarrow -w = \sqrt{\det(GG^T)} \\ \text{s.t.} & Gf_c = -F_a - f_{ciz} \leq 0 \\ & \sqrt{f_{cix}^2 + f_{ciy}^2} - \mu f_{ciz} \leq 0. \end{cases} \quad (21)$$

4.3. Optimal Grasping Position. The end of the gripper designed in this paper is on a plane. To calculate the optimal distribution of the four fingers, the above grasping model is used to calculate the two-dimensional plane. For example, an object is grasped with an elliptical cross section (without losing generality), assuming that the axis length of the ellipse in the x -axis is a and the axis length in the y -axis direction is b . Then, its polar coordinate form can be expressed as follows:

$$\begin{cases} x = a \cos(\theta), \\ y = b \sin(\theta), \end{cases} \quad (22)$$

where θ is the angle between the line between the origin and the point on the ellipse and the x -axis coordinate.

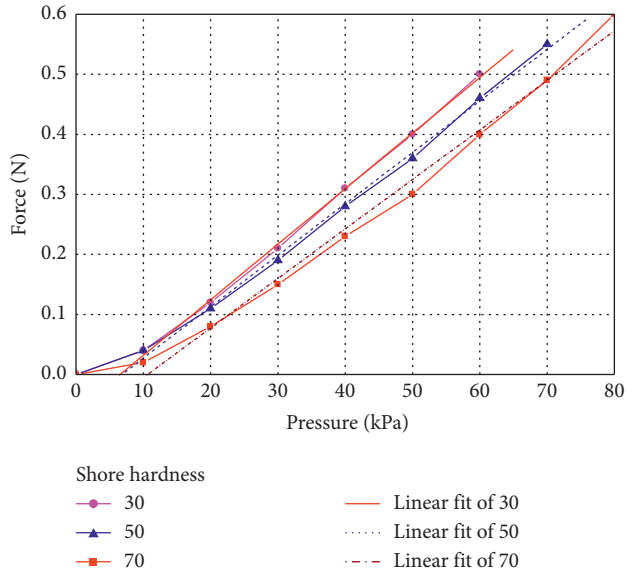
The coordinates of the two-dimensional plane contact point are known; then the grasp matrix G is obtained, and the optimal solution is calculated by the genetic algorithm. After ten groups of experiments, the results of two groups are shown in Figure 14.

Figures 14(a) and 14(b) represent the first group of calculation results, and Figures 14(c) and 14(d) represent the second group of calculation results. In the first and second groups, the ellipse axis lengths in the x -axis direction are 100 mm and 50 mm, respectively. After calculations, it can be seen that the four grasp points are approximately an arithmetic sequence. To facilitate the comparison in the calculation process, a fixed grasp point $\theta = (\pi/4) \approx 0.785$ is set. Then, through the genetic algorithm calculation to 100 generations, it is found that the θ value of the optimal position is approximately the same. In Figure 14, the optimal grasping positions θ are $[0.785, -0.77, -2.317, 2.396]$ and $[0.785, -0.783, 2.359, -2.352]$, respectively. In Figure 14, the grasp stability coefficient w values are 0.99983 and 0.999999, which are close to 1.

In this paper, the distance between two fingers in the unchangeable direction of the designed gripper is 55 mm, and the distance between the two adjacent points in the y -axis direction calculated by two groups of calculation results



(a)



(b)

FIGURE 8: Fingertip force experiment.

TABLE 4: Correlation coefficient and standard deviation of the fingertip force fitting.

Shore hardness	R	S
30	0.99944	0.00647
50	0.99889	0.00956
70	0.99596	0.01969

is approximately 56 mm. Therefore, the grasping point of this kind of object can be grasped in the optimal grasping position, and the grasping near the best position can be realized only by adjusting the distance in the x -axis direction. Because the fingers are flexible, during the process of contacting the object surface, they will be in close contact with the object surface under the control of air pressure. In the next structural design, we will add the rotation function of fingers, which may adjust the finger surface to grasp the object.

The gripper designed in this paper cannot reach the optimal grasping point for all objects because the gripper can only change the space between fingers in one direction. However, this does not mean that the gripper cannot successfully grasp the object. When the optimal position cannot be grasped, the value of the grasping stability coefficient w is slightly smaller, and most objects can also be successfully grasped.

5. Experiment and Analysis

5.1. Path Planning Experiment. A comparative experiment is carried out between our method and DWA running on ROS to test the performance. The starting point and goal point remain the same. For the mobile robot motion parameters, see Table 5.

The mobile robot running process can be seen in Figure 15, and the result is shown in Figure 16.

In Figure 16(a), the starting point is $(0.1, -0.19)$, and the goal point is $(6.5, -1.45)$. The blue line denotes a global path generated by the A* algorithm, and the red- and blue-dashed lines denote DWA and our method, respectively. It is clear that our method, which takes an error term into account, yields a better result. In Figure 16(b), the cumulative error is recorded when using different methods. Generally, the error of our method is less than that of DWA. In particular, if the global path is a zigzag pattern, our method will perform better.

5.2. Grasping Experiment. After the experimental analysis of the fingertip force and bending angle of a single finger, it is necessary to further complete the grasping experiment to test the grasping performance. The tracheas of the four fingers are connected together by a quick connector to realize the synchronous bending and grasping function of the four fingers. In the selection of fingers, because fingers with a

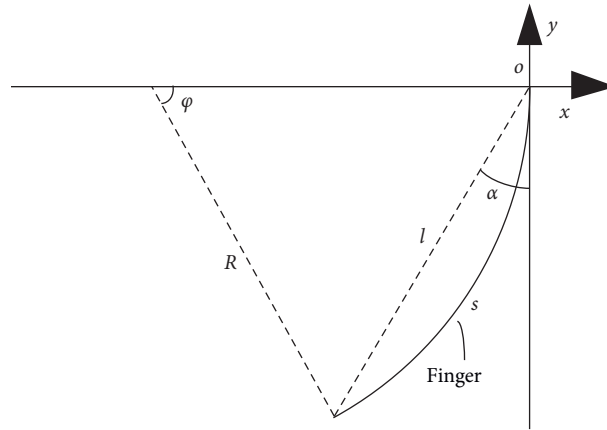


FIGURE 9: Finger bending analysis.

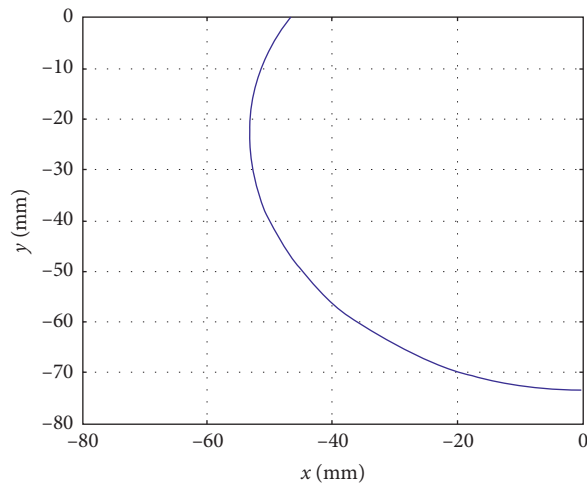


FIGURE 10: Fingertip trajectory.

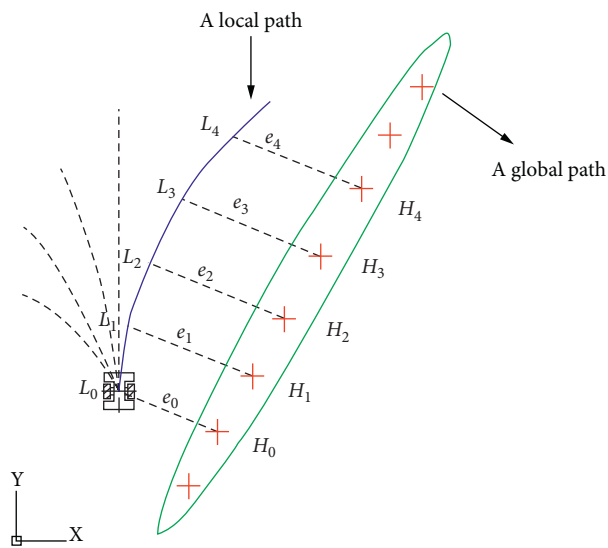


FIGURE 11: An error term definition.

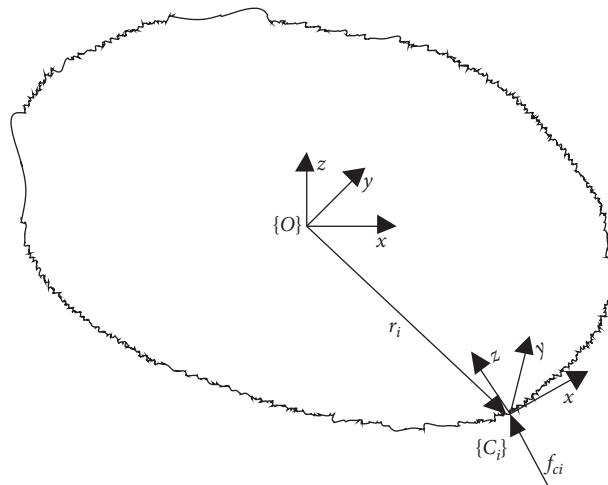


FIGURE 12: Grasp system model.

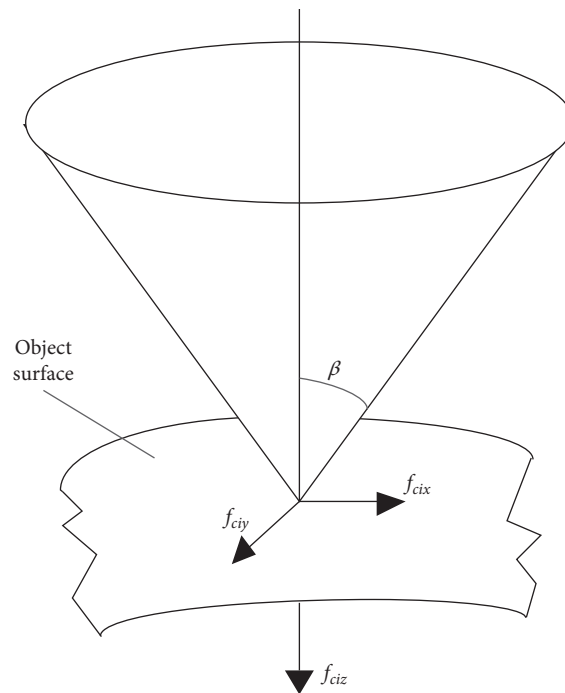


FIGURE 13: Friction model of the contact point.

Shore hardness of 30 have a greater fingertip force and are softer under the same air pressure, four fingers with a Shore hardness of 30 are selected to form a grasping hand. To verify the grasping performance, 56 kinds of object grasping experiments were conducted, and some grasping results are shown in Figure 17.

Different air pressures are used to grasp different objects; that is, larger air pressures are used for heavier objects, and smaller air pressures are used for lighter objects. The grasping objects include barrelled instant noodles, tissues, a gamepad, steamed cakes, towels, a sponge, an earpick, a toothpick, and even an old business card. During the grasping process, the advantages of the soft gripper are fully

embodied with an automatically adjustable finger spacing, which can carry out the grasping of a very small toothpick (1.7 mm in diameter) to a large barrelled instant noodle container (an upper cover with approximately 150 mm diameter). In this experiment, the largest grasp size is a sponge, and its length and width are both 200 mm.

The success rate of the objects in Figure 17 is measured. Each object is grasped ten times and held in the air for more than 10 seconds as a successful attempt. The success rate of grasping is shown in Table 6.

The stability coefficient w is related to the position and shape of the object. Because the finger space of the gripper designed in this paper can only change the finger space in

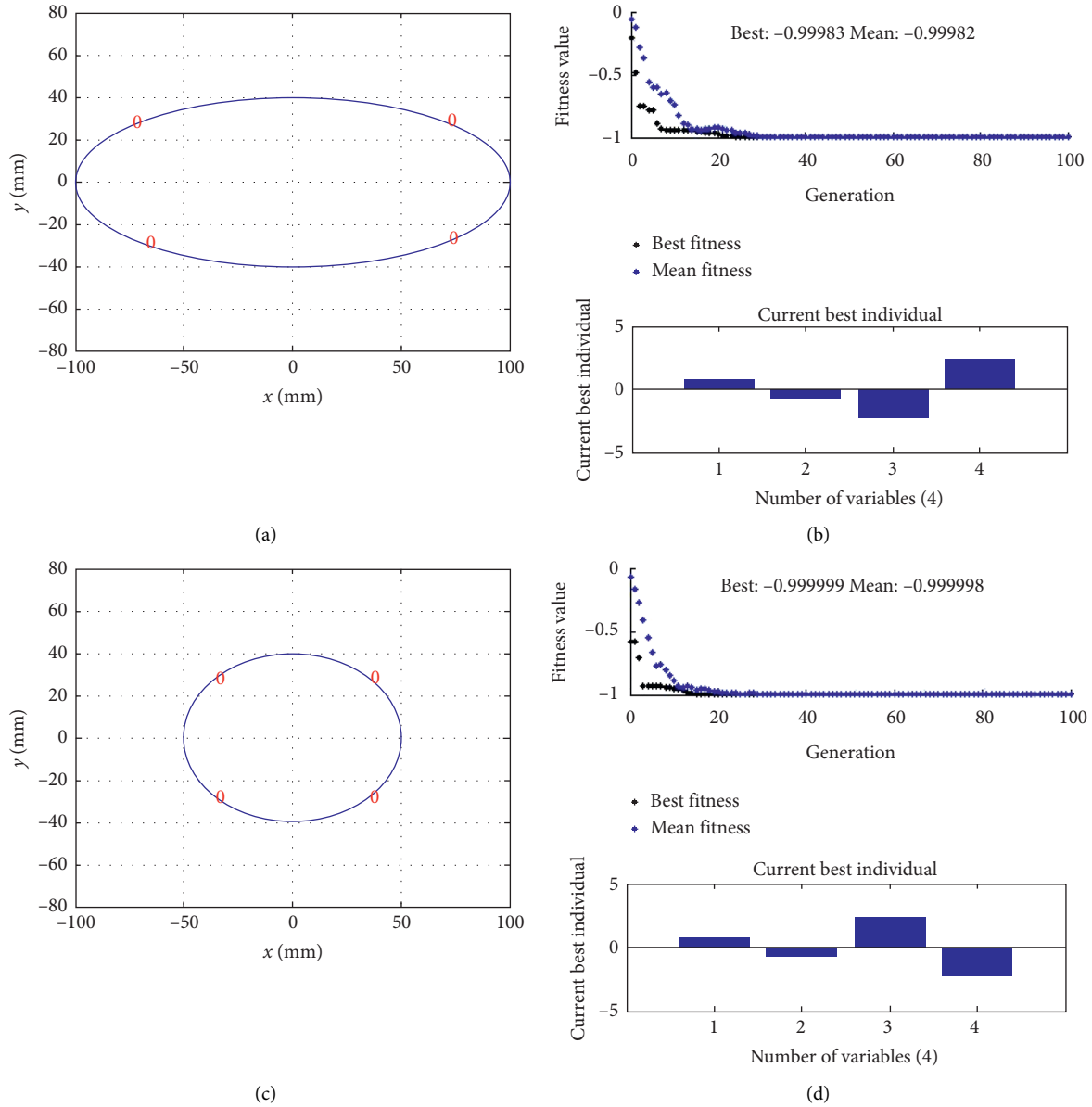


FIGURE 14: Optimal solution calculation. (a) Ellipse $a = 100$ mm and $b = 40$ mm; the calculated optimal grasp position. (b) Ellipse $a = 100$ mm and $b = 40$ mm; the upper side is the relationship between the fitness value and generation, and the lower side is the value of variable q in the optimal solution. (c) Ellipse $a = 50$ mm and $b = 40$ mm; the calculated optimal grasp position. (d) Ellipse $a = 100$ mm and $b = 40$ mm; the upper side is the relationship between the fitness value and generation, and the lower side is the value of variable q in the optimal solution.

TABLE 5: Correlation coefficient and standard deviation of the fingertip force fitting.

Parameter name	Value
v_{\max}	0.5 m/s
w_{\max}	3 rad/s
\dot{v}_{\max}	0.5 m/s ²
\dot{w}_{\max}	3 rad/s ²
v_{\min}	0 m/s
w_{\min}	-3 rad/s
\dot{v}_{\min}	-2 m/s ²
\dot{w}_{\min}	-4 rad/s ²

one direction, not all objects can reach the optimal position. Because toothpicks and earpicks are relatively small, they can be grasped with two fingers, which are not comparable to the stability coefficient of four fingers, and the stability coefficient of the two fingers is not calculated. The shape of the gamepad is more complicated, and it is dragged up when grasping, so its stability coefficient is not calculated. According to the analysis of the data, the success rate of grasping is positively related to the grasping stability coefficient and grasping air pressure and negatively related to weight. Therefore, when the stability coefficient decreases,

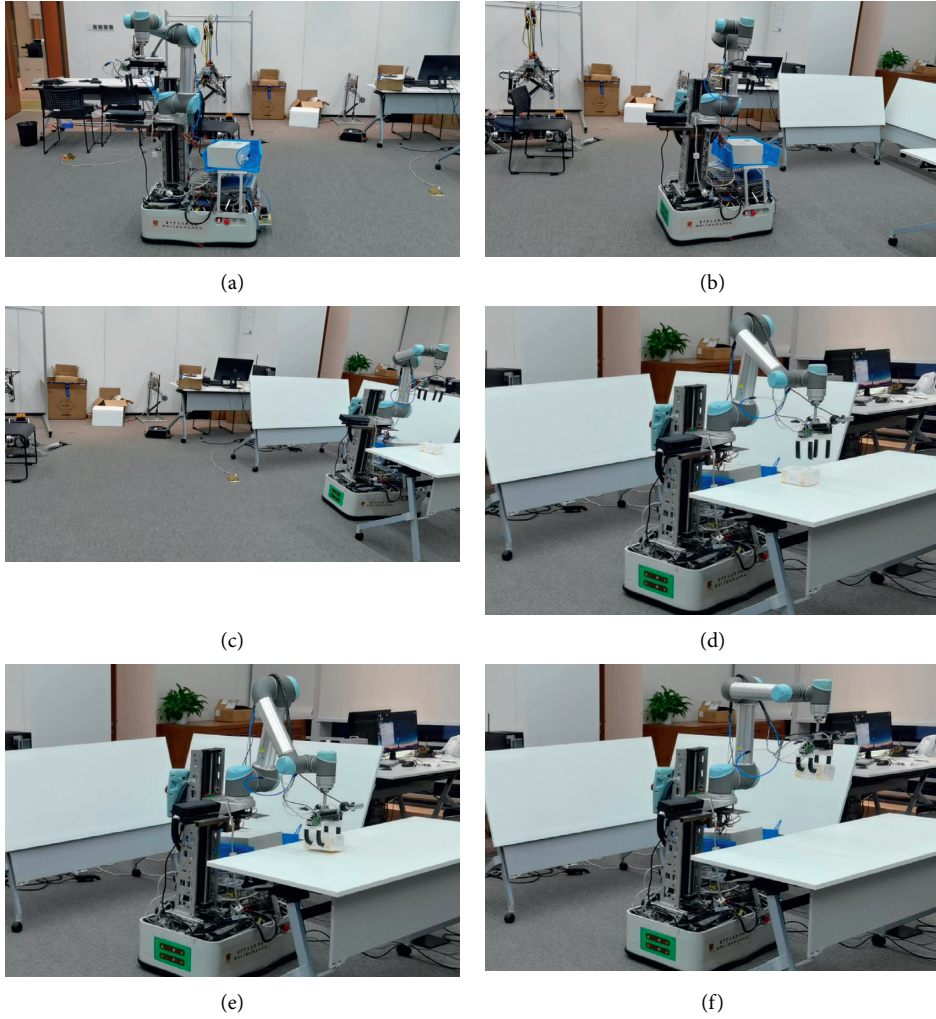
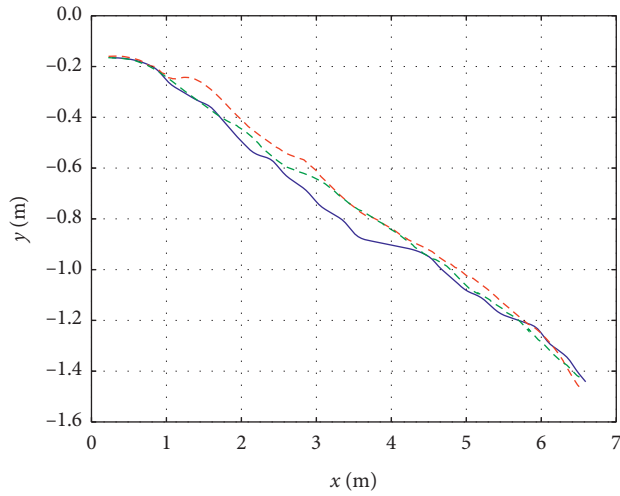
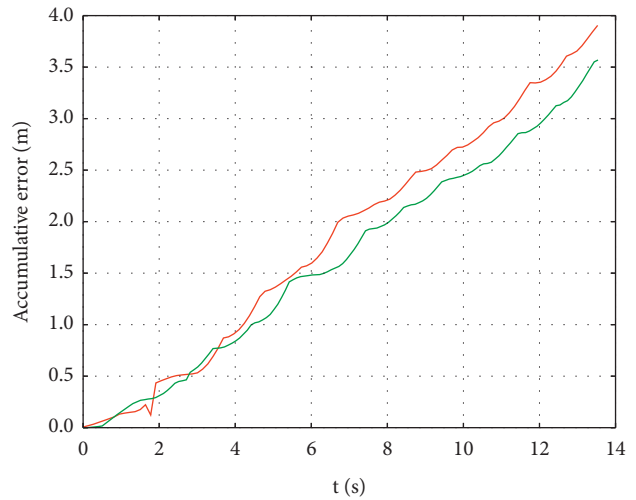


FIGURE 15: The grasping process of the mobile grasping robot. (a), (b), and (c) move to the destination of the item. (d), (e), and (f) arrive at the destination for grasping.



— A* route
 - - - DWA
 - - - Our method

(a)



— DWA
 — Our method

(b)

FIGURE 16: Experiment results. (a) Running trajectory. (b) Cumulative error.

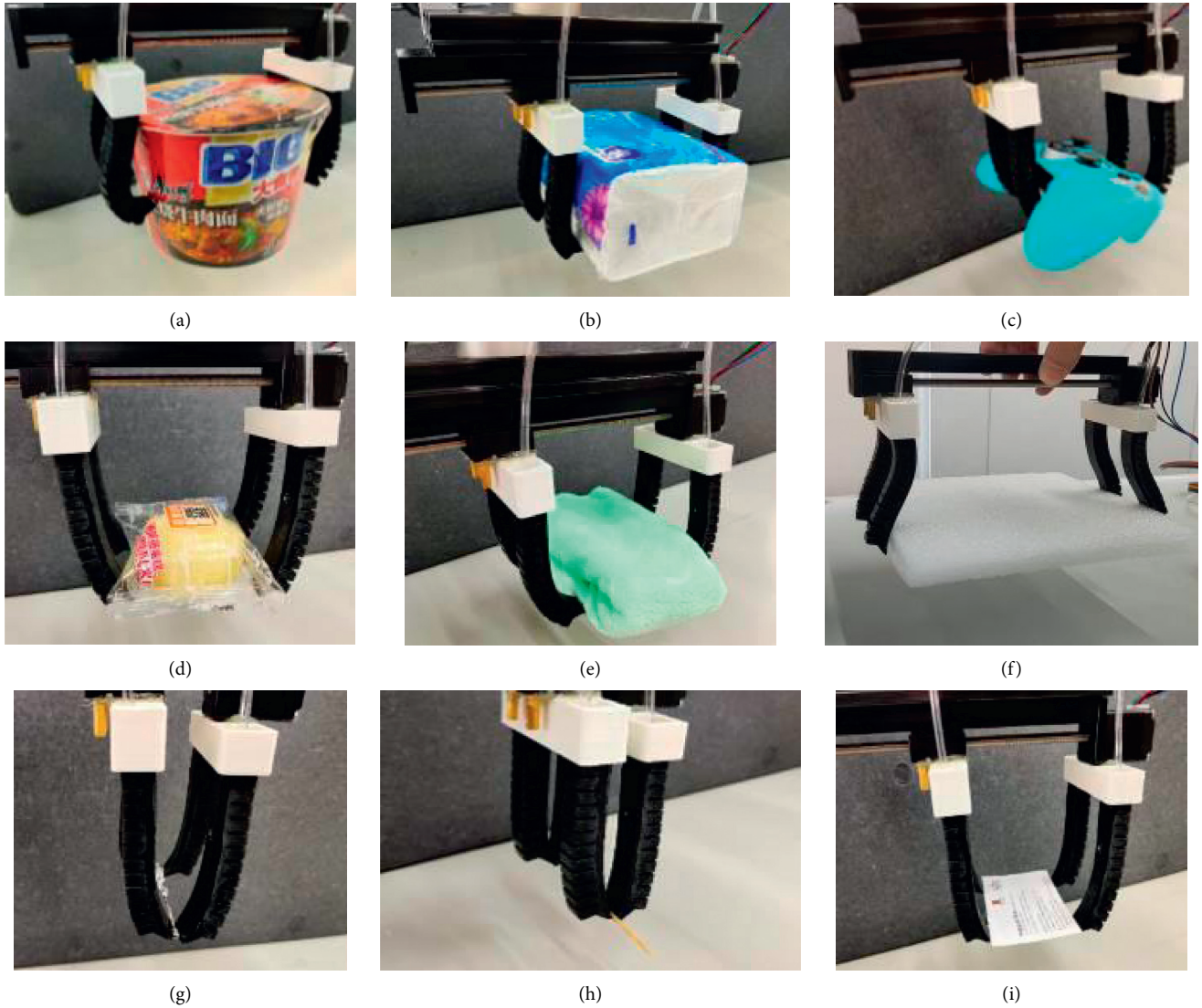


FIGURE 17: Grasping experiment. (a) Instant noodles. (b) Tissues. (c) Gamepad. (d) Steamed cake. (e) Towel. (f) Sponge. (g) Earpick. (h) Toothpick. (i) Business card.

TABLE 6: The success rate of grasping.

Object	Size (mm)	Weight (g)	Pressure (kPa)	Success rate (%)	w
Instant noodles	$\varphi 150 \times \varphi 120 \times 110$	191	60	70	0.7441
Tissue	$134 \times 98 \times 72$	132	50	90	0.8536
Gamepad	$154 \times 104 \times 56$	262	50	80	—
Steamed cake	$102 \times 60 \times 44$	27	30	100	0.9962
Towel	$140 \times 72 \times 45$	31	40	100	0.9648
Sponge	$200 \times 200 \times 15$	10	30	90	0.5114
Earpick	$\varphi 1.8 \times 75$	4	15	100	—
Toothpick	$\varphi 1.7 \times 62$	<1	10	90	—
Business card	$90 \times 56 \times 0.4$	1	15	100	0.9998

TABLE 7: Comparison of grasping results.

Reference	Min size (mm)	Max size (mm)
Our gripper	1.7	200
[16]	2	170
[17]	10	100
[18]	24	100
[10]	24	100

the success rate of grasping does not necessarily decrease. For example, although the stability coefficient w of the sponge is small, its weight is only 10 g, so the success rate can reach 100%. The success rate of the barrel surface with a large weight, small stability coefficient, and smooth surface is the lowest, only 70%. The success rate of small weight objects with a large stability coefficient is above 90%.

5.3. Comparison Analysis. The grasping results are compared with the existing references, and the comparison table is shown in Table 7.

It can be seen from Table 7 that there is a great improvement in the grasp size. References [11–15] are used to grasp specific objects, and the grasping range of [10, 16–18] is not as large as the grasping range designed in our study.

6. Conclusion and Future Work

In this paper, a new error term of the cost function is applied to DWA to improve the mobile robot navigation performance. It can make the robot travel along the global path as much as possible, reducing unnecessary running paths in the process of movement and improving the running efficiency. Additionally, a kind of pneumatic soft gripper for mobile robotic systems with automatically adjustable finger spacing is proposed. The gripper is mainly composed of four pneumatic soft fingers, a linear motor, and a connecting bracket. The fingertips of the gripper are designed to be pointed and flat in shape, and the purpose of this design is to achieve small object grasping. The free control of the distance between fingers is realized through the slide rail controlled by the stepping motor to realize the grasping strategy from small to large. In addition, the grasping stability coefficient of the grasping objects is modelled and calculated. The work of a single finger mainly includes the following: (1) In terms of the fabrication technology, the soft finger is printed via a 3D printer, which has a printing accuracy of 0.02 mm, and can print rubber-like materials with different Shore hardnesses. This printer provides a convenient and quick finger fabrication method for the experiment and speeds up the experimental progress. (2) With respect to the finger bending angle analysis, nonlinear finite element analysis is carried out by using the finite element analysis software ABAQUS, and the relationship between the finger bending and air pressure is calculated. The relationship between the bending angle and the air pressure and the relationship between the bending angle and the Shore hardness are almost the same in the actual experiment and simulation. (3) In the fingertip force test, it is

found that the smaller the Shore hardness is, the greater the fingertip force is. Combined with all the experimental results, the finger assembly with a Shore hardness of 30 was selected to carry out the grasp experiment. Finally, through a variety of different shapes and weights of the object used in the grasp experiment, the advantages of the gripper designed in this paper for grasping small to large objects are verified.

Currently, the fingertip force of the fingers studied in this paper is small, so it is difficult to grasp heavy objects. Under an air pressure of approximately 50 kPa, the pneumatic finger exhibits great bending, and it is difficult for the finger surface to have a large area contact with objects with relatively flat surfaces. Most of the objects are grasped by the fingertip, which plays a major role, without spreading the force evenly to the whole finger. In future research, we will further improve the finger's grasping performance for grasping heavy objects. We also need to improve the freedom of the finger movement, such as increasing the function of rotating fingers, increasing the multiple directions of movement, and other functions. Thus, most of the objects can be grasped in the manner of an optimal grasping position. Finally, the success rate of grasping is improved.

Data Availability

The data in this paper are obtained through experiments and references. Data are also available from the corresponding author via e-mail (yqchen@cuhk.edu.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (U1613216) and Shenzhen Fundamental Research grant (JCYJ20180508162406177) from the Chinese University of Hong Kong, Shenzhen. This work was also partially supported by funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society and the Zhejiang Lab's International Talent Fund for Young Professionals.

References

- [1] M.-S. Kang, C. Ihm, J. Lee, E.-H. Choi, and S. K. Lee, "A study on object recognition for safe operation of hospital logistics robot based on IoT," *The Journal of the Institute of Internet Broadcasting and Communication*, vol. 17, no. 2, pp. 141–146, 2017.
- [2] F. Lee and S. Ivanov, "Understanding the robotic restaurant experience: a multiple case study," 2020.
- [3] F. Coelho, S. Relvas, and A. P. F. D. Barbosa-Póvoa, "Simulation of an order picking system in a manufacturing supermarket using collaborative robots," in *Proceedings of the 32nd Conference on Modelling and Simulation*, pp. 83–88, Wilhelmshaven, Germany, May 2018.
- [4] A. Lourenço, F. Marques, R. Mendonça et al., "On the design of the ROBO-PARTNER intra-factory logistics autonomous robot," in *Proceedings of the 2016 IEEE International*

- Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2647–2652, IEEE, Budapest, Hungary, October 2016.
- [5] W. J. Lee, S. I. Kwag, and Y. D. Ko, “Optimal capacity and operation design of a robot logistics system for the hotel industry,” *Tourism Management*, vol. 76, Article ID 103971, 2020.
 - [6] A. Zeng, K. T. Yu, S. Song et al., “Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge,” in *Proceedings of the 2017 IEEE international Conference on Robotics and Automation (ICRA)*, pp. 1386–1383, IEEE, Singapore, May 2017.
 - [7] L. Zhang, L. Liu, and S. Zhang, “Design, implementation, and validation of robust fractional-order PD controller for wheeled mobile robot trajectory tracking,” *Complexity*, vol. 2020, Article ID 9523549, 12 pages, 2020.
 - [8] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.
 - [9] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
 - [10] Z. Wang, Y. Torigoe, and S. Hirai, “A prestressed soft gripper: design, modeling, fabrication, and tests for food handling,” *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1909–1916, 2017.
 - [11] Z. Wang, D. S. Chathuranga, and S. Hirai, “3D printed soft gripper for automatic lunch box packing,” in *Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 503–508, IEEE, Qingdao, China, December 2016.
 - [12] Z. Wang and S. Hirai, “A 3D printed soft gripper integrated with curvature sensor for studying soft grasping,” in *Proceedings of the 2016 IEEE/SICE International Symposium on System Integration (SII)*, pp. 629–633, IEEE, Sapporo, Japan, December 2016.
 - [13] Z. Wang and S. Hirai, “Soft gripper dynamics using a line-segment model with an optimization-based parameter identification method,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 624–631, 2017.
 - [14] Z. Wang and S. Hirai, “Geometry and material optimization of a soft pneumatic gripper for handling deformable object,” in *Proceedings of the 2018 IEEE international conference on robotics and biomimetics (ROBIO)*, pp. 612–617, IEEE, Kuala Lumpur, Malaysia, December 2018.
 - [15] Y. Kuriyama, Y. Okino, Z. Wang et al., “A wrapping gripper for packaging chopped and granular food materials,” in *Proceedings of the 2019 2nd IEEE international conference on soft robotics (RoboSoft)*, pp. 114–119, IEEE, Seoul, South Korea, April 2019.
 - [16] Y. Hao, Z. Gong, Z. Xie et al., “A soft bionic gripper with variable effective length,” *Journal of Bionic Engineering*, vol. 15, no. 2, pp. 220–235, 2018.
 - [17] G. Zhong, Y. Hou, and W. Dou, “A soft pneumatic dexterous gripper with convertible grasping modes,” *International Journal of Mechanical Sciences*, vol. 153–154, pp. 445–456, 2019.
 - [18] K. Batsuren and D. Yun, “Soft robotic gripper with chambered fingers for performing in-hand manipulation,” *Applied Sciences*, vol. 9, no. 15, p. 2967, 2019.
 - [19] H. Fu and W. Zhang, “The development of a soft robot hand with pin-array structure,” *Applied Sciences*, vol. 9, no. 5, p. 1011, 2019.
 - [20] C. K. Yoon, “Advances in biomimetic stimuli responsive soft grippers,” *Nano Convergence*, vol. 6, no. 1, pp. 1–14, 2019.
 - [21] O. A. Abubakr, M. A. K. Jaradat, and M. A. Hafez, “A reduced cascaded fuzzy logic controller for dynamic window weights optimization,” in *Proceedings of the 2018 11th International Symposium on Mechatronics and its Applications (ISMA)*, IEEE, Sharjah, UAE, March 2018.
 - [22] M. Missura and M. Bennewitz, “Predictive collision avoidance for the dynamic window approach,” in *Proceedings of the 2019 international conference on robotics and automation (ICRA)*, pp. 8620–8626, IEEE, Montreal, Canada, May 2019.
 - [23] H. Su, Y. Hu, H. R. Karimi et al., “Improved recurrent neural network-based manipulator control with remote center of motion constraints: experimental results,” *Neural Networks*, vol. 131, pp. 291–299, 2020.
 - [24] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, “Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments,” *Sensors*, vol. 20, no. 7, p. 1880, 2020.
 - [25] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, “Deep neural network approach in robot tool dynamics identification for bilateral teleoperation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2943–2949, 2020.
 - [26] I. M. Meththananda, S. Parker, M. P. Patel, and M. Braden, “The relationship between Shore hardness of elastomeric dental materials and Young’s modulus,” *Dental Materials*, vol. 25, no. 8, pp. 956–959, 2009.
 - [27] J. Gafford, Y. Ding, A. Harris et al., “Shape deposition manufacturing of a soft, atraumatic, and deployable surgical grasper,” *Journal of Mechanisms and Robotics*, vol. 7, no. 2, 2015.
 - [28] K.-J. Cho, J.-S. Koh, S. Kim, Y. Hong, and S.-H. Ahn, “Review of manufacturing processes for soft biomimetic robots,” *International Journal of Precision Engineering and Manufacturing*, vol. 10, no. 3, pp. 171–181, 2009.
 - [29] Y. Xia and G. M. Whitesides, “Soft lithography,” *Annual Review of Materials Science*, vol. 28, no. 1, pp. 153–184, 1998.
 - [30] A. D. Marchese, R. K. Katzschmann, and D. Rus, “A recipe for soft fluidic elastomer robots,” *Soft Robotics*, vol. 2, no. 1, pp. 7–25, 2015.
 - [31] B. S. Homberg, R. K. Katzschmann, M. R. Dogar et al., “Haptic identification of objects using a modular soft robotic gripper,” in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1698–1705, IEEE, Hamburg, Germany, September 2015.
 - [32] R. K. Katzschmann, A. D. Marchese, and D. Rus, “Hydraulic autonomous soft robotic fish for 3D swimming,” *Experimental Robotics*, pp. 405–420, Springer, Berlin, Germany, 2016.
 - [33] J. T. Muth, D. M. Vogt, R. L. Truby et al., “Embedded 3D printing of strain sensors within highly stretchable elastomers,” *Advanced Materials*, vol. 26, no. 36, pp. 6307–6312, 2014.
 - [34] H. Su, C. Yang, and G. Ferrigno, “Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
 - [35] W. Qi, H. Su, and A. Aliverti, “A smartphone-based adaptive recognition and real-time monitoring system for human activities,” *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 5, pp. 414–423, 2020.
 - [36] C. H. Ko and J. S. Chen, “Optimal grasping manipulation for multifingered robots using semismooth Newton method,”

Mathematical Problems in Engineering, vol. 2013, Article ID 681710, 9 pages, 2013.

- [37] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 32–44, 1988.
- [38] C. Xiong and Y. Xiong, "Stability index and contact configuration planning for multifingered grasp," *Journal of Robotic Systems*, vol. 15, no. 4, pp. 183–190, 1998.

Research Article

A Novel Human-Like Control Framework for Mobile Medical Service Robot

Xin Zhang ¹, Jiehao Li ², Wen Qi ², Xuanyi Zhou ², Yingbai Hu ³, Hao Quan ²,
and Zhen Wang ²

¹Soochow University, Suzhou 215000, China

²Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano 20133, Italy

³Department of Informatics, Technical University of Munich, Munich 85748, Germany

Correspondence should be addressed to Wen Qi; wen.qi@polimi.it

Received 17 June 2020; Revised 6 August 2020; Accepted 3 October 2020; Published 27 October 2020

Academic Editor: Yanan Li

Copyright © 2020 Xin Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, as a highly infectious disease of novel coronavirus (COVID-19) has swept the globe, more and more patients need to be isolated in the rooms of the hospitals, so how to deliver the meals or drugs to these infectious patients is the urgent work. It is a reliable and effective method to transport medical supplies or meals to patients using robots, but how to teach the robot to the destination and to enter the door like a human is an exciting task. In this paper, a novel human-like control framework for the mobile medical service robot is considered, where a Kinect sensor is used to manage human activity recognition to generate a designed teaching trajectory. Meanwhile, the learning technique of dynamic movement primitives (DMP) with the Gaussian mixture model (GMM) is applied to transfer the skill from humans to robots. A neural-based model predictive tracking controller is implemented to follow the teaching trajectory. Finally, some demonstrations are carried out in a hospital room to illustrate the superiority and effectiveness of the developed framework.

1. Introduction

In the past few months, a novel coronavirus has resulted in an ongoing outbreak of viral pneumonia in the world [1, 2]. More than 98000 people are known to be infected, and over 3400 deaths have been reported [3]. The symptoms of these patients include high temperature, cough, shortness of breath, and headache, and it is a highly infectious disease [4]. More and more patients need to be isolated in independent rooms of the hospitals. How to transport meals and medicines to patients and reduce the infection for medical staff simultaneously is a hot issue. It is a safe and useful method to transport medical supplies or meals to patients using robots. Therefore, this paper focuses on how to control the robot to the destination and to enter the door like a human at the same time.

Human-robot skill transfer is currently one of the significant topics in human-assisted systems. Under different external environmental conditions, especially in the hospital room, the main challenge for assisted medical rescue robots

is how to transport the medical supplies [5] safely. The robot system acquires the learning ability through the cognitive knowledge transmitted by humans [6]. At the same time, by learning the approach of human-in-the-loop, the performance of the robot is improved, making the system more intelligent [7, 8].

Human-robot interaction (HRI) is dedicated to the development of more intelligent and anthropomorphic robots, which is a subregion of human-computer interaction that researches the interaction between man and robot [9, 10]. In some hazardous areas, to minimize staff participation, robots are required to perform operations. HRI technology can be used to perform remote operations on robots efficiently [11]. Therefore, not only is it widely used in the research of robotic systems, but it also plays an essential role in the implementation of robotic systems. Furthermore, in some particular activities, human-computer interaction plays an important role. For example, remote operation of human-machine interaction for medical robots is the safest method in medical processes, allowing the user to interact

with the robot through tactile signals for coordinated feedback [12]. In order to improve surgeon performance, an experimental approach to characterize the human-robot-assisted surgery system is discussed. In vision-based robotic control, the visual impedance scheme was used to implement dynamic control at the activity level. To complete the integration of the vision servosystem and the conventional servosystem, Su et al. [13] proposed the visual impedance scheme in the control scheme; namely, the characteristics of the image were applied to the impedance equation. However, the flaw is that this research is still limited in terms of a self-adaptive decision. Besides, two themes are widely used in HRI [14], one of which was the human-computer interaction interface which was represented by the operation of the keyboard, and the other was the human-machine interaction represented by a touch screen operation. Therefore, this document focuses primarily on human-robot interaction with demonstration teaching.

At the same time, another advanced technology for human-machine operation is the skill transmission through demonstration teaching [15, 16], and during this process, the motion control strategies and the generalized output [17] were learned to transfer the motor skills to the robot by the movement of the demonstrator. Behavior perception, behavior representation, and behavior reproduction are three processes that are imitating the process of learning. Some special feature methods can be used to program the learning process, such as Dynamic Motion Primitives (DMP) and Hidden Markov Models (HMM) [18, 19]. The stochastic models, like the Gaussian mixture model (GMM), have some powerful capability to code and process noise so that they have the ability to handle high-dimensional problems more effectively. The trajectory-level representation that is on the basis of the probability model uses the characteristics of the stochastic model to model the motion trajectory, thereby solving the problem more efficiently. Motion trajectory reproduction and motion control belong to the category of behavior reproduction, where trajectory reproduction is a process of transmitting coded data. It is mainly transmitted for some techniques of regression, such as Gaussian Process Regression (GPR) and Gaussian Hybrid Regression (GMR), and the feedback variable is a playback behavior learned from the presenter. In other words, it is a generalized output that maps to robot motion control for motion reproduction [20–22].

The most widely used technologies for scientific research widely used in statistical models of human behavior are machine learning (ML) and deep learning (DL) technologies. In the previous work, in order to compare the recognition rates for identifying human activities, different combined sensors were adopted, and a deep convolutional neural network (DCNN) was applied to the HAR system [23]. In addition, the ML method is used to enhance HAR's adaptive identification and real-time monitoring system to overcome time-consuming strategies. These classifiers have been proven to recognize more human actions in dynamic situations, thereby improving accuracy, enhancing robustness, and achieving time-saving effects [13, 24]. Moreover, a hybrid hierarchical classification algorithm combining DL

and the method that is based on the threshold is proposed to differentiate complex events in order to calculate quickly. Although our previous research has proposed many effective frameworks, most acceptable results have been achieved with limited assumptions and conditions that cannot meet the complex environment in the medical room.

In this article, a novel human-like control framework for mobile medical service robots is presented, where Kinect sensors are used to achieve human activity recognition to generate a designed teaching trajectory. At the same time, the learning technique of dynamic movement primitives (DMP) with the Gaussian mixture model (GMM) is applied to transfer the skill from human to machine. In addition, a neural-enhanced model predictive tracking controller is implemented to follow the teaching trajectory. Finally, some demonstrations are carried out in a medical room to account for the effectiveness and superiority of the framework. The main contribution of this paper is as described below:

- (1) In order to control the mobile service robot to transport medical supplies or meals to the patient who suffers from the new coronavirus, a novel human-like control framework is discussed.
- (2) Kinect sensors technology is applied to the skill transfer of the mobile medical service robot for collecting the movement points, and the method of DMP with GMM is used to congress the points.
- (3) To efficiently track the teaching trajectory under uncertain disturbances, a neural network enhanced predictive tracking control scheme is presented. Also, some demonstrations are carried out to illustrate the developed structure.

The structure of this essay is as described below: Section 2 describes the overview of human-like control. Some demonstrations are discussed in Section 3. Finally, the conclusion and future point are summarized in Section 4.

2. Methodology

2.1. Human Activity Recognition Using Kinect Sensors. Human activity recognition technology can be applied to follow the position of an operator using a Kinect device. As shown in Figure 1, the operator selects the depth message on the Kinect sensor, and the color vision is collected in the Kinect depth image [25]. To effectively construct the movement information, it is combined with the color image and the depth image in such a way that the origin is located at the center of the depth camera. Thus, we assume that the coordination system in camera space follows a right-hand convention [26].

In this case, $\mathcal{M}(k) = (x(k), y(k), z(k))$ represents the three coordinates of the sequence of frame, and we define $D(i, j)$ and $q(i, j)$ as the depth image points and color image, respectively. Then, we can obtain the Bayes rule to evaluate the probability of $\mathcal{M}(t|q)$:

$$\mathcal{M}(t|q) = \frac{(\mathcal{M}(t|q)\mathcal{M}(t))}{\mathcal{M}(q)}, \quad (1)$$

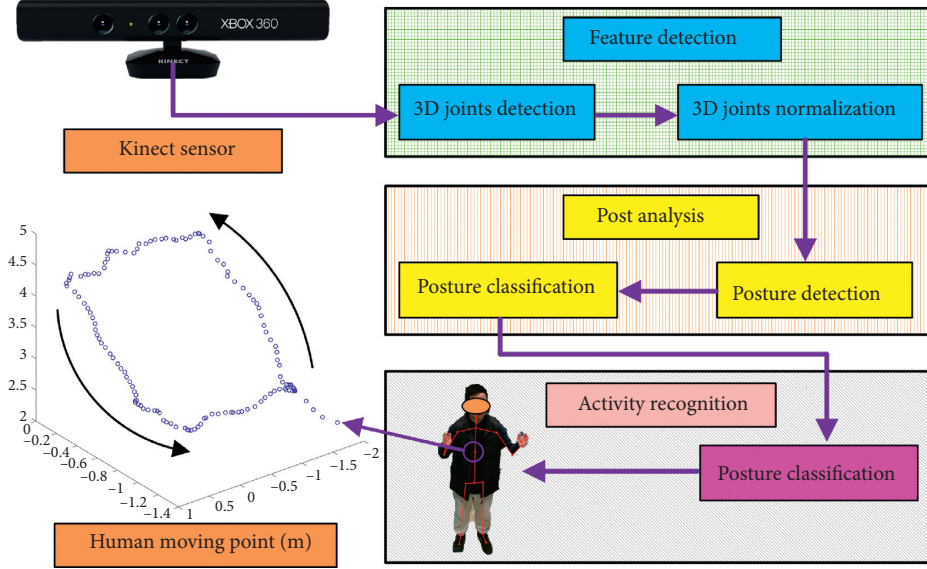


FIGURE 1: The overview of human activity recognition using Kinect sensors.

where $\mathcal{M}(\rho)$ and $\mathcal{M}(\rho|t)$ denote the exercise dataset and prior probabilities of skin color, respectively.

It is on account of self-occlusion or lacking of joint information at some stages that we need to incorporate other functions that can offer data about human shape so as to upgrade the precision of the classifier [6, 27]. We adopt orthogonal Cartesian planes on the depth map to obtain the positive 2D image and the profile obtained. Then, by converting Cartesian coordinates to polar coordinates, the silhouette of a person can be efficiently processed:

$$\begin{aligned} \mathcal{R}_i &= \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2}, \\ \theta_i &= \tan^{-1} \frac{y_i - y_j}{x_i - x_j}, \end{aligned} \quad (2)$$

where (x_i, y_i) and $(\mathcal{R}_i, \theta_i)$ represent the coordinates of the outline of the human body and Radius and angle in polar coordinates, respectively. Besides, (x_j, y_j) is the center coordinate of the human contour. The overall order of each activity with front and side views is averaged, and the average $\mathcal{S}_{\text{mean}}$ from the initial frame to the final frame is defined as follows:

$$\mathcal{S}_{\text{mean}} = \frac{1}{T} \sum_{t=1}^T I(x, y, z, t). \quad (3)$$

2.2. Trajectory Generation via DMP with GMM. After the Kinect device has collected the path information teaching by human demonstration, the mobile robot needs to learn the created trajectory [28]. The teaching trajectory is determined by dynamic movement primitive technology (DMP) and then rebuilt by the Gaussian mixture model (GMM) to generalize the movement trajectory:

$$\Psi(\Phi_l) = \sum_{\xi=1}^{\xi} \Psi_{\xi} \Psi(\Phi_l|\xi), \quad (4)$$

where $\Psi(\xi)$ is the prior probability, $\Psi(\Phi_l|\xi)$ is the conditional probability distribution, which follows the Gaussian distribution, and ξ is the number of Gaussian model distribution.

Therefore, by using a Gaussian mixture model, the entire teaching dataset can be expressed as follows:

$$\begin{aligned} \Psi(\Phi_l|\xi) &= N\left(\Phi_l, \psi_{\xi}, \sum_{\xi}\right) \\ &= \frac{1}{\sqrt{(2\pi)^E |\sum_{\xi}|}} e^{-0.5(\Phi_l - \psi_{\xi})^T \sum_{\xi}^{-1} (\Phi_l - \psi_{\xi})}, \end{aligned} \quad (5)$$

where E is the dimension of the GMR and determined by $\{\pi_{\xi}, \psi_{\xi}, \sum_{\xi}\}$.

The Gaussian distribution can be addressed as

$$\begin{aligned} \Phi_{f,\xi}|\Phi_{s,\xi} &\sim N\left(\psi'_{f,\xi}, \sum'_{f,\xi}\right), \\ \psi'_{f,\xi} &= \psi_{f,\xi} + \sum_{f,s,\xi}^{-1} (\Phi_{s,\xi} - \psi_{s,\xi}), \end{aligned} \quad (6)$$

where $\psi_{\xi} = \{\psi_{f,\xi}, \psi_{s,\xi}\}$ and $\sum_{\xi} = \{\sum_{s,f,\xi}, \sum_{f,s,\xi}\}$.

Therefore, the average ψ'_f and variance \sum'_f of GMR of the number of ξ Gaussian components can be evaluated as

$$\psi'_f = \sum_{\xi=1}^M \eta_{\xi} \psi'_{f,\xi}, \quad \sum'_f = \sum_{k=1}^M \eta_{\xi}^2 \sum_{f,\xi} \zeta_{\xi} = \frac{G(\Phi_s|\xi)}{\sum_{\xi=1}^M G(\Phi_s|i)}, \quad (7)$$

where ψ'_f is the estimation variable and Φ_s is the corresponding space parameter. (Φ'_f, Φ_s) is the generalized

points, which produces a smooth movement trajectory under the covariance constraint \sum_f' .

The GMM model, including a multidimensional probability density function, is composed of multiple Gaussian probability density functions. The Gaussian model is only related to two parameters, the mean and variance. As we all know, different learning mechanisms can directly affect the accuracy, convergence, and stability of the model. Assume that an M -order GMM is weighted and summed with M Gaussian probability density functions:

$$\mathcal{P}(X|\lambda) = \sum_{n=1}^M \Phi_n \mathcal{Q}_n(X), \quad n = 1, 2, \dots, M, \quad (8)$$

where X denotes D dimensional random vector and M is on behalf of the order of the model, while Φ_n represents the weight of each Gaussian component, satisfying $\sum_{n=1}^M \Phi_n = 1$. Furthermore, $\mathcal{Q}_n(X)$ is each Gaussian component, which is a Gaussian probability density function of D dimension and can be expressed as follows:

$$\begin{aligned} \mathcal{Q}_n(X) &= \frac{1}{\sqrt{(2\pi)^{(D/2)} |\sum_n|^{(1/2)}}} \\ & * \exp \left\{ -0.5 (x - \mu_n)^T \sum_{n=1}^M (X - \mu_n) \right\}, \end{aligned} \quad (9)$$

where μ_n is on behalf of the mean vector and \sum_n denotes the covariance matrix. Then, GMM can be expressed by the three parameters of mean vector, covariance matrix, and mixed weight. Therefore, the GMM can be described as

$$\lambda = \left\{ \mu_n, \sum_n, \Phi_n \right\}, \quad n = 1, 2, \dots, M. \quad (10)$$

2.3. Neural Approximation. In order to efficiently transfer the trajectory teaching by human demonstration, it is necessary to control the uncertain disturbance in the trajectory tracking process for the mobile robot [29–31]. To overcome the hidden safety hazards in scooter operation [32–34], a control scheme based on RBFNN was implemented for the elderly walker system. This scheme has certain disturbances and unknown dynamic characteristics. Design a constant smooth function $\mathcal{G}(\mathcal{K})$: $R^q \rightarrow R$ to connect the approximation capability, where the RBFNN control scheme is applied for evaluating the dynamics of uncertainty, such as load friction and mechanism structure [35–37]:

$$\mathcal{G}_m(\mathcal{K}_{in}) = \mathcal{F}^T \Theta(\mathcal{K}_{in}), \quad (11)$$

where $\mathcal{K}_{in} \in \Omega \subset R^q$ represents the input of RBFNN; $\Theta(\mathcal{K}_{in})$ and $\Theta_i(\mathcal{K}_{in})$ are the activation function depending on Gaussian function, respectively; and $\mathcal{F} = [\xi_1, \xi_2, \dots, \xi_m] \in R^m$ represents the weight in the hidden layer:

$$\Theta_i(\mathcal{K}_{in}) = \exp \left[\frac{-(\mathcal{K}_{in} - u_i^T)(\mathcal{K}_{in} - u_i)}{\eta_i^2} \right], \quad (12)$$

where $i = 1, 2, \dots, m$, $u_i = [u_{i1}, u_{i2}, \dots, u_{iq}]^T \in R^q$, and η_i is the variance.

Then, $\Theta(\mathcal{K}_{in})$ can be defined as

$$\|\Theta(\mathcal{K}_{in})\| \leq \omega, \quad (13)$$

where ω is a positive constant.

Then, we have

$$\mathcal{G}_m(\mathcal{K}_{in}) = \mathcal{F}^{*T} \Theta(\mathcal{K}_{in}) + \varepsilon, \quad (14)$$

where \mathcal{F}^* is the desired weight subjected to $\Phi_{\mathcal{K}_{in}} \subset R^q$ and $\|\varepsilon\| \leq \tau$.

Hence, we have

$$\mathcal{F}^* = \operatorname{argmin}_{\mathcal{K}_{in} \in R^q} \left\{ \sup |\mathcal{G}_m(\mathcal{K}_{in}) - \mathcal{F}^T \Theta(\mathcal{K}_{in})| \right\}, \quad (15)$$

where $\Theta(\mathcal{K}_{in})$ denotes the activation function depending on the Gaussian function.

2.4. Neural-Based Model Predictive Tracking Control.

Human motion points are collected by the Kinect sensor, and then the generated trajectory can be obtained with the method of DMP and GMM [18, 38, 39]. Finally, the next task of the mobile robot is to follow the teaching trajectory [40].

Figure 2 exhibits the kinematic model of the mobile medical service robot, where (x_r, y_r) and (x_f, y_f) denote the coordinate of the rear axis and front axis, respectively. P is the circle center, and R denotes the steering radius. L and M represent the wheel trajectory. v_r and v_f denote the rear speed and front speed, respectively. δ_f and φ denote the steering angle and yaw angle, respectively.

The trajectory tracking control of the mobile scroll wheel system can be represented as

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ 0 \end{bmatrix} v_r + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad (16)$$

where $\xi_s = [x_r, y_r, \varphi]^T$ is the system state and $u_s = [v_r, \omega]^T$ is the control state.

Hence, the dynamic model of the mobile robot can be addressed as

$$\begin{aligned} m\ddot{x}_r &= m\dot{y}_r \Phi_c + F_{a1} \cos \delta_f + F_{a2} \cos \delta_f + F_{a3} + F_{a4}, \\ m\ddot{y}_r &= -m\dot{x}_r \Phi_c + F_{b1} \cos \delta_f + F_{b2} \cos \delta_f + F_{b3} + F_{b4}, \\ I_z \ddot{\varphi} &= A(F_{b1} \cos \delta_f + F_{b2} \cos \delta_f) \\ &\quad - B(F_{b3} + F_{b4})M(-F_{a1} \cos \delta_f + F_{a2} \cos \delta_f - F_{a3} + F_{a4}), \end{aligned} \quad (17)$$

where F_{a1} , F_{a2} , F_{a3} , and F_{a4} are the wheel force of left front, right front, left rear, and right front, respectively. Φ_c denotes the center yaw velocity, and I_Z represents the rotational inertia.

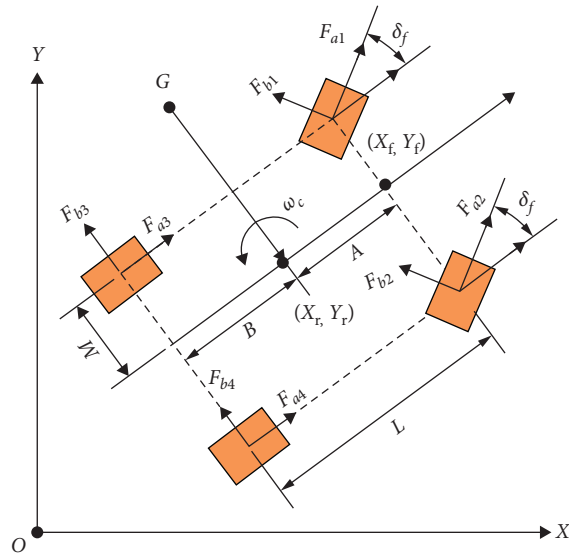


FIGURE 2: The kinematic model and the dynamic model of the mobile medical service robot.

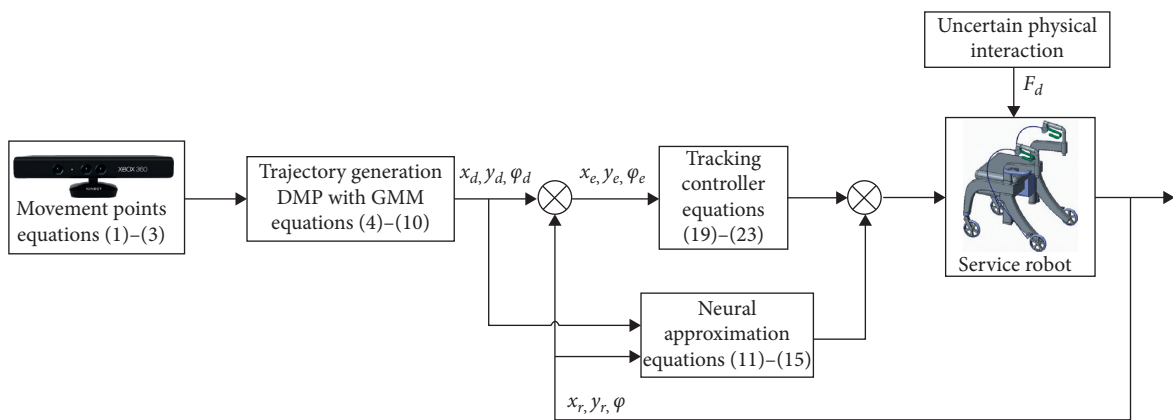


FIGURE 3: Block diagram of neural fuzzy-based tracking control.

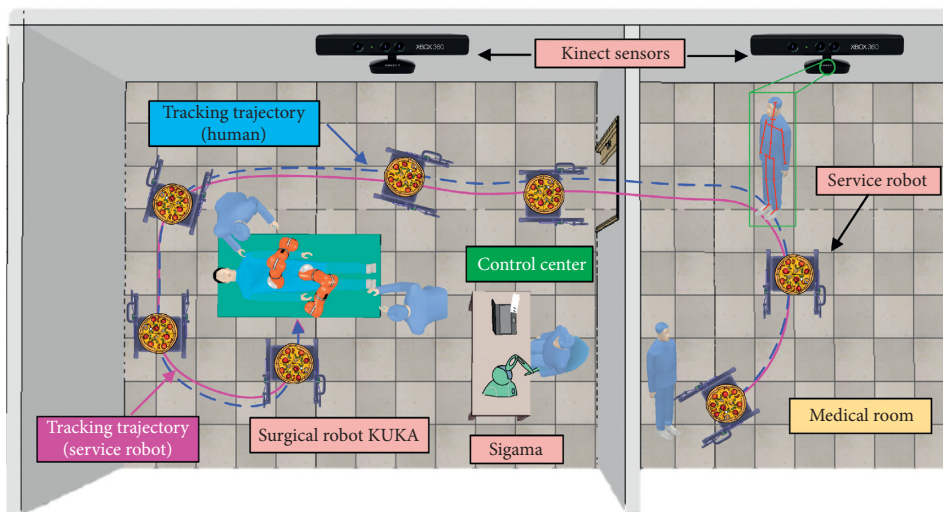


FIGURE 4: The overview scenario of the medical room to transport the meals for patients.

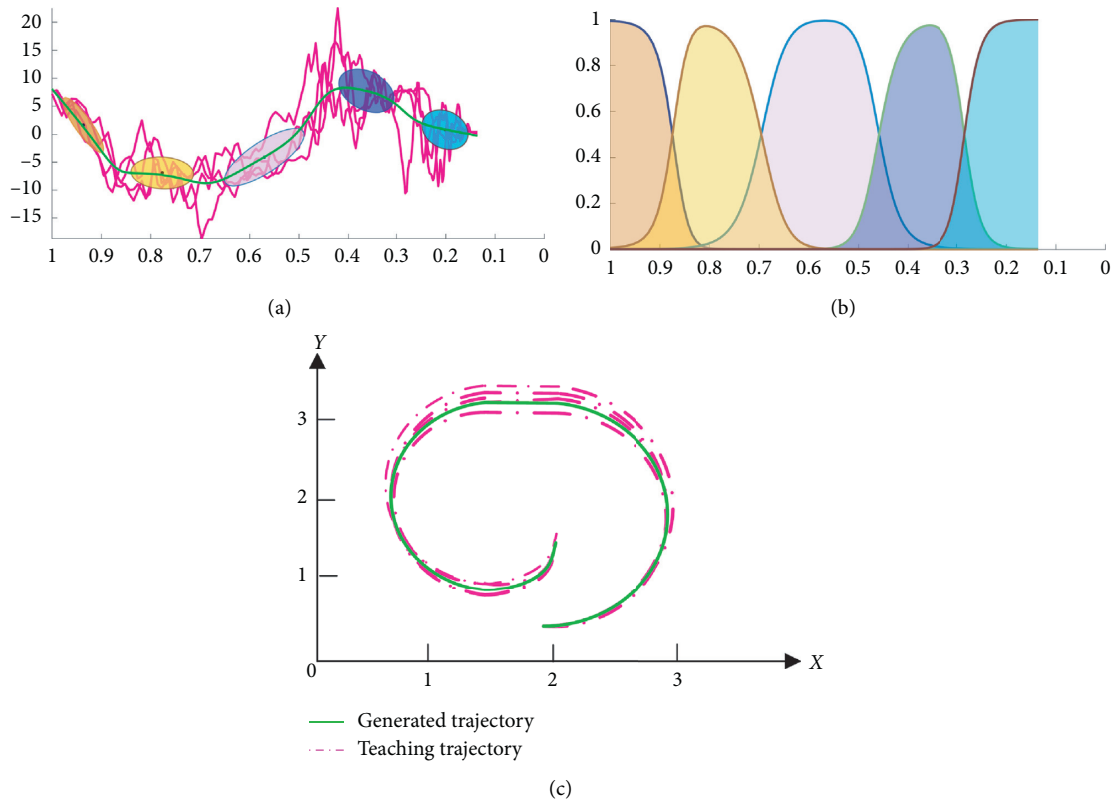


FIGURE 5: The regression result of teaching by demonstration using DMP with GMM (demonstration 1): (a) DMP to encode the trajectory points, (b) Gaussian components of GMM, and (c) regression results.

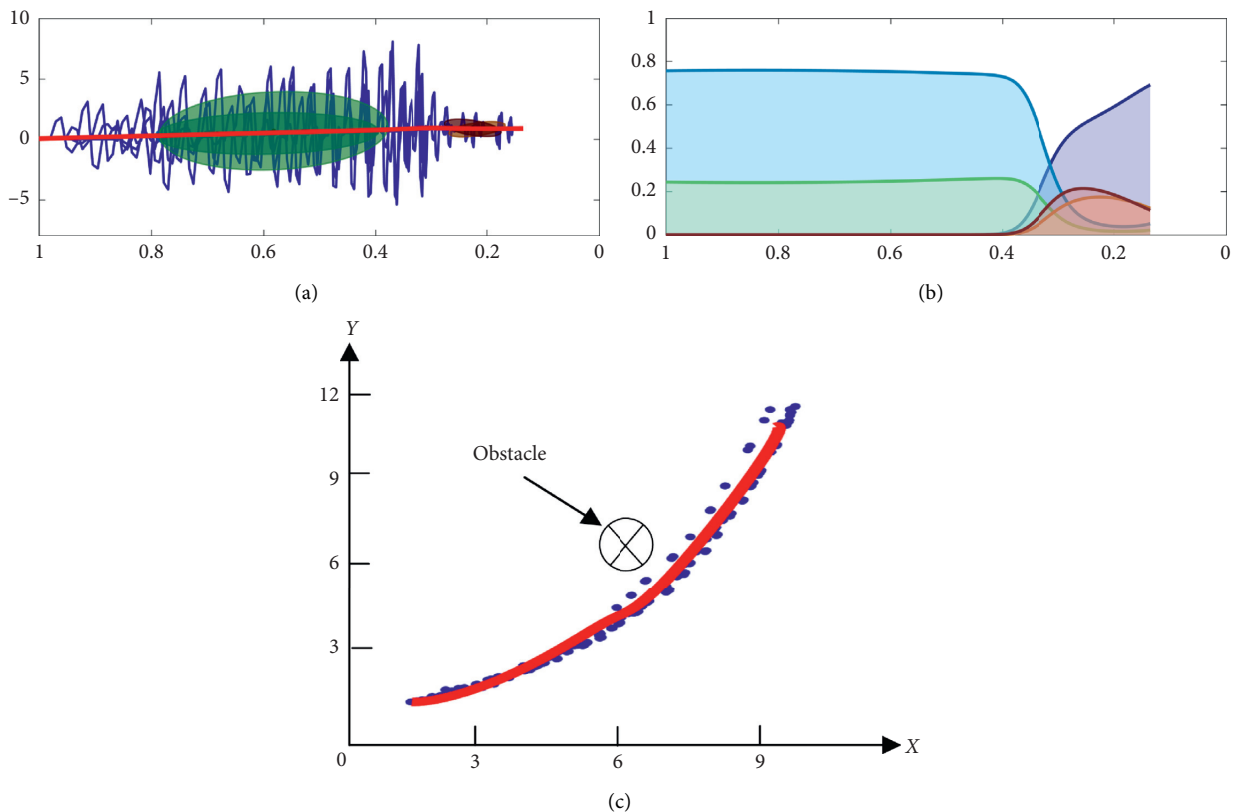


FIGURE 6: The regression result of teaching by demonstration using DMP with GMM (demonstration 2): (a) DMP to encode the trajectory points, (b) Gaussian components of GMM, and (c) regression results.

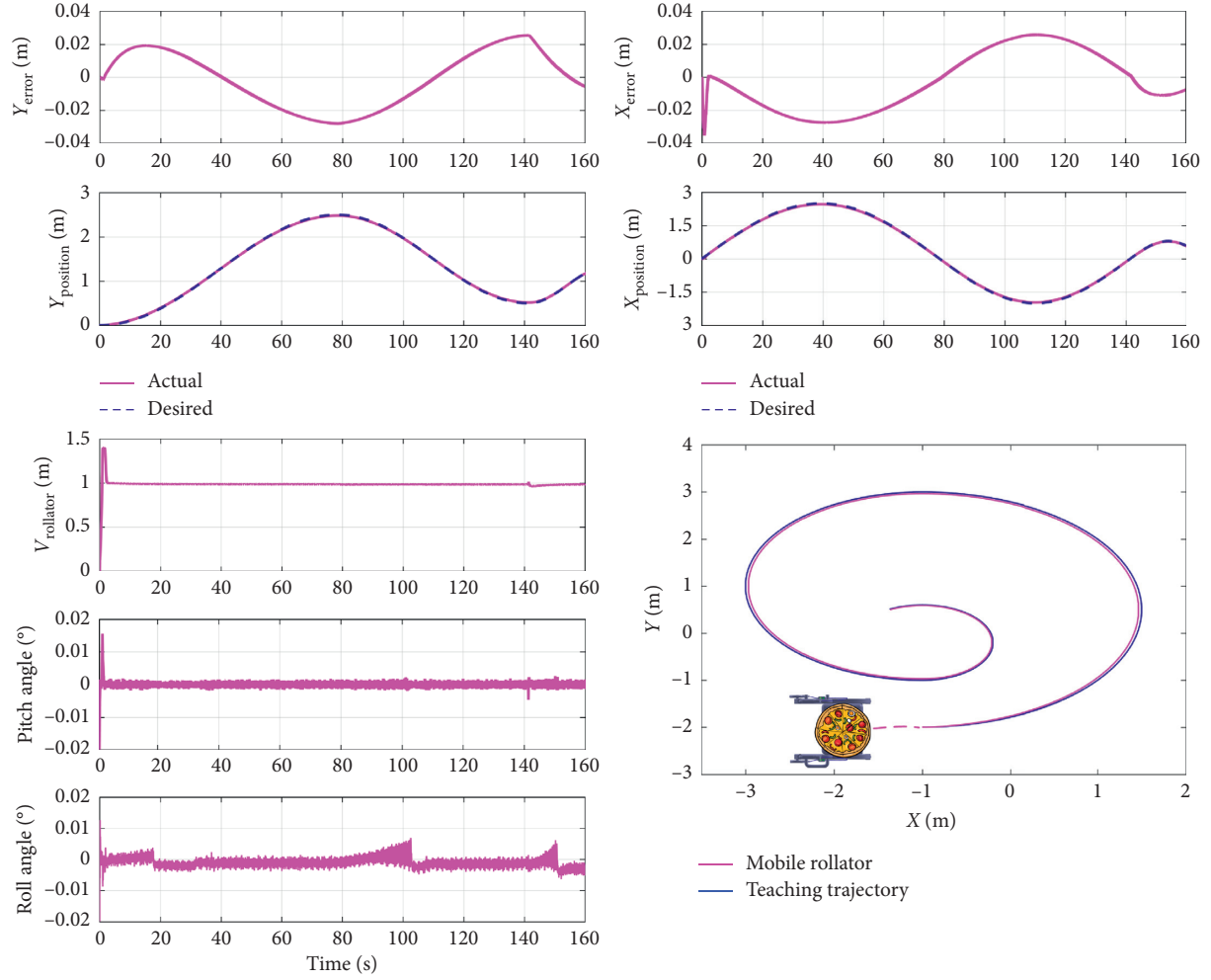


FIGURE 7: Teaching results of demonstration 1 in x -position, y -position, x -error, y -error, robot tracking velocity, roll angle, pitch angle, and tracking performance.

Besides, we assume the following condition to evaluate the lateral force of the robot tire [14]:

$$\begin{aligned}
 F_{b1} &= \psi_{\delta F} \Gamma_{\delta F}, \\
 F_{b2} &= \psi_{\delta B} \Gamma_{\delta B}, \\
 \psi_{\delta F} &= \beta + \frac{M\Phi_r}{v_x} - \delta_f, \\
 \psi_{\delta B} &= \beta + \frac{M\Phi_r}{v_x},
 \end{aligned} \tag{18}$$

where $\psi_{\delta F}$ and $\psi_{\delta B}$ are tire cornering angle. $\Gamma_{\delta F}$ and $\Gamma_{\delta B}$ denote cornering stiffness and β is the slip angle.

The tracking error can be addressed as

$$\dot{x}_e = (\dot{x}_r - \dot{x}_d) = -v_d \sin \varphi_d (x_r - x_d) + \cos \varphi_d (v - v_d),$$

$$\dot{y}_e = (\dot{y}_r - \dot{y}_d) = v_d \cos \varphi_d (y_r - y_d) + \sin \varphi_d (v - v_d),$$

$$\dot{\varphi}_e = (\dot{\varphi} - \dot{\varphi}_d) = \frac{v_d}{L \cos^2 \delta_d} (\delta - \delta_d) + \frac{\tan \delta_d}{L} (v - v_d).$$

(19)

Then, we discretize the error function as

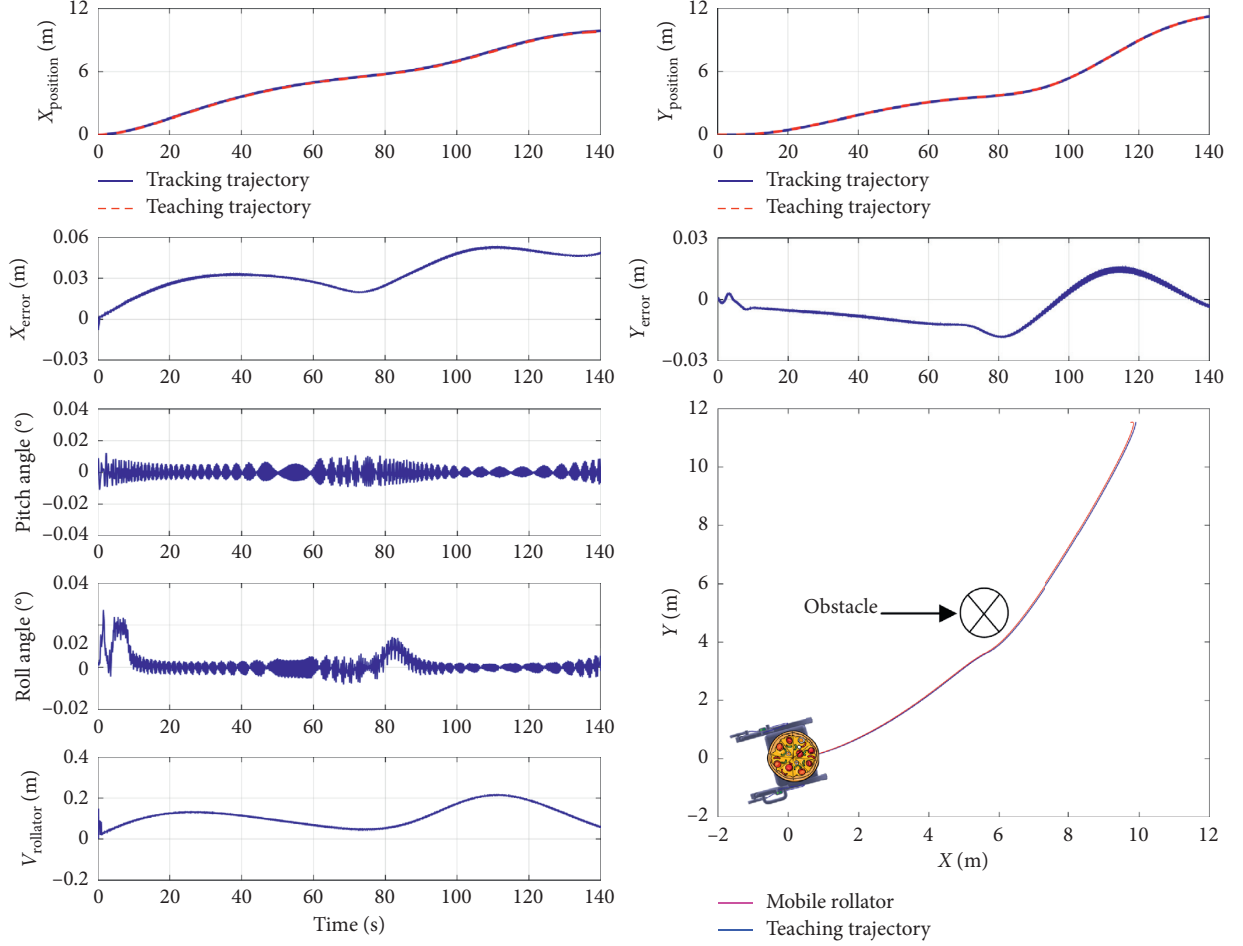


FIGURE 8: Teaching results of demonstration 2 in x -position, y -position, x -error, y -error, robot tracking velocity, roll angle, pitch angle, and tracking performance.

$$\tilde{\mathcal{X}}(n+1) = \mathcal{H}_{n,t} \tilde{\mathcal{X}}(n) + \mathcal{K}_{n,t} \tilde{u}(n), \quad (20)$$

subjected to $\mathcal{H}_{n,t} = \begin{bmatrix} 1 & 0 & -v_d T \sin \varphi_d \\ 0 & 1 & v_d T \cos \varphi_d \\ 0 & 0 & 1 \end{bmatrix}$, $\mathcal{K}_{n,t} = \begin{bmatrix} T \cos \varphi_d & 0 \\ T \sin \varphi_d & 0 \\ (\tan \delta_d / L) T & (v_d / L \cos^2 \delta_d) T \end{bmatrix}$ and T is the sampling time.

In order to reliably and smoothly grasp the desired trajectory, state errors and control parameters must be constrained:

$$\mathcal{V}(n) = \sum_{l=1}^N \tilde{\mathcal{X}}^T(n+l|n) \mathcal{Z}(n+l) + \tilde{u}^T(n+l-1) \mathcal{F} \tilde{u}(n+l-1), \quad (21)$$

where \mathcal{Z} and \mathcal{F} are weighting factors, N_p is the prediction horizon, and N_e is the control horizon. Then, the actual control variable can be determined as

$$u(t) = u(t-1) + \Delta u_t^*. \quad (22)$$

It is on account of considering the safety and stability of the robot [41, 42] that it is of necessity to restrict the control limit and control increment. Combined with the mobile robot system, the control constraint can be presented as follows:

$$\begin{bmatrix} -1.0 \\ -45 \end{bmatrix} \leq u \leq \begin{bmatrix} 1.0 \\ 45 \end{bmatrix}, \quad (23)$$

$$\begin{bmatrix} -0.1 \\ -0.2 \end{bmatrix} \leq \Delta U \leq \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}.$$

Based on the overall control scheme, the framework of neural approximation for tracking control using DMP with GMM is shown in Figure 3. The Kinect sensor detects the human movement points and then generates the teaching trajectory using the technology of DMP with GMM. Then, the neural-based model predictive tracking controller is carried out to realize the path following.

3. Results and Discussions

In this section, the overview scenario of the medical room to transport the meals for patients is presented in Figure 4.

There are two Kinect sensors (XBOX 360) used in this demonstration. The surgical medical robot (LWR4+, KUKA, Germany) is used to feed the meals to patients, where the haptic manipulator (SIGMA 7, Force Dimension, Switzerland) is applied to control the KUKA arm remotely. The main purpose of this demonstration is that the developed mobile medical service robot can safely transport the meals or medicines to the medical bed like a human without collisions.

The Kinect sensor can detect human activity points and generate a teaching trajectory based on the method of DMP and GMM. Then, the mobile robot can follow the teaching trajectory via human demonstration. The result of the learning method, including the DMP, GMM, and the regression result of teaching trajectory, is displayed in Figures 5 and 6. It is noted that there are two demonstrations considered in this section, which aims to evaluate the proposed framework for mobile medical service robot in skill transfer via teaching by demonstration.

Figure 7 exhibits teaching results of demonstration 1 in x -position, y -position, x -error, y -error, robot tracking velocity, roll angle, pitch angle, and tracking performance. It can be concluded that the mobile medical service robot can follow the teaching trajectory collected by Kinect sensors. The y -position error and x -position error can be constrained in a reasonable range within ± 0.03 meters, indicating that the mobile robot can avoid the medical devices and surgeons. On the other hand, because of the neural-based predictive tracking controller, the velocity response of the mobile robot under uncertain disturbance is smooth. In particular, the roll angle and pitch angle can maintain a stable range.

In addition, to further illustrate the improvement of skill transfer scheme using multisensors fusion technology, demonstration 2 to avoid obstacles such as medical devices and medical staff is carried out. Figure 8 displays the teaching performance in x -position, y -position, x -error, y -error, robot tracking velocity, roll angle, pitch angle, and tracking performance. From the tracking performance of x -position and y -position, the mobile medical service robot can efficiently follow the teaching trajectory and avoid obstacles. The x -position error and y -position error also can be maintained at a high accuracy, which is within ± 0.06 meters in x -position and ± 0.03 meters in y -position. At the same time, the neural-based predictive controller can constrain the mobile robot body, and the pitch angle and roll angle are within ± 0.02 degrees and ± 0.03 degrees, respectively.

4. Conclusion

In this paper, a novel human-like control framework is implemented to control a mobile service robot using a Kinect sensor and DMP with GMM. It aims to bridge the human activity recognition techniques and assist the mobile medical service robot and allows the robot to cooperate with the medical staff. The Kinect sensor is used to detect human activities to generate a set of movement points, and then the teaching method including dynamic movement primitives

with the Gaussian mixture model can generate the desired trajectory. To achieve stable tracking, a model predictive tracking control scheme based on neural networks is implemented to follow the teaching trajectory. Finally, some demonstrations are carried out in a medical room to validate the effectiveness and superiority of the developed framework.

Human-machine collaborative control based on the Internet of Things (IoT) is the future research direction. In our lastest work [43], we have successfully used IoT technology to exploit the best action in human-robot interaction for the surgical KUKA robot. Instead of utilizing compliant swivel motion, HTC VIVE PRO controllers, used as the Internet of Things technology, are adopted to detect the collision, and a virtual force is applied on the elbow of the robot, enabling a smooth rotation for human-robot interaction. Future work, combined with the IoT technology and multisensors, the concept of the intelligent medical room, will be considered to strengthen the human-robot cooperation.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2019YFC1511401 and the National Natural Science Foundation of China under Grant 61103157.

References


- [1] W.-J. Guan, Z.-Y. Ni, Y. Hu et al., "Clinical characteristics of coronavirus disease 2019 in China," *New England Journal of Medicine*, vol. 382, no. 18, 2020.
- [2] F. Pan, T. Ye, P. Sun et al., "Time course of lung changes on chest ct during recovery from 2019 novel coronavirus (covid-19) pneumonia," *Radiology*, vol. 295, no. 3, 2020.
- [3] H. Chen, J. Guo, C. Wang et al., "Clinical characteristics and intrauterine vertical transmission potential of covid-19 infection in nine pregnant women: a retrospective review of medical records," *The Lancet*, vol. 395, no. 10226, pp. 809–815, 2020.
- [4] Y. Bai, L. Yao, T. Wei et al., "Presumed asymptomatic carrier transmission of covid-19," *Journal of the American Medical Association*, vol. 323, no. 14, pp. 1406–1407, 2020.
- [5] H. Su, C. Yang, G. Ferrigno, and E. De Momi, "Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
- [6] Z. Li, B. Huang, Z. Ye, M. Deng, and C. Yang, "Physical human-robot interaction of a robotic exoskeleton by admittance control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9614–9624, 2018.
- [7] T. Klamt, M. Kamedula, H. Karaoguz et al., "Flexible disaster response of tomorrow: final presentation and evaluation of

- the centauro system,” *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 59–72, 2019.
- [8] J. Li, J. Wang, H. Peng, L. Zhang, Y. Hu, and H. Su, “Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction,” *Neurocomputing*, vol. 410, pp. 342–353, 2020.
 - [9] M. Deng, Z. Li, Y. Kang, C. P. Chen, and X. Chu, “A learning-based hierarchical control scheme for an exoskeleton robot in human-robot cooperative manipulation,” *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 112–125, 2018.
 - [10] X. Wu, Z. Li, Z. Kan, and H. Gao, “Reference trajectory reshaping optimization and control of robotic exoskeletons for human-robot co-manipulation,” *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3740–3751, 2019.
 - [11] T. Klamt, M. Schwarz, C. Lenz et al., “Remote mobile manipulation with the centauro robot: full-body telepresence and autonomous operator assistance,” *Journal of Field Robotics*, vol. 37, no. 5, pp. 889–919, 2019.
 - [12] Z. Li, F. Chen, A. Bicchi, Y. Sun, and T. Fukuda, “Guest editorial neuro-robotics systems: sensing, cognition, learning, and control,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 145–147, 2019.
 - [13] H. Su, W. Qi, C. Yang, A. Aliverti, G. Ferrigno, and E. De Momi, “Deep neural network approach in human-like redundancy optimization for anthropomorphic manipulators,” *IEEE Access*, vol. 7, pp. 124207–124216, 2019.
 - [14] Z. G. Li, Z. Ren, K. Zhao, C. Deng, and Y. Feng, “Human-cooperative control design of a walking exoskeleton for body weight support,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 2985–2996, 2019.
 - [15] Y. Hu, X. Wu, P. Geng, and Z. Li, “Evolution strategies learning with variable impedance control for grasping under uncertainty,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7788–7799, 2018.
 - [16] X. Wu and Z. Li, “Cooperative manipulation of wearable dual-arm exoskeletons using force communication between partners,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6629–6638, 2019.
 - [17] H. Su, C. Yang, H. Mdeihly, A. Rizzo, G. Ferrigno, and E. De Momi, “Neural network enhanced robot tool identification and calibration for bilateral teleoperation,” *IEEE Access*, vol. 7, pp. 122041–122051, 2019.
 - [18] Z. Cao, Y. Niu, and H. R. Karimi, “Sliding mode control of automatic electronic valve system under weighted try-once-discard protocol,” *Information Sciences*, vol. 515, pp. 324–340, 2020.
 - [19] X. Zhao, X. Wang, L. Ma, and G. Zong, “Fuzzy-approximation-based asymptotic tracking control for a class of uncertain switched nonlinear systems,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 4, pp. 632–644, 2019.
 - [20] J. Li, J. Wang, S. Wang et al., “Parallel structure of six wheel-legged robot trajectory tracking control with heavy payload under uncertain physical interaction,” *Assembly Automation*, vol. 40, no. 5, pp. 675–687, 2020.
 - [21] H. Su, S. E. Ovrur, X. Zhou, W. Qi, G. Ferrigno, and E. De Momi, “Depth vision guided hand gesture recognition using electromyographic signals,” *Advanced Robotics*, vol. 34, no. 15, pp. 985–997, 2020.
 - [22] H. Su, Y. Schmirander, S. E. Valderrama et al., “Asymmetric bimanual control of dual-arm serial manipulator for robot-assisted minimally invasive surgeries,” *Sensors and Materials*, vol. 32, no. 4, p. 1223, 2020.
 - [23] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, and A. Aliverti, “A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone,” *Sensors*, vol. 19, no. 17, p. 3731, 2019.
 - [24] W. He, T. Meng, X. He, and C. Sun, “Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances,” *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1524–1535, 2018.
 - [25] Z. Li, B. Huang, A. Ajoudani, C. Yang, C.-Y. Su, and A. Bicchi, “Asymmetric bimanual control of dual-arm exoskeletons for human-cooperative manipulations,” *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 264–271, 2017.
 - [26] Y. Hu, Z. Li, G. Li, P. Yuan, C. Yang, and R. Song, “Development of sensory-motor fusion-based manipulation and grasping control for a robotic hand-eye system,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1169–1180, 2016.
 - [27] Z. Liu, H. R. Karimi, and J. Yu, “Passivity-based robust sliding mode synthesis for uncertain delayed stochastic systems via state observer,” *Automatica*, vol. 111, Article ID 108596, 2020.
 - [28] Q. Wei, Z. Li, K. Zhao, Y. Kang, and C.-Y. Su, “Synergy-based control of assistive lower-limb exoskeletons by skill transfer,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 705–715, 2019.
 - [29] H. Peng, J. Wang, W. Shen, and D. Shi, “Cooperative attitude control for a wheel-legged robot,” *Peer-to-Peer Networking and Applications*, vol. 12, no. 6, pp. 1741–1752, 2019.
 - [30] Z. Li, J. Li, S. Zhao, Y. Yuan, Y. Kang, and C. P. Chen, “Adaptive neural control of a kinematically redundant exoskeleton robot using brain-machine interfaces,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3558–3571, 2018.
 - [31] W. He and Y. Dong, “Adaptive fuzzy neural network control for a constrained robot using impedance learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 1174–1186, 2017.
 - [32] X. Zhang, J. Li, S. E. Ovrur et al., “Novel design and adaptive fuzzy control of a lower-limb elderly rehabilitation,” *Electronics*, vol. 9, no. 2, p. 343, 2020.
 - [33] L. Zhang, Z. Li, and C. Yang, “Adaptive neural network based variable stiffness control of uncertain robotic systems using disturbance observer,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2236–2245, 2016.
 - [34] Z. Li, C. Xu, Q. Wei, C. Shi, and C.-Y. Su, “Human-inspired control of dual-arm exoskeleton robots with force and impedance adaptation,” *IEEE Transactions on Systems, Man, and Cybernetics Systems*, pp. 1–10, 2018.
 - [35] H. Su, N. Enayati, L. Vantadori, A. Spinoglio, G. Ferrigno, and E. De Momi, “Online human-like redundancy optimization for tele-operated anthropomorphic manipulators,” *International Journal of Advanced Robotic Systems*, vol. 15, 2018.
 - [36] Z. Wu, H. R. Karimi, and C. Dang, “A deterministic annealing neural network algorithm for the minimum concave cost transportation problem,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 7, pp. 699–708, 2019.
 - [37] J. Sandoval, H. Su, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, “Collaborative framework for robot-assisted minimally invasive surgery using a 7-DoF anthropomorphic robot,” *Robotics and Autonomous Systems*, vol. 106, pp. 95–106, 2018.
 - [38] J. Gong, Y. Jiang, and W. Xu, *Model Predictive Control for Self-Driving Vehicles*, Beijing Institute of Technology Press, Beijing, China, 2014.
 - [39] H. Ren, H. R. Karimi, R. Lu, and Y. Wu, “Synchronization of network systems via aperiodic sampled-data control with constant delay and application to unmanned ground

- vehicles,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 4980–4990, 2019.
- [40] B. Xiao, X. Yang, H. R. Karimi, and J. Qiu, “Asymptotic tracking control for a more representative class of uncertain nonlinear systems with mismatched uncertainties,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9417–9427, 2019.
- [41] Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, “Vision-based model predictive control for steering of a nonholonomic mobile robot,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 553–564, 2015.
- [42] H. Peng, J. Wang, W. Shen, D. Shi, and Y. Huang, “Compound control for energy management of the hybrid ultra-capacitor-battery electric drive systems,” *Energy*, vol. 175, pp. 309–319, 2019.
- [43] H. Su, S. Ertug Ovrur, Z. Li et al., “Internet of things (IoT)-based collaborative control of a redundant manipulator for teleoperated minimally invasive surgeries,” in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, September 2020.

Research Article

Simulation Framework for Cyber-Physical Production System: Applying Concept of LVC Interoperation

Byeong Soo Kim,¹ Seunghoon Nam,¹ Yooeui Jin,¹ and Kyung-Min Seo ²

¹Global Technology Center, Samsung Electronics, Suwon-si, Republic of Korea

²Department of Future Technology, Korea University of Technology and Education (KOREATECH), Cheonan-si, Republic of Korea

Correspondence should be addressed to Kyung-Min Seo; kmseo@koreatech.ac.kr

Received 8 April 2020; Revised 28 June 2020; Accepted 13 July 2020; Published 8 October 2020

Guest Editor: Hang Su

Copyright © 2020 Byeong Soo Kim 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 Industry 4.0, many manufacturers have built smart factories by ICTs (Information and Communications Technology), and simulation is one of the core technologies for smart manufacturing. Various kinds of simulations, depending on system levels, such as assembly line, logistics, worker, and process, are utilized for smart manufacturing. Manufacturers own heterogeneous simulations; however, they have difficulty integrating and interoperating them. This paper proposes a novel simulation framework for smart manufacturing based on the concept of live, virtual, and constructive (LVC) simulation. The LVC interoperation provides a synthetic simulation environment with the above three types of simulations. With the LVC interoperation, we propose a systematic and efficient architecture for smart manufacturing. To be specific, the interface technologies between the heterogeneous simulations and their interoperable methods are developed. Finally, we provide a practical LVC simulation applied in the manufacturing company and show what synergy can be created using the LVC simulation.

1. Introduction

Smart manufacturing is a broad category of manufacturing which employs computer-integrated manufacturing, high levels of adaptability and rapid design changes, digital information technology, and more flexible technical workforce training [1, 2]. In the era of the fourth industrial revolution, as ICTs such as Internet of Things (IoT), big data, artificial intelligence, and Virtual Reality/Augmented Reality (VR/AR) develop, manufacturing systems are becoming more intelligent and unmanned [3, 4]. For example, through the IoT sensors, real-time data such as various facilities, workers, transportation equipment, and factory environment can be obtained [5, 6]. It enables real-time fault detection and diagnosis [7–9]. Also, the acquired big data can be used to analyze and predict shop floor through artificial intelligence and machine-learning techniques [10]. Also, users can apply VR/AR to a manufacturing system to plan and test a complex manufacturing process or assembly process in advance [11]. Logistics and equipment management

are possible through this virtual manufacturing environment. We can utilize these ICTs to increase production efficiency [12].

Many manufacturing companies introduce a cyber-physical production system (CPPS) to innovate production using ICTs [13–16]. With smart manufacturing, it is possible to make decisions and perform tasks before actual plant construction and to solve problems that can occur in mass production in a virtual environment in advance [17]. Also, even after plant construction, it can be used for maintenance, such as the optimal operation of production lines and fault diagnosis, to increase the production competitiveness [18]. As interest in smart manufacturing increases, the importance of simulation, a key technology for this, is also increasing [19]. In manufacturing, there are various kinds of simulations depending on system levels or purposes, such as assembly line simulation, logistics simulation, process simulation, and worker simulation. Manufacturers are using these approaches to build smart factories, but they are having difficulty integrating and interoperating these heterogeneous simulations [20].

Therefore, this paper proposes a new simulation framework for CPPS which applies the concept of live, virtual, and constructive (LVC) interoperation. The LVC interoperation is a widely applied concept in the defense modeling and simulation (M&S) domain [21]. It establishes a synthetic theater of war by integrating or interoperating three heterogeneous simulations: live, virtual, and constructive [22]. In production, individual simulation technologies have developed along with smart manufacturing. An operation plan for integrating them is needed, and the LVC in the defense domain, which has relatively developed interoperation architecture, can be applied and implemented [23].

In the manufacturing domain, there were many simulations at each layer, but there were few requirements for training/analysis through simulation interoperation like defense domain. But, the recent rise of smart factories has increased the need for convergence of existing simulations. However, research on individual case studies rather than the overall interoperation framework is still the main focus. Therefore, the concept of LVC interoperation can be applied to the smart manufacturing to build a systematic and efficient smart manufacturing architecture.

There have been some studies where simulations were utilized to build the smart factory and others that applied to the smart manufacturing by interoperating process simulator and line simulator [24, 25]. They yielded more accurate simulation results through the interoperation. However, interoperation was used for distributed execution of the same level of simulators rather than interoperation with different levels of simulators [26]. In addition, they show only individual interoperation cases and do not suggest an overall architecture throughout the smart factory.

There were also studies on the simulation direction of the CPPS-based factory [27]. They built a digital twin by implementing the corresponding physical components in the production process as digital components. However, since the system levels and characteristics of many components in the production process are different, it is difficult to implement and integrate them into one environment. Therefore, they are just suggesting an overall concept or implementing some parts of it. Thus, it is important to utilize the existing simulators at each level. To this end, the interoperation architecture of manufacturing simulation and the interface construction are required [28]. This paper not only creates synergy by interoperating existing simulators for smart manufacturing but also increases reusability and modularity. In addition, through this LVC interoperation for smart manufacturing, it presents the operational plan and future for the smart manufacturing.

The remainder of the paper is organized as follows. Section 2 presents the basic knowledge about LVC interoperation. Section 3 provides a novel LVC interoperation for CPPS-based smart manufacturing and discusses an application plan of the proposed work in the shop floor of the smart manufacturing. A case study of applying the proposed approach is described in Section 4. Finally, Section 5 concludes the discussion.

2. Concept of LVC Interoperation

The LVC interoperation technology integrates and operates three heterogeneous simulators as shown in Figure 1. It is a concept commonly used in the defense system and to build a field and efficient training system by interoperation of three resources [21].

The live simulation refers to training actual forces in the actual environment. The virtual simulation refers to training actual forces in the same virtual environment as the actual equipment. The constructive simulation means training by operating virtual forces under the virtual environment.

Live simulation can increase the reality because actual forces perform in the actual environment, but it takes a lot of time/resources and is limited to operating multiple times. Virtual simulations, such as aircraft simulators or tank simulators, can simulate equipment that is difficult to operate with a small cost and little time using a virtual environment, but there is a limit to understanding the overall training situation for various situations. Constructive simulation such as war games can simulate various scenarios, but it is relatively out of touch with reality and it is difficult to describe each object in detail [29].

Thus, these three simulations have different advantages and disadvantages, and they can be interoperated to build a synthetic theater of war. Through this, the distributed operations complement the advantages and disadvantages of each other and enable low-cost, high-efficiency training. Successful interoperation of LVC requires integrability of infrastructures, interoperability of systems, and composability of models [30].

At this time, LVC can communicate using interoperation middleware such as High Level Architecture/Run Time Infrastructure (HLA/RTI), as shown in Figure 1. The HLA/RTI is a middleware that implements an international standard for distributed simulation defined in IEEE 1516. It allows real-time data exchange and time management between heterogeneous simulators [31–33]. These interoperation technologies play important roles in constructing the LVC system. In particular, the establishment of interoperation standards increases the reusability and operability of the simulators.

Meanwhile, in the production domain, various simulators have been developed along with the smart manufacturing, and some ideas are needed to integrate them. In this paper, we use the relatively more developed concept of LVC interoperation in the defense domain to implement a novel interoperation architecture in the production domain. In the next section, we apply the concept to build a systematic and efficient interoperation architecture for smart manufacturing.

3. Applying LVC Interoperation into Smart Manufacturing

In this section, we propose an LVC interoperation for CPPS-based smart manufacturing. Firstly, we explain each component of LVC. Then, we provide the overall interoperation architecture and application plan in the shop floor of smart manufacturing.

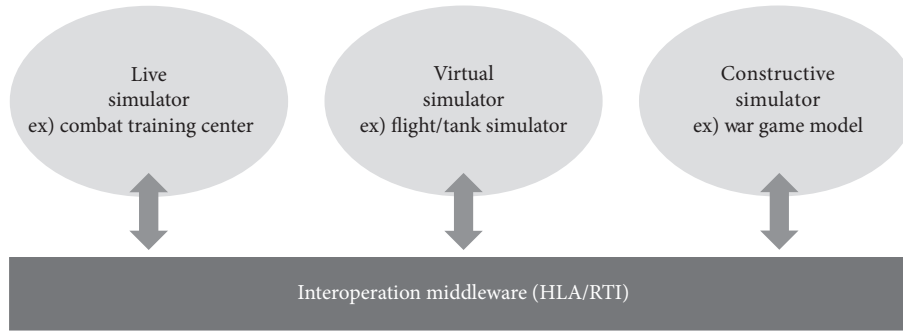


FIGURE 1: Concept of LVC interoperation.

3.1. Components of LVC in Smart Manufacturing. This section describes how to define and operate each component of LVC simulation in smart manufacturing. First, the concept of live simulation is shown in Figure 2. Live simulation in the smart manufacturing can be performed by actual workers on the assembly line or logistics on the actual shop floor. For example, workers can install IoT sensors and cameras when performing the assembly or logistics work and acquire real-time data through the attached sensors and cameras. The data obtained through the live simulation can be used to improve work proficiency and inefficient operation/behavior.

Figure 3 shows the actual live simulation that we are currently operating. When a real worker with IoT sensors performs assembly work, we can monitor operations through the sensors and the camera. The collected data allows us to measure the cycle time of the work, analyze the motion that is being wasted, and give feedback to the worker.

Next, the concept of virtual simulation is shown in Figure 4. Virtual simulation is a kind of hardware-in-the-loop (HIL) simulation in which users operate virtual simulators through hardware such as a Human Interface Device (HID). For example, a user can perform a simulation of an operation through a controller connected to a process simulator such as a machine or robot. At this time, the real-time results of the simulator return to the user in real time, and the user can perform the closed-loop simulation by constantly reflecting the feedback.

In addition, the user can experience and review the 3D Visual Factory using the VR/AR devices. As technologies such as VR/AR are developed and cooperated with existing virtual simulators, the spectrum of virtual simulation is expanding. Figure 5 is an example of a 3D virtual factory currently operating in the manufacturing facilities. The 3D virtual simulation model is implemented identically with the actual factory layout. It receives real-time data from the actual manufacturing line through the sensors. If abnormal data is detected, an alarm appears on the model and the user can detect it using equipment such as a VR Glass. This facilitates monitoring the manufacturing line and remote support. Another possible application using a VR device is an assembly test in the virtual work place. Usually implemented to validate workability of new work such as collision, reachability, or productivity, we need to build a new workshop for pilot testing. However, using VR/AR

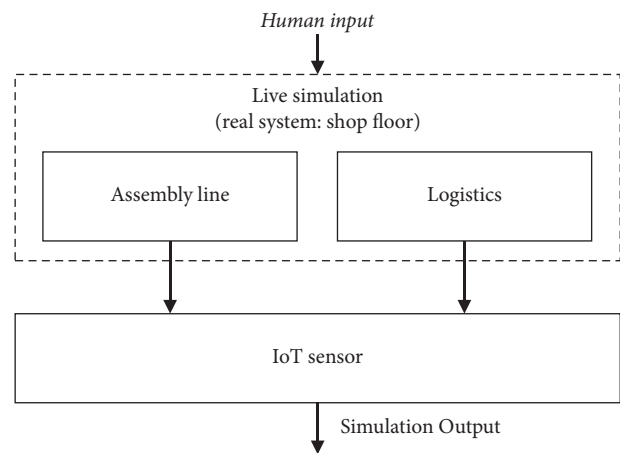


FIGURE 2: Concept of live simulation in smart manufacturing.

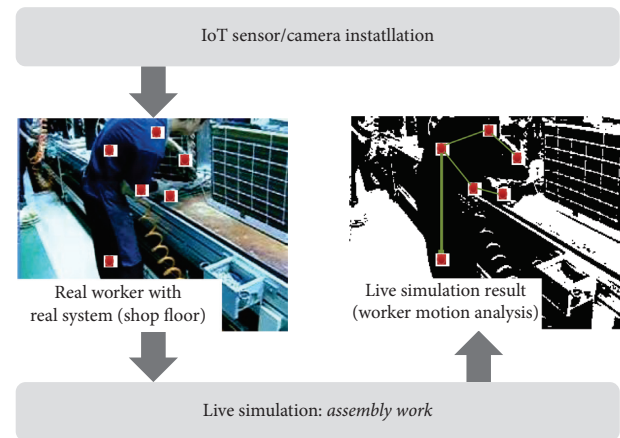


FIGURE 3: Application of live simulation in smart manufacturing.

technology, the worker can perform tasks in a virtual world without real workshop or tools.

Finally, the concept of constructive simulation is shown in Figure 6. Constructive simulator consists of constructive simulation model and constructive simulation engine. The simulation model used in this paper consists of three levels of models: the parameter-based model, the icon-based model, and the source code model so that all users in the production system can use the simulator [34]. The users can build a model in three ways

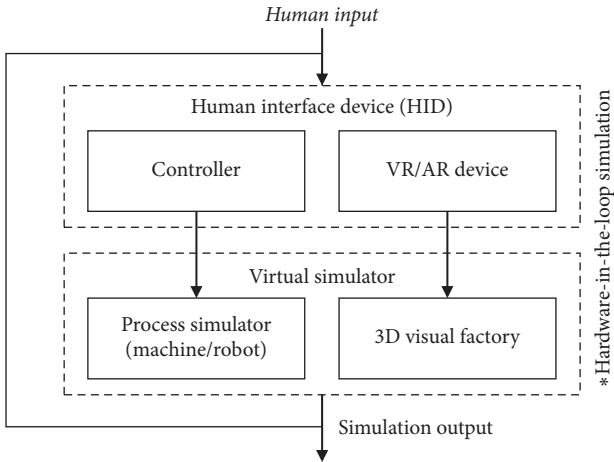


FIGURE 4: Concept of virtual simulation in smart manufacturing.

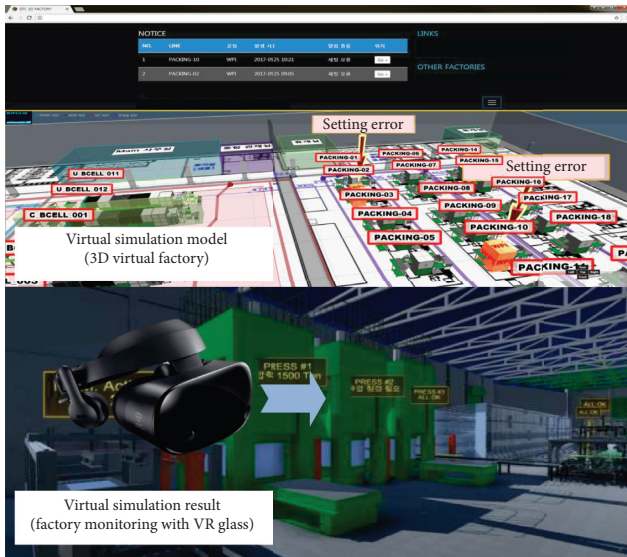


FIGURE 5: Application of virtual simulation in smart manufacturing.

depending on their simulation training level or simulation objective. The model is automatically created through the Production Line Generator and the Model Synthesizer inside the simulation engine as shown in Figure 6. Created as a discrete event simulation model, it can be simulated using a discrete event simulation engine. Although the fidelity is relatively low, the constructive simulation has the advantage of simulating production facilities quickly and repeatedly with virtual agents in a virtual environment.

An example of the constructive simulator that we are currently operating is shown in Figure 7. The upper part of Figure 7 shows the user modeling results of the assembly line and logistics through the icon-based modeling. The lower part of Figure 7 shows the simulation results using the automatically generated model from the user modeling result. Table 1 shows the results of comparing the features of three simulations in smart manufacturing.

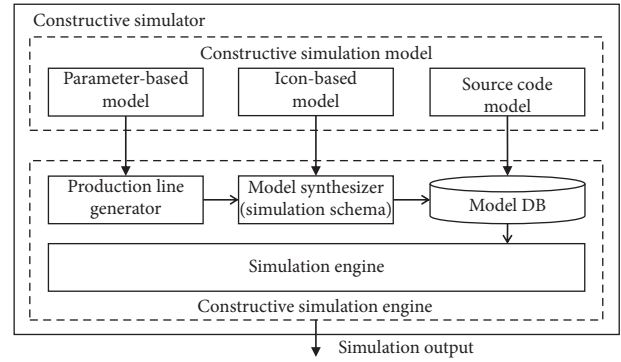


FIGURE 6: Concept of constructive simulation in smart manufacturing.

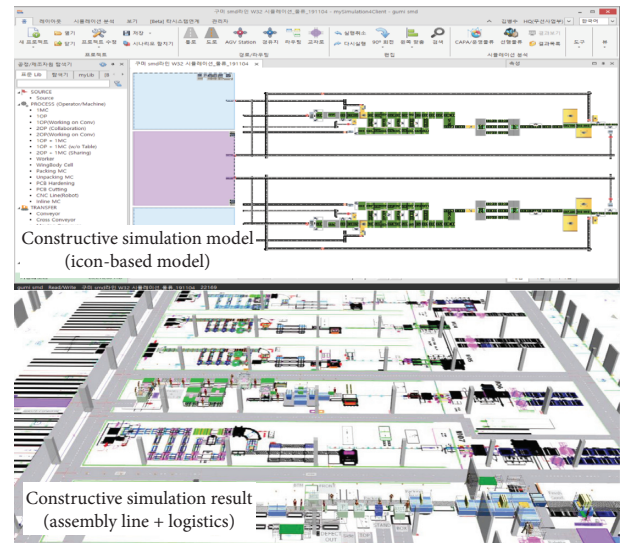


FIGURE 7: Application of constructive simulation in smart manufacturing.

3.2. Overall Architecture and Application Plan. Since modeling is objective-oriented, it is possible to model in various ways according to the objective of analysis in the manufacturing system. Although each simulation described in the previous section can be operated individually according to its purpose, it is important to create synergy by integrating them to build a more efficient smart manufacturing system. Figure 8 shows the overall architecture for LVC interoperation in smart manufacturing. Three simulations communicate using an interoperation interface, and the events they exchange are defined as shown in the figure. The interface consists of a proxy for message communication, a neutral data format, and a management tool for managing data/time. The following subsections describe the application plan for each LC, VC, and LV inside the overall interoperation architecture in detail.

First, LC interoperation will be described. The live simulation senses the operation of the worker through the IoT sensor and transmits the results to the constructive

TABLE 1: Characteristics of live, virtual, and constructive.

Characteristic	Live	Virtual	Constructive
Unit	Real	Real	Virtual
Environment	Real	Virtual	Virtual
Type	Real work	Process simulator, 3D virtual factory	Manufacturing facilities
Tool	IoT sensors, camera	HID, VR/AR device	Personal computer
Objective	Training, real data acquisition	Training	Analysis
Pros and cons	Realistic simulation, but requires a lot of cost and time	Actual environment is unnecessary	Rapid analysis time and less cost, but high abstraction level

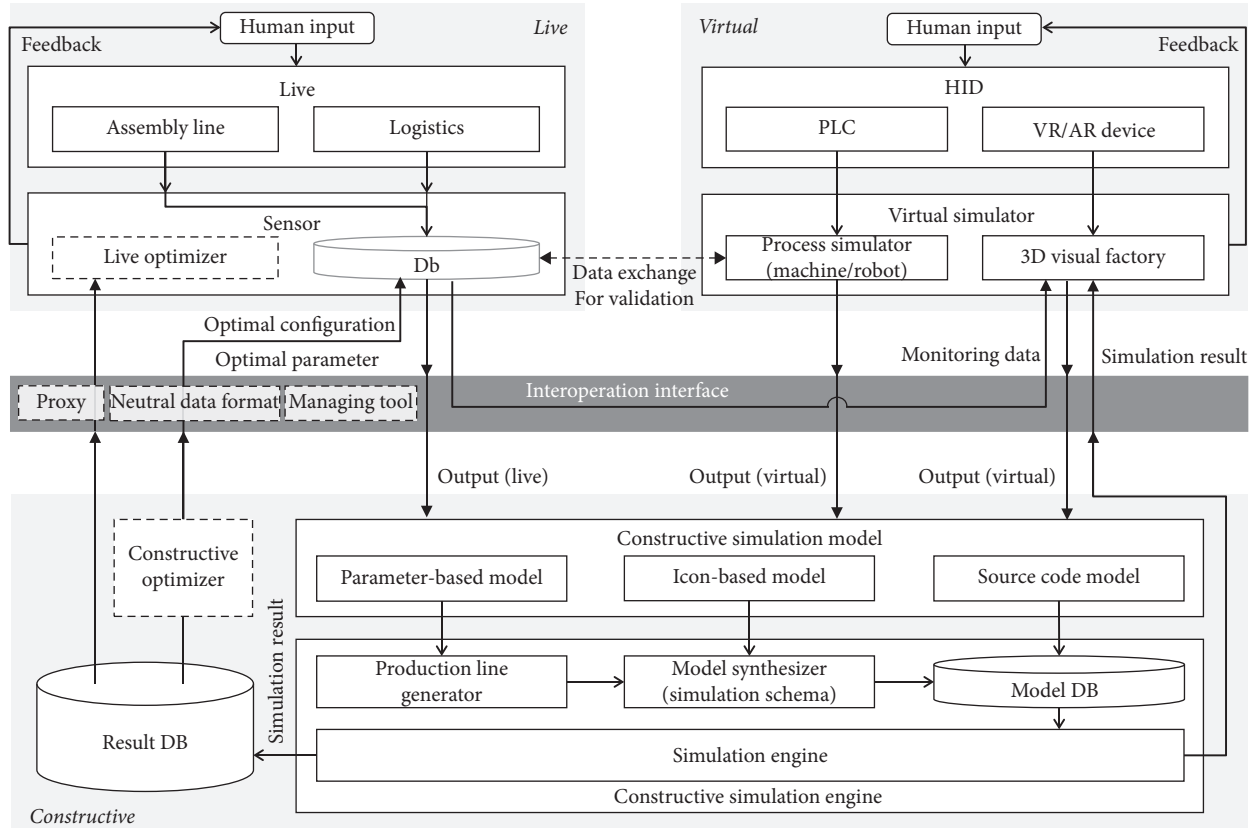


FIGURE 8: Overall architecture of LVC interoperation in smart manufacturing.

simulator in real time. The constructive simulator takes them and performs layout simulation to predict the productivity. The optimal line configurations or parameters derived from simulation-based optimization (constructive optimizer) return to the worker or shop floor to help improve the productivity. For example, by deriving the optimal cycle time for the desired throughput or line of balance (LOB), it can provide the improvement points of the operator's waste operation and the facility's performance. It can also provide a way to improve the shop floor such as with assembly line layout and logistics traffic. In other words, we can derive efficient manufacturing conditions by simulation-based optimization using interoperation of sensing data and simulation models [35, 36]. Also, we can optimize the results of the constructive simulation through the motion analysis (live optimizer) in the live simulation. Details of this procedure will be covered in the case study.

Next, one of VC interoperations is an operation with layout simulator and process simulator using HID. By replacing part of the virtual layout model with the process simulator, it is possible to do virtual commissioning that the user controls through the controller in real time. Virtual commissioning can dramatically reduce system installation costs and operating time by simulating and verifying automation equipment in a virtual environment to ensure that the equipment works as expected [37, 38]. The layout simulator can be interoperated with the virtual simulator as well as the actual machine. This enables the feasibility and interoperability test between equipment and assembly line and allows more detailed simulation of the manufacturing system through the interoperation between different levels of simulators. In addition, there is an interoperation between VR/AR device and layout simulator as another method of VC interoperation. Using the layout modeling environment,

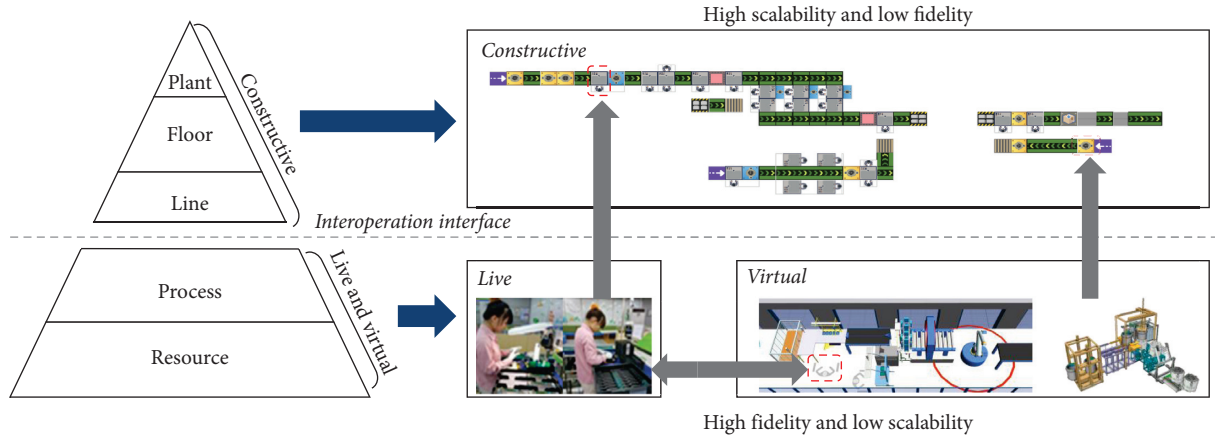


FIGURE 9: LVC interoperation example.

we can model the manufacturing plant as a 3D virtual factory and experience it using VR/AR devices. In addition, intuitive fault detection and diagnosis is possible by transmitting the real-time prediction results of the layout simulation to the virtual environment. It enables monitoring the manufacturing line with a remote support.

Next, one of LV interoperations is an analysis of virtual facilities in conjunction with actual facilities. In other words, a virtual simulator can be validated through comparison of data between virtual and actual facilities. The parameters applied to the virtual simulator can be tuned through the results of the actual equipment, and synergy can be created by performing high-scalable simulation using these validated simulators. In addition, it is possible to monitor the operating status through the 3D virtual factory by sharing the actual operation status and results of the live simulation. On the other hand, it is possible to test and validate new workshop or improved tasks in a virtual environment. Newly designed workshop can be constructed as a virtual simulation model, and a worker can perform tasks using VR without the actual workshop. At the same time, IoT sensors attached to the worker collect motion data and transmit the data to analysis.

Finally, we can operate smart manufacturing by interoperation of all three LVC simulations. Figure 9 shows a simple example of LVC interoperation. The LVC interoperation concept was applied to the hybrid simulation for advanced analysis of the manufacturing system. Among the five components that make up the factory, high-level components such as plant, floor, and line can be expressed using constructive simulation. They have high scalability but low fidelity. On the other hand, low-level components such as process and resource can be expressed by live simulation and virtual simulation depending on the type. They have low scalability but high fidelity on the contrary. By interoperating the simulations at different levels, it is possible to predict the manufacturing system more precisely. In the case study, we will show a simple example of LVC interoperation using these operation plans.

Likewise, with the LVC interoperation framework for smart manufacturing, various application plans can be

operated depending on the needs of the shop floor as described above. Through the framework, three heterogeneous simulators can achieve organic cooperation and increase interoperability/connectivity. As a result, the gap between the real and cyber world can be bridged, and CPPS-based smart manufacturing system can be finally established. In the case study, we will show a simple example of LVC interoperation using these operation plans.

4. Case Study

Generally, manufacturers use simulations to build new production lines or improve existing production lines. The simulation predicts the production of the lines in advance and finds ways to optimize the line. At this time, the process time per resource, which is the input data of the simulation model, actually utilizes the data measured in the line. This section describes a case where LVC interoperation was performed by applying this.

Figure 10 shows the layout of the linear production line used for the case study. It is one of simple test lines in our company. Five workers and one automation machine perform six separate processes from the left, and a conveyor is placed between each process to move the finished product from the previous process to the next. The user creates a model of constructive simulation based on the layout and performs the simulation. At this time, the process time per resource is obtained from live simulation and used. Table 2 is the time of the process and the process for each resource used in the case study. Here, the process time is obtained differently depending on the resource type. The worker's value is measured by attaching an IoT sensor. In the case of the machine, it is obtained by analyzing the log data about the operation of the machine through OPC UA.

Figure 11 shows the results of constructive simulation modeling by placing a library that represents the process of resources in the production line and Figure 12 shows the actual process (working, blocking, and waiting) load results for each process when the simulation is performed for 460 minutes. The legend is expressed on the right side of the graph. Through this, it can be confirmed that the assembly



FIGURE 10: Layout for production line used in case study.

TABLE 2: Process time and resource by process in the production line of the case study.

No.	Process	Process time			Resource
		Min	Avg.	Max	
1	Input	8.81	15.66	20.81	Worker
2	Preassembly	9.60	15.57	21.71	Worker
3	Assembly cell 1	16.61	18.61	21.6	Worker
4	Assembly cell 2	14.53	18.60	22.51	Worker
5	Inspection	13.95	18.00	19.50	Machine
6	Packing	13.39	16.40	20.42	Worker

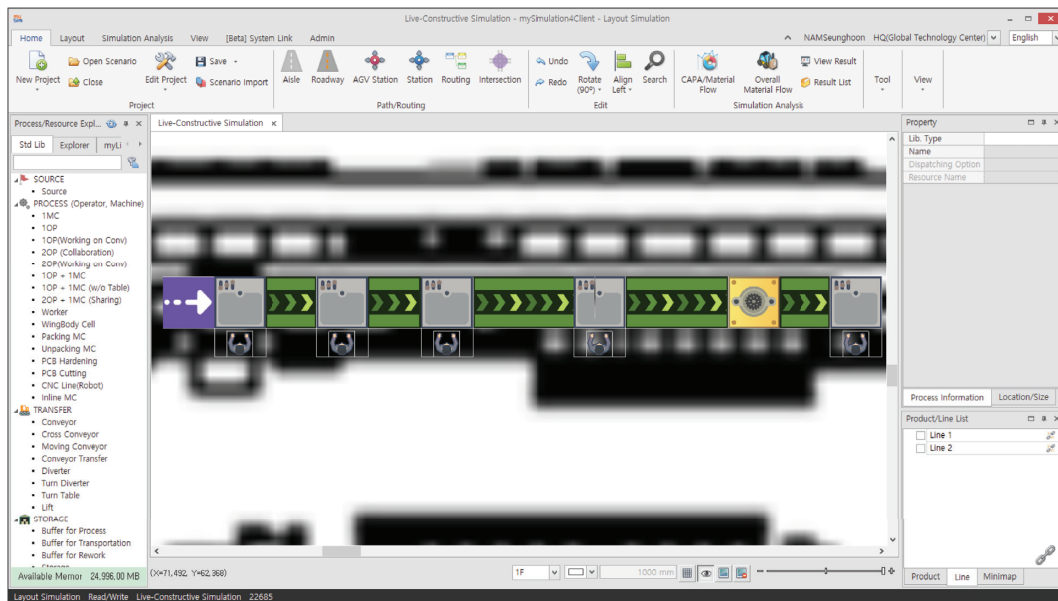


FIGURE 11: Constructive simulation modeling results for the linear production line.

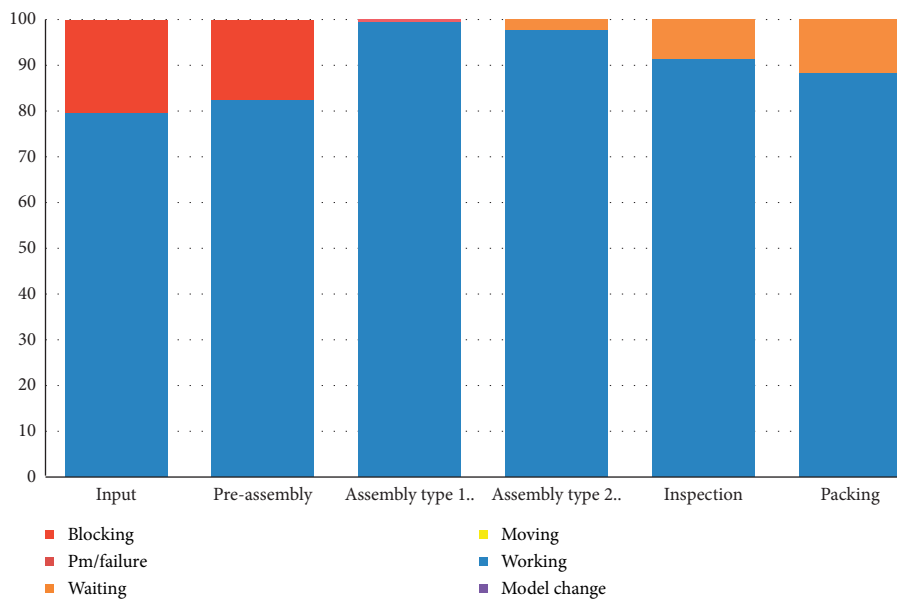


FIGURE 12: Simulation results of process load for initial measured process time.

TABLE 3: Throughput, cycle time, and working time for each process in the line.

No.	Process	Throughput	Cycle time	Working time
1	Input	1,460	18.9	15.02
2	Preassembly	1,459	18.92	15.61
3	Assembly cell 1	1,457	18.94	18.9
4	Assembly cell 2	1,455	18.97	18.49
5	Inspection	1,454	18.98	17.36
6	Packing	1,453	19	16.78

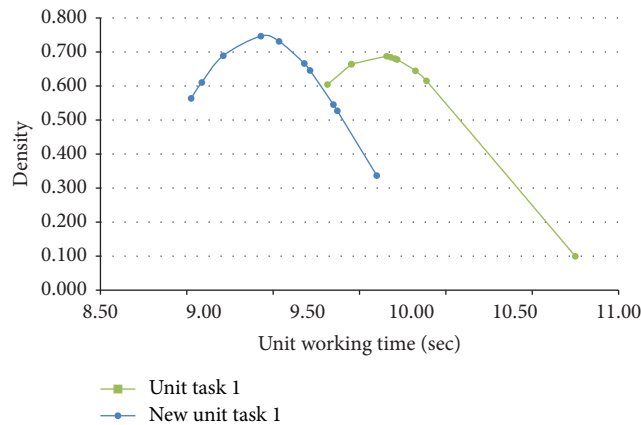


FIGURE 13: Working time distribution curve for unit task 1 of the assembly cell.

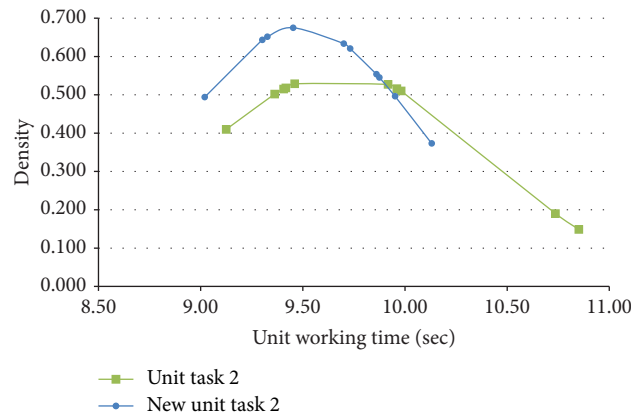


FIGURE 14: Working time distribution curve for unit task 2 of the assembly cell.

type 1 process that generates a blocking in the previous process and generates a waiting in the next process is a bottleneck process. The throughput of each process, cycle time, and working time are shown in Table 3. The total production of the line is 1,453, which is the throughput of the last process.

To optimize the line based on the results, this study restructured the process time for assembly cell 1, which was analyzed as a bottleneck through the results of constructive simulation. Assembly cell 1 consists of two unit tasks. By performing motion analysis within each unit task, unnecessary operations are removed and the locations of tool and

parts are relocated. To forecast the new process time of improved task in the suggested new workshop layout, the virtual workshop was constructed in detail so that a worker performed the new task in a virtual environment using VR device.

As a result, process time of assembly cell 1 is shortened by improving work behavior and new workshop layout. The sample data (Figures 13 and 14) show the normal distribution curve for the existing unit working time and the changed unit working time. Constructive simulation was performed again to reflect this result. The lead time of all lines was reduced and the production was increased

TABLE 4: Throughput, cycle time, and working time per process in the line according to work time reorganization.

No.	Process	Throughput	Cycle time	Working time
1	Input	1,493	18.49	15.07
2	Preassembly	1,491	18.51	15.54
3	Assembly cell 1	1,489	18.54	18.17
4	Assembly cell 2	1,487	18.56	18.5
5	Inspection	1,485	18.59	17.42
6	Packing	1,484	18.6	16.72

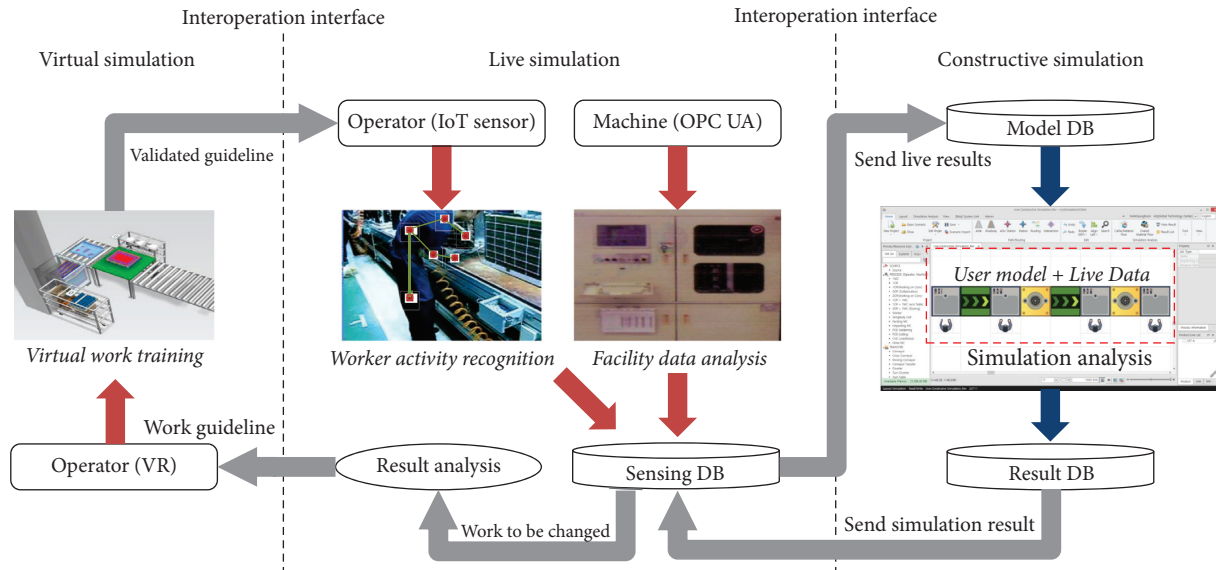


FIGURE 15: The LVC interoperation procedure performed in the case study.

(Table 4). The procedure performed in this case study is represented by a diagram as shown in Figure 15. This case study has shown that organic interoperation of three heterogeneous simulations can contribute to improve productivity and create synergy.

5. Conclusion

In the fourth industrial revolution, many manufacturing enterprises are building smart manufacturing platforms using various ICTs. As interest in smart manufacturing increases, various simulators are being developed to simulate production sites according to the objective of analysis. Manufacturers continue to utilize these various simulators to build smart manufacturing systems. However, they have difficulty integrating and interoperating existing heterogeneous simulators for smart manufacturing. In other words, we need a whole framework for interoperating them. Therefore, this paper proposed an interoperation framework of manufacturing simulations by applying the existing LVC interoperation concept operated in the military domain. This framework provides an interoperation interface and interoperable methods between each simulation. Finally, we showed how we can utilize it through the actual case study of interoperation between live and constructive simulation. It presented a method to link the constructive simulation for one production line and the live simulation of

resources in the line. It was verified by performing one example of increasing production throughput of the production line through actual data. In addition, it has been able to improve the reusability and interoperability of existing simulations.

By using LVC, the feasible integration is started from the manufacturing stage, and the comprehensive smart factory including other legacy systems, such as Manufacturing Execution System and Supply Chain Management, can be built gradually. Finally, we will build and expand a Manufacturing Lifecycle Management system that covers everything from development to manufacturing. In addition, it is required to propose the direction of the smart factory through the LVC interoperation.

Data Availability

Data in this work are not freely available due to patient privacy and commercial confidentiality.

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 Samsung Electronics Co., Ltd.

References

- [1] F. Tao and M. Zhang, "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing," *IEEE Access*, vol. 5, pp. 20418–20427, 2017.
- [2] X. Yao, J. Zhou, Y. Lin, Y. Li, H. Yu, and Y. Liu, "Smart manufacturing based on cyber-physical systems and beyond," *Journal of Intelligent Manufacturing*, vol. 30, no. 8, pp. 2805–2817, 2019.
- [3] J. Wan, M. Yi, D. Li, C. Zhang, S. Wang, and K. Zhou, "Mobile services for customization manufacturing systems: an example of industry 4.0," *IEEE Access*, vol. 4, pp. 8977–8986, 2016.
- [4] U. H. Govindarajan, A. J. C. Trappey, and C. V. Trappey, "Immersive technology for human-centric cyberphysical systems in complex manufacturing processes: a comprehensive overview of the global patent profile using collective intelligence," *Complexity*, vol. 2018, Article ID 4283634, 17 pages, 2018.
- [5] Y. Tan, W. Yang, K. Yoshida, and S. Takakuwa, "Application of IoT-aided simulation to manufacturing systems in cyber-physical system," *Machines*, vol. 7, no. 1, p. 2, 2019.
- [6] G. Hwang, J. Lee, J. Park, and T.-W. Chang, "Developing performance measurement system for Internet of Things and smart factory environment," *International Journal of Production Research*, vol. 55, no. 9, pp. 2590–2602, 2017.
- [7] K.-M. Seo and K.-P. Park, "Interface data modeling to detect and diagnose intersystem faults for designing and integrating system of systems," *Complexity*, vol. 2018, Article ID 7081501, 21 pages, 2018.
- [8] K. Y. H. Lim, P. Zheng, and C. H. Chen, "A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives," *Journal of Intelligent Manufacturing*, vol. 31, pp. 1313–1337, 2020.
- [9] A. J. H. Redelinghuys, A. H. Basson, and K. Kruger, "A six-layer architecture for the digital twin: a manufacturing case study implementation," *Journal of Intelligent Manufacturing*, vol. 31, no. 6, pp. 1383–1402, 2019.
- [10] B. S. Kim and T. G. Kim, "Modeling and simulation using artificial neural network-embedded cellular automata," *IEEE Access*, vol. 8, no. 1, pp. 24056–24061, 2020.
- [11] G. Chen, P. Wang, B. Feng, Y. Li, and D. Liu, "The framework design of smart factory in discrete manufacturing industry based on cyber-physical system," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 1, pp. 79–101, 2019.
- [12] S. Choi, C. Jun, W. B. Zhao, and S. Do Noh, "Digital manufacturing in smart manufacturing systems: contribution, barriers, and future directions," in *proceedings of the IFIP International Conference on Advances in Production Management Systems*, pp. 21–29, Tokyo, Japan, September 2015.
- [13] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.
- [14] H. Tang, D. Li, S. Wang, and Z. Dong, "CASOA: an architecture for agent-based manufacturing system in the context of Industry 4.0," *IEEE Access*, vol. 6, pp. 12746–12754, 2017.
- [15] L. Ribeiro and M. Hochwallner, "On the design complexity of cyberphysical production systems," *Complexity*, vol. 2018, Article ID 4632195, 13 pages, 2018.
- [16] R. Lovas, A. Farkas, A. C. Marosi et al., "Orchestrated platform for cyber-physical systems," *Complexity*, vol. 2018, Article ID 8281079, 16 pages, 2018.
- [17] R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015.
- [18] G. Y. Kim, J. Y. Lee, H. S. Kang, and S. D. Noh, "Digital factory wizard: an integrated system for concurrent digital engineering in product lifecycle management," *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 11, pp. 1028–1045, 2010.
- [19] J. Jeon, S. Kang, and I. Chun, "CPS-based model-driven approach to smart manufacturing systems," *The Fifth International Conference on Intelligent Systems and Applications*, vol. 68, pp. 136–146, 2016.
- [20] R. Jardim-Goncalves, A. Grilo, and K. Popplewell, "Novel strategies for global manufacturing systems interoperability," *Journal of Intelligent Manufacturing*, vol. 27, no. 1, pp. 1–9, 2016.
- [21] DoD, *DoD Modeling and Simulation (M&S) Glossary. DoD 5000.59-M*, CreateSpace Independent Publishing Platform, Sacramento, CA, USA, 2013.
- [22] W. Choi, K. Yu, B. J. Park, S. Kang, and J. Lee, "Study on LVC (Live-Virtual-Constructive) interoperation for the national defense M&S (modeling & simulation)," in *Proceedings of the 2008 International Conference on Information Science and Security (ICISS 2008)*, pp. 128–133, Hyderabad, India, December 2008.
- [23] 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.
- [24] R. Bloomfield, E. Mazhari, J. Hawkins, and Y.-J. Son, "Interoperability of manufacturing applications using the Core Manufacturing Simulation Data (CMSD) standard information model," *Computers & Industrial Engineering*, vol. 62, no. 4, pp. 1065–1079, 2012.
- [25] B. P. Gan, L. P. Chan, and S. J. Turner, "Interoperating simulations of automatic material handling systems and manufacturing processes," in *Proceedings of the 2006 Winter Simulation Conference*, pp. 1129–1135, Sacramento, CA, USA, January 2006.
- [26] G. Pedrielli, M. Sacco, W. Terkaj, and T. Tolio, "An HLA-based distributed simulation for networked manufacturing systems analysis," *Journal of Simulation*, vol. 6, no. 4, pp. 237–252, 2012.
- [27] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke, "Future modeling and simulation of CPS-based factories: an example from the automotive industry," *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 97–102, 2016.
- [28] J. Y. Lee, H. S. Kang, S. D. Noh, J. H. Woo, and P. Lee, "NESIS: a neutral schema for a web-based simulation model exchange service across heterogeneous simulation software," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 10, pp. 948–969, 2011.
- [29] M. Varshney, K. Pickett, and R. Bagrodia, "A live-virtual-constructive (LVC) framework for cyber operations test, evaluation and training," in *proceeding of the 2011 Military Communications Conference*, pp. 1387–1392, Baltimore, MA, USA, November 2011.
- [30] A. Tolk, "Interoperability and Composability," *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, pp. 403–433, John Wiley & Sons, Hoboken, NJ, USA, 2010.

- [31] IEEE Std 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules*, IEEE Standard, Piscataway, NJ, USA, 2000.
- [32] IEEE Std 1516.1-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification*, IEEE Standard, Piscataway, NJ, USA, 2001.
- [33] IEEE Std 1516.2-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification*, IEEE Standard, Piscataway, NJ, USA, 2001.
- [34] B. S. Kim, Y. Jin, and S. Nam, "An integrative user-Level customized modeling and simulation environment for smart manufacturing," *IEEE Access*, vol. 7, pp. 186637–186645, 2019.
- [35] 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.
- [36] B. S. Kim and T. G. Kim, "Cooperation of simulation and data model for performance analysis of complex systems," *International Journal of Simulation Modelling*, vol. 18, no. 4, pp. 608–619, 2019.
- [37] G. Reinhart and G. Wünsch, "Economic application of virtual commissioning to mechatronic production systems," *Production Engineering*, vol. 1, no. 4, pp. 371–379, 2007.
- [38] C. G. Lee and S. C. Park, "Survey on the virtual commissioning of manufacturing systems," *Journal of Computational Design and Engineering*, vol. 1, no. 3, pp. 213–222, 2014.

Research Article

Position Tracking of a Pneumatic-Muscle-Driven Rehabilitation Robot by a Single Neuron Tuned PID Controller

Jun Zhong ¹, Yue Zhu,¹ Chun Zhao,¹ Zhenfeng Han,² and Xin Zhang³

¹College of Mechanical & Electrical Engineering, Hohai University, Changzhou 213022, Jiangsu Province, China

²HRG Institute (Hefei) of International Innovation, Hefei 230000, Anhui Province, China

³Technology Center, Xinxing Cathay International Group, Beijing 100070, China

Correspondence should be addressed to Jun Zhong; zhongjun@hhu.edu.cn

Received 9 July 2020; Revised 9 September 2020; Accepted 26 September 2020; Published 8 October 2020

Academic Editor: Yanan Li

Copyright © 2020 Jun Zhong 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.

Pneumatic muscle actuators (PMAs) own compliant characteristics and are suitable for use in rehabilitation equipment. This paper introduces a rehabilitation robot driven by PMAs devised in the Rehabilitation and Medical Robot Laboratory. Considering high nonlinearities inside PMAs, a single neuron tuned PID controller is carefully designed. Experimental setup is built up and trials are performed. Results demonstrate the proposed advanced PID algorithm can achieve better capacity in position tracking than the conventional PID controller.

1. Introduction

Features of physiological function recession in aging process include decreased limb flexibility, osteoporosis, muscle atrophy, and significant decline in loading capacity of bone tissue. The incidence of acute cardiovascular and cerebrovascular diseases and that of neurological diseases in the aged keep at a high level, and most of these patients have symptoms of hemiplegia [1, 2]. Quantity of patients with limb dyskinesia caused by other diseases, sports injury, and traffic accidents is increasing rapidly. For these patients, besides early surgical treatments and necessary medications, correct and scientific rehabilitation training plays an important role in restoration and improvement of limb motion functions [3]. Many patients suffer from muscle atrophy and lose limb mobility because of improper training methods and failure to get effective rehabilitation training. This unfortunately makes patients suffer tremendously and causes great burden to family and society [4, 5]. A lot of clinical practices have proved that, without scientific and sufficient rehabilitation training, many patients with impairment of limb motion functions cannot restore the walking capacity to normality and have to move in a typical asymmetric attitude. Thus, improvements in patients' walking capacity

and the ability to take care of themselves in lives make a lot of sense to both patients and society. However, dependence only on physiotherapists cannot meet requirements of patients' rehabilitation because the amount of physiotherapists is small and one physiotherapist lacks the capacity of training several patients simultaneously. Another barrier is that different physiotherapists have different understandings about rehabilitation exercises and different evaluating indicators [6–10]. The above obstacles are harmful in achieving ideal training strength and effects. Besides, loss of necessary means of detection and control makes obstacles in meeting the needs of a large number of patients in modern society. Rehabilitation robots can help patients complete kinds of motion function rehabilitation training, such as arm restoring therapy and ankle rehabilitation [11–13]. Wu et al. designed a three-degrees-of-freedom lower rehabilitation robot involving hip, knee, and ankle joints and proposed an adaptive robust subcontroller for the robot to handle system uncertainties and disturbances from patients [14]. Banala et al. developed a robot-assisted gait training algorithm and used a force-field controller to achieve more effective training [15]. Besides, other rehabilitation robots have been developed [16–29]. Fateh and Khoshdel presented a new voltage-based adaptive impedance force control for a lower

limb rehabilitation robot, and gradient descent algorithm was adopted to tune impedance parameters to guarantee force controlling effect [30].

Most of the presented rehabilitation robots adopted motors as actuators, which leads the lack of compliance during recovery. In this research, a compliant ankle-rehabilitation robot driven by PMAs is presented and an advanced PID algorithm is devised to handle nonlinearities and disturbances inside the robot. This paper is arranged as follows. Section 2 introduces characteristics of the devised pneumatic muscle actuator-driven ankle rehabilitation robot. Section 3 proposes a single neuron adaptive PID controller for the robot; Section 4 establishes the experimental setup of the robot and performs the rehabilitation tests to validate the designed controller.

2. Structure of the Ankle-Rehabilitation Robot

2.1. Structure Description. The detailed structure of the ankle rehabilitation robot is addressed prudently and shown in Figure 1. Decomposition diagram of the ankle joint mechanism is displayed in Figure 2. One of the crucial issues to be handled is that placing angle and height of the training device must own the capacity of adjustment according to different patients and circumstances. Thus, the robot framework is made of aluminum profiles whose installing grooves are in unique T-shape (shown in Figure 3). By this way, arrangement angle and height of the robot can be adjusted arbitrarily in the framework. Besides, robot mounting panel has slots and holes. Slots allow the robot to move along and be fixed by bolts. Holes help the robot rotate around to modulate the arrangement angles and guarantee comfort during rehabilitation exercises. The panel framework is made up of aluminum profiles, whilst the robot is fabricated by light and high tensile aluminum. Another important issue is to guarantee safety during training exercises. This is realized by the limitation of the allowable angle range of the executive mechanism. Actually, angle limiting device (shown in Figure 4) is devised in the robot to avoid hurt on the patients' ankles. The range of the executive mechanism in the robot is $(-30^\circ, 30^\circ)$.

In traditional ankle rehabilitation robots, driving components are usually installed on one side of the ankle joint. This type of unilateral asymmetric structure (shown in Figure 5(a)) may exert uncertain force to joint, produces harmful torque on the robot joint in extra directions, and has detrimental influence on mechanical structure stability. Bilateral structure is adopted in the ankle joint design in order to preventing from producing harmful torques in extra directions. This bilateral structure keeps balance in different components of ankle rehabilitation robot and eliminates harmful torques created by drive power and forces from the feet of patients. Power supplying mechanism and transmission mechanism are both installed bilaterally about the ankle joint, which makes identical forces on the two sides of the structure. The symmetrical configuration can further guarantee the stability and reliability of mechanical transmission, make good use of space, and improve compact layout of the robot (shown in Figure 5(b)).

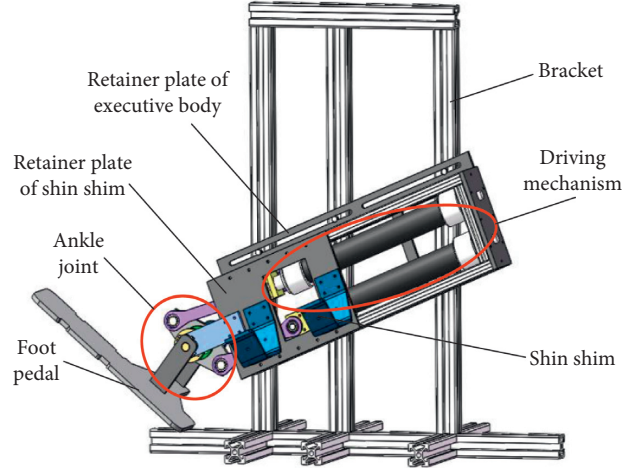


FIGURE 1: CAD model of the ankle rehabilitation robot.

2.2. Prototype. The prototype of the robot (shown in Figure 6) adopts aluminum alloy to make sure of light weight and high strength. Aluminum alloy profile frame is used to regulate posture of the robot for various patients. A torque sensor is installed in one side of the output shaft of the transmission mechanism to measure driving torque or the human ankle, and an incremental encoder is fixed with the other side of the identical output shaft to record the real time rotation angle.

3. Single Neuron Tuned Adaptive PID Controller

3.1. Single Neuron Adaptive PID Algorithm. Proportional Integral Differential (i.e., PID) algorithm is widely used in various rehabilitation and medical robots because of concision, efficiency, and reliability. However, high nonlinearity and strong hysteresis of PMAs bring a tricky job in keeping high tracking capacity because of which conventional PID algorithm owns constant values of parameters P , I , and D . Considering single neuron strategy has the excellent capacity in regulating structural parameters, a single neuron tuned PID controller is devised, as shown in Figure 7. ω_1 , ω_2 , and ω_3 are structural parameters of single neuron algorithm and represent P , I , and D of the PID controller, respectively. Values of ω_1 , ω_2 , and ω_3 are online regulated according to some criterions, and this paper adopts a hybrid of supervisory delta learning rule and nonsupervisory Hebbian rule. A cost function is defined as the evaluation indicator of the tuning algorithm in the following equation:

$$\Theta(k) = \frac{1}{2} [\theta_{\text{ref}}(k) - \theta(k)]^2. \quad (1)$$

The principle is to reduce $\Theta(k)$ by online tuning parameters ω_1 , ω_2 , and ω_3 . Gradient descent algorithm is used as follows:

$$\omega_i(k+1) = \omega_i(k) - \eta_i \frac{\partial \Theta(k)}{\partial \omega_i(k)}, \quad i = 1, 2, 3. \quad (2)$$

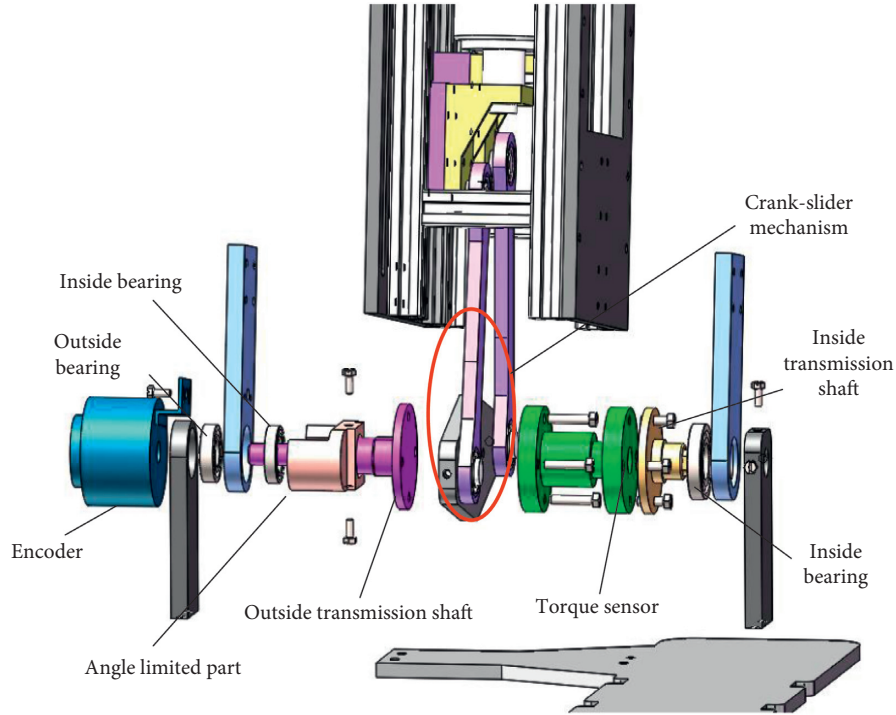


FIGURE 2: Decomposition diagram of the ankle joint mechanism.

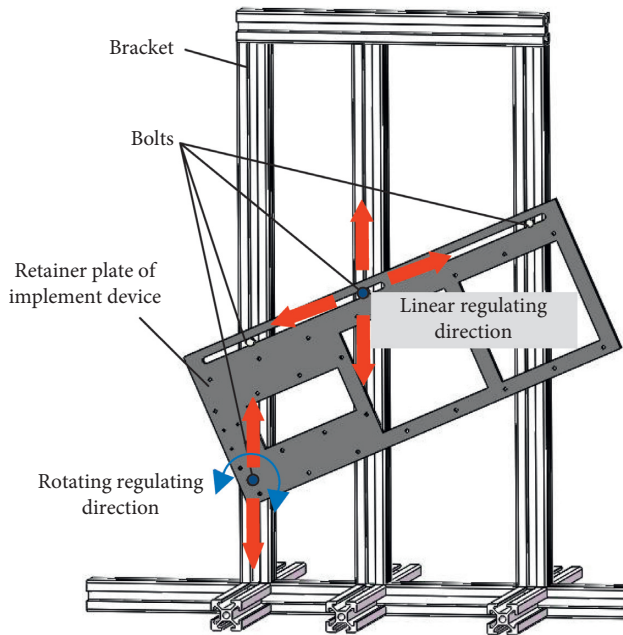


FIGURE 3: Regulating device for installing angle and height of the robot.

According to chain rule of differential equations, $\partial\Theta/\partial\omega_i(k)$ is calculated as follows:

$$\frac{\partial\Theta}{\partial\omega_i} = \frac{\partial\Theta}{\partial\theta} \frac{\partial\theta}{\partial u_{pid}} \frac{\partial u_{pid}}{\partial\omega_i} = -e(k) \cdot \frac{\partial\Theta}{\partial u_{pid}} \cdot x_i. \quad (3)$$

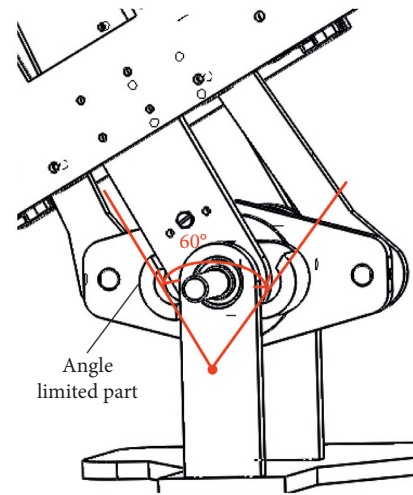


FIGURE 4: Angle limited part.

Usually, $\partial\Theta/\partial u_{pid}$ is difficult to calculate because of complexity of the actual system. To simplify the calculation, it is replaced by $\text{sgn}[\partial\theta/\partial u_{pid}]$. x_i ($i=1, 2, 3$) are inputs of neuron units ω_1 , ω_2 , and ω_3 , respectively. In the single neuron PID controller, x_i are assigned according to tracking errors of the system, i.e.,

$$\begin{aligned} x_1 &= e(k), \\ x_2 &= e(k) - e(k-1), \\ x_3 &= e(k) - 2e(k-1) + e(k-2). \end{aligned} \quad (4)$$

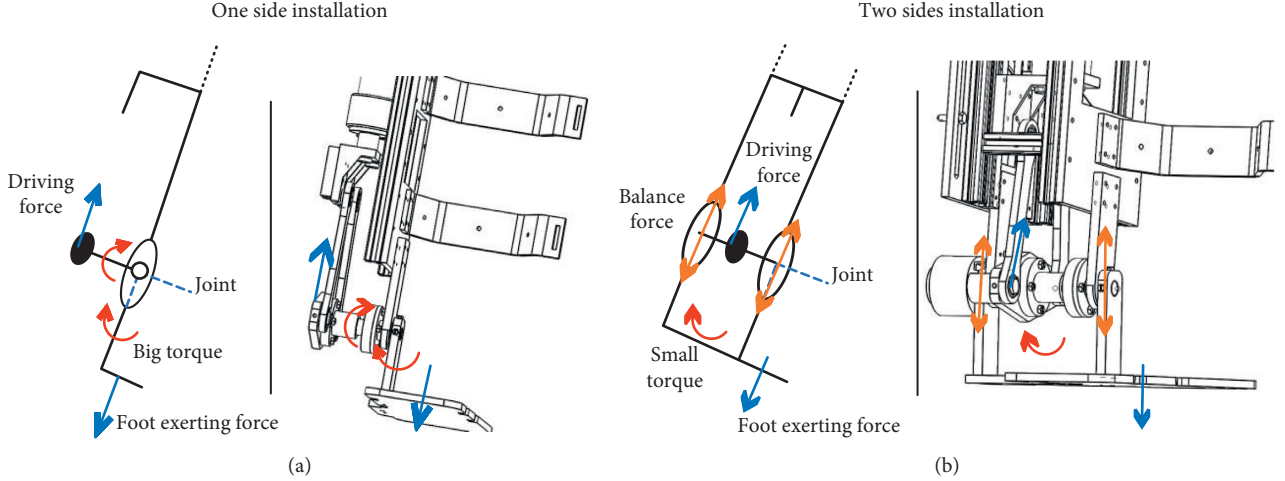


FIGURE 5: Comparison of two installing types of driving mechanism: (a) unilateral asymmetric structure; (b) symmetrical configuration.

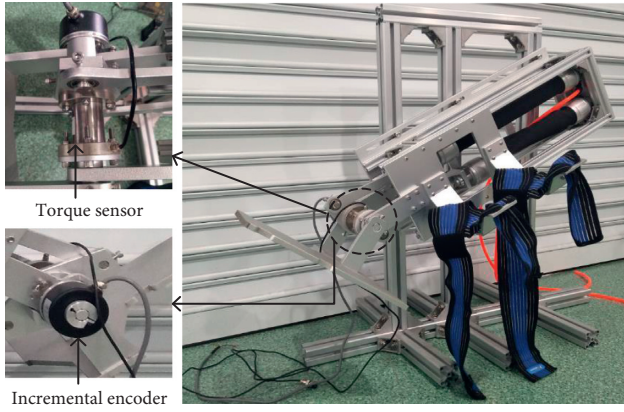


FIGURE 6: Prototype of the rehabilitation ankle robot driven by PMAs.

The output u_{pid} is as follows:

$$u_{\text{pid}} = \frac{K \sum_{i=1}^3 |x_i \omega_i|}{\sum_{i=1}^3 |\omega_i|}. \quad (5)$$

3.2. Controller Design of the Rehabilitation Robot. The controller is devised using single neuron strategy tuned PID algorithm in Figure 7. Two electrical proportional valves with type of ITV1050-212N made by SMC Corporation are used in the setup to control the pressure inside the pair of PMAs. This type valve has an output range of pressure in (0.005, 0.9) MPa and linear correspondent controlling voltage in (0, 5) V. u_{01} in Figure 8 is the initial input voltage of the first valve, i.e., the initial pressure inside the first PMA. Similarly, u_{02} is the initial input voltage of the second valve, i.e., the initial pressure inside the second PMA. Values of u_{01} and u_{02} are tuned by the trial and error. $P_{\text{valve}}(u)$ is the transfer function of the proportional valve and has the following form:

$$P_{\text{valve}}(u) = 0.179u + 0.005. \quad (6)$$

Torque switch in the controlling block is used to guarantee the safety of patients by setting a range of torque. If the sampled torque value is larger than the upper limit T_{max} , or less than lower limit T_{min} , proportional valves will output zero to PMAs and the robot stops working. T_{max} and T_{min} are acquired by trial and error.

3.3. Stability Analysis of the Proposed Controller. Controlling parameters of single neuron PID algorithm vary along with the direction of negative gradient descent of ω_1 , ω_2 , and ω_3 . Stability of the single neuron PID controller is analyzed by Lyapunov principle. Firstly, a Lyapunov function is defined as follows:

$$\mathbf{E}(\mathbf{k}) = \Theta(\mathbf{k}) = \frac{1}{2} e^2(k). \quad (7)$$

Variation of $\mathbf{E}(\mathbf{k})$ in the self-learning process of the single neuron model is expressed:

$$\Delta \mathbf{E}(\mathbf{k}) = \frac{1}{2} e^2(k+1) - \frac{1}{2} e^2(k). \quad (8)$$

Similarly, variation of \mathbf{e} can be acquired as follows:

$$e(k+1) = e(k) + \sum_{i=1}^3 \frac{\partial e(k)}{\partial \omega_i(k)} \Delta \omega_i(k). \quad (9)$$

Considering

$$\Delta \omega_i(k) = -\eta_i \frac{\partial \Theta(k)}{\partial e(k)} \frac{\partial e(k)}{\partial \omega_i(k)} = -\eta_i e(k) \frac{\partial e(k)}{\partial \omega_i(k)}, \quad (10)$$

then

$$\Delta e(k) = e \sum_{i=1}^3 \frac{\partial e(k)}{\partial \omega_i(k)} \eta_i \Delta \omega_i(k) = -e \widehat{\mathbf{O}}^T \boldsymbol{\eta} \widehat{\mathbf{O}}, \quad (11)$$

where $\widehat{\mathbf{O}} = [\partial e(k) \partial \omega_1(k) \partial e(k) \partial \omega_2(k) \partial e(k) \partial \omega_3(k)]^T$, $\boldsymbol{\eta} = \text{diag}[\eta_1 \ \eta_2 \ \eta_3]$,

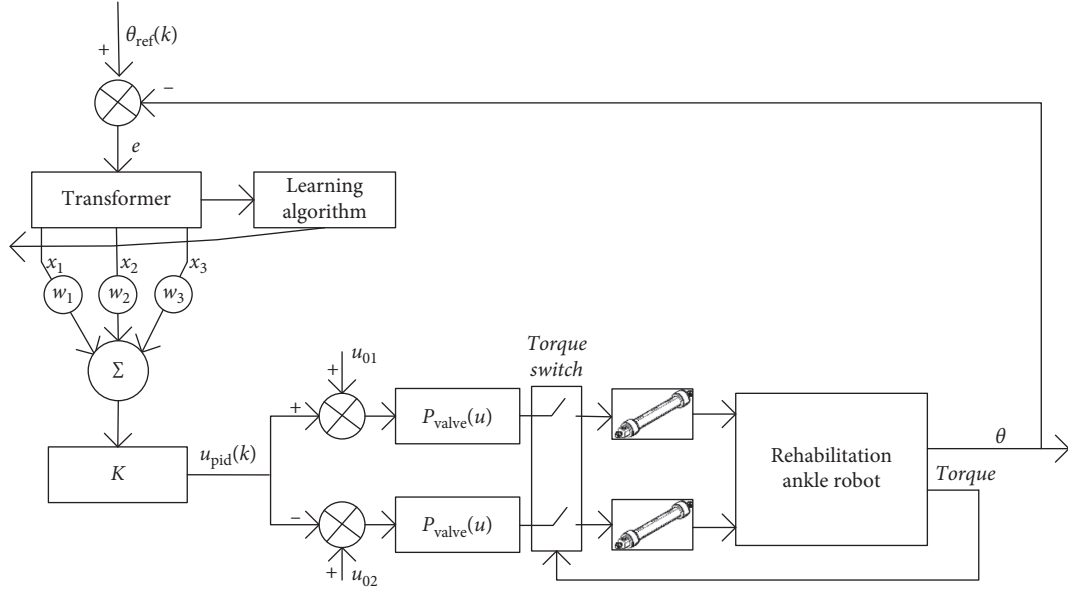


FIGURE 7: Tracking controller of the robot in passive rehabilitation therapy.

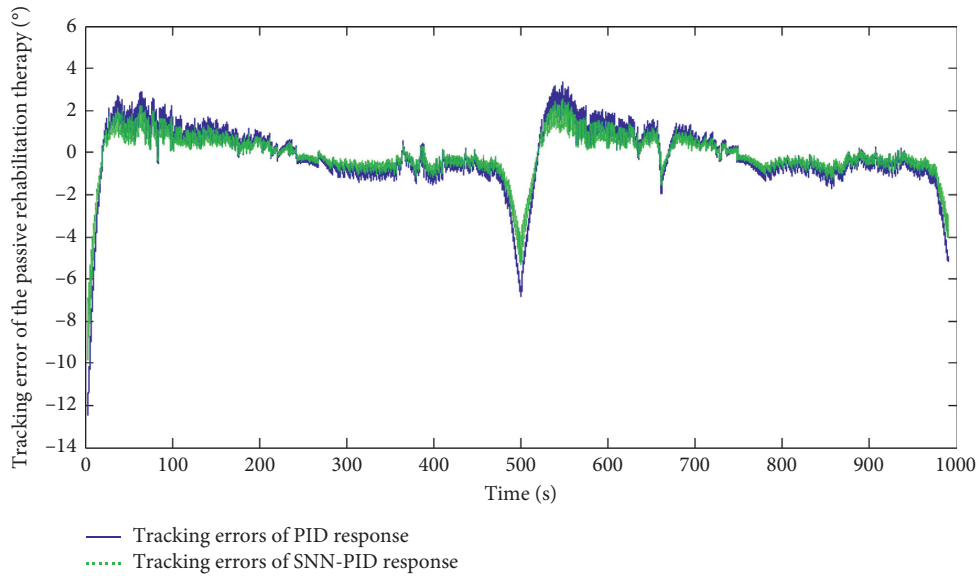


FIGURE 8: Tracking errors of passive rehabilitation therapy experiment under a half sinusoidal wave excitation of 0.002 Hz with amplitude 50°.

Thus,

$$\Delta \mathbf{E}(\mathbf{k}) = -\frac{1}{2}(\mathbf{e}(k)\hat{\mathbf{O}})^T \left[2\boldsymbol{\eta} - \boldsymbol{\eta}\hat{\mathbf{O}}\hat{\mathbf{O}}^T \boldsymbol{\eta} \right] (\mathbf{e}(k)\hat{\mathbf{O}}). \quad (12)$$

Several conclusions can be acquired by Lyapunov principle:

- $\mathbf{E}(\mathbf{k})$ is positive definite
- If and only if $2\boldsymbol{\eta} - \boldsymbol{\eta}\hat{\mathbf{O}}\hat{\mathbf{O}}^T \boldsymbol{\eta} > 0$, $\Delta \mathbf{E}(\mathbf{k})$ is negative definite, which means stability of the system depends on the learning step
- When k approaches infinity, $\mathbf{E}(\mathbf{k})$ approaches zero

Obviously, proper values of $\boldsymbol{\eta}$ make $\Delta \mathbf{E}(\mathbf{k}) < 0$, which means the controlling system keeps stable.

4. Experimental Validations and Discussion

The entire robot system, including controlling component and air compressor, is shown in Figure 9. An AD/DA card of USB3102A type from Beijing Art Technology Development Co., Ltd., is adopted to control the robot and samples all status information from sensors. An air compressor is employed as the compressed air supplier. Main components in the robot are listed in Table 1.

To validate the effectiveness of the proposed single neuron tuned PID algorithm, several trials are conducted under different excitation. Experiments are performed on a male with weight of 77 kg and height of 176 mm in this study (shown in Figure 10). It is generally known that passive rehabilitation therapy is used for serious dyskinesia of limbs and must operate

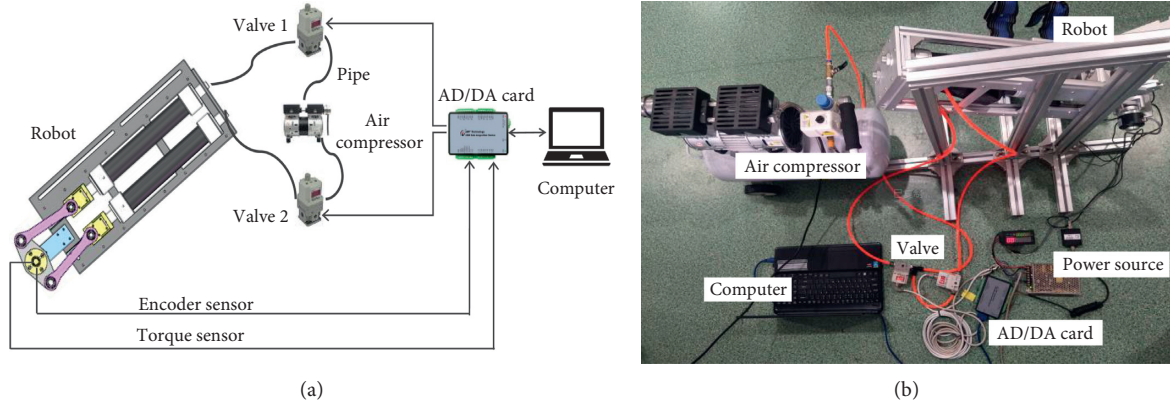


FIGURE 9: The PMAs driven rehabilitation ankle robot: (a) schematic drawing; (b) prototype of the experimental setup.

TABLE 1: Components of the PMA-driven rehabilitation ankle robot.

Name	Type	Manufacturer	Main parameters
PMA	DMSP-40-300N-RMCM	FESTO corporation	Nominal inner diameter: 40 mm Nominal length: 300 mm Pressure range: 0–0.6 MPa Maximum force: 6000 N Maximum contractile ratio: 25%
Valve	ITV1050-212N	SMC corporation	Controlling voltage range: 0–5 V Output pressure range: 0.005–0.9 MPa
Incremental encoder	ZMK60		Working voltage: DC 5 V Maximum frequency response: 300 kHz
Torque sensor	ZNNT-F	CHINO SENSOR	Load range: 2–200 Nm Output signal: 0–5 V, 0–10 V, 4–20 mA, 0–10 mA Comprehensive accuracy: 0.2%
AD/DA card	USB3102A	ART technology corporation	AD module: Resolution: 16 bit Maximum sampling speed: 250 Ksps Range: ± 10 V, ± 5 V, ± 2 V, ± 1 V DA module: Resolution: 16 bit Maximum sampling speed: 100 Ksps
Computer	INS14-3476	DELL corporation	Processor: Intel i5-7200U Storage disk: 500 G RAM: 4 GB
Air compressor	DET750-30L	DAERTUO corporation	Work power: 750 W Exhaust air capacity: 120 L/min Maximum pressure: 0.8 MPa

at a very low speed. A half sinusoidal wave excitation of 0.004 Hz with amplitude 50° is applied to the robot, and tracking responses and errors are plotted in Figures 11 and 12, respectively. Relative large errors occur at the zero position due to assembly errors of the robot and creep and high nonlinearities inside PMA. Figure 11(a) shows that the maximum tracking error of the PID controller is 5.905° at the time of 548.3 s, whilst that of single neuron tuned PID is 4.541° . Figure 11(b) shows an enlarged part for the portion A in Figure 11(a), from where the conclusion that single neuron tuned PID controller achieves smaller tracking errors than conventional PID controller can be drawn. Figure 12 shows tracking errors of passive

rehabilitation therapy in Figure 11, which further proves the better capacity of single neuron regulated PID than classic PID. Figure 13 displays a half sinusoidal wave excitation of 0.002 Hz with amplitude 50° and different responses of the adaptive PID controller and classic PID controller. Figure 13(a) shows a maximum tracking error of 3.364° from PID controller responses and a maximum tracking error of 2.603° . Figure 13(b) shows an enlarged part for the portion A in Figure 13(a), which further proves that the single neuron tuned PID controller achieves smaller tracking errors than the conventional PID controller. Figure 8 shows the corresponding tracking error performances in Figure 13, and comparison between response



FIGURE 10: Experiment validations on a male with weight of 77 kg and height of 176 mm.

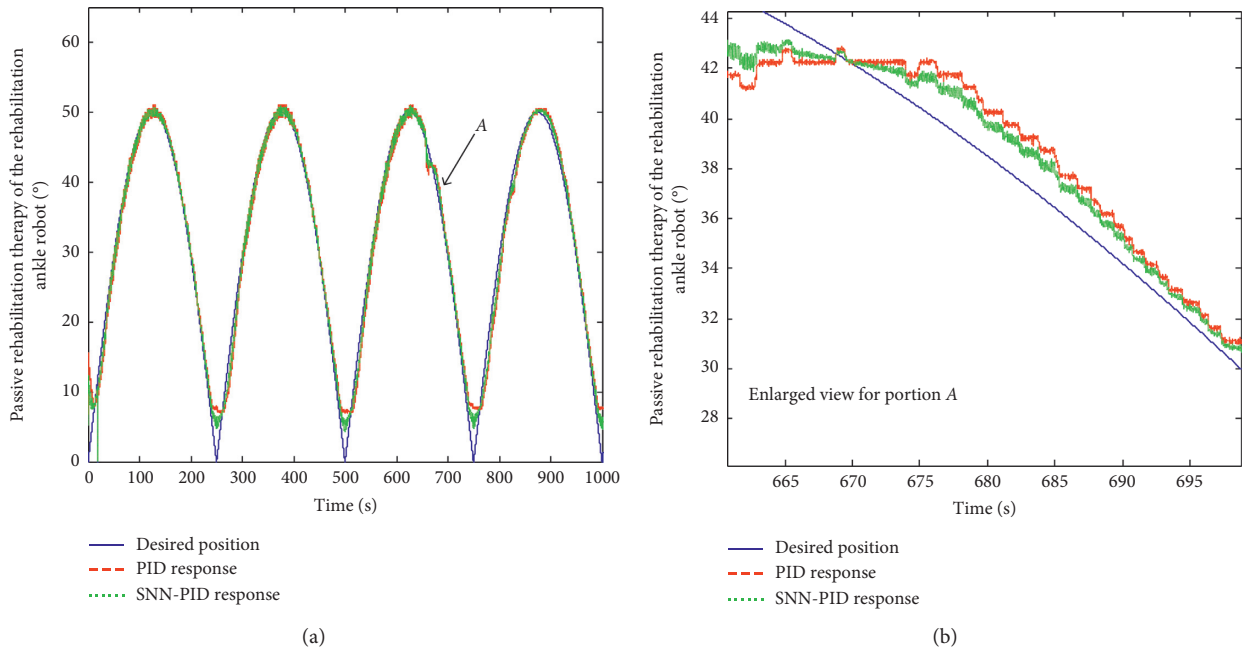


FIGURE 11: Passive rehabilitation therapy experiment under a half sinusoidal wave excitation of 0.004 Hz with amplitude 50°: (a) tracking performance; (b) enlarged view for portion A in (a).

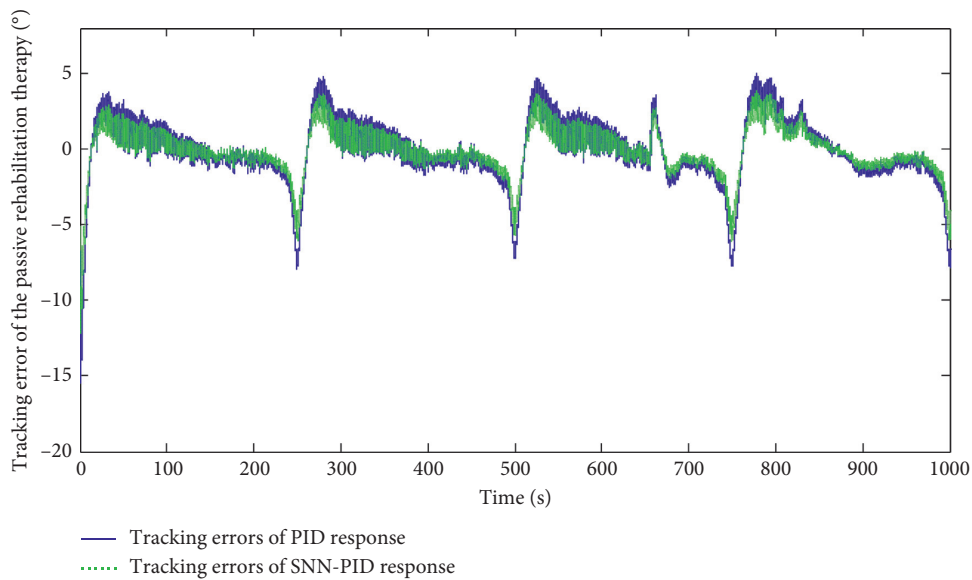


FIGURE 12: Tracking errors of passive rehabilitation therapy experiment under a half sinusoidal wave excitation of 0.004 Hz with amplitude 50°.

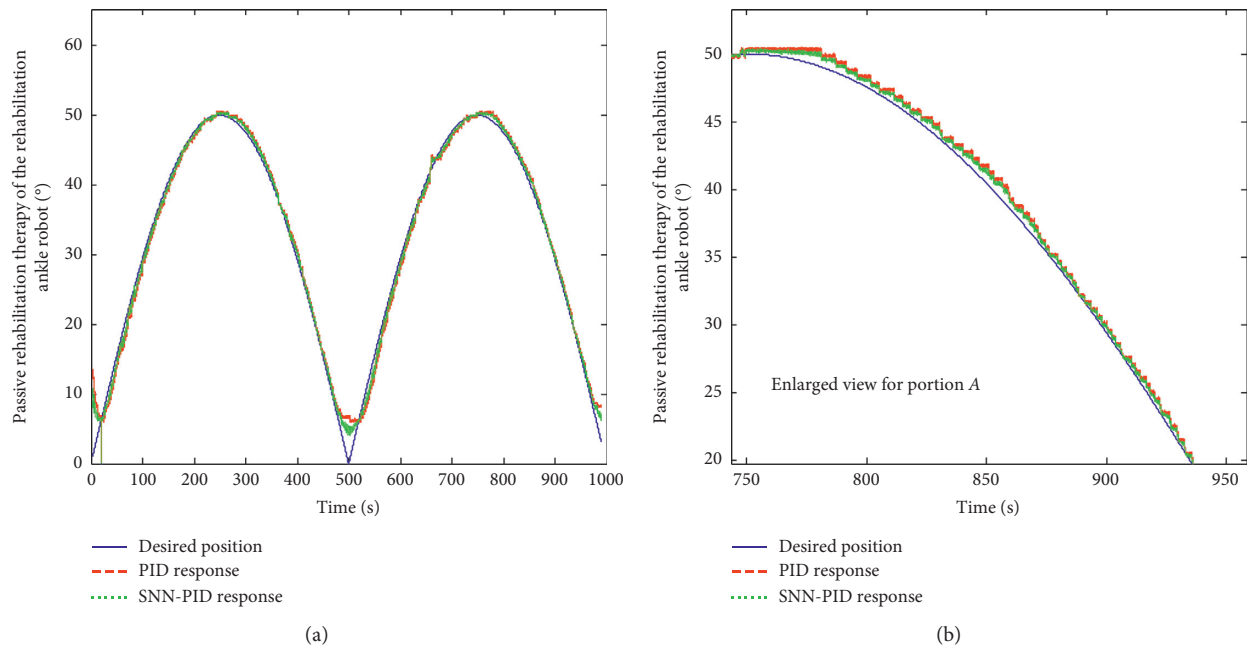


FIGURE 13: Passive rehabilitation therapy experiment under a half sinusoidal wave excitation of 0.002 Hz with amplitude 50° : (a) tracking performance; (b) enlarged view for portion A in (a).

curves of the two controllers demonstrates better capacity of single neuron tuned PID algorithm in overcoming nonlinearities and disturbances of the robot.

5. Conclusions

Compliant ankle rehabilitation robot in this research is realized by employing pneumatic muscle actuators as the power source. A pair of PMAs is arranged in antagonistic form and provides driving torque through crank-slider mechanism. Prototype is manufactured and assembled. A single neuron tuned PID controller with torque safety switch is designed for the robot. Passive rehabilitation experiments are conducted and effectiveness of the adaptive controller is validated. Conclusion of experiments demonstrates that the devised controlling algorithm can improve accuracy of position tracking of the robot.

Data Availability

The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grants 2019YFC1511400, 2019YFC1511401, and 2019YFC1511402, Pre-research

Union Fund of China Ministry of Education && PLA Equipment Development Department under Grant 6141A02033616, Fundamental Research Funds for the Central Universities under Grant B200202223, Strategic Rocket Innovation Fund of National Defense Science and Technology Industry under Grant Y18143, Natural Science Foundation of Jiangsu Province under Grant BK20170301, and Opening Fund of Jiangsu Key Laboratory of Special Robot Technology under Grant 2017JSJR01.

References

- [1] S. Okada, T. Sakaki, R. Hirata, Y. Okajima, S. Uchida, and Y. Tomita, "TEM: a therapeutic exercise machine for the lower extremities of spastic patients," *Advanced Robotics*, vol. 14, no. 7, pp. 597–606, 2001.
- [2] D. Bradley, C. Acosta-Marquez, M. Hawley, S. Brownsell, P. Enderby, and S. Mawson, "NeXOS—the design, development and evaluation of a rehabilitation system for the lower limbs," *Mechatronics*, vol. 19, no. 2, pp. 247–257, 2009.
- [3] M. Mawson, H. A. M. Rittenhouse, and H. A. Abdullah, "Design issues for therapeutic robot systems: results from a survey of physiotherapists," *Journal of Intelligent and Robotic Systems*, vol. 42, no. 3, pp. 239–252, 2005.
- [4] P. Métrailler, V. Blanchard, I. Perrin et al., "Improvement of rehabilitation possibilities with the motion maker TM," in *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics & Biomechatronics, 2006. BioRob 2006*, Pisa, Italy, February 2006.
- [5] M. L. Aisen, H. I. Krebs, N. F. Hogan, and B. T. Volpe, "The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke," *Archives of Neurology*, vol. 54, no. 4, pp. 443–446, 1997.
- [6] C. McDowell, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang, "Biologically inspired motion modeling and neural control for robot learning from demonstrations," *IEEE Transactions on*

- Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 281–291, 2019.
- [7] T. Tsuji and Y. Tanaka, “Bio-mimetic impedance control of robotic manipulator for dynamic contact tasks,” *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 306–316, 2008.
 - [8] C. P. Bechlioulis and G. A. Rovithakis, “Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems,” *Automatica*, vol. 45, no. 2, pp. 532–538, 2009.
 - [9] C. P. Bechlioulis and G. A. Rovithakis, “Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance,” *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2090–2099, 2008.
 - [10] R. Riener, S. L. Lunenburger, S. Jezernik, M. Anderschitz, G. Colombo, and V. Dietz, “Patient-cooperative strategies for robot-aided treadmill training: first experimental results,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 380–394, 2005.
 - [11] C. Dietz, J. Luo, C. Liu, M. Li, and S.-L. Dai, “Haptics electromyography perception and learning enhanced intelligence for teleoperated robot,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1512–1521, 2019.
 - [12] U. Demir, S. Kocaoglu, and E. Akdogan, “Human impedance parameter estimation using artificial neural network for modelling physiotherapist motion,” *Biocybernetics and Biomedical Engineering*, vol. 36, no. 2, Article ID S0208521616300031, 2016.
 - [13] S. Mohan, J. K. Mohanta, S. Kurtenbach, J. Paris, B. Corves, and M. Huesing, “Design, development and control of a 2PRP-2PPR planar parallel manipulator for lower limb rehabilitation therapies,” *Mechanism and Machine Theory*, vol. 112, pp. 272–294, 2017.
 - [14] J. Wu, J. Gao, R. Song, R. Li, Y. Li, and L. Jiang, “The design and control of a 3DOF lower limb rehabilitation robot,” *Mechatronics*, vol. 33, pp. 13–22, 2016.
 - [15] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, “Robot assisted gait training with active leg exoskeleton (ALEX),” *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, vol. 17, no. 1, pp. 2–8, 2009.
 - [16] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.-Y. Su, “Personalized variable gain control with tremor attenuation for robot teleoperation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1759–1770, 2017.
 - [17] R. Pérez-Rodríguez, C. Rodríguez, Ú. Costa et al., “Anticipatory assistance-as-needed control algorithm for a multijoint upper limb robotic orthosis in physical neurorehabilitation,” *Expert Systems with Applications*, vol. 41, no. 8, pp. 3922–3934, 2014.
 - [18] S. Hussain, S. Q. Xie, and P. K. Jamwal, “Adaptive impedance control of a robotic orthosis for gait rehabilitation,” *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1025–1034, 2013.
 - [19] W. Meng, Q. Liu, Z. Zhou, and Q. Ai, “Active interaction control applied to a lower limb rehabilitation robot by using EMG recognition and impedance model,” *Industrial Robot: An International Journal*, vol. 41, no. 5, pp. 465–479, 2014.
 - [20] K. Liu, J.-C. Zhao, E.-G. Cao, and X. Han, “Control methods for standing-up rehabilitation robot based on estimation and analysis of lower limb kinetics,” *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)*, vol. 45, no. 3, pp. 837–843, 2015.
 - [21] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, “Robot learning system based on adaptive neural control and dynamic movement primitives,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
 - [22] F. Yongfei, H. Wang, H. Yan, X. Wang, Z. Jin, and L. Vladareanu, “Research on safety and compliance of a new lower limb rehabilitation robot,” *Journal of Healthcare Engineering*, vol. 2017, Article ID 1523068, 11 pages, 2017.
 - [23] W. Meng, Q. Liu, Z. Zhou, Q. Ai, B. Sheng, and S. Xie, “Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation,” *Mechatronics*, vol. 31, pp. 132–145, 2015.
 - [24] J. Yoon, B. Novandy, C.-H. Yoon, and K.-J. Park, “A 6-DOF gait rehabilitation robot with upper and lower limb connections that allows walking velocity updates on various terrains,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 201–215, 2010.
 - [25] S. M. M. Rahman and R. Ikeura, “A novel variable impedance compact compliant ankle robot for overground gait rehabilitation and assistance,” *Procedia Engineering*, vol. 41, pp. 522–531, 2012.
 - [26] H. Stefan, J. Mehrholz, and C. Werner, “Robot-assisted upper and lower limb rehabilitation after stroke: walking and arm/hand function,” *Deutsches Rzteblatt International*, vol. 105, no. 18, pp. 330–336, 2008.
 - [27] X. H. Shi, H. B. Wang, L. Yuan, Z. Xu, H. W. Zhen, and Z. G. Hou, “Design and analysis of a lower limb rehabilitation robot,” *Advanced Materials Research*, vol. 490–495, pp. 2236–2240, 2012.
 - [28] N. Wang, C. Chen, and A. Di Nuovo, “A framework of hybrid force/motion skills learning for robots,” *IEEE Transactions on Cognitive and Developmental Systems*, p. 1, 2020, In press.
 - [29] Y. Pei, Y. Kim, G. Obinata, K. Hase, and D. Stefanov, “Trajectory planning of a robot for lower limb rehabilitation,” in *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1259–1263, Boston, MA, USA, August 2011.
 - [30] M. M. Fateh and V. Khoshdel, “Voltage-based adaptive impedance force control for a lower-limb rehabilitation robot,” *Advanced Robotics*, vol. 29, no. 15, pp. 961–971, 2015.

Research Article

Synchronous Analysis for Fuzzy Coupled Neural Networks with Column Pinning Controllers

Dawei Gong ¹, Shijie Song,¹ Michel Lopez,² and Edgar N. Sanchez²

¹*Mechatronics Engineering of University of Electronic Science and Technology of China, Chengdu 611731, China*

²*Advanced Studies and Research Center of the National Polytechnic Institute, Guadalajara 07360, Mexico*

Correspondence should be addressed to Dawei Gong; pzhzhx@126.com

Received 9 July 2020; Accepted 16 September 2020; Published 5 October 2020

Academic Editor: Yanan Li

Copyright © 2020 Dawei Gong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The synchronous research for fuzzy coupled neural networks (FCNNs) is studied by a new strategy of column pinning controllers. In this paper, the Lyapunov Krasovskii functional (LKF) is taken as an important element for the pinning control laws. The networks are interconnected by coupling gains that define a physical interaction graph. Different from the preset technique in traditional intermittent control, a novel additional communication control graphs of pinning control law are introduced, which has not been investigated before. The proposed control laws can achieve the control objectives of being introduced as an array of vector with Kronecker produce operation. Under the proposed framework of intermittent control, numerical simulations via MATLAB are used to confirm the availability of the suggested control laws.

1. Introduction

In recent decades, the investigation on neural networks (NNS) has aroused ever-increasing interest of researchers due to their strong application in various fields [1–5]. The coupled neural networks (CNNs) are seen as a special type of complex networks, which consist of a large set of interconnected single NNs with each individual being called node. Usually, the CNNs exhibit more unpredictable and complicated behaviors than the single NNs. Synchronization of CNNs describes a typical collective behavior and has many applications. For example, the complex oscillatory patterns were stored and retrieved as the synchronization states by presenting an architecture of CNNs in [6, 7]. A secure communication system was introduced by utilizing the coupled cellular NNs in [8]. The research on synchronization of CNNs not only opens up new opportunities in the understanding of brain science but also makes an important step forward to the practical applications [9].

However, the aforementioned results are valid when for the structures and the parameters of coupled neural networks are exactly known. In many practical models of the real world, uncertainty or vagueness is unavoidable. Fuzzy

theory [10–12] is considered an efficient tool to solve vagueness problems of the complex systems. Compared with the traditional NNS, the FCNNs have advantages for their capabilities in handling uncertain information and representing nonlinear dynamics [13–15].

In practice, many feasible control schemes can be adopted to study synchronous research of complex networks, such as the sampled-data control [16] and intermittent control [17]. Among them, the sampled-data control and impulsive control are two schemes with low control cost because their controllers are updated only at some discrete times. Besides, intermittent control is also an economic choice. In such scheme, the controller is only imposed on the systems at work time. Hence, the notion of intermittent control came into researchers' vision and has stimulated many renewed results. In [18], the quasi-synchronization of delayed chaotic systems was investigated by periodically intermittent control. In [19], the synchronization issues of complex networks were visited by designing an intermittent controller equipped with two switched periods. There are two categories of synchronization: self-synchronization and forced synchronization. Without any external force, the self-synchronization can be achieved by the connection of local

nodes. However, the networks usually cannot be synchronized by themselves. Therefore, it is more desirable to force the networks to synchronize. Due to the high dimension and complex topology, it will be expensive and literally infeasible to add controllers to all nodes. Hinted by such consideration, the strategy so-called pinning control is proposed which only controls a small location of the nodes, such as [20–24].

As far as we know, there are no pinning control results for FCNNs. So, how to solve the pinning synchronization problems for FCNNs is still challenging. Motivated by the foregoing discussion, this brief explores the synchronization of FCNNs by proposing the concept of column pinning control law. In the developed control scheme, the work conditions are decided by the dynamic relationships among the Lyapunov–Krasovskii functional (LKF) and some other column vectors. Namely, the pinning controller is imposed on the systems when the trajectory of LKF goes into the column regions. Our scheme changes the intrinsic characteristic of the existing control methods that the work

conditions are predetermined in prior. From the events' point of view, whether the controller is imposed or not is decided by the dynamic of LKF. Therefore, our scheme can be understood as a class of event-dependent column controllers. Under the framework of the proposed scheme, several simple criteria are developed to study the synchronization for the considered FCNNs.

Notations: N , $R^{n \times m}$, and R^n denote the sets of non-negative integers, $n \times m$ real matrices, and n -dimensional Euclidean space, respectively. For real symmetric matrix Y , $Y > 0$ ($Y \geq 0$) indicates that Y is positive definite (respectively, semidefinite). The superscript T stands for the transpose of a matrix. I_n denotes the n -dimensional identity matrix. $\text{diag}(\dots)$ represents the block-diagonal matrix.

2. Problem Formulation

Without the loss of generality, this brief considers the following FCNNs with N identical nodes:

$$\begin{aligned} \dot{z}_i(t) &= -Cz_i(t) + Af(z_i(t)) + Bf(z_i(t - \tau(t))) + \sum_{j=1}^N G_{ij}^{(1)} D_1 z_j(t) + \sum_{j=1}^N G_{ij}^{(2)} D_2 z_j(t - \tau(t)) + u_i(t), \quad i = 1, 2, \dots, h, \\ \dot{z}_i(t) &= -Cz_i(t) + Af(z_i(t)) + Bf(z_i(t - \tau(t))) + \sum_{j=1}^N G_{ij}^{(1)} D_1 z_j(t) + \sum_{j=1}^N G_{ij}^{(2)} D_2 z_j(t - \tau(t)), \quad i = h + 1, h + 2, \dots, N, \end{aligned} \quad (1)$$

in the formula, $z_i(t) = (z_{i1}(t), z_{i2}(t), \dots, z_{in}(t))^T \in R^n$, $A \in R^{n \times n}$, $B \in R^{n \times n}$, $f(z_i(t)) = (f_1(z_{i1}(t)), f_2(z_{i2}(t)), \dots, f_n(z_{in}(t)))^T$, $C = \text{diag}(c_1, c_2, \dots, c_n)$ are diagonal positive matrix, $G^{(1)} = (G_{ij}^{(1)})_{N \times N}$, $G^{(2)} = (G_{ij}^{(2)})_{N \times N}$ are the outer coupled matrix, and $D_1 \in R^{n \times n}$, $D_2 \in R^{n \times n}$ are the inner coupled matrix.

$\tau(t)$ satisfied

$$0 \leq \dot{\tau}(t) \leq \mu < 1, 0 \leq \tau(t) \leq \tau, \quad (2)$$

where μ and τ are known constants.

$u_i(t)$ are the pinning controllers.

The controllers are designed as

$$\begin{aligned} u_i(t) &= \sum_{j=1, j \neq i}^N k_{ij}^{(1)} D_3 (z_j(t) - z_i(t)) \\ &+ \sum_{j=1, j \neq i}^N k_{ij}^{(2)} D_4 (z_j(t - \tau(t)) - z_i(t - \tau(t))), \end{aligned} \quad (3)$$

where $k_{ij}^{(q)} > 0$, ($q = 1, 2$), $i = 1, 2, \dots, h$, and $k_{ij}^{(q)} = 0$, for $i = h + 1, h + 2, \dots, N$, and $k_{ij}^{(q)}$ are the control graph weights. Matrices $D_3, D_4 \in R^{n \times n}$ represent control gain matrices. These gain matrices are the control parameters designed to guarantee synchronization of the coupled neural networks.

Remark 1. It is noted that the physical coupling graphs combined with the communication control graphs together form a cyber-physical system, where in the physical

connection graph topology $G_{ij}^{(1)}$ and $G_{ij}^{(2)}$ and the communication connection graph topology $k_{ij}^{(1)}$ and $k_{ij}^{(2)}$ are fixed. The design freedom is in the selection of the control gain matrices D_3 and D_4 .

System (3) can be rewritten as

$$u_i(t) = \sum_{j=1}^N L_{ij}^{(1)} D_3 z_j(t) + \sum_{j=1}^N L_{ij}^{(2)} D_4 z_j(t - \tau(t)), \quad (4)$$

where matrix $L^{(q)} = (L_{ij}^{(q)})_{N \times N}$, ($q = 1, 2$), satisfied

$$\begin{cases} L_{ij}^{(q)} = k_{ij}^{(q)}, & i \neq j, \\ L_{ii}^{(q)} = - \sum_{j=1, j \neq i}^N k_{ij}^{(q)}, & i, j = 1, 2, \dots, N. \end{cases} \quad (5)$$

The initial variables are given as

$$z_i(s) = \Pi_{i0}(s) \in \Theta([- \tau, 0], R^n), \quad i = 1, 2, \dots, N. \quad (6)$$

Let

$$z(t) = (z_1^T(t), z_2^T(t), \dots, z_N^T(t))^T,$$

$$F(z(t)) = (f^T(z_1(t)), f^T(z_2(t)), \dots, f^T(z_N(t)))^T, \quad (7)$$

$$\tilde{U}(t) = (u_1^T(t), u_2^T(t), \dots, u_N^T(t))^T.$$

Combining with the sign \otimes of Kronecker product, system (1) can be rewritten as

$$\begin{aligned} \dot{z}(t) = & -(I_N \otimes C)z(t) + (I_N \otimes A)F(z(t)) \\ & + (I_N \otimes B)F(z(t - \tau(t))) + (G^{(1)} \otimes D_1)z(t) \\ & + (G^{(2)} \otimes D_2)z(t - \tau(t)) + \tilde{U}(t). \end{aligned} \quad (8)$$

From equation (15), we have

$$\tilde{U}(t) = (L^{(1)} \otimes D_3)z(t) + (L^{(2)} \otimes D_4)z(t - \tau(t)). \quad (9)$$

Remark 2. It is the first introduction of the pinning control laws as an array of vector with Kronecker produce operation.

Assumption 1 (see [25–27]). The outer-coupling matrix are assumed as

$$\begin{cases} G_{ij}^{(q)} = G_{ji}^{(q)} \geq 0, & i \neq j, q = 1, 2, \\ G_{ii}^{(q)} = - \sum_{j=1, j \neq i}^N G_{ij}^{(q)}, & i, j = 1, 2, \dots, N. \end{cases} \quad (10)$$

Assumption 2 (see [28–30]). For $j \in 1, 2, \dots, N, \forall s_1, s_2 \in R, s_1 \neq s_2$, the neural activation functions satisfy

$$\sigma_r^- \leq \frac{f_j(s_1) - f_j(s_2)}{s_1 - s_2} \leq \sigma_r^+. \quad (11)$$

We define

$$\begin{aligned} \Delta_1 &= \text{diag}(\sigma_1^+ \sigma_1^-, \dots, \sigma_n^+ \sigma_n^-), \\ \Delta_2 &= \text{diag}\left(\frac{\sigma_1^+ + \sigma_1^-}{2}, \dots, \frac{\sigma_n^+ + \sigma_n^-}{2}\right). \end{aligned} \quad (12)$$

From T-S fuzzy model concept, for the first time, a class of FNNS with pinning controllers is described here. Model 1 with T-S theory is described.

Rule l : if $\theta_1(t)$ is F_{l1} , $\theta_2(t)$ is $F_{l2}, \dots, \theta_g(t)$ is F_{lg} , then

$$\begin{aligned} \dot{z}_i(t) &= -C_l z_i(t) + A_l f(z_i(t)) + B_l f(z_i(t - \tau(t))) + \sum_{j=1}^N G_{ij}^{(1)} D_{1l} z_j(t) + \sum_{j=1}^N G_{ij}^{(2)} D_{2l} z_j(t - \tau(t)) + u_i(t), \quad i = 1, 2, \dots, h, \\ \dot{z}_i(t) &= -C_l z_i(t) + A_l f(z_i(t)) + B_l f(z_i(t - \tau(t))) + \sum_{j=1}^N G_{ij}^{(1)} D_{1l} z_j(t) + \sum_{j=1}^N G_{ij}^{(2)} D_{2l} z_j(t - \tau(t)), \quad i = h + 1, h + 2, \dots, N. \end{aligned} \quad (13)$$

The controllers of the fuzzy systems are assumed in the form

$$\begin{aligned} u_i(t) &= \sum_{j=1, j \neq i}^N k_{ij}^{(1)} D_{3l} (z_j(t) - z_i(t)) \\ &+ \sum_{j=1, j \neq i}^N k_{ij}^{(2)} D_{4l} (z_j(t - \tau(t)) - z_i(t - \tau(t))). \end{aligned} \quad (14)$$

Controller (14) can be rewritten as

$$u_i(t) = \sum_{j=1}^N L_{ij}^{(1)} D_{3l} z_j(t) + \sum_{j=1}^N L_{ij}^{(2)} D_{4l} z_j(t - \tau(t)), \quad (15)$$

where matrix $L^{(q)} = (L_{ij}^{(q)})_{N \times N}$, ($q = 1, 2$) are defined as

$$\begin{cases} L_{ij}^{(q)} = k_{ij}^{(q)}, & i \neq j, \\ L_{ii}^{(q)} = - \sum_{j=1, j \neq i}^N k_{ij}^{(q)}, & i, j = 1, 2, \dots, N. \end{cases} \quad (16)$$

The sign of \otimes is used to replace the Kronecker product, and FCNNs system 13 can be expressed as

$$\begin{aligned} \dot{z}(t) &= \sum_{l=1}^r \mu_l(\theta(t)) (-(I_N \otimes C_l)z(t) + (I_N \otimes A_l)F(z(t)) \\ &+ (I_N \otimes B_l)F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l})z(t) \\ &+ (G^{(2)} \otimes D_{2l})z(t - \tau(t)) + \tilde{U}(t), \end{aligned} \quad (17)$$

$\theta(t) = [\theta_1(t), \theta_2(t), \dots, \theta_g(t)]$, $\mu_l(\theta(t)) = (\omega_l(\theta(t)) / \sum_{l=1}^r \omega_l(\theta(t)))$, $F_{lj}(\theta_j(t))$ is the grade of membership of $\theta_j(t)$ in F_{lj} . $\mu_l(\theta(t))$ satisfied

$$\begin{aligned} \mu_l(\theta(t)) &\geq 0, \\ \sum_{l=1}^r \mu_l(\theta(t)) &= 1. \end{aligned} \quad (18)$$

The controllers of a set of fuzzy rules are written as follows.

Rule l : If $\theta_1(t)$ is F_{l1} , $\theta_2(t)$ is $F_{l2}, \dots, \theta_g(t)$ is F_{lg} , then

$$\begin{aligned} \tilde{U}(t) &= (L^{(1)} \otimes D_{3l})z(t) + (L^{(2)} \otimes D_{4l})z(t - \tau(t)), \\ & \quad l = 1, 2, \dots, r. \end{aligned} \quad (19)$$

The resulting FCNNs system can be rewritten as

$$\begin{aligned}
\dot{z}(t) = & \sum_{l=1}^r \sum_{k=1}^r \mu_l(\theta(t)) \mu_k(\theta(t)) (-I_N \otimes C_l) z(t) \\
& + (I_N \otimes A_l) F(z(t)) + (I_N \otimes B_l) F(z(t - \tau(t))) \\
& + (G^{(1)} \otimes D_{1l}) z(t) + (G^{(2)} \otimes D_{2l}) z(t - \tau(t)) \\
& + (L^{(1)} \otimes D_{3k}) z(t) + (L^{(2)} \otimes D_{4k}) z(t - \tau(t)).
\end{aligned} \quad (20)$$

Then, we will introduce some useful situations, which are very important to prove our main results.

Definition 1. System (17) is synchronized if the following equation holds:

$$\lim_{t \rightarrow \infty} \|z_i(t) - z_j(t)\| = 0, \quad i, j = 1, 2, \dots, N. \quad (21)$$

Lemma 1 (see [31]). Define $E_N = ee^T$, where $e = (1, 1, \dots, 1)^T$, and $U = NI_N - E_N$, $M \in R^{n \times n}$, $z = (z_1^T, z_2^T, \dots, z_N^T)^T$, and $x = (x_1^T, x_2^T, \dots, x_N^T)^T$ with $z_k, x_k \in R^n$, ($k = 1, 2, \dots, N$), then

$$z^T (U \otimes M) x = \sum_{1 \leq i < j \leq N} ((z_i - x_j)^T M (z_i - x_j)). \quad (22)$$

Lemma 2 (Jensen's inequality). For any real matrix $\Theta \in R^{n \times n}$, $\Theta^T = \Theta > 0$, constant $\varrho > 0$ and $\omega: [0, \varrho] \rightarrow R^n$, then

$$\varrho \int_0^\varrho \omega^T(s) \Theta \omega(s) ds \geq \left(\int_0^\varrho \omega(s) ds \right)^T \Theta \left(\int_0^\varrho \omega(s) ds \right). \quad (23)$$

Lemma 3 (see [32]). For symmetric constant matrix

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} & \Theta_{13} \\ * & \Theta_{22} & \Theta_{23} \\ * & * & \Theta_{33} \end{bmatrix} > 0, \quad \Theta_{kq} \in R^{n \times n}, 1 \leq q \leq 3, 1 \leq k \leq 3,$$

$0 < \tau(t) \leq \tau$, and vector function $\dot{z}(t): [-\tau(t), 0] \rightarrow R^{n \times N}$, then we have

$$\begin{aligned}
& -\tau(t) \int_{t-\tau(t)}^t \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix}^T \begin{bmatrix} U \otimes \Theta_{11} & U \otimes \Theta_{12} & U \otimes \Theta_{13} \\ * & U \otimes \Theta_{22} & U \otimes \Theta_{23} \\ * & * & U \otimes \Theta_{33} \end{bmatrix} \\
& \cdot \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix} ds, \\
& \leq -\psi^T(t) \begin{bmatrix} U \otimes \Theta_{11} & U \otimes \Theta_{12} & U \otimes \Theta_{13} & -U \otimes \Theta_{13} \\ * & U \otimes \Theta_{22} & U \otimes \Theta_{23} & -U \otimes \Theta_{23} \\ * & * & U \otimes \Theta_{33} & -U \otimes \Theta_{33} \\ * & * & * & U \otimes \Theta_{33} \end{bmatrix} \psi(t),
\end{aligned} \quad (24)$$

$$\text{where } \psi^T(t) = \left[\left(\int_{t-\tau(t)}^t z(s) ds \right)^T, \left(\int_{t-\tau(t)}^t F(z(s)) ds \right)^T, z^T(t), z^T(t - \tau(t)) \right].$$

3. Synchronization Results for Fuzzy System

First, we consider the synchronization results of FCNNs without control. Whereafter, we will establish some concise sufficient conditions which ensure synchronization of FCNNs.

3.1. Synchronization for FNNs without Control. In this section, we first study the synchronization criterions for TFNNs with time-varying delay and hybrid coupling:

$$\begin{aligned}
\dot{z}(t) = & \sum_{l=1}^r \mu_l(\theta(t)) (-I_N \otimes C_l) z(t) + (I_N \otimes A_l) F(z(t)) \\
& + (I_N \otimes B_l) F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l}) z(t) \\
& + (G^{(2)} \otimes D_{2l}) z(t - \tau(t)).
\end{aligned} \quad (25)$$

Theorem 1. For $l = 1, 2, \dots, r$, system 25 is synchronized if $P_z > 0$ ($z = 1, 2, 3$) and $J_1 > 0, J_2 > 0$, then the following formulas are holding for all $1 \leq i < j \leq N$:

$$\Theta_{ij}^l = \begin{bmatrix} \Pi_{11} & -NG_{ij}^{(2)} P_1 D_{2l} & P_1 A_l + J_1 \Delta_2 & P_1 B_l & 0 \\ * & -(1-\mu) P_2 - J_2 \Delta_1 & 0 & J_2 \Delta_2 & 0 \\ * & * & -J_1 & 0 & 0 \\ * & * & * & -J_2 & 0 \\ * & * & * & * & \frac{1-\mu}{\tau} P_3 \end{bmatrix} < 0, \quad (26)$$

where

$$\begin{aligned}
\Pi_{11} = & -P_1 C_l - C_l P_1 - J_1 \Delta_1 - NG_{ij}^{(1)} P_1 D_{1l} \\
& - NG_{ij}^{(1)} D_{1l}^T P_1^T + P_2 + \tau P_3.
\end{aligned} \quad (27)$$

Proof. Consider U as Lemma 1; for system (17), we have

$$\begin{aligned}
V_1(t) &= z^T(t) (U \otimes P_1) z(t), \\
V_2(t) &= \int_{t-\tau(t)}^t z^T(s) (U \otimes P_2) z(s) ds, \\
V_3(t) &= \int_{t-\tau(t)}^t \int_{\theta}^t z^T(s) (U \otimes P_3) z(s) ds d\theta.
\end{aligned} \quad (28)$$

Deriving time of system (17),

$$\begin{aligned}\dot{V}_1(t) &= 2z^T(t)(U \otimes P_1)\dot{z}(t), \\ &= 2z^T(t)(U \otimes P_1) \sum_{l=1}^r \mu_l(\theta(t)) [-(I_N \otimes C_l)z(t) + (I_N \otimes A_l)F(z(t)), \\ &\quad + (I_N \otimes B_l)F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l})z(t) + (G^{(2)} \otimes D_{2l})z(t - \tau(t))],\end{aligned}\quad (29)$$

$$\dot{V}_2(t) \leq z^T(t)(U \otimes P_2)z(t) - (1 - \mu)z^T(t - \tau(t))(U \otimes P_2)z(t - \tau(t)), \quad (30)$$

$$\begin{aligned}\dot{V}_3(t) &\leq \tau z^T(t)(U \otimes P_3)z(t) - (1 - \mu) \int_{t-\tau}^t (z^T(s))(U \otimes P_3)z(s)ds, \\ &\leq \tau z^T(t)(U \otimes P_3)z(t) - \frac{1 - \mu}{\tau} \left(\int_{t-\tau(t)}^t z(s)ds \right)^T (U \otimes P_3) \left(\int_{t-\tau(t)}^t z(s)ds \right).\end{aligned}\quad (31)$$

From reference [33] and Assumption 2, for any diagonal matrix J_1, J_2 , we have

$$\begin{aligned}0 \leq & \begin{bmatrix} z_i(t) - z_j(t) \\ f(z_i(t)) - f(z_j(t)) \end{bmatrix}^T \begin{bmatrix} -J_1 \Delta_1 & J_1 \Delta_2 \\ * & -J_1 \end{bmatrix} \begin{bmatrix} z_i(t) - z_j(t) \\ f(z_i(t)) - f(z_j(t)) \end{bmatrix}, \\ & + \begin{bmatrix} z_i(t - \tau(t)) - z_j(t - \tau(t)) \\ f(z_i(t - \tau(t))) - f(z_j(t - \tau(t))) \end{bmatrix}^T \begin{bmatrix} -J_2 \Delta_1 & J_2 \Delta_2 \\ * & -J_2 \end{bmatrix}, \\ & \times \begin{bmatrix} z_i(t - \tau(t)) - z_j(t - \tau(t)) \\ f(z_i(t - \tau(t))) - f(z_j(t - \tau(t))) \end{bmatrix}.\end{aligned}\quad (32)$$

According to (29)–(32), we obtain

$$\dot{V}(t) \leq \sum_{l=1}^r \mu_l(\theta(t)) \sum_{i=1}^{N-1} \sum_{j=i+1}^N (\xi_{ij}^T(t) \Theta_{ij}^l \xi_{ij}(t)), \quad (33)$$

in which $\xi_{ij}(t) = ((z_i(t) - z_j(t))^T, (z_i(t - \tau(t)) - z_j(t - \tau(t)))^T, (f(z_i(t)) - f(z_j(t)))^T, (f(z_i(t - \tau(t))) - f(z_j(t - \tau(t))))^T, (\int_{t-\tau(t)}^t z_i(s) - z_j(s)ds)^T)^T$, and Θ_{ij}^l is the same in Theorem 1. From Definition 1, system (25) is synchronized when $\Theta_{ij}^l < 0$.

Note that, in Theorem 1, we did not introduce free-weighting matrix. Next, we will choose other Lyapunov–Krasovskii functional and introduce more free-weighting matrices, which can add more useful conditions.

Theorem 2. For $l = 1, 2, \dots, r$, system 25 is synchronized if there exist $P_{qq} > 0, Q_{qq} > 0, W > 0$, ($q = 1, 2, 3$), P_{sq}, Q_{sq}, T , ($1 \leq s < q \leq 3$), and $J_1 > 0, J_2 > 0$; then, the following formulas are holding for all $1 \leq i < j \leq N$:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ * & P_{22} & P_{23} \\ * & * & P_{33} \end{bmatrix} > 0, \quad (34)$$

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ * & Q_{22} & Q_{23} \\ * & * & Q_{33} \end{bmatrix} > 0,$$

$$\Omega_{ij}^I = \begin{bmatrix} \Xi_{11} & \Xi_{12} & \frac{1-\mu}{\tau}Q_{33} & \Xi_{14} & \frac{1-\mu}{\tau}Q_{13}^T & \frac{1-\mu}{\tau}Q_{23}^T & \Xi_{17} & 0 \\ * & \Xi_{22} & 0 & 0 & 0 & \frac{1-\mu}{\tau}Q_{23}^T & \Xi_{27} & 0 \\ * & * & \Xi_{33} & \Xi_{34} & \frac{1-\mu}{\tau}Q_{13}^T & 0 & -NG_{ij}^{(2)}D_{2l}^T T^T & -(1-\mu)P_{13} \\ * & * & * & \Xi_{44} & 0 & 0 & B_l T^T & -(1-\mu)P_{23} \\ * & * & * & * & \frac{1-\mu}{\tau}Q_{11} & -\frac{1-\mu}{\tau}Q_{12} & 0 & 0 \\ * & * & * & * & * & -\frac{1-\mu}{\tau}Q_{22} & 0 & 0 \\ * & * & * & * & * & * & \Xi_{77} & 0 \\ * & * & * & * & * & * & * & -(1-\mu)P_{33} \end{bmatrix} < 0, \quad (35)$$

where

$$\Xi_{11} = -WC_l - C_l^T W^T + P_{11} + \tau Q_{11} - \frac{1-\mu}{\tau}Q_{33} - J_1 \Delta_1,$$

$$\Xi_{12} = P_{12} + \tau Q_{12} + WA_l - NG_{ij}^{(1)}WD_{1l} + J_1 \Delta_2,$$

$$\Xi_{14} = WB_l - NG_{ij}^{(2)}WD_{2l},$$

$$\Xi_{17} = P_{13} + \tau Q_{13} - C_l T^T - NG_{ij}^{(1)}D_{1l}^T T^T,$$

$$\Xi_{22} = P_{22} + \tau Q_{22} - J_1,$$

$$\Xi_{27} = P_{23} + \tau Q_{23} + A_l^T T^T,$$

$$\Xi_{33} = -(1-\mu)P_{11} - \frac{1-\mu}{\tau}Q_{33} - J_2 \Delta_1,$$

$$\Xi_{34} = -(1-\mu)P_{12} + J_2 \Delta_2,$$

$$\Xi_{44} = -(1-\mu)P_{22} - J_2,$$

$$\Xi_{77} = P_{33} + \tau Q_{33} - T - T^T.$$

(36)

Proof. From Assumptions 1 and 2, consider the following LKF for model (17):

$$V(t) = V_1(t) + V_2(t) + V_3(t), \quad (37)$$

where

$$V_1(t) = z^T(t)(U \otimes W)z(t),$$

$$V_2(t) = \int_{t-\tau(t)}^t \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix} \begin{bmatrix} U \otimes P_{11} & U \otimes P_{12} & U \otimes P_{13} \\ * & U \otimes P_{22} & U \otimes P_{23} \\ * & * & U \otimes P_{33} \end{bmatrix}$$

$$\cdot \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix} ds,$$

$$V_3(t) = \int_{t-\tau(t)}^t \int_{\theta}^t \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix} \begin{bmatrix} U \otimes Q_{11} & U \otimes Q_{12} & U \otimes Q_{13} \\ * & U \otimes Q_{22} & U \otimes Q_{23} \\ * & * & U \otimes Q_{33} \end{bmatrix}$$

$$\cdot \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix} ds d\theta.$$

(38)

Calculating the time derivative of system 29, then

$$\begin{aligned}
\dot{V}_1(t) &= 2z^T(t)(U \otimes W)\dot{z}(t), \\
&= 2z^T(t)(U \otimes W) \sum_{l=1}^r \mu_l(\theta(t)) [-(I_N \otimes C_l)z(t) + (I_N \otimes A_l)F(z(t)), \\
&\quad + (I_N \otimes B_l)F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l})F(z(t)), \\
&\quad + (G^{(2)} \otimes D_{2l})F(z(t - \tau(t)))], \\
\dot{V}_2(t) &= \begin{bmatrix} z(t) \\ F(z(t)) \\ \dot{z}(t) \end{bmatrix}^T \begin{bmatrix} U \otimes P_{11} & U \otimes P_{12} & U \otimes P_{13} \\ * & U \otimes P_{22} & U \otimes P_{23} \\ * & * & U \otimes P_{33} \end{bmatrix} \begin{bmatrix} z(t) \\ F(z(t)) \\ \dot{z}(t) \end{bmatrix} - (1 - \dot{\tau}(t)), \\
&\quad \times \begin{bmatrix} z(t - \tau(t)) \\ F(z(t - \tau(t))) \\ \dot{z}(t - \tau(t)) \end{bmatrix}^T \begin{bmatrix} U \otimes P_{11} & U \otimes P_{12} & U \otimes P_{13} \\ * & U \otimes P_{22} & U \otimes P_{23} \\ * & * & U \otimes P_{33} \end{bmatrix} \begin{bmatrix} z(t - \tau(t)) \\ F(z(t - \tau(t))) \\ \dot{z}(t - \tau(t)) \end{bmatrix}.
\end{aligned} \tag{39}$$

From Lemmas 2 and 3, we can acquire

$$\begin{aligned}
\dot{V}_3(t) &= \tau(t) \begin{bmatrix} z(t) \\ F(z(t)) \\ \dot{z}(t) \end{bmatrix}^T \begin{bmatrix} U \otimes Q_{11} & U \otimes Q_{12} & U \otimes Q_{13} \\ * & U \otimes Q_{22} & U \otimes Q_{23} \\ * & * & U \otimes Q_{33} \end{bmatrix} \begin{bmatrix} z(t) \\ F(z(t)) \\ \dot{z}(t) \end{bmatrix}, \\
&\quad - (1 - \dot{\tau}(t)) \int_{t-\tau(t)}^t \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix}^T \begin{bmatrix} U \otimes Q_{11} & U \otimes Q_{12} & U \otimes Q_{13} \\ * & U \otimes Q_{22} & U \otimes Q_{23} \\ * & * & U \otimes Q_{33} \end{bmatrix} \begin{bmatrix} z(s) \\ F(z(s)) \\ \dot{z}(s) \end{bmatrix} ds, \\
&\leq \tau \begin{bmatrix} z(t) \\ F(z(t)) \\ \dot{z}(t) \end{bmatrix}^T \begin{bmatrix} U \otimes Q_{11} & U \otimes Q_{12} & U \otimes Q_{13} \\ * & U \otimes Q_{22} & U \otimes Q_{23} \\ * & * & U \otimes Q_{33} \end{bmatrix} \begin{bmatrix} z(t) \\ F(z(t)) \\ \dot{z}(t) \end{bmatrix}, \\
&\quad - \frac{1 - \mu}{\tau} \psi^T(t) \begin{bmatrix} U \otimes Q_{11} & U \otimes Q_{12} & U \otimes Q_{13} & -U \otimes Q_{13} \\ * & U \otimes Q_{22} & U \otimes Q_{23} & -U \otimes Q_{23} \\ * & * & U \otimes Q_{33} & -U \otimes Q_{33} \\ * & * & * & U \otimes Q_{33} \end{bmatrix} \psi(t),
\end{aligned} \tag{40}$$

where $\psi^T(t) = \left[\left(\int_{(t-\tau t)}^t z(s) ds \right)^T, \left(\int_{(t-\tau t)}^t F(z(s)) ds \right)^T, z^T(t), z^T(t - \tau(t)) \right]$. Note that if X is a matrix with zero column sums, then $UX = NX$; from Lemma 1, we have

$$\begin{aligned} \dot{V}_1(t) = & 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N (z_i(t) - z_j(t))^T \sum_{l=1}^r \mu_l(\theta(t)) [(-WC_l)(z_i(t) - z_j(t)), \\ & + (WA_l - NG_{ij}^{(1)}WD_{1l})(f(z_i(t)) - f(z_j(t))), \\ & + (WB_l - NG_{ij}^{(2)}WD_{2l})(f(z_i(t - \tau_1(t))) - f(z_j(t - \tau_1(t))))], \end{aligned} \quad (41)$$

$$\begin{aligned} \dot{V}_2(t) \leq & \sum_{i=1}^{N-1} \sum_{j=i+1}^N \begin{bmatrix} z_i(t) - z_j(t) \\ f(z_i(t)) - f(z_j(t)) \\ \dot{z}_i(t) - \dot{z}_j(t) \end{bmatrix}^T \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ * & P_{22} & P_{23} \\ * & * & P_{33} \end{bmatrix} \begin{bmatrix} z_i(t) - z_j(t) \\ f(z_i(t)) - f(z_j(t)) \\ \dot{z}_i(t) - \dot{z}_j(t) \end{bmatrix}, \\ & - (1 - \mu) \sum_{i=1}^{N-1} \sum_{j=i+1}^N \begin{bmatrix} z_i(t - \tau(t)) - z_j(t - \tau(t)) \\ f(z_i(t - \tau(t))) - f(z_j(t - \tau(t))) \\ \dot{z}_i(t - \tau(t)) - \dot{z}_j(t - \tau(t)) \end{bmatrix}^T \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ * & P_{22} & P_{23} \\ * & * & P_{33} \end{bmatrix}, \\ & \times \begin{bmatrix} z_i(t - \tau(t)) - z_j(t - \tau(t)) \\ f(z_i(t - \tau(t))) - f(z_j(t - \tau(t))) \\ \dot{z}_i(t - \tau(t)) - \dot{z}_j(t - \tau(t)) \end{bmatrix}. \end{aligned} \quad (42)$$

As the same method, from Lemma 1, we have

$$\begin{aligned} \dot{V}_3(t) \leq & \tau \sum_{i=1}^{N-1} \sum_{j=i+1}^N \begin{bmatrix} z_i(t) - z_j(t) \\ f(z_i(t)) - f(z_j(t)) \\ \dot{z}_i(t) - \dot{z}_j(t) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ * & Q_{22} & Q_{23} \\ * & * & Q_{33} \end{bmatrix} \begin{bmatrix} z_i(t) - z_j(t) \\ f(z_i(t)) - f(z_j(t)) \\ \dot{z}_i(t) - \dot{z}_j(t) \end{bmatrix}, \\ & - \frac{1 - \mu}{\tau} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \begin{bmatrix} \int_{t-\tau(t)}^t (z_i(s) - z_j(s)) ds \\ \int_{t-\tau(t)}^t (f(z_i(s)) - f(z_j(s))) ds \\ z_i(t) - z_j(t) \\ z_i(t - \tau(t)) - z_j(t - \tau(t)) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & -Q_{13} \\ * & Q_{22} & Q_{23} & -Q_{23} \\ * & * & Q_{33} & -Q_{33} \\ * & * & * & Q_{33} \end{bmatrix}, \\ & \times \begin{bmatrix} \int_{t-\tau(t)}^t (z_i(s) - z_j(s)) ds \\ \int_{t-\tau(t)}^t (f(z_i(s)) - f(z_j(s))) ds \\ z_i(t) - z_j(t) \\ z_i(t - \tau(t)) - z_j(t - \tau(t)) \end{bmatrix}. \end{aligned} \quad (43)$$

For any matrix $T \in R^{n \times n}$, from system (17), we can easily obtain

$$\begin{aligned} 0 = & 2\dot{z}^T(t)(U \otimes T) \sum_{l=1}^r \mu_l(\theta(t))(-\dot{z}(t) - (I_N \otimes C_l)z(t) \\ & + (I_N \otimes A_l)F(z(t)), \\ & + (I_N \otimes B_l)F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l})z(t) \\ & + (G^{(2)} \otimes D_{2l})z(t - \tau(t)). \end{aligned} \quad (44)$$

Let $\zeta_{ij}^T(t) = ((z_i(t) - z_j(t))^T, f((z(t) - f(z_j(t))))^T, (z_i(t - \tau(t)) - z_j(t - \tau(t)))^T f((z_i(t - \tau(t))) - f(z_i(t - \tau(t))))^T, (\int_{t-\tau(t)}^t (z_i(s) - z_j(s))ds)^T (\int_{t-\tau(t)}^t (f(z_i(s)) - f(z_j(s)))ds)^T$,

$(s))ds)^T$, $(\dot{z}_i(t) - \dot{z}_j(t))^T, (\dot{z}_i(t - \tau(t)) - \dot{z}_j(t - \tau(t)))^T$, from (32) and (41)–(44), we can obtain

$$\dot{V}(t) \leq \sum_{l=1}^r \mu_l(\theta(t)) \sum_{i=1}^{N-1} \sum_{j=i+1}^N (\zeta_{ij}^T(t) \Omega_{ij}^l \zeta_{ij}(t)), \quad (45)$$

where Ω_{ij}^l is defined as (50). From Definition 1, it implies that system (17) is synchronized.

3.2. Synchronization for Fuzzy System with Pinning Control. This section deals with the pinning synchronization problems for the closed-loop T-S fuzzy neural networks:

$$\begin{aligned} \dot{z}(t) = & \sum_{l=1}^r \sum_{k=1}^r \mu_l(\theta(t)) \mu_k(\theta(t)) (-I_N \otimes C_l)z(t) + (I_N \otimes A_l)F(z(t)), \\ & + (I_N \otimes B_l)F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l})z(t) + (G^{(2)} \otimes D_{2l})z(t - \tau(t)), \\ & + (L^{(1)} \otimes D_{3k})z(t) + (L^{(2)} \otimes D_{4k})z(t - \tau(t)). \end{aligned} \quad (46)$$

Theorem 3. For $l = 1, 2, \dots, r$, dynamical system (46) is synchronized if there is $P_z > 0$ ($z = 2, 3, 4$) and positive diagonal matrix P_1, J_1 , and J_2 ; then, the following formulas are holding for all $1 \leq i < j \leq N$:

$$\Theta_{ij}^{lk} = \begin{bmatrix} \Pi_{11} & \Pi_{12} & A_l X + X J_1 X \Delta_2 & B_l X & 0 \\ * & \Pi_{22} & 0 & X J_2 X \Delta_2 & 0 \\ * & * & -X J_1 X & 0 & 0 \\ * & * & * & -X J_2 X & 0 \\ * & * & * & * & \frac{1-\mu}{\tau} X P_3 X \end{bmatrix} < 0, \quad (47)$$

in which

$$\begin{aligned} \Pi_{11} = & -C_l X - X C_l - X J_1 X \Delta_1 - N G_{ij}^{(1)} D_{1l} X - N L_{ij}^{(1)} z_{3k} \\ & - N G_{ij}^{(1)} X D_{1l}^T - N L_{ij}^{(1)} z_{3k}^T + X P_2 X + \tau X P_3 X, \\ \Pi_{12} = & -N G_{ij}^{(2)} D_{2l} X - N L_{ij}^{(2)} z_{4k}, \\ \Pi_{22} = & -(1 - \mu) X P_2 X - X J_2 X \Delta_1. \end{aligned} \quad (48)$$

Proof. Based on Theorem 1, the feedback gains in the fuzzy coupled system are given by $D_{3k} = z_{3k} X^{-1}$ and $D_{4k} = z_{4k} X^{-1}$. Replace $N G_{ij}^{(1)} D_{3l}$ with $N G_{ij}^{(1)} D_{1l} + N L_{ij}^{(1)} D_{3k}$, $N G_{ij}^{(2)} D_{2l}$ with $N G_{ij}^{(2)} D_{4l} + N L_{ij}^{(2)} D_{4k}$. Pre- and postmultiply 13 with $\text{diag}[X; X; X; X; X]$, where $X^{-1} = P_1$; then, we can obtain the above criteria.

Theorem 4. For $l = 1, 2, \dots, r$, system (46) is synchronized if there exists $P_{qq} > 0, Q_{qq} > 0$, ($q = 1, 2, 3$), P_{sq}, Q_{sq} , ($1 \leq s < q \leq 3$), and $J_1 > 0, J_2 > 0, W > 0$; then, the following formulas are holding for all $1 \leq i < j \leq N$:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ * & P_{22} & P_{23} \\ * & * & P_{33} \end{bmatrix} > 0, \quad (49)$$

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ * & Q_{22} & Q_{23} \\ * & * & Q_{33} \end{bmatrix} > 0,$$

$$\Omega_{ij}^{lk} = \begin{bmatrix} \Xi_{11} & \Xi_{12} & \frac{1-\mu}{\tau} XQ_{33}X & \Xi_{14} & -\frac{1-\mu}{\tau} XQ_{13}^T X & \frac{1-\mu}{\tau} XQ_{23}^T X & \Xi_{17} & 0 \\ * & \Xi_{22} & 0 & 0 & 0 & \frac{1-\mu}{\tau} XQ_{23}^T X & \Xi_{27} & 0 \\ * & * & \Xi_{33} & \Xi_{34} & \frac{1-\mu}{\tau} XQ_{13}^T X & 0 & \Xi_{37} & -(1-\mu)XP_{13}X \\ * & * & * & \Xi_{44} & 0 & 0 & XB_l & -(1-\mu)XP_{23}X \\ * & * & * & * & -\frac{1-\mu}{\tau} XQ_{11}X & -\frac{1-\mu}{\tau} XQ_{12}X & 0 & 0 \\ * & * & * & * & * & -\frac{1-\mu}{\tau} XQ_{22}X & 0 & 0 \\ * & * & * & * & * & * & \Xi_{77} & 0 \\ * & * & * & * & * & * & * & -(1-\mu)XP_{33}X \end{bmatrix} < 0, \quad (50)$$

where

$$\begin{aligned} \Xi_{11} &= -C_l X - XC_l^T + XP_{11}X + \tau XQ_{11}X \\ &\quad - \frac{1-\mu}{\tau} XQ_{33}X - XJ_1X\Delta_1, \\ \Xi_{12} &= XP_{12}X + \tau XQ_{12}X + A_l X - NG_{ij}^{(1)}D_{1l}X - NL_{ij}^{(1)}z_{3k} \\ &\quad + XJ_1X\Delta_2, \\ \Xi_{14} &= B_l X - NG_{ij}^{(2)}D_{2l}X - NL_{ij}^{(2)}z_{4k}, \\ \Xi_{17} &= XP_{13}X + \tau XQ_{13}X - XC_l - NG_{ij}^{(1)}XD_{1l}^T - NL_{ij}^{(1)}z_{3k}^T, \\ \Xi_{22} &= XP_{22}X + \tau XQ_{22}X - XJ_1X, \\ \Xi_{27} &= XP_{23}X + \tau XQ_{23}X + XA_l^T, \\ \Xi_{33} &= -(1-\mu)XP_{11}X - \frac{1-\mu}{\tau} XQ_{33}X - XJ_2X\Delta_1, \\ \Xi_{34} &= -(1-\mu)XP_{12}X + XJ_2X\Delta_2, \\ \Xi_{37} &= -NG_{ij}^{(1)}XD_{2l}^T - NL_{ij}^{(2)}z_{4k}^T, \\ \Xi_{44} &= -(1-\mu)XP_{22}X - XJ_2X, \\ \Xi_{77} &= XP_{33}X + \tau XQ_{33}X - X - X. \end{aligned} \quad (51)$$

Proof. Based on Theorem 2, let $T = W$, and the feedback gains in the fuzzy system are given by $D_{3k} = z_{3k}X^{-1}$ and $D_{4k} = z_{4k}X^{-1}$. Replace $NG_{ij}^{(1)}D_{1l}$ with $NG_{ij}^{(1)}D_{1l} + NL_{ij}^{(1)}D_{3k}$, $NG_{ij}^{(2)}D_{2l}$ with $NG_{ij}^{(2)}D_{2l} + NL_{ij}^{(2)}D_{4k}$. Pre- and postmultiply (35) with $\text{diag}[X; X; X; X; X; X; X; X]$, where $X^{-1} = W$; then, we can obtain the above criteria.

Remark 3. From these two pinning synchronized results, it is noted that the fuzzy pinning control gain matrices D_3 and D_4 can be computed, which can fix the communication connection graph topology $k_{ij}^{(1)}$ and $k_{ij}^{(2)}$. Such complex fuzzy pinning controllers are proposed for the first time.

4. Numerical Examples

This section provides a numerical example to illustrate the effectiveness of the obtained results. Assume the system without control first and then with pinning control.

4.1. Synchronization for Fuzzy Coupled Networks without Control. Consider the following TNNs model, the parameters in which are defined as

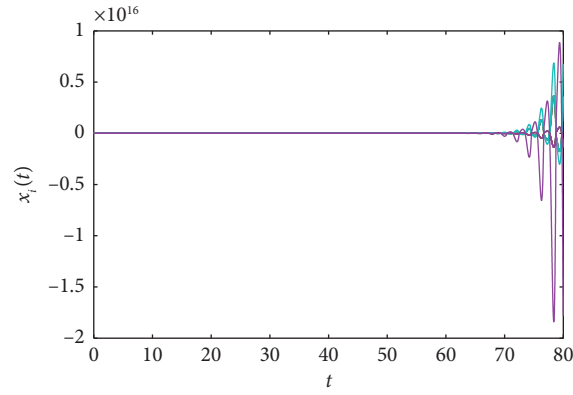


FIGURE 1: States of fuzzy neural networks (8) without pinning control ($L^{(1)} = L^{(2)} = 0$): $z_i(t), i = 1, 2, 3, 4, 5, 6$.

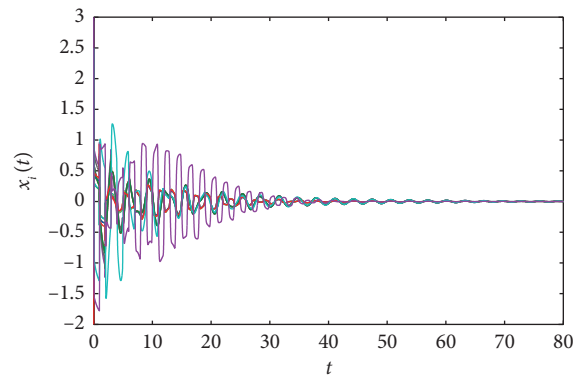


FIGURE 2: States of fuzzy neural networks (8) with pinning control: $z_i(t), i = 1, 2, 3, 4, 5, 6$.

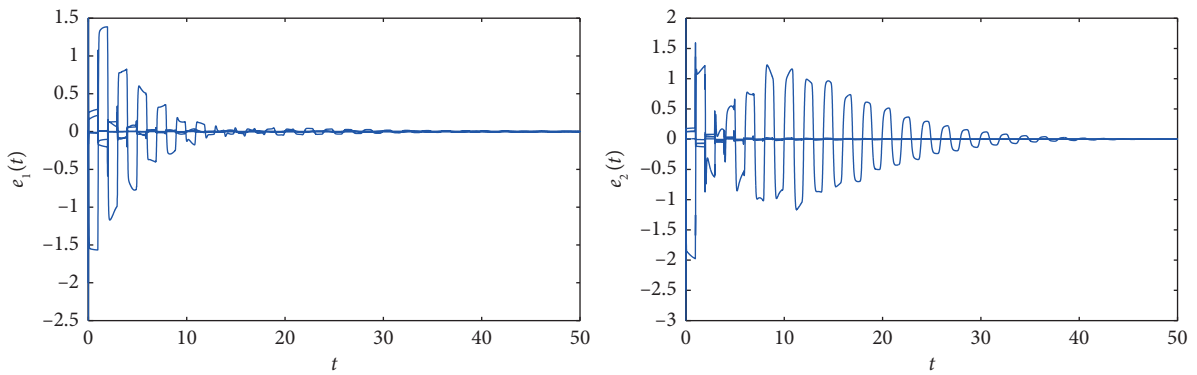


FIGURE 3: Pinning synchronization errors for fuzzy neural networks: $e_j(t), j = 1, 2$.

$$\begin{aligned} \dot{z}(t) = & \sum_{l=1}^2 \mu_l(\theta(t)) (-I_N \otimes C_l) z(t) + (I_N \otimes A_l) F(z(t)) \\ & + (I_N \otimes B_l) F(z(t - \tau(t))), \\ & + (G^{(1)} \otimes D_{1l}) z(t) + (G^{(2)} \otimes D_{2l}) z(t - \tau(t)), \end{aligned} \quad (52)$$

where

$$\begin{aligned} C_1 = & \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}, C_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, A_1 = A_2 = \begin{bmatrix} 1.8 & 10 \\ 0.1 & 1.8 \end{bmatrix}, \\ B_1 = B_2 = & \begin{bmatrix} -1.5 & 0.1 \\ 0.1 & -1.5 \end{bmatrix}, \\ D_{11} = & \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, D_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1.5 \end{bmatrix}, D_{21} = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}, D_{22} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, \\ \mu = 0, \tau = 0.97, \mu_1 = & \frac{1}{1 + \exp(-2z_1(t))}, \mu_2 = 1 - \mu_1. \end{aligned} \quad (53)$$

The outer-coupling matrix $(G_{ij}^{(q)})_{N \times N}$, $(q = 1, 2)$, are defined as

$$G^{(1)} = G^{(2)} = \begin{bmatrix} -5 & 1 & 0 & 1 & 0 & 3 \\ 1 & -5 & 1 & 0 & 0 & 3 \\ 0 & 1 & -5 & 0 & 1 & 3 \\ 1 & 0 & 0 & -5 & 1 & 3 \\ 0 & 0 & 1 & 1 & -5 & 3 \\ 3 & 3 & 3 & 3 & 3 & -15 \end{bmatrix}. \quad (54)$$

Then, we plot the states of network (52) without control in Figure 1. It is easy to see that the system cannot be synchronized by itself.

4.2. Pinning Synchronization for Fuzzy Neural Networks with Hybrid Coupling. Now, we consider the system with pinning control:

$$\begin{aligned} \dot{z}(t) = & \sum_{l=1}^r \sum_{k=1}^r \mu_l(\theta(t)) \mu_k(\theta(t)) (-I_N \otimes C_l) z(t) \\ & + (I_N \otimes A_l) F(z(t)), \\ & + (I_N \otimes B_l) F(z(t - \tau(t))) + (G^{(1)} \otimes D_{1l}) z(t) \\ & + (G^{(2)} \otimes D_{2l}) z(t - \tau(t)), \\ & + (L^{(1)} \otimes D_{3k}) z(t) + (L^{(2)} \otimes D_{4k}) z(t - \tau(t)). \end{aligned} \quad (55)$$

The controllers' parameters are as follows:

$$L^{(1)} = L^{(2)} = \begin{bmatrix} -10 & 2 & 3 & 2 & 3 & 0 \\ 2 & -10 & 2 & 3 & 3 & 0 \\ 3 & 2 & -10 & 3 & 2 & 0 \\ 2 & 3 & 2 & -10 & 2 & 0 \\ 3 & 2 & 3 & 2 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (56)$$

other parameters are the same in system (52).

According to Theorem 4, system (55) can achieve synchronization by pinning control. Solving the LMIs in Theorem 4, we can obtain the fuzzy pinning control gain matrices as follows:

$$\begin{aligned} D_{31} = & \begin{bmatrix} 6.7338 & 0 \\ 0 & 9.7243 \end{bmatrix}, \\ D_{32} = & \begin{bmatrix} 6.7872 & 0 \\ 0 & 9.6380 \end{bmatrix}, \\ D_{41} = & \begin{bmatrix} 4.2085 & 0 \\ 0 & 5.8983 \end{bmatrix}, \\ D_{42} = & \begin{bmatrix} 4.1814 & 0 \\ 0 & 5.9448 \end{bmatrix}. \end{aligned} \quad (57)$$

From the examples, network (55) without control is shown in Figure 1, and the system with pinning control is shown in Figure 2. It is easy to see that the results are very good by our methods. We also show the synchronization errors in Figure 3, where $e_j(t) = (z_{ij}(t) - z_{1j}(t))$, $i = 2, 3, 4, 5, 6$; $j = 1, 2$.

5. Conclusion

This paper has investigated the synchronization of T-S FNNs by proposing a novel pinning control scheme. Instead of the presenting technique in prior, the proposed scheme regulates the column controllers by some events which are yielded by the relationships among the LKF and three nonnegative regions. Therefore, the traditional controllers have been improved as the event-dependent one in this paper. A concise criterion has been presented to guarantee the pinning synchronization of the considered CNNs. Simulations are finally provided to display the feasibility and improvements of the proposed pinning control scheme. Our results can only be studied as theoretical research now. We expect that the innovations of this paper can shed further light on the more problems (such as ([34–36]) under column controllers law. By the similar mechanism, our further directions include (1) design an intermittent output feedback controller and (2) design an intermittent adaptive controller.

Data Availability

The data used to support the findings the 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 paper.

Acknowledgments

This work was supported by the National Defense Pre-Research Foundation of China (1126170104A, 1126180204B, and 1126190508A).

References

- [1] R. Samli, "A new delay-independent condition for global robust stability of neural networks with time delays," *Neural Networks*, vol. 66, pp. 131–137, 2015.
- [2] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
- [3] G. Peng, C. Yang, W. He, and C. L. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3138–3148, 2020.
- [4] C. Yang, G. Peng, L. Cheng, Na Jing, and Z. Li, "Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, in press, 2019.
- [5] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.
- [6] X. F. Wang and G. Chen, "Synchronization in small-world dynamical networks," *International Journal of Bifurcation and Chaos*, vol. 12, no. 1, pp. 187–192, 2002.
- [7] H. Li, X. Liao, G. Chen, D. J. Hill, Z. Dong, and T. Huang, "Event-triggered asynchronous intermittent communication strategy for synchronization in complex dynamical networks," *Neural Networks*, vol. 66, pp. 1–10, 2015.
- [8] J. Qin, H. Gao, and W. X. Zheng, "Exponential synchronization of complex networks of linear systems and nonlinear oscillators: a unified analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 510–521, 2015.
- [9] X. Chen, T. Huang, J. Cao, J. H. Park, J. Qiu, and J. Qiu, "Finite-time multi-switching sliding mode synchronisation for multiple uncertain complex chaotic systems with network transmission mode," *IET Control Theory & Applications*, vol. 13, no. 9, pp. 1246–1257, 2019.
- [10] F. Qu, J. Liu, H. Zhu, and B. Zhou, "Wind turbine fault detection based on expanded linguistic terms and rules using non-singleton fuzzy logic," *Applied Energy*, vol. 262, Article ID 114469, 2020.
- [11] F. Qu, J. Liu, H. Zhu, and D. Zang, "Wind turbine condition monitoring based on assembled multidimensional membership functions using fuzzy inference system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4028–4037, 2020.
- [12] X. Yu, W. He, H. Li, and J. Sun, "Adaptive fuzzy full-state and output feedback control for uncertain robots with output constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, in press, 2020.
- [13] H. Lin, T. Zhang, Z. Chen, H. Song, and C. Yang, "Adaptive fuzzy Gaussian mixture models for shape approximation in robot grasping," *International Journal of Fuzzy Systems*, vol. 21, no. 4, pp. 1026–1037, 2019.
- [14] J. Jian and W. Jiang, "Lagrange exponential stability for fuzzy Cohen-Grossberg neural networks with time-varying delays," *Fuzzy Sets and Systems*, vol. 277, no. 15, pp. 65–80, 2015.
- [15] X. Yang and Z. Yang, "Synchronization of TS fuzzy complex dynamical networks with time-varying impulsive delays and stochastic effects," *Fuzzy Sets and Systems*, vol. 235, no. 16, pp. 25–43, 2014.
- [16] Y. Liu, B.-Z. Guo, J. H. Park, S.-M. Lee, and S.-M. Lee, "Nonfragile exponential synchronization of delayed complex dynamical networks with memory sampled-data control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 118–128, 2018.
- [17] Z.-M. Zhang, Y. He, M. Wu, and Q.-G. Wang, "Exponential synchronization of neural networks with time-varying delays via dynamic intermittent output feedback control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 3, pp. 612–622, 2019.
- [18] S. Ding and Z. Wang, "Lag quasi-synchronization for memristive neural networks with switching jumps mismatch," *Neural Computing and Applications*, vol. 28, no. 12, pp. 4011–4022, 2017.
- [19] Y. Liang and X. Wang, "Synchronization in complex networks with non-delay and delay couplings via intermittent control with two switched periods," *Physica A: Statistical Mechanics and its Applications*, vol. 395, pp. 434–444, 2014.
- [20] T. Chen, X. Liu, and W. Lu, "Pinning complex networks by a single controller," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1317–1326, 2007.
- [21] X. Tianping Chen and T. Chen, "Cluster synchronization in directed networks via intermittent pinning control," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1009–1020, 2011.
- [22] J. Zhou, J.-A. Lu, and J. Lü, "Pinning adaptive synchronization of a general complex dynamical network," *Automatica*, vol. 44, no. 4, pp. 996–1003, 2008.
- [23] W. Yu, P. DeLellis, G. Chen, M. Di Bernardo, and J. Kurths, "Distributed adaptive control of synchronization in complex networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2153–2158, 2012.
- [24] Y. Tang, H. Gao, J. Lu, and J. Kurths, "Pinning distributed synchronization of stochastic dynamical networks: a mixed optimization approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1804–1815, 2014.
- [25] Q. Chen and J. Cao, "Synchronization of complex dynamical networks with discrete time delays on time scales," *Neurocomputing*, vol. 151, pp. 729–736, 2015.
- [26] J.-L. Wang, H.-N. Wu, and L. Guo, "Novel adaptive strategies for synchronization of linearly coupled neural networks with reaction-diffusion terms," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 429–440, 2014.
- [27] J.-L. Wang and H.-N. Wu, "Synchronization and adaptive control of an array of linearly coupled reaction-diffusion neural networks with hybrid coupling," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1350–1361, 2014.
- [28] Y. Liu, Z. Wang, and X. Liu, "Global exponential stability of generalized recurrent neural networks with discrete and distributed delays," *Neural Networks*, vol. 19, no. 5, pp. 667–675, 2006.
- [29] Y. Liu, Z. Wang, X. Liang, and X. Liu, "Stability and synchronization of discrete-time markovian jumping neural networks with mixed mode-dependent time delays," *IEEE*

- Transactions on Neural Networks*, vol. 20, no. 7, pp. 1102–1116, 2009.
- [30] Y. Liu, Z. Wang, J. Liang, and X. Liu, “Synchronization of coupled neutral-type neural networks with jumping-mode-dependent discrete and unbounded distributed delays,” *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 102–114, 2013.
 - [31] Y. Liu, Z. Wang, J. Liang, and X. Liu, “Synchronization and state estimation for discrete-time complex networks with distributed delays,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1314–1325, 2008.
 - [32] D. Gong, H. Zhang, Z. Wang, and B. Huang, “Novel synchronization analysis for complex networks with hybrid coupling by handling multitude kronecker product terms,” *Neurocomputing*, vol. 82, pp. 14–20, 2012.
 - [33] Z. Wang, Y. Wang, and Y. Liu, “Global synchronization for discrete-time stochastic complex networks with randomly occurred nonlinearities and mixed time delays,” *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 11–25, 2010.
 - [34] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, “Adaptive parameter estimation and control design for robot manipulators with finite-time convergence,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.
 - [35] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, “Mind control of a robotic arm with visual fusion technology,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3822–3830, 2018.
 - [36] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, “Admittance-based controller design for physical human-robot interaction in the constrained task space,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–13. in press, 2020.

Research Article

The Design of Compact Robotic-Assisted Needle Position System with MPC-Based Remote Control

Jing Guo,¹ Yi Liu,¹ Jin Wang,¹ Chao Zeng ,¹ Jie Huang ,² and Chao Liu³

¹School of Automation, Guangdong University of Technology, Guangzhou, China

²Guangdong Lung Cancer Institute, Guangdong Provincial Key Laboratory of Translational Medicine in Lung Cancer, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China

³Department of Robotics, LIRMM, Univ Montpellier, CNRS, Montpellier, France

Correspondence should be addressed to Chao Zeng; auczeng@gdut.edu.cn and Jie Huang; jieh2015@yahoo.com

Received 28 June 2020; Revised 7 August 2020; Accepted 30 August 2020; Published 16 September 2020

Academic Editor: Yan Wu

Copyright © 2020 Jing Guo 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 article introduces the design and control performance of a lightweight, flexible, 4-degree-of-freedom (DOF) parallel robot for percutaneous biopsy guided by computed tomography (CT). At present, the CT guidance method allows surgeons to quickly locate the lesion area; however, it is necessary to manually adjust the position of the puncture needle for insertion. In this paper, a three-dimensional assisted method is used to infer the control input required to reach the target point through the kinematic model of the robot. A Kalman filter is designed to estimate model parameters and obtain a more accurate model. To further improve the control performance of the robot system, a model-based control method—the model predictive control (MPC) controller—is used to increase the accuracy of the needle position in the developed robot system. In this way, medical efficiency is improved while reducing the burden on the surgeon.

1. Introduction

The surgeries have been advanced in the past decades thanks to novel surgical techniques [1], such as laparoscopy surgeries, in which a surgeon can operate a surgery through tiny holes on a human body for removing pathological tissue. However, the hand-held tools still make the operation hard to be performed due to limited ergonomics and accuracy. Meanwhile, the surgeons may feel fatigued when operating for long time during the surgery. In order to improve the quality of surgical operation, the robot assistant technique has been introduced into the operation room which has eased the burden of the surgeon during the surgery [2–4]. Nowadays, the robotic systems have been developed and used for various surgeries such as biopsy, brachytherapy, and tumor ablation [5–7].

Before performing a minimally invasive surgical (MIS) task, the coordinates of the robot are usually registered to the medical image coordinate system, so that the robot is controlled by analysis of the lesion's position for further

operations. According to the characteristics of MIS, the pose of the robot end-effector before needle insertion during surgery becomes very important. The needle positioning system of the surgical assistant robot can be independent of the insertion action [8–10]. The surgeon can guide the robot's needle to the insertion position of the lesion on the basis of the medical image through image guidance [11–16] and then manually insert the needle into the lesion area. Medical imaging, acting as a real-time feedback tool for the needle pose, is crucial in the process of needle positioning and control. In current needle-guided systems, CT [11–13, 17, 18] is often used. Today's commercially available minimally invasive robotic systems (MIRS) maintain and expand the flexibility of the surgeon's hand [19–22]. Navarro-Alarcon et al. [12] developed a 3-DOF needle driver for biopsy. The positioning and insertion of the needle are achieved through three interfaces, the first two of which are aligned to the target, and the third is inserted. Koseki et al. [23] established a cooperative manipulation structure which uses the optical linear encoder to measure the needle

position. However, manual remote operation control requires multiple scans using CT to obtain accurate targets. The robot-assisted positioning approach can locally record the workspace and medical images of the robot system to reduce the risk of cancer induced by repeated CT scans [24]. This approach poses challenges to the compatibility and size of the robot system.

In the development of a surgical robot, another major problem is the network system in the remote operating system, which is caused by long distance or wireless link [23–27]. Sampling and delay are involved in this problem, which is a challenge for remote controlling of the robot. The solution to the delay problem in remote is discussed in [28–31]. However, the local optimizer on the slave side of the robot will also cause a delay [32, 33]. This part of the delay will also affect the remote operating system following the feedback signal. Dong et al. [34] modified standard functions to improve the response speed. Norizuki and Uchimura [35] embedded the MPC controller into the remote operating system to reduce the impact of the optimizer’s delay in the system.

In order to improve the quality of the surgery, the accuracy of the needle pose becomes crucial. The needle pose is affected by the robot control system and medical images. However, factors such as friction in the design of the robot structure and noise in the feedback of medical images are inevitable, which makes it difficult to accurately control the needle positioning [36, 37]. Thus, the closed-loop control method becomes a good solution [38, 39]. In this paper, firstly, as shown in Figure 1, a novel light-weighted puncture robotic system is provided. A surgeon controls the medical robot through the master in the remote system. The medical robot can provide feedback of the end-effector pose and force to the surgeon. The size of the designed robot is 16 cm high, 37.6 cm long, and 37.6 cm wide. In addition, the robot is controlled by four motors, which leads to the coupling of robot control. Due to the characteristics of the robot structure, specific constraints are proposed to decouple the kinematics. Finally, we develop a related needle positioning system on this robotic platform, which uses an MPC strategy to achieve the robotic local closed-loop control of the needle tip position. This approach not only solves the problem of the time delay (TD) of the needle positioning system in local, but also reduces the cost and time of personnel training required for the operation of the robotic needle positioning system.

The structure of the paper is organized as follows: in Section 2, the overall robotic system design is introduced, including the derivation of constrained kinematic decoupling approach. Section 3 introduces model parameter estimation. Section 4 describes MPC control method. In Section 5, the experiments and the analysis of the experimental results are introduced. We conclude this article in Section 6.

In our previous work [40], we have designed a remote system which contains a master and a slave robot (see Figure 1). A surgeon can control the slave robot through the master side, and during a task, the desired position of the end-effector can be marked and recorded. The end-effector

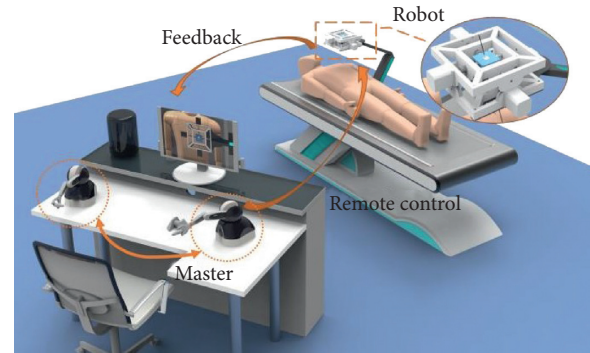


FIGURE 1: Surgical assistant robot remote operating system.

position of the slave robot is updated via medical imaging or other sensors, and such position feedback is sent to master. There are many ways to solve the remote communication scheme between the master and slave robots [28–31]. However, when the slave robot uses the optimized control method to improve the control effect, a delay will occur in the slave robot, which will affect the use of the remote operating system. Thus, the local controller design of the slave robot needs to be considered.

Because the robotic control system is a multi-input multioutput (MIMO) system, the controller selected for the slave robot should be able to adapt to the MIMO system. The proposed slave robot control system can be seen in Figure 2(a). The surgeon uses CT to mark the target lesion area or trajectory in the robot workspace. The surgical robot obtains the control trajectory which serves as the reference input for the MPC controller through kinematic model. The MPC controller optimizes the current control input to the surgical robot according to the reference trajectory and the reference state of the surgical robot. Furthermore, in order to improve the control performance, the state-space model of the robot required by the MPC controller should be accurate. In this paper, according to the structure of the surgical robot, the kinematics model is divided into a linear part and a nonlinear part. The linear part uses the model parameter estimator to identify the model parameters, and the nonlinear part is linearized using the Taylor formula. As shown in Figure 2(b), the model parameter estimator of the linear part is composed of an order judger and a parameter estimator. The order judger is used to judge the order of the linear system. The Kalman filter is selected as the parameter estimator to estimate the parameters of the linear system.

2. Kinematic Modeling

In this section, the modeling process of surgical robot kinematics will be described. The robotic kinematic is analyzed for the establishment of model parameter estimator and MPC controller. A suitable kinematics model is constructed by analyzing the mechanical structure of the parallel puncture needle robot [40]. Figure 3 shows the three-dimensional mechanical structure and sectional structure of the surgical robot. The blocks, i.e., block-*a* and block-*b*, can be moved by controlling the angle of four motors. When the

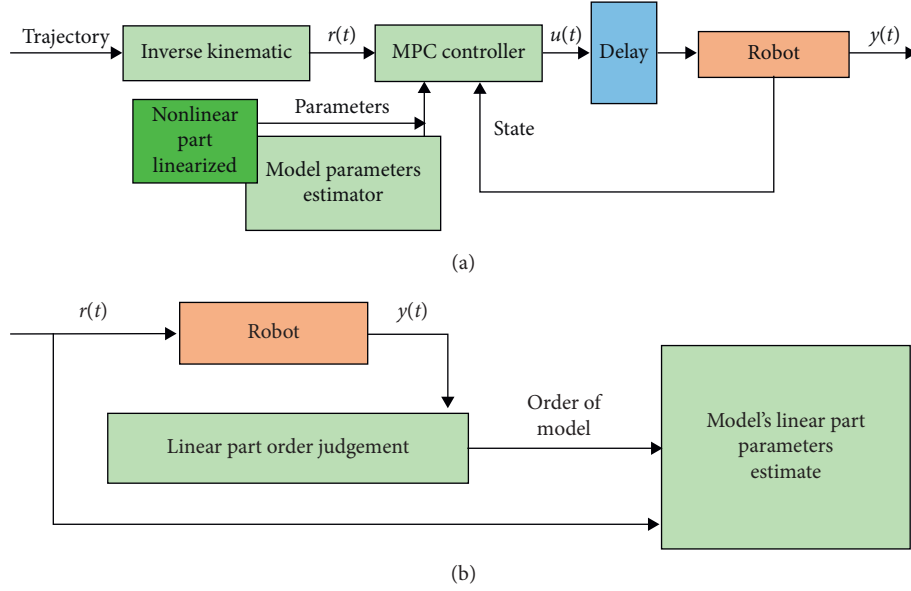


FIGURE 2: (a) The overall control framework of the surgical robot positioning system. (b) A schematic of the model parameter estimator of the linear part of the kinematics model.

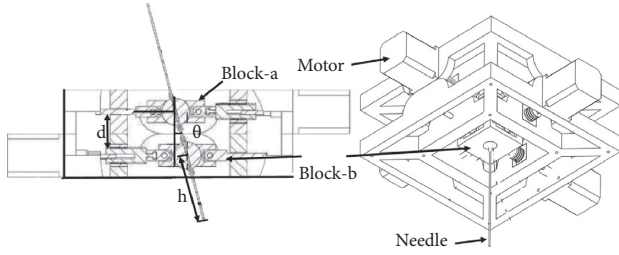


FIGURE 3: The three-dimensional mechanical structure and sectional structure of the surgical robot. It uses two parallel blocks to control the insertion direction and position of the same puncture needle.

positions of the two blocks change, the position and insertion direction of the puncture needle will change accordingly. Among this structure, the vertical distance d between the two blocks and the length h of the puncture needle below the block-b are both constant.

In order to obtain a suitable kinematics model, the coordinate relationship as shown in Figure 4 is established. The fixed world coordinate system is at the initial center position of block-b. wT_a is the coordinate system of block-a, which is used to describe the moving distance of block-a. Similarly, the value of the coordinate system wT_b is the moving distance of block-b. Coordinate system wT_e indicates the position of the tip of the puncture needle.

2.1. Forward Kinematic. The purpose of forward kinematics is to obtain the position of the tip of the puncture needle x_e , y_e , and z_e , and the forward kinematics of the surgical robot can be obtained using geometric theorem. As shown in Figure 4, firstly, the rotation of the motor will drive the block. The moving distance of the two blocks x_a , x_b , y_a , and y_b will change the angle of the puncture needle θ .

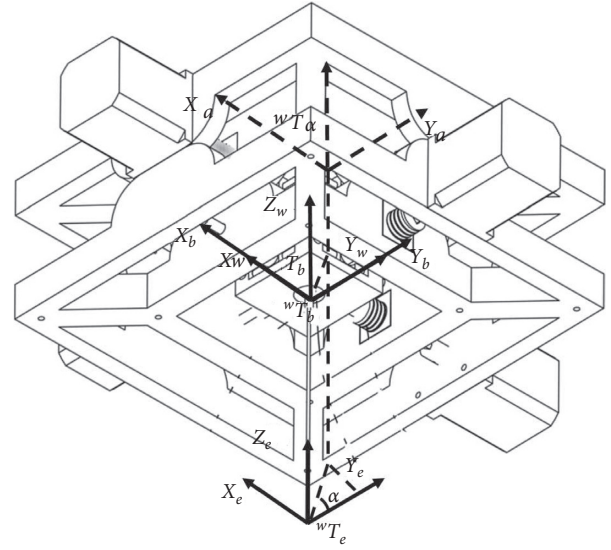


FIGURE 4: The surgical robot's coordinate relationship of the kinematic model. Define the world coordinate system and the coordinate system of each joint to describe the kinematics model.

$$x_a = \rho\beta_{a1}, \quad (1)$$

$$\theta = \arctan\left(\frac{\sqrt{(y_a - y_b)^2 + (x_a - x_b)^2}}{d}\right), \quad (2)$$

$$\alpha = \arctan\left(\frac{x_a - x_b}{y_a - y_b}\right), \quad (3)$$

where ρ is the pitch of the screw connected to the block; β is the rotation angle of the motor, and the relationship between the rotation angle of other motors and its block coordinates

is similar; α is the angle generated by two blocks on the X - Y plane; next, for coordinate system $\{{}^wT_b\}$, the change in the position of the tip of the puncture needle Δx , Δy , and Δz can be described by θ , α , and the constant length of the needle h :

$$\Delta x = h \cdot \sin(\theta) \cdot \sin(\alpha), \quad (4)$$

$$\Delta y = h \cdot \sin(\theta) \cdot \cos(\alpha), \quad (5)$$

$$\Delta z = h \cdot \cos(\theta). \quad (6)$$

Finally, the position of the puncture needle's tip x_e , y_e , and z_e relative to the fixed world coordinate system can be obtained as follows:

$$x_e = \Delta x + x_b, \quad (7)$$

$$y_e = \Delta y + y_b, \quad (8)$$

$$z_e = \Delta z. \quad (9)$$

2.2. Inverse Kinematic. The surgical robot is a parallel structure; hence, there is a coupling relationship between the control inputs. To address this problem, constraints are first defined to decouple the kinematic model. From the point of view of the positioning range of the puncture needle, it is better to have a larger range that the puncture needle can reach in a limited space. Thus, the first constraint is that the two blocks should move in opposite directions. On the other hand, from the view of the lesion area marked by surgeon, it can be seen that the position of the tip of the puncture needle in the X - Y plane is of more significance. In order to obtain a feasible solution for the motor control input, the second constraint is defined. That is, when block- a moves to the limit position and still cannot reach the lesion area, block- b is then moved. The inverse kinematics model is constructed using geometric theorem based on the constraints:

$$\begin{aligned} [x_b, y_b] &= [0, 0] \\ \text{s.t. } -\eta &\leq x_t, y_t \leq \eta, \end{aligned} \quad (10)$$

and

$$\begin{aligned} [x_a, y_a] &= [-2\text{sgn}(x_t), -2\text{sgn}(y_t)] \\ \text{s.t. } |x_t, y_t| &\leq -\eta, \end{aligned} \quad (11)$$

where η is the largest distance that the tip of the puncture needle can reach when only block- a is in motion; x_t and y_t are the lesion's target point coordinates marked by the surgeon. However, no complete feasible solution can be obtained yet. Then, the solution of the coordinates of block- a within the limit η is discussed:

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} -d \cdot \text{sgn}(x_t) \tan(\arcsin(\theta)) \sin\left(\arctan\left(\frac{x_t}{y_t}\right)\right) \\ -d \cdot \text{sgn}(y_t) \tan(\arcsin(\theta)) \cos\left(\arctan\left(\frac{x_t}{y_t}\right)\right) \end{bmatrix}, \quad (12)$$

where θ is the angle between the block and puncture needle:

$$\theta = \arccos\left(\frac{-z_t}{h}\right). \quad (13)$$

So far, a feasible solution within the limit η from equations (10) and (12) is obtained:

$$\begin{bmatrix} x_a \\ y_a \\ x_b \\ y_b \end{bmatrix} = \begin{bmatrix} -d \cdot \text{sgn}(x_t) \tan(\arcsin(\theta)) \sin\left(\arctan\left(\frac{x_t}{y_t}\right)\right) \\ -d \cdot \text{sgn}(y_t) \tan(\arcsin(\theta)) \cos\left(\arctan\left(\frac{x_t}{y_t}\right)\right) \\ 0 \\ 0 \end{bmatrix}$$

(14)

s.t. $-\eta \leq x_t, x_t \leq \eta.$

Finally, the feasible solution in the other case will be analyzed to obtain the complete feasible solution. When the lesion area exceeds the limit η , the coordinate values of block- a are fixed. Therefore, according to the angle θ between the block and the puncture needle, the relationship between x_b and x_b can be acquired:

$$\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} = d \cdot \tan(\theta), \quad (15)$$

$$\frac{x_a - x_b}{y_a - y_b} = \frac{x_t - x_b}{y_t - y_b}. \quad (16)$$

The solution of coordinate value of block- b can be acquired by equations (15) and (16):

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} \frac{-EB - \sqrt{EB^2 - 4 \cdot EA \cdot EC}}{2 \cdot EA} \\ \frac{E - 2 \cdot (x_a - x_t)}{y_a - y_t} \end{bmatrix}, \quad (17)$$

where E , EA , EB , and EC are known parameters as follows:

$$\begin{aligned} E &= \frac{h^2 - z_t^2 + x_a^2 - y_t^2 + y_a^2}{2}, \\ EA &= \frac{(x_a - x_t)^2}{(y_a - y_t)^2} + 1, \\ EB &= \frac{2d(y_a - y_t)(x_a - x_t) - 2(y_a - y_t)^2 x_t - 2E(x_a - x_t)}{(y_a - y_t)^2}, \\ EC &= \frac{(y_a - y_t)^2(x_t^2 + y_t^2 - h^2 + z_t^2) - 2dE(y_a - y_t)}{(y_a - y_t)^2}. \end{aligned} \quad (18)$$

Then, the feasible solution that exceeds the limit η is obtained by equations (11) and (17):

$$\begin{bmatrix} x_a \\ y_a \\ x_b \\ y_b \end{bmatrix} = \begin{bmatrix} -2\text{sgn}(x_t) \\ -2\text{sgn}(y_t) \\ \frac{-EB - \sqrt{EB^2 - 4 \cdot EA \cdot EC}}{2 \cdot EA} \\ \frac{E - 2 \cdot (x_a - x_t)}{y_a - y_t} \end{bmatrix} \quad (19)$$

$$\text{s.t. } |x_t, y_t| \leq -\eta.$$

Now, the complete solution of the kinematic model of the surgical robot has been obtained by equations (14) and (19). The corresponding motor control angle can be obtained through the lesion area marked by the surgeon.

3. Model Parameter Estimation

After the desired control input is obtained, it cannot, in fact, directly act on the robotic controller. In order to make the state-space model of the robot closer to the actual model, the transmission process of the surgical robot will be divided into two parts for modeling. The first part is the linear system model of the block movement, and the second part is the nonlinear system model of the puncture needle movement. The determinant ratio is used to estimate the order of the

linear system model, and a Kalman filter is established to estimate the linear system model parameters. Taylor formula is used to linearize the nonlinear system model into the controller.

3.1. Linear Part. Firstly, the linear system model is a MIMO system; however, each input corresponds to only one output. Thus, the MIMO system is equivalent to four single-input single-output (SISO) systems. The order and parameters of the SISO system are considered:

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_0 u(k) + e(k), \quad (20)$$

where $e(k)$ is the Gaussian white noise with mean zero, $u(k)$ is the input signal, and $y(k)$ is the output signal. The input and output signals at each moment are known, but parameters are unknown. Then, input signal and output signal are used to construct determinants for comparison to obtain the ratio $DR^*(\hat{n})$:

$$DR^*(\hat{n}) = \frac{\det[\mathbf{H}^*(\hat{n})]}{\det[\mathbf{H}^*(\hat{n}+1)]}, \quad (21)$$

where \hat{n} is the estimated order and \mathbf{H}^* is a matrix composed of input and output and has the following relationship:

$$\mathbf{H}_n^* = \begin{bmatrix} y(\hat{n}) & y(\hat{n}-1) & \dots & y(1) & u(\hat{n}) & u(\hat{n}-1) & \dots & u(1) \\ y(\hat{n}+1) & y(\hat{n}) & \dots & y(2) & u(\hat{n}+1) & u(\hat{n}) & \dots & u(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ y(\hat{n}+L-1) & y(\hat{n}+L-2) & \dots & y(L) & u(\hat{n}+L-1) & u(\hat{n}+L-2) & \dots & u(L) \end{bmatrix}, \quad (22)$$

$$\mathbf{H}^*(\hat{n}) = \frac{1}{L} \mathbf{H}_n^* \mathbf{H}_n^{*T},$$

where L is the sampling length of the signal. \hat{n} gradually increases from one. If only $DR^*(\hat{n})$ has a significant growth compared to $DR^*(\hat{n}-1)$, \hat{n} is the estimated order.

Next, a Kalman filter is designed to dynamically identify the system's unknown parameters a_i ($i = 1, 2, \dots, n$) and b_0 . The Kalman filter is usually used to optimally estimate the state value of the state-space equation, which is composed of the unknown parameters:

$$\begin{cases} \mathbf{X}(k+1) = \Phi \mathbf{X}(k) + \Gamma \mathbf{W}(k), \\ \mathbf{Y}(k) = \mathbf{H}(k) \mathbf{X}(k) + \mathbf{V}(k), \end{cases} \quad (23)$$

where both $\mathbf{V}(k)$ and $\mathbf{W}(k)$ are noise matrix composed of Gaussian white noise with zero mean. Γ is the process noise drive matrix. $\mathbf{X}(k)$ is the state matrix at the k -th sampling moment. Φ is the state transition matrix at the k -th sampling moment. $\mathbf{Y}(k)$ is the output matrix at the k -th sampling moment. And we assume that the variances of the two noises

are \mathbf{Q} and \mathbf{R} . Then, the update equation of Kalman filter can be derived based on Euclidean theorem:

$$\begin{cases} \hat{\mathbf{X}}(k+1|k) = \Phi \hat{\mathbf{X}}(k|k), \\ \hat{\mathbf{X}}(k+1|k+1) = \hat{\mathbf{X}}(k+1|k) + \mathbf{K}(k+1)(\mathbf{Y}(k+1) - \mathbf{H} \hat{\mathbf{X}}(k+1|k)), \\ \mathbf{K}(k+1) = \mathbf{K}(k+1|k) \mathbf{H}^T [\mathbf{H} \mathbf{P}(k+1|k) \mathbf{H}^T + \mathbf{R}]^{-1}, \\ \mathbf{P}(k+1|k) = \Phi \mathbf{P}(k|k) \Phi^T + \Gamma \mathbf{Q} \Gamma^T, \\ \mathbf{P}(k+1|k+1) = [\mathbf{I}_n - \mathbf{K}(k+1) \mathbf{H}] \mathbf{P}(k+1|k), \end{cases} \quad (24)$$

where $\hat{\mathbf{X}}(k|k)$ is the optimal estimate at the k -th sampling moment, $\mathbf{P}(k|k)$ is the error covariance of the current sampling time, and $\mathbf{K}(k+1|k)$ is the gain of the Kalman filter, which is used to update the values of $\hat{\mathbf{X}}(k|k)$ and

$\mathbf{P}(k|k)$. In this paper, the estimated state matrix with unknown parameters is constructed:

$$\begin{cases} x_1(k) = a_1(k), \\ x_2(k) = a_2(k), \\ \vdots \\ x_n(k) = a_n(k), \\ x_{n+1}(k) = b_0(k). \end{cases} \quad (25)$$

Then the system's state equation is obtained as follows:

$$\mathbf{X}(k+1) = \mathbf{X}(k) + \mathbf{W}(k), \quad (26)$$

where $\mathbf{X}(k)$ is composed of $x_i(k) (i = 1, 2, \dots, n+1)$. We use the observed input and output data to construct the matrix $\mathbf{H}(k)$ and state equation:

$$\begin{aligned} \mathbf{H}(k) &= [-y(k-1), -y(k-2), \dots, -y(k-n), u(k)], \\ \begin{cases} \mathbf{X}(k+1) = \mathbf{X}(k) + \mathbf{W}(k), \\ y(k) = \mathbf{H}(k)\mathbf{X}(k) + e(k). \end{cases} \end{aligned} \quad (27)$$

Finally, state equation (28) can be updated by equation (25) to obtain the actual system parameters. The SISO systems of four motors are combined to obtain the MIMO system of the linear part:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}, \quad (28)$$

$$\mathbf{B}^T = [0 \ 0 \ \dots \ 0 \ b_0], \quad (29)$$

$$\mathbf{C} = [0 \ 0 \ \dots \ 0 \ 1], \quad (30)$$

$$\begin{aligned} \mathbf{X}(k+1) &= \begin{bmatrix} \mathbf{A}_{x_a} & 0 & 0 & 0 \\ 0 & \mathbf{A}_{y_a} & 0 & 0 \\ 0 & 0 & \mathbf{A}_{x_b} & 0 \\ 0 & 0 & 0 & \mathbf{A}_{y_b} \end{bmatrix} \mathbf{X}(k) \\ &+ \begin{bmatrix} \mathbf{B}_{x_a} & 0 & 0 & 0 \\ 0 & \mathbf{B}_{y_a} & 0 & 0 \\ 0 & 0 & \mathbf{B}_{x_b} & 0 \\ 0 & 0 & 0 & \mathbf{B}_{y_b} \end{bmatrix} \mathbf{u}(k), \end{aligned} \quad (31)$$

$$\mathbf{Y}(k) = \begin{bmatrix} \mathbf{C}_{x_a} & 0 & 0 & 0 \\ 0 & \mathbf{C}_{y_a} & 0 & 0 \\ 0 & 0 & \mathbf{C}_{x_b} & 0 \\ 0 & 0 & 0 & \mathbf{C}_{y_b} \end{bmatrix} \mathbf{X}(k), \quad (32)$$

where the linear part's MIMO system is expressed by re-writing the state equations (32) and (33); $\mathbf{X}(k)$ consists of

samples of the output y before the k -th time step; \mathbf{A}_{x_a} , \mathbf{A}_{x_b} , \mathbf{A}_{y_a} , and \mathbf{A}_{y_b} are composed of each motor model parameter matrix \mathbf{A} ; similarly, other parameters of the MIMO system can also be expressed by equations (30) and (31).

3.2. Nonlinear Part. The nonlinear part of the system mainly represents the relationship between the rotation of the puncture needle and the coordinates of the block. The relationship between them through equations (2)–(9) can be obtained as follows:

$$(x_e, y_e, z_e) = \mathbf{f}(x_a, y_a, x_b, y_b), \quad (33)$$

where \mathbf{f} is their nonlinear mapping. Then, a first-order expansion of Taylor's formula is performed for \mathbf{f} to obtain the following linear relationship:

$$[x_e, y_e, z_e]^T = \mathbf{J}_f \begin{bmatrix} x_a - x_{ar} \\ y_a - y_{ar} \\ x_b - x_{br} \\ y_b - y_{br} \end{bmatrix} + \boldsymbol{\sigma}, \quad (34)$$

$$\mathbf{J}_f = \begin{bmatrix} \frac{\partial f_{x_e}}{\partial x_a} & \frac{\partial f_{x_e}}{\partial y_a} & \frac{\partial f_{x_e}}{\partial x_b} & \frac{\partial f_{x_e}}{\partial y_b} \\ \frac{\partial f_{y_e}}{\partial x_a} & \frac{\partial f_{y_e}}{\partial y_a} & \frac{\partial f_{y_e}}{\partial x_b} & \frac{\partial f_{y_e}}{\partial y_b} \\ \frac{\partial f_{z_e}}{\partial x_a} & \frac{\partial f_{z_e}}{\partial y_a} & \frac{\partial f_{z_e}}{\partial x_b} & \frac{\partial f_{z_e}}{\partial y_b} \end{bmatrix}, \quad (35)$$

where $\mathbf{Y}_r(k) = [x_{ar}, y_{ar}, x_{br}, y_{br}]^T$ is the target reference values obtained by the inverse kinematics model, $\boldsymbol{\sigma}$ is an infinitesimal quantity, and \mathbf{J}_f is the linearized Jacobian matrix. Finally, the complete input and output linear system model of the surgical robot can be obtained through equations (33) and (35) as follows:

$$\mathbf{X}(k+1) = \mathbf{A}_s \mathbf{X}(k) + \mathbf{B}_s \mathbf{u}(k), \quad (36)$$

$$[x_e, y_e, z_e]^T = \mathbf{J}_f [\mathbf{C}_s \mathbf{X}(k) - \mathbf{Y}_r(k)], \quad (37)$$

where

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{A}_{x_a} & 0 & 0 & 0 \\ 0 & \mathbf{A}_{y_a} & 0 & 0 \\ 0 & 0 & \mathbf{A}_{x_b} & 0 \\ 0 & 0 & 0 & \mathbf{A}_{y_b} \end{bmatrix}, \quad (38)$$

$$\mathbf{B}_s = \begin{bmatrix} \mathbf{B}_{x_a} & 0 & 0 & 0 \\ 0 & \mathbf{B}_{y_a} & 0 & 0 \\ 0 & 0 & \mathbf{B}_{x_b} & 0 \\ 0 & 0 & 0 & \mathbf{B}_{y_b} \end{bmatrix}, \quad (39)$$

$$\mathbf{C}_s = \begin{bmatrix} \mathbf{C}_{x_a} & 0 & 0 & 0 \\ 0 & \mathbf{C}_{y_a} & 0 & 0 \\ 0 & 0 & \mathbf{C}_{x_b} & 0 \\ 0 & 0 & 0 & \mathbf{C}_{y_b} \end{bmatrix}. \quad (40)$$

$$\mathbf{B}_m = \begin{bmatrix} \mathbf{B}_s \\ \mathbf{I}_{4 \times 4} \end{bmatrix}, \quad (46)$$

$$\mathbf{C}_m = [\mathbf{C}_s, 0]. \quad (47)$$

4. Model Predictive Control

Model predictive control (MPC) is now a well-known model-based control strategy, and it has gone through a period of development [41–43]. Taking the state-space model as an example, MPC uses the state of the model at the current time and the target state to estimate the control sequences at the next time under constraints. Compared with proportional-integral-derivative (PID) controller, MPC can predict the future behavior of the system. The optimal future control sequences are estimated by the known data. These sequences are evaluated at each sample time by the optimizer with constraints.

MPC consists of three components [44]: (1) a loss function, (2) constraints in the form of equality and inequality, and (3) initial conditions. The purpose of the loss function is to minimize the distance between the target state and the state at the current time [45], thereby minimizing the energy required by the control system. The equality constraints consist of the system's state equations. The limits on the state variables are considered as the inequality constraints. If all constraints are linear, the problem forms a convex set. Meanwhile, if the control system is also convex, MPC can be solved as a convex optimization problem. Similarly, the MPC can be solved as a quadratic problem if the control system is a quadratic problem.

MPC uses the system's model to predict the future control sequences [46]. As shown in Figure 5, the optimizer in MPC is to get the best control input. The linear time-invariant system will be considered in the discrete time domain as shown in equation (37). However, in equation (37), the state and input variables are added by two matrices to obtain the final output. To comprehensively consider the state and input variables, equations (37) and (38) are rewritten as follows:

$$\xi(k+1) = \mathbf{A}_m \xi(k) + \mathbf{B}_m \Delta \mathbf{u}(k), \quad (41)$$

$$[x_e, y_e, z_e]^T + \mathbf{J}_r \mathbf{Y}_r(k) = \mathbf{C}_m \xi(k), \quad (42)$$

where

$$\xi(k) = \begin{bmatrix} \mathbf{X}(k) \\ \mathbf{u}(k-1) \end{bmatrix}, \quad (43)$$

$$\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1), \quad (44)$$

$$\mathbf{A}_m = \begin{bmatrix} \mathbf{A}_s & \mathbf{B}_s \\ 0 & \mathbf{I}_{4 \times 4} \end{bmatrix}, \quad (45)$$

Equations (42) and (43) are state-space equations that consider both state and input variables. We mainly discuss the optimization problem of equation (42), namely, the state space. This equation can make predictions about the future state, that is, the prediction step required for optimization can be obtained:

$$\mathbf{Y}_\xi = \Psi \xi(k) + \Theta \Delta \mathbf{U}(k), \quad (48)$$

where

$$\mathbf{Y}_\xi = [\xi(k+1), \xi(k+2), \dots, \xi(k+N_p)]^T, \quad (49)$$

$$\Psi = [\mathbf{A}_m, \mathbf{A}_m^2, \dots, \mathbf{A}_m^{N_p}]^T, \quad (50)$$

$$\Theta = \begin{bmatrix} \mathbf{B}_m & 0 & 0 & \dots & 0 \\ \mathbf{A}_m \mathbf{B}_m & \mathbf{B}_m & 0 & \dots & 0 \\ \mathbf{A}_m^2 \mathbf{B}_m & \mathbf{A}_m \mathbf{B}_m & \mathbf{B}_m & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_m^{N_p-1} \mathbf{B}_m & \mathbf{A}_m^{N_p-2} \mathbf{B}_m & \mathbf{A}_m^{N_p-3} \mathbf{B}_m & \dots & \mathbf{A}_m^{N_p-N_c} \mathbf{B}_m \end{bmatrix}, \quad (51)$$

where N_p is the predicted step size of the system and N_c is the control step size of the system. Next, the reference trajectory $\xi_r = [\mathbf{X}_r, \mathbf{u}_r]^T$ is used to construct the following loss function:

$$\mathbf{J}_\xi = [\mathbf{Y}_\xi - \mathbf{Y}_{\xi_r}]^T \mathbf{Q}_\xi [\mathbf{Y}_\xi - \mathbf{Y}_{\xi_r}], \quad (52)$$

where \mathbf{Q}_ξ is the weight matrix. Considering that the control input cannot change too much within a period of time, the following loss function is constructed:

$$\mathbf{J}_u = \Delta \mathbf{U}^T \mathbf{P}_u \Delta \mathbf{U}, \quad (53)$$

where \mathbf{P}_u is the weight matrix. And the objective function to be optimized can be obtained by equations (53) and (54), which can be seen as a quadratic programming problem:

$$\begin{aligned} \mathbf{J} &= \mathbf{J}_\xi + \mathbf{J}_u \\ &= [\Psi \xi + \Theta \Delta \mathbf{U}]^T \mathbf{Q}_\xi [\Psi \xi + \Theta \Delta \mathbf{U}] - \mathbf{Y}_{\xi_r}^T \mathbf{Q}_\xi \mathbf{Y}_{\xi_r} + \Delta \mathbf{U}^T \mathbf{P}_u \Delta \mathbf{U} \\ &= [\Psi \xi]^T \mathbf{Q}_\xi [\Psi \xi] - \mathbf{Y}_{\xi_r}^T \mathbf{Q}_\xi \mathbf{Y}_{\xi_r} + [\Theta \Delta \mathbf{U}]^T \mathbf{Q}_\xi [\Theta \Delta \mathbf{U}] \\ &\quad + 2[\Psi \xi]^T \mathbf{Q}_\xi [\Theta \Delta \mathbf{U}] + \Delta \mathbf{U}^T \mathbf{P}_u \Delta \mathbf{U} \\ &= \Delta \mathbf{U}^T [\Theta \mathbf{Q}_\xi \Theta + \mathbf{P}_u] \Delta \mathbf{U} + 2[\Psi \xi]^T \mathbf{Q}_\xi [\Theta \Delta \mathbf{U}]. \end{aligned} \quad (54)$$

Finally, the constraints of the control variable $\Delta \mathbf{U}$ of the loss function to construct a standard form of quadratic programming will be set:

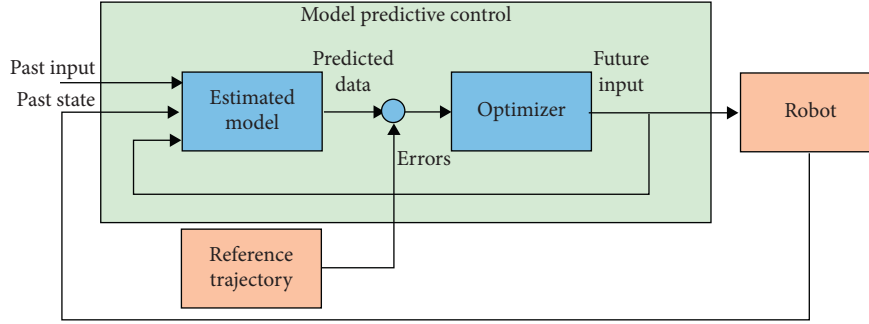


FIGURE 5: MPC controller internal rolling optimization process. The optimizer optimizes the current control input to obtain the optimal control input at the next step.

$$\begin{aligned} \min_{\Delta U} \quad & J \\ \text{s.t.} \quad & lb \leq \Delta U \leq ub, \end{aligned} \quad (55)$$

where lb and ub are the lower and upper bounds of the variable ΔU . Then, the best control sequences can be obtained by equation (45).

5. Simulation Setup and Results

This section mainly describes the simulated experimental settings and experimental results about this article. This article uses MATLAB to build a remote subsystem, which contains the MPC controller and parameter identification system. Under the function of the parameter identification system, the MPC controller obtains the robotic kinematics model realistically and outputs the optimal control sequence. In the first place, as shown in Figure 2, there is actually a TD when the MPC controller is applied to the system; thus, in the experiment, a delay system will be considered for comparison. We will, in addition, divide it into two experiments: (1) order and parameter identification of the system's linear part and (2) performance of MPC controller and PID controller in delay system.

5.1. Model Identification Experiment. Above all, only the model identification of the linear part of the system is considered. And the order needs to be identified through offline data. The system's order can be obtained by equation (21). In this paper, the system's order $\hat{n} = 2$, the pitch of the screw $\rho = 0.02$, and the largest movement distance of the block $\eta = 2$. Next, the Kalman filter is designed to identify the system parameters. The state equation of the system is rewritten to identify the parameters of the system, and the result is represented by \mathbf{H} in equation (28). The results of parameter identification are shown in Figure 6.

It can be seen from Figure 6 that the three parameters can stabilize at a certain value $a_1 = 0.0125$, $a_2 = -0.0257$, and $b = 1.818$ within the step size. However, it is not known whether a system composed of stable parameters can approximate the real system. Thus, as shown in Figure 7, the output of the actual system is compared with the output of the identified model. The figure shows the target curve, the output curve of the identification model, and the error curve

between them. The value of the error curve floats above and below zero.

5.2. MPC Performance. First, the model of the linear part of the control system can be obtained by equation (28), and the parameters of the model are obtained in the previous experiment, which contains the effective length of the needle $h = 0.05$ m and the distance between blocks $d = 0.03$ m. And the linearized model of the nonlinear part of the control system can be obtained by equation (35). Then, the two models are combined by equation (37) to obtain a complete model of the control system. In addition, in order to comprehensively consider the control input and system state, the state-space equation of the control system is rewritten by equations (42) and (43). Finally, the prediction equation (49) required by the MPC controller is obtained by equation (42), which is used for rolling optimization in the MPC controller. The constraints of quadratic programming $lb = -0.002$ and $ub = 0.002$ are constraints on the difference values of control sequence. The optimizer of the MPC controller outputs the optimal control sequence through equation (55). In this paper, traditional PID controller is designed to compare with the MPC controller. In fact, as the controller takes time to run, there is a delay in the output of the controller, which is equivalent to the control delay system. Thus, we set different TDs in the control system. We set the weight matrix \mathbf{Q}_ξ and \mathbf{P}_u in the MPC controller as the identity matrix and the parameters of the PID controller $K_p = 13$, $K_i = 2$, and $K_d = 0$.

Figure 8 shows the Y-axis motion trajectory of block- a , which illustrates that with increase in the delay time, the response time of the system also increases and the desired trajectory can be tracked by the MPC controller. As the delay increases, the response speed of the system also becomes slower, and the output of the MPC controller will produce 2% overshoot. However, the PID controller does not perform well in the case of TD. The output of the PID controller diverges when $TD = 250$ ms, making the system unstable. When $TD = 150$ ms, the PID controller makes the system gradually stable, but the setting time $t_s = 4$ s and the 10% overshoot exceed the constraints of the surgical robot.

Then, the output of the position of the tip of the puncture needle will be analyzed. As shown in Figure 9, in the open-loop state without controller, the output value of the system

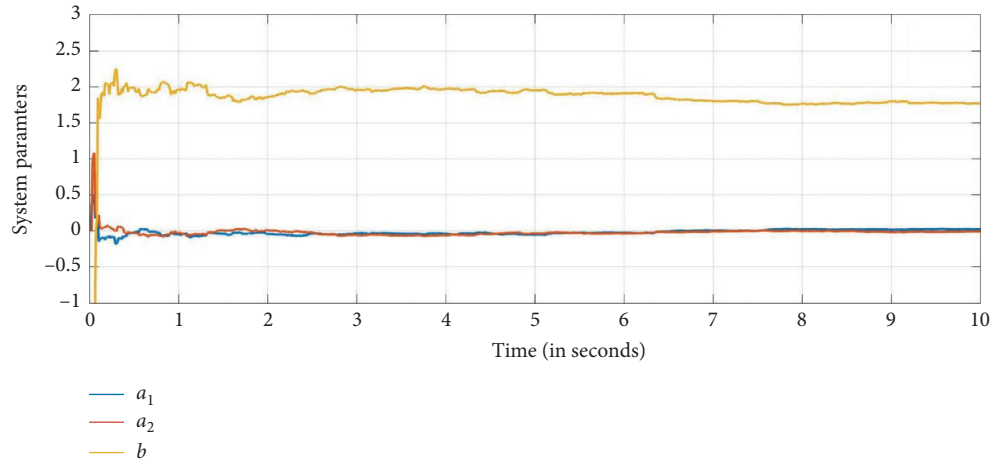


FIGURE 6: The identification result of the system parameters, parameters a_1 and a_2 are the coefficients of the output term, and parameter b is the coefficient of the input term.

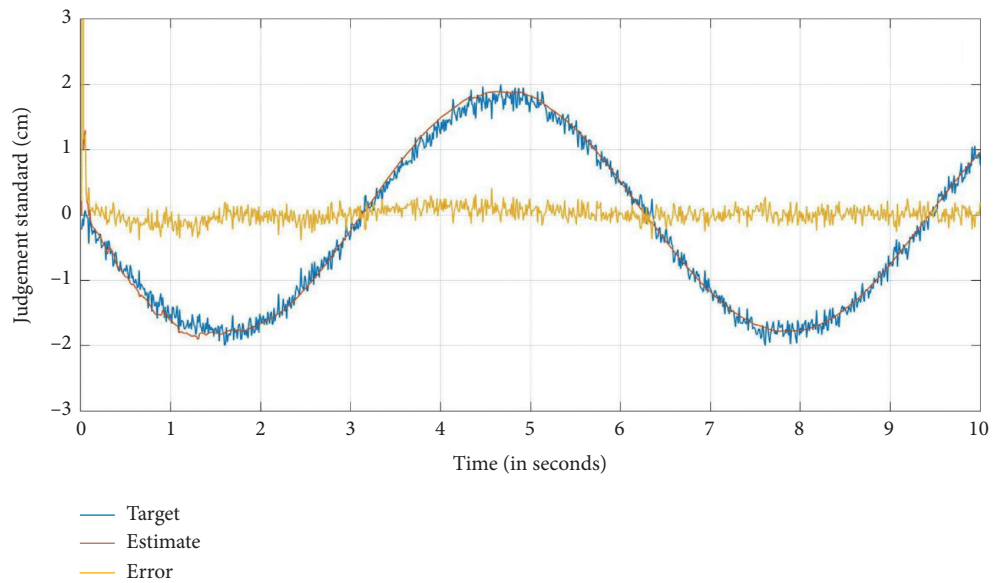


FIGURE 7: Comparison of system parameter identification. The error is the difference between the target value and the estimated value.

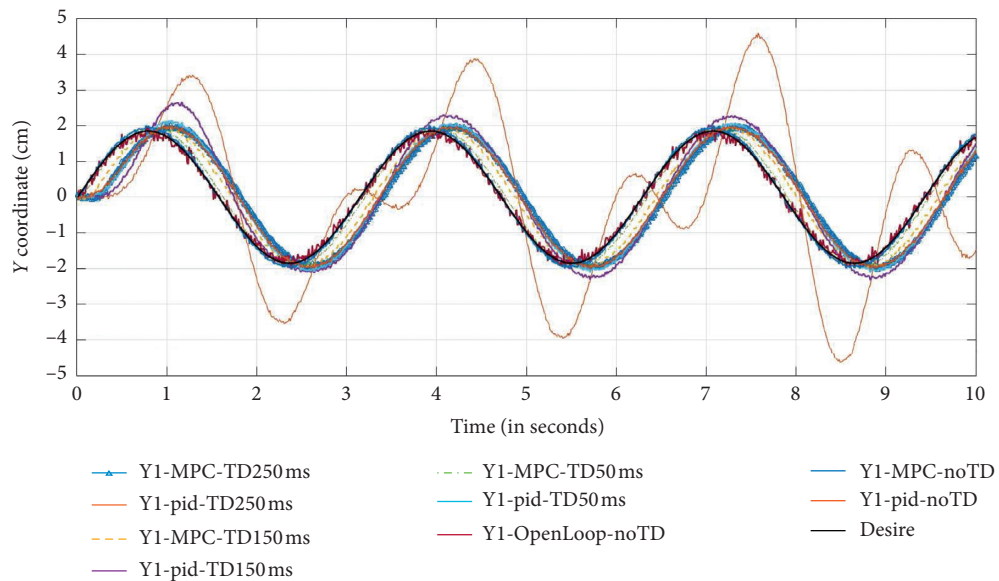


FIGURE 8: The performance of the Y-axis motion trajectory of block A under different delays and different controllers.

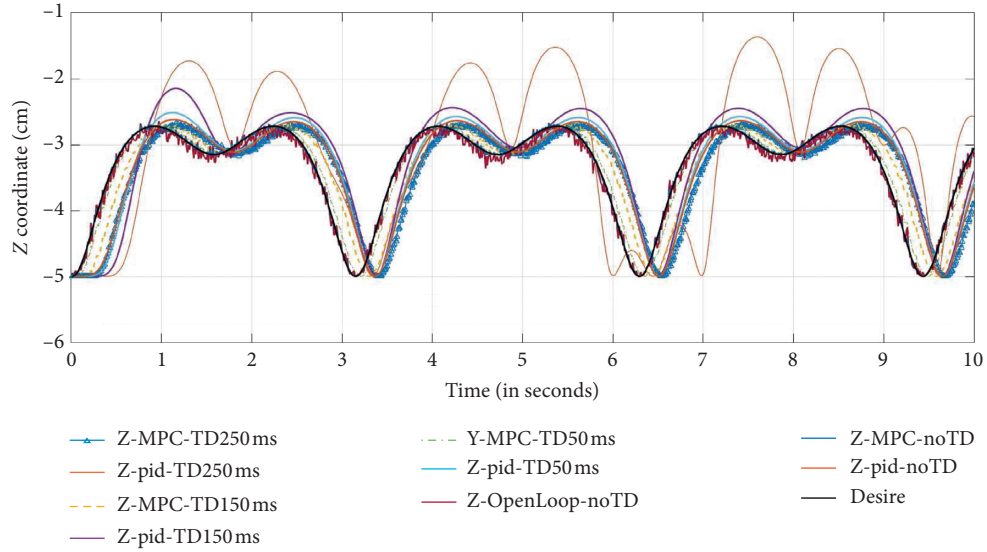


FIGURE 9: The performance of the Z-axis motion trajectory of the tip of puncture needle under different delays and different controllers.

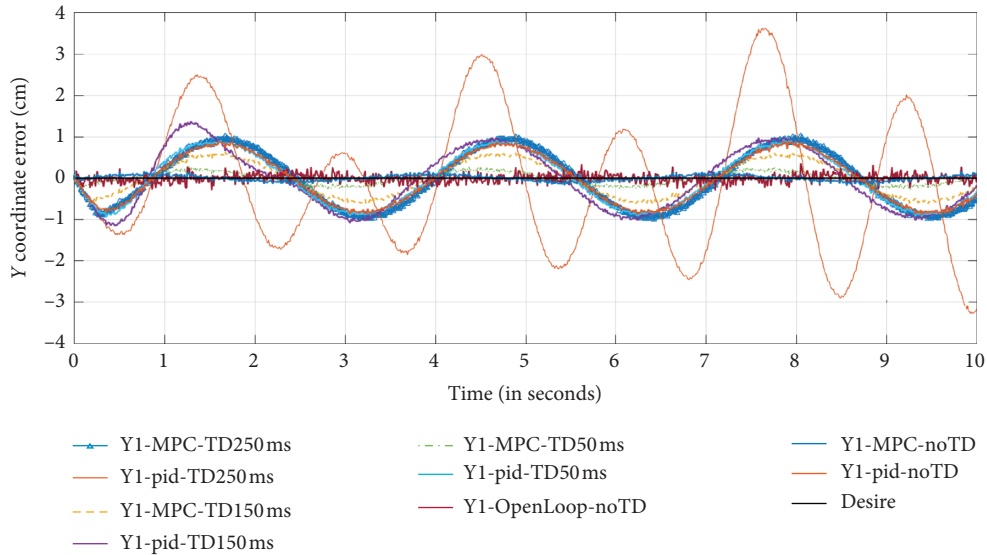


FIGURE 10: The performance of the coordinate difference in the Y-axis direction of the block. The TD will cause the regular periodic error of the sine wave.

has a large fluctuation around the desire trajectory. The closed-loop system formed by the MPC controller can track the desired trajectory with TD, which will produce 2% overshoot in the delay system. However, the closed-loop system formed by the PID controller is sensitive to TD. When there is a large TD in the system, it will produce a larger overshoot than the MPC controller and even make the system unstable.

Finally, in order to more clearly show the advantages of the MPC controller, a differential operation is performed on the state value of the sine wave shown in Figure 8 to obtain the distance from the target value shown in Figure 10. In the figure, the output of the MPC controller applied to the delay system produces phase shifts at different distances relative to the expected value. However,

when the PID controller acts on a nondelayed system, phase shift occurs. As the delay increases, even the output diverges. The above experimental results show that the effect of MPC controller is better than that of the PID controller. The error variance is shown in Table 1, the desired error variance is zero, and the closer to the ideal error variance, the more ideal the system's outputs are. The error variance of the MPC controller in the TD system is smaller than the error variance generated by the PID controller. And when $TD = 0$, the error variance generated by the MPC controller is even smaller than the error variance generated by the nondelay system with noise. From the result, we can get that for the parallel mechanism robot involved in this article, MPC has a superior control effect because MPC can control the MIMO system.

TABLE 1: Error variance (cm²) under different TD.

Y1-MPC-250	Y1-MPC-150	Y1-MPC-50	Y1-MPC-0	Y1-pid-250	Y1-pid-150	Y1-pid-50	Y1-PID-0	Y1-reality-0
0.6625	0.4048	0.1512	0.0728	1.6730	0.7067	0.6178	0.6860	0.1077

6. Conclusion and Future Work

We introduced a surgical robotic puncture needle positioning platform that can improve the efficiency of biopsy. Kinematic analysis of the platform illustrates that the puncture needle can work in the working space through multiple angles that can be obtained by decoupling the robotic kinematic. In this paper, two constraints are defined for decoupling the kinematic, which ensure the robotic largest workspace. The designed MPC controller makes the control flow of the positioning device form a closed loop. And the state equation of the MPC controller needed is divided into two parts: linear part and nonlinear part. In order to guarantee the simulation of robotic system closer to the reality, the Kalman filter system parameters identification approach is used to obtain the system's parameters in the linear part. In the nonlinear part, Taylor formula is used to linearize the system, and the linearization system only contains a Jacobian matrix. Next, the two parts of the system are combined into a complete robot system state equation. In simulation, the closed-loop control significantly improves the stability of robot needle positioning. And the performance of MPC controller is better than that of the PID controller. Finally, the system is open source in Github (<https://github.com/tKsome/MPC-Puncture-Robot>) to reduce the amount of development time for other researchers involved in the control of the medical robotic MIMO system. The system is flexible and may be useful for most image-based intervention procedures. In future work, we will test the performance of the proposed approach on the real-work robotic platform. Furthermore, a force control strategy will be proposed such that a force feedback can be provided to the surgeon through remote operation.

Data Availability

The data included in this study are available upon proper request by contact with the corresponding author.

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 with grant no. 61803103 and Science and Technology Program of Guangdong with grant nos. 2019B010140002 and 2019B010142001.

References

- [1] P. Kulkarni, S. Sikander, P. Biswas, S. Frawley, and S.-E. Song, "Review of robotic needle guide systems for percutaneous intervention," *Annals of Biomedical Engineering*, vol. 47, p. 7, 2019.
- [2] B. Schell, K. Eichler, M. Mack et al., "Roboterunterstützte punktion in einem hochfeld-kernspintomografen—erste klinische ergebnisse," *RöFo—Fortschritte auf dem Gebiet der Röntgenstrahlen und der Bildgebenden VERFAHREN*, vol. 184, no. 1, pp. 42–47, 2012.
- [3] N. Hungr, I. Bricault, P. Cinquin, and C. Fouard, "Design and validation of a CT- and MRI-guided robot for percutaneous needle procedures," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 973–987, 2016.
- [4] N. Simaan, R. M. Yasin, and L. Wang, "Medical technologies and challenges of robot assisted minimally invasive intervention and diagnostics," 2018, <https://arxiv.org/abs/1807.03731>.
- [5] Y. Chen, S. Xu, A. Squires et al., "MRI-guided robotically assisted focal laser ablation of the prostate using canine cadavers," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 7, pp. 1434–1442, 2017.
- [6] S. Lim, K. Sharma, P. Li et al., "Robotically assisted long bone biopsy under MRI: cadaver study results," *International Journal of Computer-Assisted Radiology and Surgery*, vol. 14, no. 1, pp. 147–156, 2018.
- [7] M. Giannakou, C. Yiallouras, G. Menikou, C. Ioannides, and C. Damianou, "MRI-guided frameless biopsy robotic system with the inclusion of unfocused ultrasound transducer for brain cancer ablation," *The International Journal of Medical Robotics and Computer Assisted Surgery :MRCAS*, vol. 15, p. e1951, 2019.
- [8] B. Fallahi, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, "Partial estimation of needle tip orientation in generalized coordinates in ultrasound image-guided needle insertion," in *Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1604–1609, Banff, Canada, 2016.
- [9] Y. Kurose, Y. M. Baek, Y. Kamei et al., "Preliminary study of needle tracking in a microsurgical robotic system for automated operations," in *Proceedings of the 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 627–630, Gwangju, Republic of Korea, 2013.
- [10] A. Melzer, B. Gutmann, T. Remmele et al., "Innomotion for percutaneous image-guided interventions," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 3, pp. 66–73, 2008.
- [11] H. J. Won, N. Kim, G. Kim, J. B. Seo, and H. Kim, "Validation of a CT-guided intervention robot for biopsy and radio-frequency ablation: experimental study with an abdominal phantom," *Diagnostic and Interventional Radiology*, vol. 23, pp. 233–237, 2017.
- [12] D. Navarro-Alarcon, S. Singh, T. Zhang et al., "Developing a compact robotic needle driver for MRI-guided breast biopsy in tight environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1648–1655, 2017.
- [13] S. Park, J.-G. Kim, K.-W. Lim et al., "A magnetic resonance image-guided breast needle intervention robot system: overview and design considerations," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, pp. 1319–1331, 2017.

- [14] T. Zhang, D. Navarro-Alarcon, K. W. Ng, M. K. Chow, Y. Liu, and H. L. Chung, "A novel palm-shape breast deformation robot for MRI-guided biopsy," in *Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 527–532, Qingdao, China, December 2016.
- [15] Y. Chen, A. Squires, R. Seifabadi et al., "Robotic system for MRI-guided focal laser ablation in the prostate," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 107–114, 2017.
- [16] Z. Cao, T. Liao, W. Song, Z. Chen, and C. Li, "Detecting the shuttlecock for a badminton robot: a yolo based approach," *Expert Systems with Applications*, vol. 164, Article ID 113833, 2020.
- [17] W. Wang, A. N. Viswanathan, A. L. Damato et al., "Evaluation of an active magnetic resonance tracking system for interstitial brachytherapy," *Medical Physics*, vol. 42, no. 12, p. 7114, 2015.
- [18] Y. Chen, W. Wang, E. J. Schmidt et al., "Design and fabrication of MR-tracked metallic stylet for gynecologic brachytherapy," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 956–962, 2016.
- [19] J. Ladoiye, D. Neculescu, and J. Sasiadek, "Force control of surgical robot with time delay using model predictive control," in *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics*, Porto, Portugal, 2018.
- [20] A. Squires, J. Oshinski, J. Lamanna, and Z. T. H. Tse, "Spinotemplate: a platform for MRI-guided spinal cord injections," *Journal of Medical Robotics Research*, vol. 1, no. 1, Article ID 1640006, 2016.
- [21] L. Zhang, Z. Chen, W. Cui et al., "Wifi-based indoor robot positioning using deep fuzzy forests," *IEEE Internet of Things Journal*, 2020.
- [22] W. Chen, T. Liao, Z. Li et al., "Using FTOC to track shuttlecock for the badminton robot," *Neurocomputing*, vol. 334, pp. 182–196, 2019.
- [23] Y. Koseki, D. De Lorenzo, K. Chinzei, and A. Okamura, "Coaxial needle insertion assistant for epidural puncture," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, 2011.
- [24] J. Kettenbach and G. Kronreif, "Robotic systems for percutaneous needle-guided interventions," *Minimally Invasive Therapy & Allied Technologies*, vol. 24, 2014.
- [25] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.
- [26] S. Zampieri, "Trends in networked control systems," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 17, 2008.
- [27] X.-M. Zhang, Q.-L. Han, X. Ge et al., "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, 2020.
- [28] D. Uygun, "Low-power and cost-effective speed control card for remotely-controlled dc motors," *Electronics World*, vol. 121, pp. 16–19, 2015.
- [29] T. Savi and M. Radonjic, "One solution for remote control through internet application," in *Proceedings of the 2016 Information Technologies Conference*, Žabljak, Montenegro, 2016.
- [30] H. Su, C. Yang, G. Ferrigno, and E. De Momi, "Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
- [31] H. Su, Y. Schmirander, S. Valderrama-Hincapie et al., "Asymmetric bimanual control of dual-arm serial manipulator for robot-assisted minimally invasive surgeries," *Sensors and Materials*, vol. 32, p. 1223, 2020.
- [32] C. Liu, W. H. Chen, and J. Andrews, "Model predictive control for autonomous helicopters with computational delay," in *Proceedings of the UKACC International Conference on Control*, Coventry, UK, 2010.
- [33] P. Choomkasien, P. Chomphooyod, and D. Banjerdpongchai, "Design of model predictive control for industrial process with input time delay," in *Proceedings of the 2017 17th International Conference on Control, Automation and Systems (ICCAS)*, Jeju, Republic of Korea, October 2017.
- [34] N. Dong, Y. Feng, X. Han, and A. Wu, "An improved model-free adaptive predictive control algorithm for nonlinear systems with large time delay," in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, China, May 2018.
- [35] H. Norizuki and Y. Uchimura, "Design of model predictive control considering time delay in remote excavation work," in *Proceedings of the 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, Tokyo, Japan, March 2018.
- [36] C. Cheng, S. Jiang, J. Liu, and J. Lou, "Design and kinematics analysis of a novel MR-compatible robot for needle insertion," "Design and kinematics analysis of a novel MR-compatible robot for needle insertion," in *Advances in Reconfigurable Mechanisms and Robots I*, J. S. Dai, M. Zoppi, and X. Kong, Eds., pp. 443–453, Springer, London, UK, 2012.
- [37] S. Jiang, F. Sun, W. Feng, L. F. Hofman, and Y. Yu, "Analysis of a novel high-precision 5-degrees of freedom magnetic resonance imaging-compatible surgery robot for needle-insertion prostate brachytherapy," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 5, pp. 865–876, 2014.
- [38] E. G. Christoforou, I. Seimenis, E. Andreou, E. Eracleous, and N. V. Tsekos, "A novel, general-purpose, MR-compatible, manually actuated robotic manipulation system for minimally invasive interventions under direct MRI guidance," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 10, no. 1, pp. 22–34, 2014.
- [39] S. Guo, Y. Wang, N. Xiao, Y. Zhao, Y. Zeng, and J. Wu, "A novel master-slave robotic system with close loop control for vascular interventional surgery," in *Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, Tokyo, Japan, 2018.
- [40] J. Guo, Y. Liu, Q. Qiu et al., "A novel robotic guidance system with eye gaze tracking control for needle based interventions," *IEEE Transactions on Cognitive and Developmental Systems*, p. 1, 2019.
- [41] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82D, pp. 35–45, 1959.
- [42] Y. Shang and T. Lu, "Optimal tracking control of mini unmanned helicopter based iterative approximating," in *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2019.
- [43] R. Tanaka and T. Koga, "An approach to linear active disturbance rejection controller design with a linear quadratic regulator for a non-minimum phase system," in *Proceedings of the 2019 Chinese Control Conference (CCC)*, Gangzhou, China, 2019.
- [44] M. Dominici and R. Cortesão, "Model predictive control architectures with force feedback for robotic-assisted beating heart surgery," in *Proceedings of the 2014 IEEE International*

Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

- [45] R. Abe, T. Satoh, N. Saito, J. Nagase, and N. Saga, "Disturbance observer-based model predictive control using time-varying constraints," in *Proceedings of the 2015 World Congress on Industrial Control Systems Security (WCICSS)*, London, UK, 2015.
- [46] J. Jerez, E. Kerrigan, and G. Constantinides, "A condensed and sparse QP formulation for predictive control," in *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, 2011.

Research Article

A Novel Method to Evaluate Patient-Ventilator Synchrony during Mechanical Ventilation

Liming Hao,¹ Shuai Ren ,¹ Yan Shi,¹ Na Wang ,¹ Yixuan Wang ,¹ Zujin Luo ,² Fei Xie ,³ Meng Xu ,⁴ Jian Zhang,¹ and Maolin Cai¹

¹School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

²Department of Respiratory and Critical Care Medicine,

Beijing Engineering Research Center of Respiratory and Critical Care Medicine, Beijing Institute of Respiratory Medicine, Beijing Chao-Yang Hospital, Capital Medical University, Beijing 100043, China

³Department of Pulmonary and Critical Care Medicine, Chinese PLA General Hospital, Beijing 100039, China

⁴Department of Orthopedics, Chinese PLA General Hospital, Beijing 100039, China

Correspondence should be addressed to Na Wang; lion_na987@buaa.edu.cn, Zujin Luo; xmjb2002@163.com, Fei Xie; xiifei0522@163.com, and Meng Xu; 13501175839@163.com

Received 20 May 2020; Revised 12 June 2020; Accepted 17 June 2020; Published 15 September 2020

Guest Editor: Hang Su

Copyright © 2020 Liming Hao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The synchrony of patient-ventilator interaction affects the process of mechanical ventilation which is clinically applied for respiratory support. The occurrence of patient-ventilator asynchrony (PVA) not only increases the risk of ventilator complications but also affects the comfort of patients. To solve the problem of uncertain patient-ventilator interaction in the mechanical ventilation system, a novel method to evaluate patient-ventilator synchrony is proposed in this article. Firstly, a pneumatic model is established to simulate the mechanical ventilation system, which is verified to be accurate by the experiments. Then, the PVA phenomena are classified and detected based on the analysis of the ventilator waveforms. On this basis, a novel synchrony index (SI_{hao}) is established to evaluate the patient-ventilator synchrony. It not only solves the defects of previous evaluation indexes but also can be used as the response parameter in the future research of ventilator control algorithms. The accurate evaluation of patient-ventilator synchrony can be applied to the adjustment of clinical strategies and the pathological analyses of patients. This research can also reduce the burden on clinicians and help to realize the adaptive control of the mechanical ventilation and weaning process in the future.

1. Introduction

Nowadays, respiratory diseases have become some of the deadliest diseases in the world [1, 2]. Sudden global epidemic diseases, such as COVID-19, also bring great threat to people's life. In order to provide people with the necessary life support, the application and research of mechanical ventilation in clinical treatment are getting more and more attention from experts. Mechanical ventilation is an effective means of clinical treatment for respiratory failure and pneumonia and of respiratory support in critically ill patients. The indications for mechanical ventilation include acute respiratory distress syndrome (ARDS), chronic obstructive pulmonary diseases (COPD), pneumonia, severe

acute respiratory syndrome (SARS), middle east respiratory syndrome (MERS), and corona virus disease (COVID-19) [2–6]. In the treatment of the COVID-19 that has swept the world, mechanical ventilation has played an important role. During mechanical ventilation, patient-ventilator asynchrony (PVA) is a common problem [7]. The incidence of severe PVA can be as high as 25% in spontaneously breathing patients with acute respiratory failure during invasive ventilation [8, 9]. During noninvasive ventilation, PVA also maintains a high incidence [10]. Clinically, PVA could result in prolonged time on mechanical ventilation, increased use of sedation, complications of ventilator, and decrease in patient's comfort accompanied by the physiological and psychological changes [11–15]. For example,

during the treatment of COVID-19, PVA occurs in most patients, which increases the need for sedation and muscle relaxants [16, 17]. In addition, the respiratory gland is stimulated, which increases the secretion of sputum and causes sputum stasis, resulting in difficulty in suction [18–20]. Therefore, the research and clinical evaluation of patient-ventilator synchrony are vital for the process of mechanical ventilation.

In clinical practice, ventilator waveforms are often used to detect PVAs manually. Nilsestuen and Hargett summarized the method of using ventilator waveforms to identify PVAs [21]. The patient's pulmonary system and the ventilator are described as two pumps. The asynchrony between the two pumps can be identified by comparing the pressure, flow, and volume waveforms. However, they did not propose a standard index to evaluate the degrees of patient-ventilator synchrony. And the detection work is performed manually with lower intelligence degrees and real-time performance. Diaphragm electrical activity (EAdi) waveform can reflect patient's respiratory effort intuitively. Compared to conventional ventilator waveforms, EAdi can effectively monitor the interaction between the patient and the ventilator [22, 23]. However, the acquisition of the EAdi signal requires the placement of an esophageal electrode catheter, which is complicated in clinical practice and may interfere with mechanical ventilation. Therefore, it is not easy to popularize the application of synchrony index using EAdi clinically.

In addition, Sinderby et al. have proposed a neural index to quantify patient-ventilator interaction [24]. In their research, the patient-ventilator synchrony was evaluated by comparing ventilator pressure and EAdi waveforms. The two waveforms were analyzed for their timings, and the error between the waveforms was quantified, thus obtaining the synchrony index that is standardized and automated. However, this index still has certain limitations. In addition to the difficulty of measuring EAdi signals, this index lacks the synchrony evaluation on the space of ventilator waveforms. For example, amplitude fluctuations of pressure and flow waveforms caused by insufficient ventilation or over-ventilation cannot be effectively evaluated.

In this article, a novel method to evaluate patient-ventilator synchrony is proposed based on a pneumatic model of the mechanical ventilation system, by comparing changes in ventilator waveforms during the breathing cycles of patients. This method can automatically detect the main asynchrony phenomena considering the feasibility of clinical operation and the effectiveness of the evaluation. Different from the contribution of the research work completed by other scientists, the method proposed in this article does not require invasive operations and can comprehensively evaluate the patient-ventilator synchrony. The PVAs are classified according to different phases of a ventilation cycle. And the accuracy of the model and detection method is verified by a series of simulations. The synchrony index (SI_{hao}) is established in the form of a vector, and its elements correspond to various aspects of the asynchrony phenomena. This index intuitively reflects the detection and quantification results of the patient-ventilator synchrony. It is

significant for the diagnosis of the patient, the prediction of the development of patient's condition, and the clinical regulation of mechanical ventilation.

2. Materials and Methods

2.1. Pneumatic Model of the Mechanical Ventilation System.

The patient and the ventilator are driven by two different pneumatic systems. The patient's spontaneous breathing is driven by respiratory muscle, of which the work of diaphragm accounts for 60%~80% [25–27]. The ventilator is driven by an external air source or fan to generate pressure and flow to provide respiratory support to the patient. The interaction between the patient and the ventilator affects the changes of the ventilator waveform. A pneumatic model is established to simulate the process of mechanical ventilation, as shown in Figure 1.

In the pneumatic model, the vacuum pump simulates the generation of negative pressure when the patient inhales. The lungs can be regarded as a variable-volume container, whose elasticity corresponds to pulmonary compliance. The airway resistance is represented by the friction loss of the throttle valve. The brake valve simulates the switching between inspiration and expiration. The ventilator pneumatic system is established based on the working principle of a medical ventilator that can achieve pressure support ventilation (PSV). These two pneumatic systems are connected by the ventilation tube whose compliance and air resistance can be negligible ideally.

The working principle of the pneumatic model corresponds to the physiological principle of the patient's respiratory system. During a normal breathing cycle, the negative pressure is first generated inside the pleural cavity, to bring a pressure difference between the inside and outside of the human body. Then the air gets into the lungs through the airway. This process can be completed by the vacuum pump and the brake valve in the pneumatic model. When the patient starts exhaling, the chest wall retracts, expelling excess air in the lungs. This process can be completed by the variable-volume container which is elastic. The breathing support of the mechanical ventilation system is provided by the ventilator pneumatic system as shown in Figure 1. When the patient's compliance changes, the elasticity of the variable-volume container changes accordingly, and their change rules are inversely proportional. When the patient's airflow resistance changes, the effective sectional area of the throttle valve changes accordingly, and their change rules are proportional. Therefore, the pneumatic model in this article can effectively simulate the changes and effects of respiratory mechanical parameters in the mechanical ventilation system.

2.2. Mathematical Model of the Mechanical Ventilation System

2.2.1. Driving Pressure of Patient's Spontaneous Breathing.

Patient's spontaneous breathing is driven by the contraction of respiratory muscles. The formula of pressure changes in the lungs produced by respiratory muscles can be fitted from

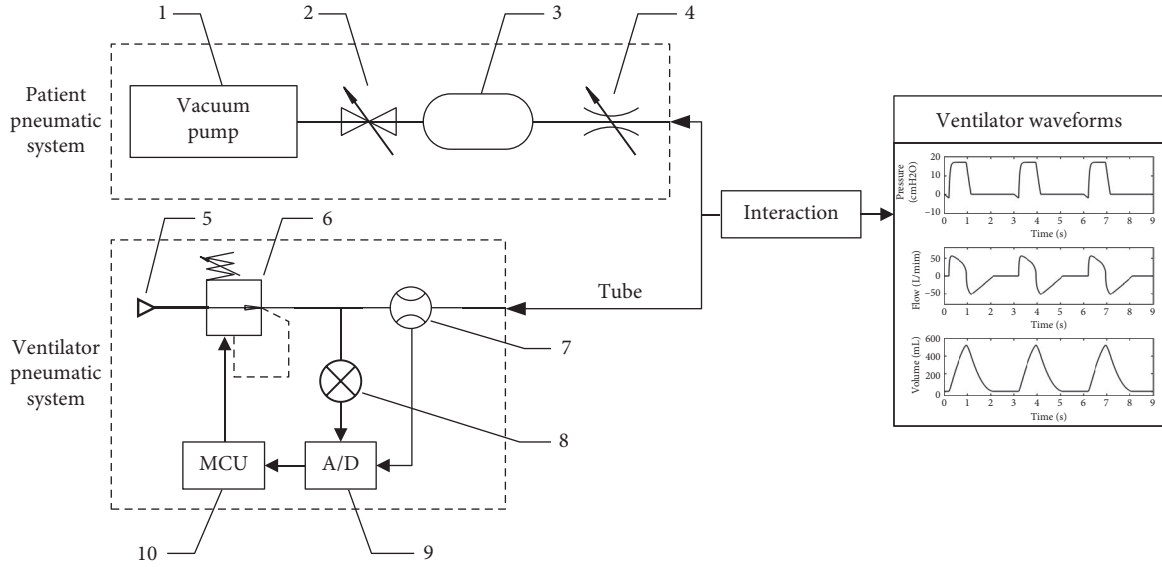


FIGURE 1: Principle diagram of the pneumatic model of the mechanical ventilation system. 1: vacuum pump, 2: brake valve, 3: variable-volume container, 4: throttle valve, 5: air supply resource, 6: control valve, 7: flow sensor, 8: pressure sensor, 9: data acquisition module, 10: microprogrammed control unit.

the pressure curve produced by the lung simulator (ASL5000), and the simulator curve can accurately represent the curve of pressure changes generated by the patient's spontaneous breathing [28–30]. In this article, the value of this pressure changes is denoted as P_{mus} . In previous research, both Vicario and Hao have revealed this law [30, 31],

and P_{mus} can be expressed as a piecewise function, as shown in equation (1). In this formula, $P_{mus,max}$ represents the maximum value of the pressure change, t_{dec} is the time for pressure dropping, t_{inc} is the start of the pressure rising, t_{ins} is the termination of the pressure rising, and T_{cycle} is the time for a breathing cycle.

$$P_{mus} = \begin{cases} -P_{mus,max} \cdot \sin\left(\frac{\pi t}{2 \cdot t_{dec}}\right), & 0 \leq t < t_{dec}, \\ -P_{mus,max}, & t_{dec} \leq t < t_{inc}, \\ -P_{mus,max} \cdot \sin\left(\frac{\pi(t + t_{ins} - 2 \cdot t_{inc})}{2(t_{ins} - t_{inc})}\right), & t_{inc} \leq t < t_{ins}, \\ 0, & t_{ins} \leq t < T_{cycle}. \end{cases} \quad (1)$$

The pressure curve (see Figure 2) can be simulated according to equation (1). The red dotted line was obtained by the data acquisition of the lung simulator (ASL5000), and the black solid line is the fitted curve based on equation (1).

2.2.2. Operating Pressure of the Ventilator. For the PSV mode, the operating pressure of the ventilator can be expressed as a piecewise function, as shown in equation (2) [29–31]. In the formula, P_V is the operating pressure of the

ventilator, $P_{trigger}$ is the trigger pressure, P_{peak} is the plateau pressure, P_{PSV} is the maximum support pressure set by the ventilator, $t_{trigger}$ is the trigger time, t_{rise} is the rise time from $P_{trigger}$ to P_{peak} (the value of P_{peak} is approximately equal to $(P_{PSV} + P_{trigger} + PEEP)$), t_{ter} is the termination time of inhalation, and t_{drop} is the drop time from P_{peak} to PEEP. When the airway pressure drops more than the trigger pressure, the ventilator is triggered and starts working. The value of t_{rise} is preset by the clinician and t_{ter} is based on a percentage of peak flow. This percentage is preset or varies

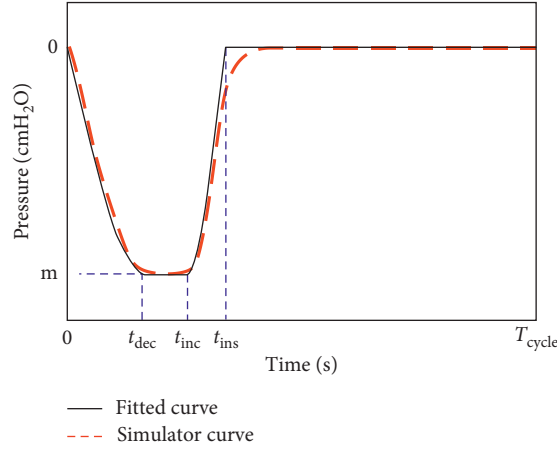


FIGURE 2: Curve of the pressure changes produced by respiratory muscle. The point m represents $P_{\text{mus,max}}$; the black solid line represents the fitted curve; and the red dotted line represents the pressure changes curve produced by the lung simulator.

depending on the algorithm of the ventilator. PEEP means positive end expiratory pressure, which maintains a certain pressure in the airway at the end of expiration.

$$P_v = \begin{cases} (P_{\text{PSV}} + P_{\text{trigger}}) \cdot \left(1 - e^{-(t-t_{\text{trigger}}/t_{\text{rise}})}\right) + \text{PEEP}, & t_{\text{trigger}} \leq t < (t_{\text{trigger}} + t_{\text{rise}}), \\ P_{\text{peak}}, & (t_{\text{trigger}} + t_{\text{rise}}) \leq t < t_{\text{ter}}, \\ P_{\text{peak}} - (P_{\text{peak}} - \text{PEEP}) \cdot \frac{(t - t_{\text{ter}})}{t_{\text{drop}}}, & t_{\text{ter}} \leq t < (t_{\text{ter}} + t_{\text{drop}}), \\ \text{PEEP}, & (t_{\text{ter}} + t_{\text{drop}}) \leq t < T_{\text{cycle}}. \end{cases} \quad (2)$$

2.2.3. Differential Equations and Transformation Relations of Physical Quantities. During the mechanical ventilation, the pneumatic model can be regarded as an isothermal system [32, 33]. By deforming the ideal gas equation, the differential equation for intrapulmonary pressure can be obtained, as shown in equation (3) [31, 34–36]. In the formula, P_{pl} means the intrapulmonary pressure (pa), ρ means the air density (1.293 kg/m^3), R means the gas constant ($297 \text{ J/(kg}\cdot\text{K)}$), T means the temperature (K), Q means the volumetric flow rate, V means the lung volume, C means the pulmonary compliance, and m means the air mass.

$$\frac{dP_{\text{pl}}}{dt} = \frac{\rho RT Q V}{V^2 + CmRT}. \quad (3)$$

In the pneumatic model, the ratio of downstream pressure (P_d) to upstream pressure (P_u) of the throttle valve is bigger than 0.528, which means the airflow is subsonic [31, 34–36]. The equation of volumetric flow rate can be described as follows:

$$Q = \frac{nA_e P_u \sqrt{1-r}}{\rho \sqrt{RT}} \sqrt{1 - \left(\frac{(P_d/P_u) - r}{1-r}\right)^2}, \quad (4)$$

where n is the flow coefficient (equal to 1 during inhalation; equal to -1 during exhalation), A_e is the effective sectional area of the throttle valve, and r is critical pressure ratio (0.528).

According to the definition of respiratory compliance, the differential equation of lung volume can be described as follows [31, 34–36]:

$$dV = C \cdot dP_{\text{pl}}. \quad (5)$$

During the mechanical ventilation, the difference between the operating pressure of the ventilator and the patient's intrapulmonary pressure produces inspiratory and expiratory airflows. The mathematical model as shown in equations (3)–(5) illustrates the correlation between the pressure and flow in the pneumatic model. During the clinical process, the pressure and flow in the mechanical system are affected by the respiratory mechanical parameters. The compliance and the airflow resistance are regarded as the main respiratory mechanical parameters [31]. The compliance reflects pulmonary ability to deform. The higher compliance results in the greater change in the lung volume at unit pressure, which is described as equation (5). The airflow resistance affects the volumetric flow rate. The

greater resistance results in the smaller volumetric flow rate. In the pneumatic model proposed in this article, the effect of the airflow resistance is equivalent to a throttle valve, as shown in Figure 1. Therefore, the effective sectional area of the throttle valve (A_e) is directly proportional to the airflow resistance. In equation (4), the effective sectional area of the throttle valve (A_e) is used to describe the effect of airflow resistance on the volumetric flow rate. According to the mathematical model, the simulation calculation can be carried out. In the calculation, the units of each physical quantity are the basic units of the international system of units. In the reports of clinical diagnosis and treatment or the display of parameters on the ventilator, the units of each physical quantity are converted into the units commonly used in the clinic. The corresponding units of each physical quantity are shown in Table 1.

2.2.4. Clinical Operation Principle and Simulation Parameter Settings. The clinician needs to make some presets on the ventilator according to the physical condition of the patient before the operation of mechanical ventilation. The clinical operation process of the mechanical ventilation is shown in Figure 3. The ventilator waveforms are affected by the interaction between the patient and the ventilator. Therefore, unreasonable presets, changes in the patient's physical condition, and changes in the clinical environment can all lead to the occurrence of PVA, thus producing specific changes in the ventilator waveforms.

In this article, the PSV mode is the main ventilation mode for the research of patient-ventilator synchrony. This common ventilation mode has a wide range of applications and is involved in invasive and noninvasive ventilation and in weaning process. Based on the mathematical model of the pneumatic system of the patient and the ventilator, the mechanical ventilation process can be simulated. According to the parameter preset of clinical operation, the settings of simulation parameters are shown in Table 2 [30, 31, 37].

The ventilator waveforms are simulated as shown in Figure 4. The green dotted lines indicate the successful triggering of the breathing cycle. The red dotted lines indicate the termination of the inspiratory phase and the beginning of the expiratory phase.

In order to verify the accuracy of the model, an experimental prototype was established to obtain the experimental data to be compared with the simulation data. The experimental prototype consists of a lung simulator (ASL5000), a ventilator (Philips Respironics BiPAP ST30), and a PC, as shown in Figure 5. The lung simulator can simulate the natural respiratory process of the patients, which has been verified to be accurately fitted in clinical medicine [28–30]. The PC is the control center of the lung simulator, which can adjust the respiratory mechanical parameters to simulate the pathological difference in the patients. The ventilator is connected to the airway of the lung simulator to provide breathing support. By setting the ventilator to the PSV mode, the mechanical ventilation experiments in this article can be performed. The regulation of the ventilator parameters is regarded as the clinician's

TABLE 1: The units of basic physical quantity.

Physical quantity	Calculation unit	Reported unit
Time	s	s
Pressure	Pa	cmH ₂ O
Volume	m ³	cmH ₂ O/L/s
Compliance	m ³ /Pa	mL/cmH ₂ O
Mass	kg	—
Effective area	m ²	mm ²

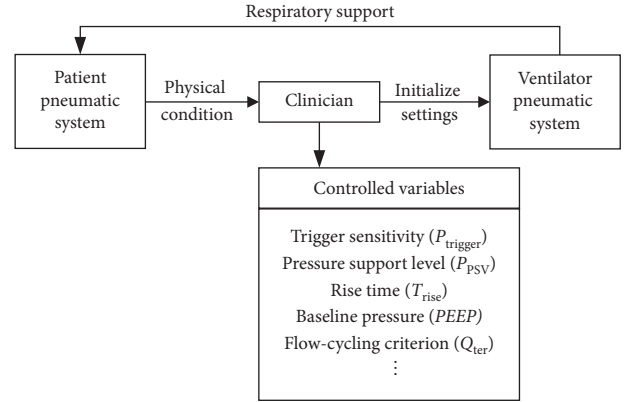


FIGURE 3: Clinical operation process of the mechanical ventilation.

TABLE 2: The settings of simulation parameters.

Physical quantity	Value	Unit
P_{PSV}	17	cmH ₂ O
$P_{trigger}$	2	cmH ₂ O
$PEEP$	5	cmH ₂ O
t_{rise}	0.2	s
$*k_{ter}$	10%	
A	15	mm
C	50	mL/cmH ₂ O
$P_{mus,max}$	5	cmH ₂ O
t_{dec}	0.5	s
t_{inc}	0.75	s
t_{ins}	1	s
T_{cycle}	3	s

* k_{ter} represents the ratio of flow rate to peak flow rate at the end of inspiration, which determines the value of t_{ter} .

operation process. The clinical operation process of the mechanical ventilation is illustrated in Figure 3. The goodness of fit (R^2) between the experimental curve and the simulation curve is calculated by equation (6). In the formula, \bar{d}_i means the value of the simulation data, \bar{d} means the average value of the experimental data, d_i means the value of the experimental data, and m means the number of samples. In Figure 4, R^2 values of the pressure curve, the flow curve, and the volume curve are 0.9423, 0.9378, and 0.9628. The values of R^2 verify the good fitting between the experimental curve and the simulation curve, thus verifying the validity of the pneumatic model and the accuracy of the mathematical model.

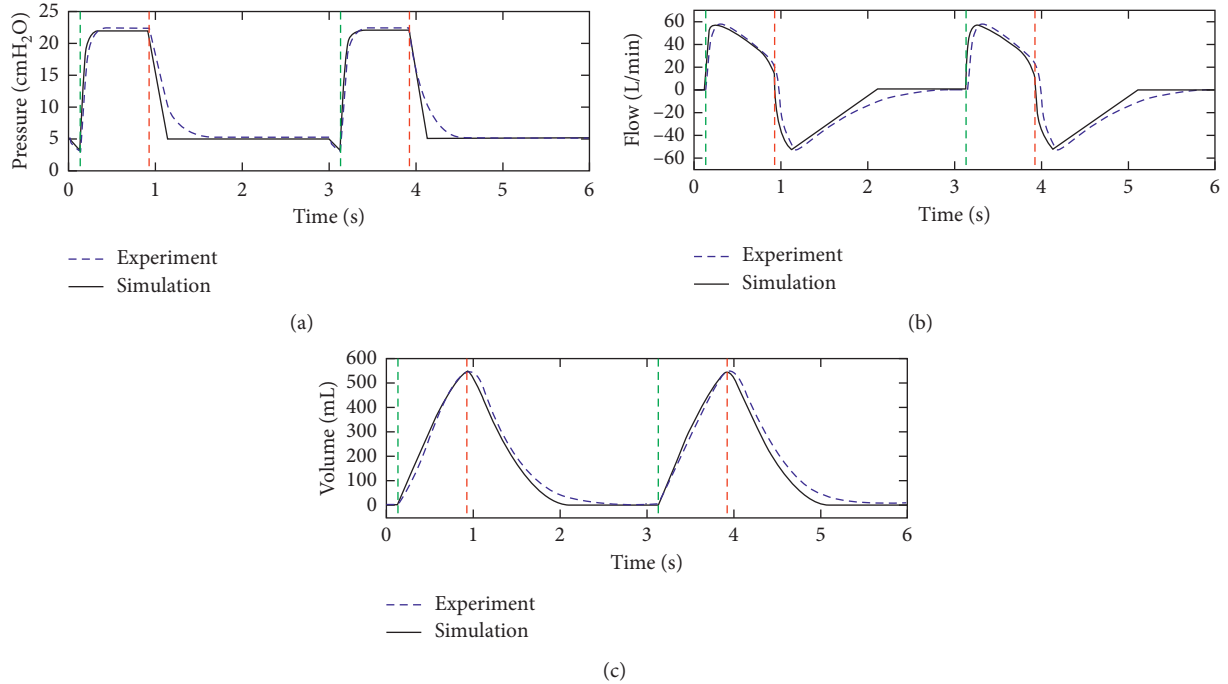


FIGURE 4: The ventilator waveforms during pressure support ventilation. The waveforms include the pressure curve, the flow curve, and the volume curve.

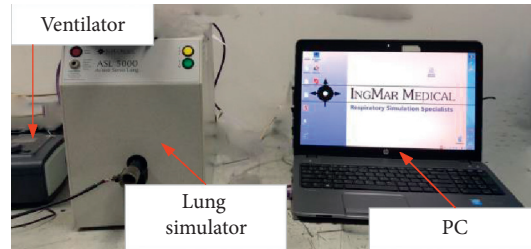


FIGURE 5: Experimental prototype. The experimental prototype consists of a lung simulator, a ventilator, and a PC.

$$R^2 = \frac{\sum_{i=1}^m (\hat{d}_i - \bar{d})^2}{\sum_{i=1}^m (d_i - \bar{d})^2}. \quad (6)$$

2.3. Classification and Detection of PVA. Considering the process of a typical breath, PVA usually occurs in four phases of each breathing cycle. According to these four phases, PVA can be divided into triggering asynchrony, inspiration asynchrony, switching asynchrony, and exhalation asynchrony. These phenomena can be detected and classified by observing and comparing pressure and flow waveforms. In this part, several main PVA phenomena in PSV ventilation mode are classified and reproduced through simulation, and the automatic detection method of PVA is described.

2.3.1. Triggering Asynchrony

(1) Ineffective Triggering. During clinical mechanical ventilation, the ventilator is usually triggered by the pressure or

flow, which depends on the type of ventilation mode. The effort to breathe results in the variation of the pressure or flow in the airway. When the variation reaches the preset triggering level, the ventilator will be triggered and work. In this article, the ventilator works in PSV mode. Therefore, the triggering pressure is regarded as the trigger signal. The value is set to 2 cmH₂O as shown in Table 2. It is designed according to the general clinical standard in pressure support ventilation [30, 31, 37].

When the effort of patient's respiratory muscle cannot reach the triggering level, the ventilator fails to be triggered and will not work. This phenomenon is called ineffective triggering. Usually, this phenomenon is caused by the sensitivity setting of the ventilator being too low (the trigger level is set too high) or the patient's pathological changes (such as excessive airway resistance). As shown in Figure 6, the red arrows point to the occurrence of the ineffective triggering, and the blue dotted line indicates the start of ineffective effort. In the ventilator waveforms, the pressure is slightly reduced into a concave shape, corresponding to the short-term rise of the flow waveform, which constitutes a

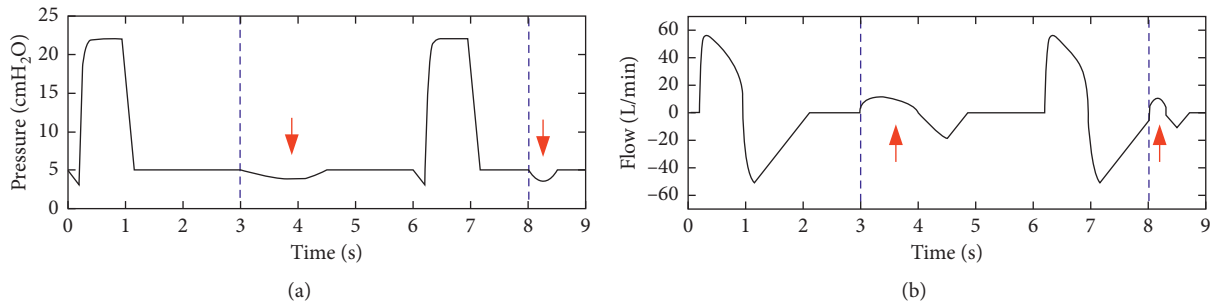


FIGURE 6: Ventilator waveforms of ineffective triggering.

short-term inspiratory flow. This reflects the patient's efforts to inhale.

It can be seen from Figure 6 that the ineffective triggering may also occur in the expiratory phase. This is usually caused by the patient's sudden inspiratory demand or a slight leak in the ventilation system.

(2) *Double Triggering*. When the patient's breathing demand suddenly increases, the patient produces a second inspiratory effort in the same respiratory cycle and successfully triggers the ventilator, which will produce a double triggering asynchrony. As shown in Figure 7, the red arrows point to the occurrence of double triggering. It can be seen from the curve that when double triggering occurs, the exhalation time is less than half of the inhalation time, which can be regarded as an automatic detection standard. The peak flow of the second triggered cycle is often higher than the average peak flow, as shown by the red dotted line in Figure 7.

In addition, when premature switching occurs, if the patient's continuous breathing effort is intense, a second respiratory cycle may be triggered, causing the double triggering. A cough may also trigger multiple respiratory cycles within a short period of time, but this is not considered to be double triggering.

(3) *Autotriggering*. In the mechanical ventilation system, there may be leakage in the mask, ventilation pipeline, pipeline interface, etc., which is especially common in noninvasive ventilation. When the leakage reaches the trigger level, the ventilator is triggered to work, causing the autotriggering asynchrony. As shown in Figure 8, the exhalation time of the autotriggering cycle is less than the average value. Different from the double triggering, the peak flow of the second cycle produced by the autotriggering is lower than the average level. This can be regarded as the detection criteria.

2.3.2. *Inspiration Asynchrony*. For the PSV mode, the inspiration asynchrony usually occurs during the pressure rise phase. The pressure rise time (t_{rise}) depends on the valve-opening rate of the ventilator. The value of t_{rise} can be preset by the clinician. If the rise time is too slow, the time to reach the effective ventilation level will be significantly delayed, and the ventilation flow will not reach the average level, resulting in insufficient tidal volume, as shown by the red arrow and red dotted line in Figure 9.

Conversely, if the pressure rise time is too fast, there will be a spike at the initial position of the inspiratory phase of the plateau pressure of the pressure-time waveform. Correspondingly, there will be an overshoot at the initial position of the inspiratory flow. By appropriately reducing the valve-opening rate or increasing the setting value of t_{rise} , this phenomenon can be effectively improved.

2.3.3. Switching Asynchrony

(1) *Delayed Switching*. The accurate switching between inhalation and exhalation of mechanical ventilation affects the patient's comfort and the effect of mechanical ventilation. If the ventilator is still supplying air after the termination of the patient's inhalation, there will be a confrontation between the patient and the ventilator. When the confrontation occurs, the patient's expiratory efforts produce changes of the intrapulmonary pressure, which makes the airway pressure suddenly rise at the end of the plateau pressure. Corresponding to the pressure, the patient's inspiratory flow suddenly drops at the end of the inspiratory phase. In Figure 10, the red dotted line and the red arrow indicate the characteristic performance of the delayed switching.

(2) *Premature Switching*. For PSV ventilation mode, the switching time depends on the percentage of the peak flow, that is, the termination level of the inspiratory phase. If the termination level is set too high, the ventilator supply will be terminated prematurely. At the end of the ventilator supply, the patient has not yet completed the inhalation, and the inspiratory effort causes a brief reversal of the ventilator waveforms. As shown in Figure 11, the beginning of the expiratory phase of the flow waveform suddenly rebounds toward the baseline. Corresponding to the flow waveform, the pressure suddenly drops at the beginning of the expiratory phase and then rises again, showing a concave shape. The red dotted line and the red arrow indicate the characteristic performance of the delayed switching, and the blue dotted line shows the simulated waveform without confrontation. If the premature switching asynchrony is serious, it will cause double triggering.

2.3.4. *Exhalation Asynchrony*. Exhalation asynchrony is usually caused by too short or prolonged exhalation time. Premature switching will cause the prolonged exhalation

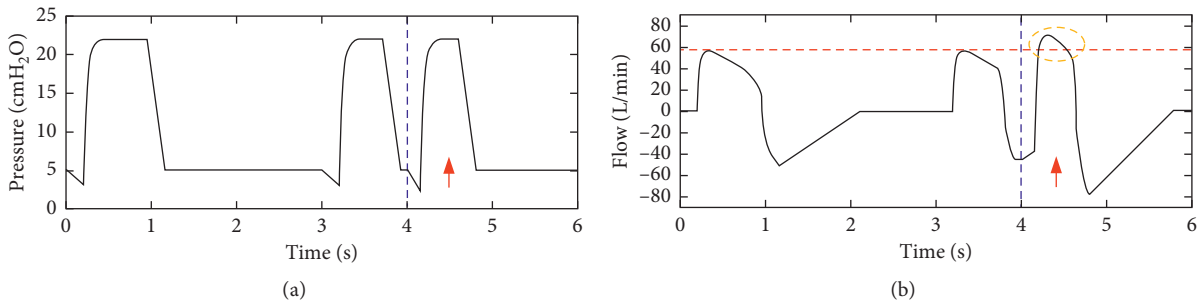


FIGURE 7: Ventilator waveforms of double triggering.

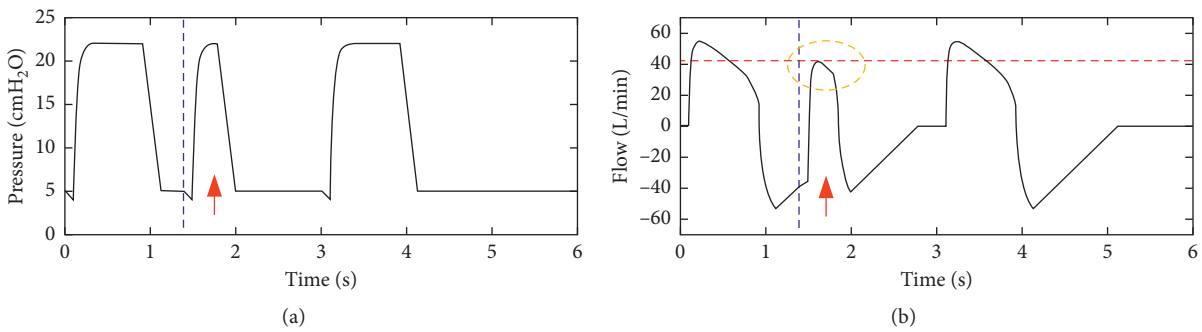


FIGURE 8: Ventilator waveforms of autotriggering.

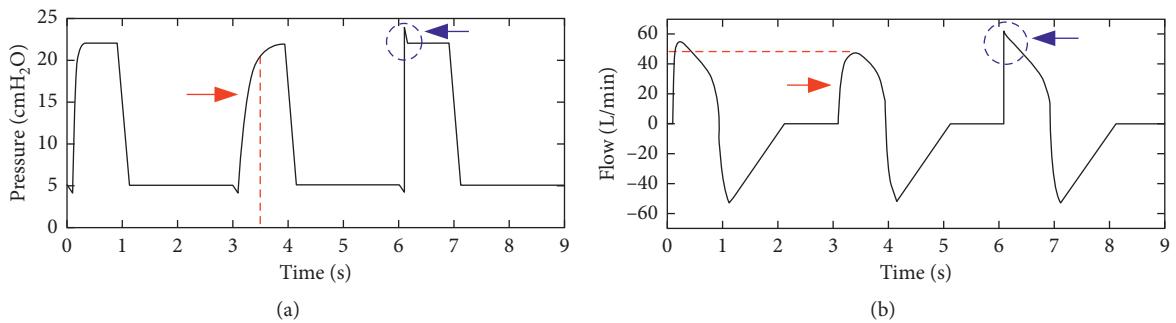


FIGURE 9: Ventilator waveforms of inspiration asynchrony.

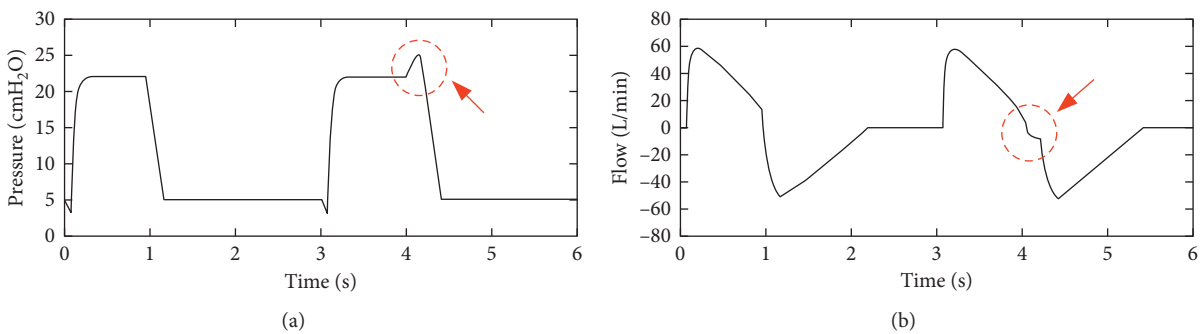


FIGURE 10: Ventilator waveforms of delayed switching.

time. If the exhalation time is too long, it will cause hypoventilation. Usually, prolonged exhalation is of little consequence to the subsequent cycles. However, shorten

exhalation usually has the risk of causing endogenous positive end expiratory pressure (PEEPi). The presence of PEEPi often causes ineffective triggering of the subsequent

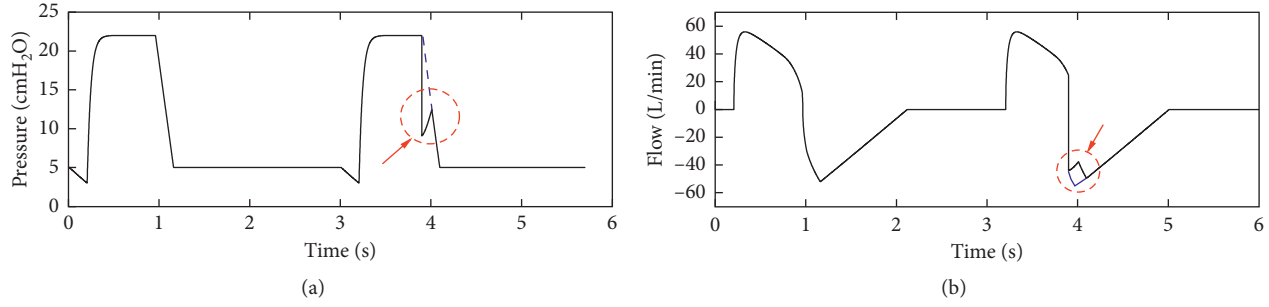


FIGURE 11: Ventilator waveforms of premature switching.

cycles, as the patient has to overcome the additional positive end expiratory pressure to reach the trigger level. The presence of PEEP_i can be detected by observing whether the inspiratory flow can return to baseline before the next breathing cycle. As shown in Figure 12, the red dotted line and red arrows point to the performance of the exhalation asynchrony. At the end of the expiratory phase, there is a certain difference between the intrapulmonary pressure and the airway pressure.

In addition, PEEP_i is closely related to pathological changes in patients. When patients have pathological changes, such as chronic bronchitis, chronic obstructive pulmonary disease (COPD), and other diseases, which lead to structural destruction of the bronchial wall, thickening of the tube wall, stenosis of the lumen, and significant increase in expiratory resistance, the isobaric point is shifted inward. The expiratory process is not completed, but the bronchial collapse occurs, and positive pressure is formed in the alveoli.

2.4. Definition of Patient-Ventilator Synchrony Index. In the pressure support ventilation mode, the clinical regulation of mechanical ventilation is achieved through the control and adjustment of ventilator pressure. From the classification of PVA, it can be concluded that PVA usually occurs at different phases of each breathing cycle. Therefore, to evaluate patient-ventilator synchrony during mechanical ventilation, each breathing cycle is divided into four phases for analysis: (1) triggering phase; (2) rising phase; (3) inspiratory phase; and (4) expiratory phase. These four phases are determined based on the airway pressure waveform of the ventilator. As shown in Figure 13, point a is the triggering phase, the interval from point a to point b is the rising phase, the interval from point b to point c is the inspiratory phase, and the interval from point c to point d is the expiratory phase. ΔP_{ins} represents the difference between the abnormal pressure value and the normal pressure waveform.

Corresponding to the performance of patient-ventilator synchrony at different phases, a novel evaluation index of patient-ventilator synchrony, SI_{hao} , is proposed in this article.

$$SI_{hao} = (SI_{tri}, SI_{timing}, SI_a, SI_{peepi}). \quad (7)$$

SI_{hao} is a vector, whose elements can be used as indicators for different problems of patient-ventilator

synchrony. The set of SI_{tri} is $\{-1, 0, 1, 2\}$. In the case of normal ventilation, the value of SI_{tri} is 1. Different values represent different triggering asynchrony phenomena: 0 represents ineffective triggering; 2 represents double triggering; and -1 represents autotriggering. When a triggering problem occurs, the ventilator control parameters must be adjusted in time. Therefore, SI_{tri} can not only evaluate the triggering synchrony over time but also serve as an alarm for triggering asynchrony.

$$SI_{timing} = \frac{t_{effective}}{T_{cycle}} \times 100\%. \quad (8)$$

As shown in equation (8), the value of SI_{timing} is equal to the ratio of $t_{effective}$ and T_{cycle} . When the airway pressure rises to 95% of P_{PSV} , it can be considered that the ventilator pressure has reached an effective support level. If the pressure rise time of the ventilator is set too long, or the trigger level is set too high to trigger the ventilator immediately, the value of $t_{effective}$ will increase significantly, delaying the effective ventilation and increasing the patient's work of breathing. Therefore, when the value of SI_{timing} exceeds the normal threshold value, it may correspond to the inspiration asynchrony caused by the excessively slow rise time, and the peak flow and tidal volume will significantly reduce. At the same time, the change of SI_{timing} can also reflect the triggering synchrony, as the increase of SI_{timing} reflects the delay of the trigger of the ventilator. Thus, when SI_{timing} is abnormal, the trigger sensitivity can be fine-tuned if inspiration asynchrony is not detected.

As shown in Figure 14, ΔP_{ins} represents the difference between the abnormal pressure value and the normal pressure waveform. When the delay switching occurs, there will be an overshoot at the end of the plateau pressure, and the value of ΔP_{ins} is positive. When the premature switching occurs, there will be a sudden drop during the transition between the inspiratory phase and the expiratory phase, showing a concave shape on the pressure curve, and the value of ΔP_{ins} is negative. Therefore, ΔP_{ins} can be used as the characteristic quantity for synchrony evaluation.

$$SI_a = \frac{\Delta P_{ins}}{P_{PSV}} \times 100\%. \quad (9)$$

The value of SI_a , as shown in equation (9), reflects the patient-ventilator synchrony in amplitude of ventilator waveform. It corresponds to some phenomena of the

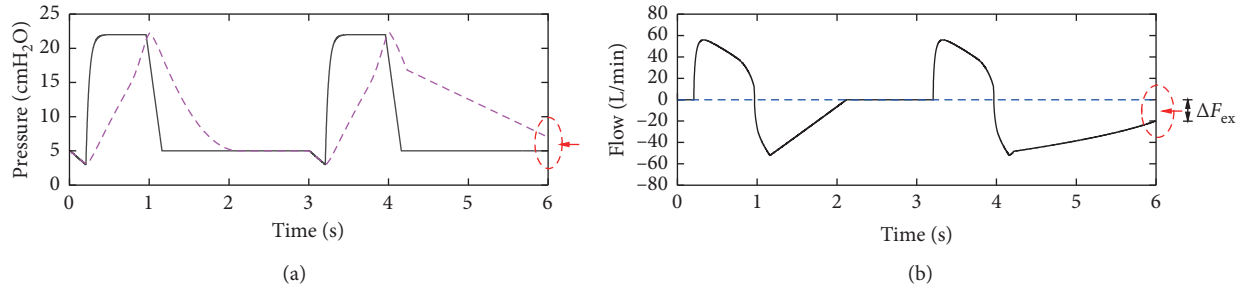


FIGURE 12: Ventilator waveforms of breathing cycles with the presence of PEEPi. The pink dotted line represents the curve of the intrapulmonary pressure, and the blue dotted line represents the baseline of the flow curve.

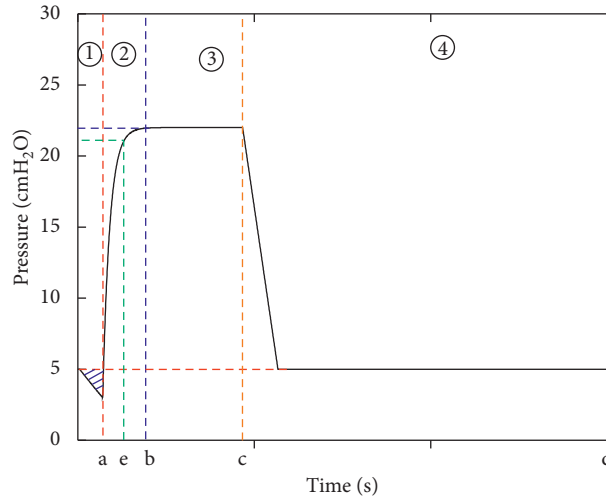


FIGURE 13: Four phases of a breathing cycle. a: $t_{trigger}$, the trigger time; b: t_{rise} , the rise time; c: t_{ter} , the termination of inhalation; d: T_{cycle} , the termination of a breathing cycle; e: $t_{effective}$, the time to reach effective ventilation level (95% of the P_{PSV}).

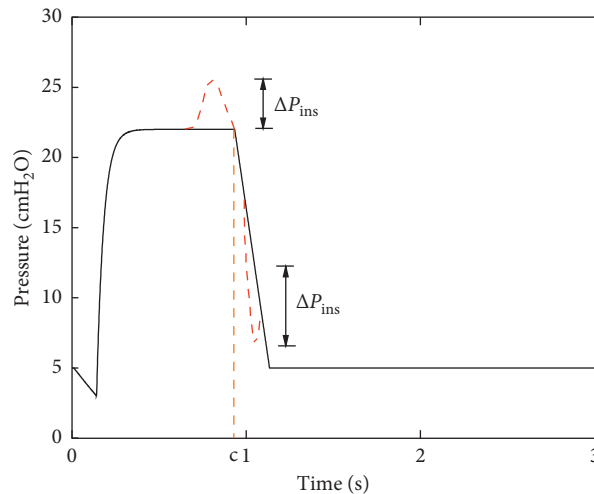


FIGURE 14: The characteristics of PVA in the amplitude of pressure waveform.

inspiration asynchrony and switching asynchrony. When the switching asynchrony occurs, or during the normal ventilation, the value range of SI_a is $(-1, 1)$. And the closer the value of SI_a is to zero, the better the synchrony is.

However, when the rise time of the pressure is too short, there will be a short-term spike at the beginning of the plateau pressure, as shown in Figure 15. In this case, the value of ΔP_{ins} is also positive. In order to distinguish it from

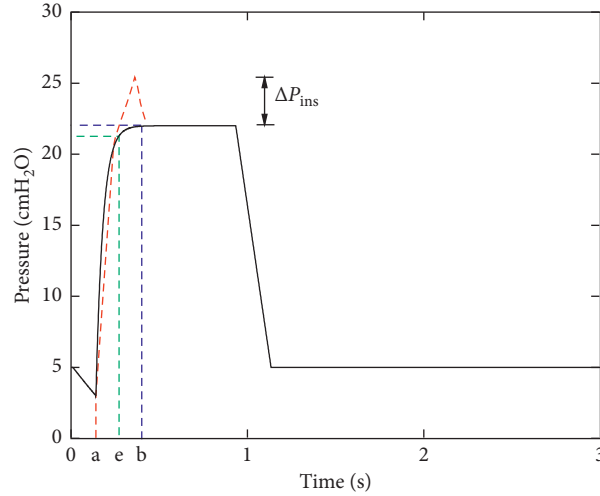


FIGURE 15: The illustration of the short-term spike during inspiration asynchrony.

delayed switching, when the above phenomenon occurs, the calculation of SI_a adds 1 to the result of equation (9), and the value range becomes (1, 2).

SI_{peepi} corresponds to the phenomenon of exhalation asynchrony, which is mainly manifested as the increase of endogenous positive end expiratory pressure (PEEPi). The value of PEEPi is closely related to pathological changes in patients and PVA caused by improper mechanical ventilation, so continuous monitoring of PEEPi is required during ventilation. However, the measurement of the intrapulmonary pressure imposes high requirements on the equipment and is not easy to achieve. Therefore, the judgement of the occurrence and increase of PEEPi is achieved through the detection of expiratory flow instead of the detection of intrapulmonary pressure. When the PEEP provided by the ventilator cannot offset PEEPi, the expiratory flow cannot return to zero before the next breathing cycle, as shown in Figure 16.

$$SI_{peepi} = \frac{\Delta F_{ex}}{F_{exmax}} \times 100\%. \quad (10)$$

In equation (10), the value of ΔF_{ex} represents the difference between the end expiratory flow and zero, as shown in Figure 16. And the value of F_{exmax} represents the absolute value of the peak flow during the expiratory phase. Therefore, the higher value of SI_{peepi} means the more serious phenomenon of exhalation asynchrony, and the value of PEEP need to be increased to offset the influence of PEEPi. During normal ventilation, the value of SI_{peepi} is equal to 0.

3. Results and Discussion

Through the definition of SI_{hao} , it can be concluded that the evaluation index of patient-ventilator synchrony proposed in this paper is a vector index, and its four elements correspond to different aspects of PVA phenomena.

In the past, the asynchrony index was usually used as the evaluation index of patient-ventilator synchrony. The asynchrony index proposed by Thille is defined as follows [8]:

$$AI_{Thille} = \frac{\text{Number of asynchrony phenomena}}{\text{Total breathing cycles (ventilator cycles + ineffective efforts)}} \times 100\%, \quad (11)$$

which counts the number of asynchrony phenomena within a certain period, such as 30 breathing cycles. This asynchrony index cannot analyze the synchrony of a single breathing cycle. Besides, Sinderby et al. proposed a neural index to automatically evaluate the patient-ventilator interaction. This index can reflect the timing synchrony and triggering synchrony between the patient's spontaneous breathing and ventilator ventilation. However, the

measurement requirement of EAdi makes it more difficult to obtain this index. Besides, it cannot reflect the PVA caused by the improper rise time of ventilator pressure and the increase of PEEPi, which makes the index have certain defects when evaluating the patient-ventilator synchrony.

Different from the traditional evaluation index, SI_{hao} can realize the single-period judgement, with a better real-time performance. As a vector index, it covers several major

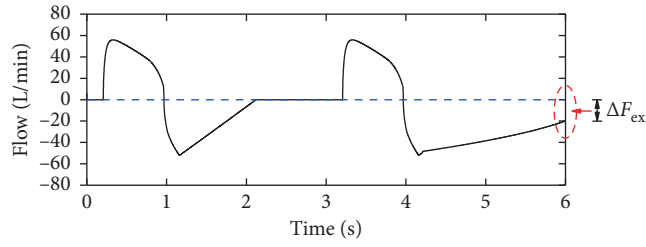


FIGURE 16: The effect of PEEPi on expiratory flow.

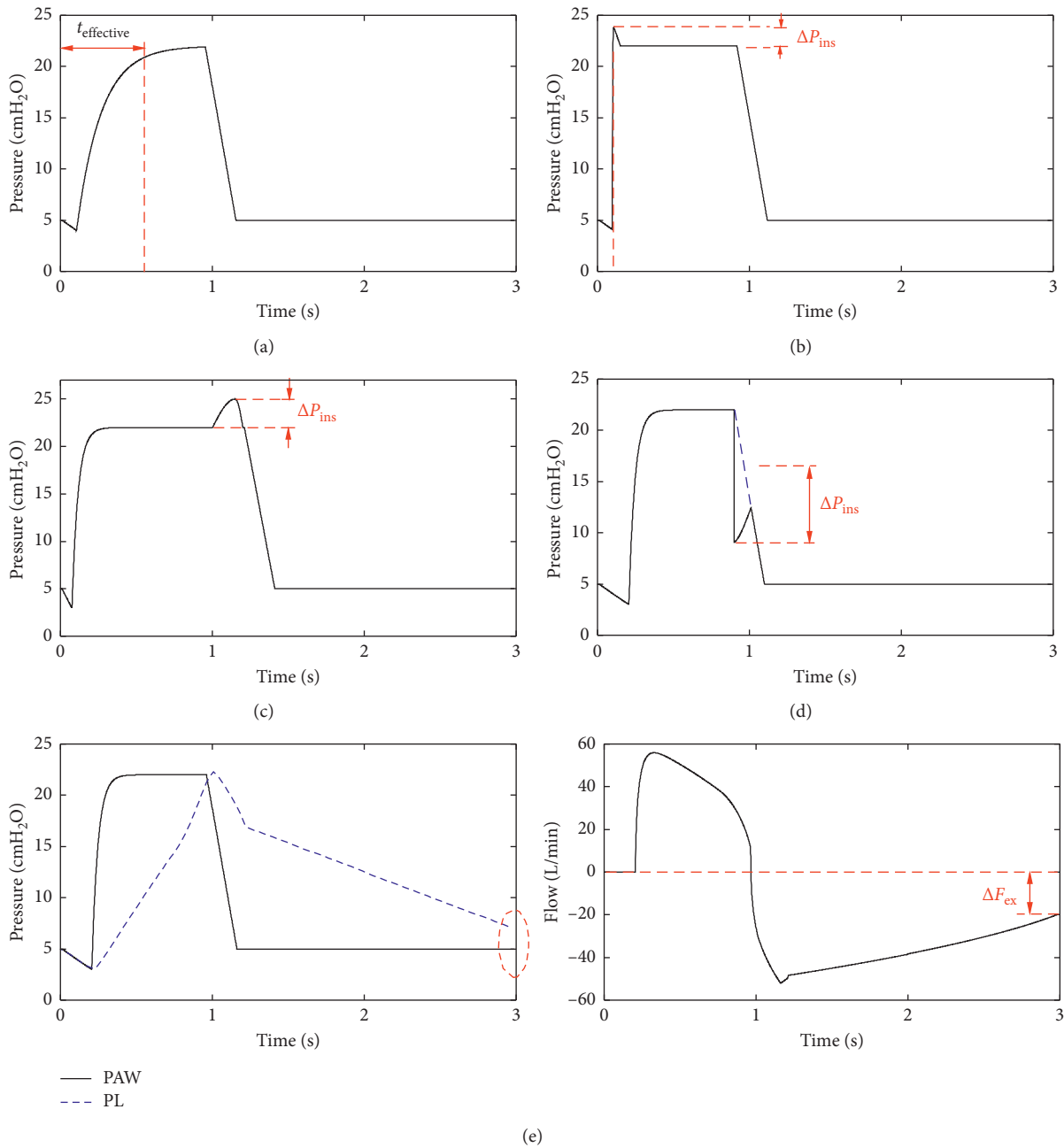


FIGURE 17: Evaluating patient-ventilator synchrony of the breathing cycles with asynchrony phenomena.

asynchrony phenomena under PSV mode, which makes this index more convincing. Taking several different breathing cycles shown in Figure 17 as examples, the determination of corresponding asynchrony phenomena and SI_{hao} is shown as follows:

(a) Inspiration asynchrony (longer rise time)

$$t_{\text{effective}} = 0.53 \text{ s}$$

$$SI_{hao} = (1, 17.67\%, 0, 0)$$

(b) Inspiration asynchrony (shorter rise time)

$$t_{\text{effective}} = 0.14 \text{ s}$$

$$\Delta P_{\text{ins}} = 1.8 \text{ cmH}_2\text{O}$$

$$SI_{hao} = (1, 4.67\%, 1.11, 0)$$

(c) Switching asynchrony (delayed switching)

$$t_{\text{effective}} = 0.32 \text{ s}$$

$$\Delta P_{\text{ins}} = 3 \text{ cmH}_2\text{O}$$

$$SI_{hao} = (1, 10.67\%, 17.65\%, 0)$$

(d) Switching asynchrony (premature switching)

$$t_{\text{effective}} = 0.32 \text{ s}$$

$$\Delta P_{\text{ins}} = -10 \text{ cmH}_2\text{O}$$

$$SI_{hao} = (1, 10.67\%, -58.82\%, 0)$$

(e) Exhalation asynchrony

$$t_{\text{effective}} = 0.32 \text{ s}$$

$$\Delta F_{\text{ex}} = 20.3 \text{ L/min}$$

$$SI_{hao} = (1, 10.67\%, 0, 36.25\%)$$

To define the pros and cons of the patient-ventilator synchrony, it is necessary to set a certain threshold value for SI_{hao} . The setting of the threshold depends on the detection accuracy of each parameter and the normal variation range. According to the clinical experience of physicians [30, 38–40], P_{trigger} is usually set to 1 to 2 cmH_2O , t_{rise} is usually set to 0.1 to 0.3 s, and k_{ter} is usually set to 5~10%. The values of P_{trigger} and t_{rise} influence the variation of $t_{\text{effective}}$. The value of k_{ter} influences the termination of the inspiratory phase and then affects the performance characteristics of switching asynchrony. Therefore, the thresholds of the elements of SI_{hao} are set as shown in Table 3.

The corresponding threshold value also indicates the automatic detection standard of PVA. When the value of SI_{tr} is not displayed as 1, it means that the triggering asynchrony has occurred (0 represents ineffective triggering; 2 represents double triggering; and -1 represents autotriggering). When the value of SI_{timing} exceeds 15%, it means that the rising time from the triggering to the plateau pressure is too long to provide effective respiratory support in time. This abnormal value usually means the occurrence of inspiration asynchrony (the rising time is set too long) or delayed triggering. When the value of SI_a falls in (10%, 100%), it means the occurrence of delayed switching asynchrony. Correspondingly, when the value of SI_a falls in (-100%, -10%), it means the occurrence of premature switching asynchrony. It should be noted that when the value of SI_a exceeds 1, it means the occurrence of the inspiration asynchrony (the

TABLE 3: The thresholds of the evaluation indices.

Evaluation index	Normal value/threshold
SI_{tr}	1
SI_{timing}	<15%
SI_a	(-10%, 10%)
SI_{peepi}	0

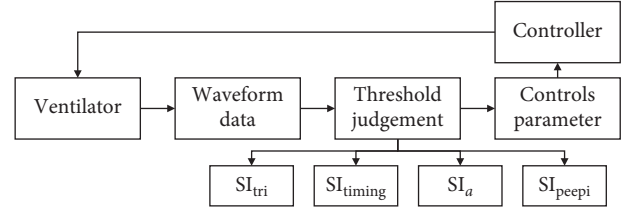


FIGURE 18: Controlling principle for using SI_{hao} as the response parameter.

rising time is set too short). When the value of SI_{peepi} is not equal to 0, it indicates that the presence of PEEP_i results in exhalation asynchrony. The support degree of PEEP needs to be adjusted according to the value of SI_{peepi} to offset the influence of PEEP_i.

In addition, SI_{hao} can also be used as the response parameter of ventilator regulation. If the asynchrony phenomenon occurs, the control parameters of the ventilator can be adjusted according to the corresponding evaluation index. As shown in Figure 18, the ventilator waveform data of each breathing cycle is analyzed to obtain the threshold range of the four elements of the vector, SI_{hao} . After the threshold judgement, the control parameters can be adjusted according to the judgement results. If the value of the synchrony index exceeds the threshold, it will correspond to the specific asynchrony phenomena mentioned above. Then the control parameters could be adjusted according to the corresponding asynchrony phenomena. For example, when premature asynchrony occurs, the value of k_{ter} is appropriately lower. The adjustment of control parameters will realize the feedback control of the ventilator through the controller [41–44]. In future research, the control law will be designed to realize the adaptive control of the mechanical ventilation system.

4. Conclusion

In this research, a novel pneumatic model of the mechanical ventilation system is established to simulate the mechanical ventilation process, and different PVA phenomena are simulated. An experimental prototype is established to verify the validity of the pneumatic model and the accuracy of the mathematical equations. The goodness of fit between the experimental curve and the simulation curve exceeds 90%.

By analyzing the morphological characteristics of different asynchrony phenomena, the PVA phenomena are divided into four categories, namely, (1) triggering asynchrony; (2) inspiration asynchrony; (3) switching

asynchrony; and (4) exhalation asynchrony. A method to detect different PVA phenomena by morphology was proposed in this article. Based on this method, almost all the PVA phenomena in pressure support ventilation are analyzed.

A new type of evaluation index, SI_{hao} , is established in this study, which completes the evaluation of patient-ventilator synchrony of each breathing cycle during mechanical ventilation in the form of a vector. It reflects the real-time performance of patient-ventilator synchrony with the use of a morphology-based method and gives the index for automatic evaluation. The four elements of the evaluation index correspond ingeniously to different types of PVA, establishing the link between synchrony evaluation and asynchrony detection. Therefore, the evaluation index, SI_{hao} , can also be used as a response parameter in the control of ventilator, optimizing the control algorithm, achieving the intelligent adjustment of parameters, and improving the patient-ventilator synchrony.

This work proposes a novel method to evaluate patient-ventilator synchrony during mechanical ventilation at the current stage. It solves certain problems of the existing results in clinical application. The existing asynchrony index cannot evaluate the performance of synchrony, while the synchrony index proposed in this article can comprehensively evaluate the patient-ventilator synchrony. The existing neural index needs to be obtained by the invasive operations, which adds difficulty to clinical promotion. In contrast, the method in this work is noninvasive, which is safer and more convenient.

This research can be used to adjust clinical strategies during mechanical ventilation. And it is helpful for the pathological analysis of patients during the treatment. It aims at reducing the burden on clinicians, optimizing the mechanical ventilation process, effectively reducing the complications caused by PVA, and realizing the adaptive control of the ventilator and weaning process in the future.

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 paper.

Authors' Contributions

Liming Hao and Shuai Ren contributed equally to this work.

Acknowledgments

The research was funded by grants of China Postdoctoral Science Foundation [2019M660391], Open Foundation of the State Key Laboratory of Fluid Power and Mechatronic Systems [GZKF-201920], Outstanding Young Scientists in

Beijing [BJJWZYJH01201910006021], and National Key Research and Development Project [2019YFC0121700].

References

- [1] K. D. Kochanek, S. L. Murphy, J. Q. Xu, and E. Arias, "Deaths: final data for 2017," in *National Vital Statistics Reports*, vol. 68, no. 9, National Center for Health Statistics, Hyattsville, MD, USA, 2019.
- [2] E. Arias and J. Xu, "United states life tables, 2017," in *National Vital Statistics Reports*, vol. 68, no. 7, National Center for Health Statistics, Hyattsville, MD, USA, 2019.
- [3] GOLD, "Global initiative for chronic obstructive lung disease. Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease," 2009, <http://www.goldcopd.com/>.
- [4] M. Vitacca, E. Clini, R. Porta, K. Foglio, and N. Ambrosino, "Acute exacerbations in patients with COPD: predictors of need for mechanical ventilation," *European Respiratory Journal*, vol. 9, no. 7, pp. 1487–1493, 1996.
- [5] T. D. Girard and G. R. Bernard, "Mechanical ventilation in ARDS," *Chest*, vol. 131, no. 3, pp. 921–929, 2007.
- [6] S. A. Nămendys-Silva, "Respiratory support for patients with COVID-19 infection," *The Lancet Respiratory Medicine*, vol. 8, no. 4, p. e18, 2020.
- [7] K. G. Mellott, M. J. Grap, C. L. Munro, C. N. Sessler, and P. A. Wetzel, "Patient-ventilator dyssynchrony: clinical significance and implications for practice," *Critical Care Nurse*, vol. 29, no. 6, pp. 41–55, 2009.
- [8] A. W. Thille, P. Rodriguez, B. Cabello, F. Lellouche, and L. Brochard, "Patient-ventilator asynchrony during assisted mechanical ventilation," *Intensive Care Medicine*, vol. 32, no. 10, pp. 1515–1522, 2006.
- [9] G. Murias, A. Villagra, and L. Blanch, "Patient-ventilator dyssynchrony during assisted invasive mechanical ventilation," *Minerva Anestesiologica*, vol. 79, no. 4, pp. 434–444, 2013.
- [10] G. Carteaux, A. Lyazidi, A. Cordoba-Izquierdo et al., "Patient-ventilator asynchrony during noninvasive ventilation," *Chest*, vol. 142, no. 2, pp. 367–376, 2012.
- [11] D. Tassaux, M. Gannier, A. Battisti, and P. Joliet, "Impact of expiratory trigger setting on delayed cycling and inspiratory muscle workload," *American Journal of Respiratory and Critical Care Medicine*, vol. 172, no. 10, pp. 1283–1289, 2005.
- [12] L. Blanch, A. Villagra, B. Sales et al., "Asynchronies during mechanical ventilation are associated with mortality," *Intensive Care Medicine*, vol. 41, no. 4, pp. 633–641, 2015.
- [13] G. Carteaux, A. Córdoba-Izquierdo, A. Lyazidi, L. Heunks, A. W. Thille, and L. Brochard, "Comparison between neurally adjusted ventilatory assist and pressure support ventilation levels in terms of respiratory effort," *Critical Care Medicine*, vol. 44, no. 3, pp. 503–511, 2016.
- [14] J. Doorduyn, L. H. Roesthuis, D. Jansen, J. G. Van Der Hoeven, H. W. H. Van Hees, and L. M. A. Heunks, "Respiratory muscle effort during expiration in successful and failed weaning from mechanical ventilation," *Anesthesiology*, vol. 129, no. 3, pp. 490–501, 2018.
- [15] C. Subirà, C. De Haro, R. Magrans, R. Fernández, and L. Blanch, "Minimizing asynchronies in mechanical ventilation: current and future trends," *Respiratory Care*, vol. 63, no. 4, pp. 464–478, 2018.
- [16] M. Cascella, M. Rajnik, A. Cuomo, S. C. Dulebohn, and R. D. Napoli, "Features, evaluation and treatment coronavirus

- (COVID-19),” in *StatPearls [Internet]*, StatPearls Publishing, Treasure Island, FL, USA, 2020.
- [17] K. Liu, Y. Chen, R. Lin, and K. Han, “Clinical features of COVID-19 in elderly patients: a comparison with young and middle-aged patients,” *Journal of Infection*, vol. 80, no. 6, pp. e14–e18, 2020.
 - [18] S. Ren, W. Li, L. Wang et al., “Numerical analysis of airway mucus clearance effectiveness using assisted coughing techniques,” *Scientific Reports*, vol. 10, no. 1, p. 2030, 2020.
 - [19] S. Ren, J. Niu, Z. Luo et al., “Cough expired volume and cough peak flow rate estimation based on GA-BP method,” *Complexity*, vol. 2020, Article ID 9036369, 9 pages, 2020.
 - [20] S. Ren, Y. Shi, M. Cai, H. Zhao, Z. Zhang, and X. D. Zhang, “ANSYS-MATLAB co-simulation of mucus flow distribution and clearance effectiveness of a new simulated cough device,” *International Journal for Numerical Methods in Biomedical Engineering*, vol. 34, no. 6, p. e2978, 2018.
 - [21] J. O. Nilsestuen and K. D. Hargett, “Using ventilator graphics to identify patient-ventilator asynchrony,” *Respiratory Care*, vol. 50, no. 2, pp. 202–234, 2005.
 - [22] D. Colombo, G. Cammarota, M. Alemani et al., “Efficacy of ventilator waveforms observation in detecting patient-ventilator asynchrony*,” *Critical Care Medicine*, vol. 39, no. 11, pp. 2452–2457, 2011.
 - [23] M. Ramsay, S. Mandal, E.-S. Suh et al., “Parasternal electromyography to determine the relationship between patient-ventilator asynchrony and nocturnal gas exchange during home mechanical ventilation set-up,” *Thorax*, vol. 70, no. 10, pp. 946–952, 2015.
 - [24] C. Sinderby, S. Liu, D. Colombo et al., “An automated and standardized neural index to quantify patient-ventilator interaction,” *Critical Care*, vol. 17, no. 5, p. R239, 2013.
 - [25] M. E. Ward, D. Eidelman, D. G. Stubbing, F. Bellemare, and P. T. Macklem, “Respiratory sensation and pattern of respiratory muscle activation during diaphragm fatigue,” *Journal of Applied Physiology*, vol. 65, no. 5, pp. 2181–2189, 1988.
 - [26] R. W. Jodat, J. D. Horgan, and R. L. Lange, “Simulation of respiratory mechanics,” *Biophysical Journal*, vol. 6, no. 6, pp. 773–785, 1966.
 - [27] J. S. Mecklenburgh, T. A. A. Al-Obaidi, and W. W. Mapleson, “A model lung with direct representation of respiratory muscle activity[†],” *British Journal of Anaesthesia*, vol. 68, no. 6, pp. 603–612, 1992.
 - [28] A. W. Thille, A. Lyazidi, J.-C. M. Richard, F. Galia, and L. Brochard, “A bench study of intensive-care-unit ventilators: new versus old and turbine-based versus compressed gas-based ventilators,” *Intensive Care Medicine*, vol. 35, no. 8, pp. 1368–1376, 2009.
 - [29] A. G. Polak and J. Mroczka, “Nonlinear model for mechanical ventilation of human lungs,” *Computers in Biology and Medicine*, vol. 36, no. 1, pp. 41–58, 2006.
 - [30] F. Vicario, A. Albanese, N. Karamolegkos, D. Wang, A. Seiver, and N. W. Chbat, “Noninvasive estimation of respiratory mechanics in spontaneously breathing ventilated patients: a constrained optimization approach,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 4, pp. 775–787, 2015.
 - [31] L. Hao, Y. Shi, M. Cai et al., “Dynamic characteristics of a mechanical ventilation system with spontaneous breathing,” *IEEE Access*, vol. 7, pp. 172847–172859, 2019.
 - [32] Y. Shi, J. Niu, M. Cai, and W. Xu, “Dimensionless study on dynamics of pressure controlled mechanical ventilation system,” *Journal of Mechanical Science and Technology*, vol. 29, no. 2, pp. 431–439, 2015.
 - [33] S. Ren, Y. Shi, M. Cai, and W. Xu, “Influence of airway secretion on airflow dynamics of mechanical ventilated respiratory system,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1660–1668, 2018.
 - [34] Y. Shi, Y. Wang, M. Cai, B. Zhang, and J. Zhu, “An aviation oxygen supply system based on a mechanical ventilation model,” *Chinese Journal of Aeronautics*, vol. 31, no. 1, pp. 197–204, 2018.
 - [35] J. Niu, Y. Shi, Z. Cao et al., “Study on air flow dynamic characteristic of mechanical ventilation of a lung simulator,” *Science China Technological Sciences*, vol. 60, no. 2, pp. 243–250, 2017.
 - [36] S. Ren, M. Cai, Y. Shi, W. Xu, and X. D. Zhang, “Influence of bronchial diameter change on the airflow dynamics based on a pressure-controlled ventilation system,” *International Journal for Numerical Methods in Biomedical Engineering*, vol. 34, no. 3, p. e2929, 2018.
 - [37] F. Vicario, A. Albanese, D. Wang, N. Karamolegkos, and N. W. Chbat, “Constrained optimization for noninvasive estimation of work of breathing,” in *Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5327–5330, IEEE, Milan, Italy, August 2015.
 - [38] S. M. Donn and S. K. Sinha, “Pressure support ventilation,” *Manual of Neonatal Respiratory Care*, Springer, Cham, Switzerland, pp. 297–300, 2017.
 - [39] C. Alexopoulou, E. Kondili, M. Plataki, and D. Georgopoulos, “Patient-ventilator synchrony and sleep quality with proportional assist and pressure support ventilation,” *Intensive Care Medicine*, vol. 39, no. 6, pp. 1040–1047, 2013.
 - [40] I. S. Sehgal, S. Dhooria, A. N. Aggarwal, D. Behera, and R. Agarwal, “Asynchrony index in pressure support ventilation (PSV) versus neurally adjusted ventilator assist (NAVA) during non-invasive ventilation (NIV) for respiratory failure: systematic review and meta-analysis,” *Intensive Care Medicine*, vol. 42, no. 11, pp. 1813–1815, 2016.
 - [41] Z. Li, C. Y. Su, G. Li, and H. Su, “Fuzzy approximation-based adaptive backstepping control of an exoskeleton for human upper limbs,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 555–566, 2015.
 - [42] H. Su, S. E. Ovrur, X. Zhou, W. Qi, G. Ferrigno, and E. De Momi, “Depth vision guided hand gesture recognition using electromyographic signals,” *Advanced Robotics*, pp. 1–13, 2020.
 - [43] H. Su, C. Yang, G. Ferrigno, and E. De Momi, “Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
 - [44] H. Su, J. Sandoval, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, “Safety-enhanced collaborative framework for tele-operated minimally invasive surgery using a 7-DoF torque-controlled robot,” *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2915–2923, 2018.

Research Article

Numerical Calibration Method for Vehicle Velocity Data from Electronic Registration Identification of Motor Vehicles Based on Mobile Edge Computing and Particle Swarm Optimization Neural Network

Jingfeng Yang ^{1,2}, Zhiyong Luo,³ Nanfeng Zhang,⁴ Honggang Wang,⁵ Ming Li ^{6,7}
and Jinchao Xiao ^{1,2}

¹Shenyang Institute of Automation (Guangzhou) Chinese Academy of Sciences, Guangzhou 511458, China

²Shenyang Institute of Automation Chinese Academy of Sciences, Shenyang 110016, China

³School of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou 510006, China

⁴Technical Center of Huangpu Customs District China, Guangzhou 510730, China

⁵School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China

⁶South China Agricultural University, Guangzhou 510642, China

⁷Yaz Technology Co., Ltd., Guangzhou 510630, China

Correspondence should be addressed to Ming Li; liming@yazcn.com and Jinchao Xiao; xiaojinchao@sia.cn

Received 20 April 2020; Revised 13 June 2020; Accepted 13 July 2020; Published 30 August 2020

Guest Editor: Yanan Li

Copyright © 2020 Jingfeng Yang 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 the development of technology for smart cities, the installation and deployment of electronic motor vehicle registration identification have attracted great attention in terms of smart transportation in recent years. Vehicle velocity measurement is one of the fundamental data collection efforts for motor vehicles. The velocity detection using electronic registration identification of motor vehicles is constrained by the detection algorithm, the material of the automobile windshield, the placement of the decals, the installation method of the signal reader, and the angle of the antenna. The software and hardware for electronic motor vehicle registration identification produced in the standard manner cannot meet the accuracy of velocity detection for all scenarios. Based on the actual application requirements, we propose a calibration method for the numerical output of the automobile velocity detector based on edge computing of the optimized multiple reader/writer velocity values and based on a particle swarm-optimized radial basis function (RBF) neural network. The proposed method was tested on a two-way eight-lane road, and the test results showed that it can effectively improve the accuracy of velocity detection using electronic registration identification of motor vehicles. Compared with the actual velocity, 87.12% of all the data samples had an error less than 5%, and 91.76% of the data samples for vehicles in the center lane had an error less than 5%. By calibrating the electronic vehicle velocity based on the registration identification, the accuracy of velocity detection in different application environments can be improved. Moreover, the method can establish an accurate foundation for application in traffic flow management, environmental protection, traffic congestion fee collection, and special vehicle traffic management.

1. Introduction

As urban development expands, the rapid increase in the number of vehicles on roads has caused various urban management problems and has led to the construction of

large-scale, traffic-related infrastructures and the construction and the deployment of extensive vehicle-related facilities and equipment. These are specifically manifested in the use of various traffic sensing and road monitoring techniques and equipment including video cameras, GPS,

geomagnetic, and radar. These devices are deployed on the road by local public security departments to ensure the safety and smooth operation of road traffic, strengthen the enforcement of traffic regulations, and ensure the intelligent control of city-level traffic. Intelligent control of traffic has become an important component of smart cities. However, video-based vehicle information recognition technology is susceptible to interference from environmental factors, such as weather, lighting, and distance from the vehicle. As a result, the effective recognition rate of video technology cannot meet the requirements of intelligent traffic management and cannot automatically identify false license plates, overlay plates, and intentionally obscured plates. Electronic registration identification of motor vehicles based on a single video image cannot satisfy the needs of smart traffic and the need to maintain a general social order [1]. To meet these needs, the application of electronic registration identification (ID) is proposed and promoted on a national level by establishing a national standard. Electronic vehicle registration identifications have been promoted in a limited number of cities and is used for intelligent parking [2], smart signal control and application [3, 4], web-linked smart vehicles [5], special vehicle passing management and special electronic registration identification of motor vehicles and control [6], and traffic operation supervision and environmental driving restrictions [7]. However, due to various technical problems, the vehicle velocity data collected using electronic vehicle registration identification still cannot serve for traffic law enforcement; this has restricted the scope of application of electronic vehicle registration identification. In terms of increasing the accuracy of collected data with spatial attributes, many researchers have conducted studies on multiobjective optimization algorithms in recent years. Grid-based methods have found good applications [8–10]. For example, Wu [11], Luo [12], and Kong [13] located robot arm movement by constructing a grid; Leong [14] maintained the diversity of solutions by constructing a grid; Knowles [8] used adaptive grids to store the obtained nondominated vectors; Yang [15] used grid technology to apply evolutionary algorithms to solve high-dimensional multiobjective optimization problems; Li [16] used grids to simultaneously characterize the convergent and distributive characteristics and proposed a multitarget particle swarm optimization algorithm based on grid sorting. The algorithm uses coordinate mapping, ordering of element optimization in the grid coordinate system, and the Euclidean distance between the element and the boundary of the approximate optimum. To improve the accuracy of the velocity data collected using electronic registration identification of motor vehicles, we combine the results of previous researchers and propose a numerical calibration method for vehicle velocity data collected by electronic registration identification of motor vehicles based on a particle swarm optimization neural network. By comparing the optimized vehicle calibration velocity and the data collected by the OBD port of the test vehicle, the reliability of the acquired velocity data is further improved.

2. Vehicle Velocity Measurement Method Using Electronic Registration Identification of Motor Vehicles

2.1. Installation and Connection Method for Reader/Writer of Electronic Vehicle Registration Identification. The collection of vehicle velocity data based on electronic registration identification of motor vehicles involves installing a comprehensive sensing base station at each key intersection or public security checkpoint, expressway ramp, and checkpoints at city entries. A detection coil should be installed 15 meters away from the law enforcement monitoring station or the key intersection on the main highway. To inspect the vehicles passing through the road, a gantry should be installed at a distance of 15 meters from the detection coil and an electronic registration identification of the motor vehicle device should be installed on the gantry to automatically identify the license plate of the vehicle. The reader/writer antenna installed on the gantry acquires the data from passing vehicles equipped with an electronic registration identification of motor vehicles. When the vehicle comes in contact with the coil, the detection coil emits two signals, one signal goes to the high-definition video camera and the other signal goes to the reader/writer antenna. The vehicle license plate and the electronic registration identification of the motor vehicle signal of the vehicle are captured simultaneously. After the signal is read, it is uploaded to the comprehensive sensing base station for data processing.

The system at the comprehensive sensing base station consists of a license plate recognition system and a vehicle electronic registration identification of motor vehicles decal read and write system. The system as a whole performs data collection, verification, transmission, and processing in practical applications. The overall layout is shown in Figures 1 and 2.

An appropriate location at the traffic checkpoint or the road junction is chosen to install the gantry. The induction coil is installed 15 meters from the gantry. Installed on the gantry are the electronic registration identification of motor vehicles' antenna, the high-definition video camera, and the fill-in light. Two trigger signals are sent out through the induction coil, one to the video camera and one to the vehicle electronic antenna. The antenna of the electronic registration identification of motor vehicles is connected to the read/write control of the vehicle. The video camera is connected to a front-end computer through a network or to an exchanger.

2.2. Vehicle Velocity Detection Based on Reader/Writer of Electronic Registration Identification of Motor Vehicles. The main method of using electronic registration identification of motor vehicles and readers/writers to acquire vehicle velocity data is the ultrahigh-frequency radio frequency identification technology. The radio frequency identification reader, using ultrahigh-frequency identification technology, interacts with the radio frequency electronic registration identification of motor vehicle tag installed on the wind shield of the vehicle through the ultrahigh-frequency horizontally polarized directional

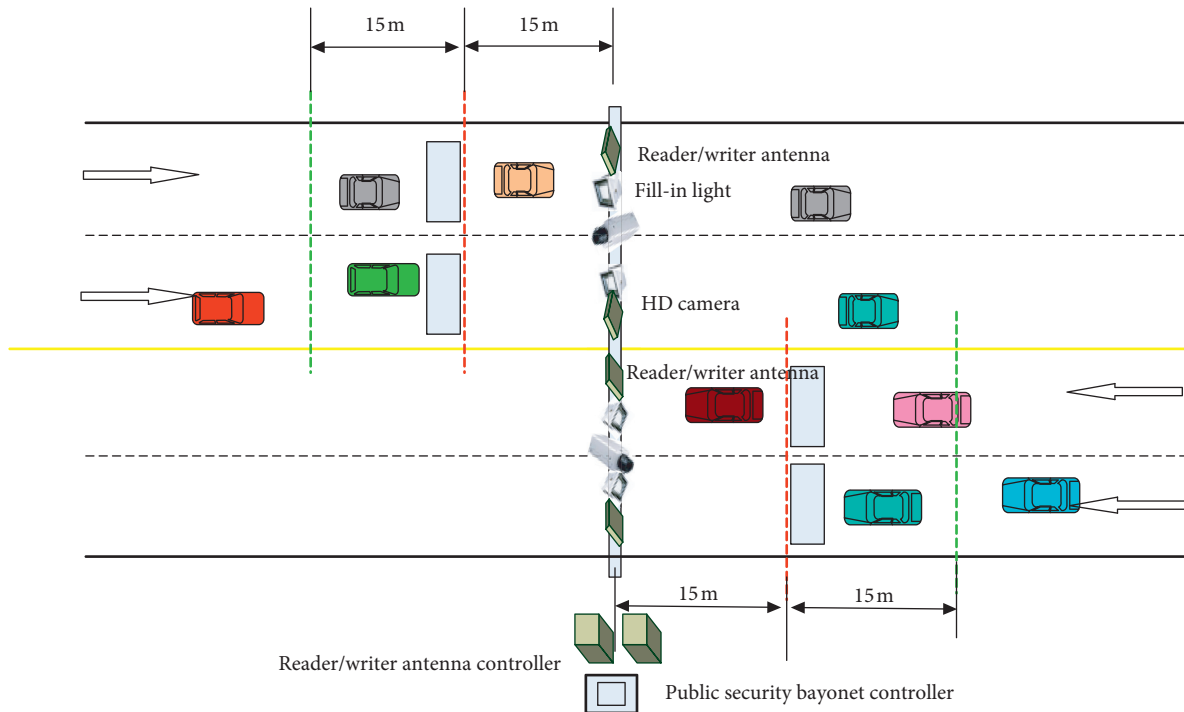


FIGURE 1: Overall layout diagram.



FIGURE 2: Picture of the actual installation.

antenna. The velocity of the vehicle is calculated by noting the time difference for the vehicle to go through a fixed-length identification zone. Under normal circumstances, the transmission power of the RFID reader is 30–33 dBm, and the antenna gain is 10–12 dBi. Combined with comprehensive factors such as the sensitivity of the reader and the tag, the reading and writing coverage of the RFID reader is usually between 0 and 35 meters. The steps to calculate the velocity of a vehicle using an electronic registration identification of motor vehicle tag and a reader are as follows:

Step 1. For an arbitrary vehicle shown in Figure 1, the identification record and time data are recorded as the vehicle enters the identification section of the electronic reader/writer; the same records are also recorded as the vehicle leaves the identification section.

Step 2. Interruption in the recognition record is checked to determine that the vehicle has left the identification section.

Step 3. The velocity is calculated using the formula:

$$V = (S / (T_o - T_i)),$$

where V is the velocity with which the vehicle passes through the recognition section, S is the direct distance according to the electronic registration identification of motor vehicle reader, that is, the length of the recognition section, T_i is the time when the vehicle enters the recognition section, and T_o is the time when the vehicle leaves the recognition section.

However, constrained by factors such as the material and angle of the vehicle's windshield, any presence of a film, location of the decal, or any other interference, different electronic registration identification of motor vehicle readers/writers often have different maximum reading distances. The differences in the maximum reading distance prevents the electronic reader from determining where the corresponding vehicle enters the recognition section, that is, where it was first read. As a result, the reader cannot accurately calculate the vehicle velocity based on the distance and time it read.

Given the various interferences on the maximum read distance of the electronic registration identification of motor vehicle reader/writer, the calculation of the vehicle velocity in actual practice is often combined with the received signal strength indication (RSSI), that is, the strength of the signal returned by the electronic registration identification of the motor vehicle tag. The higher the RSSI value, the closer the vehicle is to the antenna, and vice versa. Specifically, the position of the vehicle at different times may be calculated by using the RSSI value returned from the electronic registration identification of the motor vehicle tag of the vehicle at different times, and the velocity of the vehicle may then be

calculated according to the time differences of the corresponding positions. That is, by analyzing the RSSI value of the feedback signal that is first recognized when the corresponding vehicle enters the recognition section, it can be determined whether the vehicle position at the first recognition is at the boundary between the direct shot area and the blind area or at the farthest point in the reflection area. The formula $V = (S/(T_o - T_i))$ can then be used to calculate the velocity of the vehicle. In actual implementation, however, the presence of blind spots and the emergence of environmental conditions may affect the results of velocity calculation. Analysis shows that although a blind spot may disappear, the radio frequency identification effect in the cross section of the blind area will still be different from that in the direct area. This is judged by the fluctuation of the number of recognitions per unit time from when the vehicle enters the recognition section to when it leaves the recognition section. If the fluctuation is not large, the vehicle has been recognized in the direct shot area the whole time. If the number of recognitions per unit time varies greatly and if one period shows a very high number of recognitions per unit time and the other period shows a relatively low number of recognitions, or if a vehicle in another lane is recognized and calculated by the electronic registration identification of the motor vehicle reader, then it is possible that a blind spot had disappeared. In a real environment, it is necessary to set up different calculation methods for calculating vehicle velocity in different situations. Since it is often difficult to accurately select the correct calculation method, the calculated vehicle velocity often deviates from the actual values.

3. Velocity Calibration Method Based on Particle Swarm Optimization Neural Network Algorithm

Since vehicle velocity data are affected by various environmental and equipment factors, we use the velocity data read by the OBD interface on the vehicle as the reference standard. We establish a velocity calibration model based on a particle swarm optimization neural network algorithm and incorporate various types of data influenced by environmental factors. In order to preserve the diversity of velocity data acquisition methods using electronic registration identification of motor vehicles, the source of the velocity data comes from data collected by the electronic registration identification of the motor vehicle reader/writer on vehicles in the lane of interest and in other lanes. On the two-way, eight-lane road section, the velocity of vehicles passing through the recognition section can theoretically be calculated from the data read by four electronic registration identification of motor vehicle readers/writers. This completes the detection network of electronic registration identification of motor vehicle readers/writers set up to improve the accuracy of velocity calibration. In addition, to improve the calculation efficiency of the neural network, attention is given to the distribution efficiency of sensing tasks and the collection efficiency of sensing data in the edge network in order to improve the scope of coverage and

sensing efficiency of the reader application. The readers that have deployed the task distribution service will serve as the source of task distribution and continuously distribute vehicle-borne velocity detection tasks to mobile devices. During the data collection process, the vehicle-borne devices serve as the source of the velocity data detection, and the data are transmitted back to the readers involved in data collection [17, 18].

3.1. RBF Neural Network. The RBF neural network is an artificial neural network that uses local adjustment to perform function mapping. It has strong input and output mapping functions and is an optimal network for performing mapping functions in feed-forward networks. It has a strong nonlinear approximation capability, a simple network architecture, and a fast learning velocity. The output matrix of the hidden layer after iterative convergence has a linear relationship with the output. It is an ideal algorithm for calculating the degree of influence [19, 20].

Using RBF as the “base” of the hidden unit constitutes the hidden layer space, so that the input vector can be directly mapped to the hidden space, without the need for a weighted connection. After the center point of the RBF is determined, this mapping relationship is also determined. The mapping of the hidden layer space to the output space is linear, that is, the output of the network is a linear weighted sum of the output of the hidden unit, and the weights here are the tunable parameters of the network. Here, the role of the hidden layer is to map the vector from a low dimension to a high dimension, so that the linearly inseparable case of the low dimension can become linearly separable in the high dimension. Thus, the network’s mapping from the input to the output is nonlinear, while the network’s output is linear with respect to the adjustable parameters. The weights of the network can be solved directly from the linear equations, which greatly speed up the learning and avoids local minima.

In the two-staged learning process of the RBF neural network, unsupervised learning mainly determines the radial basis vector and normalized parameters of the Gaussian basis function of each node of the hidden layer based on the input samples; that is, it determines the center and variance of the hidden layer basis function. In the supervised learning phase, the weights of the hidden layer and the output layer are calculated using the least square method after determining the hidden layer parameters. The output of the i -th hidden layer of the RBF neural network is [20–22]

$$h_i(t) = \phi_i(\|x(t) - c_i\|) = \exp\left(-\sum_{i=1}^n \frac{(x_i(t) - c_i)^2}{2s_i^2}\right), \quad (1)$$

where $x(t)$ is the input vector of the network at time t , c_i is the central vector of the i -th unit of the hidden layer, s_i is the shaper parameter of the Gaussian function, where $s_i > 0$, $1 \leq i \leq L$, and L is the number of nodes of the hidden layer.

The overall output of the RBF neural network is as follows:

$$\begin{aligned}
y &= \sum_{i=1}^L w_i h_i, \\
&= \sum_{i=1}^L w_i \phi_i(\|x(t) - c_i\|), \\
&= H^T W,
\end{aligned} \tag{2}$$

where $H = [h_1, h_2, \dots, h_L]^T$ and $W = [w_1, w_2, \dots, w_L]$.

For an RBF network with k input nodes, m output nodes, and n learning samples, the error objective function is

$$\begin{aligned}
E(n) &= \frac{1}{2} \sum_{k=1}^n \lambda^{n-k} \sum_{i=1}^M \delta_i(n), \\
&= \frac{1}{2} \sum_{k=1}^n \lambda^{n-k} \sum_{i=1}^M [\bar{y}_i(n) - y_i(n)]^2, \\
&= \lambda E(n-1) + \frac{1}{2} \sum_{i=1}^M [\bar{y}_i(n) - H^T(n)W_i(n-1)]^2,
\end{aligned} \tag{3}$$

where λ is the forgetting factor and $\delta_i(n)$, $\bar{y}_i(n)$, and $y_i(n)$ are, respectively, the error of the output node, the expected output, and the actual output.

The operation performance of the RBF neural network mainly depends on the center and radial basis of the hidden layer function and the weight of the output layer. Additionally, the number and the center of hidden nodes in the RBF network are difficult to determine, which affects the accuracy of the entire network. When the selection of the training samples contains a high degree of randomness, the number of errors will increase rapidly, and it is difficult to meet requirements. These errors will contribute directly to the calculation of the predicted results. In addition, when determining these parameters, the learning strategy of the traditional RBF neural network has a great disadvantage, that is, it can only find the optimal solution in local space. If these parameters are not set properly, the accuracy of the approximation will be reduced or the network will diverge. Furthermore, since the input samples contain various types of data, including discrete values, continuous values, and missing values, the training samples are usually obtained by random extraction. The center of the RBF neural network hidden layer basis function is selected from the input sample set, which has a large dependence on the training samples. In many cases, it is difficult to reflect the true input-output relationship of the system, and it is easy for data pathology to occur when there are too many initial center points. This sample data selection dilemma is a key problem to be solved when an RBF neural network is used for nonlinear system modeling. To this end, we introduce in this paper, a particle swarm optimization algorithm to optimize the parameters of the RBF neural network, and we apply it to the numerical calibration of vehicle velocity detection based on electronic registration identification of motor vehicle reading.

3.2. Particle Swarm Optimization Algorithm. Particle swarm optimization (PSO) is an efficient heuristic parallel search algorithm. Since it converges rapidly during optimization and does not require gradient information of the objective function, it enjoys the advantage of being simple and easy to implement, and thus has been applied in the field of engineering technology [23, 24].

The solution of the PSO algorithm for the optimization problem is abstracted as an entity, the ‘‘particle,’’ without weight and volume. Each particle has a fitness value determined by the optimized function, and this value is searched in the search space at a certain velocity through the cooperation and competition among the particles. The specific process is to first initialize a group of random particles; these particles update their velocity and position by tracking the two optimal particles. The two optimal particles are the individual optimal particle (the optimal solution found by the particle itself, known as the individual optimal position) and the global optimal particle (the optimal solution found by the entire population up to now, known as the global best position).

Let the search space be an n -dimensional space and the particle swarm consists of n_s particles; $x_i^{(t)} = (x_{i,1}^{(t)}, x_{i,2}^{(t)}, \dots, x_{i,n}^{(t)})$ and $v_i^{(t)} = (v_{i,1}^{(t)}, v_{i,2}^{(t)}, \dots, v_{i,n}^{(t)})$ are the position and velocity, respectively, of the i -th particle at time t in the search space, $p_i^{(t)} = (p_{i,1}^{(t)}, p_{i,2}^{(t)}, \dots, p_{i,n}^{(t)})$ is the personal best position (referred to as *pbest*) of the i -th particle at time t , and $g_i^{(t)} = (g_{i,1}^{(t)}, g_{i,2}^{(t)}, \dots, g_{i,n}^{(t)})$ is the global best position (referred to as *gbest*) at time t . According to the PSO algorithm, the update formula for particle velocity and position is as follows [23–25]:

$$v_{i,j}^{(t+1)} = v_{i,j}^{(t)} + c_1 r_1 (p_{i,j}^{(t)} - x_{i,j}^{(t)}) + c_2 r_2 (g_{i,j}^{(t)} - x_{i,j}^{(t)}), \tag{4}$$

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + v_{i,j}^{(t+1)}, \tag{5}$$

where c_1 and c_2 are the learning factors or acceleration constants, respectively, for adjusting the step size of the particle flying to the personal best position and the global best position. In order to reduce the probability of the particle to flying out of the search space, the velocity of the particle in each dimension is usually limited to a certain value within the range. Also, r_1 and r_2 are random numbers obeying a uniform distribution in the interval $[0, 1]$, $i = 1, 2, \dots, N_s$, N_s is the scale of the particle swarm, and $j = 1, 2, \dots, n$.

Traditional PSO algorithms generally follow the following steps to complete the iteration [26]: step one, initializing the velocity and position of the particle swarm; step two, calculating the fitness value of each particle in the particle swarm; step three, updating the personal best position of each particle; step four, updating the global best position of each particle; step five, updating the velocity and position of the particles and taking certain measures to ensure that the particles remain in the search space; step six, determining whether the termination conditions are met (terminate the algorithm if yes, or return to step two).

However, in the updating process of the traditional PSO algorithm, the inertia weight, learning factor, and maximum

velocity jointly determine the balance between the global exploration and local development capabilities of the algorithm, which has a direct impact on the search performance of the algorithm.

3.3. RBF Neural Network Based on Particle Swarm Optimization. The key point of the seeking process is that the parameters of the RBF neural network are combined into a particle vector by coding, and the individual optimal value and the global optimal value of particles are reduced to the corresponding parameters of the RBF neural network after initialization; the individual utility extreme value of particles is calculated by comparison, and the individual optimal value and the global optimal value are updated after comparison with the current individual optimal value and the global optimal value. Finally, the global optimal solution is decoded, restored to the parameters of the network, and the RBF network prediction is carried out. Before neural network training, the convergence of the RBF neural network can be ensured by normalizing all sample data.

By combining the calibration requirement vehicle velocity detection data using electronic registration identification of motor vehicles with characteristics of PSO and the RBF neural network, we arrived at the following procedure for optimizing the kernel function parameter of the RBF neural network with the PSO algorithm:

Step 1. Data Normalization Processing. In order to improve the generalization ability, the convergence velocity, the correlation between influencing factors, and the fitting effect of the neural network, we use a maximum and minimum value method to normalize the sample data to the $[0, 1]$ range: $y(k) = (x(k) - \min(x(n)))/\max(x(n) - \min(x(n)))$, $k = 1, 2, \dots, N$, where $\min(x(n))$ is the minimum value of the sample data and $\max(x(n))$ is the maximum value of the sample data.

Step 2. The particle swarm and the neural network are initialized, the scale is determined, i.e., the dimension of the particle swarm, and the mapping is established between the dimensional space of the PSO particles and the connection weight of the neural network. The population size is set as N_s and the maximum number of iterations as T_{\max} . The population location matrix $\text{Rand}(x)$ and the velocity matrix $\text{Rand}(v)$ are randomly generated as follows:

$$\text{Rand}(x) = \begin{bmatrix} x_{1,1} & \cdots & x_{1,M} \\ \vdots & & \vdots \\ x_{N,1} & \cdots & x_{N,M} \end{bmatrix}, \quad (6)$$

$$\text{Rand}(v) = \begin{bmatrix} v_{1,1} & \cdots & v_{1,M} \\ \vdots & & \vdots \\ v_{N,1} & \cdots & v_{N,M} \end{bmatrix},$$

$$x_{i,j}^{(t+1)} = x_{\min}^{(t)} + (x_{\max}^{(t)} - x_{\min}^{(t)}) \times \text{Rand}(x), \quad (7)$$

$$v_{i,j}^{(t+1)} = -v_{\max}^{(t)} + 2v_{\max}^{(t)} \times \text{Rand}(v), \quad (8)$$

where $M = k(n + 1)$, k is the number of nodes in the hidden layer, n is the dimension of the input feature vector, $\{x_{i,j} | i = 1, 2, \dots, N; j = 1, 2, \dots, M\}$ represents the j -th dimension component of the i -th particle position vector, $(x_{i,1}, x_{i,2}, \dots, x_{i,M})$ corresponds to the radial basis parameter of the RBF neural network, representing the j -th dimension component of the velocity vector of the i -th particle, and the value of $v_{\max}^{(t)}$ is related to $x_{\max}^{(t)}$ and $x_{\min}^{(t)}$.

Step 3. The fitness of each particle is calculated, the mean square error of the neural network is used as the fitness function of PSO. The fitness function is a measure of the position of particles in space. The greater the value of the fitness function, the better the position of the particle. It is defined as follows:

$$\text{FitnessFunc}_i = \frac{1}{1 + D(x_{i,1}, x_{i,2}, \dots, x_{i,M})}, \quad (9)$$

where FitnessFunc_i is the fitness function value of the i -th particle and $D(x_{i,1}, x_{i,2}, \dots, x_{i,M})$ is the sum of the average errors squared of the neural network when the kernel function parameter is equal to $(x_{i,1}, x_{i,2}, \dots, x_{i,M})$.

Step 4. For each particle, its fitness is compared with the fitness of the best location it has experienced, and if the former is better, p_{best} is updated.

Step 5. For each particle, its fitness is compared with the fitness of the best location experienced by the swarm, and if the former is better, g_{best} is updated.

Step 6. The velocity and position of the particles are updated, and dynamic adaptive adjustment is performed using the inertial weights. Steps 3 through 5 are repeated until the calculation requirements are met. When the algorithm iteration stops, the weights and thresholds of the neural network correspond to the global optimum, which is the optimal solution to the problem.

Step 7. The vehicle velocity data are calibrated using the trained RBF neural network while simultaneously assessing the accuracy of the velocity data.

3.4. Numerical Optimization of Multireader Velocity Data Using Mobile Edge Computing. The shortcomings of the RBF neural network, including the local optima and the effects on PSO by inertial weight, learning factor, and maximum velocity, have direct impact on the calibration results in the probability sense. In order to further improve the accuracy of velocity calibration, the capability of having multiple readers simultaneously implementing velocity detection of the same electronic registration identification of the motor vehicle tag had been achieved before our study was conducted. After each reader/writer calibrates the velocity data using the RBF neural network based on the particle swarm

optimization algorithm, the k-means method is used to find the cluster center value of different reader/writer values as the final velocity calibration value. The calculation method is as follows.

Assuming that the cluster partitioning is (c_1, c_2, \dots, c_k) , the goal is to minimize the squared error E :

$$E = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2, \quad (10)$$

where μ_i is the mean vector of cluster C_i and $\mu_i = (1/|C_i|) \sum_{x \in C_i} x$.

In order to improve the computing efficiency of the reader/writer, the emphasis of the optimization algorithm is placed on the distribution efficiency of the sensing tasks and the collection efficiency of the sensing data in the edge network of the reader/writer. By designating a certain reader/writer as the main computing reader/writer, readers/writers already deployed for task distribution services will serve as sources for task distribution in the task distribution computation process. The main computing reader/writer will continually distribute the vehicle terminal velocity detection computing tasks to different readers. During the data collection and optimization process, all the readers/writers serve as the source of vehicle-borne devices in acquiring data and return the data of completed computing tasks to the main computing reader/writer for final calculation.

4. Tests and Results

4.1. Test Method and Data Description. The tests were conducted in a road section where the electronic registration identification of the motor vehicle reader has already been installed for testing. The road was a two-way, eight-lane urban expressway, with a gantry equipped with internet-connected testing instruments. There were eight sets of electronic registration identification of the motor vehicles readers, with each traffic lane aligned with at least one reader for direct acquisition of the electronic registration identification of motor vehicle data in the vertical direction.

Since the automobile electronic registration identification of the motor vehicle reader has already acquired the capability of group acquisition, data can be collected simultaneously when multiple electronic registration identification of motor vehicles pass through at the same time, and we placed 1000 electronic registration identification of motor vehicles simultaneously on the test vehicles and scattered them at different locations inside the vehicles. In this manner, we demonstrated the interference immunity performance between the electronic registration identification of motor vehicles.

The data read by the OBD interface on the vehicle is the true velocity of the moving vehicle. With the temporal data of the OBD interface synchronized with the temporal data of the reader, we selected three sedans as test vehicles, each carrying 1000 automotive electronic registration identification of motor vehicle cards, to pass under the gantry's readers with different velocity. The sample data set was

formed with the data read from the 1000 vehicle-borne electronic registration identification of motor vehicles as the input, with the actual velocity data read by the OBD interface as the output, all matched with the time data. It should be noted that when the vehicles passed under the gantry, the velocity data of the electronic registration identification of motor vehicles in the vehicles were read by multiple readers, and each electronic registration identification of motor vehicles can form velocity data in different readers. The final complete sample data set was obtained by testing the three sedans in different lanes at different velocity, and with the data from repeated tests matched. A comparison of the true values of the velocity and the measured values of the velocity is shown in Figure 3. The abscissa is for the measured value of the velocity and the ordinate is for the true value of the velocity. If the two values are the same, the point falls on the straight line of $y = x$. If the true value is less than the measured value, the point falls below the straight line, and vice versa.

Based on the data distribution shown in Figure 3, the differences between the true velocity and the measured velocity are shown in Figure 4.

4.2. Test Results and Analysis. The experimental results show that the best effect is that the number of hidden layer centers of the RBF neural network is 9, and the network structure is 4-12-1. PSO algorithm parameters are divided into $c_1 = c_2 = 2.0$, the number of particle population is 46, and the maximum times of iterations is set to 500. Before neural network training, the convergence of the RBF neural network can be ensured by normalizing all sample data. After many experiments, when the control accuracy is 0.005, the average times of iterations using only the RBF neural network is about 320, while the average times of iterations using the PSO-RBF neural network is about 95.

The errors in the collected detection data reflects the difference between the data collected by the traditional velocity detection method and the true values. Most of the data had an error rate within 5%, which can fully meet the needs of practical applications; to a certain extent, this is better than the error rate of video, geomagnetic, and other methods for collecting velocity data and is closer to the needs of practical applications. However, in some industries or specific applications, detected data need to be closer to the actual velocity, and higher requirements for controlling the data error rate are imposed. In this paper, we use the particle swarm optimization neural network method to calibrate the velocity detected with electronic registration identification of motor vehicles so as to obtain a detected velocity that is closer to the actual data.

Test data for different velocity and different lanes were screened for perfect time matches; a total of 11086 sets of complete data were obtained. The data acquired for the eight lanes showed an approximately uniform distribution. It should be noted that since the driver can only control the velocity at a certain interval when driving the vehicle through the gantry and the velocity on urban expressways is not less than 20 km/h, all data values below 20 km/h were

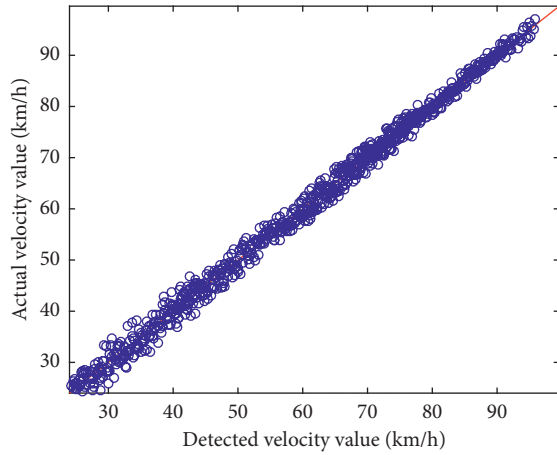


FIGURE 3: Actual velocity value and the detected velocity value.

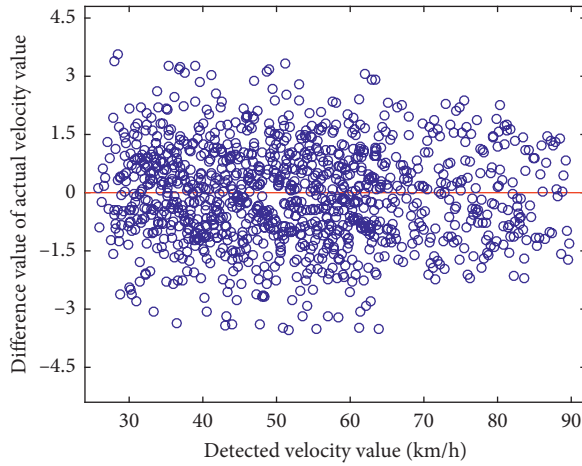


FIGURE 4: The error distribution.

discarded. In the meantime, requirements for safe driving forbids drivers from driving over 100 km/h under normal circumstances (although the electronic registration identification of motor vehicle readers/writers can accurately detect velocities up to 180 km/h), and therefore no tests were carried out over 100 km/h.

Using the same display method as in Figure 3, the actual velocity and the detected velocity are displayed for four lanes of the selected side of the road (Figure 5). Figures 5(a)–5(d) show, respectively, the velocity values of vehicles on the four lanes. The lane in Figure 5(a) is the closest to the center line of the road.

It can be seen from Figure 5 that, after the detection position of the electronic registration identification of the motor vehicle reader is fixed on the gantry, the detection results are also relatively fixed; detection in different lanes would not cause a large difference. However, since all the velocity values are detected by hardware, the velocity calculation method belongs to a generic algorithm. As long as its error is within a preset range, it meets the user requirement. For this reason, the error rate is still too high for certain specific applications. In addition, the distribution results of the detected velocity also showed that the errors of

the two lanes near the center line and the emergency lane are slightly higher than that of the middle two lanes. It can also be seen that the faster the velocity, the lower the error. And, the errors are slightly larger at lower velocities.

The tests are conducted on a selected middle lane, and the velocity detection results are calibrated using the following methods. The RBF neural network is based on particle swarm optimization, and multireader/writer velocity values are optimized with the RBF neural network based on PSO. The calibration results are shown in Figures 6–9.

As can be seen from Figures 6–9, the optimal output calibration result is the value of the multireader velocity with numerical optimization based on the particle swarm-optimized RBF neural network. Especially in the low-velocity stage, the detected velocity value can be better calibrated, and the velocity error value can be further reduced. The four scatter plots above show that, in the 25–40 km/h range, the detected velocity values and the actual velocity values are greater than those in other regions. The mean square error value for all data is 0.0965, and the Pearson coefficient $r = (\sum (x_i - \bar{x})(y_i - \bar{y}) / \sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2})$, a measure of correlation, is calculated to be 0.9301.

In order to further compare the differences in the calibration process of velocity detection in different lanes, we statistically analyzed all the test data. The statistical results are shown in Table 1. It should be noted that in the second column of the table, labeled “optimization algorithm,” “uncalibrated” represents data obtained without calibration, “Algorithm 1” represents the detected velocity output value calibrated by the RBF neural network, “Algorithm 2” represents the detected velocity output value calibrated by the RBF neural network based on particle swarm optimization, and “Algorithm 3” represents the detected velocity output value calibrated by the RBF neural network based on particle swarm optimization and on multireader velocity numerical optimization.

Table 1 lists the calibration results of the detected velocity values for different lanes using different algorithms. As far as the trend of calibration accuracy is concerned, the accuracy of the data from the one-way middle lane is relatively high. For detected velocity values calibrated with multireader numerical optimization based on the PS-optimized RBF neural network, the percentages of data with errors less than or equal to 5% are 92.39%, 92.10%, 91.41%, and 91.11% for lane nos. 2, 3, 6, and 7, respectively. The average percentage of data with errors less than or equal to 5% is 91.76% for the middle lane, 82.13% for the edge lane, and 87.12% for all lanes combined. In terms of actual applications, results with an error rate less than 10% have an application value, and the calibrated velocity detection data have met this requirement. A lower error rate not only is indicative of the method’s application value, but also reflects the real traffic condition more realistically. In the comprehensive statistical analysis of the error rate of the detected velocity value, the rate for the middle lane is lower than that for the edge lane. After calibration, the error values for different testing velocities are statistically consistent.

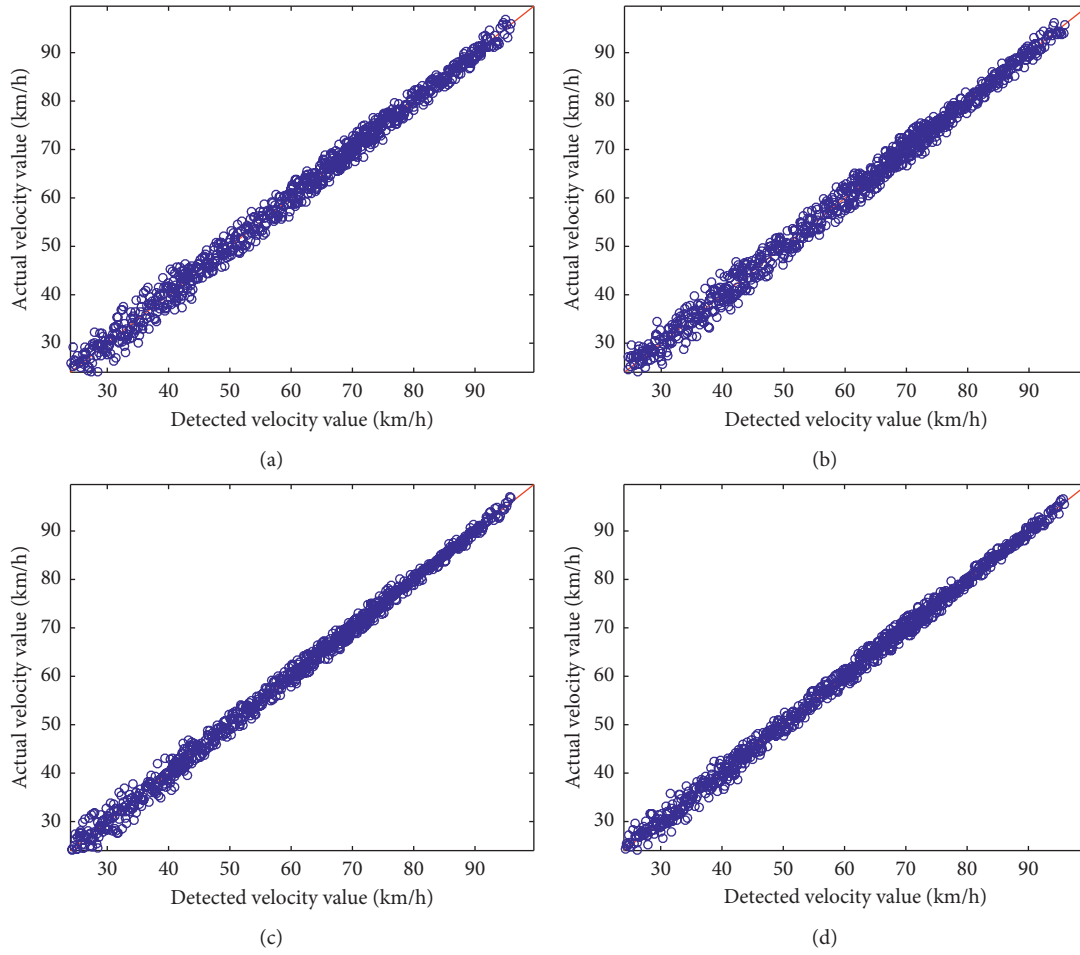


FIGURE 5: Actual velocity value and the detected velocity value for four different lanes.

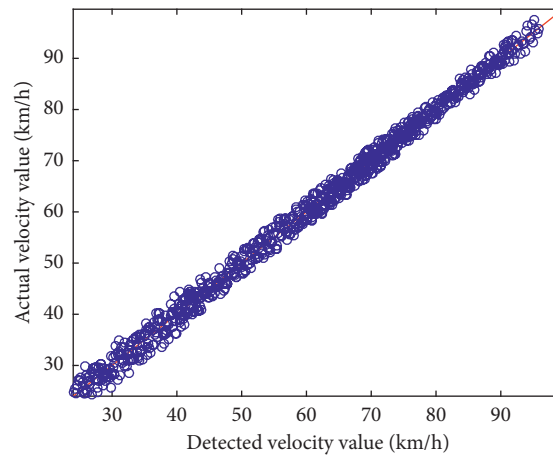


FIGURE 6: Actual velocity value and the detected velocity value for one lane.

We calculated the mean square error and the Pearson coefficient of the detected and calibrated velocity values with multireader numerical optimization based on the PS-optimized RBF neural network. From lane nos. 1 through 8, the mean square error values are 0.0856, 0.0653, 0.0623, 0.0841, 0.0901, 0.0679, 0.0688, and

0.0831, respectively; the corresponding Pearson coefficients are 0.9116, 0.9412, 0.9332, 0.9193, 0.9113, 0.9401, 0.939, and 0.9128, respectively. The mean square error value and the Pearson coefficient value both demonstrate that the velocity calibration value in the middle lane is accurate.

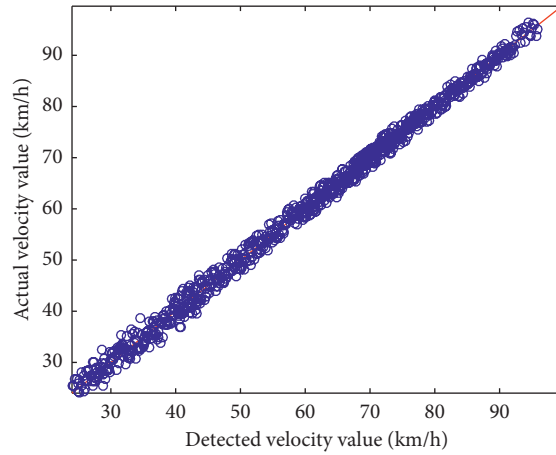


FIGURE 7: Actual velocity value and the detected velocity value calibrating based on the RBF neural network.

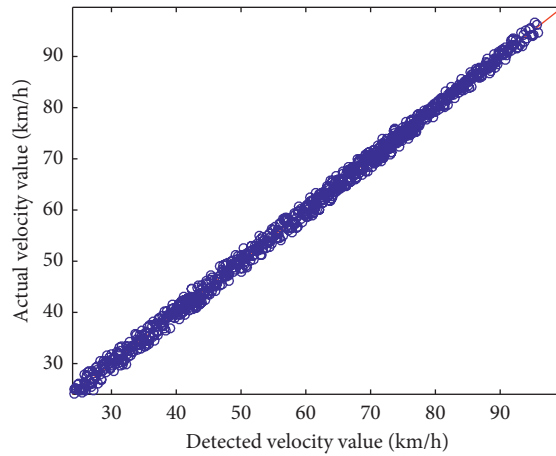


FIGURE 8: Actual velocity value and the detected velocity value calibrating based on the RBF neural network based on particle swarm optimization.

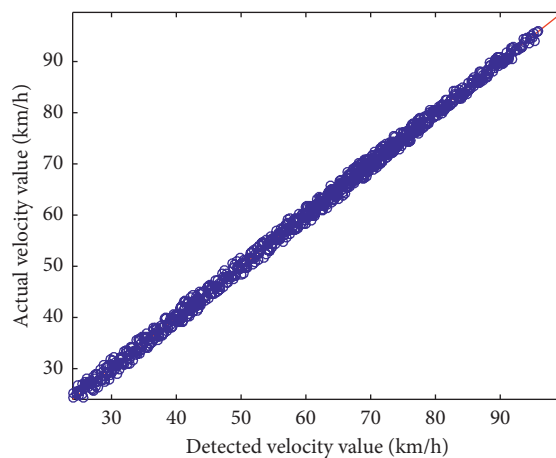


FIGURE 9: Actual velocity value and the detected velocity value calibrating based on numerical optimization of multireader velocity based on the particle swarm-optimized RBF neural network.

To further analyze the test results, we plot in Figure 10 the error values of the uncalibrated detected velocity values and compare them with the error values of the calibrated

detected velocity values using multireader numerical optimization based on the particle swarm optimization of the RBF neural network. The blue circle represents the error

TABLE 1: The calibration results of detected velocity values under different algorithms.

Lane number (number of samples)	Optimization method	Number of samples with error $\leq 1\%$	Proportion of samples with error $\leq 1\%$	Number of samples with error $\leq 3\%$	Proportion of samples with error $\leq 3\%$	Number of samples with error $\leq 5\%$	Proportion of samples with error $\leq 5\%$	Number of samples with error $> 5\%$	Proportion of samples with error $> 5\%$
Lane no. 1 (1256)	Uncalibrated	284	22.61	539	42.91	798	63.54	458	36.46
	Algorithm 1	302	24.04	674	53.66	878	69.90	378	30.10
	Algorithm 2	332	26.43	699	55.65	909	72.37	347	27.63
	Algorithm 3	386	30.73	759	60.43	1053	83.84	203	16.16
Lane no. 2 (1432)	Uncalibrated	332	23.18	683	47.70	995	69.48	437	30.52
	Algorithm 1	399	27.86	796	55.59	1092	76.26	340	23.74
	Algorithm 2	448	31.28	807	56.35	1245	86.94	187	13.06
	Algorithm 3	498	34.78	893	62.36	1323	92.39	109	7.61
Lane no. 3 (1494)	Uncalibrated	353	23.63	698	46.72	1002	67.07	492	32.93
	Algorithm 1	387	25.90	769	51.47	1104	73.90	390	26.10
	Algorithm 2	472	31.59	821	54.95	1265	84.67	229	15.33
	Algorithm 3	531	35.54	901	60.31	1376	92.10	118	7.90
Lane no. 4 (1379)	Uncalibrated	301	21.83	592	42.93	879	63.74	500	36.26
	Algorithm 1	344	24.95	673	48.80	953	69.11	426	30.89
	Algorithm 2	380	27.56	733	53.15	1034	74.98	345	25.02
	Algorithm 3	413	29.95	853	61.86	1143	82.89	236	17.11
Lane no. 5 (1402)	Uncalibrated	267	19.04	582	41.51	949	67.69	453	32.31
	Algorithm 1	309	22.04	672	47.93	1012	72.18	390	27.82
	Algorithm 2	350	24.96	749	53.42	1044	74.47	358	25.53
	Algorithm 3	401	28.60	829	59.13	1097	78.25	305	21.75
Lane no. 6 (1386)	Uncalibrated	341	24.60	603	43.51	1015	73.23	371	26.77
	Algorithm 1	422	30.45	692	49.93	1098	79.22	288	20.78
	Algorithm 2	478	34.49	795	57.36	1164	83.98	222	16.02
	Algorithm 3	503	36.29	832	60.03	1267	91.41	119	8.59
Lane no. 7 (1429)	Uncalibrated	332	23.23	586	41.01	998	69.84	431	30.16
	Algorithm 1	441	30.86	689	48.22	1086	76.00	343	24.00
	Algorithm 2	476	33.31	807	56.47	1239	86.70	190	13.30
	Algorithm 3	513	35.90	841	58.85	1302	91.11	127	8.89
Lane no. 8 (1308)	Uncalibrated	290	22.17	490	37.46	878	67.13	430	32.87
	Algorithm 1	304	23.24	572	43.73	954	72.94	354	27.06
	Algorithm 2	343	26.22	689	52.68	1003	76.68	305	23.32
	Algorithm 3	376	28.75	772	59.02	1097	83.87	211	16.13

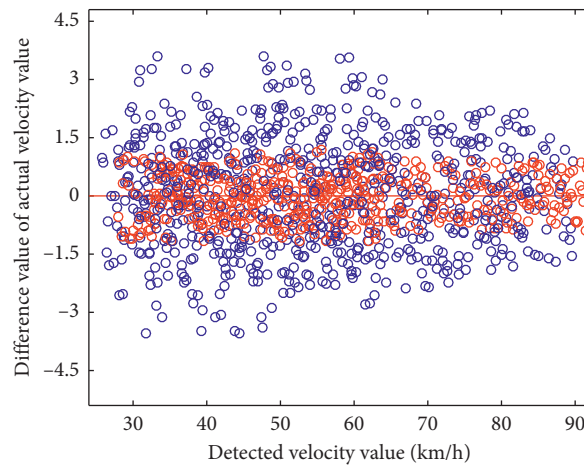


FIGURE 10: The calibration error distribution.

value of the uncalibrated detected velocity values and the red circle represents the error of the detected velocity values calibrated by the RBF neural network based on particle swarm optimization after multireader numerical optimization.

The scatter plot of the error distribution for the uncalibrated and calibrated detected velocity values shows that the algorithm proposed here can achieve accurate calibration of the velocity detected by electronic registration identification of motor vehicles and that the error value can be controlled within an acceptable range. In actual application, if the various influencing factors are fully considered, including the external environment, electromagnetic interference, windshield material, angle, different vehicle types, presence of film, decal placement location, and other interference factors, then the test results can accurately reflect the value of the real velocity, and the algorithm can provide data even closer to the real situation of practical application scenarios for improving traffic management efficiency such as in traffic rule enforcement, traffic status detection, traffic congestion index calculation, and navigation and parking guidance.

5. Conclusion

Based on vehicle velocity values collected by electronic registration identification of motor vehicles and by on-board (OBD) equipment and combined with actual application needs, we calibrated the detected velocity output value of electronic registration identification of motor vehicles using an RBF neural network, particle swarm optimization-based RBF neural network, and multireader numerical optimization based on a particle swarm-optimized RBF neural network and compared the data with that data collected by OBD. We then analyzed the errors. The method proposed here can greatly improve the accuracy of the velocity detected by the electronic registration identification of motor vehicles and make it closer to the real value through the use of an algorithm. The test results show that the results predicted by the method can more accurately reflect the actual driving velocity. The proposed method can be an important reference for improving the manufacture and application of electronic registration identification of motor vehicle software and hardware. In addition, since the two methods of data collection using electronic registration identification of motor vehicles and using on-board (OBD) devices both have their limitations, such as data acquisition cycle and time matching, the method proposed here is established on the velocity values selected from the same data by time matching. As a result, part of the time series data is incomplete, and less than 30% of all the data collected are used as test samples. With only 30% of the data, it is difficult to fit or predict velocity values using the time series method, and thus the data analyzed in this paper are slightly lacking in the time dimension. This would be one area to strengthen in future studies, and the focus should be on the fitting of velocity data with different time stamps. The fitted velocity detection data, together with the algorithm described here, would then be able to completely describe the vehicle

trajectory and provide important guidance for algorithm selection and model optimization.

Data Availability

The original data used to support the findings of this study are restricted by the relevant law enforcement departments in order to protect vehicle information privacy and law enforcement basis. Data are available from relevant law enforcement departments for researchers who meet the criteria for access to confidential data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Key Research and Development Program (nos. 2017YFD0700602, 2018YFB2003500, and 2018YFB1700200) and the Key Research and Development Plan of Shaanxi Province (2018ZDXM-GY-041).

References

- [1] X. Gu, J. Hu, B. Qian et al., "Application and study on electronic identification of motor vehicles in intelligent transport," *Journal of Highway and Transportation Research and Development*, vol. 34, no. S2, pp. 110–117, 2017.
- [2] H. Jiang, W. Fang, and J. Huang, "Design of parking management system based on the electronic registration identification," *China Science and Technology Information*, vol. 28, no. 23, pp. 78–79, 2016.
- [3] R. Wu and H. Yang, "Application and research of the electronic registration identification in traffic management field," *Police Technology*, vol. 7, no. 3, pp. 24–26, 2017.
- [4] Y. Tao, X. Yan, T. Wang et al., "Intelligent transportation system development strategy for future intelligent society," *Science & Technology Review*, vol. 37, no. 34, pp. 48–53, 2016.
- [5] W. Lang, S. Tian, Schwarcz et al., "A review of application of driving simulation technology in R&D of vehicular intelligence technology," *Journal of Highway and Transportation Research and Development*, vol. 34, no. 12, pp. 140–150, 2017.
- [6] C. Zuo, Z. You, and X. Niu, "The application of the electronic registration identification in public security traffic management," *Transportation Enterprise Management*, vol. 32, no. 1, pp. 91–93, 2017.
- [7] Z. L. Sun, "The application and practice of the electronic registration identification," *China Information Security*, vol. 37, no. 10, pp. 68–70, 2016.
- [8] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100–116, 2003.
- [9] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.
- [10] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.

- [11] M. Wu, S.-L. Dai, and C. Yang, "Mixed reality enhanced user interactive path planning for omnidirectional mobile robot," *Applied Sciences*, vol. 10, no. 3, p. 1135, 2020.
- [12] J. Luo, Z. Lin, Y. Li, and C. Yang, "A teleoperation framework for mobile robots based on shared control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 377–384, 2020.
- [13] H. Kong, C. Yang, G. Li, and S.-L. Dai, "A sEMG-based shared control system with no-target obstacle avoidance for omnidirectional mobile robots," *IEEE Access*, vol. 8, no. 1, pp. 26030–26040, 2020.
- [14] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1270–1293, 2008.
- [15] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.
- [16] L. Li, "Ranking-based multi-objective particle swarm optimization: research and application," Doctoral Dissertation of Zhejiang University of Technology, 2017.
- [17] W. Li, C. Yang, Y. Jiang, X. Liu, and C.-Y. Siu, "Motion planning for omni-directional wheeled mobile robot by potential field method," *Journal of Advanced Transportation*, vol. 2017, Article ID 4961383, 11 pages, 2017.
- [18] D. Zhao, H. Ma, S. Tang, and X.-Y. Li, "Coupon: a cooperative framework for building sensing maps in mobile opportunistic networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 392–402, 2015.
- [19] J. Yang, N. Zhang, M. Li et al., "Vehicle information influence degree screening method based on GEP optimized RBF neural network," *Complexity*, vol. 2018, Article ID 1067927, 12 pages, 2018.
- [20] L. Wang, J. Yang, N. Zhang et al., "Time-space relationship analysis model on the bus driving characteristics of different drivers based on the traffic performance index system," *Tehnicki Vjesnik-Technical Gazette*, vol. 25, no. 1, pp. 236–244, 2018.
- [21] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
- [22] J. Luo, C. Yang, N. Wang, and M. Wang, "Enhanced teleoperation performance using hybrid control and virtual fixture," *International Journal of Systems Science*, vol. 50, no. 3, pp. 451–462, 2019.
- [23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, WA, Australia, December 1995.
- [24] E. Zhang, "Research on multi-objective particle swarm optimization algorithm and applications," Doctoral Dissertation of Nanjing University of Science and Technology, 2016.
- [25] J. Luo, C. Liu, Y. Feng, and C. Yang, "A method of motion recognition based on electromyographic signals," *Advanced Robotics*, vol. 35, pp. 1–9, 2020.
- [26] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.

Research Article

Research on Discriminative Skeleton-Based Action Recognition in Spatiotemporal Fusion and Human-Robot Interaction

Qiubo Zhong ^{1,2}, Caiming Zheng,¹ and Haoxiang Zhang¹

¹Robotics Institute, Ningbo University of Technology, Ningbo 315211, China

²State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710054, China

Correspondence should be addressed to Qiubo Zhong; zhongqiubo@nbut.edu.cn

Received 18 June 2020; Revised 25 July 2020; Accepted 29 July 2020; Published 25 August 2020

Guest Editor: Hang Su

Copyright © 2020 Qiubo Zhong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel posture motion-based spatiotemporal fused graph convolutional network (PM-STGCN) is presented for skeleton-based action recognition. Existing methods on skeleton-based action recognition focus on independently calculating the joint information in single frame and motion information of joints between adjacent frames from the human body skeleton structure and then combine the classification results. However, that does not take into consideration of the complicated temporal and spatial relationship of the human body action sequence, so they are not very efficient in distinguishing similar actions. In this work, we enhance the ability of distinguishing similar actions by focusing on spatiotemporal fusion and adaptive feature extraction for high discrimination information. Firstly, the local posture motion-based attention (LPM-TAM) module is proposed for the purpose of suppressing the skeleton sequence data with a low amount of motion in the temporal domain, and the representation of motion posture features is concentrated. Besides, the local posture motion-based channel attention module (LPM-CAM) is introduced to make use of the strongly discriminative representation between different action classes of similarity. Finally, the posture motion-based spatiotemporal fusion (PM-STF) module is constructed which fuses the spatiotemporal skeleton data by filtering out the low-information sequence and enhances the posture motion features adaptively with high discrimination. Extensive experiments have been conducted, and the results demonstrate that the proposed model is superior to the commonly used action recognition methods. The designed human-robot interaction system based on action recognition has competitive performance compared with the speech interaction system.

1. Introduction

With the development of artificial intelligence technology, human-robot interaction technology has become a research hotspot. Compared with speech and image signals, vision-based human-robot interaction technology is more stable, and it attracts a lot of research interest. The key to human-centered visual interaction technology is to understand human activities [1] and human social behaviors [2]. Therefore, action recognition plays an important role in the field of human-robot interaction [3]. The two main approaches of human action recognition are RGB-based and skeleton-based. The RGB-based method makes full use of the image data and can obtain higher performance in the recognition rate. However, this method usually needs to process

every pixel in the image to extract features. Therefore, high-cost computing resources are required and real-time processing can hardly be achieved. It is also vulnerable to poor lighting conditions and background noise. In the skeleton sequence method, the 2D or 3D coordinates are expressed as human joint positions. Due to the limited number of joints in the human skeleton, only a few dozen, some modest computing resources would be enough for real-time applications. It is also robust to dynamic environments and complex backgrounds. Many widely available devices are suitable for extracting human skeleton features, such as Microsoft Kinect, OpenPose [4], and CPN [5].

The conventional deep learning-based methods convert the skeleton sequence as a set of joint vector sequences, input them to RNNs [6], or extract features by feeding 2D

pseudoimages representing skeleton sequences into CNNs [7], and then predict the action classes. However, neither the joint vector sequence nor the 2D pseudoimage can represent the correlation between human joints effectively. Recently, graph convolutional neural networks (GCNs) have extended the convolution operation from 2D image structure to graph structure and have shown good performance in many applications. Yan et al. [8] used GCNs for the first time in skeleton-based action recognition and proposed a spatial-temporal graph model. Subsequently, the methods for optimizing spatial feature extraction were proposed. Yang et al. [9] presented a finite-time convergence adaptive fuzzy control method for a dual-arm robot with an unknown number of kinematics and dynamics. Shi et al. [10] used adaptive graph convolutional layer and attention mechanism to increase the flexibility of the model, first-order joint information, second-order bone information, and motion information as inputs to construct multistream networks. Liu et al. [11] proposed multiscale aggregation across spatial and temporal dimensions effectively to eliminate the importance of neighbor nodes for long-range modeling. Yang et al. proposed a personalized variable gain control with tremor attenuation for robot teleoperation [12] and used adaptive parameter estimation and control design for robot manipulators with finite-time convergence [13]. Peng et al. [14] used high-order representations of skeleton adjacency graphs and dynamic graph modeling mechanisms to find implicit joint correlations. Obinata and Yamamoto [15] modeled the spatiotemporal graph by adding extra edges on the interframe to extract the relevant features of the human joints. However, all these methods ignore the fusion of posture motion and skeleton joint features in the temporal domain.

In the existing research work, the spatial information and motion information of the spatiotemporal graph are not fused to achieve end-to-end training effectively. The proposed novel posture motion-based spatiotemporal graph convolution networks (PM-STFGCNs) use the posture motion-based spatiotemporal fusion (PM-STF) module to perform feature fusion of motion and skeleton representation in the spatiotemporal domain for enhancing skeleton features adaptively. The defined local posture motion-based attention module (LPM-TAM) is used to constrain the disturbance information in the temporal domain and learn the representation of motion posture. The introduced local posture motion-based channel attention module (LPM-CAM) is employed to learn the strong discrimination representation between similar action classes in order to enhance the ability to distinguish fuzzy action. Extensive experiments have been performed on two large-scale skeleton datasets. Compared with common methods, the proposed method can further improve the recognition performance which combines with the method of optimizing the spatial graph convolution only. In addition, a human action recognition interactive system was designed to compare with speech interaction.

The main contributions of our methods are the following:

- (1) A novel local posture motion-based attention module (LPM-TAM) filters out low motion information in the temporal domain that helps to improve the ability of relevance motion feature extraction
- (2) Local posture motion-based channel attention module (LPM-CAM) is employed to enhance the ability to distinguish similar actions for learning the strong discriminative representation adaptively between different action classes
- (3) The posture motion-based spatiotemporal fusion (PM-STF) module is used, which integrates LPM-TAM and LPM-CAM to effectively fuse the spatiotemporal feature information and extract high-discriminative feature for improving the ability to distinguish similar actions
- (4) The effectiveness of the proposed method has been verified through extensive experiments, compared with other common methods to evaluate the competitiveness of the proposed method and applied in humanoid robots successfully to verify that action interaction is better than speech interaction

2. Related Work

2.1. Spatial Graph Convolution Networks. The spatial-temporal graph convolutional neural network [8] represents the connection relationship of the joints with the self-connected identity matrix I and the adjacency matrix A^S . In the case of a single frame, the convolution operation of the spatial dimension is performed as follows:

$$f_{\text{out}} = \sum_{k=1}^K (\Lambda_k^{-1/2} A_k^S \Lambda_k^{-1/2} \otimes A_S) f_{\text{in}} M_k, \quad (1)$$

where $f_{\text{in}} \in \mathbb{R}^{C_{\text{in}} \times T \times N}$ is the feature map with input dimension of (C_{in}, T, N) tensor, N is the number of joints, A_k^S is the $N \times N$ adjacency-like matrix, $A_k^{S_{i,j}} = 1$ denotes the vertex v_i in the subset of the vertex v_j , and $\Lambda_k^{ii} = \sum_j (A_k^{S_{ij}}) + \lambda$ is the normalized diagonal matrix, where $\lambda = 0.001$. K represents the numbers of different subsets in spatial dimension based on spatial distance partition strategies. There are three different subsets, namely, $K = 3$. A_0^S represents the connection of the vertex itself, A_1^S represents the connection of the centripetal subset, and A_2^S represents the connection of the centrifugal subset. $M_k \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times 1 \times 1}$ is 1×1 convolution weight. A_S is a spatial attention feature map of the $N \times N$ dimension, which denotes the importance of each joint. \otimes is the multiplication of the corresponding elements of the matrix, which means that it can only affect the vertices connected to the current target.

2.2. Temporal Graph Convolutional Networks. The literature [10] proposed a temporal attention module, and the attention coefficient is calculated as follows:

$$A_T = \sigma(\theta(\text{AvgPool}(f_{\text{in}}))), \quad (2)$$

where $\mathbf{f}_{in} \in \mathbb{R}^{C_{in} \times T \times N}$ is an input feature map. AvgPool is an average pooling operation. θ is a 1×1 convolution operation, and the weight matrix $\mathbf{M}_\theta \in \mathbb{R}^{1 \times C_{in} \times S}$, where S is the size of the convolution kernel. σ refers to the Sigmoid activation function. The attention feature map $\mathbf{A}_T \in \mathbb{R}^{1 \times T \times 1}$, which denotes the importance of the skeleton graph at a temporal dimension, and T refers to the length in time.

The literature [8] defined the temporal graph convolution based on a simple strategy. In equation (1), they use the kernel size $\Theta \times 1$ in the temporal dimension to perform graph convolution. Therefore, the sampling area on the vertex v_{ti} is $S_a(v_{ti}) = \{v_{qi} \mid |q - t| \leq \lfloor \Theta/2 \rfloor\}$, where Θ is the kernel size in temporal dimension, which is set to 9 in [8].

3. Posture Motion-Based Spatiotemporal Fusion Graph Convolution

3.1. Posture Motion Representation. The posture motion represents the motion information of the corresponding joint in a series of consecutive frames, for example, the j _{th} joint of the given frame $u + 1$, i.e., $\mathbf{v}_{u+1,j} = (x_{u+1,j}, y_{u+1,j}, z_{u+1,j})$ and the j _{th} joint of frame u , i.e., $\mathbf{v}_{u,j} = (x_{u,j}, y_{u,j}, z_{u,j})$, which posture motion is represented as $\tau_{u,j} = (x_{u+1,j} - x_{u,j}, y_{u+1,j} - y_{u,j}, z_{u+1,j} - z_{u,j})$. $\tau_{u,j}$ is the posture motion representation of the j _{th} joint of frame u .

3.2. Local Posture Motion-Based Temporal Attention Module. A novel local posture motion-based temporal attention module (LMP-TAM) is proposed for suppressing a large amount of disturbance information in the temporal dimension. As shown in Figure 1, the posture motion feature map of each vertex in the spatiotemporal graph is calculated as follows:

$$\Omega = \sum_{t=1}^T \theta_t(\varepsilon_m(\mathbf{f}_{in}^t, \mathbf{f}_{in}^{t-1})), \quad (3)$$

where $\mathbf{f}_{in}^t \in \mathbb{R}^{C_{in} \times 1 \times N}$ is the input feature map at time t . $\theta_t \in \mathbb{R}^{C_{in}/2 \times C_{in} \times 1 \times 1}$ is the convolution weight matrix of 1×1 . ε_m extracted the posture motion representation from the input feature map. $\Omega \in \mathbb{R}^{C_{in}/2 \times T \times N}$ is the posture motion feature map, where the channel of the feature map is half of the input channel.

Human motion is body movements, which involve part or all of the limbs. The attention map Φ_T of skeleton sequence in the spatiotemporal graph is represented by the attention Φ_L of local limbs in the temporal dimension. The importance of local limb in temporal dimension is determined by motion information in the local perception domain D . $D = \{d_0, d_1, d_2, d_3, d_4\}$, where d_0 denotes left hand, d_1 denotes right hand, d_2 denotes left leg, d_3 denotes right leg, and d_4 denotes other limb parts. $\Phi_L \in \mathbb{R}^{1 \times T \times \eta}$, where η refers to the number of limbs being denoted, and η has been set 5 in this work. The temporal attention of local posture motion-based is calculated as follows:

$$\Phi_T = \sigma(\text{AvgPool}(\text{MaxPool}(\Gamma \otimes \Omega))), \quad (4)$$

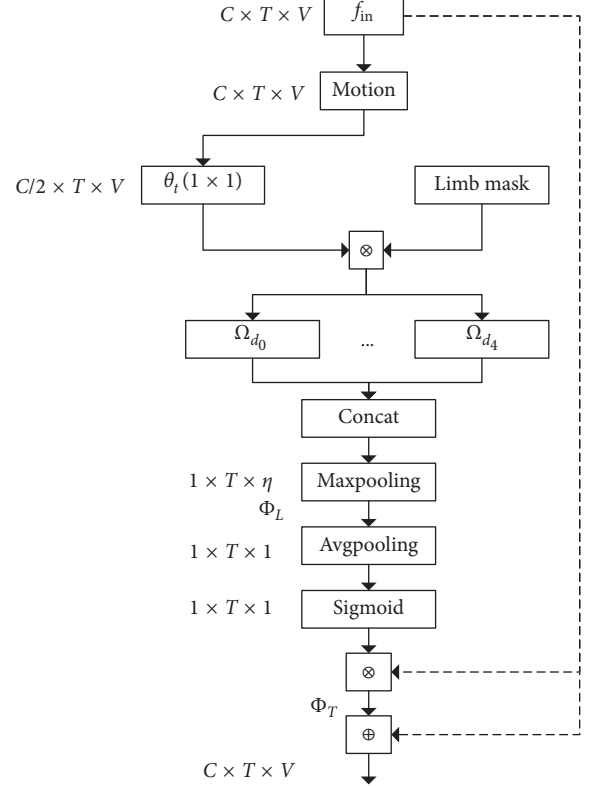


FIGURE 1: Local posture motion-based temporal attention (LMP-TAM) module.

where $\Phi_T \in \mathbb{R}^{1 \times T \times 1}$ refers to the importance of each frame of the spatiotemporal graph with a time length of T . Ω is the motion feature map of joints, and $\Gamma \in \mathbb{R}^{1 \times 1 \times V}$ is the local limb mask set D . $\Gamma \otimes \Omega$ is the attention Φ_L based on local limbs in the temporal dimension. \otimes is the multiplication of the corresponding elements of the matrices. σ is the sigmoid activation function. The final output is as follows:

$$\mathbf{f}_{out} = \mathbf{f}_{in} \otimes \Phi_T \otimes \mathbf{f}_{in}. \quad (5)$$

The input feature map is multiplied by the attention feature map Φ_T in a residual manner to calculate adaptive feature enhancement, and \otimes refers to the addition of corresponding matrix elements.

3.3. Local Posture Motion-Based Channel Attention Module. The local posture motion-based channel attention module (LPM-CAM) has been proposed to improve the ability to learn the strong discrimination representation between different postures. As shown in Figure 2, the input includes the posture motion feature map $\Omega \in \mathbb{R}^{C \times T \times V}$ extracted based on the local posture motion-based temporal attention module and generated temporal attention, which multiplied of each other to obtain the spatial-temporal graph after attention. The temporal sequence action segment with rich action semantic information is paid more attention. The channel attention coefficient is calculated as follows:

$$\Phi_C = \sigma(\text{AvgPool}(\text{MaxPol}(\rho(\Gamma \otimes (\Omega \otimes \Phi_T))))). \quad (6)$$

The motion feature map after attention is denoted as $\Omega \otimes \Phi_T$, which is decomposed into several local limbs in the local perception field to represent local posture movement. The action sequence with important semantic on the spatial-temporal graph has been screened out by the temporal attention module, and the channel attention selects the strong discriminative representations between different posture movements for action recognition. ρ is marked as ReLU nonlinear activation function. Concat refers to concatenate the local limb feature map:

$$\mathbf{f}_{\text{out}} = \mathbf{f}_{\text{in}} \otimes \Phi_C \otimes \mathbf{f}_{\text{in}}. \quad (7)$$

The input feature map is multiplied by the channel attention feature map in way of residual connection to achieve adaptive feature enhancement.

3.4. Posture Motion-Based Spatiotemporal Fusion. In order to fuse skeleton joints information and motion features to achieve an end-to-end learning manner, the posture motion-based spatiotemporal fusion module (PM-STF) is proposed to fuse spatial and temporal features and enhance the discriminative feature adaptively. The output of temporal convolution module at the i_{th} vertex of frame u is

$$f_{\text{out}}(v_i) = f_{\text{in}}(v_i) + \sum_{v_j \in S'_a(v_i)} \frac{1}{Z_{ui}(v_j)} \tau(v_j) \cdot \delta(\gamma_j(v_j)). \quad (8)$$

This is different from formula (1), and the input is a posture motion feature map extracted from the spatio-temporal graph and adopts a residual connection to enhance the motion feature. $\tau(v_j)$ is the posture motion feature of the neighborhood vertex v_j . δ is the weighting function. $\gamma_j(v_j)$ refers to the mapping label of the subset of the neighborhood vertex v_j , which is divided into three subsets S'_{a0} , S'_{a1} , and S'_{a2} based on the spatial distance partition strategy.

To implement the PM-STF, equation (8) is transformed into

$$\mathbf{f}_{\text{out}} = \mathbf{f}_{\text{in}} \oplus \sum_{k=1}^K (\Lambda_k^{-1/2} \mathbf{A}_k^S \Lambda_k^{-1/2} \otimes \mathbf{A}_S) (\mathbf{M}_m \Omega) \mathbf{M}_k, \quad (9)$$

where $\Omega \in \mathbb{R}^{C_{\text{in}}/2 \times T \times N}$ is posture motion feature map and $\mathbf{M}_m \in \mathbb{R}^{C_{\text{in}} \times C_{\text{in}}/2 \times 1 \times 1}$ is a 1×1 convolution weight matrix, increasing the channel of the same posture motion feature map as input channel. $\mathbf{M}_k \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times 1 \times 1}$ is a 1×1 convolution weight vector. $\mathbf{A}_S \in \mathbb{R}^{1 \times T \times N}$ is a spatial attention map which is used to distinguish the importance of vertices. \otimes refers to the multiplication of the corresponding elements of matrices. \mathbf{A}_k^S is adjacency-like matrix, and

$$\mathbf{A}_k^{S_i, j} = \begin{cases} 1, & \text{if vertex } v_j \text{ in the subset } S'_{ak} \text{ of vertex } v_i, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

3.5. Implementation of PM-STFGCN. The implementation of our module is combined with the model of optimizing the spatial graph convolution only, such as ST-GCN and

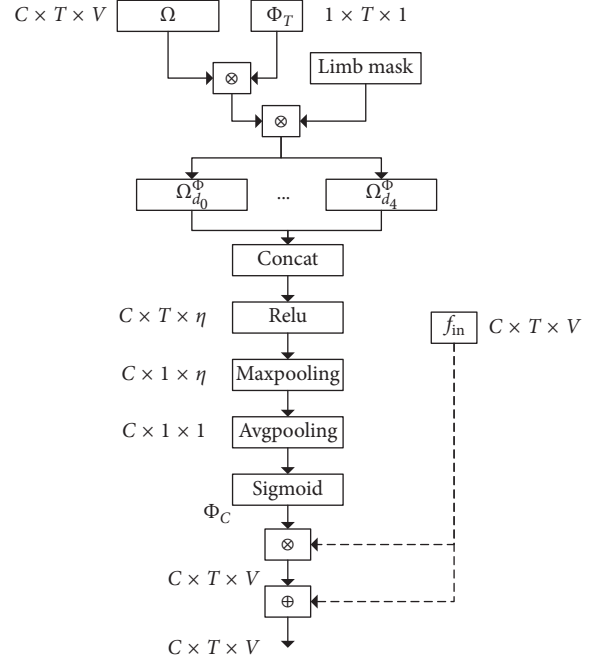


FIGURE 2: Local posture motion-based channel attention module (LMP-CAM).

2s-AGCN. Taking ST-GCN as an example, shown in Figure 3, the implementation of our module PMSTF-GCN is added between S-GCN and T-GCN. Each layer of PMSTF-GCN contains LPM-TAM, LMP-CAM, and PM-STF. S-GCN and T-GCN are named as the spatial graph convolution layer and temporal graph convolution layer of the original model. GAP is a global average pooling layer, and FCN is marked as a fully connected network layer. Finally, a spatiotemporal fusion graph convolution block is constructed. The overall architecture of the network consists of several STFGCN blocks. The batch normalization layer is added to the skeleton data input to normalize the input data. Finally, the global average pooling layer is implemented to pool the feature graphs to the same size, and the followed layer is a SoftMax classifier to obtain the prediction.

3.6. Implementation in Human-Robot Interaction. The presented action recognition schemes were applied on a real system, which consists of a Pepper robot and an external Kinect v2 depth camera. The implementation in human-robot interaction was performed as follows (Algorithm 1).

4. Experiments

4.1. Datasets

4.1.1. NTU-RGB + D. NTU-RGB + D [16] is the largest and most widely used multimodality dataset for skeleton-based action recognition. Each action segment was performed by 40 volunteers aged 10 to 35 and captured by three camera sensors at the same height but from different horizontal

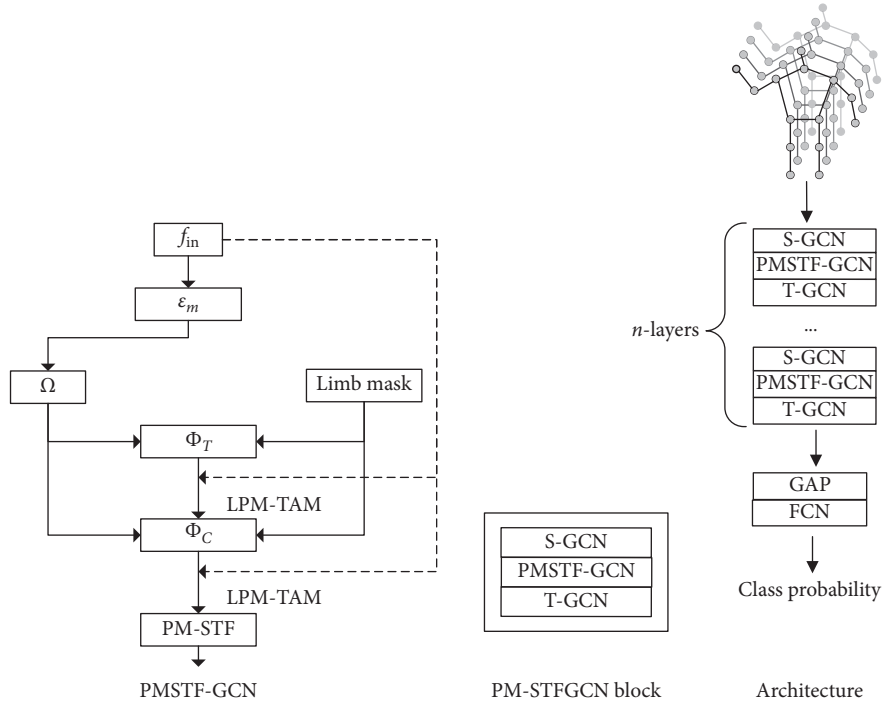


FIGURE 3: ST-GCN based spatiotemporal fusion graph convolution neural networks.

- (1) Initialize action recognition network f
- (2) Initialize threshold $\alpha = 0.75$
- (3) **while**
- (4) obtain the skeleton data series D from Kinect v2
- (5) $C_{class}, P_{probability} \leftarrow \arg \max_{probability} (f(D))$
- (6) **if** $P_{probability} \geq \alpha$
- (7) digital instruction $I \leftarrow C_{class}$
- (8) pass the instruction I to Pepper robot
- (9) **end**

ALGORITHM 1: Human-robot action interaction.

angles: -45° , 0° , and 45° . The human skeleton has 25 joint points, and the number of skeletons in each video is no more than 2. It contains 60 action classes and 56880 video clips. There are two kinds of training benchmarks, and [16] recommends as follows: cross-subject (CS) and cross-view (CV). In cross-subject (CS) benchmarks, the training dataset contains 40320 action samples, and the testing dataset contains 16560 action samples. In cross-view (CV) benchmarks, the training dataset contains 37920 action samples taken by camera sensors 2 and 3, and the testing dataset contains 18960 action samples taken by camera sensors 1. In the following experiments, we test the top-1 accuracy on two benchmark datasets.

4.1.2. Kinetics-Skeleton. Kinetics-Skeleton [17] is a large dataset for skeleton-based action recognition. Kinetic contains 300000 action video clips, and a total of 400 classes [8]

used the publicly available OpenPose toolbox [4] to estimate the pose of 18 joints in each fragment frame. There are a total of 300 frames for each action video frame. According to the average joint confidence, two people are selected as multi-person clips in each frame. The training dataset contains 240000 video clips, and the testing dataset contains 20000 video clips. We make use of the training dataset and then perform experiments to verify the accuracy of top-1 and top-5 on the testing dataset.

4.2. Ablation Study. The effectiveness of the proposed method has been verified over two large skeleton datasets in Kinetics-Skeleton and NTU-RGB+D. The local posture motion-based attention module (LPM-TAM), local posture motion-based channel module (LPM-TAM), and posture motion-based spatiotemporal fusion (PM-STF) module are represented by PM-STFGCN. Two sets of comparisons are

made between ST-GCN [8] and ST-GCN + PM-STFGCN, and between 2s-AGCN [18] and 2s-AGCN + PM-STFGCN. The results show that the performance has been improved over the original models and verified the effectiveness of LPM-TAM, LPM-CAM, and PM-STF.

As shown in Tables 1 and 2 for ST-GCN [8] and ST-GCN + PM-STFGCN, PM-STFGCN improves the top-1 accuracy of the CS and CV benchmarks by 4.2% and 1.6%, respectively, and the accuracy of the top-1 and top-5 of the Kinetics-Skeleton dataset by 2.5% and 1.9%, respectively. For 2s-AGCN [18] and 2s-AGCN + PM-STFGCN, PM-STFGCN improves the top-1 accuracy of CS and CV benchmarks by 3.3% and 1.3%, respectively, and the accuracy of the top-1 and top-5 of the Kinetics-Skeleton dataset has been improved by 1.4% and 2.0%, respectively. 2s-AGCN + PM-STFG performed best on NTU-RGB + D and Kinetics-Skeleton datasets.

4.2.1. Attention Module. Experiments were also performed to verify the local posture motion-based temporal attention module (LPM-TAM) and channel attention module (LPM-CAM). In the ST-GCN [8] and 2s-AGCN [18] networks, only the LPM-TAM or LPM-CAM is added to the convolutional layer of the spatial-temporal graph. The results are shown in Tables 1 and 2. Compared with ST-GCN, the LPM-TAM module improves the top-1 accuracy of the CS and CV benchmarks by 2.6% and 0.6%, respectively, and the accuracy of top-1 and top-5 of Kinetics-Skeleton by 1.0% and 0.9%, respectively. The LPM-CAM module improved the top-1 accuracy of the CS and CV benchmarks by 3.0% and 0.7%, respectively, and the accuracy of top-1 and top-5 of Kinetics-Skeleton by 1.4% and 1.1%, respectively. Compared with 2s-AGCN, the LPM-TAM module improves the top-1 accuracy of the CS and CV benchmarks by 1.7% and 0.5%, respectively, and the accuracy of the top-1 and top-5 of Kinetics-Skeleton by 0.4% and 0.9, respectively. The LPM-CAM module improves the top-1 accuracy of the CS and CV benchmarks by 2.1% and 0.7%, respectively, and the accuracy of the top-1 and top-5 of Kinetics-Skeleton by 0.5% and 1.1%, respectively. The temporal and channel attention module improved the recognition performance than the original model which verifies the effectiveness of the feasibility of the attention modules.

4.2.2. Spatiotemporal Fusion Module. Experiments were also carried out on the posture motion-based spatiotemporal fusion (PM-STF) module. In the ST-GCN [8] and 2s-AGCN [18] networks, only the PM-STF is added to the convolutional layer of the spatial-temporal graph. The results are shown in Tables 1 and 2. Compared with ST-GCN, the PM-STF module improved the top-1 accuracy of CS and CV benchmarks by 3.2% and 0.9%, respectively, and the accuracy of top-1 and top-5 of Kinetics-Skeleton by 1.6% and 1.2%, respectively. Compared with 2s-AGCN, the PM-STF module improves the top-1 accuracy of the CS and CV benchmarks by 2.5% and 0.7%, respectively, and the accuracy of the top-1 and top-5 of the dataset Kinetics-Skeleton

TABLE 1: Ablation study on the benchmark of NTU-RGB + D.

Methods	CS (%)	CV (%)
ST-GCN [8]	81.5	88.3
2s-AGCN [18]	88.6	95.2
ST-GCN + LPM-TAM (ours)	84.1	88.9
2s-AGCN + LPM-TAM (ours)	90.3	95.7
ST-GCN + LPM-CAM (ours)	84.5	89.0
2s-AGCN + LPM-CAM (ours)	90.7	95.9
ST-GCN + PM-STF (ours)	84.7	89.2
2s-AGCN + PM-STF (ours)	91.1	95.9
ST-GCN + PM-STFGCN (ours)	85.7	89.9
2s-AGCN + PM-STFGCN (ours)	91.9	96.5

TABLE 2: Ablation study on the skeleton-based dataset Kinetics-Skeleton.

Methods	Top-1 (%)	Top-5 (%)
ST-GCN [8]	32.5	54.9
2s-AGCN [18]	36.7	59.8
ST-GCN + LPM-TAM (ours)	33.5	55.8
2s-AGCN + LPM-TAM (ours)	37.1	60.7
ST-GCN + LPM-CAM (ours)	33.9	56.0
2s-AGCN + LPM-CAM (ours)	37.2	60.9
ST-GCN + PM-STF (ours)	34.1	56.1
2s-AGCN + PM-STF (ours)	37.5	61.1
ST-GCN + PM-STFGCN (ours)	35.0	56.8
2s-AGCN + PM-STFGCN (ours)	38.1	61.8

by 0.8% and 1.3%, respectively. Compared with the original model, the spatiotemporal fusion module has a greater contribution to the improvement of the recognition performance which verifies the effectiveness and necessity of spatiotemporal fusion.

4.3. Comparison with State-of-the-Art Schemes. The proposed method is compared with some of the state-of-the-art schemes, and the results are shown in Tables 3 and 4. Among them, 2s-AGCN + PM-STFGCN achieved very good performance on CS and CV. On the Kinetics-Skeleton dataset, the accuracy of top-1 and top-5 of 2s-AGCN + PM-STFGCN also showed decent performance.

4.4. Human-Robot Interaction Demonstration. To further evaluate the robustness of the proposed action recognition schemes to distinguish similar action classes, action recognition is applied to a real system that consists of a Pepper robot and an external Kinect v2. As shown in Table 5, there is a correspondence between action semantics and interactive action.

The designed correspondence between action semantic and interactive activities ranges from partial and limb movements of hands to whole-body movements with more complexity. For example, waving the hand, touching the ear, holding the head with hands, and applauding are all hand movements. Among them, the hand movements of the first three movements are related to the head with high similarity. Also, squatting, sitting

TABLE 3: Comparison of CS and CV benchmarks with state-of-the-art schemes.

Methods	CS (%)	CV (%)
STA-LSTM [6]	73.4	81.2
ST-GCN [8]	81.5	88.3
CNN-based [7]	83.2	89.3
GCN-NAS [14]	89.4	95.7
MS-AAGCN [10]	90.0	96.2
2s-AGCN + PM-STFGCN (ours)	91.9	96.5

TABLE 4: Comparison with Kinetics-Skeleton dataset with state-of-the-art schemes.

Methods	Top-1 (%)	Top-5 (%)
ST-GCN [8]	30.7	52.8
GCN-NAS [14]	37.1	60.1
MS-AAGCN [10]	37.4	60.6
2s-AGCN + PM-STFGCN (ours)	38.1	61.8

TABLE 5: Correspondence between action semantic and interactive action.

Action semantics	Interactive action
Wave the hand	Raise the hand
Touch the ear	Nod the head
Hold the head	Look left
Applaud	Look right
Squat	Sit down
Sit down	Wake up
Jump up	Stand up

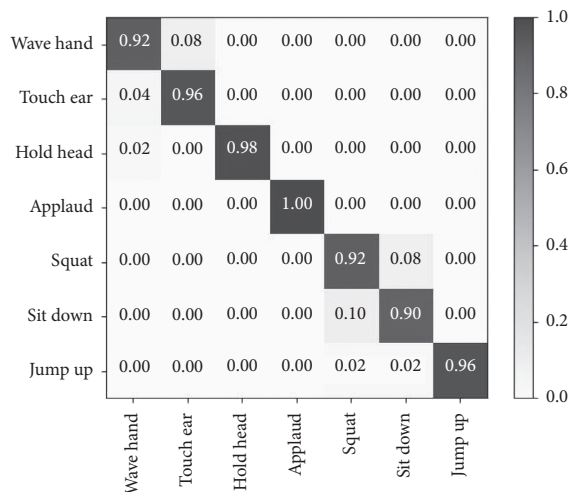


FIGURE 4: The confusion matrix of action interaction with the Pepper robot.

down, and jumping involve movements of the whole body with high similarity.

As shown in Figure 4, the measurement results are obtained in this work after conducting 50 experiments.

Among them, each similar action has a high recognition accuracy which means our method can effectively distinguish each different action. Each action sequence can be seen as a combination of many steps. For example, waving the hand can be divided into two steps: first, raise your right hand to above the head; second, swing the hand around the head. Similarly, a video can be decomposed into multiple frames of images.

4.4.1. Strong Discrimination Analysis. As shown in Figures 5–8, there are examples of human-robot interaction with similar actions. A period of time action sequence has been calculated, and the classification result with the highest probability is selected as the recognized result. Skeleton sequence with low motion information can be filtered out well by LPM-TAM, which helps to identify the process from raising hands to the head and swinging, and more purposefully recognize interactive actions. The main action of touching the ear is the process of raising the hand to the ear. Compared with waving the hand, the main difference is the movement of the hand swinging near the head. The characteristics of strong discrimination have been paid more attention by LPM-CAM to constraint similar movement processes, such as the process of raising the hand which serves as the basis for action recognition. The action of holding the head with both hands is similar to touching the ears. However, the main difference is that holding the head with both hands is the movement of the left and right hands, while touching the ears is the movement of the limbs with one hand. The main difference between similar movements in the local limb area can be captured by LPM-CAM effectively that enables the proposed method to extract stronger and discrimination representation. The human-robot interaction experiments verified that similar action did not affect the recognition result at all and has a strong discrimination of similar actions.

4.4.2. Comparison with Speech Interaction. In this work, the two indicators of accuracy and real-time performance are compared with speech interaction. The accurate times of these interaction methods were recorded 50 times to verify the reliability of the action interaction. Figure 9 shows the confusion matrix of the Pepper robot speech interaction recognition. In the testing phase, it only needs to speak out the corresponding action, such as wave the hand or touch the ear. The recognition result is regarded as “jumping” if the result of speech interaction has not been recognized within the specified test time. The recognition result of speech interaction is easily affected by external noise and distance which cause recognition errors, or no recognition results. From the experimental results, the average recognition rate of action interaction and speech interaction is 95.7% and 94.8%, respectively. Compared with speech interaction, our scheme has highly competitive which verifies the reliability of the action interaction in the recognition effect.

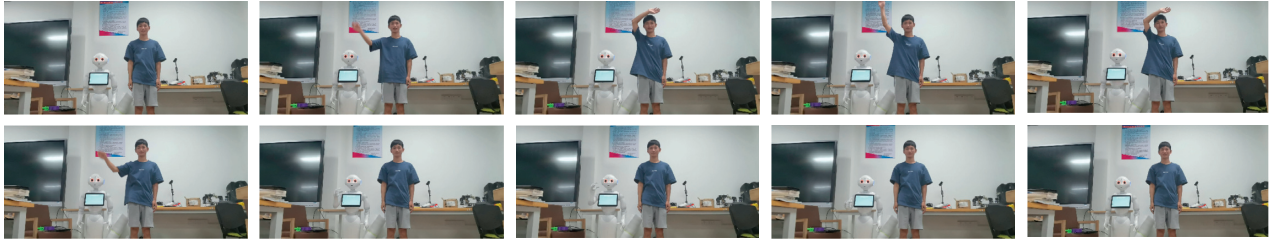


FIGURE 5: Action interaction with waving the hand as an example.

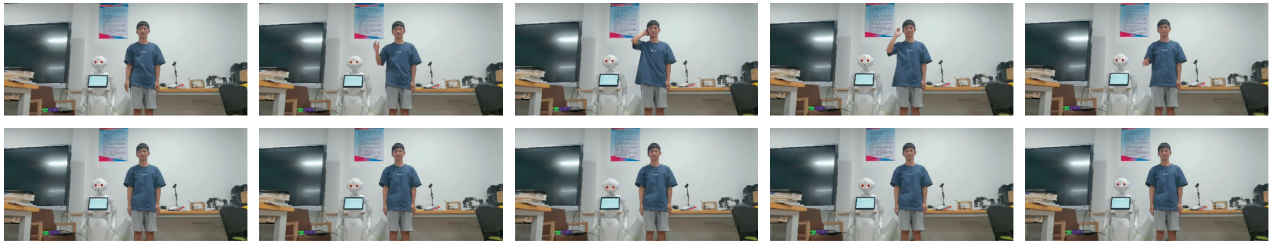


FIGURE 6: Action interaction with touching the ear as an example.

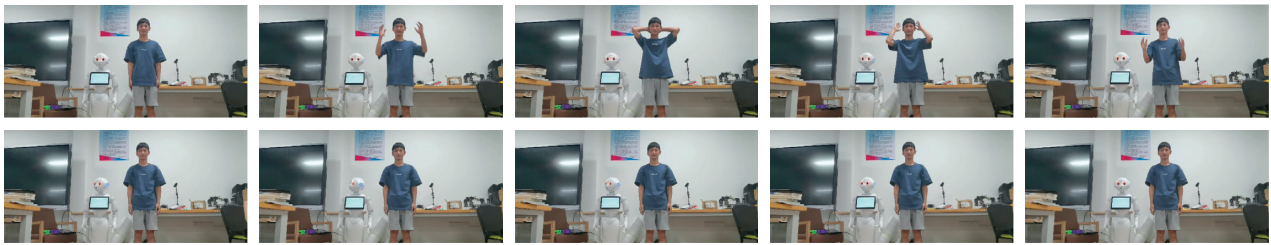


FIGURE 7: Action interaction with holding hands as an example.

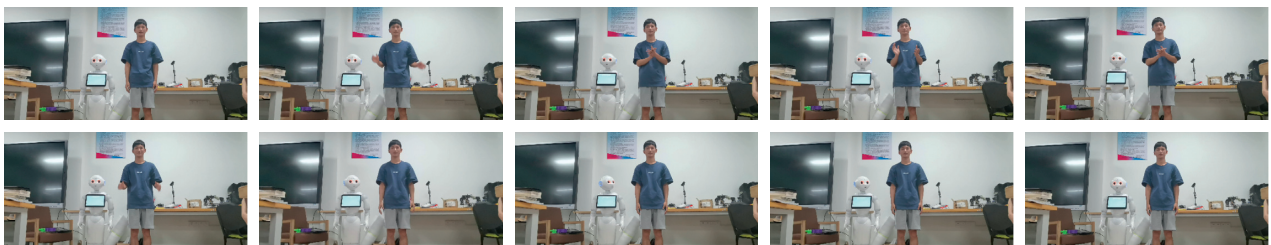


FIGURE 8: Action interaction with applause as an example.

4.4.3. *Comparison of Response Time.* As shown in Figure 10, comparison with the response time of speech and action interaction shows the average time of the 10 test results. Due to the different durations of each action, using the same time segment as inputs will cause fluctuations of response time. We try to do a few more experiments to eliminate the differences among the action response time. The results show that the response time of action interaction is shorter than speech interaction because of the robustness to external environment noise. The average response time of speech and action interaction

is 2.05 s and 1.86 s, respectively. Compared with speech interaction, the proposed scheme reduced the responding time by 0.19 s in real-time. The main reason is that video frames within a certain time range are used for recognition and shorter processing time for the action recognition network.

In conclusion, through the experimental comparison of two human-robot interaction ways, the action recognition has its advantages: it is not affected by environmental noise or spatial distance; it provides better real-time response during the interaction.

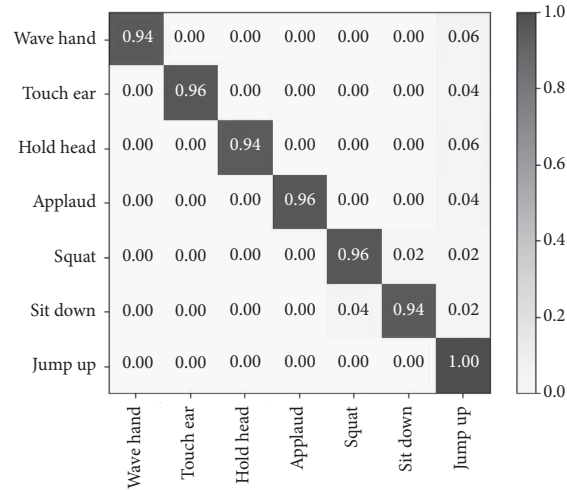


FIGURE 9: The confusion matrix of speech interaction with the Pepper robot.

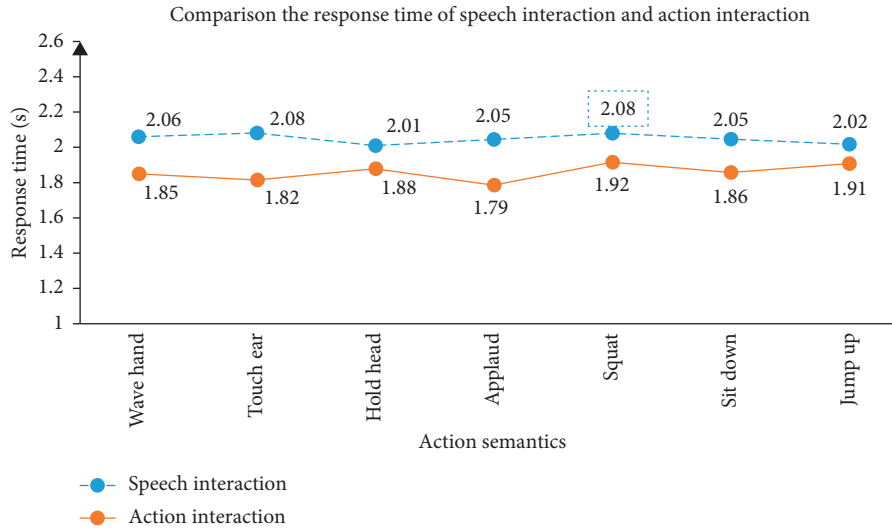


FIGURE 10: Comparison of the response time of speech and action interaction.

5. Conclusion

Previous works in the literature mostly make use of modeling of motion information and skeleton joint information independently, which cannot fully express the relationship between them. The posture motion-based spatiotemporal fusion graph convolution network (PM-STFGCN) is presented to fuse temporal and spatial features and enhance the posture motion features adaptively with high discrimination. A novel local posture motion-based temporal attention (LPM-TAM) module is introduced to suppress the disturbance information with low motion in the temporal domain efficiently and fully learn the representation of the posture motion. The local posture motion-based channel attention module (LPM-CAM) is proposed for the purpose of learning strong discrimination representation between different motion postures which improved the ability to discriminate

action classes, and the posture motion-based spatiotemporal fusion module (PM-STF) is adopted to fuse the motion feature and skeleton representation effectively. Extensive experiments were performed on two large skeleton datasets, and the constructed scheme shows substantial improvement over some other methods. The proposed action recognition interaction system has a competitive performance in accuracy and response time compared with speech interaction.

Data Availability

The data used to support the findings of this study are included with the supplementary information files.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This material is based upon work funded by the State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University Foundation of China, under grant no. sklms2019011, "13th Five-Year Plan" Talent Training Project of Higher Education in Zhejiang Province under grant no. jg20190487, and Research Project of Educational Science Planning in Zhejiang Province under grant no. 2020SCG090.

References

- [1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014.
- [2] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: end-to-end multi-person action localization and collective activity recognition," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [3] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3551–3558, Sydney, Australia, December 2013.
- [4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [5] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [6] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3d human action recognition," *Computer Vision-ECCV 2016*, vol. 9907, pp. 816–833, 2016.
- [7] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops ICMEW*, Hong Kong, China, July 2017.
- [8] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018, <http://arxiv.org/abs/1801.07455v2>.
- [9] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.
- [10] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," 2019, <https://arxiv.org/abs/1912.06971>.
- [11] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," 2020, <https://arxiv.org/abs/2003.14111>.
- [12] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.
- [13] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.-Y. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1759–1770, 2018.
- [14] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 3, pp. 2669–2676, 2020.
- [15] Y. Obinata and T. Yamamoto, "Temporal extension module for skeleton-based action recognition," 2020, <https://arxiv.org/abs/2003.08951>.
- [16] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB + D: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019, Las Vegas, NV, USA, June 2016.
- [17] W. Kay, J. Carreira, K. Simonyan et al., "The kinetics human action video dataset," 2017, <https://arxiv.org/abs/1705.06950>.
- [18] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.

Research Article

Adaptive Fuzzy Control for Attitude Stabilization of Spacecraft with Deployable Composite Laminated Solar Array

Wei Zhang,¹ Weibing Zhu ,¹ Shijie Zhang,² and Xiangtian Zhao²

¹College of Aerospace and Civil Engineering, Harbin Engineering University, Harbin 150001, China

²School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Weibing Zhu; zhuweibing@hrbeu.edu.cn

Received 15 April 2020; Accepted 11 June 2020; Published 4 August 2020

Guest Editor: Ning Wang

Copyright © 2020 Wei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modern spacecraft are often equipped with large-scale, complex, and lightweight solar arrays whose deployment involves a highly dynamic movement. This paper proposed a novel adaptive proportional-derivative typed fuzzy logic control scheme for the attitude stabilization of a flexible spacecraft during the deployment of a composite laminated solar array. First, a constrained rigid-flexible coupling spacecraft model consisting of a rigid main body and a flexible solar array was proposed. The solar array, which is composed of composite laminated shells, was described by the absolute nodal coordinate formulation. Then, the detailed derivation of the adaptive fuzzy PD controller for attitude stabilization of the spacecraft was discussed. In addition, the spacecraft dynamic model which integrated the adaptive fuzzy PD controller was derived as a set of differential-algebraic equations. Several simulations were developed to investigate the solar array deployment dynamics and to verify the effectiveness of the proposed adaptive fuzzy PD controller. The results suggested that the proposed dynamic model is able to exactly describe the deployment dynamics of the composite laminated solar array. The solar array deployment causes obvious translational and rotational motions of the spacecraft. The proposed adaptive fuzzy PD control scheme has better performance in terms of the control precision and time response in stabilizing spacecraft during the deployment of the composite laminated solar array, comparing with that of the conventional PD controller.

1. Introduction

Modern spacecraft often employ large, complex, and lightweight solar arrays to achieve multiple functionalities and to provide sufficient power supply during flight [1]. The solar array deployment is a highly dynamic movement that may affect the spacecraft's motion [2–4]. In particular, when the locking operation is performed, the induced impulsive forces and moments may cause strong vibrations in large-scale and flexible solar panels. Consequently, it will severely affect spacecraft's motions, even leading to a disaster for a space mission. Therefore, an effective control scheme should be carried out to stabilize the spacecraft's motions during the deployment of solar arrays.

The first challenge is how to exactly describe the deployment dynamics of the solar array and evaluate its influence on the spacecraft main body. The spacecraft system is a typical

constrained rigid-flexible coupling multibody system. In addition, these solar arrays are commonly composed of laminated shells involving fiber-reinforced composite materials, due to its high reliability, superior mechanical properties, high stiffness-to-weight ratio, and low fabrication cost [5–7]. The deployment of the composite laminated solar array exhibits a strong nonlinearity coupling between the large-rotation and large-deformation motions. Thus, an accurate dynamic model, which can well capture nonlinear characteristics (mainly including the material and geometric nonlinearity) of the solar array, plays a crucial role for the control scheme. Over the past few decades, several scholars have investigated the dynamics of the flexible multibody system involving the composite laminated plate/shell structures. Neto et al. [8] described the elastic deformations of a composite laminated plate undergoing large rigid body rotations by using the floating frame of the reference formulation (FFRF). Neto et al. [9] and Ambrósio et al. [10]

adopted the FFRF to study the deployment of a synthetic aperture radar antenna comprising composite laminated plates. The FFRF is a widely used method describing flexible multibody systems in the field of aerospace. However, mode reduction methods cannot be applied when the flexible multibody system experiences large deformations [11]. Additionally, the incremental finite element formulation and the large-rotation vector formulation are also used to describe the flexible multibody systems. These two formulations are capable of describing the large deformations of structures. However, the incremental finite element formulation cannot obtain the exact models of rigid motions when nonisoparametric elements are involved [12], and the large-rotation vector formulation will lead to singularity and unrealistic shear forces problems due to the redundancy [12]. The absolute nodal coordinate formulation (ANCF), originally developed by Shabana [13], utilizes global position vector gradients to model the rotation and deformation fields of the element, which can well describe the flexible body with large deformation in multibody applications. Also, it can avoid the coordinate redundancy problem [12] and singularities emerging from the parameterization of rotations [14] and can also induce a constant mass matrix. At present, several scholars started to show interest in studying the deployment dynamics of flexible multibody systems using ANCF. Li et al. [15] and Li et al. [16] investigated the deployment dynamics of a flexible solar array. The solar array was formulated with the planar deformable ANCF beam element. The planar deformable beam element is not enough when the solar array is a plate-formed structure in which the transverse shearing and the in-plane shearing exert nonnegligible influences on the dynamic response of the spacecraft. Liu et al. [17] adopted the ANCF to study the deployment dynamics of a flexible satellite antenna system comprising composite laminated plates. However, some convergence problems will emerge when the plate element is used to formulate very thin and stiff plates because of the plane stress assumption. Therefore, the deployment dynamics of composite laminated solar arrays still require further improved.

In the past few decades, spacecraft attitude control has gained huge attention. The conventional Proportional-Integration-Derivative (PID) or Proportional-Derivative (PD) control is the most popular industrial control method for simple structures due to its simplicity, reliability, and low-cost implementation [18–23]. The conventional PID/PD control has been widely applied to spacecraft attitude control systems and showed its feasibility and reliability in practical applications/implications [24, 25]. When designing a PD controller, the determination of design parameters is crucial. However, these design parameters cannot be accurately determined in most systems. In addition, the conventional PID/PD controller can accommodate disturbances to a certain extent, where the performance tends to be degraded under circumstances of significant disturbing sources [26]. Therefore, it is difficult for the conventional PID/PD controllers to achieve satisfactory performances with increasing functional requirements of spacecraft [20]. In recent years, the intelligent control scheme, which has evolved from the conventional control laws, has drawn more and more attention and performed better under the uncertainties

condition due to its high degree autonomy [27–33]. The fuzzy logic approach is one of the most widely used methods to observe the uncertainties in complex nonlinear systems [34–40]. Moreover, researchers have revealed that combining PID/PD with fuzzy logic techniques would result in a much better control scheme. Wang and Kwok [41] designed an intelligent control system by using the fuzzy set theory and the PID principle. Boubertakh et al. [42] proposed auto-tuning fuzzy PD and PI controllers using the reinforcement learning algorithm for single-input single-output and two-input two-output systems. Duan et al. [43] revealed a property of inherent saturation in the fuzzy PID controller. Further, Kumar and Kumar [44], Wang et al. [45], and Goma Haroun et al. [46, 47] have introduced new methods to improve the performance of the fuzzy PID/PD controller. These above studies confirmed that the adaptive fuzzy PID/PD control scheme could effectively improve the control performance and efficiency. The adaptive fuzzy PID/PD control scheme has also been gradually applied to the spacecraft attitude control. Kosari et al. [48] proposed a fuzzy PID control scheme based on genetic algorithms during a docking manoeuvre of two spacecraft. Calvo et al. [49] compared the adaptive fuzzy logic PID controller with conventional PID controller for the attitude control of a nanosatellite. The results showed that the adaptive fuzzy PID controller was significantly more efficient than the conventional PID. Chen et al. [50] proposed an adaptive fuzzy PD+ controller for the attitude manoeuvre of a rigid spacecraft. Chak et al. [51] discussed a new composite control scheme combining disturbance observer-based control and fuzzy PD control for flexible spacecraft attitude control in the presence of external and internal disturbances caused by flexible appendages. The results showed that the fuzzy PD controller can effectively manoeuvre the spacecraft to the nadir attitude reference trajectory in the presence of multiple disturbances. Najafizadeh et al. [52] designed a novel fuzzy PID controller for geostationary satellite attitude control, which achieved faster convergence rates and higher accuracy. Li et al. [4, 53] designed a fuzzy PD controller to compensate for the attitude change of the spacecraft caused by the deployment of solar arrays. Although the above mentioned studies have investigated the adaptive fuzzy PID/PD control scheme for attitude control of the spacecraft, there are very few researches on eliminating the attitude and position drift caused by the deployment of solar arrays. In addition, modern spacecraft often employ large, complex, and lightweight solar arrays which are commonly composed of laminated plates with fiber-reinforced composite materials. The flexibility of solar arrays, nonlinearity of composite material, and other external disturbances may cause unpredictable rotational and translational motion of the spacecraft during the deployment of solar arrays. Therefore, a robust enough control scheme is required to overcome the model uncertainty and nonlinearity, structural vibrations of the flexible solar arrays, and the other disturbances in the environment. To the authors' best knowledge, studies related to this topic are not sufficient.

Our work aims to develop a constrained rigid-flexible coupling dynamic model of a spacecraft system equipped

with laminated solar arrays, involving fiber-reinforced composite materials, and further to propose an effective control scheme to stabilize the spacecraft main body during the deployment of the solar arrays. The key points of our work can be briefly described as follows. (1) A constrained rigid-flexible coupling spacecraft dynamic model was developed based on ANCF. The dynamic model can accurately describe the deployment dynamics of the composite laminated solar array undergoing large-rotation and large-deformation motions. (2) An adaptive fuzzy PD control scheme, which can be easily integrated into the spacecraft dynamic model, was proposed. (3) The computation strategies used to solve equations of motion of the spacecraft were provided in detail. The remainder of this paper is organized as follows. Section 2 first depicts the structure of the spacecraft system. The rigid-flexible coupling spacecraft dynamic model was then derived in Section 3. After that, Section 4 reveals the adaptive fuzzy PD control scheme. Section 5 gives equations of motion with the constrained rigid-flexible coupling dynamics and their solution strategies. After that, numerical simulations were conducted and analysed in Section 6. Finally, the whole work was summarised in Section 7.

2. Structure of Spacecraft System

This section illustrates the structure of a spacecraft with a deployable composite laminated solar array. As shown in Figure 1, the spacecraft system is simplified to a main body and a deployable solar array. The solar array mainly contains three types of critical devices, namely, the spring-damper mechanism (Figure 2(a)), the closed cable loop (CCL) mechanism (Figure 2(b)), and the latch mechanism (Figure 2(c)). These devices generate torque to realize the deployment of the solar array. The torque model is given in Figure 2(d). Three assumptions are made for the spacecraft system. (1) The main body and yoke are considered to be a completely free rigid unit. (2) The solar array is assembled with two flexible solar panels, connected by revolute joints. (3) The solar panel is assumed to be a perfect composite laminated shell; that is, there is no relative motion between any two layers of the composite laminated shell.

The spring-damper mechanism, which is located in the revolute joint, provides driving torque to deploy the folded solar array with the expression as

$$T_{\text{drive},k} = K_{\text{drive},k}(\theta_{\text{pre},k} - \theta_k) - C_k \frac{d\theta_k}{dt}, \quad (1)$$

where $K_{\text{drive},k}$ is the torsion stiffness coefficient of the k -th spring-damper, C_k is the damping coefficient of the k -th spring-damper, and $\theta_{\text{pre},k}$ and θ_k denote the preload angle and practical deployment angle of the solar panel, respectively.

The CCL mechanism comprises synchronous wheels fixed on the revolute joint and provides two passive control torques to synchronize the deployment angles of the solar panels [54]. The control torques can be simply modelled as [15]

$$\begin{cases} T_{\text{ccl},1} = K_{\text{ccl},1}(2\theta_1 - \theta_2), \\ T_{\text{ccl},2} = K_{\text{ccl},2}(2\theta_1 - \theta_2), \end{cases} \quad (2)$$

where $T_{\text{ccl},1}$ and $T_{\text{ccl},2}$ are the equivalent synchronous torques and $K_{\text{ccl},1}$ and $K_{\text{ccl},2}$ are the equivalent torsional stiffness of the wheels. In addition, θ_1 and θ_2 are the deployment angles of the first and the second solar panel, respectively.

A typical latch mechanism is demonstrated in Figure 2(c). Body A is connected to the body B , and both of them can rotate around joint C . Cam E is fixed on body A , and pin F can move on cam E 's surface during the deployment process. Pin F slides into groove D when the deployment angle reaches the preset lock angle, and, thus, body A and body B are latched at the expected position. The STEP and BISTOP functions are adopted to simulate this locking process. When the deployment angle θ_k increases from φ_1 to φ_2 , the STEP function correspondingly increases from h_1 to h_2 . Once the deployment angle θ_k reaches the collision point ($\varphi_3 \sim \varphi_4$), the BISTOP function begins to produce torques to push pin F toward the expected angle. These lock torques can be expressed as [15, 53]

$$T_{\text{lock},k} = \text{STEP}(\theta_k, \varphi_1, 0, \varphi_2, 1) \times \text{BISTOP}(\theta_k, \dot{\theta}_k, \varphi_3, \varphi_4, K_s, e, c, d), \quad (3)$$

where

$$\text{STEP}(\theta_k, \varphi_1, h_1, \varphi_2, h_2) = \begin{cases} 0 & \text{if } \theta_k < \varphi_1, \\ h_1 + (h_2 - h_1) \left(\frac{\theta_k - \varphi_1}{\varphi_2 - \varphi_1} \right)^2 \left(3 - 2 \times \frac{\theta_k - \varphi_1}{\varphi_2 - \varphi_1} \right) & \text{if } \varphi_1 \leq \theta_k < \varphi_2, \\ 1 & \text{if } \varphi_2 < \theta_k, \end{cases} \quad (4)$$

$$\text{BISTOP}(\theta_k, \dot{\theta}_k, \varphi_3, \varphi_4, K_s, e, c, d) = \begin{cases} \text{Max}(K_s(\varphi_3 - \theta_k)^e - \dot{\theta}_k \text{STEP}(\theta_k, \varphi_3 - d, c, \varphi_3, 0), 0) & \text{if } \theta_k < \varphi_3, \\ \text{Min}(-K_s(\theta_k - \varphi_4)^e - \dot{\theta}_k \text{STEP}(\theta_k, \varphi_4, 0, \varphi_4 + d, c), 0) & \text{if } \theta_k > \varphi_4, \end{cases}$$

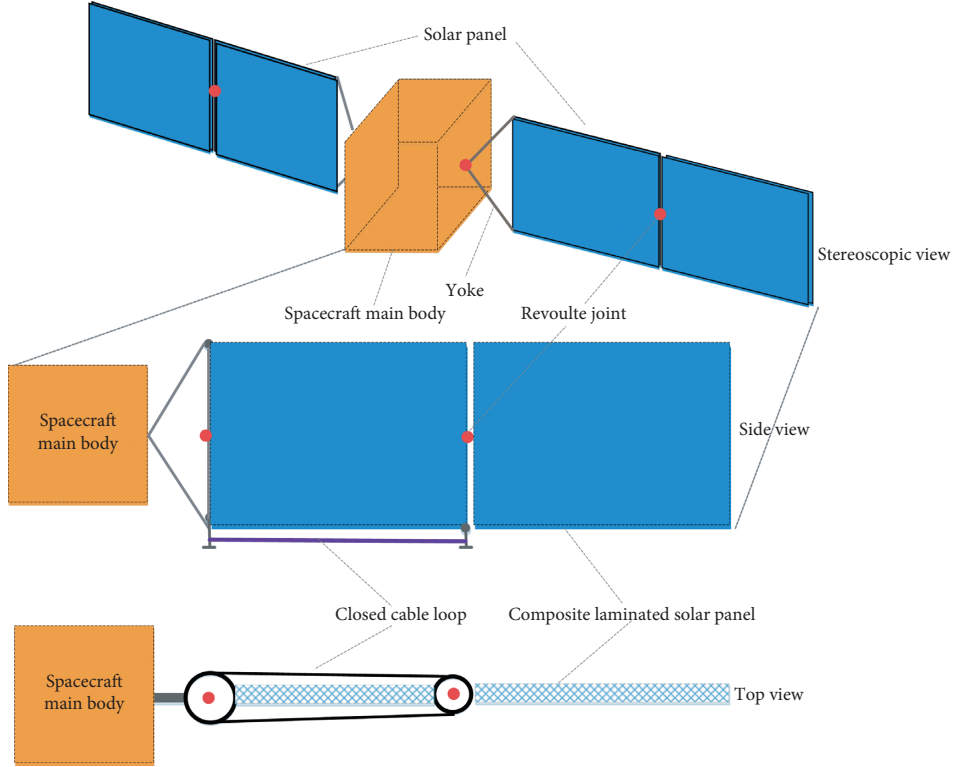


FIGURE 1: Scheme diagram of the spacecraft structure.

where $\dot{\theta}_k$ is the relative rotation velocity of the k -th solar panel. K_s and c are the equivalent stiffness and damping coefficients of the latch mechanism, respectively, d denotes the distance depth, and e is an exponent.

3. Formulation of Rigid-Flexible Coupling Spacecraft System

This subsection is divided into three parts to introduce the formulation of the spacecraft system, corresponding to the motion of the rigid main body, formulation of the composite laminated solar panel, and the motion of constraints. Without loss of generality, the spacecraft main body is regarded as a rigid body of arbitrary shape. As exhibited in Figure 3, an arbitrary point Q is attached to the rigid body. The global position vector of the point Q can be written as

$$r_Q = r + As', \quad (5)$$

where $r \in \mathbb{R}^{3 \times 1}$ is the global position vector of the origin of local coordinate frame x - y - z which is fixed on the rigid body. It should be noted that the local coordinate frame is selected with its origin at the center of mass of the body to simplify the form of the formulation. In equation (5), the local position of the point $s' = [x \ y \ z]^T$ is a constant vector because of the rigid body assumption, and A is the

transformation matrix that defines the orientation of the local coordinate frame with respect to the global reference frame. Thus, the velocity of the point Q can be expressed as

$$\dot{r}_Q = \dot{r} + \dot{A}s' = \dot{r} + \tilde{\omega}As', \quad (6)$$

and the acceleration can be written as

$$\ddot{r}_Q = \ddot{r} + \ddot{A}s' = \ddot{r} + \tilde{\dot{\omega}}As' + \tilde{\omega}\tilde{\omega}As', \quad (7)$$

where ω is the angular velocity of the local coordinate frame with respect to the global coordinate frame. Without the singularity problem, the quaternion p is applied to describe the attitude of the spacecraft main body with the expression as

$$p = \left[\cos \frac{\theta}{2}, u \sin \frac{\theta}{2} \right]^T = \left[\cos \frac{\theta}{2}, u_1 \sin \frac{\theta}{2}, u_2 \sin \frac{\theta}{2}, u_3 \sin \frac{\theta}{2} \right]^T, \quad (8)$$

where $u = [u_1 \ u_2 \ u_3]$ represents the Euler axis and θ is the angle of the rotation. And then, the generalized coordinate vector of the body can be defined with the quaternion as

$$q_b = \begin{bmatrix} r \\ p \end{bmatrix}. \quad (9)$$

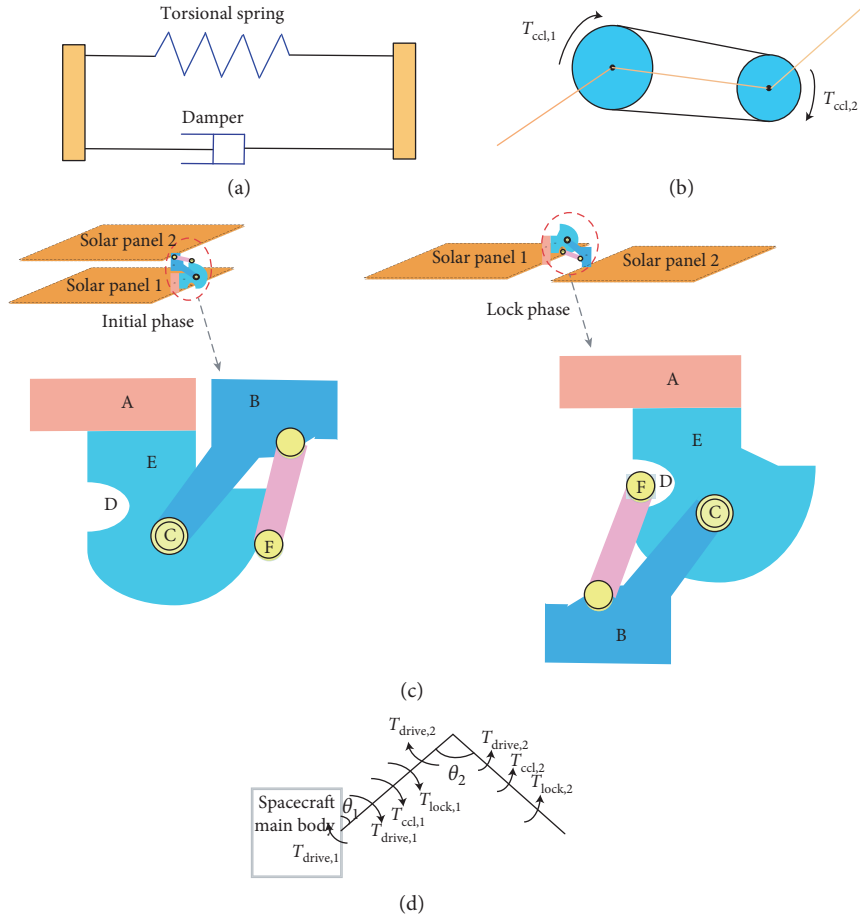


FIGURE 2: Representations of the solar array's mechanisms and torque models. (a) Spring-damper mechanism. (b) CCL mechanism. (c) Latch mechanism. (d) Torque model of the solar array.

The virtual work done by the inertial force can be expressed as [55]

$$\delta W_{\text{ine}}^b = -\delta r^T m \ddot{r} - \delta p^T (4G^T J G \ddot{p} - 8\dot{G}^T J \dot{G} \dot{p}) = -\delta q_b^T \begin{bmatrix} m \\ 4G^T J G \end{bmatrix} \ddot{q}_b + \delta q_b^T \begin{bmatrix} 0 \\ 8\dot{G}^T J \dot{G} \end{bmatrix} \dot{q}_b, \quad (10)$$

where m is the mass of the body, J is the constant inertia tensor which is defined as

$$J = \int_m \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xy & -yz & x^2 + y^2 \end{bmatrix} dm, \quad (11)$$

and G is the function of the quaternion p with the following expression:

$$G = \left[-u \sin \frac{\theta}{2}, -\tilde{u} \sin \frac{\theta}{2} + \cos \frac{\theta}{2} I \right]. \quad (12)$$

The generalized mass matrix of the body can be defined as

$$M_b = \begin{bmatrix} m \\ 4G^T J G \end{bmatrix}. \quad (13)$$

If an external force F_b acts on the body at point Q , the virtual work of the body done by the external force can be obtained by

$$\delta W_{\text{ext}}^b = \delta r^T F_b + \delta p^T (2G^T n) = \delta q_b^T \begin{bmatrix} F_b \\ 2G^T n \end{bmatrix}, \quad (14)$$

where n is the torque generated by the external forces acting at a distance from the origin of the local coordinate frame. The detailed derivation of the virtual work done by the inertial force and the external force can be found in Appendix A. Using equations (10) and (14), we can define the generalized external force as

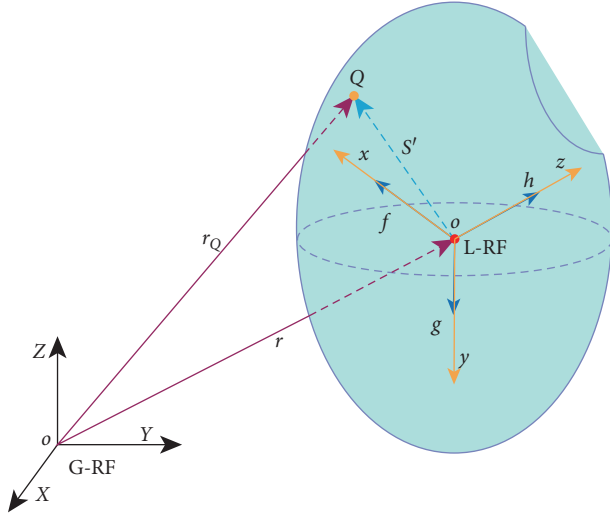


FIGURE 3: Kinematic description of the rigid body.

$$Q_{\text{ext}}^b = \begin{bmatrix} F_b \\ 2G^T n + 8\dot{G}^T J \dot{G} p \end{bmatrix}. \quad (15)$$

The deformable ANCF laminated shell element, originally proposed by Yamashita et al. [56, 57], is employed to formulate the composite laminated solar panel. As shown in Figure 4, the global position vector of an arbitrary point P whose local coordinate is $(x \ y \ z)$ in shell element can be expressed as

$$r_P = r_m + z r_n = S_m q_m + z S_n q_n = \begin{bmatrix} S_m & z S_n \end{bmatrix} \begin{bmatrix} q_m \\ q_n \end{bmatrix}, \quad (16)$$

where r_m is the global position vector of the vertical projection point of P on the middle surface of the element and r_n is the transverse gradient vector that describes the orientation and deformation of the infinitesimal volume in the element. The global position vector of four corner nodes in the middle surface of the element and their transverse displacement gradient can be, respectively, written as

$$\begin{cases} q_m = \begin{bmatrix} r_1^T & r_2^T & r_3^T & r_4^T \end{bmatrix}^T, \\ q_n = \begin{bmatrix} (\frac{\partial r_1}{\partial z})^T & (\frac{\partial r_2}{\partial z})^T & (\frac{\partial r_3}{\partial z})^T & (\frac{\partial r_4}{\partial z})^T \end{bmatrix}^T, \end{cases} \quad (17)$$

where $r_i \in \mathbb{R}^{3 \times 1}$ ($i = 1, 2, 3, 4$) is the global position vector of each node. In (16), S_m is the element shape function defined as [43]

$$S_m = [S_1 I \ S_2 I \ S_3 I \ S_4 I], \quad (18)$$

where $I \in \mathbb{R}^{3 \times 3}$ is the identity matrix and S_i ($i = 1, 2, 3, 4$) are components of the shape function obtained by

$$\begin{cases} S_1 = \frac{1}{4} (1 - \xi)(1 - \eta), & S_2 = \frac{1}{4} (1 + \xi)(1 - \eta), \\ S_3 = \frac{1}{4} (1 + \xi)(1 + \eta), & S_4 = \frac{1}{4} (1 - \xi)(1 + \eta), \end{cases} \quad (19)$$

where $\xi = 2x/l$ and $\eta = 2y/w$. l and w are lengths along the x and y axes, respectively. Then, equation (16) can be simplified as

$$r_P = S q_e \quad (20)$$

where

$$\begin{cases} \mathbf{S} = [S_m \ z S_n], \\ \mathbf{q}_e = [q_m \ q_n]^T. \end{cases} \quad (21)$$

According to the kinematic description of the element, the Green-Lagrange strain tensor is defined as

$$\varepsilon = \frac{1}{2} \left[\left(\frac{\partial r_P}{\partial x} \left(\frac{\partial X}{\partial x} \right)^{-1} \right)^T \left(\frac{\partial r_P}{\partial x} \left(\frac{\partial X}{\partial x} \right)^{-1} \right) - I \right], \quad (22)$$

where x and X denote the local and the global position vector of the material point at the reference configuration, respectively. The element locking may occur in the deformable ANCF shell element due to the use of low-order polynomials in the shape function. The element lock leads to slower convergence in numerical iterations and erroneous stiffer bending behaviour. Thus, the assumed natural strain in [58, 59] and the enhanced assumed strain in [60, 61] are adopted to systematically alleviate the element locking. The modified strain vector can be defined as

$$\hat{\varepsilon} = \varepsilon + \varepsilon^{\text{EAS}}, \quad (23)$$

where ε^{EAS} denotes the enhanced assumed strain vector which is referred in literature [56]. All layers with different fiber angles are assumed to be bonded together to produce the desired material properties. Therefore, the virtual work done by elastic forces of the laminated shell element with n layers can be expressed as [57]

$$\delta W_{\text{ela}}^e = -\delta q_e^T \sum_{i=1}^n \int_{V^i} \left(\frac{\partial \varepsilon^i}{\partial q_e} \right)^T \frac{\partial W^i(\hat{\varepsilon}^i)}{\partial \varepsilon^i} dV^i, \quad (24)$$

where V^i is the element volume of the i -th layer at the reference configuration and W^i is an elastic energy density function that is described in Appendix B. Correspondingly, the generalized elastic forces of the element can then be written as

$$Q_{\text{ela}} = \sum_{i=1}^n \int_{V^i} \left(\frac{\partial \varepsilon^i}{\partial q_e} \right)^T \frac{\partial W^i(\hat{\varepsilon}^i)}{\partial \varepsilon^i} dV^i. \quad (25)$$

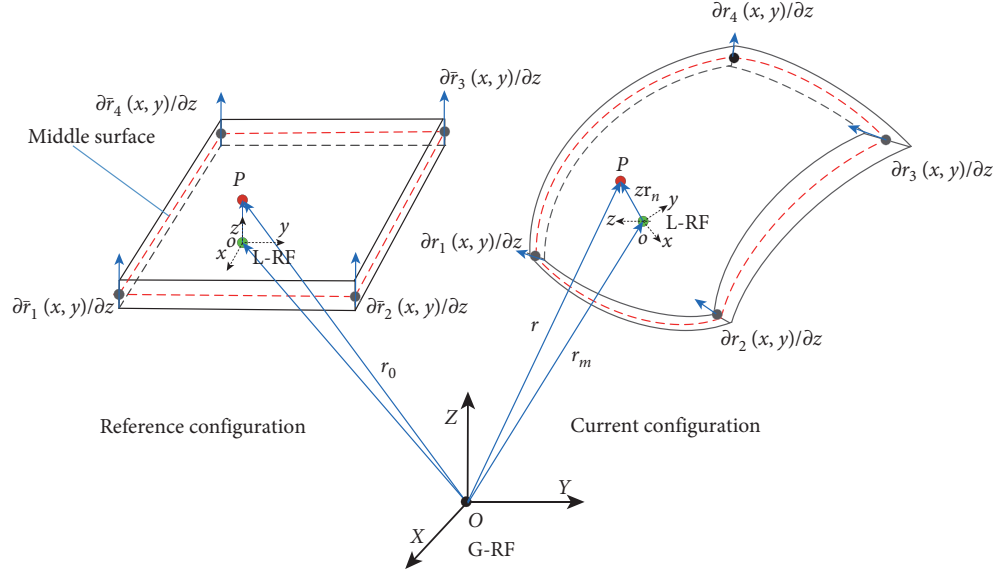


FIGURE 4: Kinematic description of the deformable ANCF shell element.

The shape function of the element is not time-dependent, and, then, the element's kinetic energy can be defined as

$$T = \frac{1}{2} \sum_{k=1}^n \int_{V^i} \rho^i (\dot{r}_P)^T \dot{r}_P dV^i = \frac{1}{2} \dot{q}_e^T \sum_{k=1}^n \int_{V^i} \rho^i S^T S dV^i \dot{q}_e, \quad (26)$$

where ρ^i is the material density of the i -th layer at the reference configuration. Then, the constant mass matrix can be obtained by

$$M_e = \sum_{k=1}^n \int_{V^i} \rho^i S^T S dV^i. \quad (27)$$

Hence, the virtual work done by the inertia forces on the element can be expressed as

$$\delta W_{ine}^e = -\delta q_e^T (M_e \ddot{q}_e). \quad (28)$$

Denoting by F_e the external force acts on the element, we can write the following virtual work:

$$\delta W_{ext}^e = \delta r_P^T F_e = \delta q_e^T (S^T F_e). \quad (29)$$

The generalized external force vector of the element can be obtained by

$$Q_{ext}^e = S^T F_e. \quad (30)$$

It should be noted that the driving torques of the spring-dampers in equation (1), the synchronization torques of the CCL mechanisms in equation (2), and the lock torques of the latch mechanisms in equation (3) are regarded as the external forces of the shell. These torques are then integrated into the generalized external force vector. With the assumption that the structural damping exists in the shell, the virtual work done by the damping forces can be defined as

$$\delta W_{dam}^e = -\delta q_e^T D \dot{q}_e, \quad (31)$$

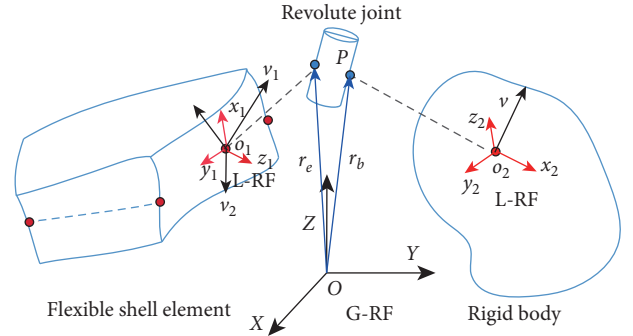


FIGURE 5: Kinematic description of the revolute joint.

where $D = \zeta M_e$ is the damping matrix and ζ is the damping coefficient.

The rigid main body and flexible solar panels are linked by revolute joints, which can be represented by a constraint equation. As depicted in Figure 5, a shell element is connected to a rigid body at point P by a revolute joint, and the constraint equation can be defined as [58, 59]

$$\Phi(q_e, q_b, t) = \begin{bmatrix} r_e - r_b \\ v_1^T v \\ v_2^T v \end{bmatrix} = 0, \quad (32)$$

where q_e and q_b are the generalized coordinate vectors of point P defined on element and rigid body, respectively, r_e and r_b are the global position vectors of point P defined on the element and rigid body, respectively, and v_i ($i = 1, 2$) and v are vectors defined along the joint axis on shell element and rigid body, respectively. We can then obtain the virtual work done by constraint forces [62]:

$$\delta W_{\text{con}} = \lambda^T \delta \Phi = \lambda^T \frac{\partial \Phi}{\partial q_e} \delta q_e + \lambda^T \frac{\partial \Phi}{\partial q_b} \delta q_b = \delta q_e^T (\Phi_{q_e}^T \lambda) + \delta q_b^T (\Phi_{q_b}^T \lambda), \quad (33)$$

where λ is the Lagrange multipliers vector corresponding to the vector Φ . $\Phi_{q_e} = \partial \Phi / \partial q_e$ and $\Phi_{q_b} = \partial \Phi / \partial q_b$ are Jacobian matrices with expression as follows:

$$\begin{bmatrix} \frac{\partial \Phi}{\partial q_e} & \frac{\partial \Phi}{\partial q_b} \end{bmatrix} = \begin{bmatrix} \frac{\partial r_e}{\partial q_e} & \frac{\partial r_b}{\partial q_b} \\ v_1^T \frac{\partial v_1}{\partial q_e} & v_1^T \frac{\partial v}{\partial q_b} \\ v_2^T \frac{\partial v_2}{\partial q_e} & v_2^T \frac{\partial v}{\partial q_b} \end{bmatrix}. \quad (34)$$

4. Adaptive Fuzzy PD Control Scheme

As discussed in the previous section, the deployment of solar arrays may cause attitude and position perturbations within spacecraft. We first observe the conventional PD control scheme in the elimination of position deviations and attitude stabilization during the deployment of solar arrays.

It is well known that reaction wheel actuators are commonly employed to provide control forces and torques, which can accommodate position deviations and finite rotations caused by the deployment of solar arrays. Assuming that $r_a(t) = [r_{a,x} \ r_{a,y} \ r_{a,z}]$ and $r_d(t) = [r_{d,x} \ r_{d,y} \ r_{d,z}]$ are the actual and the desired global position coordinate vectors of the main body, respectively, and $\dot{r}_a = [\dot{r}_{a,x} \ \dot{r}_{a,y} \ \dot{r}_{a,z}]$ and $\dot{r}_d = [\dot{r}_{d,x} \ \dot{r}_{d,y} \ \dot{r}_{d,z}]$ are the actual and desired velocity vectors of the main body, respectively, a PD controller for eliminating position deviations is presented as

$$\tau^f(t) = P^f e_p(t) + D^f \dot{e}_p(t), \quad (35)$$

where

$$\begin{aligned} P^f &= \begin{bmatrix} K_{P,x}^f & 0 & 0 \\ 0 & K_{P,y}^f & 0 \\ 0 & 0 & K_{P,z}^f \end{bmatrix}, \\ D^f &= \begin{bmatrix} K_{D,x}^f & 0 & 0 \\ 0 & K_{D,y}^f & 0 \\ 0 & 0 & K_{D,z}^f \end{bmatrix}, \\ e_p(t) &= \begin{bmatrix} r_{d,x} - r_{a,x} \\ r_{d,y} - r_{a,y} \\ r_{d,z} - r_{a,z} \end{bmatrix}, \\ \dot{e}_p(t) &= \begin{bmatrix} \dot{r}_{d,x} - \dot{r}_{a,x} \\ \dot{r}_{d,y} - \dot{r}_{a,y} \\ \dot{r}_{d,z} - \dot{r}_{a,z} \end{bmatrix}, \end{aligned} \quad (36)$$

in which $K_{P,i}^f (i = x, y, z)$ and $K_{D,i}^f (i = x, y, z)$ are the proportional and differential gains of the PD controller, respectively. Then, the virtual work done by these control forces can be expressed as

$$\delta W_f = \delta r^T \tau^f. \quad (37)$$

According to Euler's theorem of finite rotations of the rigid body, an arbitrary finite rotation of a rigid body over a fixed point can always be represented by the Euler axis and the angle of rotation [63]. In view of the attitude stabilization, we exploit three quaternions, including error quaternion p_e , actual quaternion p_a , and desired quaternion p_d , to describe the attitude deviation of the main body. The mathematical relationships of these quaternions are

$$p_e = (p_a)^{-1} p_d. \quad (38)$$

Now, three equivalent error angles $[\theta_x \ \theta_y \ \theta_z]$ are defined using the error quaternion:

$$\begin{cases} \theta_x = p_e(2)p_e(1), \\ \theta_y = p_e(3)p_e(1), \\ \theta_z = p_e(4)p_e(1), \end{cases} \quad (39)$$

where $p_e(i)$, ($i = 1, 2, 3, 4$) are four components of the error quaternion p_e . Denoting the actual and the desired angular velocity vectors by $\omega_a = [w_{a,x} \ w_{a,y} \ w_{a,z}]$ and $\omega_d = [w_{d,x} \ w_{d,y} \ w_{d,z}]$, respectively, we can present the PD controller for attitude stabilization as

$$\tau^t(t) = P^t e_a(t) + D^t \dot{e}_a(t), \quad (40)$$

where

$$\begin{aligned} P^t &= \begin{bmatrix} K_{P,x}^t & 0 & 0 \\ 0 & K_{P,y}^t & 0 \\ 0 & 0 & K_{P,z}^t \end{bmatrix}, \\ D^t &= \begin{bmatrix} K_{D,x}^t & 0 & 0 \\ 0 & K_{D,y}^t & 0 \\ 0 & 0 & K_{D,z}^t \end{bmatrix}, \end{aligned} \quad (41)$$

$$e_a(t) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix},$$

$$\dot{e}_a(t) = \begin{bmatrix} w_{d,x} - w_{a,x} \\ w_{d,y} - w_{a,y} \\ w_{d,z} - w_{a,z} \end{bmatrix},$$

in which $K_{P,i}^t (i = x, y, z)$ and $K_{D,i}^t (i = x, y, z)$ are the proportional and differential gains of the PD controller, respectively. Thus, the virtual work done by these control torques can be obtained by

$$\delta W_t = \delta p^T \tau^t. \quad (42)$$

It is a challenging task to determine proper PD parameters when considering the flexibility of solar arrays, nonlinearity of the composite material, and other uncertainties. As a result, an adaptive fuzzy PD control scheme was proposed. The block diagram of the proposed control scheme is depicted in Figure 6. The adaptive fuzzy PD controller primarily consists of two parts: one is a fuzzy logic system and the other is a PD controller. In the

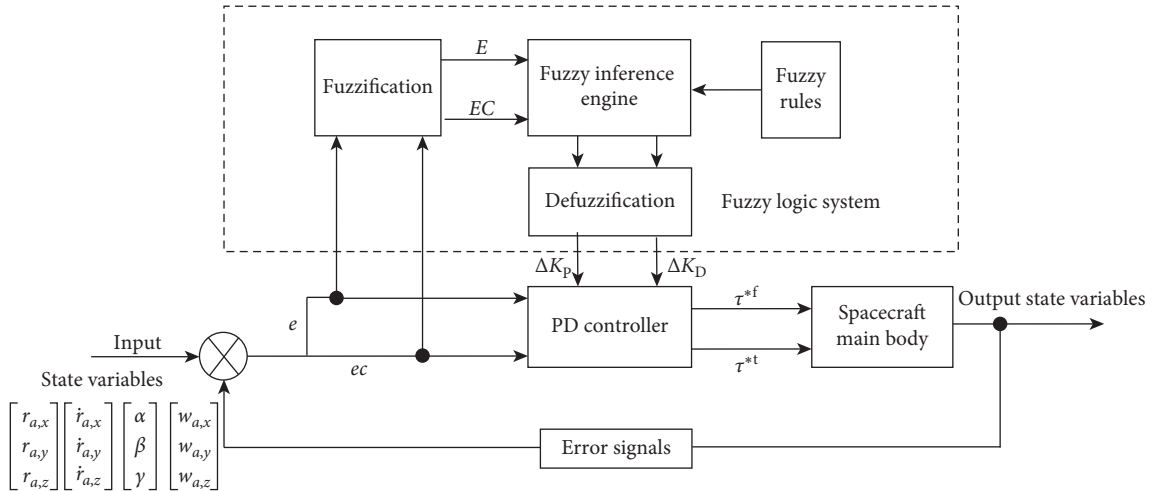


FIGURE 6: Schematic of the adaptive fuzzy PD control scheme.

control loop, the control gains are modified online using the fuzzy rules and sent to the PD controller. Firstly, error signals are computed according to the state variables of the spacecraft and the desired variables. The symbol e in the diagram includes actual position errors $r_d - r_a$ and attitude angle errors $[\theta_x \ \theta_y \ \theta_z]$ while the symbol ec represents velocity errors $\dot{r}_d - \dot{r}_a$ and angular velocity errors $\omega_d - \omega_a$. These error signals are fuzzified by using membership functions and sent to the fuzzy inference engine. After that, the fuzzy inference engine performs the fuzzy reasoning on the basis of fuzzy rules to obtain the fuzzy proportional and differential gains. Finally, the PD controller updates its feedback gains in real time according to these fuzzy gains. The control forces and torques generated from the PD controller act on the spacecraft main body. Control actuators remain active until those error signals reach zero. The fuzzy logic system mainly includes four components, namely, fuzzification, fuzzy rules, fuzzy inference engine, and defuzzification. Fuzzification maps the input state errors (e and ec) into two fuzzy variables (E and EC) that are defined with three fuzzy sets where the associated linguistic terms are negative (N), zero (Z), and positive (P) by using membership functions. The fuzzy IF-THEN rules are adopted to relate these fuzzy sets to output fuzzy gains with the Mamdani-type fuzzy inference [53]. Defuzzification converts the output linguistic variables into precise numerical values. The center-of-gravity method is used to defuzzify the subset in the fuzzy logic system. Now, we give the expression of the control forces as follows:

$$\tau^{*f}(t) = P^{*f} e_p(t) + D^{*f} \dot{e}_p(t), \quad (43)$$

where

$$P^{*f} = \begin{bmatrix} K_{P,x}^f + \Delta K_{P,x}^f & 0 & 0 \\ 0 & K_{P,y}^f + \Delta K_{P,y}^f & 0 \\ 0 & 0 & K_{P,z}^f + \Delta K_{P,z}^f \end{bmatrix}, \quad (44)$$

$$D^{*f} = \begin{bmatrix} K_{D,x}^f + \Delta K_{D,x}^f & 0 & 0 \\ 0 & K_{D,y}^f + \Delta K_{D,y}^f & 0 \\ 0 & 0 & K_{D,z}^f + \Delta K_{D,z}^f \end{bmatrix},$$

in which $\Delta K_{P,i}^f$ and $\Delta K_{D,i}^f$ ($i = x, y, z$) are the fuzzy increments of the proportional and differential gains, respectively. The expression of the control forces can be written as

$$\tau^{*t}(t) = P^{*t} e_p(t) + D^{*t} \dot{e}_p(t), \quad (45)$$

where

$$P^{*t} = \begin{bmatrix} K_{P,x}^t + \Delta K_{P,x}^t & 0 & 0 \\ 0 & K_{P,y}^t + \Delta K_{P,y}^t & 0 \\ 0 & 0 & K_{P,z}^t + \Delta K_{P,z}^t \end{bmatrix},$$

$$D^{*t} = \begin{bmatrix} K_{D,x}^t + \Delta K_{D,x}^t & 0 & 0 \\ 0 & K_{D,y}^t + \Delta K_{D,y}^t & 0 \\ 0 & 0 & K_{D,z}^t + \Delta K_{D,z}^t \end{bmatrix}, \quad (46)$$

in which $\Delta K_{P,i}^t$ and $\Delta K_{D,i}^t$ ($i = x, y, z$) are the fuzzy increments of the proportional and differential gains, respectively. The total virtual work done by these forces and torques can be expressed as

$$\delta W_{fuz}^b = \delta W_f^* + \delta W_t^* = \delta r^T \tau^{*f} + \delta p^T \tau^{*t} = \delta q_b^T \begin{bmatrix} \tau^{*f} \\ \tau^{*t} \end{bmatrix}, \quad (47)$$

and then the generalized control force can be defined as

$$Q_{fuz}^b = \begin{bmatrix} \tau^{*f} \\ \tau^{*t} \end{bmatrix}. \quad (48)$$

5. Equations of Motion and Solve Strategy

According to the principle of virtual work, the variation equations of the whole rigid-flexible coupling spacecraft system can be written as

$$(\delta W_{ine}^b + \delta W_{ext}^b) + (\delta W_{ine}^e + \delta W_{ela}^e + \delta W_{ext}^e + \delta W_{dam}^e) + \delta W_{con} = 0. \quad (49)$$

Substituting equations (10), (13)–(15), (24), (25), (28)–(31), (33), and (47) into (49) yields

$$\begin{aligned} & [-\delta q_b^T (M_b \ddot{q}_b) + \delta q_b^T Q_{\text{ext}}^b + \delta q_b^T Q_{\text{fuz}}^b] + \delta q_e^T (\Phi_q^T \lambda) + \delta q_b^T (\Phi_{q_b}^T \lambda) \\ & + [-\delta q_e^T (M_e \ddot{q}_e) + \delta q_e^T Q_{\text{ela}}^e + \delta q_e^T Q_{\text{ext}}^e - \delta q_e^T D \dot{q}_e] = 0. \end{aligned} \quad (50)$$

Using equation (32), we can obtain equations of motion of the spacecraft:

$$\begin{cases} M \ddot{q} + \Phi_q^T \lambda + Q(q) = F(q, \dot{q}), \\ \Phi(q, t) = 0, \end{cases} \quad (51)$$

where

$$\begin{aligned} M &= \begin{bmatrix} M_b & 0 \\ 0 & M_e \end{bmatrix}, \\ Q(q) &= \begin{bmatrix} 0 \\ -Q_{\text{ela}}^e \end{bmatrix}, \\ F(q, \dot{q}) &= \begin{bmatrix} Q_{\text{ext}}^b + Q_{\text{fuz}}^b \\ Q_{\text{ext}}^e - D \dot{q}_e \end{bmatrix}, \\ q &= \begin{bmatrix} q_b \\ q_e \end{bmatrix}. \end{aligned} \quad (52)$$

The Newmark method is used to discretize equation (51)

as

$$\begin{cases} M \ddot{q}_{n+1} + \Phi_q^T \lambda_{n+1} + Q(q_{n+1}) = F(q_{n+1}, \dot{q}_{n+1}), \\ \Phi(q_{n+1}, t_{n+1}) = 0, \end{cases} \quad (53)$$

where

$$\begin{cases} q_{n+1} = q_n + h \dot{q}_n + \frac{h^2}{2} ((1 - 2\beta) \ddot{q}_n + 2\beta \ddot{q}_{n+1}), \\ \dot{q}_{n+1} = \dot{q}_n + h((1 - \gamma) \ddot{q}_n + \gamma \ddot{q}_{n+1}), \end{cases} \quad (54)$$

where β and γ are parameters for determining iteration convergence and accuracy, respectively. The Newmark method are generally poor in solving the dynamics problems of the multibody system with high-frequency responses [64]. Additionally, spurious high-frequency responses may exist in the spacecraft system, mainly due to the flexibility of solar arrays. Thus, it is desirable for an algorithm with a controllable numerical dissipation which can well-preserve the low-frequency responses and damp the high-frequency responses. Negrut et al. [65] proposed an HHT-based algorithm for the index 3 differential-algebraic equations of multibody systems, known as HHT-I3. According to ANCF, it has been demonstrated that the HHT-I3 method's high-frequency responses can be filtered out while preserving accuracy [66]. According to the HHT-I3 method, the iterative form can be written as [67]

$$\begin{bmatrix} \frac{\partial N_1}{\partial \ddot{q}_{n+1}} & \frac{\partial N_1}{\partial \lambda_{n+1}} \\ \frac{\partial N_2}{\partial \ddot{q}_{n+1}} & \frac{\partial N_2}{\partial \lambda_{n+1}} \end{bmatrix} \begin{bmatrix} \Delta \ddot{q}_{n+1} \\ \Delta \lambda_{n+1} \end{bmatrix} = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}, \quad (55)$$

where

$$\begin{cases} N_1 = \frac{1}{1 + \eta} M \ddot{q}_{n+1} + \Phi_q^T \lambda_{n+1} + Q(q_{n+1}) - F(q_{n+1}, \dot{q}_{n+1}) - \frac{\eta}{1 + \eta} [\Phi_q^T \lambda_n + Q(q_n) - F(q_n, \dot{q}_n)], \\ N_2 = \frac{1}{\beta h^2} \Phi(q_{n+1}, t_{n+1}), \\ \frac{\partial N_1}{\partial \lambda_{n+1}} = \Phi_q^T, \\ \frac{\partial N_2}{\partial \lambda_{n+1}} = 0, \\ \frac{\partial N_1}{\partial \ddot{q}_{n+1}} = \frac{1}{1 + \eta} M + \frac{\partial Q}{\partial q_{n+1}} \frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} - \left(\frac{\partial F}{\partial q_{n+1}} \frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} + \frac{\partial F}{\partial \dot{q}_{n+1}} \frac{\partial \dot{q}_{n+1}}{\partial \ddot{q}_{n+1}} \right), \\ = \frac{1}{1 + \eta} M + \left(\frac{\partial Q}{\partial q_{n+1}} - \frac{\partial F}{\partial q_{n+1}} \right) \beta h^2 - \frac{\partial F}{\partial \dot{q}_{n+1}} h \gamma, \\ \frac{\partial N_2}{\partial \ddot{q}_{n+1}} = \frac{1}{\beta h^2} \frac{\partial N_2}{\partial q_{n+1}} \frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} = \Phi_q, \end{cases} \quad (56)$$

where $\beta = (1 - \eta)^2/4$ and $\gamma = 0.5 - \eta$, $\eta \in [-(1/3) 0]$, and the parameter η is a numerical damping index. Converting equation (55) into the explicit format at iteration k yields

$$\begin{bmatrix} \frac{1}{1 + \eta} M + \left(\frac{\partial Q}{\partial q_{n+1}} - \frac{\partial F}{\partial q_{n+1}} \right) \beta h^2 - \frac{\partial F}{\partial q_{n+1}} h \gamma & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix}^{(k)} \begin{bmatrix} \Delta \ddot{q}_{n+1} \\ \Delta \lambda_{n+1} \end{bmatrix}^{(k)} = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}^{(k)}. \quad (57)$$

Thus, the iteration can be written as

$$\begin{bmatrix} \ddot{q}_{n+1} \\ \lambda_{n+1} \end{bmatrix}^{(k+1)} = \begin{bmatrix} \ddot{q}_{n+1} \\ \lambda_{n+1} \end{bmatrix}^{(k)} - \begin{bmatrix} \Delta \ddot{q}_{n+1} \\ \Delta \lambda_{n+1} \end{bmatrix}^{(k)}, \quad (58)$$

where the initial step can be set as [67]

$$\begin{cases} \dot{q}_0 = q_0 = 0, \\ \ddot{q}_0 = M^{-1}F. \end{cases} \quad (59)$$

6. Results and Discussion

6.1. Model Parameters. The solar array deployment is divided into three phases, namely, the initial phase, the deployment phase, and the postlock phase (see Figure 7). During the initial phase, the solar panels are folded and set parallel to each other. Once the spacecraft is launched into its free-flying orbit, the solar panels are driven to deploy by the preloaded spring-damper mechanism, and the CCL mechanism ensures the synchronous deployment. When the solar panels are deployed on the same spatial plane, the lock torques provided by the latch mechanism begin to confine relative rotation of the panels. At this time and thereafter, the solar array deployment enters its postlock phase. The global coordinate system, which is the absolute reference for the spacecraft system's motions, is set up and its origin is at the center of mass of spacecraft main body. The geometrical dimensions of the vital components of the system are labeled in the figure.

Several simulations are conducted to confirm the effectiveness of the proposed control scheme. It is assumed that the geometrical dimensions and material properties of the main body as well as physical parameters of the solar array mechanisms keep identical in all simulations. Simulation parameters are listed in Table 1. The numerical simulations for the proposed adaptive fuzzy PD control scheme were performed under cosimulation of Visual Studio 2017 and MATLAB R2017 software. In these numerical calculations, the adaptive fuzzy PD control algorithms are implemented by MATLAB/Simulink, and the numerical algorithms for solving the equations of the proposed rigid-flexible coupling dynamic model are implemented by C++ programs. Each flexible solar panel is divided into 120 elements.

6.2. Model Validation. In this subsection, two simulations are carried out to verify the proposed model. The first simulation presents a cantilevered shell subjected to a vertical force to reveal the mechanical properties of the

deformable ANCF laminated shell element adopted in this paper. As shown in Figure 8, a rectangular cantilevered composite laminated shell with four layers (0/45/-45/0) is subjected to a force in the Z direction. The geometric size of the composite laminated shell is $5 \times 3 \times 0.008$ m. The material properties of these four layers are the same: Young's modulus $E_1 = 6.78 \times 10^8$ Pa, $E_2 = E_3 = 4.08 \times 10^8$ Pa, Shear modulus $G_{12} = G_{13} = 1.20 \times 10^8$ Pa, $G_{23} = 4.50 \times 10^8$ Pa, Poisson's ratio $\nu_{12} = \nu_{13} = \nu_{23} = 0.3$, and material density $\rho = 150$ kg/m³. A vertical force in the Z direction is applied on point B which is located at the far right corner of the shell (viewed from the clamped boundary).

A contrast study is made between the deformable ANCF laminated shell and the laminated shell with finite membrane strains modelled by ABAQUS S4R elements. As shown in Figure 9, the displacement responses at point B are compared along the X (Figure 9(a)), Y (Figure 9(b)), and Z (Figure 9(c)) directions. The displacement curves obtained by the deformable ANCF laminated shell elements match well with those obtained by the ABAQUS S4R laminated shell element, which indicate that the deformable ANCF laminated shell element in this paper clearly describes a flexible body undergoing large deformations.

The spacecraft in this study is a typical rigid-flexible coupling system with a flexible solar array formulated with deformable ANCF composite laminated shell element. An identical rigid-flexible coupling model was achieved by using the cosimulation of ADAMS-ABAQUS. The rigid main body is modelled by ADAMS software, and the flexible solar panels are constructed of largely deformable ABAQUS S4R laminated shell element. Figure 10 shows the comparison results for the angular displacements of the two solar panels. As illustrated in Figures 10(a) and 10(b), the angular displacement curves obtained by the proposed model almost coincide with those obtained by ADAMS-ABAQUS software, suggesting that the proposed model is valid.

6.3. Dynamic Responses of Spacecraft System. This subsection presents the position and attitude responses of the main body during the deployment of the solar array. Figure 11 shows the solar panel deployment process. Figures 11(a) and 11(b) depict the angular displacements of solar panel 1 and solar panel 2, respectively. In the initial phase, the solar panels are folded, as well as setting parallel to each other, and their angular displacements are zero or near zero. After the solar array is released, the two solar panels are deployed through the torsional spring-damper mechanisms. The deployment angles are controlled by the CCL mechanism, with a proportionality coefficient of 0.5. The system is then locked by the latch mechanism at about the 18th second, when the lock torques are produced by the latch mechanism to prevent relative rotation of the two panels. In the postlock phase, there are some slight oscillations in the angular displacements of the panels. These oscillations are mainly caused by impact forces from the latch mechanisms. After that, the system maintains a relatively steady deployment

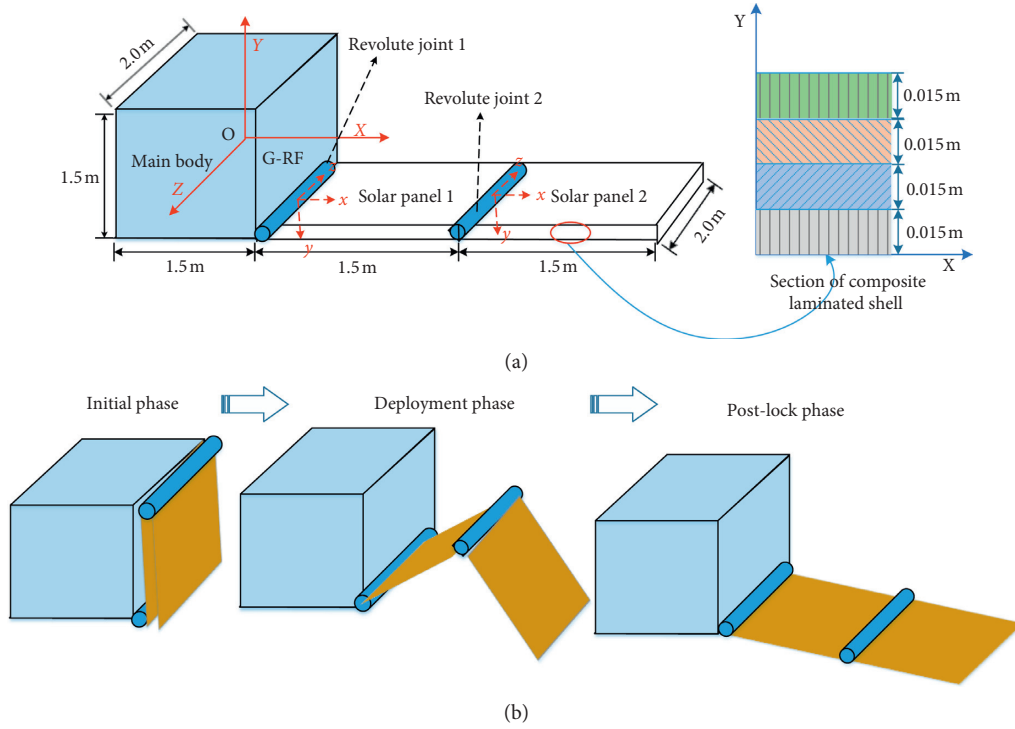


FIGURE 7: Deployment process of the solar array.

TABLE 1: Physical parameters of the spacecraft system.

Parameter item	Parameter value
Size of the main body (m)	$1.5 \times 1.5 \times 2.0$
Material density of the main body (kg/m^3)	2400
Length of panels (m)	1.5
Width of panels (m)	2.0
Thickness of panels (m)	0.06
Material density of panels (kg/m^3)	100
Young's modulus of panels (Pa)	$E_1 = 6.78 \times 10^8, E_2 = E_3 = 4.08 \times 10^8$
Poisson's ratio of panels (-)	$\nu_{12} = \nu_{13} = \nu_{23} = 0.3$
Shear modulus of panels (Pa)	$G_{12} = G_{13} = 1.2 \times 10^8, G_{23} = 4.5 \times 10^8$
Layer number of panels (-)	4
Fiber angle relative to the Y direction ($^\circ$)	0/90/90/0
Material thickness of panels (m)	0.015/0.015/0.015/0.015
Torsional stiffness of spring-damper 1 (N m/rad)	0.25
Torsional stiffness of spring-damper 2 (N m/rad)	0.085
Damping coefficient of spring-damper 1 (N/(m/s))	0.025
Damping coefficient of spring-damper 2 (N/(m/s))	0.015
Equivalent cable stiffness of CCL 1 (N m/rad)	60
Equivalent cable stiffness of CCL 2 (N m/rad)	15
Equivalent stiffness of latch 1 (N m/rad)	2.5×10^4
Equivalent stiffness of latch 2 (N m/rad)	1×10^4
Damping coefficient of latch 1 (N/(m/s))	2.5×10^3
Damping coefficient of latch 2 (N/(m/s))	1×10^3
Exponent of latch 1 and latch 2 (-)	1.5
Distance depth of latch 1 and latch 2 ($^\circ$)	0.01
Time step (s)	0.001
Total time (s)	40

state in the plane with deployment angles of nearly 0.5π for panel 1 and nearly π for panel 2.

The torque responses of three types of mechanisms mentioned in the previous section are critical for the whole

spacecraft system. Figure 12 shows the torque responses of these mechanisms for solar array deployment. Figure 12(a) shows that the driving torques of the two spring-dampers reach the peak during the initial phase, and, then, both

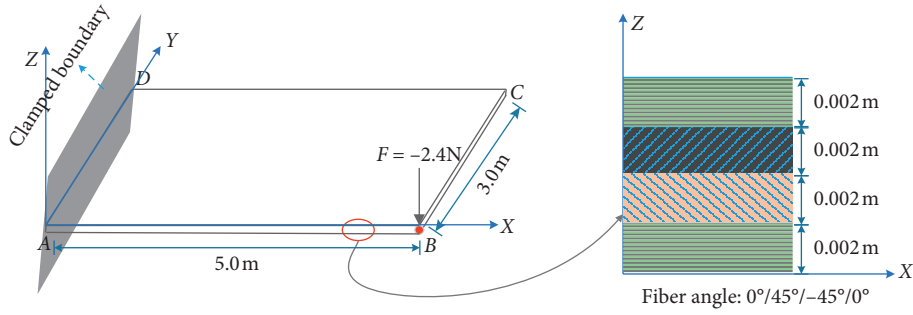


FIGURE 8: A cantilever composite laminated shell subjected to a vertical force.

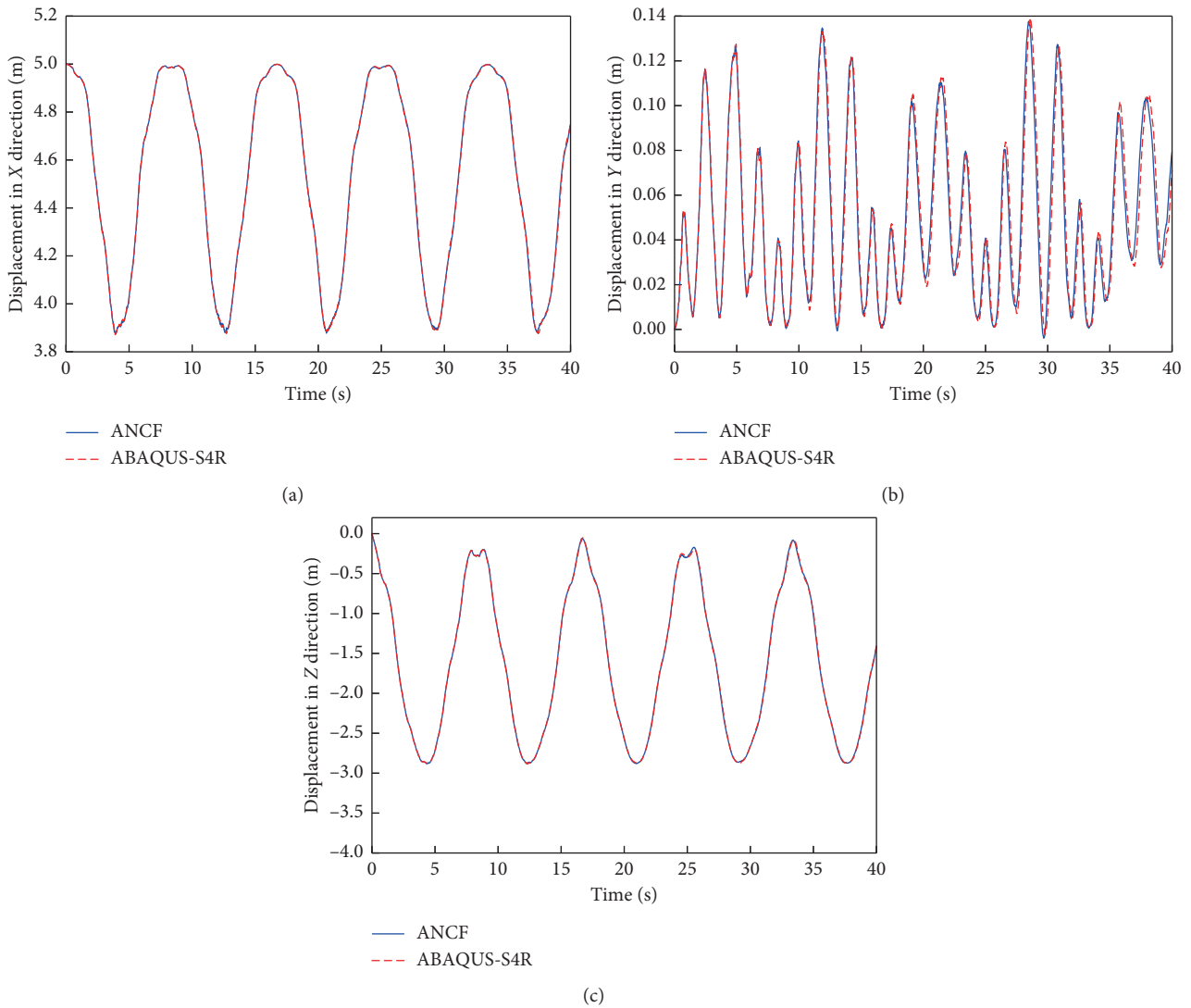


FIGURE 9: Displacement responses of the cantilever composite laminated shell.

torques gradually converge to zero during the deployment phase and reach a small neighbourhood of zero during the postlock phase. The driving torques of the two spring-dampers vary with the phases of the solar array deployment. Moreover, the torques of the two spring-dampers

still demonstrate a trend of obvious oscillations in the neighbourhood of zero in the postlock phase. However, the oscillations gradually decrease over time due to damping from the spring-dampers. Figure 12(b) illustrates the trends of synchronous control torque through

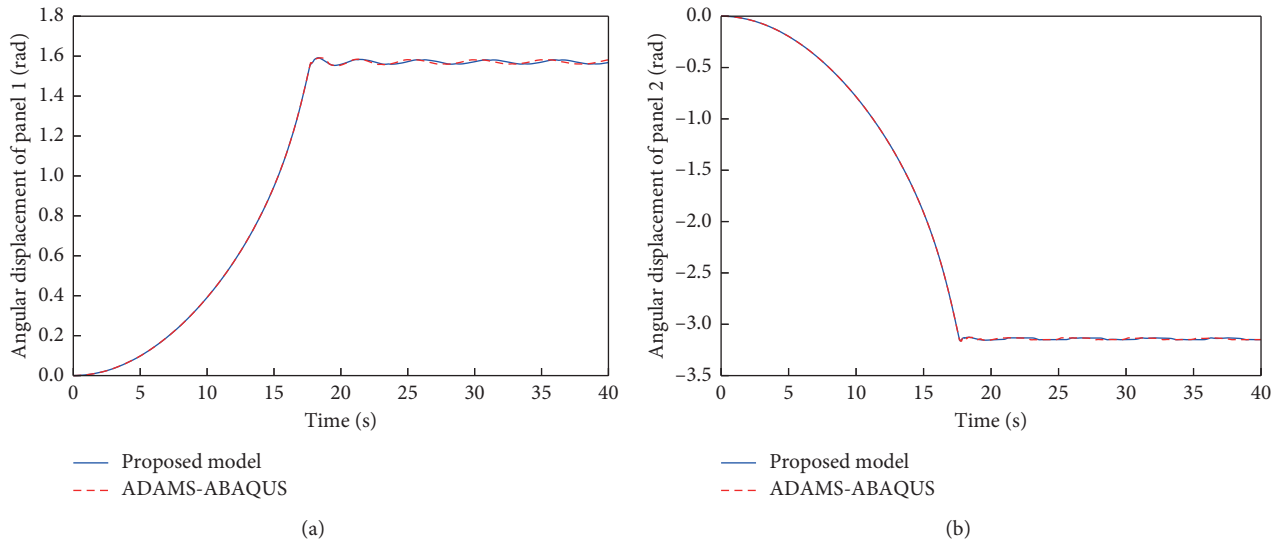


FIGURE 10: Comparison results of angular displacements of two solar panels between the proposed method and cosimulation of ADAMS-ABAQUS software.

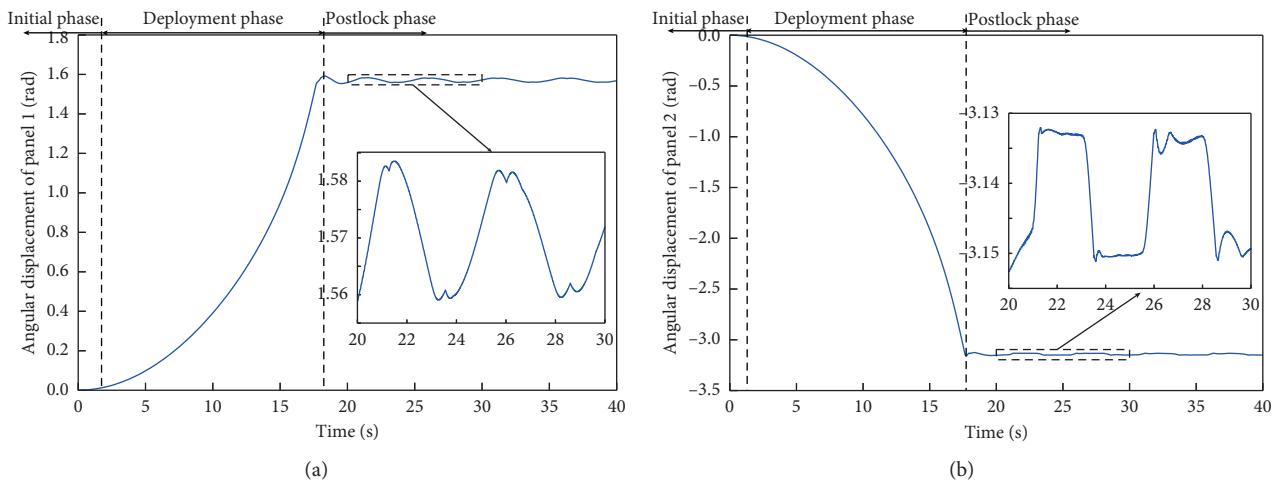


FIGURE 11: Angular displacements of two solar panels.

the CCL mechanism. The control torques of the two CCL mechanisms increase with small fluctuations in the early deployment phase and then reach the larger fluctuations in the later deployment phase. Figure 12(b) indicates that greater control torques produced from the CCL mechanism are required when the first panel becomes more out of sync with the second panel. After the two panels are locked, the control torques fluctuate around the domain of zero, indicating asynchronous oscillations within the two panels. The lock torque curves from the latch mechanism are illustrated in Figure 12(c). These results show that impulsive forces are induced when the locking operation is performed to lock the panels in a proper position. These impulsive forces are generally much larger than those from the other two mechanism types, which can lead to strong vibrations in the solar panels and may also cause the position deviations and attitude changes of the spacecraft main body. Moreover, the impulsive forces of the second latch mechanism are much

greater than those of the first latch mechanism. Therefore, several methods should be applied to avoid excessive impulsive forces from the latch mechanism.

Figure 13 presents the displacement responses of the main body during the deployment of the solar array. As shown in Figures 13(a), 13(b), and 13(c), the main body gradually deviates from its original position during the deployment phase. Moreover, the displacements of the main body in the X and Y directions are much larger than those in the Z direction. On the other hand, it is also shown in Figures 13(a) and 13(b) that the deviations climb to the top and remain steady in the postlock phase. The results demonstrate that the deployment of the solar array mainly causes the position deviations in the X and Y directions in the deployment phase. Therefore, the position changes in the X and Y directions are the main considerations in the controller design, while the change in Z direction is negligible.

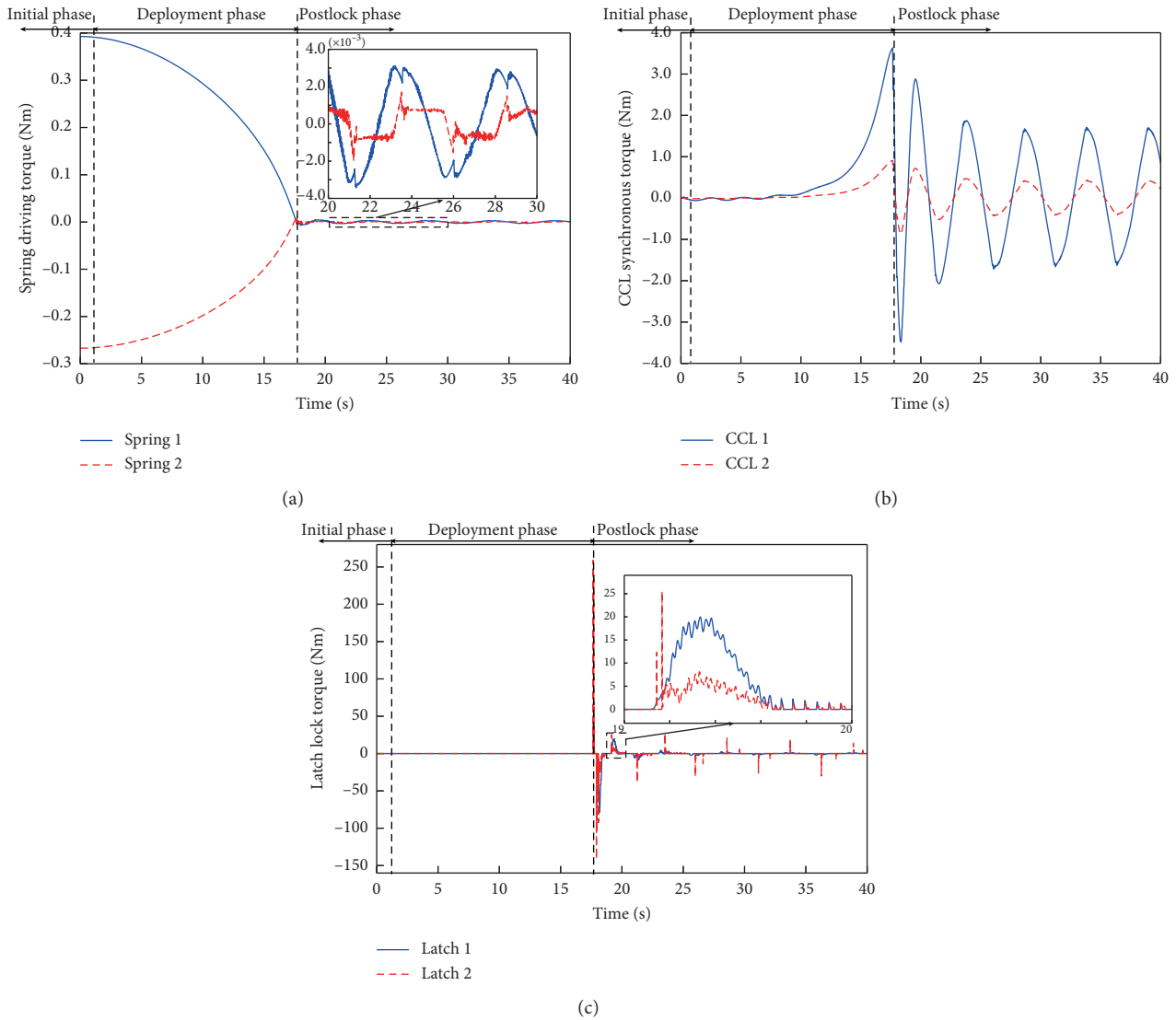


FIGURE 12: Torque responses of the solar array mechanisms.

Figure 14 illustrates the velocity responses of the main body during the deployment of solar arrays. The reference line in the following figures indicates that the value always keeps zero. As shown in Figure 14(a), the velocity signal gradually increases in the first 15 seconds and then decreases dramatically along the X direction throughout the deployment phase. From the zoomed part in Figure 14(a), there is a minor fluctuation in the postlock phase. As is illustrated in Figure 14(b), the velocity of the spacecraft main body increases steadily in the Y direction during the deployment phase and reaches a peak of 0.8×10^{-3} m/s. This decreases significantly after the solar panels are locked by the latch mechanisms. The velocity in the Y direction oscillates at a higher amplitude than it does in the X direction during the postlocking phase. The velocity in the Z direction is five orders of magnitude lower compared with that in the other two directions (see Figure 14(c)). Velocities in the X and Y

directions are therefore more significant in the controller design.

Figure 15 reveals the attitude responses of the main body during the deployment of the solar array system. As demonstrated in Figure 15(a), the rotation vector of the main body is almost constant with values (0.0, 0.0, 1.0) during all phases, which indicates that the main body approximately rotates around the Z axis during the deployment of solar arrays. As illustrated in Figures 15(b) and 15(c), the rotation angle and angular velocity of the main body have similar change tendencies during the deployment phase and reach the highest peak when the latch mechanisms lock the solar panels. After the solar panels are locked, the rotation angle and velocity of the main body present obvious fluctuations. Consequently, the rotation angle and velocity around the Z axis are indispensable variables to be taken into account in the controller design.

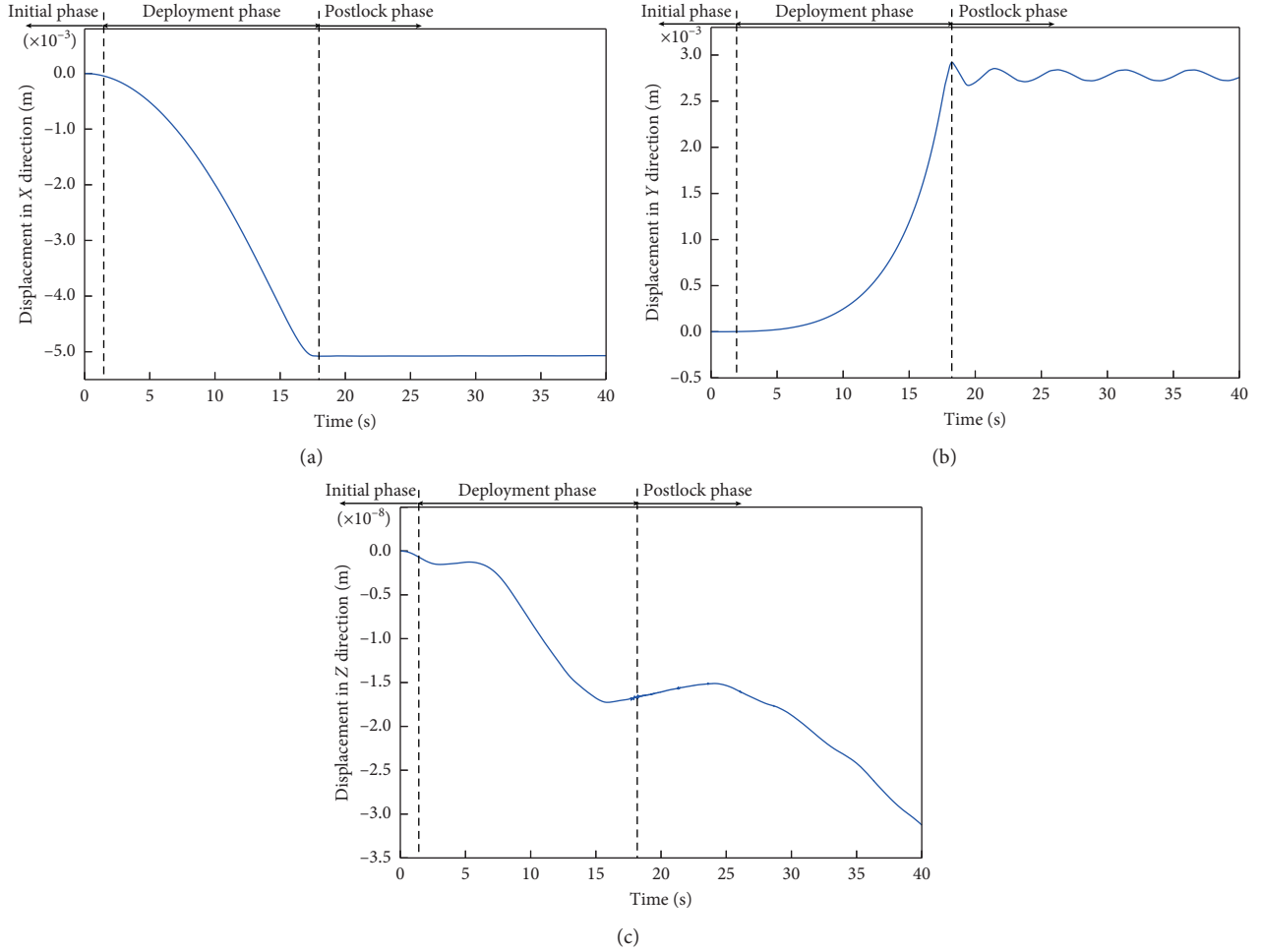


FIGURE 13: Displacement responses of the spacecraft main body in the three directions.

6.4. Conventional PD Control Scheme. The subsection presents a conventional PD control scheme to eliminate the position deviations and attitude changes of the spacecraft main body. As mentioned in the previous sections, the position deviations (X and Y directions) and attitude change (rotation around the Z axis) are major considerations in the control design. The desired displacements $[r_{d,x} \ r_{d,y}]$ and velocities $[\dot{r}_{d,x} \ \dot{r}_{d,y}]$ in equation (43), the desired rotation angle γ , and angular velocity $w_{d,z}$ in equation (45) are set to be zero. Two groups of the simulation cases were performed to investigate the effects of the PD parameters on the control performance. The parameters of these simulation cases are listed in Table 2. Figure 16 depicts the evolution of the position and attitude responses of the spacecraft in these simulation cases.

Figure 16 shows that all the state variables of the spacecraft main body converge nearly to zero, indicating that the spacecraft system is stabilized successfully under the conventional PD control scheme. In Group 1, the differential gain K_D is fixed, while the proportional gain K_P gradually increases in these simulation cases. As illustrated in Figures 16(a) and 16(c), the displacement deviations of the spacecraft in the X and Y directions significantly decrease and expedite the convergence to zero with the increase of the

proportional gain K_P . The attitudes of the spacecraft are also obviously stabilized as shown in Figure 16(e). The proportional term of the PD controller provides an overall control action proportional to the error signals through the gain factor, which means that the steady-state error and rise time of the system reduced by increasing the proportional gain factor. However, an increase in the proportional gain factor also decreases the stability margins and may induce oscillations in the system. It is insufficient to use only the proportional gain factor to improve the system responses. In Group 2, the proportional gain K_P is fixed, while the differential gain K_D increases in order to evaluate the effects of the differential term of the PD controller. As depicted in Figures 16(b) and 16(d), the displacement responses of the spacecraft in the X and Y directions have been improved to a certain extent with the increase of the differential gain K_D . After the solar panels are locked, the rotation angle of the spacecraft has obvious fluctuations. From the zoomed part in Figure 16(f), the amplitude of these fluctuations significantly decreases with the increase of the differential gain K_D . The differential term of the PD controller is capable of predicting the trend of error signals and thus improving the transient responses and system stability through high-frequency compensation. However, the differential term of the

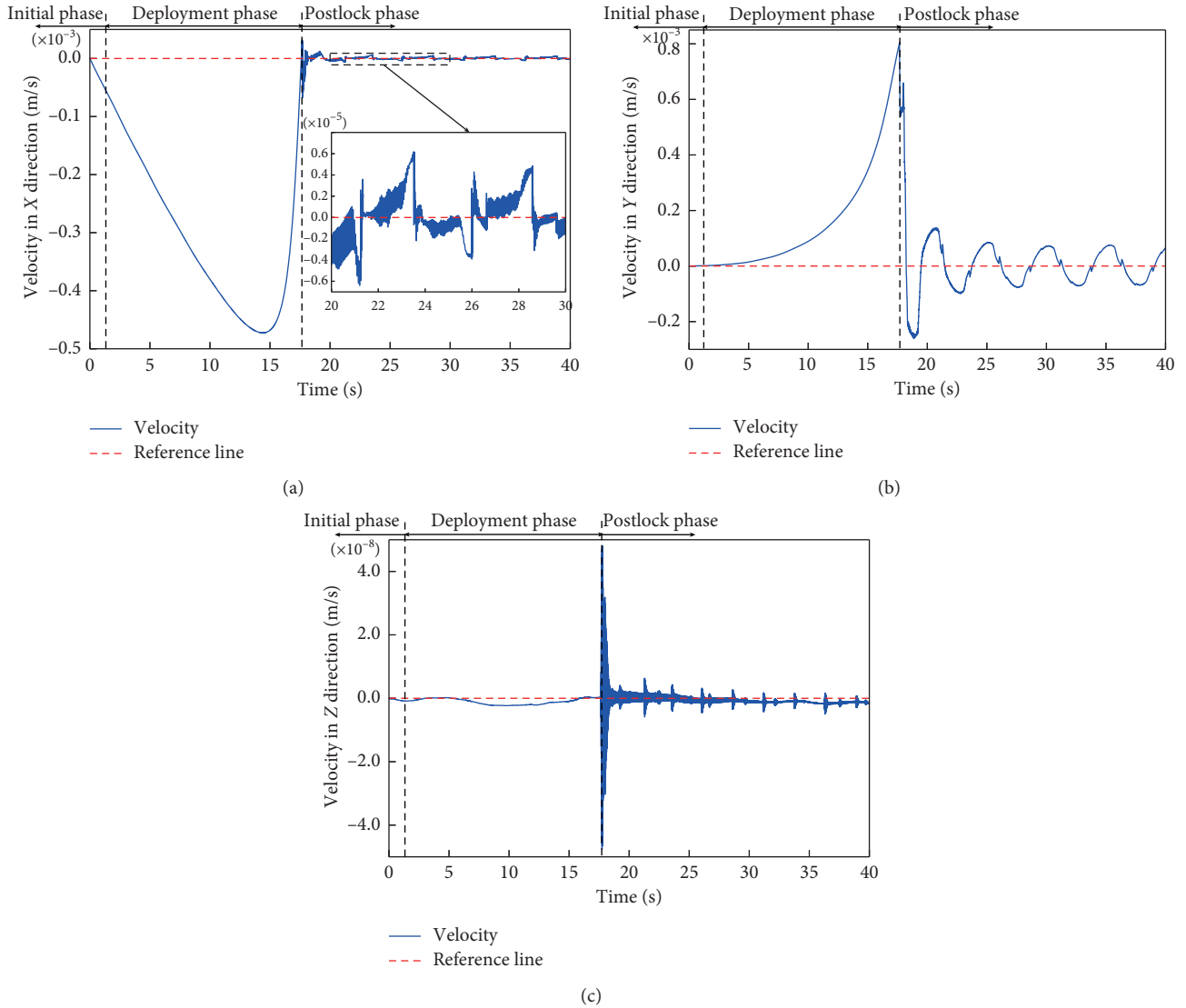


FIGURE 14: Velocity responses of the spacecraft main body in the three directions.

PD controller is sensitive to the interference noises of the system. The improper differential gain factor is likely to reduce the ability to suppress the interference noises. Therefore, the proportional and differential terms of the PD controller must be tuned jointly for optimum performance. An online adjustment for the PD controller is required to stabilize the spacecraft.

6.5. Adaptive Fuzzy PD Control Scheme. As discussed in the previous subsection, the conventional PD controller can stabilize the spacecraft system successfully. However, the parameters of the PD controller cannot be accurately determined in the presence of uncertainty and nonlinearity within the spacecraft system. This subsection presents an adaptive fuzzy PD control scheme. According to the discussion about the effects of the PD control parameters on the control performance in the previous section, we designed the fuzzy rules of the fuzzy PD controller as listed in Table 3 [4].

The membership functions were chosen to be triangular and S-shaped, due to their simplicity and sensitivity to small variations. Figure 17 shows these membership functions chosen for both inputs and outputs of each fuzzy tuner. Figures 17(a) and 17(c) depict the membership functions of the fuzzy tuner to characterize the position errors in the X and Y directions, respectively, and Figure 17(e) illustrates the membership functions to deal with attitude angle errors. Figures 17(b), 17(d), and 17(f) show the corresponding membership functions for the velocity errors. Figures 17(g) and 17(h) depict the membership functions for the increments of the proportional and differential gains, respectively.

Figure 18 compares the position and attitude responses between the fuzzy PD control scheme and the conventional PD control scheme. As shown in Figures 18(a) and 18(c), the position deviations of the spacecraft main body in the X and Y directions for the fuzzy PD controller are generally less than those for the

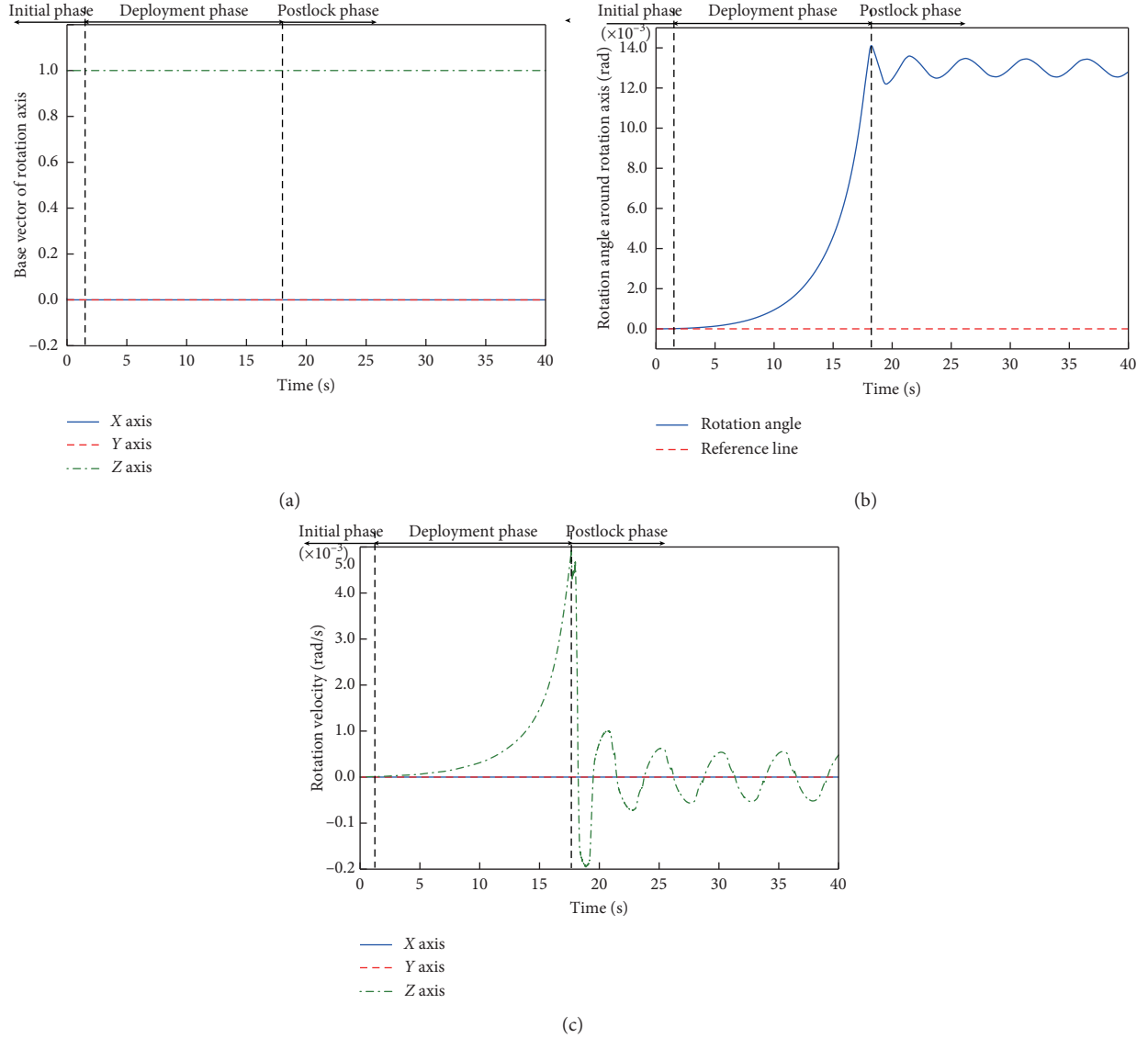


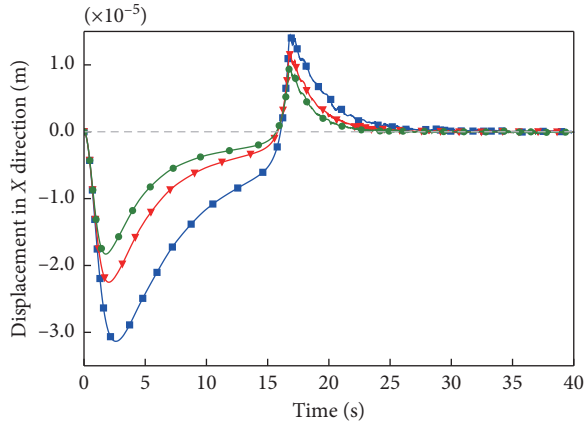
FIGURE 15: Attitude responses of the spacecraft main body.

TABLE 2: Parameters of the PD controller in the simulation cases.

Group list	Case list	K_P	K_D
Group 1	Case 1	6.5	12.5
	Case 2	12.0	12.5
	Case 3	17.0	12.5
Group 2	Case 1	17.0	5.5
	Case 2	17.0	12.5
	Case 3	17.0	25.6

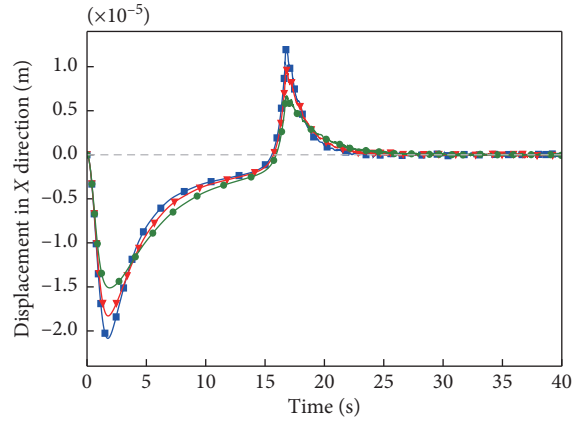
conventional PD control scheme. The conventional PD controller requires about 30 s to successfully stabilize the spacecraft system in the X direction, while this takes about 20 s for the fuzzy PD controller. In the Y direction, the spacecraft main body presents obvious fluctuations after

the solar array is locked and requires more time than the X direction to stabilize. The amplitude of these fluctuations for the fuzzy PD controller is generally smaller than those for the conventional PD controller. As depicted in Figures 18(b) and 18(d), the spacecraft's velocities fluctuate near the zero values, of which the amplitude under the fuzzy PD control is smaller than that under the conventional PD control. The results indicate that the position responses of the spacecraft main body under the fuzzy PD control are significantly improved in comparison with those under the conventional PD control. Figures 18(e) and 18(f) illustrate the contrast of the attitude responses between the fuzzy PD control and the conventional PD control. To reveal the details, the rotation angles in the deployment phase are zoomed, where the rotation angles under the fuzzy PD control are shown



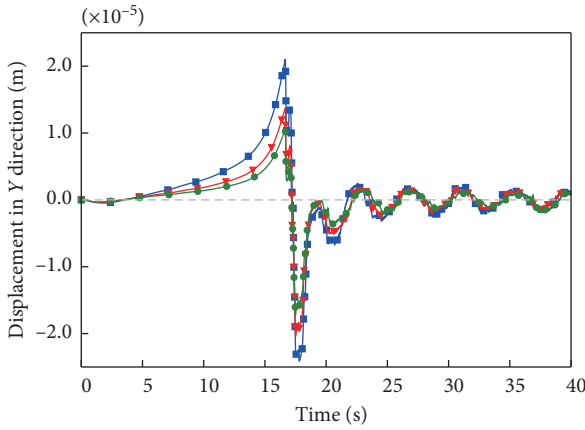
Group 1
 ■ $K_p = 6.5, K_D = 12.5$
 ▼ $K_p = 12.5, K_D = 12.5$
 ● $K_p = 17.0, K_D = 12.5$

(a)



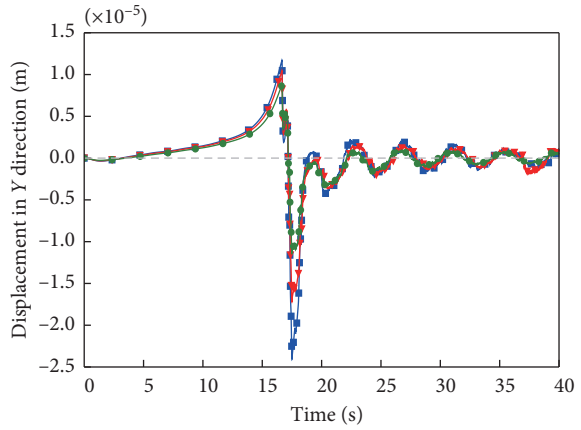
Group 2
 ■ $K_p = 17.0, K_D = 5.5$
 ▼ $K_p = 17.0, K_D = 12.5$
 ● $K_p = 17.0, K_D = 25.6$

(b)



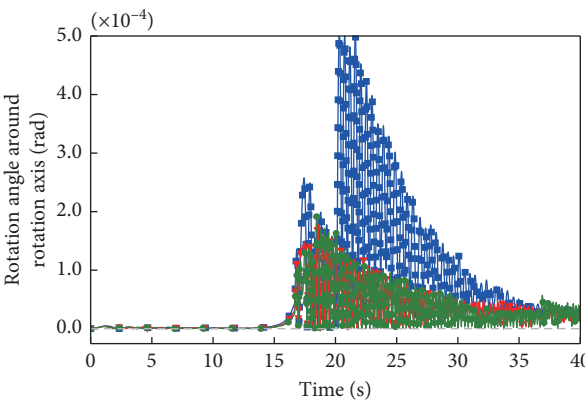
Group 1
 ■ $K_p = 6.5, K_D = 12.5$
 ▼ $K_p = 12.5, K_D = 12.5$
 ● $K_p = 17.0, K_D = 12.5$

(c)



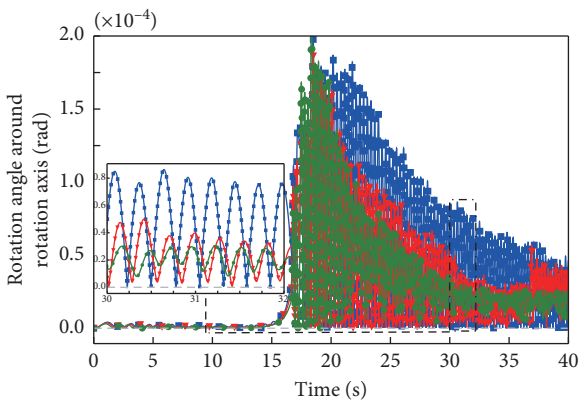
Group 2
 ■ $K_p = 17.0, K_D = 5.5$
 ▼ $K_p = 17.0, K_D = 12.5$
 ● $K_p = 17.0, K_D = 25.6$

(d)



Group 1
 ■ $K_p = 6.5, K_D = 12.5$
 ▼ $K_p = 12.5, K_D = 12.5$
 ● $K_p = 17.0, K_D = 12.5$

(e)



Group 2
 ■ $K_p = 17.0, K_D = 5.5$
 ▼ $K_p = 17.0, K_D = 12.5$
 ● $K_p = 17.0, K_D = 25.6$

(f)

FIGURE 16: Position and attitude responses of the spacecraft varying with PD controller parameters.

TABLE 3: Fuzzy rules of the fuzzy PD controller.

Rule index	Rule statements
Rule 1	If (E is N) and (EC is N), then (ΔK_P is P) and (ΔK_D is P)
Rule 2	If (E is N) and (EC is Z), then (ΔK_P is P) and (ΔK_D is Z)
Rule 3	If (E is N) and (EC is P), then (ΔK_P is N) and (ΔK_D is P)
Rule 4	If (E is Z) and (EC is N), then (ΔK_P is Z) and (ΔK_D is P)
Rule 5	If (E is Z) and (EC is Z), then (ΔK_P is Z) and (ΔK_D is Z)
Rule 6	If (E is Z) and (EC is P), then (ΔK_P is Z) and (ΔK_D is P)
Rule 7	If (E is P) and (EC is N), then (ΔK_P is N) and (ΔK_D is P)
Rule 8	If (E is P) and (EC is Z), then (ΔK_P is P) and (ΔK_D is Z)
Rule 9	If (E is P) and (EC is P), then (ΔK_P is P) and (ΔK_D is P)

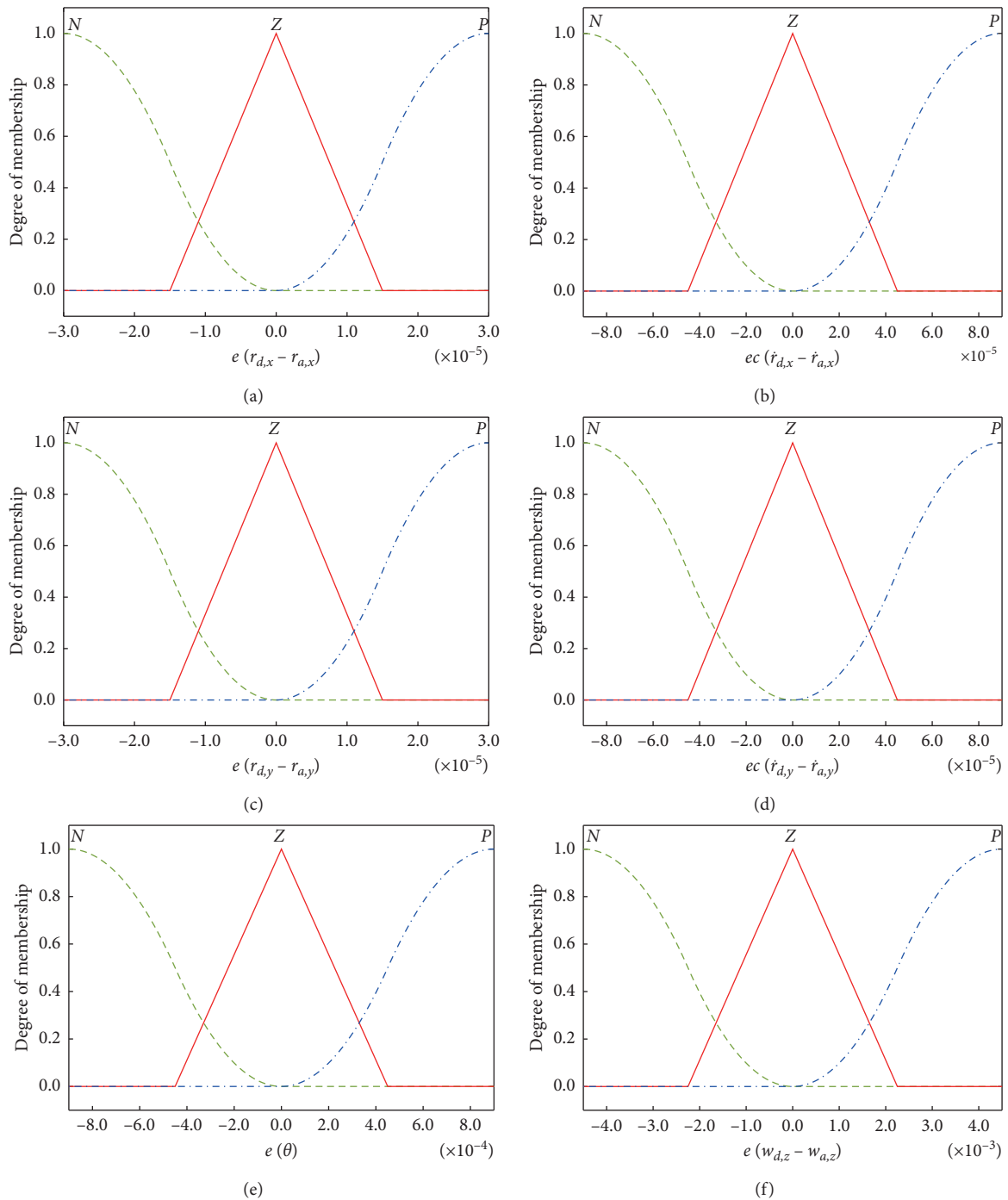


FIGURE 17: Continued.

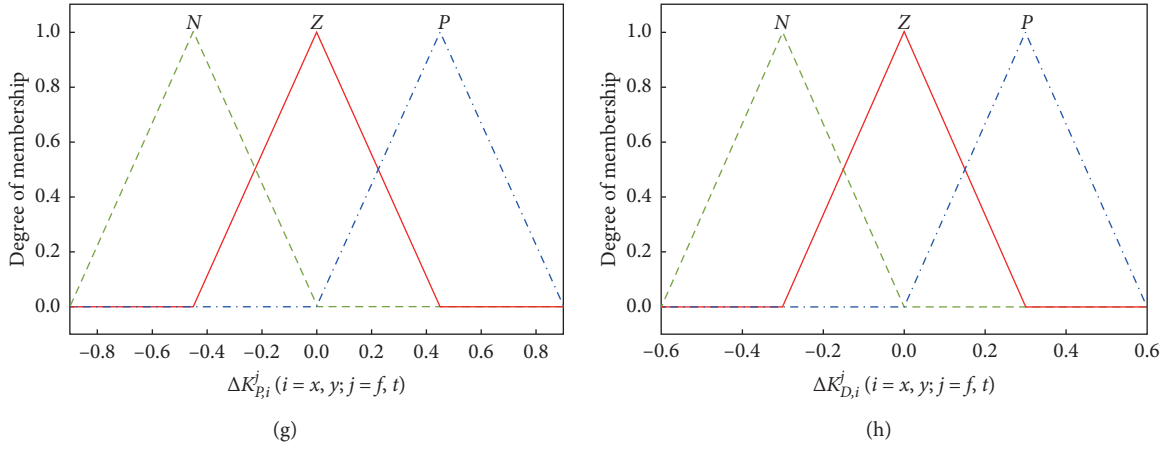


FIGURE 17: Membership functions (MFs) chosen for both inputs and outputs of each fuzzy tuner.

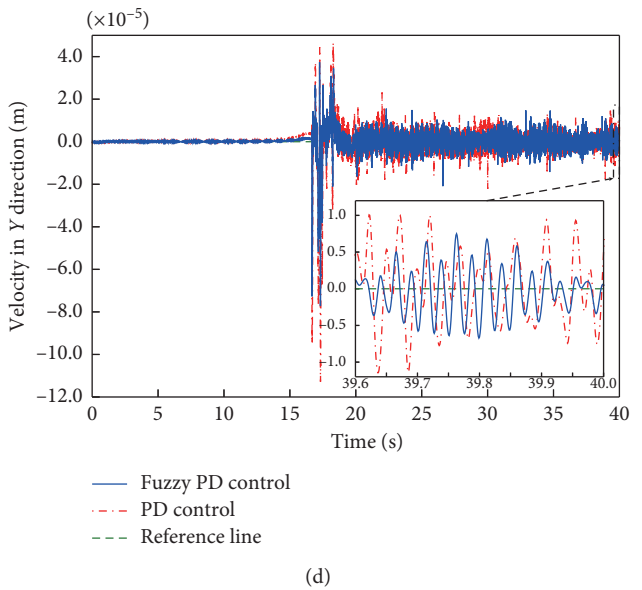
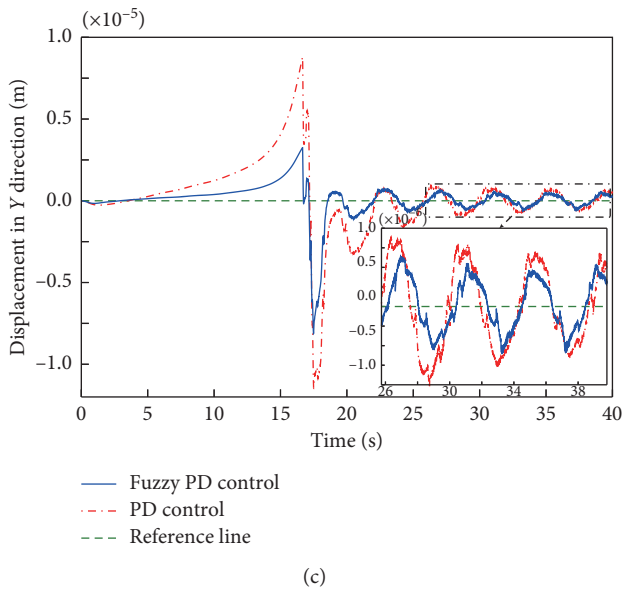
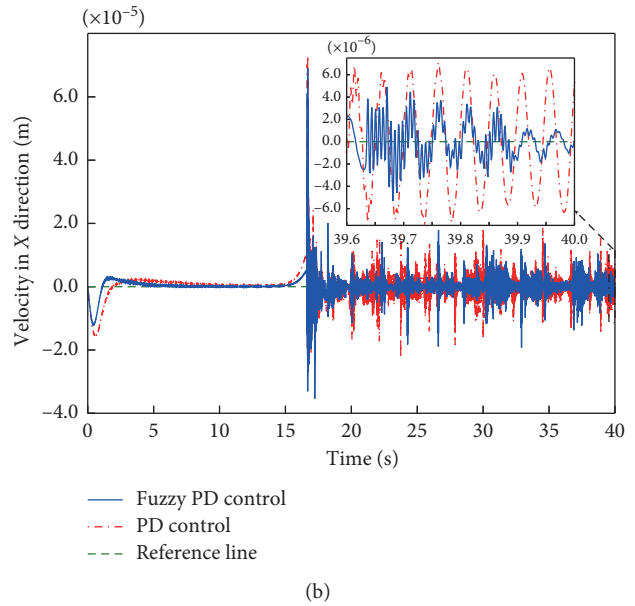
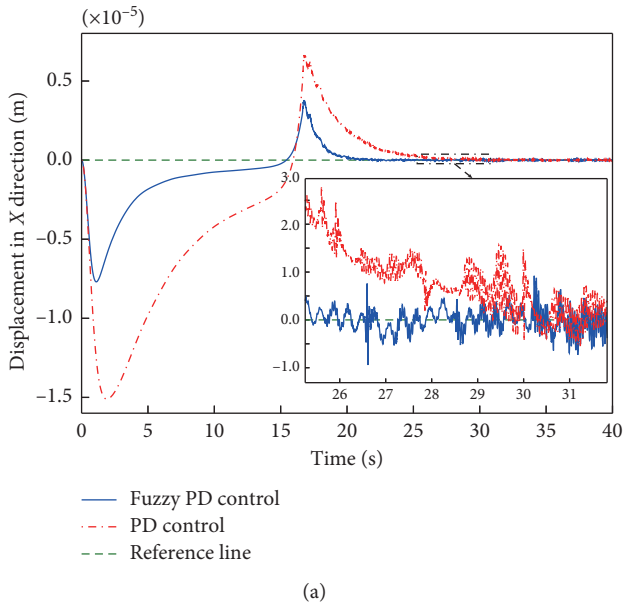


FIGURE 18: Continued.

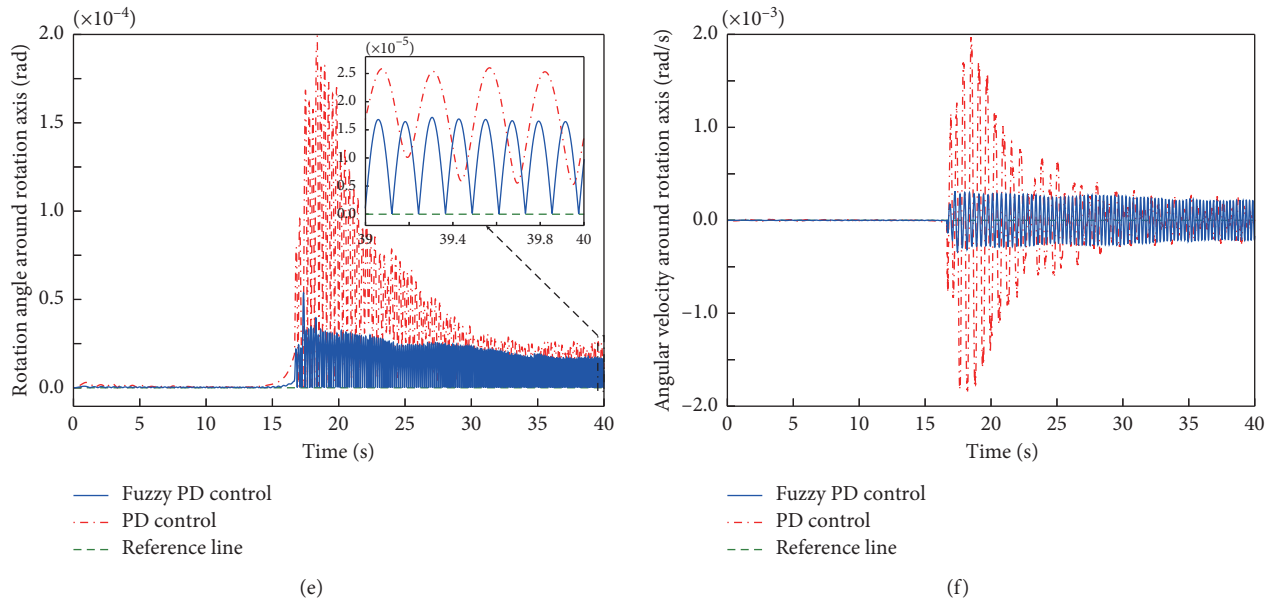


FIGURE 18: Comparisons of position and attitude responses between fuzzy PD control and conventional PD control.

to be smaller than those under the conventional PD control. When the solar panels are locked, the rotation angles of the spacecraft suddenly increase as the result of the induced impulsive forces generated from the latch mechanisms. After that, the rotation angles of the spacecraft with the fuzzy PD controller converge to zero faster than those with the conventional PD controller. The angular velocities of the spacecraft follow the similar trends. The results indicate that the fuzzy PD controller is more robust than the conventional PD controller in terms of the attitude stabilization of the spacecraft.

The adaptive fuzzy control scheme has much better performance in terms of the control precision and time response, especially in complex systems with uncertainties. Similar conclusions were also reported by previous studies. Yang et al. [68] proposed an adaptive fuzzy control scheme for coordinated robot arms in the presence of system uncertainties and compared it with a model-based controller and a conventional PD controller. The comparisons have shown that the adaptive fuzzy controller obtained the best control performance and the lowest tracking errors. Najafizadeh et al. [52] designed an adaptive fuzzy PID controller for the attitude control of the geostationary satellite. The results have shown that the adaptive fuzzy PID controller can achieve faster convergence time and higher performance. Calvo et al. [49] developed a fuzzy controller for attitude control of a satellite and compared it with the PID controller. The comparisons have proved that the fuzzy controller has superior performance on the control precision. In recent years, a versatile intelligent control scheme has appeared and performed better under the uncertainties condition. Yang et al. [69] developed a neural network-based controller to track the generated motions of a robot. Tsai et al. [32] combined the fuzzy wavelet neural networks with a

novel adaptive predictive PID control for a kind of highly nonlinear discrete-time system with time delay. In the future work, we will attempt to combine the more intelligent techniques with the classical control scheme to improve the dynamic responses of the flexible spacecraft with uncertainties and nonlinearities.

7. Conclusions

This paper presented a novel adaptive control scheme, which combines the fuzzy logic technique and PD control for attitude stabilization of a flexible spacecraft during the deployment of the solar array. The dynamic model of a constrained rigid-flexible coupling spacecraft system with a composite laminated solar array was first proposed. The validity of our dynamic model was verified by comparing with the cosimulation of the ADAMS-ABAQUS software. The comparison results showed that the proposed model can adequately describe the deployment dynamics of a solar array that composed of composite laminated shells. The dynamic responses of the spacecraft during the deployment of the solar array were then investigated. The deployment of the composite laminated solar array mainly causes the position deviations in X and Y directions, as well as the spacecraft main body approximately rotating around the Z axis. Moreover, a control scheme for attitude stabilization of the flexible spacecraft was proposed. The parameters of a PD controller play an evident role in the control performance for improving the displacement and attitude responses of the spacecraft. However, these parameters cannot be accurately determined. The effectiveness of the proposed adaptive fuzzy PD control scheme was evaluated by comparing with that of the conventional PD controllers. The comparison results revealed the superiorities of the proposed adaptive fuzzy

PD controller over the conventional PD controller. The results further provide inspiration for the design of control strategies for complex spacecraft with uncertainty and nonlinearity.

Appendix

A. Virtual Work for Inertial Force and External Force

Assuming that the point Q is associated with an infinitesimal mass element dm , the virtual work done by the inertial force can be expressed as

$$\begin{aligned}\delta W_{\text{ine}}^b &= - \int_m (\delta r_Q)^T \ddot{r}_Q dm, \\ &= - \int_m (\delta r^T + \delta \pi^T \tilde{s}' A^T) (\ddot{r} + A \tilde{\omega} s' + A \tilde{\omega} \tilde{\omega} s') dm, \\ &= - \delta r^T \ddot{r} \int_m dm - \delta r^T (A \tilde{\omega} + A \tilde{\omega} \tilde{\omega}) \int_m s' dm - \delta \pi^T \\ &\quad \int_m s' dm A^T \ddot{r} - \delta \pi^T \int_m \tilde{s}' \tilde{\omega} s' dm - \delta \pi^T \int_m \tilde{s}' \tilde{\omega} \tilde{\omega} s' dm.\end{aligned}\quad (\text{A.1})$$

If an external force f (per unit mass) act at point Q , the virtual work done by the external force can be obtained by

$$\begin{aligned}\delta W_{\text{ext}}^b &= \int_m \delta(r^P)^T f dm, \\ &= \int_m (\delta r^T + \delta \pi^T \tilde{s}' A^T) f dm, \\ &= \delta r^T \int_m f dm + \delta \pi^T \int_m \tilde{s}' A^T f dm, \\ &= \delta r^T \int_m f dm + \delta \pi^T \int_m \tilde{s}' f dm.\end{aligned}\quad (\text{A.2})$$

Since the origin of the local coordinate frame x - y - z is located at the center of mass of the body, the total mass of the body can be expressed as

$$m = \int_m dm. \quad (\text{A.3})$$

The total external force is acting on the body with the following expression:

$$F_b = \int_m f dm. \quad (\text{A.4})$$

The torque of the external forces with respect to the origin of the local coordinate frame is

$$n = \int_m \tilde{s}' f dm. \quad (\text{A.5})$$

The constant inertia tensor can be defined as

$$J = - \int_m \tilde{s}' \tilde{s}' dm = \int_m \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xy & -yz & x^2 + y^2 \end{bmatrix} dm. \quad (\text{A.6})$$

And the following mathematical relation satisfies

$$\int_m s' dm = 0. \quad (\text{A.7})$$

Substituting (A.3), (A.6), and (A.7) into (A.1) yields

$$\delta W_{\text{ine}}^b = -\delta r^T m \ddot{r} - \delta \pi^T J \dot{\omega} - \delta \pi^T \tilde{\omega} J \tilde{\omega}. \quad (\text{A.8})$$

Substituting (A.4) and (A.5) into (A.2) yields

$$\delta W_{\text{ext}}^b = \delta r^T F_b + \delta \pi^T n. \quad (\text{A.9})$$

Using (8) and (12), the following expression can be obtained as

$$\begin{aligned}\dot{\omega} &= 2G\ddot{p}, \\ \tilde{\omega} &= 2G\dot{G}^T, \\ \delta \pi &= 2G^T \delta p.\end{aligned}\quad (\text{A.10})$$

Then, (A.8) can be rewritten as

$$\delta W_{\text{ine}}^b = -\delta r^T m \ddot{r} - \delta p^T (4G^T J G \ddot{p} - 8\dot{G}^T J \dot{G} p). \quad (\text{A.11})$$

And (A.9) can be rewritten as

$$\delta W_{\text{ext}}^b = \delta r^T F_b + \delta p^T (2G^T n). \quad (\text{A.12})$$

B. Saint Venant–Kirchhoff Material Model

With regard to the orthotropic Saint Venant–Kirchhoff nonlinear material, the second Piola–Kirchhoff stress can be expressed as

$$S = \frac{\partial W}{\partial \varepsilon} = C : \varepsilon, \quad (\text{B.1})$$

where C is the fourth-order material modulus. The material modulus can be defined as [70]

$$C^{ijkl} = (b^i \cdot a_a)(b^j \cdot a_b)(b^k \cdot a_c)(b^l \cdot a_d) \bar{C}^{abcd}, \quad (\text{B.2})$$

where $\{a_1 a_2 a_3\}$ is the fiber coordinate frame, $\{b^1 b^2 b^3\}$ is the global coordinate frame, and \bar{C}^{abcd} is the tangent material modulus in the fiber coordinate frame. The tangent material modulus can be defined as

$$[\bar{C}^{abcd}] = \begin{bmatrix} \bar{C}^{1111} & \bar{C}^{1122} & 0 & \bar{C}^{1133} & 0 & 0 \\ \bar{C}^{1122} & \bar{C}^{2222} & 0 & \bar{C}^{2233} & 0 & 0 \\ 0 & 0 & \bar{C}^{1212} & 0 & 0 & 0 \\ \bar{C}^{1133} & \bar{C}^{2233} & 0 & \bar{C}^{3333} & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{C}^{2323} & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{C}^{1313} \end{bmatrix}, \quad (\text{B.3})$$

where

$$\begin{aligned}
\bar{C}^{1111} &= \frac{E_1(1 - \nu_{23}\nu_{32})}{\Delta} & \bar{C}^{2222} &= \frac{E_2(1 - \nu_{13}\nu_{31})}{\Delta} & \bar{C}^{3333} &= \frac{E_3(1 - \nu_{12}\nu_{21})}{\Delta} \\
\bar{C}^{1122} &= \frac{E_1(\nu_{21} + \nu_{31}\nu_{23})}{\Delta} & \bar{C}^{1133} &= \frac{E_3(\nu_{13} + \nu_{12}\nu_{23})}{\Delta} & \bar{C}^{2233} &= \frac{E_2(\nu_{32} + \nu_{12}\nu_{31})}{\Delta} \\
\bar{C}^{1212} &= G_{12} & \bar{C}^{2323} &= G_{23} & \bar{C}^{1313} &= G_{13} \\
\Delta &= 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}
\end{aligned} \tag{B.4}$$

where E_1 , E_2 , and E_3 are Young's moduli; G_1 , G_2 , and G_3 are the shear moduli; and ν_{12} , ν_{13} , ν_{23} , ν_{21} , ν_{31} , and ν_{32} are Poisson's ratios.

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 paper.

References

- [1] S.-J. Guo, H.-Q. Li, and G.-P. Cai, "Deployment dynamics of a large-scale flexible solar array system on the ground," *The Journal of the Astronautical Sciences*, vol. 66, no. 3, pp. 225–246, 2019.
- [2] A. Kote, K. Balaji, B. S. Nataraju, and V. C. Aralimatti, "Effect of solar array deployment on spacecraft attitude," *Journal of Spacecraft Technology*, vol. 17, pp. 1–8, 2007.
- [3] Z. Bai and Y. Zhao, "Attitude motion of spacecraft during oblique solar panel deployment," *Aircraft Engineering and Aerospace Technology*, vol. 84, 2012.
- [4] H.-Q. Li, X.-F. Liu, L.-C. Duan, and G.-P. Cai, "Deployment and control of spacecraft solar array considering joint stick-slip friction," *Aerospace Science and Technology*, vol. 42, pp. 342–352, 2015.
- [5] A. Gloria, D. Ronca, T. Russo et al., "Technical features and criteria in designing fiber-reinforced composite materials: from the aerospace and aeronautical field to biomedical applications," *Journal of Applied Biomaterials & Biomechanics*, vol. 9, 2011.
- [6] S. A. Emam and D. J. Inman, "A review on bistable composite laminates for morphing and energy harvesting," *Applied Mechanics Reviews*, vol. 67, 2015.
- [7] B. Adhikari and B. N. Singh, "An efficient higher order non-polynomial Quasi 3-D theory for dynamic responses of laminated composite plates," *Composite Structures*, vol. 189, pp. 386–397, 2018.
- [8] M. A. Neto, J. A. C. Ambrósio, and R. e. P. Leal, "Flexible multibody systems models using composite materials components," *Multibody System Dynamics*, vol. 12, no. 4, pp. 385–405, 2004.
- [9] M. A. Neto, J. A. C. Ambrósio, and R. P. Leal, "Composite materials in flexible multibody systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 50–51, pp. 6860–6873, 2006.
- [10] J. A. C. Ambrósio, M. A. Neto, and R. P. Leal, "Optimization of a complex flexible multibody systems with composite materials," *Multibody System Dynamics*, vol. 18, no. 2, pp. 117–144, 2007.
- [11] J. Gerstmayr, H. Sugiyama, and A. Mikkola, "Review on the absolute nodal coordinate formulation for large deformation analysis of multibody systems," *Journal of Computational and Nonlinear Dynamics*, vol. 8, pp. 31012–31016, 2013.
- [12] A. A. Shabana, "Flexible multibody dynamics: review of past and recent developments," *Multibody System Dynamics*, vol. 1, no. 2, pp. 189–222, 1997.
- [13] A. A. Shabana, "Definition of the slopes and the finite element absolute nodal coordinate formulation," *Multibody System Dynamics*, vol. 1, pp. 339–348, 1997.
- [14] K. Nachbagauer, "State of the art of ANCF elements regarding geometric description, interpolation strategies, definition of elastic forces, validation and the locking phenomenon in comparison with proposed beam finite elements," *Archives of Computational Methods in Engineering*, vol. 21, no. 3, pp. 293–319, 2014.
- [15] Y. Li, C. Wang, and W. Huang, "Dynamics analysis of planar rigid-flexible coupling deployable solar array system with multiple revolute clearance joints," *Mechanical Systems and Signal Processing*, vol. 117, pp. 188–209, 2019.
- [16] Y. Li, C. Wang, and W. Huang, "Rigid-flexible-thermal analysis of planar composite solar array with clearance joint considering torsional spring, latch mechanism and attitude controller," *Nonlinear Dynamics*, vol. 96, 2019.
- [17] C. Liu, Q. Tian, and H. Hu, "Dynamics of a large scale rigid-flexible multibody system composed of composite laminated plates," *Multibody System Dynamics*, vol. 26, no. 3, pp. 283–305, 2011.
- [18] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 559–576, 2005.
- [19] M. S. Can and O. F. Ozguven, "PID tuning with neutrosophic similarity measure," *International Journal of Fuzzy Systems*, vol. 19, 2017.
- [20] J. Ye, "PID tuning method using single-valued neutrosophic cosine measure and genetic algorithm," *Intelligent Automation and Soft Computing*, 2018.
- [21] Q. Lu, M. Xiao, B. Tao et al., "Complex dynamic behaviors of a congestion control system under a novel PD1n control law: stability, bifurcation and periodic oscillations," *Chaos, Solitons and Fractals*, vol. 126, 2019.
- [22] Y. Ji, "Synchronization analysis for master and slave system under communication time delay using fractional-order PD α control," *International Journal of Dynamics and Control*, vol. 7, 2019.

- [23] S. Shi, M. Xiao, L. N. Rong et al., "Stability and bifurcation control of a neuron system under a novel fractional-order PD controller," *Science China Technological Sciences*, vol. 62, 2019.
- [24] H. Bang, M.-J. Tahk, and H.-D. Choi, "Large angle attitude control of spacecraft with actuator saturation," *Control Engineering Practice*, vol. 11, no. 9, pp. 989–997, 2003.
- [25] M. J. Sidi, *Spacecraft Dynamics and Control*, Cambridge University Press, Cambridge, UK, 2014.
- [26] H. Bang, J. Kim, and Y. Jung, "Spacecraft attitude control compensating internal payload motion using disturbance observer technique," *International Journal of Aeronautical and Space Sciences*, vol. 20, 2019.
- [27] X. Xu, H. Chen, C. Lian, and D. Li, "Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, 2018.
- [28] G. L. O. Serra and C. P. Bottura, "Multiobjective evolution based fuzzy PI controller design for nonlinear systems," *Engineering Applications of Artificial Intelligence*, vol. 19, 2006.
- [29] L. E. Ramos-Velasco, O. A. Domínguez-Ramírez, and V. Parra-Vega, "Wavenet fuzzy PID controller for nonlinear MIMO systems: experimental validation on a high-end haptic robotic interface," *Applied Soft Computing*, vol. 40, 2016.
- [30] H. Arpacı and O. F. Ozguven, "Design of adaptive fractional-order PID controller to enhance robustness by means of adaptive network fuzzy inference system," *International Journal of Fuzzy Systems*, vol. 19, 2017.
- [31] Q. Zhou, W. Wang, H. Liang, M. Basin, and B. Wang, "Observer-based event-triggered Fuzzy adaptive bipartite containment control of multi-agent systems with input quantization," *IEEE Transactions on Fuzzy Systems*, 2019.
- [32] C.-C. Tsai, F.-C. Tai, Y.-L. Chang, and C.-T. Tsai, "Adaptive predictive PID control using fuzzy wavelet neural networks for nonlinear discrete-time time-delay systems," *International Journal of Fuzzy Systems*, vol. 19, no. 6, pp. 1718–1730, 2017.
- [33] Q. Liu, J. Leng, D. Yan et al., "Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system," *Journal of Manufacturing Systems*, 2020.
- [34] Q. Zhou, P. Du, H. Li, R. Lu, and J. Yang, "Adaptive fixed-time control of error-constrained pure-feedback interconnected nonlinear systems," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 2020.
- [35] W. Wang, H. Liang, Y. Pan, and T. Li, "Prescribed performance adaptive fuzzy containment control for nonlinear multi-agent systems using disturbance observer," *IEEE Transactions on Cybernetics*, 2020.
- [36] C. C. Tsai, Z. C. Wang, C. T. Lee, and Y. Y. Li, "Intelligent adaptive trajectory tracking control for an autonomous small-scale helicopter using fuzzy basis function networks," *Asian Journal of Control*, vol. 17, 2015.
- [37] J. Zhang, Y. Zhang, and C. Xu, "Backing up a truck on Gaussian and non-Gaussian impulsive noise with extended kalman filter and fuzzy controller," *International Journal of Fuzzy Systems*, vol. 20, 2018.
- [38] Q. Zhou, W. Wang, H. Ma, and H. Li, "Event-triggered fuzzy adaptive containment control for nonlinear multi-agent systems with unknown Bouc-Wen hysteresis input," *IEEE Transactions on Fuzzy Systems*, 2019.
- [39] J. Qiu, K. Sun, T. Wang, and H. Gao, "Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 11, pp. 2152–2162, 2019.
- [40] H. Liang, X. Guo, Y. Pan, and T. Huang, "Event-triggered fuzzy bipartite tracking control for network systems based on distributed reduced-order observers," *IEEE Transactions on Fuzzy Systems*, 2019.
- [41] P. Wang and D. P. Kwok, "Analysis and synthesis of an intelligent control system based on fuzzy logic and the PID principle," *Intelligent Systems Engineering*, vol. 1, 1992.
- [42] H. Boubertakh, M. Tadjine, P.-Y. Glorennec, and S. Labiod, "Tuning fuzzy PD and PI controllers using reinforcement learning," *ISA Transactions*, vol. 49, no. 4, pp. 543–551, 2010.
- [43] X. G. Duan, H. X. Li, and H. Deng, "Robustness of fuzzy PID controller due to its inherent saturation," *Journal of Process Control*, vol. 22, 2012.
- [44] A. Kumar and V. Kumar, "A novel interval type-2 fractional order fuzzy PID controller: design, performance evaluation, and its optimal time domain tuning," *ISA Transactions*, vol. 68, 2017.
- [45] Y. Wang, Q. Jin, and R. Zhang, "Improved fuzzy PID controller design using predictive functional control structure," *ISA Transactions*, vol. 71, 2017.
- [46] A. H. Gomaa Haroun and Y. ya Li, "A novel optimized hybrid fuzzy logic intelligent PID controller for an interconnected multi-area power system with physical constraints and boiler dynamics," *ISA Transactions*, vol. 71, 2017.
- [47] A. H. Gomaa Haroun and Y.-Y. Li, "Ant lion optimized fractional order fuzzy pre-compensated intelligent pid controller for frequency stabilization of interconnected multi-area power systems," *Applied System Innovation*, vol. 2, 2019.
- [48] A. Kosari, H. Jahanshahi, and S. A. Razavi, "An optimal fuzzy PID control approach for docking maneuver of two spacecraft: orientational motion," *Engineering Science and Technology, an International Journal*, vol. 20, 2017.
- [49] D. Calvo, T. Avilés, V. Lapuerta, and A. Laverón-Simavilla, "Fuzzy attitude control for a nanosatellite in low Earth orbit," *Expert Systems with Applications*, vol. 58, pp. 102–118, 2016.
- [50] Z. Chen, L. Zhong, X. Liu, and B. Cong, "Adaptive fuzzy PD+ control for attitude maneuver of rigid spacecraft," *Asian Journal of Control*, vol. 18, no. 2, pp. 631–641, 2016.
- [51] Y.-C. Chak, R. Varatharajoo, and Y. Razoumny, "Disturbance observer-based fuzzy control for flexible spacecraft combined attitude & sun tracking system," *Acta Astronautica*, vol. 133, pp. 302–310, 2017.
- [52] N. Najafizadeh, H. Jahanshahi, and M. Fakoor, "Adaptive fuzzy PID control strategy for spacecraft attitude control," *International Journal of Fuzzy Systems*, vol. 21, 2019.
- [53] H.-Q. Li, L.-C. Duan, X.-F. Liu, and G.-P. Cai, "Deployment and control of flexible solar array system considering joint friction," *Multibody System Dynamics*, vol. 39, no. 3, pp. 249–265, 2017.
- [54] F. Birhanu, Z.-B. Chen, and W. Ma, "Modeling and simulation of satellite solar panel deployment and locking," *Information Technology Journal*, vol. 9, 2010.
- [55] E. J. Haug, "Computer aided kinematics and dynamics of mechanical systems. volume 1: basic methods," *Journal of Manufacturing Systems*, vol. 11, 1992.
- [56] H. Yamashita, A. I. Valkeapää, P. Jayakumar, and H. Sugiyama, "Continuum mechanics based bilinear shear deformable shell element using absolute nodal coordinate formulation," *Journal of Computational and Nonlinear Dynamics*, vol. 10, 2015.
- [57] H. Yamashita, P. Jayakumar, and H. Sugiyama, "Development of shear deformable laminated shell element and its application to ANCF tire model," in *Proceedings of the Volume 6: 11th International Conference on Multibody Systems*,

Nonlinear Dynamics, and Control, Boston, MA, USA, August 2015.

- [58] E. N. Dvorkin and K. J. Bathe, "A continuum mechanics based four-node shell element for general non-linear analysis," *Engineering Computations*, vol. 1, no. 1, pp. 77–88, 1984.
- [59] P. Betsch and E. Stein, "An assumed strain approach avoiding artificial thickness straining for a non-linear 4-node shell element," *Communications in Numerical Methods in Engineering*, vol. 11, no. 11, pp. 899–909, 1995.
- [60] U. Andelfinger and E. Ramm, "EAS-Elements for two-dimensional, three-dimensional, plate and shells and their equivalence to HR-elements," *International Journal for Numerical Methods in Engineering*, vol. 36, pp. 1413–1449, 1993.
- [61] J. C. Simo and M. S. Rifai, "A class of mixed assumed strain methods and the method of incompatible modes," *International Journal for Numerical Methods in Engineering*, vol. 29, 1990.
- [62] D. García-Vallejo, J. L. Escalona, J. Mayo, and J. Domínguez, "Describing rigid-flexible multibody systems using absolute coordinates," *Nonlinear Dynamics*, vol. 34, no. 1/2, pp. 75–94, 2003.
- [63] H. Cheng and K. C. Gupta, "An historical note on finite rotations," *Journal of Applied Mechanics*, vol. 56, 1989.
- [64] M. Geradin and D. Rixen, *Mechanical Vibration*, Wiley, Hoboken, NJ, USA, 1997.
- [65] D. Negrut, R. Rampalli, G. Ottarsson, and A. Sajdak, "On an implementation of the hilber-hughes-taylor method in the context of index 3 differential-algebraic equations of multibody dynamics (DETC2005-85096)," *Journal of Computational and Nonlinear Dynamics*, vol. 2, no. 1, pp. 73–85, 2006.
- [66] B. Hussein, D. Negrut, and A. A. Shabana, "Implicit and explicit integration in the solution of the absolute nodal coordinate differential/algebraic equations," *Nonlinear Dynamics*, vol. 54, no. 4, pp. 283–296, 2008.
- [67] Y. Zhang, Q. Tian, L. Chen, and J. Yang, "Simulation of a viscoelastic flexible multibody system using absolute nodal coordinate and fractional derivative methods," *Multibody System Dynamics*, vol. 21, 2009.
- [68] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.
- [69] C. Yang, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang, "Biologically inspired motion modeling and neural control for robot learning from demonstrations," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, pp. 281–291, 2019.
- [70] L. Vu-Quoc and X. G. Tan, "Optimal solid shells for non-linear analyses of multilayer composites. I. Statics," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 9-10, pp. 975–1016, 2003.

Research Article

Optimization of Fuel Injection Control System of Two-Stroke Aeroengine of UAV

Yixuan Wang,¹ Yan Shi ,^{1,2} Maolin Cai,¹ Weiqing Xu ,¹ Jian Zhang,¹ Wei Zhong,^{2,3} and Na Wang ¹

¹School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

²Jiangsu Provincial Key Laboratory of Advanced Manufacture and Process for Marine Mechanical Equipment, Zhenjiang 212003, China

³School of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China

Correspondence should be addressed to Yan Shi; yesoyou@gmail.com and Na Wang; lion_na987@buaa.edu.cn

Received 1 May 2020; Accepted 8 June 2020; Published 9 July 2020

Guest Editor: Juan Sandoval

Copyright © 2020 Yixuan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Power efficiency of two-stroke spark-ignition engine is generally low because improper amount of fuel injection leads to a lot of unburned fuel loss during the engine working process. However, parameters of the fuel injection system are hard to confirm by aviation experiments due to expensive test costs. This paper proposes a method of calibrating injection parameters of two-stroke spark-ignition engine based on thermodynamic simulation and parameter optimum algorithm. Firstly, the one-dimensional thermodynamic model is built according to the internal structure and thermodynamic process of the engine; then, the model parameters are corrected according to the operating principle of the injector; after experimental verification of the model, considering both the engine power sufficiency and fuel economy, Analytic Hierarchy Process method is applied to look for the optimal injection amount and fuel injection advance angle at different engine working speeds; finally, an aeroengine experiment station with an electronic fuel injector system is built. Through simulation and experiment studies, it can be seen that when the engine speed changes from 3000 to 3500 RPM, the oil consumption rate of the optimal results is higher than that of the previous ones; when the aeroengine speed is higher than 4000 RPM, the oil consumption rate results of the optimal method are 10% to 27% higher than the original results. This paper can be a reference in the optimization of UAV aircraft engine.

1. Introduction

Two-stroke engine has been widely applied in the power system of small aerial equipment fuel-powered UAV because of the advantages of strong explosive and small size [1]. However, model selection of engine is always difficult for the design of fuel-powered UAV's power system. That is because, in the flight simulation environment of UAV, the output characteristics of the engine are hard to accurately predict, especially it is hard to find a matching fuel supply system. Traditional fuel supply method for two-stroke engine is using carburetor which can mechanically atomize fuel during the engine working process [2]. Nevertheless, automatic control cannot be achieved in engine with a carburetor, and the engine can hardly automatically adapt to

the flight condition variation of UAV. Electronic fuel injector (EFI) has been widely developed in the area of engine fuel supply due to its superiorities of controllability and favourable characteristics [3]. Performance of the engine with the EFI system has been generally studied by setting up experimental stations which can test the output speed, torque, air-fuel ratio (AFR), cylinder pressure, and exhaust contents, and the researchers do a lot of experiments to optimize the engine structure or control method [4, 5].

Although the engine experiments are designed more and more realistic in recent time, there is still some distance between test results to the real application. In addition, the traditional two-stroke engine test stations are always designed more suitable for the ground vehicles because the test torque is always added by means of electromagnetism,

which is hard to test the output power of an aeroengine with a propeller. Furthermore, in order to get accurate results, the experiment conditions have to be prepared strictly such as high precision sensors and stable environment, which will greatly increase the research cost. Last but not the least, it is dangerous and inaccurate to simulate extreme working conditions by engine station tests.

Therefore, research studies have paid more and more attention to the engine working process simulation by mathematical models. In order to estimate engine performance such as cylinder pressure, heat release rate, and fuel consumption, Venkatraman and Devaradjane [6] build the working mathematical model of a 4-stroke engine. The simulation model includes cylinder state equation, heat transfer process, ignition delay, combustion duration, and NO_x formation. In addition, based on the mathematical model, Venkatraman and Devaradjane [7] design engine experiments for demonstration. In their works, the heat release rate, brake thermal efficiency, carbon monoxide, hydrocarbon, and so on are predicted through the model, and the experiments verify that the rectification model coincides with the reality. Furthermore, in the combustion model, Wiebe heat release function is applied based on the exponential rate of the chemical reactions. Wiebe equations have been implemented by Miyamoto et al. [8], and one of the equation factors is considered to be important which is called "rate of heat release." Ganapathy et al. [9] have employed a thermodynamic model based on two-zone Wiebe heat release function to simulate the performance of new fuel engine. Raut [10] also use an exponential rate-based Wiebe heat release model, and the Pflaum formula is applied in the estimation of empirical coefficient of the heat transfer process. From these works, it can be seen that engine performance study by using the mathematical model method is effective.

GT-Power is the leading engine simulation software based on one-dimensional gas dynamic which represents the flow and transfer in the components of the engine system, and more and more scientists and engineers have applied the computing tool in engine prediction in order to improve the control performance or reduce the emission. Kassa et al. [11] have leveraged experimental data from a 6-cylinder engine to a GT-Power model to better understand the distribution of the port-injected fuel across cylinders under several operating conditions. Rahimi-Gorji et al. [12] have optimized the performance and fuel consumption according to the weather conditions by combining the artificial neural network and GT-Power model, and pressure, temperature, and humidity of the incoming air are considered in the network to obtain a better engine performance. Alves et al. [13] apply GT-Power in the engine intake system design, and the best intake runner length and diameter configuration of each speed for a four-stroke and single cylinder engine is found to get the optimum volume efficiency. Trajkovic et al. [14] build the GT-Power model of a 2-stroke engine to study the effect of different parameters and their effect on pneumatic hybrid performance. From the works above, the mathematical model built by GT-Power is proved to be effective to predict and improve the engine characteristics. However, these papers mainly focus on the engine structure rather than the control strategy of the EFI system.

In order to match the power system of a kind of downsized fuel-powered UAV, the characteristics of the aeroengine, including output speed and output power, should be analyzed based on the GT-Power model with a fixed structure. Furthermore, key control parameters of the matched EFI system should be confirmed for the aeroengine application. Calculations of the engine power based on the GT-Power module have been researched. Yang and Zhu [15] have developed a mixed valve and crank-based engine model for a dual-stage turbocharged engine. Under different loading states, the output torque and released AFR of the engine are simulated, and values of the fuel pulse width are calculated for a reference for the engine control unit (ECU) design. Menacer and Bouchetara [16] have applied the GT-Power model to study the effect of the inject fuel mass flow on the brake power and indicate power under the certain ignition advanced angle, compression ratio, and output speed. In their work, the maximum power and economy corresponding to the optimal speed are determined. Wei et al. [17] have adopted a serial of experiment data in a GT-Power model of a water-cooled four-stroke engine, and lengths of the opening and closing delay times are optimized, and an optimal inject fuel mass flow is optimized. Moreover, Yang et al. [18] have designed the controlled fuel process and studied the different intake air parameters to improve the engine dynamic performances. However, in these works, the engine confirmatory experiments are far away from the real application of the aeroengine because the torque propeller mainly comes from the propeller air resistance. In addition, some of the works are short of detailed experiment description and relative theory basis, so it is important for us to provide a theoretical model reference for the aeroengine fuel supply system in order to avoid multiple engine parameter tests which can cause huge development costs. Furthermore, ECU controls the injector of the EFI system of the aeroengine, and the electrified injector is opened and atomizes the input high pressure gasoline into the engine manifold [19]. However, because of the electromagnetic force characteristic of the injector, the dynamic response of the injected fuel mass flow will affect the precision of the supplied fuel. Therefore, based on the model results of the theory fuel flow, it is necessary to analyze the dynamic response of the injector and compensate the fuel spray, and then we can get a confirmed EFI control parameter which can provide optimum performance for the aeroengine.

In this paper, we firstly analyze the structure of the aeroengine, and one-dimensional GT-Power mode of the engine is established. Furthermore, several parameter correction methods are proposed. Based on the simulation results of the corrected model, the analytic hierarchy method is applied to optimize the fuel injection control system. Engine experiment results which use the optimize injection MAP demonstrate that the oil consumption rate can be improved differently.

2. Methodology of Model

2.1. Subject Introduction. In this paper, the studied two-stroke aeroengine with the model of DLE170 has two

opposing twin cylinders and mainly includes two air cylinders chambers, two pistons, one crankcase, and one crankshaft. As shown in Figure 1, each of the cylinders has a scavenging channel and an exhaust vent, and all the ports are without valves. That means that opening and closing of the holes on the cylinder chambers depend on the movement of the pistons. In addition, main parameters of the aeroengine are shown in Table 1.

When the two-stroke aeroengine starts to work, during the first stroke, firstly, the air-fuel mixture is sucked into the crankcase, and the scavenging port is opened when the pistons move from the bottom dead center (BDC) until the crankshaft rotates to the intake valve closed (IVC) angle, which can be seen in Figure 2. The exhaust port is opened from BDC until the crankshaft rotates to the emission valve closed (EVC) angle; then, the air-fuel mixture is compressed, and at the ignition advance angle before the top dead center (TDC), the engine is sparked. During the piston power process before the exhaust valve opened (EVO) angle, both the exhaust and scavenging ports are closed, and the cylinder chambers are hermetic which can ensure that the piston gets the maximum power. The ignition advanced angle (θ) is usually set to 5~15 degrees ahead of TDC. Because the compressed ratio of the engine is relatively high and the rated engine speed is fast, the ignition advanced angle is set at 15 degrees. The fuel in the crankcase comes from the electronic injector as a certain air-fuel ratio (λ), and the ratio is determined by the average intake fuel flow (\dot{m}_f) and air flow into the manifold (\dot{m}_{man}), which is controlled by inject fuel pulse width (P_w), inject fuel pressure (p_f), and throttle opening degree (α). The relative parameters can also be seen in Table 1.

The engine working process is designed, as shown in Figure 3, according to the working principle of the one-dimensional simulation software GT-Power. In this picture, it can be seen that connected to the inlet port there are two symmetrical crankcase chambers with numbers 1 and 2. When the air-fuel mixture is flows into the crankcase, it is generally assumed that two homogeneous mounts of oil and gas are divided by the crankshaft and then flow into the two cylinders. Furthermore, there are also symmetrical scavenging passages, intake ports, cylinder chambers, exhaust ports, and exhaust passages. The opening and closing degrees of the two kinds of ports determine the intake time of the air-fuel mixture and the exhaust time of the emissions. According to the actual measurement results, the opening areas with the crankshaft angles are as shown in Figures 4(a) and 4(b). It should be noted that, in these figures, range of the x -coordinates is 0 to 180 degrees which is in the first working stroke. The changing area with the shaft angle is symmetrical in the second working stroke.

2.2. Modeling Method. Then, the one-dimensional GT-Power model of the two-stroke aeroengine can be set, as shown in Figure 5, according to the aeroengine working process. Structure of the aeroengine is based on the actual measure results.

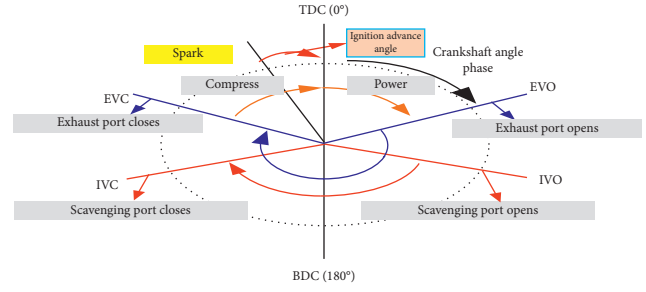


FIGURE 1: Main parts of the two-stroke aeroengine.

TABLE 1: Specifications of DLE170 engine.

Parameter	Value
Cylinder bore (mm)	52
Engine stroke (mm)	40
Connecting rod length (mm)	175
Compression ratio	9.5
TDC clearance height (mm)	2
Displacement (cc)	85X2
Intake fuel pressure (MPa)	0.3
EVO (degrees)	65
IVO (degrees)	123
Intake pressure (bar)	1
Intake temperature (K)	298
Exhaust temperature (K)	700
Exhaust pressure (bar)	1.2
Maximum performance	13 kw@ 7500 RPM
Minimum idle (RPM)	1000
Range of the throttle opening degree (degrees)	10~90
EVC (degrees)	-65
IVC (degrees)	-123

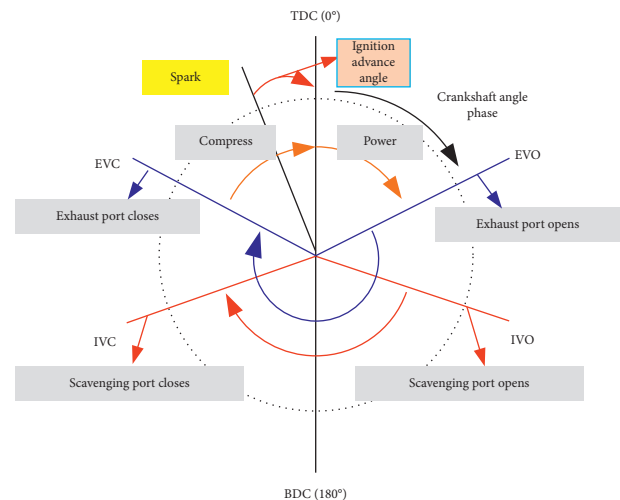


FIGURE 2: Working schematic diagram of the two-stroke aeroengine.

Main setting parameters of the fuel injector are \dot{m}_f and the set air-fuel ratio (λ_{set}). In practice, ECU controls the injector work and breaks through pulse signal with a certain

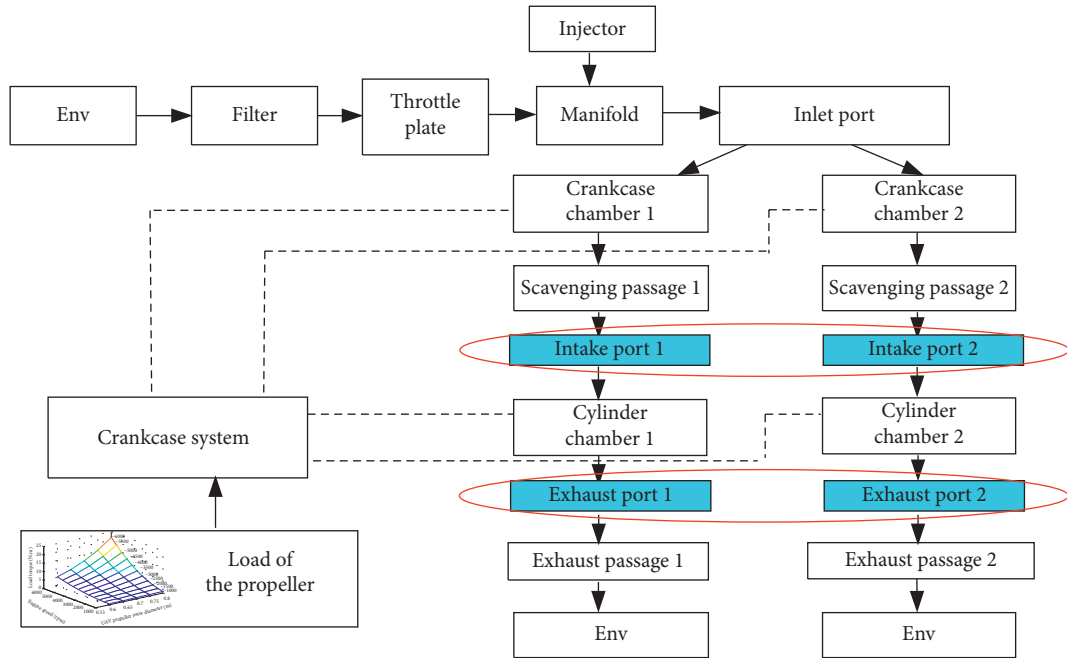


FIGURE 3: Working process of the two-stroke aeroengine.

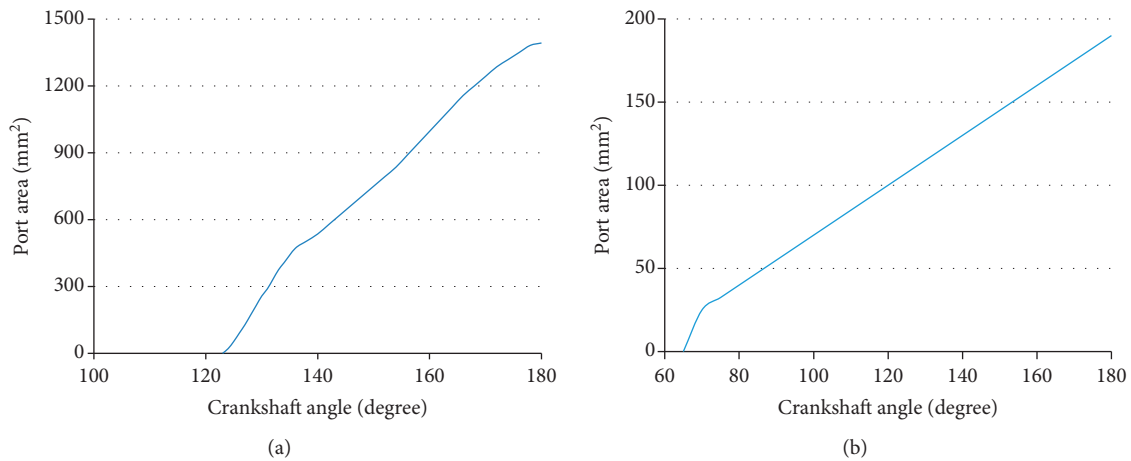


FIGURE 4: Working process of the two-stroke aeroengine. (a) Intake port area. (b) Exhaust port area.

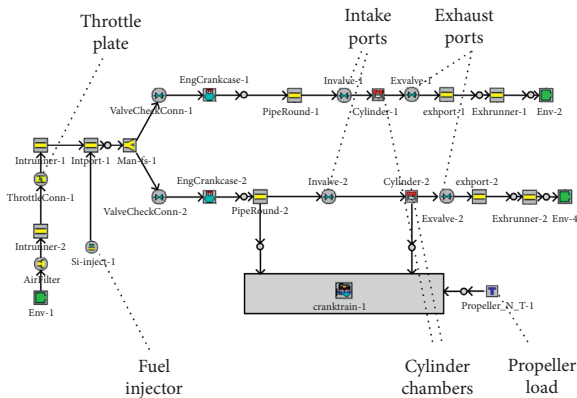


FIGURE 5: One-dimensional GT-Power model of the two-stroke aeroengine.

width. Relationship between \dot{m}_f , λ_{set} and the inject fuel pulse width (P_w) is shown in the following equation:

$$\dot{m}_f = \frac{\eta_v \rho_{ref} V_D \lambda_{set}}{(\#CYL) P_w}, \quad (1)$$

where η_v is volumetric efficiency, ρ_{ref} is reference air density used to calculate volumetric efficiency, V_D is the engine displacement, and $\#CYL$ is the number of cylinders. From this equation, we can see that P_w directly determines the injected fuel flow. In order to improve the comprehensive performance of the aeroengine, the inject fuel flow rates are calibrated under different working conditions. So, as to make the setup more intuitive, in this paper, the calibrating standard is based on the expected air-fuel ratio, and then the ECU can calculate the output P_w in the real practice. However, because the injector is driven by electromagnetic

force, the dynamic action of the needle valve should be considered in the compensation of the set pulse width.

Fuel injector working progress mainly includes three steps: the injector receives the pulse signal from ECU, the electromagnetic coil is gradually energized, and the needle valve starts to move when overcoming the spring preload; when the magnetized solenoid coil is saturated, the needle valve stops at the mechanical limit position; when the pulse signal becomes zero, the magnetic flux of the solenoid coil reduces gradually, and the needle valve will return to the normal position. Basically, all the nozzles of the electromagnetic type work in this way, and the response delay impact on the dynamic inject fuel flow caused by mechanical factors cannot be ignored. In order to deeply analyze the dynamic injection and formulate the calibration strategy, the dynamic model of the injector is built as follows:

$$U_0 = Ri + N \frac{d\Phi_b}{dt}, \quad \text{when electrified,} \quad (2)$$

$$0 = (R + R_0)i + N \frac{d\Phi_b}{dt}, \quad \text{when not electrified.}$$

These equations are magnetic flux when the injector is electrified and not electrified, where R is the basic resistance of the electrified coil loop, R_0 is the protective resistance, Φ_b represents total magnetic circuit, N is the number of the coil, i is the current in the loop, and U_0 is the driving voltage. The electromagnetic force (F_m) on the needle valve when the coil is electrified is as follows:

$$F_m = \frac{\mu_0 (iN)^2 S}{2\delta^2}, \quad (3)$$

where μ_0 represents permeability of vacuum, S is cross-section of the air gap, and δ means length of the working air gap. Kinetic equation of the magnetic needle valve is

$$F_m - F_0 - kx + F_{\text{fuel}} = m_v \frac{d^2 x}{dt^2}, \quad (4)$$

where F_0 is the initial tension of spring, k is the spring stiffness, x is the displacement of the needle valve, m_v is the mass of needle valve, and F_{fuel} is fuel pressure force on the needle valve. When the needle valve is opened, the high pressure fuel erupts and produces spray into the manifold. The equation of the fuel flow is as follows according to orifice compensation principle:

$$\dot{m}_{\text{fi}} = C_d A_0 \sqrt{2\rho_f (p_f - p_m)}, \quad (5)$$

where \dot{m}_{fi} is the instantaneous inject fuel mass flow, C_d is discharge coefficient, A_0 is aperture area, ρ_f is the fuel density, and p_m is the atmosphere pressure. Based on the equations, the dynamic displacement of the needle valve is calculated under different spring stiffness values. As shown in Figure 6(a), when the initial spring tension force F_0 is set at 5.5 N, the response of the valve displacement will not keep pace with the control signal. However, when F_0 is set at 9.5 N, pulse width of the needle displacement is much shorter than the control signal, which can be shown in Figure 6(b), and that will lead to insufficient of the inject fuel.

Therefore, the pulse width of the valve displacement can be adjusted to be the same with that of the control signal by setting the spring tension force F_0 . As shown in Figure 6(c), pulse width of the dynamic displacement of the needle valve is approximate to the control signal except at the beginning of the period, where there is a rise process which causes injection control error.

This paper proposes a compensation method for the injection control error. As shown in Figure 7, it can be seen that the displacement compensation time is equal to the current delay time. Compensation area of the rise process is approximate as a triangle. Therefore, the compensation width (P_c) is shown in the following equation:

$$P_c = \frac{D_i}{2}, \quad (6)$$

where D_i is the current time delay. Equation (1) can be amended as follows:

$$\dot{m}_f = \frac{\eta_v \rho_{\text{ref}} V_D \lambda_{\text{set}}}{(\#CYL)(P_w + P_c)}. \quad (7)$$

Propeller load can be calculated according to different working conditions based on standard strip analysis. As known from the calculation, load torque of the propeller mainly depends on engine speed (n) and propeller rotor diameter (r_p). Main formula of the torque is as follows:

$$M_R = \sum (\Delta D \cos \beta + \Delta L \sin \beta) r_p, \quad (8)$$

where M_R is the propeller torque, ΔD is differential form of the drag force, ΔL is differential form of the lift force, β is the intake air flow angle, and r_p is the propeller radius. Then, the parameters are confirmed according to a blade material; then, we can get the torque MAP in horizontal direction under different working conditions, which is shown in Figure 8. From Figure 8, we can see that the load torque does not increase linearly with increasing engine speed and the UAV propeller rotor diameter. However, we can substitute the torque MAP into the one-dimensional model by the linear interpolation method.

3. Experiments and Optimization

3.1. Experimental Verification. Numerical simulation cannot completely replace experiment analysis, and if we want to make the simulation results reflect the engine mechanism as precisely as possible, the mathematical simulation and experiment analysis should be combined. The mathematical model needs to be verified by experiment results which mainly includes two parts: the engine structure and the combustion model. The engine structure can be verified by intake air flow experiments. That is because the engine is driven by the oil and gas combustion, and if the detected air flow is consistent with the simulation result in different conditions, we can see that the built engine model structure can provide an equal inlet air mass flow. In addition, the combustion model should be demonstrated by the cylinder pressure test. The reason is that the output power of the internal combustion engine mainly comes from the in-

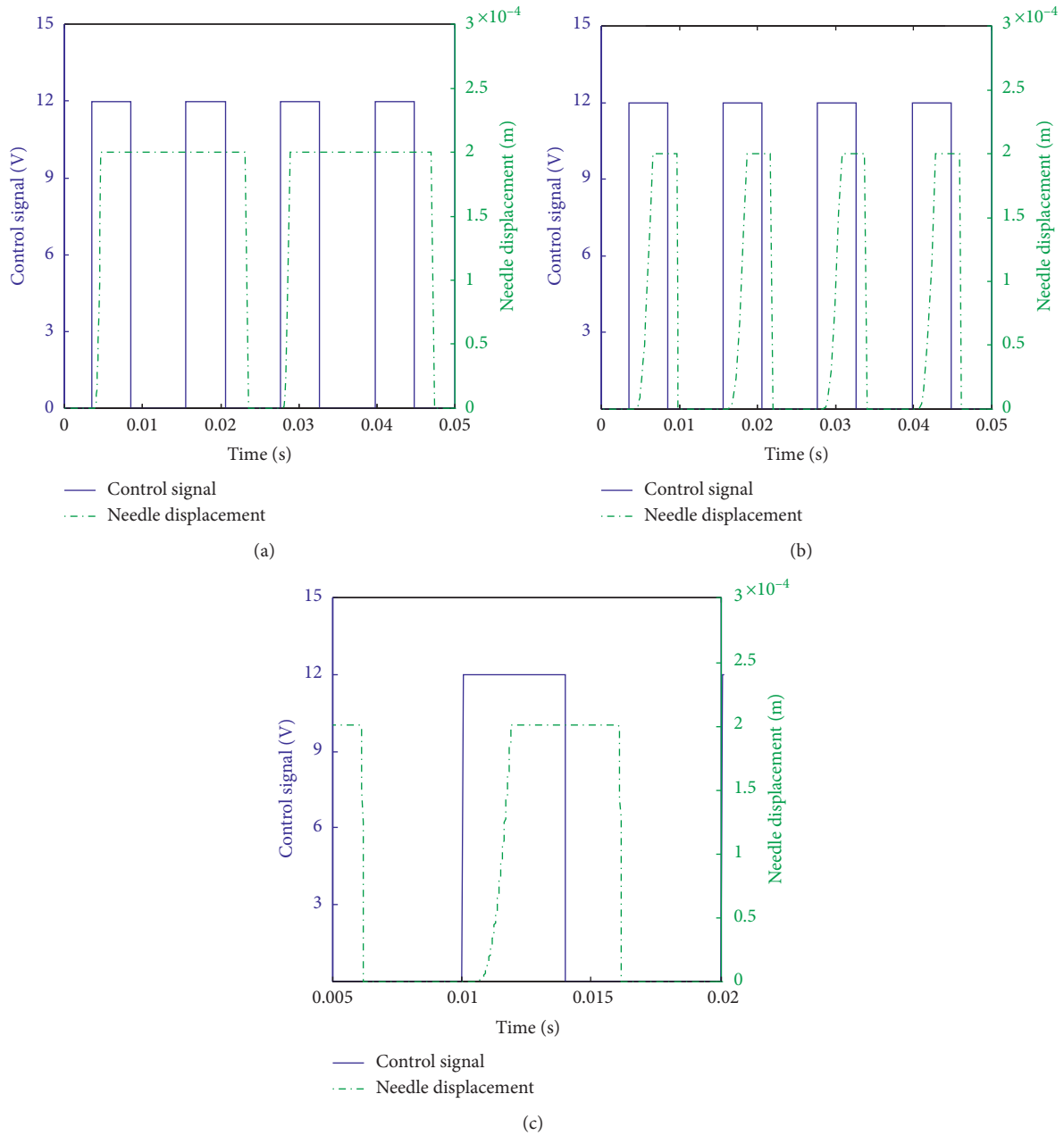


FIGURE 6: Dynamic displacement of the needle valve with different spring initial tension F_0 . (a) $F_0 = 5.5$ N. (b) $F_0 = 9.5$ N. (c) $F_0 = 8.0$ N.

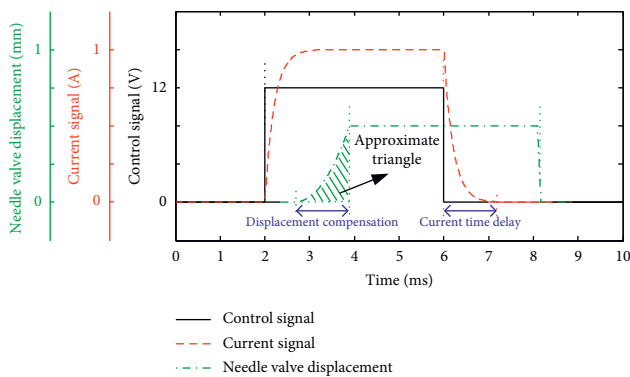


FIGURE 7: Schematic diagram of needle valve displacement compensation.

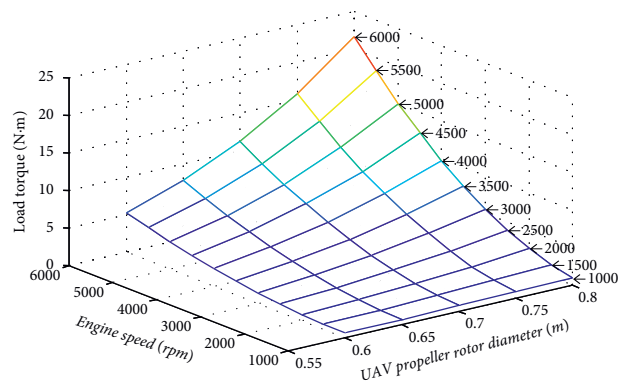


FIGURE 8: Horizontal torque of the propeller under different working conditions.

cylinder pressure, and if the detected pressure fits well with the simulation result, it can be seen that the combustion prediction model in the simulation is effective. Therefore, an engine intake air flowmeter is connected with the intake manifold, and a high frequency pressure sensor is setup on the engine cylinder. Then, we can get the air mass flow and cylinder pressure compare curves, as shown in Figures 9(a) and 9(b). From these two pictures, we can see that the result errors are no more than 5% and can demonstrate the mathematic model to be effective.

Injection fuel compensation can be demonstrated by ECU experiments. ECU gets trigger signal and outputs pulse signal with a certain pulse width. According to above research result, the inject fuel pulse is compensated by the delay time of the current through the electrified coil loop. According to Figure 10, an ECU with the above function is designed and tested. The current signal as well as the control voltage through the fuel injector is tested. According to Figure 11, we can obtain that the current delay time is about 4 ms. Therefore, in the model, we compensate for the inject pulse by 2 ms. Repetitive experiments with different control signal widths are conducted, and according to the real application, the signal width is controlled within the range of 3.5 ms to 5.0 ms, and we obtain that the current delay time is the same. That is because their lowering processes of the current are the same. So, in the simulation model, we can set the current delay time as constant 2 ms.

Through the simulation based on the model above, the relative working parameters can be calculated. The basic simulation setting parameters are throttle opening degree (α) and the set air-fuel ratio (λ_{set}). Generally, the most concerned characteristics and evaluation indicators of the engine mainly include engine speed (n), output power (P_o), power efficiency (η), and rotational fuel consumption (γ). η and γ can be calculated as follows:

$$\eta = \frac{P_o}{P_i} = \frac{nT_o}{9550\dot{m}_f H_u}, \quad (9)$$

$$\gamma = \frac{n}{\dot{m}_f},$$

where P_i is the input power of the engine, T_o is the output torque of the engine, and H_u is the gas calorific value which is about 46000 KJ/kg. By changing the setting parameters α and λ_{set} , a group of output parameters are obtained.

3.2. Simulation Results. As shown in Tables 2–4, there are several arrays of input and output parameters. In addition, all the parameters are recorded when the engine simulations tend to be stable.

Tables 2–4 represent a part of simulation results. In this paper, the throttle opening degree (α) is changed from 10° to 90° , and the engine speed is from 2500 RPM to 6000 RPM according to the real application. What needs illustration is that, according to our a large number of experiment results, the output AFR of the engine can only be controlled within a precision of 0.5, and the general range of AFR during the engine working process is from 12 to 15.5. Therefore, in the

simulation, the input AFR value is set to every 0.5 from 12 to 15.5.

As shown in the three tables, T_o reflects the load-carrying capacity, and generally it is considered as the main indicator of grade ability in the area of ground gasoline. However, in the application of the rotorcraft UAV field, the lift force of the UAV is primarily determined by the engine speed. So, the parameter T_o is mainly considered in the start and acceleration processes. P_o is the output power of the engine, and in the case of the same displacement, output power should be bigger. However, in this paper, the fuel economy is treated as a priority, so in the engine fuel injection control, weight of η should be put more. Considering that the engine speed directly influences the lift of the UAV, the rotational fuel consumption γ reflects the fuel consumption rate at constant speed. In addition, P_c is the maximum cylinder pressure of one crankshaft rotate cycle.

Since these output characteristics affect each other, the influence rules of the injection parameter should be analyzed in order to assist in the formulation of the optimization strategies. The set AFR directly affects the oil injection flow rate, and its value always combines with that of the throttle opening degree (α). Here, α is controlled stably as 40° because the single opening degrees value can reflect the whole principle. Engine speed (n) which is as the final control quantity of the UAV power system should be simulated by stages. According to the application requirement, the interval is set as 500 RPM from 2500 RPM to 6000 RPM. Then, the tendency charts which show the relationships between the input parameters and the output characteristics are obtained, as shown in Figures 12–15.

As shown in Figures 12 and 13, it can be seen that when the throttle opening degree is constant, the output torque curves, and output power curves will have a peak at a same speed. However, it is difficult to find a regular rule between the set AFR and these two output characteristics. That is because when the throttle opening degree is set constantly, there must be a primal AFR setting value with different working conditions which is in accordance with the engine external characteristics. According to the settled weight, the optimization should be comprehensively considered. In addition, from the two figures, the curve trends are almost the same, so only one of the items can be considered when setting weights in order to reduce the amount of calculation.

It is a bit of mess in Figure 14 which represents the efficiency of the engine system. Curves in Figure 14 almost have peak values, and the peak values basically independent of the output torque and power. Therefore, the characteristic of power efficiency can be considered independently. Power efficiency is significant for the fuel economy improvement. Nevertheless, it should be secondary to the torque and power in the startup and acceleration processes in order to ensure the safety of the UAV flight.

Since the engine speed is kept steady during the flight, the oil consumption of rotation speed (γ) is the most important characteristic for fuel saving and flight endurance extension. As shown in Figure 15, the curves have troughs at the same speed point with the torque and power curves. However, in Figure 15, the arrange regular way of the curves

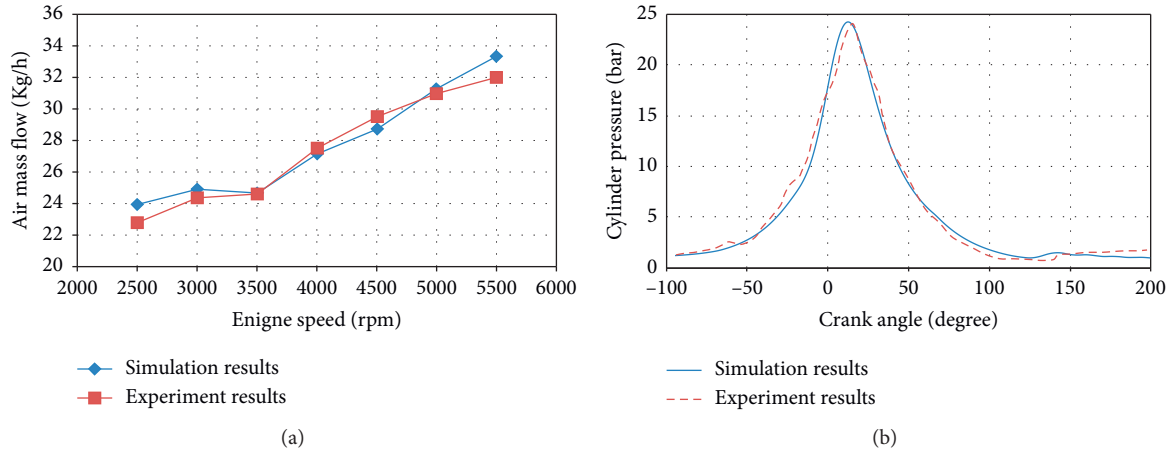


FIGURE 9: Air mass flow and cylinder pressure compare curves. (a) Air mass flow at different engine speeds. (b) Cylinder pressure when engine speed is 5000 RPM.

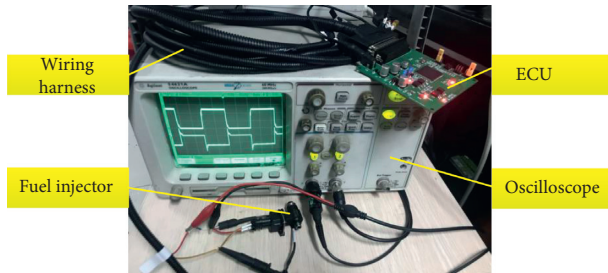


FIGURE 10: Injector test picture.

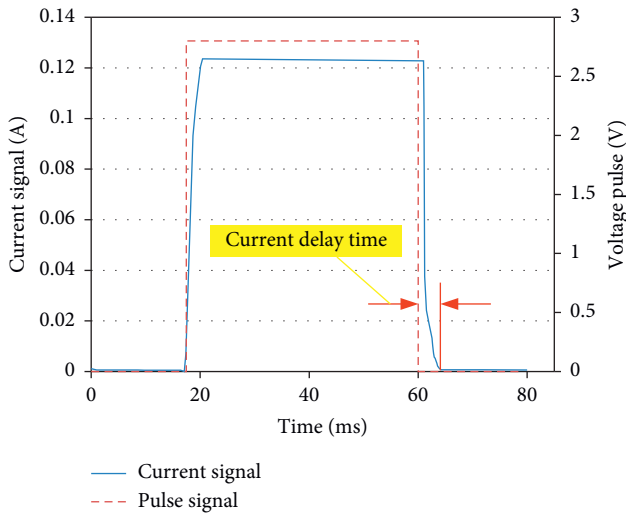


FIGURE 11: Results of the current delay time test.

of different set AFR values is different from those in Figures 12 and 13. So, in the optimization work, the oil consumption of rotation speed (γ) can be independently considered according to the setting weight in the stable flight process of UAV.

TABLE 2: Results of the GT-power simulation when $\alpha = 10^\circ$.

α ($^\circ$)	n (RPM)	λ_{set}	T_o (Nm)	P_o (kw)	η	γ (RPM-h/Kg)	P_c (MPa)
10	2500	12	9.551	2.501	0.177	2257.016	3.066
10	3000	12	7.352	2.310	0.175	2892.084	2.045
10	3500	12	7.195	3.633	0.189	2316.357	2.786
10	4000	12	8.874	3.717	0.185	2537.999	3.219
10	4500	12	14.272	6.725	0.204	1736.989	3.987
10	5000	12	10.635	5.568	0.194	2219.936	2.986
10	5500	12	6.053	3.486	0.163	3280.692	2.760
10	6000	12	5.634	3.539	0.156	3362.353	2.687

TABLE 3: Results of the GT-power simulation when $\alpha = 40^\circ$.

α ($^\circ$)	n (RPM)	λ_{set}	T_o (Nm)	P_o (kw)	η	γ (RPM-h/Kg)	P_c (MPa)
40	2500	14.5	9.122	2.388	0.149	1987.442	3.142
40	3000	14.5	8.869	2.786	0.140	1918.287	2.308
40	3500	14.5	12.132	4.447	0.213	2134.503	3.228
40	4000	14.5	12.926	5.414	0.206	1934.364	3.806
40	4500	14.5	15.451	7.281	0.179	1410.437	4.054
40	5000	14.5	13.459	7.047	0.179	1621.358	3.952
40	5500	14.5	10.109	5.823	0.203	2445.253	3.660
40	6000	14.5	9.111	5.725	0.165	2201.615	3.128

TABLE 4: Results of the GT-power simulation when $\alpha = 80^\circ$.

α ($^\circ$)	n (RPM)	λ_{set}	T_o (Nm)	P_o (kw)	η	γ (RPM-h/Kg)	P_c (MPa)
80	2500	15.5	7.003	1.833	0.141	2453.110	2.510
80	3000	15.5	9.582	3.010	0.194	2457.778	2.424
80	3500	15.5	11.009	4.035	0.229	2529.551	2.974
80	4000	15.5	13.881	5.815	0.231	2020.441	3.869
80	4500	15.5	14.826	6.987	0.167	1368.566	4.293
80	5000	15.5	12.003	6.285	0.188	1904.895	4.107
80	5500	15.5	8.315	4.789	0.138	2018.235	3.044
80	6000	15.5	6.094	3.829	0.205	4088.545	2.391

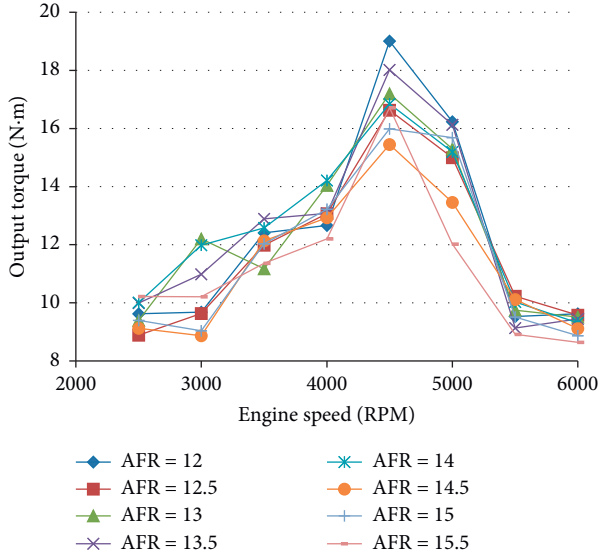


FIGURE 12: Output torque at different engine speeds.

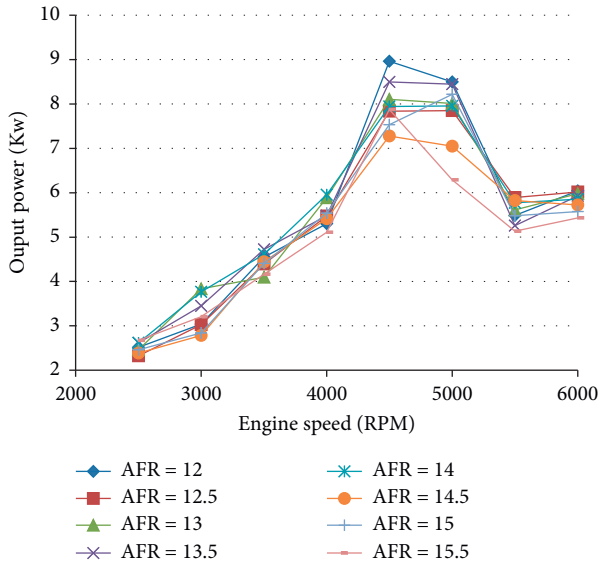


FIGURE 13: Output power at different engine speeds.

3.3. *Optimization Method.* Through the improved model, the optimization work is conducted. Several key performance parameters are selected as the multiple optimization objectives, such as output power (P_{out}), output speed (n), power efficiency (η), and oil consumption of rotation speed (γ). In this paper, firstly, we calculate the key characteristics by using the engine model while changing the input engine control parameters. Then, according to the users' requirement, we can artificially set the engine control parameters based on the engine working conditions. The basic principle of the optimization is as the following equations:

$$n(\alpha_1) < n(\alpha_2) < \dots < n(\alpha_n) (\alpha_1 < \alpha_2 < \dots < \alpha_n), \quad (10)$$

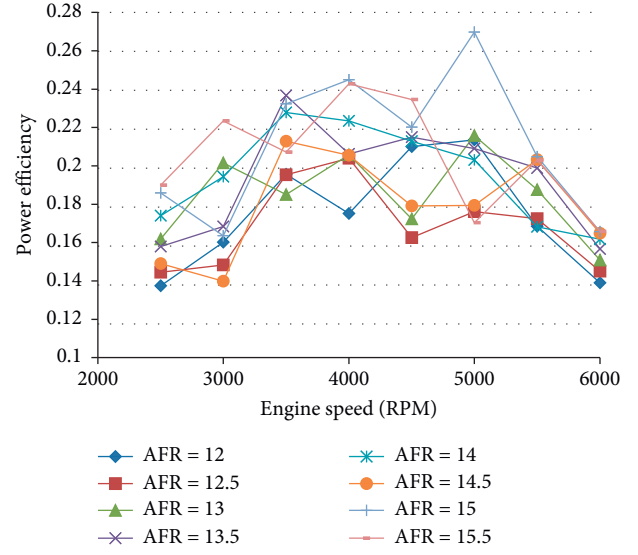


FIGURE 14: Power efficiency at different engine speeds.

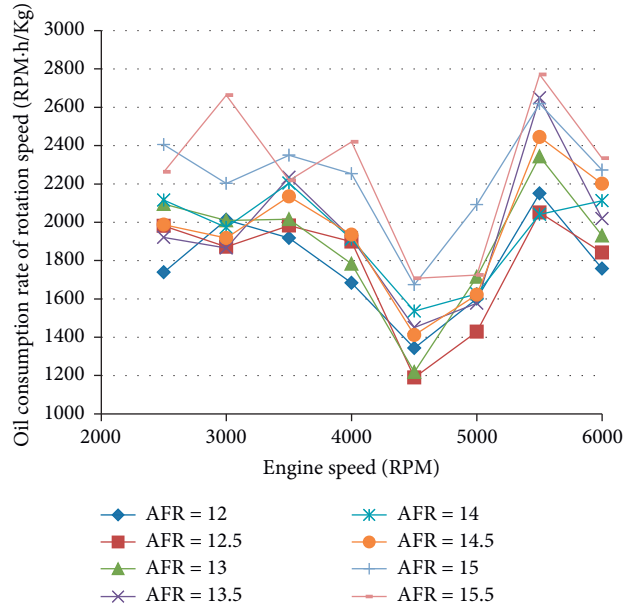


FIGURE 15: Oil consumption rate of rotation speed.

where $n(\alpha_i)$ ($i = 1, 2, \dots, n$) means engine speed with throttle degree of α_i . Equation (10) represents that the higher the engine speed is risen, the larger the throttle opening degree is. The evaluation system of the set fuel injection parameter relies on the developed evaluation function $f(\theta, n)$:

$$f(\theta, n) = \mathbf{W}_1(\theta, n)T_o + \mathbf{W}_2(\theta, n)P_o + \mathbf{W}_3(\theta, n)\eta + \mathbf{W}_4(\theta, n)\gamma, \quad (11)$$

where $\mathbf{W}_i(\theta, n)$ ($i = 1, 2, 3$, and 4) represents evaluation weight function of T_o , P_o , η , and γ , respectively. In this paper, the weights to be calculated can be expressed as matrixes \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{W}_3 , and \mathbf{W}_4 , and \mathbf{W}_i ($i = 1, 2, 3$ and 4) $\in R^{a \times b}$, where a and b are the numbers of different throttle opening degree values and engine speed values, respectively. Then, the

matrix of the evaluation function can be expressed as $F(\theta, n) \in R^{a \times b}$. Matrix A is the set AFR matrix. The basic principle of the optimization method is as shown in Figure 16.

The most important step is Step 2 which aims to obtain the weight matrixes. This paper applies a well-known multicriteria decision-making method named Analytic Hierarchy Process (AHP) to obtain evaluation weights for different groups of throttle opening degree and engine speed [20]. The above four alternatives are compared with each other based on self-defined Saaty scale, as shown in Table 5.

According to the experimental experience of the aero-engine characteristics in the application of aircraft flight, principles of setting the weights are as follows:

- (a) When the aeroengine starts, the speed gets to idle state, and the throttle degree is relative small, the output power of the engine should be firstly ensured, and oil consumption should be adequate in order to avoid engine speed suddenly dropping.
- (b) When the engine works from idle state to intermediate speed (approximate 4000 RPM), according to Figure 8, the load increase is not obvious. However, the engine noise is big which means the load efficiency is relatively low. The load efficiency is defined as load torque/output torque. At the same time, this process is general when the aircraft takes off and lands, and in order to prevent accidents, the output torque should be primarily guaranteed, and a certain amount of fuel consumption is to be sacrificed.
- (c) When the engine speed transits to the rated value, the fuel injection is always controlled based on the output AFR [21, 22]. However, in spite of the relevant regulation of the AFR control, the small aero-engine always leaves out the three-way catalytic unit so as to reduce the whole weight of the aircraft. Therefore, the stoichiometric AFR value is usually not the control target. When the aircraft regularly works, the fuel consumption is the first item to consider because load efficiency of this stage which is relatively high according to the experiment results [23, 24]. That is because the working noise is regular and varies uniformly along with the rising speed. It can be inferred that it is an uncommon occurrence of drop speed of aeroengine [25, 26]. As for the application in UAV, a certain speed is corresponding to a certain lift force, so the oil consumption of rotation speed (γ) should be firstly considered.
- (d) When the engine speed is over the rated value, the reason can be, firstly, there is an urgent external disturbance such as mutations in the air, and the flight attitude should be adjusted; secondly, UAV meets the obstacle while moving forward. At this moment, the consideration of output torque and power should be enhanced.

Through the AHP optimizing calculation, the calibration results of fuel injection parameter can be obtained, as shown in Table 6, and the fuel injection MAP is shown in Figure 17.

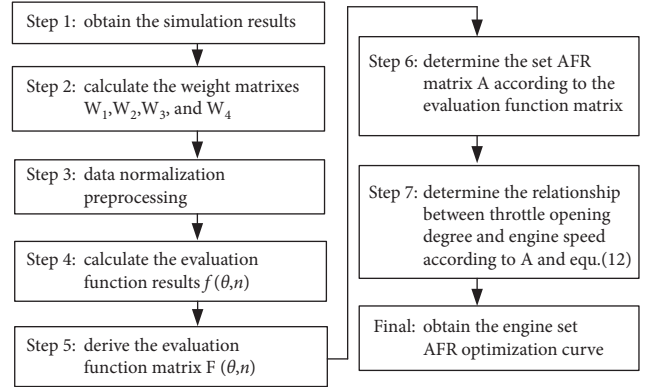


FIGURE 16: Optimization process.

TABLE 5: Definition and explanation of preference weights based on Saaty's theory.

Preference weights	Definition	Explanation
1	Equally preferable	Two factors contribute equally to the objective
3	Moderate preferred	Experience and judgement slightly favour one over other
5	Strongly preferred	Experience and judgement strongly favour one over the other
7	Very strongly preferred	Experience and judgement very strongly favour one over the other
9	Extremely preferred	The evidence favour one over the other is of the highest possible validity
2, 4, 6, 8	Intermediates values	Used to represent compromise between the preferences listed
Reciprocals	Reciprocals for inverse comparison	—

TABLE 6: Optimize calibration data of fuel injection parameter.

α ($^\circ$)/n (RPM)	2500	3000	3500	4000	4500	5000	5500	6000
10	15.5	15	15.5	15.5	15.5	15.5	15.5	15.5
20	15	14	15.5	15.5	15.5	15.5	15.5	15.5
30	14.5	12.5	15	15	15.5	15.5	15.5	15.5
40	15	15.5	15	15.5	15.5	15	15.5	15.5
50	15.5	15.5	12.5	15	15.5	15.5	13	15.5
60	15	15.5	14.5	15.5	15.5	15.5	15	14.5
70	15	15.5	15.5	15.5	15	13.5	13.5	14
80	13.5	15	15	15	14	15	14.5	15.5
90	15.5	15.5	15.5	15.5	14.5	15.5	13.5	13.5

From Figure 17, results of the control target of the output AFR of the two-stroke aeroengine of UAV can be summarized as follows:

- (1) Engine working conditions are corresponding to different control values in order to achieve the optimal optimization indicator.

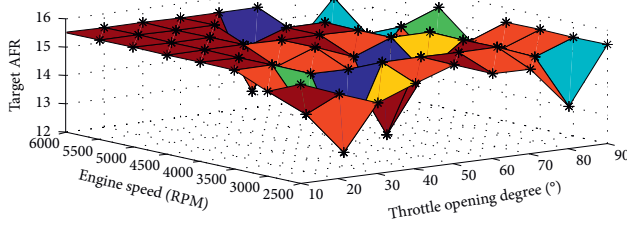


FIGURE 17: Fuel injection MAP.

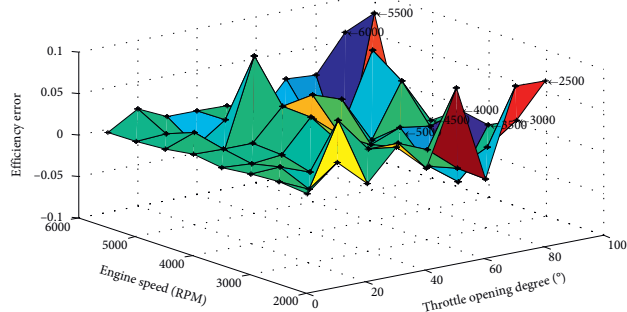


FIGURE 18: Efficiency error MAP.

- (2) Through the optimization method based on analytic hierarchy process, efficiency values under the whole working conditions can get a promotion, as shown in Figure 18. At the rated engine working condition (when the engine speed is at the range of 4500 to 6000 rpm), the improved efficiency is at the range of 5% to over 10%.

3.4. Optimization Results. Experiments of the aeroengine are designed so as to test the optimal results in the real application. In Figure 19, the analysis computer is connected to a data acquisition card which can gather real-time data from the AFR ratio and engine speed sensors. The oil consumption rate of rotation speed can be calculated according to the collected data including the speed and the fuel consumption within a certain period of time. According to the characteristics of the aeroengine, the power of the load is constant at a certain engine speed. Therefore, in the experiments, the throttle opening degree is stair-stepping settled, and the corresponding oil consumption is recorded.

From Figure 20, results of the aeroengine experiments are obtained through the comparison of the previous open-loop control when the injection width is constantly at 4.5 ms. When the engine speed changes from 3000 to 3500 RPM, the oil consumption rate of the optimal results is higher than that of the previous ones because at low-level speed the engine needs more fuel to guarantee the output power when the engine starts. When the aeroengine speed is higher than 4000 RPM, the oil consumption rate results of the optimal method are 10% to 27% higher than the original results.

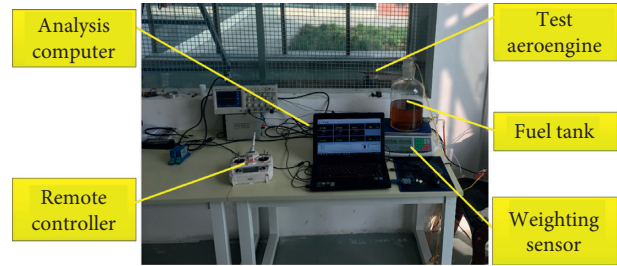


FIGURE 19: Aeroengine test station.

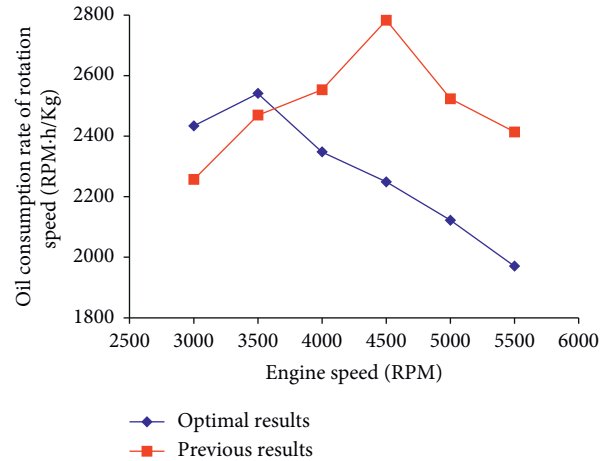


FIGURE 20: Comparison results of the optimal and previous oil consumption rate of the engine speed.

4. Conclusion

This paper proposes a method to optimize the fuel injection control system of two-stroke aeroengine of UAV based on one-dimensional fluid model and analytic hierarchy process. Key parameters of the one-dimensional model are calibrated and verified by calculation reasoning and experimentation. Expert experience is integrated into the rules of the analytic hierarchy calculation process. Through the optimization results from the experiments, it can be seen that when the engine speed changes from 3000 to 3500 RPM, the oil consumption rate of the optimal results is higher than that of the previous ones; when the aeroengine speed is higher than 4000 RPM, the oil consumption rate results of the optimal method are 10% to 27% higher than the original results. This method can be a reference for the efficiency optimization of the engine control system.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This work was supported by the Open Project Funding of Jiangsu Provincial Key Laboratory of Advanced Manufacturing and Process for Marine Mechanical Equipment.

References

- [1] C. Stöcker, R. Bennett, F. Nex, M. Gerke, and J. Zevenbergen, "Review of the current state of UAV regulations," *Remote Sensing*, vol. 9, no. 5, p. 459, 2017.
- [2] A. D. Sonparate, S. P. Gadpayle, and P. P. Bajpai, "Performance testing of 2-stroke SI engine by using external vaporized carburetor," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 8, pp. 1470–1478, 2015.
- [3] H. W. Gitano, R. Chim, and J. Loh, "The application of a resistive type O₂ sensor to a small engine EFI system," in *Proceedings of the SAE Technical Paper Series*, No. 2014-32-0073, Pisa, Italy, November 2014.
- [4] M. K. Balki, C. Sayin, and M. Canakci, "The effect of different alcohol fuels on the performance, emission and combustion characteristics of a gasoline engine," *Fuel*, vol. 115, pp. 901–906, 2014.
- [5] N. Kumar, "Performance evaluation and emission analysis of variable compression ratio direct injection diesel engine," *MATTER: International Journal of Science and Technology*, vol. 2, no. 2, pp. 32–47, 2016.
- [6] M. Venkatraman and G. Devaradjane, "Computer modeling of a CI engine for optimization of operating parameters such as compression ratio, injection timing and injection pressure for better performance and emission using diesel-diesel biodiesel blends," *American Journal of Applied Sciences*, vol. 8, no. 9, pp. 897–902, 2011.
- [7] M. Venkatraman and G. Devaradjane, "Simulation studies of a CI engine for better performance and emission using diesel-diesel biodiesel blends," *International Journal on Design and Manufacturing Technologies*, vol. 5, no. 2, pp. 14–21, 2011.
- [8] N. Miyamoto, T. Chikahisa, T. Murayama, and R. Sawyer, "Description and analysis of diesel engine rate of combustion and performance using Wiebe's functions," in *Proceedings of the SAE Technical Paper*, Detroit, MI, USA, No. 850107, Detroit, MI, USA, 1985.
- [9] T. Ganapathy, K. Murugesan, and R. P. Gakkhar, "Performance optimization of Jatropa biodiesel engine model using Taguchi approach," *Applied Energy*, vol. 86, no. 11, pp. 2476–2486, 2009.
- [10] L. P. Raut, "Computer simulation of CI engine for diesel and biodiesel blends," *International Journal of Innovative Technology and Exploring Engineering*, vol. 3, no. 2, pp. 2278–3075, 2013.
- [11] M. Kassa, C. Hall, A. Ickes, and T. Wallner, "Cylinder-to-cylinder variations in power production in a dual fuel internal combustion engine leveraging late intake valve closings," *SAE International Journal of Engines*, vol. 9, no. 2, pp. 1049–1058, 2016.
- [12] M. Rahimi-Gorji, M. Ghajar, A.-H. Kakaee, and D. Domiri Ganji, "Modeling of the air conditions effects on the power and fuel consumption of the SI engine using neural networks and regression," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 2, pp. 375–384, 2017.
- [13] L. O. F. Alves, M. G. D. dos Santos, A. B. Urquiza, J. H. Guerrero, J. C. de Lira, and V. Abramchuk, "Design of a new intake manifold of a single cylinder engine with three stages," in *Proceedings of the SAE Technical Paper*, No. 2017-36-0172, Sao Paulo, Brazil, November 2017.
- [14] S. Trajkovic, P. Tunestål, and B. Johansson, "Simulation of a pneumatic hybrid powertrain with VVT in GT-power and comparison with experimental data," in *Proceedings of the SAE Technical Paper*, No. 2009-01-1323, Detroit, MI, USA, 2009.
- [15] X. Yang and G. G. Zhu, "A mixed mean-value and crank-based model of a dual-stage turbocharged SI engine for hardware-in-the-loop simulation," in *Proceedings of the 2010 American Control Conference (ACC)*, pp. 3791–3796, IEEE, Baltimore, MD, USA, 2010.
- [16] B. Menacer and M. Bouchetara, "Parametric study of the performance of a turbocharged compression ignition engine," *Simulation*, vol. 90, no. 12, pp. 1375–1384, 2014.
- [17] C. Wei, M. Chen, and Y. Jiang, "Electronic control fuel injection system based on GT-POWER and MotoTron," *Procedia Engineering*, vol. 174, pp. 773–779, 2017.
- [18] X. Yang, C. Liao, and J. Liu, "Harmonic analysis and optimization of the intake system of a gasoline engine using GT-power," *Energy Procedia*, vol. 14, pp. 756–762, 2012.
- [19] J. H. Spurk, T. Betzel, and N. Simon, "Interaction of nonlinear dynamics and unsteady flow in fuel injectors," in *Proceedings of the SAE Technical Paper*, No. 920621, Detroit, MA, USA, 1992.
- [20] T. L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill International Book Co., Columbus, OH, USA, 1980.
- [21] Y. Wang, Y. Shi, M. Cai, W. Xu, and Q. Yu, "Optimization of air-fuel ratio control of fuel-powered UAV engine using adaptive fuzzy-PID," *Journal of the Franklin Institute*, vol. 355, no. 17, pp. 8554–8575, 2018.
- [22] Y. Wang, Y. Shi, M. Cai, W. Xu, and Q. Yu, "Efficiency optimized fuel supply strategy of aircraft engine based on air-fuel ratio control," *Chinese Journal of Aeronautics*, vol. 32, no. 2, pp. 489–498, 2018.
- [23] Z. Li, C.-Y. Su, G. Li, and H. Su, "Fuzzy approximation-based adaptive backstepping control of an exoskeleton for human upper limbs," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 555–566, 2014.
- [24] H. Su, C. Yang, G. Ferrigno, and E. De Momi, "Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
- [25] H. Yang, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2943–2949, 2020.
- [26] W. Qi and A. Aliverti, "A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity," *IEEE Journal of Biomedical and Health Informatics*, vol. 99, pp. 1–10, 2020.

Research Article

DOA Estimation without Source Number for Cyber-Physical Interactions

Jianzhong Li ¹, Xiaobo Gu ¹, Ruidian Zhan ², Xiaoming Xiong ¹ and Yuan Liu ¹

¹School of Automation, Guangdong University of Technology, Guangzhou 510000, China

²Chipeye Microelectronics Foshan Ltd., Foshan, China

Correspondence should be addressed to Yuan Liu; eliuyuan@gdut.edu.cn

Received 23 May 2020; Revised 7 June 2020; Accepted 18 June 2020; Published 8 July 2020

Guest Editor: Hang Su

Copyright © 2020 Jianzhong 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, a direction of arrival (DOA) estimator is proposed to improve the cyber-physical interactions, which is based on the second-order statistics without a priori knowledge of the source number. The impact of noise will firstly be eliminated. Then the relationship between the processed covariance matrix and the steering matrix is studied. By applying the elementary column transformation, an oblique projector will be designed without the source number. At last, a rooting method will be adopted to estimate the DOAs with the constructed projector. Simulation results show that the proposed method performs as well as other methods, which requires that the source number must be known.

1. Introduction

Nowadays, the requirement for different objects to communicate with each other is rapidly rising in many fields of the practical life. However, the network combining all the objects is very complicated, and the communications among different nodes faces many problems [1]. On one hand, energy diffusion of the transmitted signal would lower down the quality of communication. The energy collected by the target receiver is very poor in this situation. On the other hand, the currently adopted omnidirectional antennas also increase the risk of being attacked during the communication. Therefore, it is very important to omit the signal in the desired direction, and the technique of finding direction of arrival (DOA) can help to improve the performance of cyber-physical interactions.

DOA estimation has been an important research topic for array signal processing [2–4]. This research topic is widely applied in many fields such as sonar and electronic surveillance [5], where the signals are often nonstationary [6]. The wavefront of a far-field source signal can be assumed to be plane when it impinges on the receiver array. Each source can be localized with its corresponding DOA.

Plenty of researchers over the world have been making efforts to contribute to the research of DOA estimation, and

there are already many achievements, such as the multiple signal classification (MUSIC) in [7], estimating signal parameters via rotational invariance techniques (ESPRIT) in [8], root-MUSIC in [9], and oblique projection operator method (LOFNS) in [10]. However, all these existing methods require a priori knowledge of the source number to guarantee their successful application.

In this paper, we propose a method to localize far-field sources without any priori knowledge of the source number. Firstly, the impact of noise is eliminated by taking advantage of the property of nonstationary signal. Then the relationship between the steering matrix and covariance matrix is studied. The elementary column transformation is applied to get rid of the dependency of the source number and an orthogonal matrix is designed based on this relationship. At last, a rooting method is applied for the estimation of DOA, reducing the computational complexity.

The rest of this paper is organized as follows. Section 2 presents the signal model and some common assumptions. In Section 3, the proposed method is described in detail. The complexity analysis is also given to illustrate the improvement of the proposed method. In Section 4, several simulations are provided. At last, the conclusion of the whole paper is made in Section 5.

In this paper, T represents the transpose operation, H the conjugate transpose, and $*$ the complex conjugate. A bold capital letter symbolizes a matrix, and a bold letter in lowercase stands for a vector, such as \mathbf{A} and \mathbf{a} , respectively.

2. Signal Model

As shown in Figure 1, K narrow-band far-field source signals are received with a uniform linear array (ULA). $M + 1$ sensors are distributed in the array with intersensor spacing being d . The output of the m th ($m \in [0, M]$) sensor can be expressed as

$$y_m(t) = \sum_{k=1}^K s_k(t) e^{j\omega_k m} + n_m(t), \quad t = 1, 2, \dots, T, \quad (1)$$

$$\omega_k = \frac{2\pi d}{\lambda} \sin \theta_k,$$

where θ_k is the DOA of the k th source, λ is the wavelength of the signal satisfying $\lambda \geq 2d$, and $n_m(t)$ is the corresponding Gaussian white noise with the variance σ^2 . The noises are assumed to be independent from each other and from all the source signals. Written in matrix form, the received signal can be expressed as

$$\mathbf{y}(t) = \mathbf{A}(\theta)\mathbf{s}(t) + \mathbf{n}(t), \quad (2)$$

where

$$\begin{aligned} \mathbf{y}(t) &= [y_0(t), y_1(t), \dots, y_M(t)]^T, \\ \mathbf{s}(t) &= [s_1(t), s_2(t), \dots, s_K(t)]^T, \\ \mathbf{A}(\theta) &= [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_K)], \\ \mathbf{a}(\theta_k) &= [e^{j[0]\omega_k}, e^{j[1]\omega_k}, \dots, e^{j[M]\omega_k}]^T, \\ \mathbf{n}(t) &= [n_0(t), n_1(t), \dots, n_M(t)]^T. \end{aligned} \quad (3)$$

Without loss of generality, we make the following assumptions, which are the same as those in [10–17]:

- (1) The kurtosis of the source signal is nonzero
- (2) All the DOAs are different from each other
- (3) The Gaussian noise $n_m(t)$ is independent of the source signals, and the K source signals are independent of each other

3. Proposed Algorithm

Generally in order to estimate DOAs with an oblique projection, the source number should be known to divide the covariance matrix, like LOFNS in [10]. Indeed, after a deep analysis into the basic principle of the oblique projection based methods, we observed that the division with the source number is mainly to ensure that the desired matrix is full column rank, such that the inverse operation can work properly while constructing the oblique projection. Therefore, we propose a rooting method to localize sources with nonstationary signal that can get rid of the dependence of the source number.

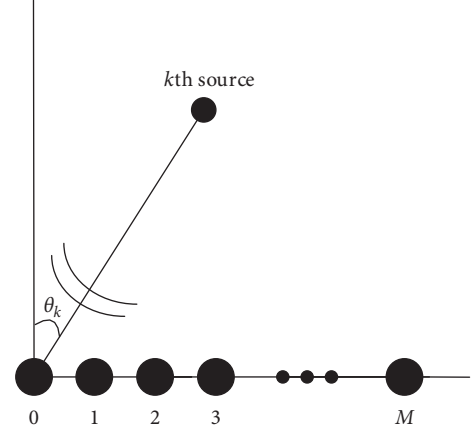


FIGURE 1: Far-field source localization with ULA.

When the signal is nonstationary and the noise is stationary, two covariance matrices can be obtained with two different group of snapshots as follows:

$$\begin{aligned} \mathbf{R}_1 &= \frac{2}{T} \sum_{t=1}^{T/2} [\mathbf{A}(\theta)\mathbf{y}(t)\mathbf{y}^H(t)\mathbf{A}^H(\theta) + \mathbf{n}(t)\mathbf{n}^H(t)] \\ &= \mathbf{A}(\theta)\mathbf{R}_{s1}\mathbf{A}^H(\theta) + \sigma^2\mathbf{I}, \end{aligned} \quad (4)$$

$$\begin{aligned} \mathbf{R}_2 &= \frac{2}{T} \sum_{t=(T/2)+1}^T [\mathbf{A}(\theta)\mathbf{y}(t)\mathbf{y}^H(t)\mathbf{A}^H(\theta) + \mathbf{n}(t)\mathbf{n}^H(t)] \\ &= \mathbf{A}(\theta)\mathbf{R}_{s2}\mathbf{A}^H(\theta) + \sigma^2\mathbf{I}, \end{aligned}$$

where \mathbf{R}_1 is estimated with the first $T/2$ snapshots and \mathbf{R}_2 with the last $T/2$ ones. The impact of the noise can be eliminated by constructing another matrix [18]:

$$\mathbf{R} = \mathbf{R}_1 - \mathbf{R}_2 = \mathbf{A}(\theta)\mathbf{R}_s\mathbf{A}^H(\theta), \quad (5)$$

where $\mathbf{R}_s = \mathbf{R}_{s1} - \mathbf{R}_{s2} \neq \mathbf{0}$.

From (5), it can be learned that all the columns of \mathbf{R} are the linear combination of the whole steering matrix. The relationship between \mathbf{R} and $\mathbf{A}(\theta)$ can still be maintained if we apply the elementary column transformation to \mathbf{R} :

$$\frac{r_{p(M+1-p)}}{r_{(p+q)(M+1-p)}} \mathbf{r}_{p+q} - \mathbf{r}_p \longrightarrow \mathbf{r}_{p+q}, \quad (6)$$

where $p \in [1, M]$, $q \in [1, M + 1 - p]$, \mathbf{r}_p is the p th column of \mathbf{R} , and r_{pq} is the pq th element. For every iteration, we fix the parameter p , and make q traverse all the legal values. The procedure will be repeated before we get a matrix which is upper triangular-like. All the nonzero columns we obtain after the transformation operation can form a maximal linearly independent subsystem of \mathbf{R} , and can form a full column-rank matrix \mathbf{U}_s . Construct an oblique projection as

$$\mathbf{U}_n = \mathbf{I}_{M+1} - \mathbf{U}_s(\mathbf{U}_s^H\mathbf{U}_s)^{-1}\mathbf{U}_s^H, \quad (7)$$

where \mathbf{I}_{M+1} is an identity matrix of $(M + 1) \times (M + 1)$. It can be easily calculated that

$$\mathbf{U}_s^H \mathbf{U}_n = \mathbf{0}_{M+1}. \quad (8)$$

Due to the fact that \mathbf{U}_s and $\mathbf{A}(\theta)$ span the same column space, (8) equals

$$\mathbf{A}^H(\theta) \mathbf{U}_n = \mathbf{0}_{M+1}. \quad (9)$$

Based on this orthogonality, a spectrum can be used for DOA estimation. In order to avoid the spectrum search which is computationally expensive, we propose the application of a rooting method. A polynomial is designed as follows:

$$f(z) = z^{2M+1} \mathbf{a}^T(z^{-1}) \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}(z), \quad (10)$$

where $z = e^{j\omega}$. The roots closest to the unit circle are the desired ones for the estimation of ω :

$$\hat{\omega}_k = \angle(z_k), \quad (11)$$

where $\angle(z)$ is to take the angle of z . The DOAs of the sources are estimated through

$$\hat{\theta}_k = \frac{\lambda \omega_k}{2\pi d}. \quad (12)$$

4. Simulation

In this section, the performance of the proposed method will be studied, which is examined with the root mean square error (RMSE). The definition of RMSE is given by

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^N |\hat{\theta}_n - \theta_{\text{true}}|^2}{N}}, \quad (13)$$

where $\hat{\theta}_n$ represents the DOA estimate of the n th simulation, θ_{true} is the real DOA and N means the total number of Monte Carlo simulations. The performance of the proposed method will be compared with other methods such as root-MUSIC in [9] and LOFNS in [10]. Two sources are considered in the simulations, whose DOAs are 6° and 23° , respectively. The inner space between sensors of the array $d = \lambda/2$. For the proposed method, the source number K can be unknown. For the other methods, K must be an exact priori knowledge. Specifically, the Cramer-Rao lower bound (CRLB) is also illustrated to make a better comparison.

The first simulation is designed to examine the relationship between the RMSEs and signal-noise ratio (SNR), which is defined as

$$\text{SNR} = 10 \log_{10} \frac{\sum_{k=1}^K P_{s_k}}{\sigma^2}, \quad (14)$$

with P_{s_k} being the signal power. Assume that there are 8 sensors in the array (i.e. $M = 7$) and 400 snapshots are received with the array. The SNR varies from 0 dB to 30 dB. As shown in Figures 2 and 3, by eliminating the effect of noise, both the proposed method and LOFNS outperform root-MUSIC, even though LOFNS does not perform well when the SNR is low. The proposed method shows a robust output for all the SNR. As for the estimates of different

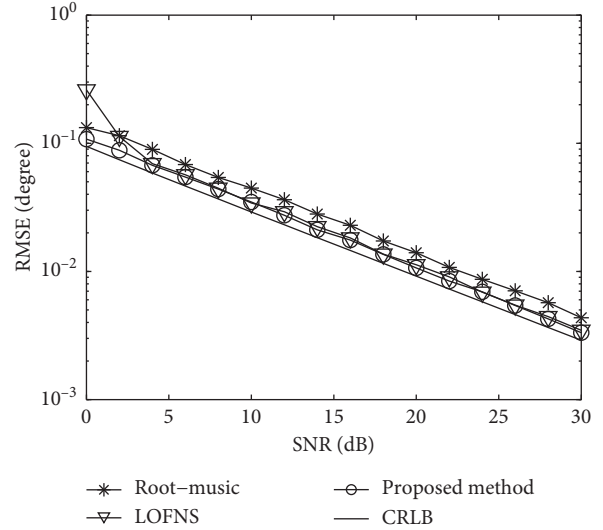


FIGURE 2: RMSEs versus SNR: 1st source.

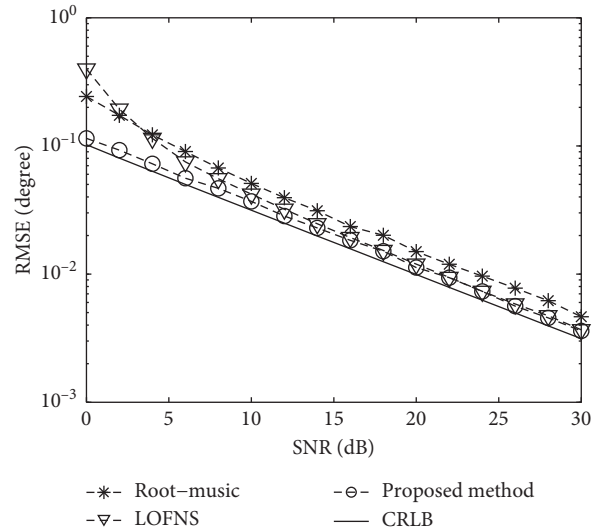


FIGURE 3: RMSEs versus SNR: 2nd source.

sources, we can see that the RMSEs with the same method are almost the same. The different directions of sources do not affect the estimation accuracy.

The second simulation studies the RMSEs in terms of the number of snapshots. Set the SNR at 15 dB, and the number of snapshots changes from 10 to 10000. The array is the same as that in the first simulation. The corresponding results are displayed in Figures 4 and 5. Similar to those RMSEs in Figures 2 and 3, the proposed method provides the most robust performance while root-MUSIC performs the poorest.

The computational efficiency will be studied in the third simulation. 1000 simulations are run in a PC, whose CPU is Intel (R) Core (TM) I7 8700 3.2 GHz and RAM is 8 GB, to get the total proceeding time for different methods. The results with 400 snapshots are shown in Table 1. We can see that the results of proceeding time of the three methods are almost

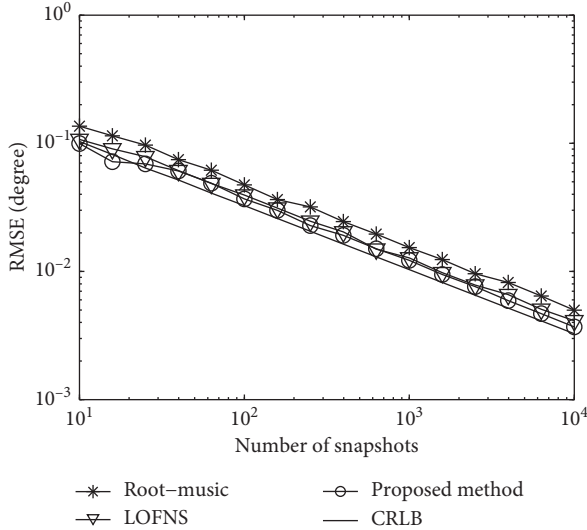


FIGURE 4: RMSEs versus number of snapshots.

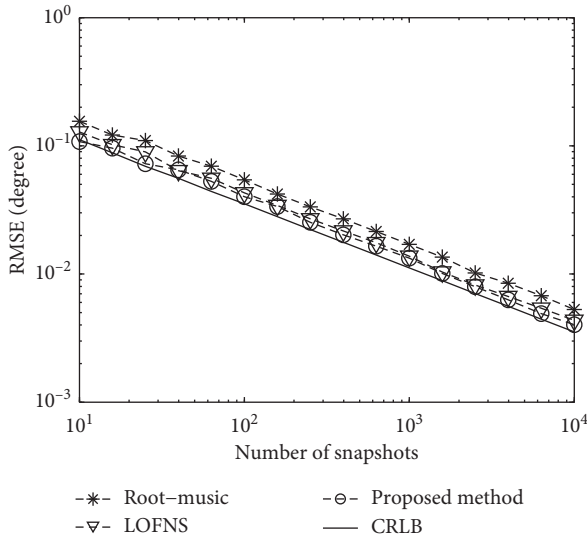


FIGURE 5: RMSEs versus number of snapshots.

TABLE 1: Total proceeding time with Matlab for 1000 times (in second).

Methods	Root-music	LOFNS	Proposed method
$M = 5$	0.3645	0.3545	0.4575
$M = 7$	0.5551	0.6222	0.6636
$M = 9$	0.6923	0.6929	0.7018
$M = 11$	0.7757	0.8646	0.7518
$M = 13$	1.1122	1.1589	1.1240
$M = 15$	1.7455	1.7939	1.7756

the same. As for the average proceeding time for one simulation, the difference is negligible.

5. Conclusion

In order to get rid of the dependency on the source number, a rooting method based on the oblique projection is

proposed in this paper. By taking advantage of the property of nonstationary signal, the effect of noise can be eliminated. The elementary column transformation is applied and a matrix orthogonal with the steering matrix is constructed without the number of sources. At last, a rooting method is applied with the constructed orthogonal matrix. Simulation results verify the effectiveness of the proposed method, which can perform better than other existing methods even with the known exact source number.

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 Key-Area Research and Development Program of Guangdong Province (2019B010145001, 2019B010142001, and 2019B010141002) and Science and Technology Program of Guangdong (2019A141401005).

References

- [1] V. Behzadan, "Cyber-physical attacks on uas networks-challenges and open research problems," 2017, <http://arxiv.org/abs/1702.01251>.
- [2] F.-G. Yan, S. Liu, J. Wang, M. Jin, and Y. Shen, "Fast doa estimation using co-prime array," *Electronics Letters*, vol. 54, no. 7, pp. 409-410, 2018.
- [3] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67-94, 1996.
- [4] Z. Zheng, J. Sun, W.-Q. Wang, and Y. Haifen, "Classification and localization of mixed near-field and far-field sources using mixed-order statistics," *Signal Processing*, vol. 143, pp. 134-139, 2018.
- [5] A. Liu, Q. Yang, X. Zhang, and W. Deng, "Modified root music for co-prime linear arrays," *Electronics Letters*, vol. 54, no. 15, pp. 949-951, 2018.
- [6] S. Chen, X. Dong, G. Xing, Z. Peng, W. Zhang, and G. Meng, "Separation of overlapped non-stationary signals by ridge path regrouping and intrinsic chirp component decomposition," *IEEE Sensors Journal*, vol. 17, no. 18, pp. 5994-6005, 2017.
- [7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276-280, 1986.
- [8] H. Chen, W. Zhu, and M. Swamy, "Real-valued esprit for two-dimensional doa estimation of noncircular signals for acoustic vector sensor array," in *Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015.
- [9] B. Friedlander, "The root-MUSIC algorithm for direction finding with interpolated arrays," *Signal Processing*, vol. 30, no. 1, pp. 15-29, 1993.
- [10] W. Zuo, J. Xin, N. Zheng, and A. Sano, "Subspace-based localization of far-field and near-field signals without

- eigendecomposition,” *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4461–4476, 2018.
- [11] A. Ahmed, Y. D. Zhang, and B. Himed, “Effective nested array design for fourth-order cumulant-based DOA estimation,” in *Proceedings of the Radar Conference*, pp. 0998–1002, Seattle, WA, USA, May 2017.
- [12] Y. Hu, Y. Liu, and X. Wang, “DOA estimation of coherent signals on coprime arrays exploiting fourth-order cumulants,” *Sensors*, vol. 17, no. 4, 2017.
- [13] Z. Zheng, M. Fu, W.-Q. Wang, and H. C. So, “Mixed far-field and near-field source localization based on subarray cross-cumulant,” *Signal Processing*, vol. 150, pp. 51–56, 2018.
- [14] R. Challa and S. Shamsunder, “High-order subspace-based algorithms for passive localization of near-field sources,” in *Proceedings of the Conference Record of the Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, November 1995.
- [15] J. M. Mendel, “Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications,” *Proceedings of the IEEE*, vol. 79, no. 3, pp. 278–305, 1991.
- [16] K. Wang, L. Wang, J. R. Shang, and X. X. Qu, “Mixed near-field and far-field source localization based on uniform linear array partition,” *IEEE Sensors Journal*, vol. 16, no. 22, pp. 8083–8090, 2016.
- [17] Z. Zheng, M. Fu, D. Jiang, W.-Q. Wang, and S. Zhang, “Localization of mixed far-field and near-field sources via cumulant matrix reconstruction,” *IEEE Sensors Journal*, vol. 18, no. 18, pp. 7671–7680, 2018.
- [18] A. Paulraj and T. Kailath, “Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 13–20, 1986.

Research Article

Output Feedback Adaptive Dynamic Surface Sliding-Mode Control for Quadrotor UAVs with Tracking Error Constraints

Guoqiang Zhu,^{1,2} Sen Wang,^{1,2} Lingfang Sun ,^{1,2} Weichun Ge,³ and Xiuyu Zhang ^{1,2}

¹School of Automation Engineering, Northeast Electric Power University, Jilin, China

²Jilin Province International Research Center of Precision Drive and Intelligent Control, Jilin, China

³State Grid Liaoning Electric Power Supply Co., Ltd., Shenyang, China

Correspondence should be addressed to Xiuyu Zhang; zhangxiuyu80@163.com

Received 11 April 2020; Accepted 22 May 2020; Published 15 June 2020

Guest Editor: Ning Wang

Copyright © 2020 Guoqiang Zhu 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, a fuzzy adaptive output feedback dynamic surface sliding-mode control scheme is presented for a class of quadrotor unmanned aerial vehicles (UAVs). The framework of the controller design process is divided into two stages: the attitude control process and the position control process. The main features of this work are (1) a nonlinear observer is employed to predict the motion velocities of the quadrotor UAV; therefore, only the position signals are needed for the position tracking controller design; (2) by using the minimum learning technology, there is only one parameter which needs to be updated online at each design step and the computational burden can be greatly reduced; (3) a performance function is introduced to transform the tracking error into a new variable which can make the tracking error of the system satisfy the prescribed performance indicators; (4) the sliding-mode surface is introduced in the process of the controller design, and the robustness of the system is improved. Stability analysis proved that all signals of the closed-loop system are uniformly ultimately bounded. The results of the hardware-in-the-loop simulation validate the effectiveness of the proposed control scheme.

1. Introduction

Quadrotor unmanned aerial vehicles (UAVs), as a new product in the field of small UAVs, have become a research hotspot among research and scholars all over the world [1–5]. The main advantages of quadrotor UAVs, such as flying in any direction, take off and land vertically, and hover at an ideal attitude, make the quadrotor UAVs widely used in more important fields, such as providing medical assistance, transporting special resources, disaster monitoring, and agricultural mapping. However, quadrotor UAVs are a complex physical system with the following characteristics, such as multivariate, nonlinearity, underactuating, and strong coupling, which make it very difficult to design an effective adaptive robust flight controller.

In the recent years, various effective control techniques have been developed for quadrotor UAVs to achieve stabilized or automatic flight, such as adaptive PID linear quadratic regulator (LQR) control [2, 6, 7], LMI-based linear control [8, 9], and H_∞ control [10, 11].

With the development of intelligence control theory, different kinds of advanced nonlinear control methods, which combine the linear control methods with intelligence control theory, such as feedback linearization control [12], model predictive control [13, 14], adaptive backstepping control [15, 16], adaptive sliding-mode control (SMC) [17–20], fault-tolerant control [21], dynamic surface control (DSC) [22–25], and adaptive fuzzy control [26, 27], have been proposed to achieve attitude and position trajectory tracking performance of quadrotor UAVs. In [28], a novel neural network-based output feedback controller is developed for a group of quadrotor UAVs. In [29], the prescribed performance backstepping dynamic surface control (DSC) scheme is proposed to solve the problem of trajectory tracking control for a quadrotor UAV with control input saturation. In [5], a fuzzy-based compound quantized control strategy is applied to the Quanser Qball-X4 quadrotor experimental platform, which achieved precise position control and tracking performance.

Among the above control schemes, the backstepping strategy has been widely used in the controller design for nonlinear systems. For instance, in [30], the trajectory tracking controller based on the backstepping approach was developed for the quadrotor model, while the PD control was used to attenuate the effects caused by system uncertainties. In [16], a nonlinear disturbance observer-based backstepping control method has been proposed to address the problem of loss of actuators' effectiveness. However, one drawback of backstepping is the "explosion of complexity" caused by the recurrent derivation of the virtual control law in each design step. To deal with this problem, the DSC control method has been proposed for a class of nonlinear systems, by introducing a first-order low-pass filter in each design step, and the shortcoming was overcome [22, 31–33].

An effective way to deal with the uncertainty of system parameters and unmodeled dynamics is to design an adaptive controller using the general approximation ability of the fuzzy logic system (FLS) and neural networks (NNs) [34–36]. The number of adaptive laws which depends on the fuzzy rules or the NN weights will be significantly increased as the number of fuzzy rules or the NN weights increase. To overcome this problem, a new method by estimating the norm rather than each item of the weight vector was proposed in [37, 38]. Thus, the number of adaptive laws is reduced significantly. Actually, the quadrotor UAVs are not only time-varying coupled and nonlinear systems, but also suffer from perturbations such as payload variations and nonlinear friction. Therefore, it is necessary to design a controller with adaptive capability, fast convergence, and robust performance for the quadrotor UAVs.

As a widely used nonlinear control algorithm, the sliding-mode control (SMC) is known for its excellent performance properties for complex high-order nonlinear systems in the presence of uncertain conditions [39, 40]. In [18], the SMC trajectory tracking controller was proposed for quadrotor UAVs by considering the wind perturbations and external disturbance components. In [19], a hierarchical control strategy based on the double-loop integral sliding-mode controller was designed for the position and attitude tracking of quadrotor UAVs with sustained disturbances and parameter uncertainties. Most of the existing literature focuses on using the SMC method to solve the attitude control of quadrotor UAVs instead of the position trajectory tracking control design because the transformed dynamic equation has a preferred form for the attitude control.

However, a major constraint in the controller design of quadrotor UAVs is that all the state variables of the system are required to be measurable. But in practical application, under some unpredictable factors, it will cause the measurement sensor to fail. [41–45]. In [41], an adaptive output feedback control scheme has been proposed for a class of uncertain SISO nonlinear systems under the constraint that only the system output can be obtained. In [43], a fuzzy state observer-based control method is designed for an uncertain MIMO nonlinear system, and by using the state observer, the problem of state immeasurability has been solved. Traditionally, the tracking performance in adaptive control schemes has been confined to ensure that the tracking error

converges to a residual set, the size of which is determined by the explicit design parameters and some unknown bounded terms. The upper bounds of the tracking error are difficult to calculate, so it is a very practical work to make the prior selection of the tracking performance satisfy certain steady state behavior. In [46, 47], a prescribed performance control scheme has been proposed for a class of nonlinear systems, and by constructing a prescribed performance function, the tracking error of the system was transformed into a new variable to ensure that the convergence rate was no less than a prespecified value, and the steady-state error remains within the prescribed range. However, limited attention has been paid to this issue for the controller design of quadrotor UAVs.

Inspired by the aforementioned discussions, an adaptive output feedback dynamic surface sliding-mode control for a class of quadrotor UAV system is presented where the fuzzy approximators are used to approximate the unknown items of the system. The main contributions of the proposed control scheme are as follows:

Firstly, to our best knowledge, this is the first time to use the dynamic surface control techniques with the sliding-mode method to design and test the robust controller of quadrotor UAVs in the platform of hardware-in-the-loop simulation, leading to a greatly simplified structure of the controller and improved robustness of the system.

Secondly, by introducing performance and error transformed functions in the controller, the tracking error of the quadrotor UAVs is transformed into a new error constraint variable which can ensure the prescribed transient performance of the system.

Thirdly, by estimating the norm of the FLS weights instead of estimating each variables of the weight vector, there is only one parameter needed to be updated at each step. Thus, the computing time is reduced.

Finally, the nonlinear state observer is introduced to predict the unmeasurable state of the quadrotor such as the angular velocity state of the quadrotor. Then, only the measurable attitude and position information are required in the implementation of the controller of the quadrotor.

The rest of this paper is organized as follows. In Section 2, problem statement and preliminaries including the mathematical model of the quadrotor, fuzzy logic systems (FLSs), and performance function are introduced. The process of the controller design and analysis of stability are given in Sections 3 and 4, respectively. Section 5 shows the results of the hardware-in-the-loop simulation to validate the effectiveness of the proposed control scheme.

2. Problem Statement and Preliminaries

2.1. Dynamic Model of Quadrotor UAVs. The schematic configuration of the quadrotor in this paper is shown in Figure 1. The basic movements are vertical movement, front and back movement, left and right movement, pitch rotation, roll rotation, and yaw rotation. On changing the rotor speed altogether with the same quantity, the lift forces will change, in this case, affecting the attitude of the vehicle. The complicated motions of a quadrotor can be divided into two

typical parts, and each part represents a subsystem with coupled terms. The first part is associated with the translational positions, and the second part is associated with the rotational angles. And in this section, we will deduce the mathematical model of a quadrotor UAV, including navigation equations and moment equations.

Define $\Lambda = [\phi, \theta, \psi]^T \in R^3$ and $w = [p, q, r]^T$, where ϕ, θ , and ψ represent the roll angle, pitch angle, and yaw angle with respect to the inertia frame and p, q , and r denote the angular velocity of the roll, pitch, and yaw with respect to the body-fixed frame. Let R_{BG} denote the transformation matrix between the inertia frame and the body-fixed frame using Euler–Lagrange formulation, which can be expressed as

$$R_{BG} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix}, \quad (1)$$

where $S_{(\cdot)}$ and $C_{(\cdot)}$ denote the $\sin(\cdot)$ and $\cos(\cdot)$, respectively.

Let $P = [x, y, z]^T \in R^3$ denote the position with respect to the inertia frame. According to Newton's second law, the relationship between combined force F_G and acceleration in the ground coordinate is

$$F_G = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \frac{d}{dt} (mV) = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \quad (2)$$

and we can get the translational dynamic equations of the quadrotor

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} F_G = \frac{1}{m} \left(R_{BG} \begin{bmatrix} 0 \\ 0 \\ U_z \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} \right), \quad (3)$$

where $D_x = d_x \dot{x}$, $D_y = d_y \dot{y}$, and $D_z = d_z \dot{z}$, in which d_x, d_y , and d_z are the air drag coefficients; $U_z = k(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$ is the lift force generated by the rotors with respect to the body coordinate system, in which $\Omega_i, i = 1, \dots, 4$ denote the rotary speed of the front, right, rear, and left rotors and k is the lift coefficient of the rotor.

According to the kinetic equation, the relationship between Λ and w can be described as

$$\dot{\Lambda} = Q(\Lambda)w = \begin{bmatrix} 1 & T_\theta S_\phi & T_\theta C_\phi \\ 0 & C_\phi & -S_\phi \\ 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad (4)$$

where $T_{(\cdot)}$ denotes $\tan(\cdot)$ and the transformation matrix $Q(\Lambda)$ is bounded according to $\|Q(\Lambda)\|_F < Q(\Lambda)_{\max}$ for a known constant $Q(\Lambda)_{\max}$ provided $-(\pi/2) < \phi < (\pi/2)$ and $-(\pi/2) < \theta < (\pi/2)$ [23]. M_{B0} is defined as the torque

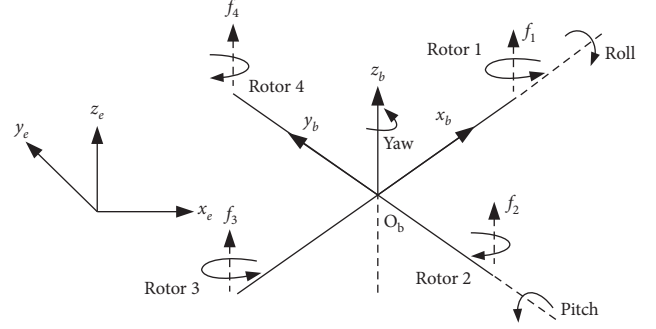


FIGURE 1: Schematic diagram of the quadrotor UAV.

provided by the rotors with respect to the body-fixed frame and is described as follows:

$$M_{B0} = \begin{bmatrix} M_{B0x} \\ M_{B0y} \\ M_{B0z} \end{bmatrix} = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ \tau_2 + \tau_4 - \tau_1 - \tau_3 \end{bmatrix}, \quad (5)$$

where $\tau_i (i = 1, \dots, 4) = k_\psi F_i$, k_ψ is a constant, l is the distance between a rotor and the center of mass of the quadrotor, $F_i (i = 1, \dots, 4) = k\Omega_i^2$ denotes the lift provided by the rotor, and we get $\tau_i = k_\psi k\Omega_i^2 = \tau\Omega_i^2$, in which τ represents the antitorque coefficient. Using the Newton–Euler equation, we can get the rotational dynamic equation of the quadrotor:

$$M_{B0} = (w \times J_B w) + J_B \dot{w} + M_r + M_d, \quad (6)$$

where $J_B = \text{diag}(J_{xx}, J_{yy}, J_{zz})$ is a symmetric positive definite constant matrix with J_{xx}, J_{yy} , and J_{zz} being the rotary inertia with respect to the $O_b x_b, O_b y_b$, and $O_b z_b$ axes, the signal \times represents the cross multiplication, and M_r and M_d are the resultant torques due to the gyroscopic effects and the resultant of the aerodynamic frictions torque. They are given as

$$M_r = \sum_{i=1}^4 w \times J_R [0, 0, (-1)^{i+1} \Omega_i]^T, \quad (7)$$

$$M_d = [d_\phi \dot{\phi}, d_\theta \dot{\theta}, d_\psi \dot{\psi}]^T,$$

where J_R represents the moment of inertia of each rotor and d_ϕ, d_θ , and d_ψ are the corresponding aerodynamic drag coefficients.

From (6), the following equation can be obtained:

$$\dot{w} = \frac{1}{J_B} [M_{B0} - M_r - M_d - w \times (J_B w)], \quad (8)$$

where

$$w \times J_B w = \begin{bmatrix} -r(J_{yy}q) + q(J_{zz}r) \\ r(J_{xx}p) - p(J_{zz}r) \\ -q(J_{xx}p) + p(J_{zz}q) \end{bmatrix}. \quad (9)$$

With the help of (4), the following equations can be obtained:

$$\begin{aligned}
\phi &= \frac{1}{J_{xx}} \left[M_{B0x} + \dot{\theta} \dot{\psi} (J_{yy} - J_{zz}) - J_R \dot{\theta} \Omega - d_\phi \dot{\phi} \right], \\
\theta &= \frac{1}{J_{yy}} \left[M_{B0y} + \dot{\phi} \dot{\psi} (J_{zz} - J_{xx}) + J_R \dot{\phi} \Omega - d_\theta \dot{\theta} \right], \\
\psi &= \frac{1}{J_{zz}} \left[M_{B0z} + \dot{\theta} \dot{\phi} (J_{xx} - J_{yy}) - d_\psi \dot{\psi} \right],
\end{aligned} \quad (10)$$

where $\Omega = \Omega_1 + \Omega_3 - \Omega_2 - \Omega_4$.

Remark 1. It is worth noting to point out that the roll, pitch, and yaw angles are limited to $(-\pi/2, \pi/2)$ which is physically meaningful.

By combing (3) and (10), a nonlinear equation of the quadrotor UAV is given as follows:

$$\dot{X} = f(X) + g(X)U, \quad (11)$$

where

$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ is the state vector and $f(X)$ and $g(X)$ are the smooth functions. The dynamic of quadrotor UAVs can be described as follows:

$$\begin{cases}
\dot{x}_1 = x_2, \\
\dot{x}_2 = (C_{x_7} S_{x_9} C_{x_{11}} + S_{x_7} S_{x_{11}}) U_1 - a_1 x_2, \\
\dot{x}_3 = x_4, \\
\dot{x}_4 = (C_{x_7} S_{x_9} S_{x_{11}} - S_{x_7} C_{x_{11}}) U_1 - a_2 x_4, \\
\dot{x}_5 = x_6, \\
\dot{x}_6 = (C_{x_7} C_{x_9}) U_1 - g - a_3 x_6, \\
\dot{x}_7 = x_8, \\
\dot{x}_8 = a_4 x_{10} x_{12} + a_5 x_{10} \Omega - a_6 x_8 + U_2, \\
\dot{x}_9 = x_{10}, \\
\dot{x}_{10} = a_7 x_8 x_{12} + a_8 x_8 \Omega - a_9 x_{10} + U_3, \\
\dot{x}_{11} = x_{12}, \\
\dot{x}_{12} = a_{10} x_8 x_{10} - a_{11} x_{12} + U_4,
\end{cases} \quad (12)$$

where g is the gravity acceleration and $U_i, i = 1, \dots, 4$ are the control inputs which can be expressed as

$$\begin{cases}
U_1 = \frac{k(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)}{m}, \\
U_2 = \frac{lk(\Omega_4^2 - \Omega_2^2)}{J_{xx}}, \\
U_3 = \frac{lk(\Omega_3^2 - \Omega_1^2)}{J_{yy}}, \\
U_4 = \frac{\tau(\Omega_4^2 + \Omega_2^2 - \Omega_1^2 - \Omega_3^2)}{J_{zz}}.
\end{cases} \quad (13)$$

Dividing the unknown parameters $a_i (i = 4, \dots, 11)$ into two parts, known part a_{iN} and unknown part Δa_i , it is expressed as follows:

$$\begin{aligned}
a_1 &= \frac{d_x}{m}, \\
a_2 &= \frac{d_y}{m}, \\
a_3 &= \frac{d_z}{m}, \\
a_{4N} &= \frac{J_{yy} - J_{zz}}{J_{xx}}, \\
a_{5N} &= \frac{J_R}{J_{xx}}, \\
a_{6N} &= \frac{d_\phi}{J_{xx}}, \\
a_{7N} &= \frac{J_{zz} - J_{xx}}{J_{yy}}, \\
a_{8N} &= \frac{J_R}{J_{yy}}, \\
a_{9N} &= \frac{d_\theta}{J_{yy}}, \\
a_{10N} &= \frac{J_{xx} - J_{yy}}{J_{zz}}, \\
a_{11N} &= \frac{d_\psi}{J_{zz}}.
\end{aligned} \quad (14)$$

2.2. Fuzzy Logic Systems (FLSs). The FLS is composed of three main components: fuzzy rule base, fuzzification, and defuzzification operators. The fuzzy rule base comprises a collection of fuzzy ‘‘IF-THEN’’ rules of the following form:

R^l : if x_1 is F_1^l and x_2 is F_2^l, \dots and x_n is F_n^l , then y is G^l , $l = 1, 2, \dots, N$,

where $x = [x_1, \dots, x_n]^T \in U$ and y are the FLS input and output, respectively, N is the number of rules, and fuzzy sets F_i^l and G^l are associated with the fuzzy membership functions $\mu_{F_i^l}(x_i)$ and $\mu_{G^l}(y)$. Through the singleton function, center average defuzzification, and product inference, the FLS can be expressed as

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

where $\bar{y}_l = \max_{y \in R} \mu_{G^l}(y)$. The fuzzy basis function is defined as

$$\xi_l = \frac{\prod_{i=1}^n \mu_{F_i}(x_i)}{\sum_{l=1}^N \left(\prod_{i=1}^n \mu_{F_i}(x_i) \right)}. \quad (16)$$

Denoting $W^T = [\bar{y}_1, \bar{y}_2, \dots, \bar{y}_N] = [W_1, W_2, \dots, W_N]$ and $\xi(x) = [\xi_1(x), \xi_2(x), \dots, \xi_N(x)]^T$, FLS (15) can be rewritten as

$$y(x) = W^T \xi(x). \quad (17)$$

Lemma 1. *According to [34], FLSs can effectively approximate any continuous nonlinear function with any small approximated error in a compact set. It can be expressed as*

$$\sup_{x \in \Omega} |F(x) - W^T \xi(x)| < \varepsilon, \quad (18)$$

where the continuous nonlinear function $F(x): \Omega \rightarrow \mathbb{R}$ with $\Omega \in \mathbb{R}^n$ is a compact set, $W^T \xi(x)$ is an FLS (17), and $\varepsilon > 0$ is the approximated error. Therefore, $F(x)$ can be described as

$$F(x) = W^{*T} \xi(x) + \sigma^*, \quad \forall x \in \Omega \subset \mathbb{R}^n, \quad (19)$$

where the minimum approximated error $|\sigma^*| \leq \varepsilon$ and W^* is an ideal weight vector and can be defined as

$$W^* = \arg \min_{W \subset \mathbb{R}^n} \left\{ \sup_{x \in \Omega} |F(x) - y(x)| < \varepsilon \right\}. \quad (20)$$

2.3. Performance and the Error Transformation Functions. Similar to [46], the mathematical expression of the prescribed tracking performance is given by

$$-\kappa_i p_i(t) < e_i(t) < \beta_i p_i(t), \quad (21)$$

where $e_i(t) = y_i - x_{id}$, $i = 1, \dots, 6$, are the tracking errors, the performance function $p_i(t)$ is defined as a smooth and decreasing positive function, and κ_i and β_i are the given positive constants. Moreover, $\kappa_i p_i(0)$ and $\beta_i p_i(0)$ represent the lower and upper bounds of the undershoot of $e_i(t)$ and $-\kappa_i p_i(\infty)$ and $\beta_i p_i(\infty)$ are the maximum allowable size of $e_i(t)$.

The error transformation function is chosen as

$$Y_i(S_i) = \frac{e_i(t)}{p_i(t)}, \quad (22)$$

where S_i is the transformed error variable and $Y_i(S_i)$ is a smooth strictly increasing function with the following properties:

$$\begin{aligned} \lim_{S_i \rightarrow -\infty} Y_i(S_i) &= -\kappa_i, \\ \lim_{S_i \rightarrow +\infty} Y_i(S_i) &= \beta_i. \end{aligned} \quad (23)$$

Note that if S_i is kept bounded, we have $-\kappa_i < Y_i(S_i) < \beta_i$, and thus (21) holds. The inverse transformation of $Y_i(S_i)$ can be expressed as

$$S_i = Y_i^{-1} \left(\frac{e_i(t)}{p_i(t)} \right) := \Theta_i \left(\frac{e_i(t)}{p_i(t)} \right), \quad (24)$$

where S_i , $e_i(t)$, and $p_i(t)$, are the transformed errors, the output tracking errors, and their corresponding performance functions.

In this paper, we choose

$$S_i = \Theta_i \left(\frac{e_i(t)}{p_i(t)} \right) = \ln \left(\frac{\kappa_i + e_i(t)/p_i(t)}{\beta_i - e_i(t)/p_i(t)} \right), \quad (25)$$

and differentiating (25) yields

$$\dot{S}_i = \eta_i \dot{y}_i - \eta_i v_i, \quad (26)$$

where $\eta_i = (\partial \Theta / \partial (e_i/p_i))(1/p_i)$ ($i = 1, \dots, 6$) and $v_i = \dot{x}_{id} + e_i \dot{p}_i/p_i$, where x_{id} are the reference signals and $e_i(t)$ are the output tracking errors. From the properties of the transformation, it is clear that η_i and v_i are bounded and $0 < \eta_{i0} \leq \eta_i$.

Remark 2. It can be seen that a new variable S_i is introduced through the above transformation process (21)–(25). If the designed control system can guarantee that S_i is bounded, then the tracking error e_i is bounded and meets formula (21). This means that the tracking error is always kept within the range $[-\kappa_i p_i(t), p_i(t)]$ or $[-p_i(t), \beta_i p_i(t)]$. The control objective is to design an adaptive controller so that S_i is bounded.

2.4. Nonlinear Observer. For a class of nonlinear systems with (A_0, C) in the observer canonical form is given by

$$\begin{cases} \dot{x} = A_0 x + \sum_{i=1}^n b_i f_i(x) + b_n u, \\ y = C^T x, \end{cases} \quad (27)$$

with

$$\begin{aligned} A_0 &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \\ b_i &= \begin{bmatrix} 0, \dots, 0, 1, 0, \dots, 0 \end{bmatrix}^T, \end{aligned} \quad (28)$$

with $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $u \in \mathbb{R}$, $b_i \in \mathbb{R}^n$ ($i \geq 2$), and $f(x)$ is the unknown smooth function. The vector b is general and not in a restricted form. Only the output y is assumed to be

measurable [48]. For the uncertain system (27), the nonlinear state observer is established as

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + \sum_{i=1}^n b_i \hat{f}_i(\hat{x}) + b_n u + Ky, \\ \hat{y} = C^T \hat{x}, \end{cases} \quad (29)$$

with

$$A = \begin{bmatrix} -k_1 & 1 & 0 & \cdots & 0 \\ -k_2 & 0 & 1 & \cdots & 0 \\ & & \vdots & & \\ -k_{n-1} & 0 & 0 & 1 & 0 \\ -k_n & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad (30)$$

$$K = \begin{bmatrix} -k_1 \\ -k_2 \\ \vdots \\ -k_{n-1} \\ -k_n \end{bmatrix},$$

where \hat{x} is the estimation of the state x and K is the observer gain vector; K is chosen so that the characteristic polynomial of $A - KC^T$ is strictly Hurwitz. Thus, for a given a matrix $Q_1 = Q_1^T > 0$, there exists a positive definite matrix P such that

$$A^T P + PA = -Q_1. \quad (31)$$

The function $\hat{f}(\hat{x})$ is the estimation of $f(x)$. In the next section, we choose FLSs to approximate $f(x)$. According to (27) and (29), the observer error can be expressed as

$$\dot{\tilde{x}} = A\tilde{x} + \sum_{i=1}^n b_i \tilde{W}_i^T \xi(\hat{x}_i) + \sum_{i=1}^n b_i \varepsilon_i = A\tilde{x} + \sum_{i=1}^n b_i \tilde{W}_i^T \xi(\hat{x}_i) + \bar{\varepsilon}, \quad (32)$$

where $\tilde{x} = x - \hat{x}$, $\tilde{W}_i = W_i^* - \hat{W}_i$, and $\bar{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_n]^T$.

3. The Process of the Controller Design

In this section, an adaptive FLS dynamic surface sliding-mode control scheme is proposed for position and attitude trajectory tracking control. The structure of the proposed control scheme is shown in Figure 2. The recursive design procedure contains two parts. Part 1 is the position trajectory tracking control and part 2 is the attitude trajectory tracking control. Each part contains three design steps, which are shown in Tables 1 and 2. The details of the controller design process are shown in Appendix A.

In Table 1, S_i , ($i = 1, \dots, 6$) are the surface errors and \bar{x}_{j_d} , ($j = 2, 4, 6$) are the virtual control signals in Step 1, Step 3, and Step 5, respectively; (T1.3), (T1.9), and (T1.15) represent the virtual control signal pass through a first-order filter to obtain a new variable x_{j_d} , ($j = 2, 4, 6$) with the time constant τ_j , ($j = 2, 4, 6$); (T1.6), (T1.12), and (T1.18) are the adaptive laws, and c_i , ($i = 1, \dots, 6$), λ_j, μ_j , ($j = 1, 2, 3$) are the design positive parameters. It should be noted that χ_i , ($i = 1, 2, 3$) are the virtual control given by

$\chi_1 = (C_{x_7} S_{x_9} C_{x_{11}} + S_{x_7} S_{x_{11}})U_1$, $\chi_2 = (C_{x_7} S_{x_9} S_{x_{11}} - S_{x_7} C_{x_{11}})U_1$, and $\chi_3 = (C_{x_7} C_{x_9})U_1$.

In Table 2, S_i , ($i = 7, \dots, 12$) are the surface errors and \bar{x}_{j_d} , ($j = 8, 10, 12$) are the virtual control signals in Step 7, Step 9, and Step 11, respectively; (T2.4), (T2.11), and (T2.18) represent the virtual control signal pass through a first-order filter to obtain a new variable x_{j_d} , ($j = 8, 10, 12$) with the time constant τ_j , ($j = 8, 10, 12$); (T2.7), (T2.14), and (T2.21) are the adaptive laws, and c_i , ($i = 7, \dots, 12$), λ_j, μ_j , ($j = 4, 5, 6$) are the design positive parameters; k_1 and k_2 are the observer gain.

It should be noted that $\hat{\theta}_i$, ($i = 1, \dots, 6$) are the estimations of θ_i with $\theta_i = \|W_i^*\|^2$, and $\|W_i^*\|^2$ and $\xi_i(X_i)$ are the ideal weight vector and fuzzy basis function vector of FLSs which are used to approximate the unknown continuous nonlinear function at each design step.

Remark 3. For the attitude trajectory tracking control system, χ_1, χ_2 , and χ_3 can be regarded as known and the input U_1 can be solved. The denominator of U_1 will not cause singularity since the yaw angle is limited to $(-\pi/2, \pi/2)$.

Remark 4. In the traditional sliding-mode control method, the existence of the signum function will cause chattering in the control system. In practical applications, the saturation function $\text{sat}(\cdot)$ [49] or the hyperbolic tangent function $\tanh(\cdot)$ [50] are generally used to eliminate the chatting phenomenon.

4. Stability and Prescribed Tracking Performance Analysis

First of all, define the filter error

$$y_i = x_{i_d} - \bar{x}_{i_d}, \quad i = 2, 4, 6, 7, 8, 9, 10, 12, \quad (33)$$

from (A.3), (A.13), (A.23), and (A.37). We have

$$\dot{x}_{i_d} = -\frac{y_i}{\tau_i}, \quad i = 2, 4, 6, 7, 8, 9, 10, 12. \quad (34)$$

Differentiating the boundary errors yields

$$\dot{y}_2 = \dot{x}_{2_d} - \dot{\bar{x}}_{2_d} = -\frac{y_2}{\tau_2} + k_1(\dot{x}_1 - \dot{\hat{x}}_1) - \dot{v}_1 + \frac{\dot{S}_1 c_1}{\eta_1} - \frac{S_1 c_1}{\eta_1^2}, \quad (35)$$

from which we have

$$\dot{y}_2 = -\frac{y_2}{\tau_2} + B_2(S_1, S_2, y_2, \ddot{x}_{1_d}). \quad (36)$$

By the same token, one has

$$\dot{y}_i = -\frac{y_i}{\tau_i} + B_i(\cdot), \quad (37)$$

where $B_i(\cdot)$, $i = 2, 4, 6, 7, 8, 9, 10, 12$, are the continuous functions. From (37), the following inequality holds:

$$y_i \dot{y}_i \leq -\frac{y_i^2}{\tau_i} + B_i |y_i|. \quad (38)$$

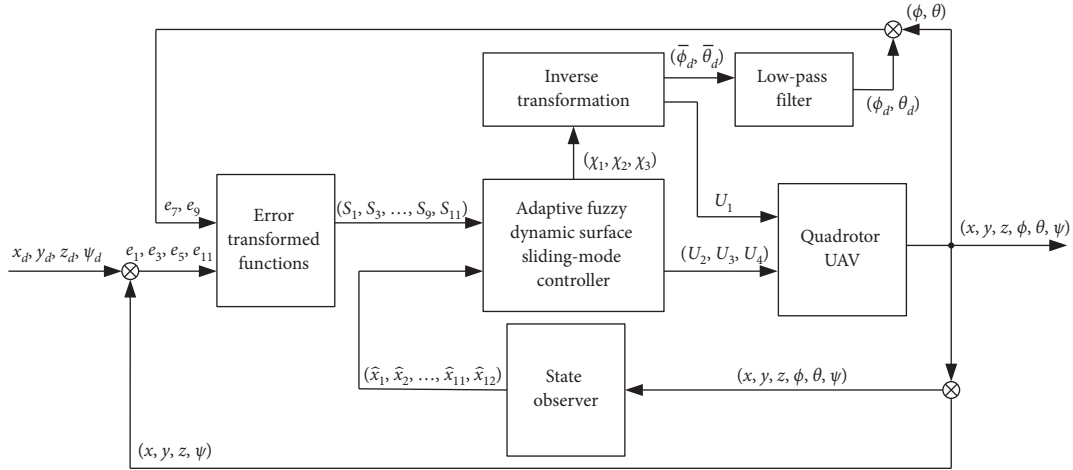


FIGURE 2: Structure of the proposed control scheme.

TABLE 1: The proposed DSCSM design for position trajectory control.

<i>Step 1</i>	
$S_1 = \Theta_1((e_1(t))/(p_1(t))),$	(T1.1)
$\hat{x}_1 = \hat{x}_2 + k_1(x_1 - \hat{x}_1), \hat{x}_2 = \hat{f}(\hat{x}_2) + \chi_1 + k_2(x_1 - \hat{x}_1),$	(T1.2)
$\bar{x}_{2d} = -k_1(x_1 - \hat{x}_1) + v_1 - S_1 c_1 / \eta_1, \tau_2 \dot{x}_{2d} + x_{2d} = \bar{x}_{2d}, x_{2d}(0) = \bar{x}_{2d}(0).$	(T1.3)
<i>Step 2</i>	
$S_2 = \hat{x}_2 - x_{2d},$	(T1.4)
$\chi_1 = -c_2 S_2 - (1/2)S_2 \hat{\theta}_1 \xi_1^T(X_1) \xi_1(X_1) - k_2(x_1 - \hat{x}_1) + \dot{x}_{2d} - \mu_1 \text{sgn}(S_2),$	(T1.5)
$\hat{\theta}_1 = (1/2)S_2^2 \xi_1^T(X_1) \xi_1(X_1) - \lambda_1 \hat{\theta}_1, \hat{\theta}_1(0) \geq 0.$	(T1.6)
<i>Step 3</i>	
$S_3 = \Theta_3((e_3(t))/(p_1(t))),$	(T1.7)
$\hat{x}_3 = \hat{x}_4 + k_1(x_3 - \hat{x}_3), \hat{x}_4 = \hat{f}(\hat{x}_4) + \chi_2 + k_2(x_3 - \hat{x}_3),$	(T1.8)
$\bar{x}_{4d} = -k_1(x_3 - \hat{x}_3) + v_2 - S_3 c_3 / \eta_2, \tau_4 \dot{x}_{4d} + x_{4d} = \bar{x}_{4d}, x_{4d}(0) = \bar{x}_{4d}(0).$	(T1.9)
<i>Step 4</i>	
$S_4 = \hat{x}_4 - x_{4d},$	(T1.10)
$\chi_2 = -c_4 S_4 - (1/2)S_4 \hat{\theta}_2 \xi_2^T(X_2) \xi_2(X_2) - k_2(x_3 - \hat{x}_3) + \dot{x}_{4d} - \mu_2 \text{sgn}(S_4),$	(T1.11)
$\hat{\theta}_2 = (1/2)S_4^2 \xi_2^T(X_2) \xi_2(X_2) - \lambda_2 \hat{\theta}_2, \hat{\theta}_2(0) \geq 0.$	(T1.12)
<i>Step 5</i>	
$S_5 = \Theta_5((e_5(t))/(p_1(t))),$	(T1.13)
$\hat{x}_5 = \hat{x}_6 + k_1(x_5 - \hat{x}_5), \hat{x}_6 = \hat{f}(\hat{x}_6) + \chi_3 + k_2(x_5 - \hat{x}_5),$	(T1.14)
$\bar{x}_{6d} = -k_1(x_5 - \hat{x}_5) + v_3 - S_5 c_5 / \eta_3, \tau_6 \dot{x}_{6d} + x_{6d} = \bar{x}_{6d}, x_{6d}(0) = \bar{x}_{6d}(0).$	(T1.15)
<i>Step 6</i>	
$S_6 = \hat{x}_6 - x_{6d},$	(T1.16)
$\chi_3 = -c_6 S_6 - (1/2)S_6 \hat{\theta}_3 \xi_3^T(X_3) \xi_3(X_3) - k_2(x_5 - \hat{x}_5) + \dot{x}_{6d} - \mu_3 \text{sgn}(S_6),$	(T1.17)
$\hat{\theta}_3 = (1/2)S_6^2 \xi_3^T(X_3) \xi_3(X_3) - \lambda_3 \hat{\theta}_3, \hat{\theta}_3(0) \geq 0.$	(T1.18)

Consider the Lyapunov Function candidate as

$$\Gamma = \frac{1}{2} \tilde{x}^T P \tilde{x} + \sum_{i=1}^6 \Gamma_i + \sum_{i=1}^6 \frac{1}{2} y_{2i}^2 + \frac{1}{2} y_7^2 + \frac{1}{2} y_9^2, \quad (39)$$

where $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{12}]^T$ and Γ_i are defined in (A.6)–(A.45).

Theorem 1. For the quadrotor system (12) and the state observer (29), the adaptive laws (T1.6), (T1.12), (T1.18), (T2.7), (T2.14), and (T2.21) and the control input (T2.6), (T2.13), and (T2.20) are given in Table 1 and Table 2; if $\Gamma(0)$ satisfies $\Gamma(0) \leq P$, ($P \geq 0$), then, by properly selecting the design parameters c_i , ($i = 1, \dots, 12$), λ_j, μ_j , ($j = 1, \dots, 6$) and the time constant of the low-pass filter

$\tau_2, \tau_4, \tau_6, \tau_7, \tau_8, \tau_9, \tau_{10}$, and τ_{12} appropriately, all the signals in the closed-loop system are semiglobal uniformly bounded. In addition, the tracking error of position and attitude angle can converge to an arbitrarily residual set and is always kept in the prespecified cures.

Proof. Please see Appendix B for details. \square

5. Hardware in the Loop Simulation Results

In this paper, the hardware-in-loop testing platform is used to verify the effectiveness of the proposed control scheme. The experiment environment and the experimental system architecture are shown in Figures 3 and 4, where the following components are included:

TABLE 2: The proposed DSCSM design for attitude trajectory control.

<i>Step 7</i>	
$S_7 = \Theta_7((e_7(t))/(p_\psi(t))),$	(T2.1)
$\hat{x}_7 = \hat{x}_8 + k_1(x_7 - \hat{x}_7),$	(T2.2)
$\hat{x}_8 = a_{4N}\hat{x}_{10}\hat{x}_{12} + a_{5N}\hat{x}_{10}\Omega - a_{6N}\hat{x}_8 + \hat{f}(\hat{x}_8, \hat{x}_{10}, \hat{x}_{12}) + U_2 + k_2(x_7 - \hat{x}_7),$	(T2.3)
$\bar{x}_{8d} = -k_1(x_7 - \hat{x}_7) + v_4 - S_7c_7/\eta_4, \tau_8\dot{x}_{8d} + x_{8d} = \bar{x}_{8d}, x_{8d}(0) = \bar{x}_{8d}(0).$	(T2.4)
<i>Step 8</i>	
$S_8 = \hat{x}_8 - x_{8d},$	(T2.5)
$U_2 = -c_8S_8 - (1/2)S_8\hat{\theta}_4\xi_4^T(X_4)\xi_4(X_4) - k_2(x_7 - \hat{x}_7) - a_{4N}\hat{x}_{10}\hat{x}_{12} - a_{5N}\hat{x}_{10}\Omega + a_{6N}\hat{x}_8 + \dot{x}_{8d} - \mu_4\text{sgn}(S_8),$	(T2.6)
$\hat{\theta}_4 = (1/2)S_8^2\xi_4^T(X_4)\xi_4(X_4) - \lambda_4\hat{\theta}_4, \hat{\theta}_4(0) \geq 0.$	(T2.7)
<i>Step 9</i>	
$S_9 = \Theta_9((e_9(t))/(p_\psi(t))),$	(T2.8)
$\hat{x}_9 = \hat{x}_{10} + k_1(x_9 - \hat{x}_9),$	(T2.9)
$\hat{x}_{10} = a_{7N}\hat{x}_8\hat{x}_{12} + a_{8N}\hat{x}_8\Omega - a_{9N}\hat{x}_{10} + \hat{f}(\hat{x}_8, \hat{x}_{10}, \hat{x}_{12}) + U_3 + k_2(x_9 - \hat{x}_9),$	(T2.10)
$\bar{x}_{10d} = -k_1(x_9 - \hat{x}_9) + v_5 - S_9c_9/\eta_5, \tau_{10}\dot{x}_{10d} + x_{10d} = \bar{x}_{10d}, x_{10d}(0) = \bar{x}_{10d}(0).$	(T2.11)
<i>Step 10</i>	
$S_{10} = \hat{x}_{10} - x_{10d},$	(T2.12)
$U_3 = -c_{10}S_{10} - (1/2)S_{10}\hat{\theta}_5\xi_5^T(X_5)\xi_5(X_5) - k_2(x_9 - \hat{x}_9) - a_{7N}\hat{x}_8\hat{x}_{12} - a_{8N}\hat{x}_8\Omega + a_{9N}\hat{x}_{10} + \dot{x}_{10d} - \mu_5\text{sgn}(S_{10}),$	(T2.13)
$\hat{\theta}_5 = (1/2)S_{10}^2\xi_5^T(X_5)\xi_5(X_5) - \lambda_5\hat{\theta}_5, \hat{\theta}_5(0) \geq 0.$	(T2.14)
<i>Step 11</i>	
$S_{11} = \Theta_{11}((e_{11}(t))/(p_\psi(t))),$	(T2.15)
$\hat{x}_{11} = \hat{x}_{12} + k_1(x_{11} - \hat{x}_{11}),$	(T2.16)
$\hat{x}_{12} = a_{10N}\hat{x}_8\hat{x}_{10} - a_{11N}\hat{x}_{12} + \hat{f}(\hat{x}_8, \hat{x}_{10}, \hat{x}_{12}) + U_4 + k_2(x_{11} - \hat{x}_{11}),$	(T2.17)
$\bar{x}_{12d} = -k_1(x_{11} - \hat{x}_{11}) + v_6 - S_{11}c_{11}/\eta_6, \tau_{12}\dot{x}_{12d} + x_{12d} = \bar{x}_{12d}, x_{12d}(0) = \bar{x}_{12d}(0).$	(T2.18)
<i>Step 12</i>	
$S_{12} = \hat{x}_{12} - x_{12d},$	(T2.19)
$U_4 = -c_{12}S_{12} - (1/2)S_{12}\hat{\theta}_6\xi_6^T(X_6)\xi_6(X_6) - k_2(x_{11} - \hat{x}_{11}) - a_{10N}\hat{x}_8\hat{x}_{10} + a_{11N}\hat{x}_{12} + \dot{x}_{12d} - \mu_6\text{sgn}(S_{12}),$	(T2.20)
$\hat{\theta}_6 = (1/2)S_{12}^2\xi_6^T(X_6)\xi_6(X_6) - \lambda_6\hat{\theta}_6, \hat{\theta}_6(0) \geq 0.$	(T2.21)

- (i) NI PXIe-1082, the MT real-time simulator (RTS) with Kintex-7 325T FPGA chip and 16bits synchronized analog I/O with a data transfer rate of 1 MS/s. The simulator supports FPGA simulation for the quadrotor UAV system. The device accepts the control signal and calculates the response of the system in real time and outputs to the controller box.
- (ii) NI PXIe-1071, the MT Rapid Control Prototype (RCP), with Kintex-7 325T FPGA @Xilinx and 16 analog I/O channels with a transmission rate of 1 MS/s. This device is used to realize the real-time running of the control code and send the control signals to the quadrotor UAV simulation model which is running on the MT real-time simulator.
- (iii) Adapter plate: it is used to realize the signal connection between the RTS and RCP. The RTS, RCP, and the signal adapter board comprise a closed loop experimental system.
- (iv) Host computer: uses the StarSim RCP software to download the Matlab/simulink quadrotor UAV system model and the control algorithm into the RTS and RCP, respectively.

In this section, the effectiveness and the performance of the proposed adaptive dynamic surface sliding-mode output feedback control method are showed by the following experiments. Different scenarios are considered, including normal case and model uncertainty cases to demonstrate the



FIGURE 3: Actual experimental environment.

robustness of the proposed controller. The parameters of the quadrotor UAV adopted in this study are described in Table 3 [17]. In the experiment, the desired trajectory of the position and yaw angle desired trajectory $\{x_d(t), y_d(t), z_d(t), \psi_d(t)\}$ is chosen as $\{\cos(t), \sin(t), 0.5(t), \sin(0.5t)\}$. The performance functions are selected as $p_1(t) = (p_{10} - p_{1\infty})e^{-lt} + p_{1\infty}$, with the parameters $p_{10} = 1.5, p_{1\infty} = 0.055, l = 1$, and $\kappa_1 = \beta_1 = 1$. $p_\psi(t) = 0.6 * e^{-t} + 0.04$. The controller parameters chosen for simulation are $c_1 = c_2 = c_3 = c_4 = 0.6, c_5 = c_6 = 0.55, c_7 = c_8 = c_9 = c_{10} = 0.6, c_{11} = c_{12} = 0.6, \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0.02$, and $\tau_i = 0.005, (i = 2, 4, 6, 7, 8, 9, 10, 12)$. To demonstrate the effectiveness of the proposed controller, the following different cases are considered and comparisons are conducted:

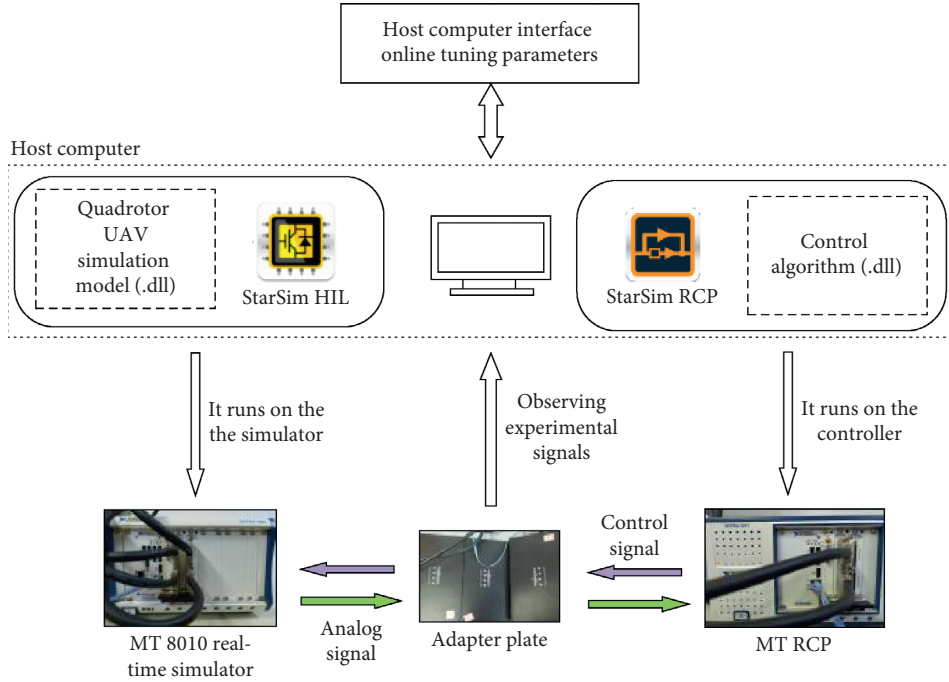


FIGURE 4: The experimental system architecture.

TABLE 3: Quadrotor parameters.

Symbol	Case 1	Case 2	Case 3	Case 4	Units
m	1.4	1.4	1.4	1.4	kg
k	2.98	2.98	2.98	2.98	$10^{-6} \text{ N} \cdot \text{s}^2 \cdot \text{rad}^{-2}$
l	0.2	0.2	0.2	0.2	m
τ	1.14	1.14	1.14	1.14	$10^{-7} \text{ N} \cdot \text{s}^2 \cdot \text{rad}^{-2}$
d_ϕ, d_θ, d_ψ	1.2	1.2	1.2	1.2	$10^{-2} \text{ N} \cdot \text{s} \cdot \text{rad}^{-1}$
J_R	5	5	5	5	$10^{-5} \text{ N} \cdot \text{s}^2 \cdot \text{rad}^{-1}$
J_{xx}, J_{yy}	0.03	0.03	0.03	0.03	$\text{N} \cdot \text{s}^2 \cdot \text{rad}^{-1}$
J_{zz}	0.04	0.046	0.052	0.06	$\text{N} \cdot \text{s}^2 \cdot \text{rad}^{-1}$

Case 1: normal case: we assume that there are no uncertainties in the model, and all parameters of the quadrotor are normal. The initial state vector is set to be $x(0) = [0.02, 0, 0.02, 0, 0, 0, 0, 0, 0, 0, 0.1, 0]^T$. And, the fuzzy membership function is adopted as $\mu_{F_k^l} = \exp(-((\hat{x}_k - 6 + 2l)^2)/2)$, where $l = 1, \dots, 5$ and $k = 2, 6, 8, 10, 12$.

Cases 2, 3, and 4: uncertainty (15%, 30%, and 50% added) in rotary inertia: in these cases, we consider three different model uncertainties 15%, 30%, and 50% separately added in the yaw axis.

The experimental results are shown in Figures 5–15. Figures 5–9 illustrate the comparison experimental results of the tracking trajectory in Case1 between the proposed control method and the traditional PID control method. From Figures 6–9, it can be seen clearly that all the tracking errors of the position and yaw angles of the proposed control scheme are always kept within the performance function curves. That is, the control method proposed in this paper obtains much better control performance by comparing with the traditional PID control scheme. Figure 10 shows the

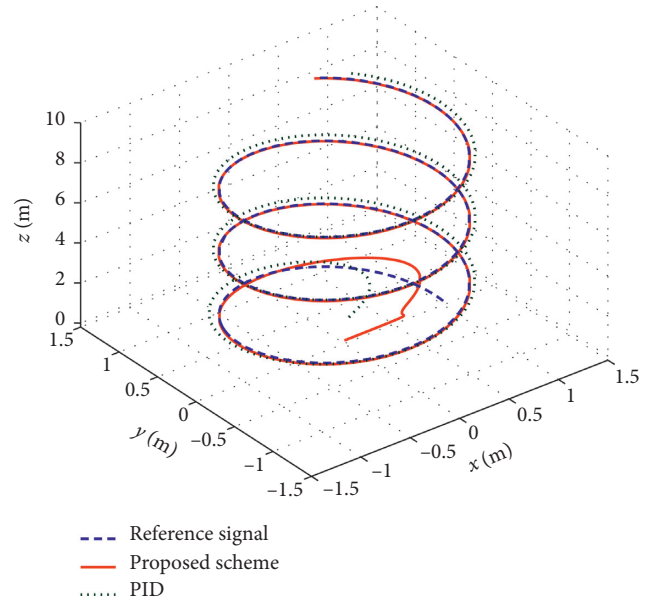
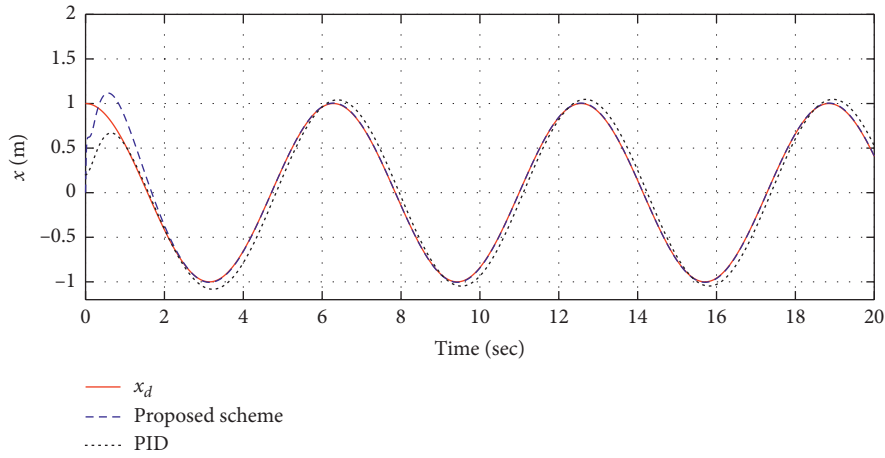
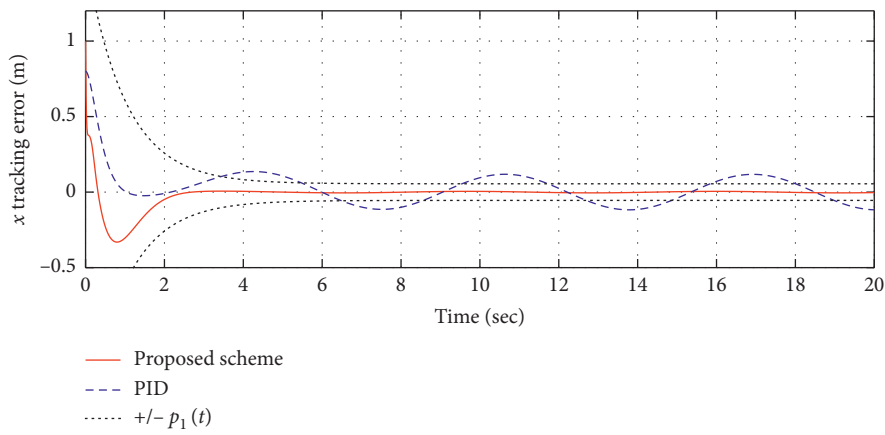


FIGURE 5: Space diagram of position in the normal case.

control signals. Figure 11 shows the response curves of roll and pitch angles. Figures 12 and 13 show the change of six adaptive parameters. Figures 14 and 15 show the trajectories of x_i and \hat{x}_i , ($i = 2, 4, 6, 8, 10, 12$). We can see that the proposed state observer can quickly approximate the output of the system. Figure 16 shows the 3D tracking trajectory with uncertainties 15%, 30%, and 50% added in the yaw rotating axes. Figure 17 illustrates the results of the tracking error under cases 2, 3, and 4. Also, the maximum value of the tracking error (MVTE) and the root mean square value of the tracking error (RMSVTE) in the steady state ($t > 5$ s) are

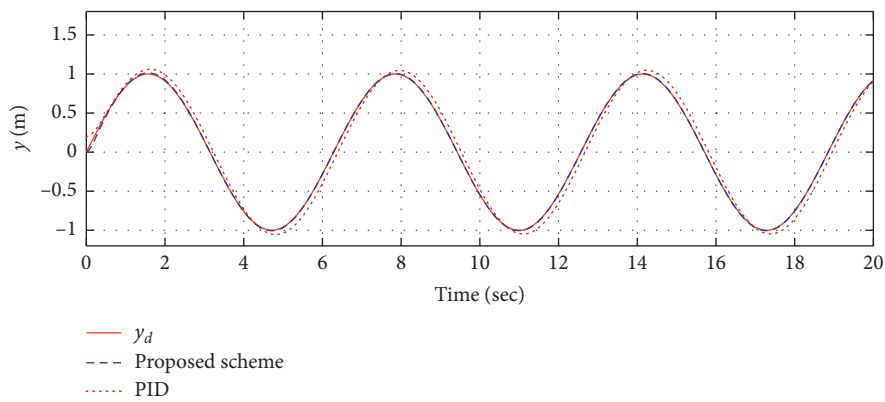


(a)



(b)

FIGURE 6: The tracking trajectory of x_d , x , and the tracking error.



(a)

FIGURE 7: Continued.

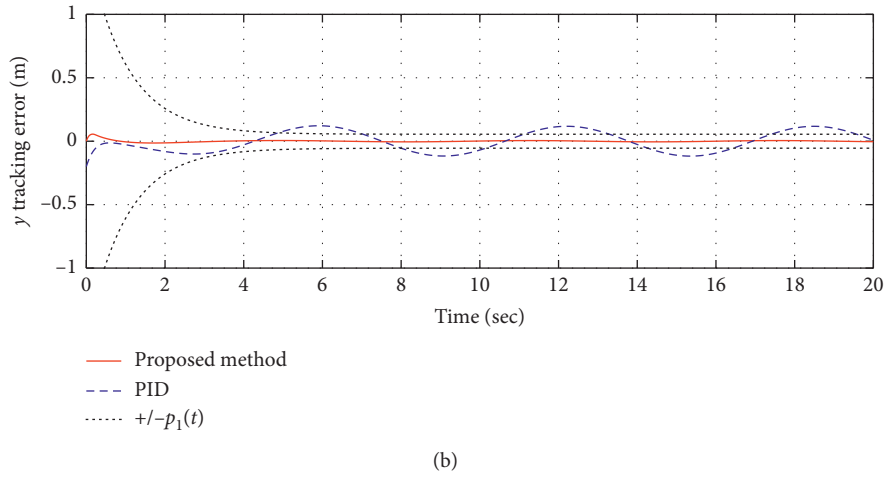


FIGURE 7: The tracking trajectory of y_d , y , and the tracking error.

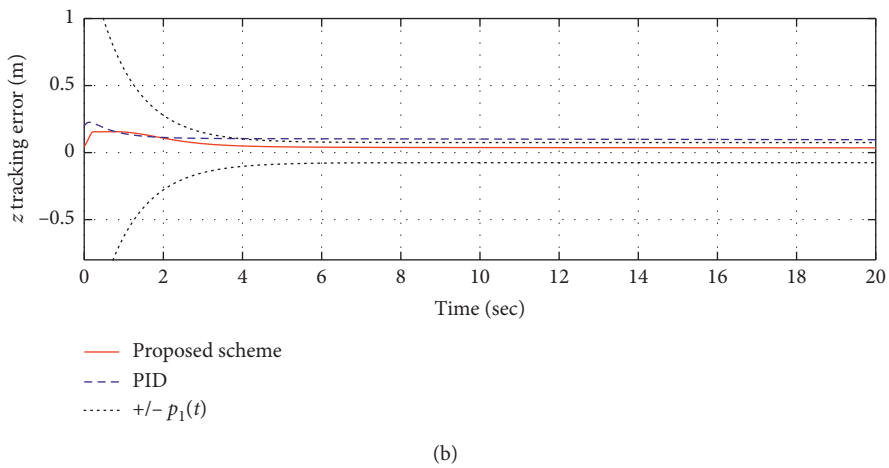
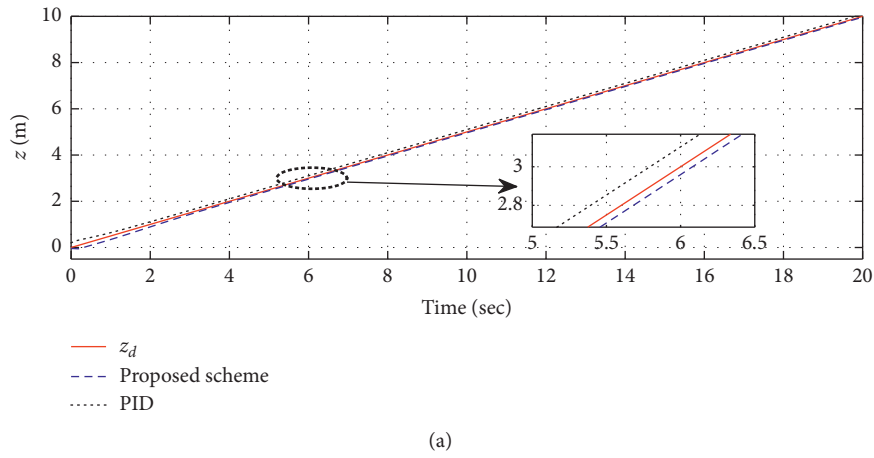
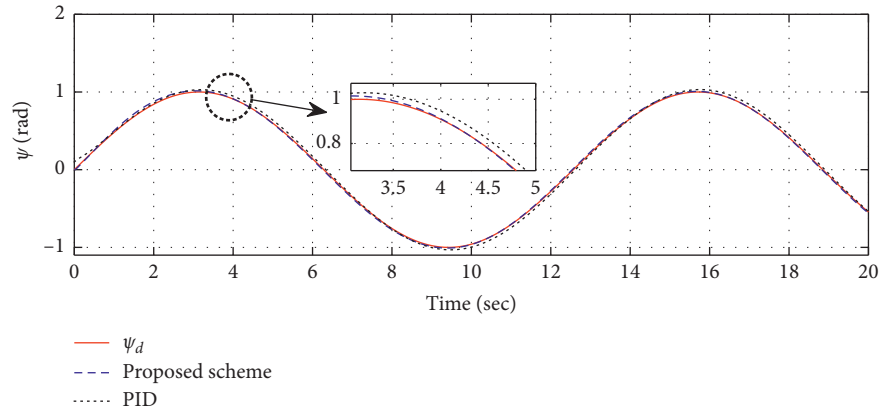
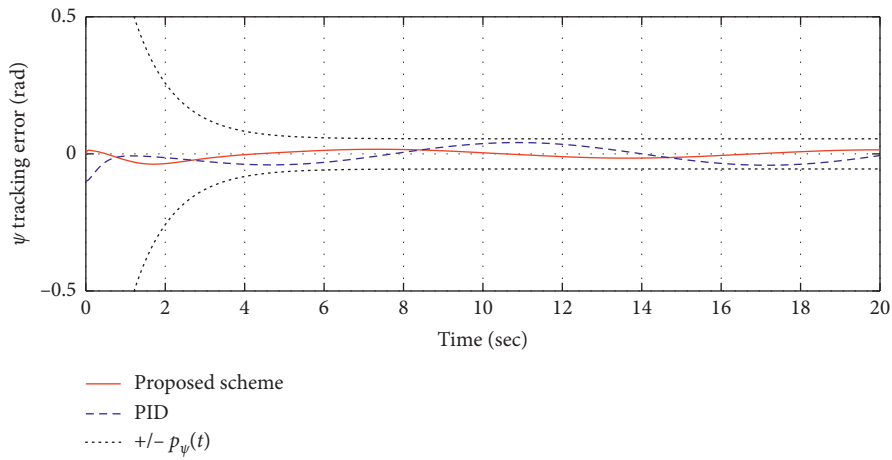


FIGURE 8: The tracking trajectory of z_d , z , and the tracking error.



(a)



(b)

FIGURE 9: The tracking trajectory of ψ_d , ψ , and the tracking error.

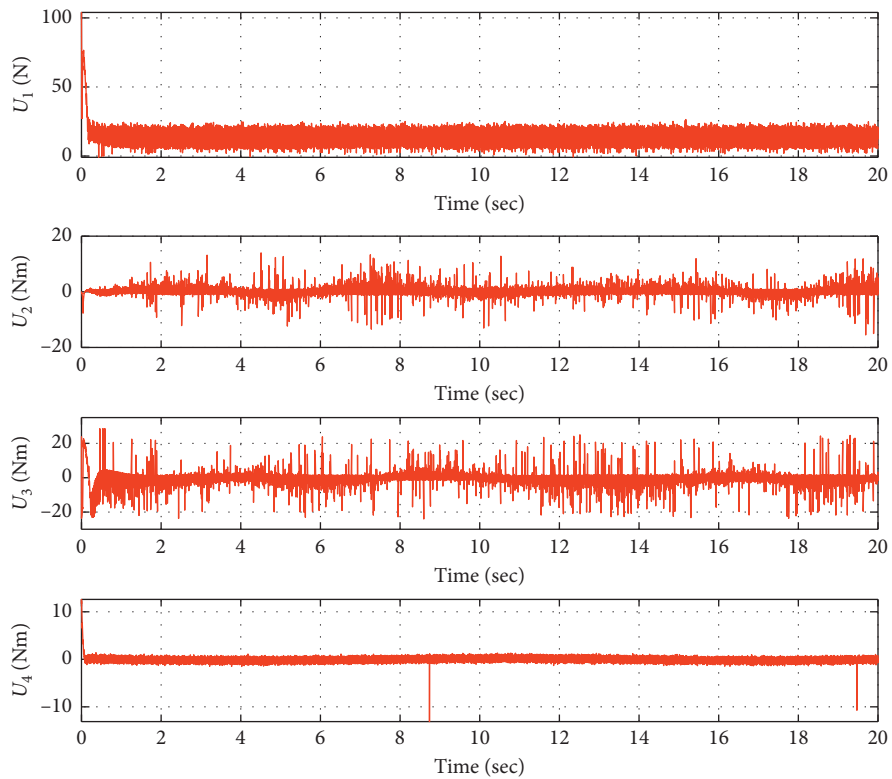


FIGURE 10: Control signals.

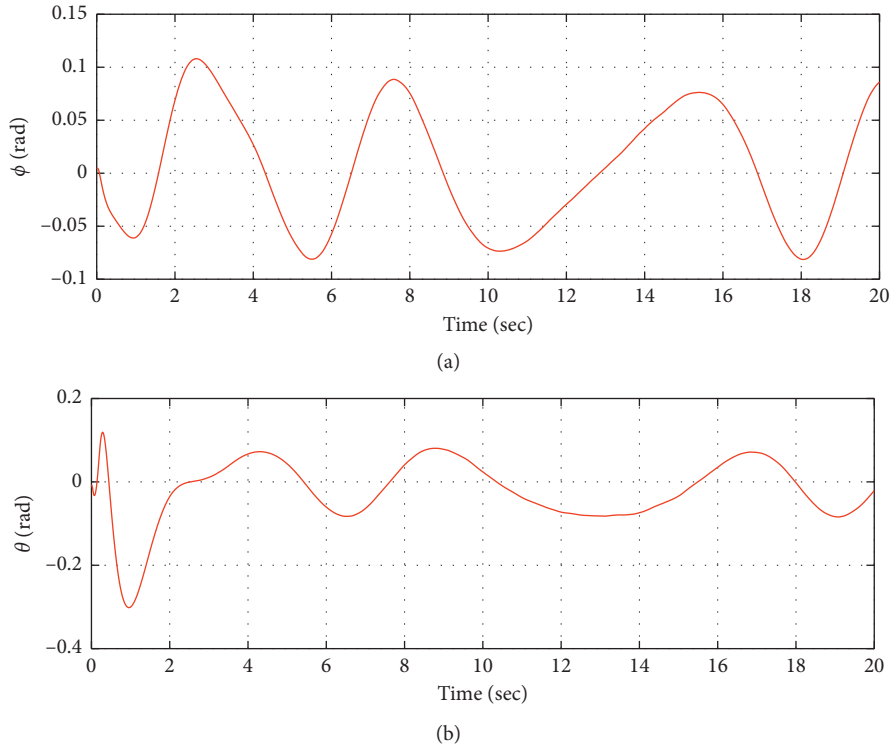


FIGURE 11: Change of roll and pitch angles.

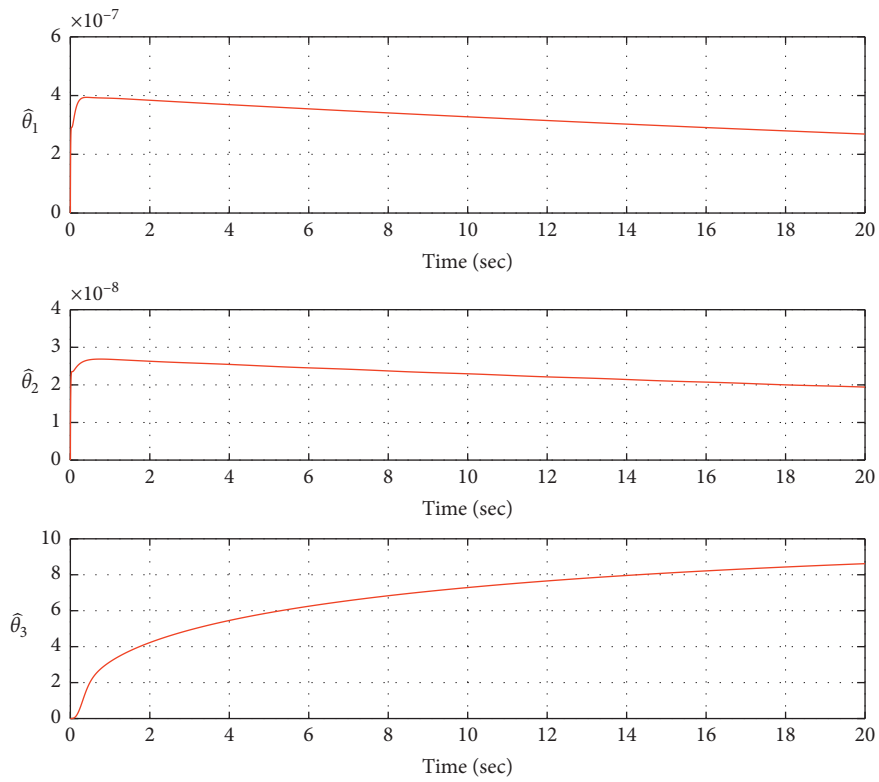


FIGURE 12: Adaptive parameters $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$.

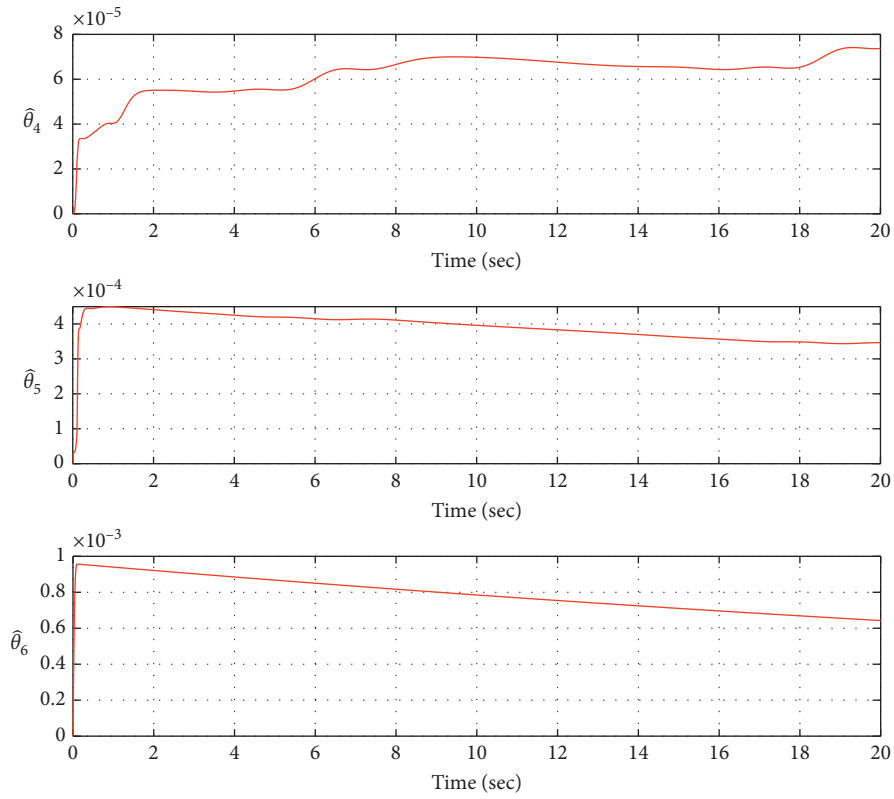


FIGURE 13: Adaptive parameters $\hat{\theta}_4$, $\hat{\theta}_5$, and $\hat{\theta}_6$.

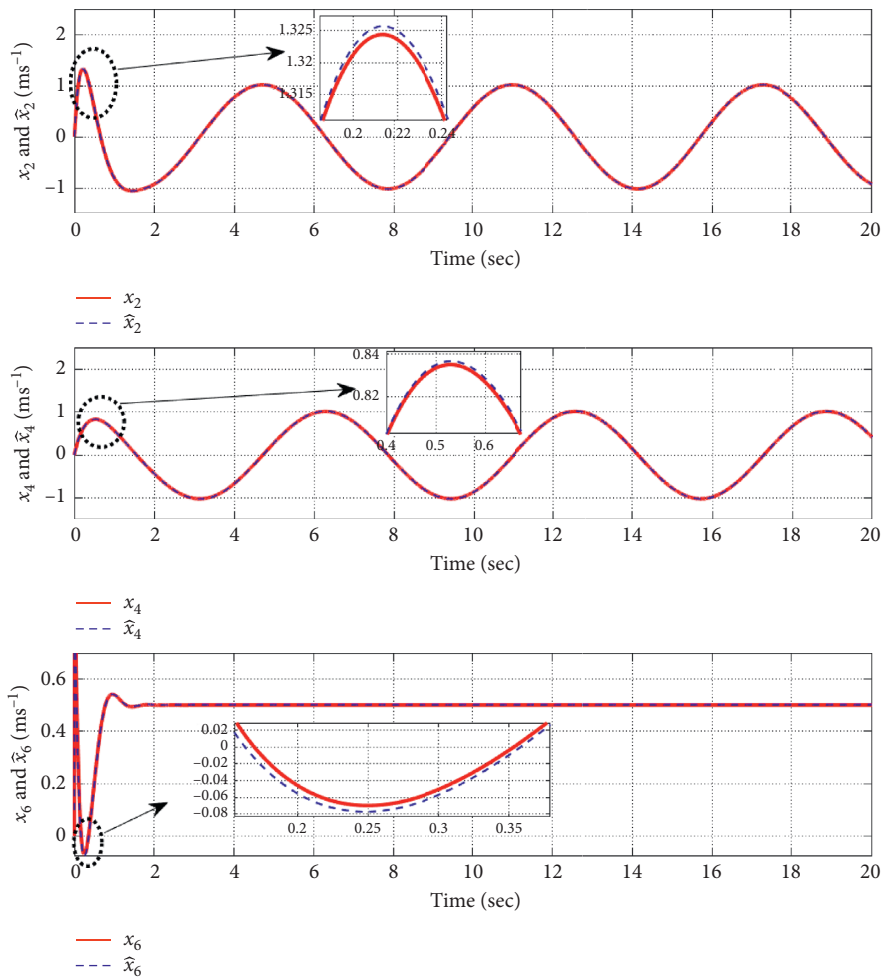


FIGURE 14: Actual value and estimated value.

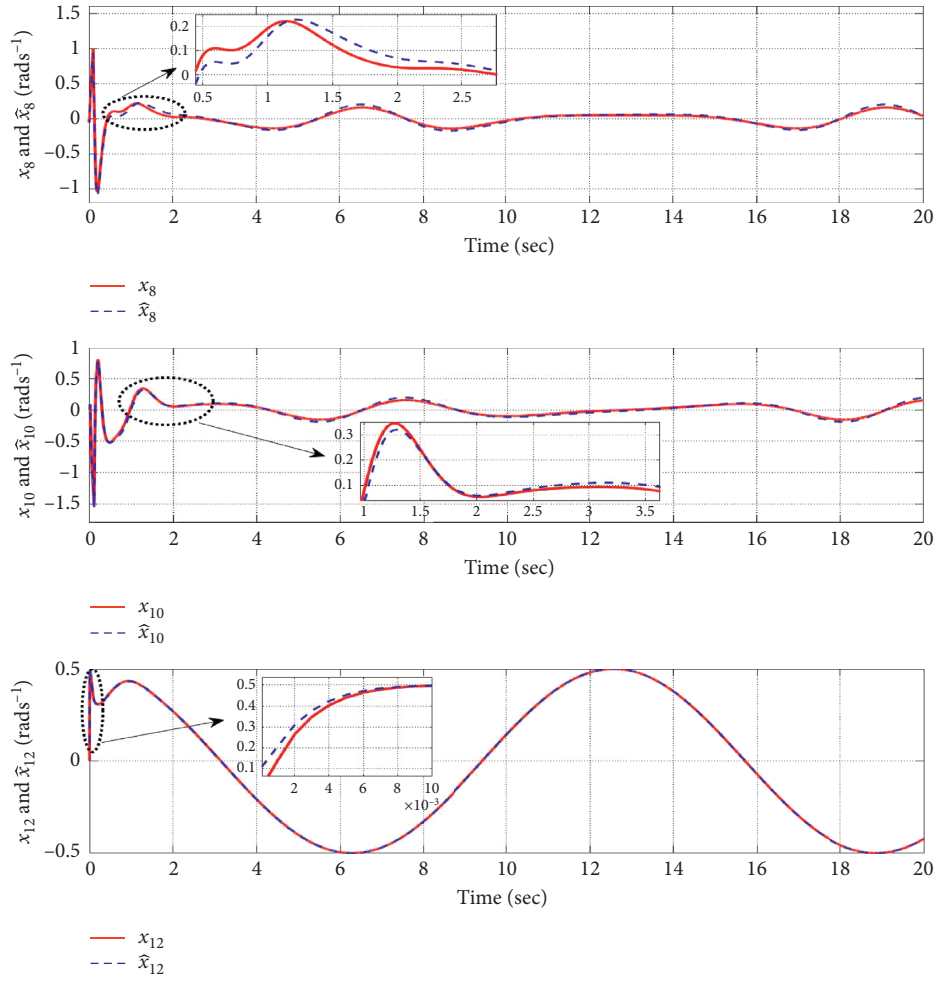


FIGURE 15: Actual value and estimated value.

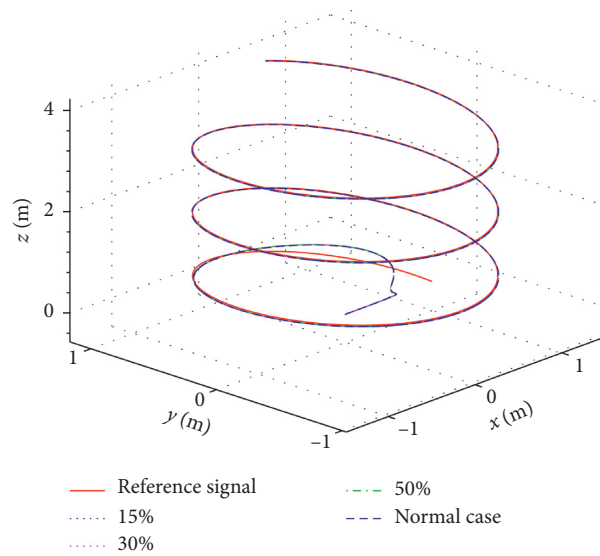


FIGURE 16: Space diagram of the position with normal, 15%, 30%, and 50% parameter uncertainty.

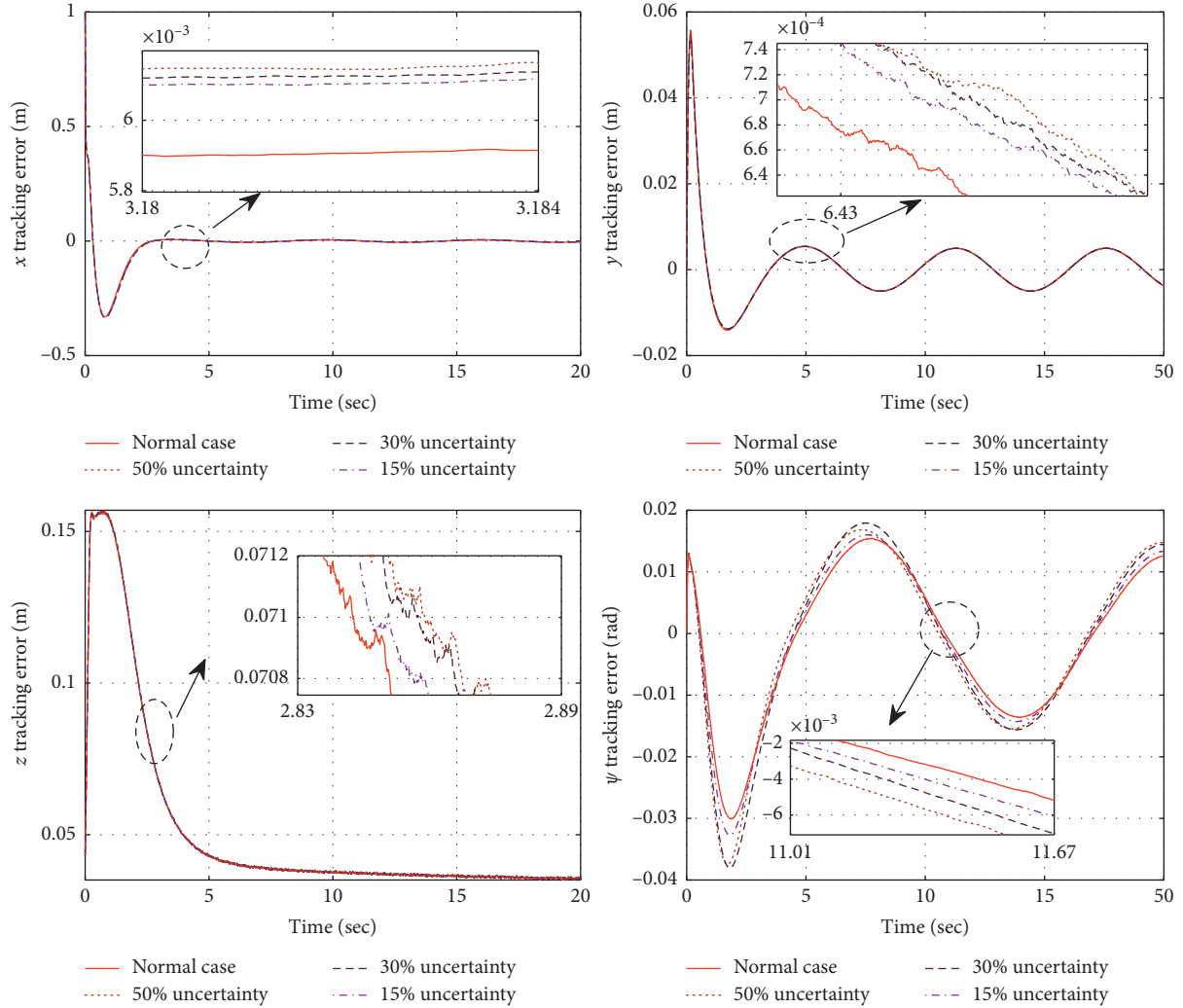


FIGURE 17: The tracking errors of different uncertainty cases added in the z rotating axis.

TABLE 4: The MVTE and RMSVTE.

Kind of errors	Normal case	Uncertainty 15%	Uncertainty 30%	Uncertainty 50%
MVTE	x (m)	$5.1254e^{-03}$	$5.5325e^{-03}$	$5.7471e^{-03}$
	y (m)	$4.9051e^{-03}$	$5.2051e^{-03}$	$5.3593e^{-03}$
	z (m)	$4.3216e^{-02}$	$4.4239e^{-02}$	$4.4799e^{-02}$
	ψ (rad)	$1.5415e^{-02}$	$1.6041e^{-02}$	$1.6869e^{-02}$
RMSVTE	x (m)	$3.6036e^{-03}$	$3.9049e^{-03}$	$4.1057e^{-03}$
	y (m)	$3.4844e^{-03}$	$3.4966e^{-03}$	$3.5137e^{-03}$
	z (m)	$3.7271e^{-02}$	$3.7414e^{-02}$	$3.7710e^{-02}$
	ψ (rad)	$0.9939e^{-02}$	$1.0468e^{-02}$	$1.1347e^{-02}$

shown in Table 4. From Table 4, we can see that the proposed control scheme has strong robustness, and even the uncertainty in yaw rotary inertia is up to 50%.

6. Conclusion

In this paper, an adaptive dynamic surface sliding-mode output feedback controller has been proposed for attitude and position control of a class of quadrotor UAVs with consideration of parametric uncertainties and disturbances.

By using the norm estimation approach, there is only one parameter which needs to be updated online at each design step regardless of the plant order and input-output dimension. Also, by introducing an error transformed function, the tracking performance of the quadrotor UAV has been achieved. The proposed control scheme can not only eliminate the problem of “explosion of complexity” existing in the backstepping control scheme but also improve the robustness of the system. The results of the hardware-in-loop simulation validate the effectiveness of the proposed

control algorithm. Furthermore, our future research work will focus on applying the control algorithm of this paper to the quadrotor UAV test platform for experimental verification.

Appendix

A. The Procedures of Controller Design

The controller design process.

Step 1. After error transformation, let S_1 given by (25) be the first error variable. Then, the derivative of S_1 can be expressed as

$$\dot{S}_1 = \eta_1 \dot{\hat{x}}_1 - \eta_1 v_1. \quad (\text{A.1})$$

According to adaptive laws (T1.2), (A.1) can be rewritten as

$$\dot{S}_1 = \eta_1 [\hat{x}_2 + k_1 (x_1 - \hat{x}_1)] - \eta_1 v_1, \quad (\text{A.2})$$

which suggests that we choose the virtual control signal \bar{x}_{2d} as $\bar{x}_{2d} = -k_1 (x_1 - \hat{x}_1) + v_1 - S_1 c_1 / \eta_1$, where c_1 is a positive constant. Introduce a new state variable x_{2d} , which can be obtained by the following first-order filter:

$$\tau_2 \dot{x}_{2d} + x_{2d} = \bar{x}_{2d}, \quad x_{2d}(0) = \bar{x}_{2d}(0). \quad (\text{A.3})$$

Define the error surface (T1.4), and the time derivative of S_2 is

$$\dot{S}_2 = \dot{\hat{x}}_2 - \dot{x}_{2d} = F_1(X_1) + \chi_1 + k_2 (x_1 - \hat{x}_1) - \dot{x}_{2d}, \quad (\text{A.4})$$

where variables $\chi_1 = (C_{x_7} S_{x_9} C_{x_{11}} + S_{x_7} S_{x_{11}}) U_1$ and $F_1(X_1) = \hat{f}(\hat{x}_2)$, $X_1 = \hat{x}_2$ are introduced. Since $F_1(X_1)$ is unknown, we use FLSs to approximate the function $F_1(X_1)$:

$$F_1(X_1) = W_1^* T \xi_1(X_1) + \sigma_1^*, \quad |\sigma_1^*| \leq \varepsilon_1, \quad (\text{A.5})$$

with respect to the unknown optimal weight vector in (A.5), define $\theta_1 = \|W_1^*\|^2$, and since θ_1 is unknown, let $\hat{\theta}_1$ be the estimation of θ_1 and $\tilde{\theta}_1 = \theta_1 - \hat{\theta}_1$. Choosing the following proper sliding surface $\sigma_{s1} = S_2$, consider the first Lyapunov function:

$$\Gamma_1 = \frac{1}{2} (S_1^2 + S_2^2 + \tilde{\theta}_1^2), \quad (\text{A.6})$$

where the differential of Lyapunov Function Γ_1 can be found as follows:

$$\begin{aligned} \dot{\Gamma}_1 &= S_1 \dot{S}_1 + S_2 \dot{S}_2 - \tilde{\theta}_1 \dot{\hat{\theta}}_1 \\ &= -c_1 S_1^2 + S_1 \eta_1 (\hat{x}_2 - \bar{x}_{2d}) + S_2 W_1^* T \xi_1(X_1) + S_2 \sigma_1^* - \frac{1}{2} S_2^2 \\ &\quad + S_2 \chi_1 + S_2 k_2 (x_1 - \hat{x}_1) - S_2 \dot{x}_{2d} - \tilde{\theta}_1 \dot{\hat{\theta}}_1. \end{aligned} \quad (\text{A.7})$$

Using Young's inequality, it can be verified that

$$\begin{aligned} S_2 W_1^* T \xi_1(X_1) &\leq \frac{1}{2} S_2^2 \|W_1^*\|^2 \xi_1^T(X_1) \xi_1(X_1) + \frac{1}{2} \\ &\leq \frac{1}{2} S_2^2 \theta_1 \xi_1^T(X_1) \xi_1(X_1) + \frac{1}{2}, \end{aligned} \quad (\text{A.8})$$

$$S_2 \sigma_1^* \leq \frac{1}{2} S_2^2 + \frac{1}{2} \varepsilon_1^2.$$

Then, (A.7) can be rewritten as

$$\begin{aligned} \dot{\Gamma}_1 &\leq -c_1 S_1^2 + \frac{1}{2} S_2^2 \tilde{\theta}_1 \xi_1^T(X_1) \xi_1(X_1) + S_2 \chi_1 + S_2 k_2 (x_1 - \hat{x}_1) - S_2 \dot{x}_{2d} \\ &\quad + \frac{1}{2} + \frac{1}{2} \varepsilon_1^2 - \tilde{\theta}_1 \left[\dot{\hat{\theta}}_1 - \frac{1}{2} S_2^2 \xi_1^T(X_1) \xi_1(X_1) \right] \\ &\quad + S_1 \eta_1 (\hat{x}_2 - \bar{x}_{2d}) + \frac{1}{2} S_2^2. \end{aligned} \quad (\text{A.9})$$

The stabilization of Γ_1 can be obtained by designing the virtual control (T1.5) and the adaptation law (T1.6), where c_2 , μ_1 , and λ_1 are the positive constants and ε_1 is an arbitrarily small positive constant. For the external disturbance d_{t1} encountered in the quadrotor flight process, the sliding surface is added to maintain system stability with $\mu_1 \geq |d_{t1}|$. Substituting (T1.5) and (T1.6) into (A.9), we get

$$\dot{\Gamma}_1 \leq -c_1 S_1^2 + S_1 \eta_1 (\hat{x}_2 - \bar{x}_{2d}) - \left(c_2 - \frac{1}{2} \right) S_2^2 + \frac{1}{2} + \frac{1}{2} \varepsilon_1^2 + \lambda_1 \tilde{\theta}_1 \hat{\theta}_1. \quad (\text{A.10})$$

Similar design procedures can be used to design adaptive DSC sliding-mode laws for trajectory tracking control of y axis position (x_3) and z axis position (x_5). Introduce two variables $\chi_2 = (C_{x_7} S_{x_9} S_{x_{11}} - S_{x_7} C_{x_{11}}) U_1$ and $\chi_3 = (C_{x_7} C_{x_9}) U_1$. The corresponding control laws and adaptive laws are designed as follows.

Step 2. Let S_3 given by (26) be the second error variable. Then, the derivative of S_3 can be expressed as

$$\dot{S}_3 = \eta_2 \dot{\hat{x}}_3 - \eta_2 v_2. \quad (\text{A.11})$$

According to (29) and (T1.8), (A.11) can be rewritten as

$$\dot{S}_3 = \eta_2 [\hat{x}_4 + k_1 (x_3 - \hat{x}_3)] - \eta_2 v_2, \quad (\text{A.12})$$

choosing the virtual control signal $\bar{x}_{4d} = -k_1 (x_3 - \hat{x}_3) + v_2 - S_3 c_3 / \eta_2$, where c_3 is a positive constant. Introduce a new state variable x_{4d} , which can be obtained by the following first-order filter:

$$\tau_4 \dot{x}_{4d} + x_{4d} = \bar{x}_{4d}, \quad x_{4d}(0) = \bar{x}_{4d}(0). \quad (\text{A.13})$$

Define the error surface (T1.10), and the time derivative of S_4 is

$$\dot{S}_4 = \dot{\hat{x}}_4 - \dot{x}_{4d} = F_2(X_2) + \chi_2 + k_2 (x_3 - \hat{x}_3) - \dot{x}_{4d}, \quad (\text{A.14})$$

where $F_2(X_2) = \hat{f}(\hat{x}_4)$, $X_2 = \hat{x}_4$. FLSs are used to approximate the unknown function $F_2(X_2)$:

$$F_2(X_2) = W_2^{*T} \xi_2(X_2) + \sigma_2^*, \quad |\sigma_2^*| \leq \varepsilon_2, \quad (\text{A.15})$$

with respect to the unknown optimal weight vector in (A.15), define $\theta_2 = \|W_2^*\|^2$, and since θ_2 is unknown, let $\hat{\theta}_2$ be the estimation of θ_2 and $\tilde{\theta}_2 = \theta_2 - \hat{\theta}_2$. Choosing the following proper sliding surface $\sigma_{s_2} = S_4$, consider the second Lyapunov function:

$$\Gamma_2 = \frac{1}{2} \left(S_3^2 + S_4^2 + \tilde{\theta}_2^2 \right). \quad (\text{A.16})$$

Differentiating Γ_2 , we obtain

$$\begin{aligned} \dot{\Gamma}_2 &= S_3 \dot{S}_3 + S_4 \dot{S}_4 - \tilde{\theta}_2 \dot{\hat{\theta}}_2 \\ &= -c_3 S_3^2 + S_3 \eta_2 (\hat{x}_4 - \bar{x}_{4d}) + S_4 W_2^{*T} \xi_2(X_2) + S_4 \sigma_2^* - \frac{1}{2} S_4^2 \\ &\quad + S_4 \chi_2 + S_4 k_2 (x_3 - \hat{x}_3) - S_4 \dot{x}_{4d} - \tilde{\theta}_2 \dot{\hat{\theta}}_2. \end{aligned} \quad (\text{A.17})$$

Using Young's inequality, it can be verified that

$$\begin{aligned} S_4 W_2^{*T} \xi_2(X_2) &\leq \frac{1}{2} S_4^2 \|W_2^*\|^2 \xi_2^T(X_2) \xi_2(X_2) + \frac{1}{2} \\ &\leq \frac{1}{2} S_4^2 \theta_1 \xi_2^T(X_2) \xi_2(X_2) + \frac{1}{2}, \end{aligned} \quad (\text{A.18})$$

$$S_4 \sigma_2^* \leq \frac{1}{2} S_4^2 + \frac{1}{2} \varepsilon_2^2.$$

Then, (A.17) can be rewritten as

$$\begin{aligned} \dot{\Gamma}_2 &\leq -c_3 S_3^2 + \frac{1}{2} S_4^2 \tilde{\theta}_2 \xi_2^T(X_2) \xi_2(X_2) + S_4 \chi_2 \\ &\quad + S_4 k_2 (x_3 - \hat{x}_3) - S_4 \dot{x}_{4d} \\ &\quad + \frac{1}{2} + \frac{1}{2} \varepsilon_2^2 - \tilde{\theta}_2 \left[\dot{\hat{\theta}}_2 - \frac{1}{2} S_4^2 \xi_2^T(X_2) \xi_2(X_2) \right] \\ &\quad + \frac{1}{2} S_4^2 + S_3 \eta_2 (\hat{x}_4 - \bar{x}_{4d}). \end{aligned} \quad (\text{A.19})$$

The stabilization of Γ_2 can be obtained by designing the virtual control (T1.11) and the adaptation law (T1.12), where c_4, μ_2 , and λ_2 are the positive constants and ε_2 is an arbitrarily small positive constant. For the external disturbance d_t encountered in the quadrotor flight process, the sliding surface is added to maintain system stability with $\mu_1 \geq |d_{t2}|$. Substituting (T1.11) and (T1.12) into (A.20), we get

$$\dot{\Gamma}_2 \leq -c_3 S_3^2 + S_3 \eta_2 (\hat{x}_4 - \bar{x}_{4d}) - \left(c_4 - \frac{1}{2} \right) S_4^2 + \frac{1}{2} + \frac{1}{2} \varepsilon_2^2 + \lambda_2 \tilde{\theta}_2 \dot{\hat{\theta}}_2. \quad (\text{A.20})$$

Step 3. Let S_5 given by (26) be the third error variable. Then, the derivative of S_5 can be expressed as

$$\dot{S}_5 = \eta_3 \hat{x}_5 - \eta_3 v_3. \quad (\text{A.21})$$

According to (29) and adaptive laws (T1.14), (A.21) can be rewritten as

$$\dot{S}_5 = \eta_3 [\hat{x}_6 + k_1 (x_5 - \hat{x}_5)] - \eta_3 v_3, \quad (\text{A.22})$$

and the virtual control signal can be chosen as \bar{x}_{6d} as $\bar{x}_{6d} = -k_1 (x_5 - \hat{x}_5) + v_3 - S_5 c_5 / \eta_3$ with c_5 being a positive constant. Introduce a new state variable x_{6d} , which can be obtained by the following first-order filter:

$$\tau_6 \dot{x}_{6d} + x_{6d} = \bar{x}_{6d}, \quad x_{6d}(0) = \bar{x}_{6d}(0). \quad (\text{A.23})$$

Define the error surface (T1.16), and the time derivative of S_6 is

$$\dot{S}_6 = \hat{x}_6 - \dot{x}_{6d} = F_3(X_3) + \chi_3 + k_2 (x_5 - \hat{x}_5) - \dot{x}_{6d}, \quad (\text{A.24})$$

where $F_3(X_3) = \hat{f}(\hat{x}_6)$, $X_3 = \hat{x}_6$. Utilizing FLSs to approximate the unknown function $F_3(X_3)$, we obtain

$$F_3(X_3) = W_3^{*T} \xi_3(X_3) + \sigma_3^*, \quad |\sigma_3^*| \leq \varepsilon_3. \quad (\text{A.25})$$

Define $\theta_3 = \|W_3^*\|^2$, and let $\hat{\theta}_3$ be the estimation of θ_3 and $\tilde{\theta}_3 = \theta_3 - \hat{\theta}_3$. Choosing the following proper sliding surface $\sigma_{s_3} = S_6$, consider the third Lyapunov function:

$$\Gamma_3 = \frac{1}{2} \left(S_5^2 + S_6^2 + \tilde{\theta}_3^2 \right). \quad (\text{A.26})$$

The differentiation of Γ_3 is as follows:

$$\begin{aligned} \dot{\Gamma}_3 &= S_5 \dot{S}_5 + S_6 \dot{S}_6 - \tilde{\theta}_3 \dot{\hat{\theta}}_3 \\ &= -c_5 S_5^2 + S_5 \eta_3 (\hat{x}_6 - \bar{x}_{6d}) + S_6 W_3^{*T} \xi_3(X_3) + S_6 \sigma_3^* - \frac{1}{2} S_6^2 \\ &\quad + S_6 \chi_3 + S_6 k_2 (x_5 - \hat{x}_5) - S_6 \dot{x}_{6d} - \tilde{\theta}_3 \dot{\hat{\theta}}_3. \end{aligned} \quad (\text{A.27})$$

Using Young's inequality, it can be verified that

$$\begin{aligned} S_6 W_3^{*T} \xi_3(X_3) &\leq \frac{1}{2} S_6^2 \|W_3^*\|^2 \xi_3^T(X_3) \xi_3(X_3) + \frac{1}{2} \\ &\leq \frac{1}{2} S_6^2 \theta_3 \xi_3^T(X_3) \xi_3(X_3) + \frac{1}{2}, \end{aligned} \quad (\text{A.28})$$

$$S_6 \sigma_3^* \leq \frac{1}{2} S_6^2 + \frac{1}{2} \varepsilon_3^2.$$

Then, (A.27) can be rewritten as

$$\begin{aligned} \dot{\Gamma}_3 &\leq -c_5 S_5^2 + \frac{1}{2} S_6^2 \tilde{\theta}_3 \xi_3^T(X_3) \xi_3(X_3) + S_6 \chi_3 \\ &\quad + S_6 k_2 (x_5 - \hat{x}_5) - S_6 \dot{x}_{6d} \\ &\quad + \frac{1}{2} + \frac{1}{2} \varepsilon_3^2 - \tilde{\theta}_3 \left[\dot{\hat{\theta}}_3 - \frac{1}{2} S_6^2 \xi_3^T(X_3) \xi_3(X_3) \right] + \frac{1}{2} S_6^2 \\ &\quad + S_5 \eta_3 (\hat{x}_6 - \bar{x}_{6d}). \end{aligned} \quad (\text{A.29})$$

The stabilization of Γ_3 can be obtained by designing the virtual control (T1.17) and the adaptive law (T1.18), where c_6, μ_3 , and λ_3 are the positive constants and ε_3 is an arbitrarily small positive constant. For the external disturbance d_{t_3} , the sliding surface is added to maintain system stability with $\mu_3 \geq |d_{t_3}|$. Substituting (T1.17) and (T1.18) into (A.29), we get

$$\dot{\Gamma}_3 \leq -c_5 S_5^2 + S_5 \eta_3 (\hat{x}_6 - \bar{x}_{6d}) - \left(c_6 - \frac{1}{2} \right) S_6^2 + \frac{1}{2} + \frac{1}{2} \varepsilon_3^2 + \lambda_3 \bar{\theta}_3 \hat{\theta}_3. \quad (\text{A.30})$$

By associating χ_1, χ_2 , and χ_3 , the virtual controllers are obtained as

$$\begin{aligned} \chi_1 &= (C_{x_7} S_{x_9} C_{x_{11}} + S_{x_7} S_{x_{11}}) U_1, \\ \chi_2 &= (C_{x_7} S_{x_9} S_{x_{11}} - S_{x_7} C_{x_{11}}) U_1, \\ \chi_3 &= (C_{x_7} C_{x_9}) U_1. \end{aligned} \quad (\text{A.31})$$

Notably, (A.31) has four degrees of freedom, namely, x_7, x_9, x_{11} , and U_1 . We consider the reference trajectory for yaw angle x_{11d} , which is usually given in advance, and the corresponding DSC sliding-mode law U_4 is directly designed in the next section to ensure the rapid convergence of x_{11} to x_{11d} . Thus, we regarded x_{11} as known that can be replaced by x_{11d} in the controller, and the degrees of freedom in (A.31) is reduced so that x_7, x_9 , and U_1 can be solved. The programs are as follows:

$$\begin{cases} \bar{x}_{7d} = \arctan\left(\frac{C_{x_9} \beta \chi_1 - \alpha \chi_2}{\chi_3}\right), \\ \bar{x}_{9d} = \arctan\left(\frac{\alpha \chi_1 + \beta \chi_2}{\chi_3}\right), \\ U_1 = \frac{\chi_3}{C_{x_7} C_{x_9}}, \end{cases} \quad (\text{A.32})$$

where U_1 is one of the ultimate control laws, and in addition, $\alpha = \cos(x_{11d})$ and $\beta = \sin(x_{11d})$. Introduce two new state variables x_{7d} and x_{9d} , which can be obtained by the following first-order filters:

$$\begin{aligned} \tau_7 \dot{x}_{7d} + x_{7d} &= \bar{x}_{7d}, & x_{7d}(0) &= \bar{x}_{7d}(0), \\ \tau_9 \dot{x}_{9d} + x_{9d} &= \bar{x}_{9d}, & x_{9d}(0) &= \bar{x}_{9d}(0). \end{aligned} \quad (\text{A.33})$$

For attitude trajectory tracking control, by taking $\{x_{7d}, x_{9d}, x_{11d}\}$ as the desired attitude trajectory, the design of control laws contains three steps. The attitude dynamic system can be extracted as follows:

$$\begin{cases} \dot{x}_7 = x_8, \\ \dot{x}_8 = a_4 x_{10} x_{12} + a_5 x_{10} \Omega - a_6 x_8 + U_2, \\ \dot{x}_9 = x_{10}, \\ \dot{x}_{10} = a_7 x_8 x_{12} + a_8 x_8 \Omega - a_9 x_{10} + U_3, \\ \dot{x}_{11} = x_{12}, \\ \dot{x}_{12} = a_{10} x_8 x_{10} - a_{11} x_{12} + U_4. \end{cases} \quad (\text{A.34})$$

Step 4. Let S_7 given by (26) be the fourth error variable. Then, the derivative of S_7 can be expressed as

$$\dot{S}_7 = \eta_4 \dot{\hat{x}}_7 - \eta_4 v_4. \quad (\text{A.35})$$

According to (29) and adaptive laws (T2.2) and (T2.3), where $\hat{f}(\hat{x}_8, \hat{x}_{10}, \hat{x}_{12}) = \Delta a_4 \hat{x}_{10} \hat{x}_{12} + \Delta a_5 \hat{x}_{10} \Omega - \Delta a_6 \hat{x}_8$, (A.35) can be rewritten as

$$\dot{S}_7 = \eta_4 [\hat{x}_8 + k_1 (x_7 - \hat{x}_7)] - \eta_4 v_4, \quad (\text{A.36})$$

which suggests that we choose the virtual control signal \bar{x}_{8d} as $\bar{x}_{8d} = -k_1 (x_7 - \hat{x}_7) + v_4 - S_7 c_7 / \eta_4$, where c_7 is a positive constant. Introduce a new state variable x_{8d} , which can be obtained by the following first-order filter:

$$\tau_8 \dot{x}_{8d} + x_{8d} = \bar{x}_{8d}, \quad x_{8d}(0) = \bar{x}_{8d}(0). \quad (\text{A.37})$$

Define the error surface (T2.5), and the time derivative of S_8 is

$$\begin{aligned} \dot{S}_8 &= \hat{x}_8 - \dot{x}_{8d} = a_{4N} \hat{x}_{10} \hat{x}_{12} + a_{5N} \hat{x}_{10} \Omega - a_{6N} \hat{x}_8 \\ &\quad + F_4(X_4) + U_2 + k_2 (x_7 - \hat{x}_7) - \dot{x}_{8d}, \end{aligned} \quad (\text{A.38})$$

where $F_4(X_4) = \hat{f}(\hat{x}_8, \hat{x}_{10}, \hat{x}_{12})$, $X_4 = [\hat{x}_8, \hat{x}_{10}, \hat{x}_{12}]^T$. Utilizing FLSs to approximate the unknown function $F_4(X_4)$, we obtain

$$F_4(X_4) = W_4^* T \xi_4(X_4) + \sigma_4^*, \quad |\sigma_4^*| \leq \varepsilon_4. \quad (\text{A.39})$$

Define $\theta_4 = \|W_4^*\|^2$, and let $\hat{\theta}_4$ be the estimation of θ_4 and $\tilde{\theta}_4 = \theta_4 - \hat{\theta}_4$. Choosing the following proper sliding surface $\sigma_{s4} = S_8$, consider the following Lyapunov function:

$$\Gamma_4 = \frac{1}{2} (S_7^2 + S_8^2 + \tilde{\theta}_4^2), \quad (\text{A.40})$$

where the differential of Lyapunov function Γ_4 can be found as follows:

$$\begin{aligned} \dot{\Gamma}_4 &= S_7 \dot{S}_7 + S_8 \dot{S}_8 - \tilde{\theta}_4 \dot{\hat{\theta}}_4 \\ &= -c_7 S_7^2 + S_7 \eta_4 (\hat{x}_8 - \bar{x}_{8d}) + S_8 W_4^* T \xi_4(X_4) + S_8 \sigma_4^* - \frac{1}{2} S_8^2 \\ &\quad + S_8 U_2 + S_8 k_2 (x_7 - \hat{x}_7) - S_8 \dot{x}_{8d} + \tilde{\theta}_4 \dot{\hat{\theta}}_4. \end{aligned} \quad (\text{A.41})$$

Using Young's inequality, it can be verified that

$$\begin{aligned} S_8 W_4^* T \xi_4(X_4) &\leq \frac{1}{2} S_8^2 \|W_4^*\|^2 \xi_4^T(X_4) \xi_4(X_4) + \frac{1}{2} \\ &\leq \frac{1}{2} S_8^2 \theta_4 \xi_4^T(X_4) \xi_4(X_4) + \frac{1}{2}, \end{aligned} \quad (\text{A.42})$$

$$S_8 \sigma_4^* \leq \frac{1}{2} S_8^2 + \frac{1}{2} \varepsilon_4^2.$$

Then, (A.41) can be rewritten as

$$\begin{aligned}
\dot{\Gamma}_4 \leq & -c_7 S_7^2 + \frac{1}{2} S_8^2 \widehat{\theta}_4 \xi_4^T(X_4) \xi_4(X_4) + S_8 (a_4 N \widehat{x}_{10} \widehat{x}_{12} \\
& + a_{5N} \widehat{x}_{10} \Omega - a_{6N} \widehat{x}_8) + S_8 U_2 + S_8 k_2 (x_7 - \widehat{x}_7) \\
& - S_8 \dot{x}_{8d} + \frac{1}{2} + \frac{1}{2} \varepsilon_4^2 + \frac{1}{2} S_8^2 \\
& + S_7 \eta_4 (\widehat{x}_8 - \bar{x}_{8d}) \\
& - \widehat{\theta}_4 \left[\dot{\theta}_4 - \frac{1}{2} S_8^2 \xi_4^T(X_4) \xi_4(X_4) \right].
\end{aligned} \tag{A.43}$$

The stabilization of Γ_4 can be obtained by designing the virtual control (T2.6) and the adaptation law (T2.7), where c_8, μ_4 , and λ_4 are positive constants and ε_1 is an arbitrarily small positive constant. For the external disturbance d_{t4} encountered in the quadrotor flight process, the sliding surface is added to maintain system stability with $\mu_4 \geq |d_{t4}|$. Substituting (T2.6) and (T2.7) into (A.43), we get

$$\dot{\Gamma}_4 \leq -c_7 S_7^2 + S_7 \eta_4 (\widehat{x}_8 - \bar{x}_{8d}) - \left(c_8 - \frac{1}{2} \right) S_8^2 + \frac{1}{2} + \frac{1}{2} \varepsilon_4^2 + \lambda_4 \widehat{\theta}_4 \widehat{\theta}_4. \tag{A.44}$$

Similarly, the adaptive DSC sliding-mode laws for trajectory tracking control of pitch angle (x_9) and yaw angle (x_{11}) can be designed as (T2.13), (T2.20), (T2.14), and (T2.21), where $S_{10} = \widehat{x}_{10} - x_{10d}$, $S_{12} = \widehat{x}_{12} - x_{12d}$, $X_5 = [\widehat{x}_8, \widehat{x}_{10}, \widehat{x}_{12}]^T$, $X_6 = [\widehat{x}_8, \widehat{x}_{10}, \widehat{x}_{12}]^T$, x_{10d} and x_{12d} are the output of each first-order filter, and $c_9, c_{10}, c_{11}, c_{12}, \lambda_5, \lambda_6, \mu_5$, and μ_6 are the positive constants.

The derivative of Lyapunov candidate for pitch angle (x_9) and yaw angle (x_{11}) are designed as follows:

$$\begin{aligned}
\dot{\Gamma}_5 \leq & -c_9 S_9^2 + S_9 \eta_5 (\widehat{x}_{10} - \bar{x}_{10d}) - \left(c_{10} - \frac{1}{2} \right) S_{10}^2 + \frac{1}{2} \\
& + \frac{1}{2} \varepsilon_5^2 + \lambda_5 \widehat{\theta}_5 \widehat{\theta}_5, \\
\dot{\Gamma}_6 \leq & -c_{11} S_{11}^2 + S_{11} \eta_6 (\widehat{x}_{12} - \bar{x}_{12d}) - \left(c_{12} - \frac{1}{2} \right) S_{12}^2 + \frac{1}{2} \\
& + \frac{1}{2} \varepsilon_6^2 + \lambda_6 \widehat{\theta}_6 \widehat{\theta}_6.
\end{aligned} \tag{A.45}$$

B. Proof of Theorem 1

Taking the time derivative of Γ_i , ($i = 1, \dots, 6$) and combining (38) yields

$$\begin{aligned}
\dot{\Gamma}_i \leq & -c_{(2i-1)} S_{(2i-1)}^2 + S_{(2i-1)} \eta_i (S_{2i} + y_{2i}) - \left(c_{2i} - \frac{1}{2} \right) S_{2i}^2 \\
& + \frac{1}{2} + \frac{1}{2} \varepsilon_i^2 + \lambda_i \widehat{\theta}_i \widehat{\theta}_i,
\end{aligned}$$

$$y_{2i} \dot{y}_{2i} \leq -\frac{y_{2i}^2}{\tau_{2i}} + B_{2i} |y_{2i}|, \quad i = 1, \dots, 6,$$

$$y_7 \dot{y}_7 \leq -\frac{y_7^2}{\tau_7} + B_7 |y_7|,$$

$$y_9 \dot{y}_9 \leq -\frac{y_9^2}{\tau_9} + B_9 |y_9|.$$

(B.1)

Consider the sets

$$\begin{aligned}
Y_1 := & \left\{ (x_d, \dot{x}_d, x_d, y_d, \dot{y}_d, y_d, \dot{y}_d, z_d, \dot{z}_d, z_d, \psi_d, \dot{\psi}_d, \psi_d): x_d^2 + \dot{x}_d^2 + x_d^2 + y_d^2 + \dot{y}_d^2 + y_d^2 \right. \\
& \left. + z_d^2 + \dot{z}_d^2 + z_d^2 + \psi_d^2 + \dot{\psi}_d^2 + \psi_d^2 \right\} \leq B_0,
\end{aligned} \tag{B.2}$$

$$Y_2 := \left\{ \sum_{i=1}^6 \Gamma_i + \sum_{i=1}^6 \frac{1}{2} y_{2i}^2 + \frac{1}{2} y_7^2 + \frac{1}{2} y_9^2 \leq p \right\},$$

where $Y_1 \times Y_2$ is also in a compact set. Then, the continuous functions $B_{2i}(\cdot)$, ($i = 1, \dots, 6$), B_7, B_9 have maximums on $Y_1 \times Y_2$, say, M_i ($i = 1, \dots, 6$), M_7, M_9 . Thus,

$$\begin{aligned}
y_{2i}y_{2i}^2 &\leq -\frac{y_{2i}^2}{\tau_{2i}} + \frac{M_i^2|y_{2i}^2|}{2\mathcal{Q}} + \frac{\mathcal{Q}}{2}, \\
y_7y_7^2 &\leq -\frac{y_7^2}{\tau_7} + \frac{M_7^2|y_7^2|}{2\mathcal{Q}} + \frac{\mathcal{Q}}{2}, \\
y_9y_9^2 &\leq -\frac{y_9^2}{\tau_9} + \frac{M_9^2|y_9^2|}{2\mathcal{Q}} + \frac{\mathcal{Q}}{2},
\end{aligned} \tag{B.3}$$

and using the following inequalities,

$$\eta_i S_{(2i-1)}(S_{2i} + y_{2i}) \leq \eta_i^2 S_{(2i-1)}^2 + \frac{1}{2}S_{2i}^2 + \frac{1}{2}y_{2i}^2, \quad i = 1, \dots, 6, \tag{B.4}$$

which together with (B.1) implies that

$$\begin{aligned}
\dot{\Gamma} &\leq \frac{1}{2}\dot{\tilde{x}}^T P \tilde{x} + \frac{1}{2}\tilde{x}^T P \dot{\tilde{x}} - \sum_{i=1}^6 (c_{(2i-1)} - \eta_i^2) S_{(2i-1)}^2 - \sum_{i=1}^6 (c_{2i} - 1) S_{2i}^2 \\
&\quad + \sum_{i=1}^6 \lambda_i \tilde{\theta}_i \hat{\theta}_i + \sum_{i=1}^6 \left(\frac{1}{2} + \frac{1}{2}\varepsilon_i^2 \right) - \sum_{i=1}^6 \left(\frac{1}{\tau_{2i}} - \frac{M_i^2}{2\mathcal{Q}} - \frac{1}{2} \right) y_{2i}^2 \\
&\quad - \left(\frac{1}{\tau_7} - \frac{M_7^2}{2\mathcal{Q}} \right) y_7^2 - \left(\frac{1}{\tau_9} - \frac{M_9^2}{2\mathcal{Q}} \right) y_9^2 + 4\mathcal{Q} \\
&\leq -\frac{1}{2}\lambda_{\min}(Q_1)\|\tilde{x}\|^2 + \tilde{x}^T P \sum_{i=1}^6 b_i \tilde{W}_i^* \xi_i(\tilde{X}_i) \\
&\quad + \tilde{x}^T P \varepsilon - \sum_{i=1}^6 (c_{(2i-1)} - \eta_i^2) S_{(2i-1)}^2 - \sum_{i=1}^6 (c_{2i} - 1) S_{2i}^2 + \sum_{i=1}^6 \lambda_i \tilde{\theta}_i \hat{\theta}_i \\
&\quad + \sum_{i=1}^6 \left(\frac{1}{2} + \frac{1}{2}\varepsilon_i^2 \right) - \sum_{i=1}^6 \left(\frac{1}{\tau_{2i}} - \frac{M_i^2}{2\mathcal{Q}} - \frac{1}{2} \right) y_{2i}^2 \\
&\quad - \left(\frac{1}{\tau_7} - \frac{M_7^2}{2\mathcal{Q}} \right) y_7^2 - \left(\frac{1}{\tau_9} - \frac{M_9^2}{2\mathcal{Q}} \right) y_9^2 + 4\mathcal{Q},
\end{aligned} \tag{B.5}$$

where $\varepsilon = [0, \varepsilon_1, 0, \varepsilon_2, 0, \varepsilon_3, 0, \varepsilon_4, 0, \varepsilon_5, 0, \varepsilon_6]^T$ and $\lambda_{\min}(Q_1)$ is the smallest eigenvalue of matrix Q_1 . Applying Young's inequality $ab \leq (a^2 + b^2)/2$ and the fact $\xi_i^T(\tilde{X}_i)\xi_i(\tilde{X}_i) \leq 1$, the following inequation can be obtained:

$$\tilde{x}^T P \sum_{i=1}^6 b_i \tilde{W}_i^* \xi_i(\tilde{X}_i) \leq \frac{1}{2}\lambda_{\max^2}(P)\|\tilde{x}\|^2 + \frac{1}{2}\|\tilde{W}^*\|^2 \leq \frac{1}{2}\lambda_{\max}^2(P)\|\tilde{x}\|^2 + \frac{1}{2}\tilde{\theta}, \tag{B.6}$$

where $\tilde{\theta} = \tilde{\theta}_1 + \tilde{\theta}_2 + \tilde{\theta}_3 + \tilde{\theta}_4 + \tilde{\theta}_5 + \tilde{\theta}_6$ and $\lambda_{\max}^2(P)$ is the largest eigenvalue of matrix P .

$$\tilde{x}^T P \varepsilon \leq \frac{1}{2}\|\tilde{x}\|^2 + \frac{1}{2}\lambda_{\max}^2(P)\|\varepsilon^*\|^2. \tag{B.7}$$

By using Young's inequality, we obtain

$$2\tilde{\theta}_i \hat{\theta}_i \leq \theta_i^2 - \tilde{\theta}_i^2. \tag{B.8}$$

Substituting (B.6) and (B.7) into (B.5), we have

$$\begin{aligned}
\dot{\Gamma} &\leq -\mu_1\|\tilde{x}\|^2 + \frac{1}{2}\tilde{\theta} + \frac{1}{2}\lambda_{\max^2}(P)\|\varepsilon^*\|^2 - \sum_{i=1}^6 (c_{(2i-1)} - \eta_i^2) S_{(2i-1)}^2 \\
&\quad - \sum_{i=1}^6 (c_{2i} - 1) S_{2i}^2 \\
&\quad - \sum_{i=1}^6 \frac{1}{2}\lambda_i \tilde{\theta}_i^2 - \sum_{i=1}^6 \left(\frac{1}{2} + \frac{1}{2}\varepsilon_i^2 \right) - \sum_{i=1}^6 \left(\frac{1}{\tau_{2i}} - \frac{M_i^2}{2\mathcal{Q}} - \frac{1}{2} \right) y_{2i}^2 \\
&\quad - \left(\frac{1}{\tau_7} - \frac{M_7^2}{2\mathcal{Q}} \right) y_7^2 - \left(\frac{1}{\tau_9} - \frac{M_9^2}{2\mathcal{Q}} \right) y_9^2 + 4\mathcal{Q} + \sum_{i=1}^6 \frac{1}{2}\lambda_i \theta_i^2 \\
&\leq -\mu_1\|\tilde{x}\|^2 - \sum_{i=1}^6 (c_{(2i-1)} - \eta_i^2) S_{(2i-1)}^2 - \sum_{i=1}^6 (c_{2i} - 1) S_{2i}^2 \\
&\quad - \sum_{i=1}^6 \frac{1}{2}\lambda_i \tilde{\theta}_i^2 + \frac{1}{2}\tilde{\theta} \\
&\quad - \sum_{i=1}^6 \left(\frac{1}{\tau_{2i}} - \frac{M_i^2}{2\mathcal{Q}} - \frac{1}{2} \right) y_{2i}^2 - \left(\frac{1}{\tau_7} - \frac{M_7^2}{2\mathcal{Q}} \right) y_7^2 - \left(\frac{1}{\tau_9} - \frac{M_9^2}{2\mathcal{Q}} \right) y_9^2 + C,
\end{aligned} \tag{B.9}$$

where

$$\mu_1 = \left(\frac{1}{2} \right) \lambda_{\min}(Q_1) - \left(\frac{1}{2} \right) \lambda_{\max}^2(P) - 1,$$

$$C = \frac{1}{2}\lambda_{\max}^2(P)\|\varepsilon^*\|^2 + 3 + 4\mathcal{Q} + \sum_{i=1}^6 \frac{1}{2}\lambda_i \theta_i^2 - \sum_{i=1}^6 \left(\frac{1}{2} + \frac{1}{2}\varepsilon_i^2 \right). \tag{B.10}$$

Choose the suitable design parameters to make $\mu_1 > 0$ and satisfy $\sqrt{C/\mu_1} < \|\tilde{x}\|$. The following conditions hold:

$$\begin{aligned}
c_{(2i-1)} &\geq \eta_i^2 + r, \\
c_{2i} &\geq 1 + r, \\
\frac{1}{\tau_{2i}} &\geq \frac{1}{2} + \frac{M_i^2}{2\mathcal{Q}} + r, \\
\frac{1}{2}\lambda_i &\geq \frac{1}{8} + r, \\
\frac{1}{\tau_7} &\geq \frac{M_7^2}{2\mathcal{Q}} + r, \\
\frac{1}{\tau_9} &\geq \frac{M_9^2}{2\mathcal{Q}} + r,
\end{aligned} \tag{B.11}$$

where r is the positive constant. Thus, we have $\dot{\Gamma} \leq 0$. Hence, all the signals of the closed-loop system are semiglobal bounded. Particularly, the tracking errors of position and attitude angle can converge to an arbitrarily residual set and are always kept in the prespecified cures. This completes the proof.

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 work was supported in part by the NSF of China under Grant 61673101 and 61304015, the Natural Science Foundation of Jilin Province under Grant 20180201009SF and 20170414011GH, the Thirteenth Five-Year Science Research Plan of Jilin Province under Grant JJKH20200119KJ, and the Jilin Technological Innovation Development Plan under Grant 201831719.

References

- [1] Z. Bo, B. Xian, Z. Yao, and Z. Xu, "Nonlinear robust adaptive tracking control of a quadrotor uav via immersion and invariance methodology," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2891–2902, 2015.
- [2] P. E. I. Pounds and A. M. Dollar, "Stability of helicopters in compliant contact under PD-PID control," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1472–1486, 2014.
- [3] T. Huang, D. Huang, Z. Wang, and A. Shah, "Robust tracking control of a quadrotor UAV based on adaptive sliding mode controller," *Complexity*, vol. 2019, Article ID 7931632, 15 pages, 2019.
- [4] H. Kong, C. Yang, G. Li, and S.-L. Dai, "A semg-based shared control system with no-target obstacle avoidance for omnidirectional mobile robots," *IEEE Access*, vol. 8, pp. 26030–26040, 2020.
- [5] X. Zhang, Y. Wang, G. Zhu et al., "Compound adaptive fuzzy quantized control for quadrotor and its experimental verification," *IEEE Transactions on Cybernetics*, pp. 1–13, 2020.
- [6] M. Taimoor, S. Iqbal, U. I. Bhatti, and A. A. Khan, "Autonomous flight of unmanned aerial vehicle (UAV) by using linear quadratic regulator (LQR)," *International Journal of Control and Automation*, vol. 8, no. 1, pp. 9–28, 2015.
- [7] F. Rinaldi, S. Chiesa, and F. Quagliotti, "Linear quadratic control for quadrotors uavs dynamics and formation flight," *Journal of Intelligent and Robotic Systems*, vol. 70, no. 1–4, pp. 203–220, 2013.
- [8] T. Ryan and H. Jin Kim, "LMI-based gain synthesis for simple robust quadrotor control," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 4, pp. 1173–1178, 2013.
- [9] M. Emam and A. Fakharian, "Attitude tracking of quadrotor uav via mixed H₂/H_∞ controller: an LMI based approach," in *Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED)*, Athens, Greece, June 2016.
- [10] J. P. Ortiz, L. I. Minchala, and M. J. Reinoso, "Nonlinear robust H-infinity PID controller for the multivariable system quadrotor," *IEEE Latin America Transactions*, vol. 14, no. 3, pp. 1176–1183, 2016.
- [11] M. Kerma, A. Mokhtari, B. Abdelaziz, and Y. Orlov, "Nonlinear H_∞ control of a quadrotor (UAV), using high order sliding mode disturbance estimator," *International Journal of Control*, vol. 85, no. 12, pp. 1876–1885, 2012.
- [12] H. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," in *Proceedings of the IEEE International Conference on Mechatronics*, pp. 1–6, Malaga, Spain, April 2009.
- [13] K. Alexis, G. Nikolakopoulos, and A. Tzes, "On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances," *Asian Journal of Control*, vol. 16, no. 1, pp. 209–224, 2014.
- [14] A. Hao, H. Xia, and C. Wang, "Barrier lyapunov function-based adaptive control for hypersonic flight vehicles," *Nonlinear Dynamics*, vol. 88, no. 3, pp. 1833–1853, 2017.
- [15] P. Deng, L. Zeng, and Y. Liu, "RBF neural network backstepping sliding mode adaptive control for dynamic pressure cylinder electrohydraulic servo pressure system," *Complexity*, vol. 2018, Article ID 4159639, 16 pages, 2018.
- [16] F. Chen, W. Lei, K. Zhang, G. Tao, and B. Jiang, "A novel nonlinear resilient control for a quadrotor UAV via backstepping control and nonlinear disturbance observer," *Nonlinear Dynamics*, vol. 85, no. 2, pp. 1281–1295, 2016.
- [17] F. Chen, R. Jiang, K. Zhang, B. Jiang, and G. Tao, "Robust backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5044–5056, 2016.
- [18] G. Perozzi, D. Efimov, J. M. Biannic, and L. Planckaert, "Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains," *Journal of the Franklin Institute*, vol. 355, no. 12, pp. 4809–4838, 2018.
- [19] S. Li, Y. Wang, J. Tan, and Y. Zheng, "Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft," *Neurocomputing*, vol. 216, pp. 126–134, 2016.
- [20] Y. Yin, H. Niu, and X. Liu, "Adaptive neural network sliding mode control for quad tilt rotor aircraft," *Complexity*, vol. 2017, Article ID 7104708, 13 pages, 2017.
- [21] W. Hao and B. Xian, "Nonlinear adaptive fault-tolerant control for a quadrotor UAV based on immersion and invariance methodology," *Nonlinear Dynamics*, vol. 90, no. 4, pp. 2813–2826, 2017.
- [22] X. Shao, J. Liu, H. Cao, C. Shen, and H. Wang, "Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 7, pp. 2700–2719, 2018.
- [23] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.
- [24] G. Zhu, L. Nie, Z. Lv, L. Sun, X. Zhang, and C. Wang, "Adaptive fuzzy dynamic surface sliding mode control of large-scale power systems with prescribe output tracking performance," *ISA Transactions*, vol. 99, pp. 305–321, 2020.
- [25] X. Zhang, Y. Wang, X. Chen et al., "Decentralized adaptive neural approximated inverse control for a class of large-scale nonlinear hysteretic systems with time delays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2424–2437, 2019.
- [26] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.
- [27] X. Zhang, Z. Xu, C.-Y. Su et al., "Fuzzy approximator based adaptive dynamic surface control for unknown time delay nonlinear systems with input asymmetric hysteresis nonlinearities," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2218–2232, 2017.

- [28] H. M. Lin, T. Dierks, S. Jagannathan, and L. Acar, "Modified consensus-based output feedback control of quadrotor UAV formations using neural networks," *Journal of Intelligent and Robotic Systems*, vol. 94, no. 1, pp. 283–300, 2019.
- [29] Z. Shen, F. Li, X. Cao, and C. Guo, "Prescribed performance dynamic surface control for trajectory tracking of quadrotor uav with uncertainties and input constraints," *International Journal of Control*, pp. 1–11, 2020.
- [30] M. A. M. Basri, A. R. Husain, and K. A. Danapalasingam, "Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle," *Journal of Intelligent and Robotic Systems*, vol. 79, no. 2, pp. 295–321, 2015.
- [31] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1893–1899, 2000.
- [32] X. Zhang, Y. Wang, C. Wang, C.-Y. Su, Z. Li, and X. Chen, "Adaptive estimated inverse output-feedback quantized control for piezoelectric positioning stage," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2106–2118, 2019.
- [33] X. Zhang, Z. Li, C.-Y. Su, Y. Lin, and Y. Fu, "Implementable adaptive inverse control of hysteretic systems via output feedback with application to piezoelectric positioning stages," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5733–5743, 2016.
- [34] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.
- [35] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
- [36] X. Zhang, B. Li, X. Chen, Z. Li, Y. Peng, and S. Chunyi, "Adaptive implicit inverse control for a class of discrete-time hysteretic nonlinear systems and its application," *IEEE Transactions on Mechatronics*, 2020.
- [37] C. Wang and Y. Lin, "Multivariable adaptive backstepping control: a norm estimation approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 989–995, 2012.
- [38] X. Zhang, C.-Y. Su, Y. Lin, L. Ma, and J. Wang, "Adaptive neural network dynamic surface control for a class of time-delay nonlinear systems with hysteresis inputs and dynamic uncertainties," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2844–2860, 2015.
- [39] Y. Zhou, M. Chen, and C. Jiang, "Robust tracking control of uncertain MIMO nonlinear systems with application to UAVs," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 1, pp. 25–32, 2015.
- [40] H. Wang, B. Su, Y. Wang, and J. Gao, "Adaptive sliding mode fixed-time tracking control based on fixed-time sliding mode disturbance observer with dead-zone input," *Complexity*, vol. 2019, Article ID 8951382, 14 pages, 2019.
- [41] Y. Li, S. Tong, L. Liu, and G. Feng, "Adaptive output-feedback control design with prescribed performance for switched nonlinear systems," *Automatica*, vol. 80, pp. 225–231, 2017.
- [42] Y. Li and S. Tong, "Adaptive fuzzy output-feedback stabilization control for a class of switched nonstrict-feedback nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1007–1016, 2017.
- [43] L. Zhang, Y. Li, and S. Tong, "Adaptive fuzzy output feedback control for MIMO switched nonlinear systems with prescribed performances," *Fuzzy Sets and Systems*, vol. 306, pp. 153–168, 2017.
- [44] L. Xing, C. Wen, Y. Zhu, H. Su, and Z. Liu, "Output feedback control for uncertain nonlinear systems with input quantization," *Automatica*, vol. 65, pp. 191–202, 2016.
- [45] X. Zhang, X. Chen, G. Zhu, and C.-Y. Su, "Output feedback adaptive motion control and its experimental verification for time-delay nonlinear systems with asymmetric hysteresis," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6824–6834, 2020.
- [46] C. P. Bechlioulis and G. A. Rovithakis, "Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems," *Automatica*, vol. 45, no. 2, pp. 532–538, 2009.
- [47] C. P. Bechlioulis and G. A. Rovithakis, "Prescribed performance adaptive control for multi-input multi-output affine in the control nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1220–1226, 2010.
- [48] Y. H. Kim, F. L. Lewis, and C. T. Abdallah, "A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems," *Automatica*, vol. 33, no. 8, pp. 1539–1543, 1997.
- [49] C.-K. Chan and C.-C. Tsai, "Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball-riding robot," in *Proceedings of the 2012 International Conference on Fuzzy Theory and Its Applications (IFUZZY2012)*, Taichung, Taiwan, November 2012.
- [50] Y. Fang, J. Fei, and T. Hu, "Adaptive backstepping fuzzy sliding mode vibration control of flexible structure," *Journal of Low Frequency Noise Vibration and Active Control*, vol. 37, no. 4, pp. 1079–1096, 2018.

Research Article

A Bi-Level Probabilistic Path Planning Algorithm for Multiple Robots with Motion Uncertainty

Jingchuan Wang ¹, Ruochen Tai ², and Jingwen Xu ¹

¹Department of Automation, Key Laboratory of System Control and Information Processing of Ministry of Education of China, Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, China

²School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 637121, Singapore

Correspondence should be addressed to Jingchuan Wang; jchwang@sjtu.edu.cn

Received 18 April 2020; Accepted 15 May 2020; Published 5 June 2020

Guest Editor: Ning Wang

Copyright © 2020 Jingchuan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For improving the system efficiency when there are motion uncertainties among robots in the warehouse environment, this paper proposes a bi-level probabilistic path planning algorithm. In the proposed algorithm, the map is partitioned into multiple interconnected districts and the architecture of proposed algorithm is composed of topology level and route level generating from above map: in the topology level, the order of passing districts is planned combined with the district crowdedness to achieve the district equilibrium and reduce the influence of robots under motion uncertainty. And in the route level, a MDP method combined with probability of motion uncertainty is proposed to plan path for all robots in each district separately. At the same time, the number of steps for each planning is dependent on the probability to decrease the number of planning. The conflict avoidance is proved, and optimization is discussed for the proposed algorithm. Simulation results show that the proposed algorithm achieves improved system efficiency and also has acceptable real-time performance.

1. Introduction

In the running of multiple robots in warehouses, motion uncertainties happen unavoidably since there exist some disabled components of robots, the instability of networks, and the interference of people walking. Under this case, robots would not follow the designed paths and the coupled relationship between temporal and space domain for paths is broken. And there is no doubt that other robots are disturbed by the ones where motion uncertainties happen. Finally, the whole multirobot system would be influenced and this brings about chaos or even breakdown of the whole system. It can be seen that this poses a great challenge to the path planning of multiple robots in warehouses. Therefore, taking motion uncertainty into consideration in the path planning of multiple robots is an issue worthy of attention and research.

As a kind of method taking motion uncertainty into consideration for the discretized map, Ma et al. [1] designed

the path planning algorithm which contains relevant restricted principles that an agent cannot enter the vertex that is occupied by some other agents at the last time step. The conflicts can still occur and this would cause the breakdown of the multirobot system in warehouses. Another kind of method which can deal with the issues caused by motion uncertainties needs to replan the paths of multiple robots when the samplings of robots contain abnormal information. Li et al. [2] proposed a two-stage trajectory planning scheme where a directed graph with cost variable connection is established and cost variable depends on the reachability, the risk of collision, and different state constraints. Gombolay et al. [3] presented a centralized algorithm that handles tightly intercoupled temporal and spatial constraints and scales to environment with large number of robots, which aims at improving the efficiency firstly. However, the above method cannot guarantee the real-time performance. In [4], the authors proposed a hybrid architecture which combines centralized coordination with distributed freedom of action

to achieve an appropriate interplay. The architecture takes the new sampling for the state of robots at each time step, and path planning algorithm would be adopted to coordinate paths of multiple robots when sampling information containing motion uncertainties. However, the real-time performance cannot be guaranteed and the efficiency of the whole system is reduced due to too frequent planning. The algorithm introduced in [5, 6] based on generalized probabilistic roadmaps (GPRM) also coordinates paths of multiple robots under motion uncertainty by using the feedback information. In the algorithm, the passive planning strategy is introduced in the multiple traveling salesman problem (MTSP) solution methodology. However, the system efficiency is low and the scalability of the algorithm cannot be guaranteed because far more points need to be sampled. A novel method in [7] based on the sampling considering the motion uncertainty would propagate this delay. Robots' paths need to be replanned adopting the Pareto-optimal plan repair scheme. However, the problem concerned with the decreased system efficiency is still not solved.

As a classical method devoted to solving the uncertainty problem, the Markov decision process (MDP) is introduced to the field of multiple agents. Corresponding multiagent MDPs (MMDPs) have been designed for multiple agent planning. In [8], approximating methods are introduced to be combined with MMDPs for the path planning problem of a team of homogeneous microair vehicles (MAVs) towards a set of goals. This algorithm can guarantee that there are no conflicts. Ma et al. [9] proposed the use of robust plan-execution policies to control how each robot proceeds along its path with a 2-level MDP solver that generates valid plans. To optimize the solutions, a novel optimal solver is designed in [10] for the problem of motion uncertainties in the transition-independent MMDPs. In [11], an equilibrium policy has been proposed in MMDPs. Based on the notion of macroactions, the path planning problem is transferred into the decentralized partially observable Markov decision processes (DecPOMDPs) in [12, 13]. Multiagent reinforcement learning (MARL) [14–20] solves the learning task in multiagent systems (MASs). Nevertheless, the reasoning agent would deal with learning tasks which grow exponentially according to the number of agents. Therefore, the computational complexity of these methods based on MDPs is so high that the real-time scalability performance cannot be guaranteed.

Recently, the planning methods based on homotopy classes have been proposed to solve the problem of motion uncertainties. In [21, 22], the coordination space is established, which is the base for the later path planning. By selecting homotopy classes of the paths which are collision-free, a centralized controller is designed for the path planning problem under motion uncertainties. Furthermore, the encoding based on priorities, which is called the priority graphs, is introduced for the homotopic solutions to solve the coordination problem in [23]. For [21–23], the planning method based on homotopy classes can be proved that there are no conflicts for the coordinating paths and make sure that all robots will eventually reach their destinations even when some robots are temporarily stopped by a delay disturbance. However, the quality of solutions is

decreased to a great extent since this kind of method only considers the next step of motions and all the robots would follow the previous planned paths.

Although the above methods ensure that conflict can be avoided among the planning of multiple agents under motion uncertainty, problem still exists. The efficiency of a multirobot system with motion uncertainty cannot be guaranteed. Therefore, the first objective of this paper is to design an algorithm which can plan conflict-free paths for multiple robots even when motion uncertainty happens. And another objective is to improve the system efficiency that includes the makespan of the whole system and the number of planning.

To attain these goals, in this paper, the map is partitioned into multiple districts and we propose a bi-level planning architecture that is composed of topology level and route level generating from above map.

In the topology level, the passing orders of districts are planned based on the weight of paths in topology map. And in this process, the weight changes dynamically according to the distribution of robots; the aim of planning in the topology level is to achieve the district equilibrium, which can in turn reduce the influence of robots under motion uncertainty and promote the system efficiency. In the route level, a MDP-based probabilistic method is proposed to plan path for all robots in each district separately. The MDP is combined with the probability of motion uncertainty to improve efficiency. And the number of steps for each planning is dependent on the probability to decrease the number of planning. We proved that the designed algorithm is conflict-free. And we also validated the system efficiency improvement of designed algorithm from the optimization discussion and simulation.

The remainder of this paper is organized as follows. Section 2 provides the problem statement. In Section 3, the path planning algorithm for multiple robots is introduced. Section 4 proves that the proposed algorithm is conflict-free and discusses the optimization. Simulation validation results are reported in Section 5. Section 6 concludes our research.

2. Problem Formulation

The problem that this paper attempts to solve is the path planning of multiple AGVs with motion uncertainty in warehouses. In this problem, warehouse environment is discretized and simplified as a graph $G(V, E)$ whose vertices V correspond to locations and whose edges E correspond to transitions between locations. Figure 1 shows the discretized map.

Let $R = \{r_{(1)}, r_{(2)}, \dots, r_{(n)}\}$ be a set of robots. Time is discretized into time steps. To formally describe a plan, the path for a robot $r_{(i)}$ is a map $\mathbf{p}_{(i)}$ which means that the location $p_{(i)}(t)$ of $r_{(i)}$ in each time step $t \in N^*$ belongs to the vertices V . A robot can either remain stationary or move to an adjacent vertex. Robots are allowed to move simultaneously but restricted to move on the same edge in opposite directions or move at the same vertex simultaneously to ensure that there are no conflicts among multiple robots. Figure 2 gives the illustrations of these restricted conditions.

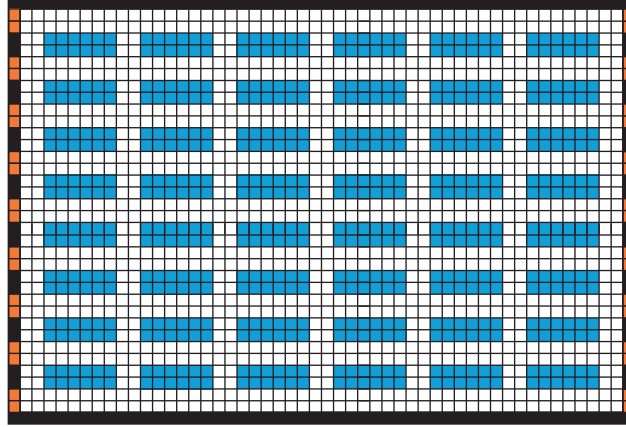


FIGURE 1: The discretized map of the warehouse. The orange and blue squares represent pickup points of the shelves and storage points, respectively. Other squares represent feasible ways.

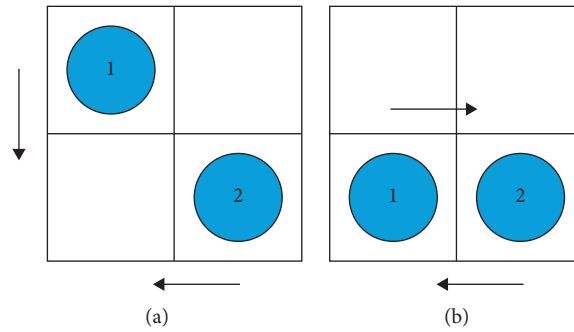


FIGURE 2: Restricted condition for motions of multiple robots in warehouses.

Let $T = \{t_{(1)}, \dots, t_{(v)}\}$ be a set of tasks which are generated continually. For each robot that has been assigned a task, it has a start position (pickup point) and a goal position (storage point). And the makespan of all the tasks is M . During the running of robots, motion uncertainty, which means that robots have to stop immediately, could happen at any time. And the probability of the uncertainty is p_u .

This paper assumes the following about the scheduling of multiple AGVs in warehouses:

- (i) All feasible routes are two-way bidirectional streets, which means that two robots can drive side by side and robots can drive in both directions on each feasible way.
- (ii) The warehouse is made up of shelves and feasible ways, which is shown in Figure 1 (not limited to this scenario).
- (iii) The locations of all the robots are known.
- (iv) The probability of the uncertainty is known.
- (v) The robots in the state of delay will return to normal. And after the recovery, these robots will be put into operation in the warehouse system.
- (vi) The task allocation is not taken into consideration in this paper.

In this paper, letters of the subscripts which are not in the parentheses are used for identification. Letters or numbers of the subscripts which are in the parentheses are used for index.

Therefore, the problem studied in this paper can be formulated in the following equationfd1:

$$\min M \text{ s.t. } \begin{cases} p_{(i)}(t) \neq p_{(j)}(t+1), & \forall i, j \in [1, n], i \neq j, \\ p_{(j)}(t) \neq p_{(i)}(t+1), & \forall i, j \in [1, n], i \neq j, \\ p_{(i)}(t) \neq p_{(j)}(t), & \forall i, j \in [1, n], i \neq j. \end{cases} \quad (1)$$

3. The Bi-Level Path Planning Algorithm for Multiple Robots

In this section, the bi-level path planning algorithm is introduced. The aim of this structure is that district equilibrium would be taken into consideration in the topology level for reducing the influence of robots under motion uncertainty and distributed path planning in each district can improve system efficiency in the route level even when motion uncertainty occurs. The architecture of the algorithm is shown in Figure 3. It is divided into two levels generating from the map partition: topology level and route level. In the

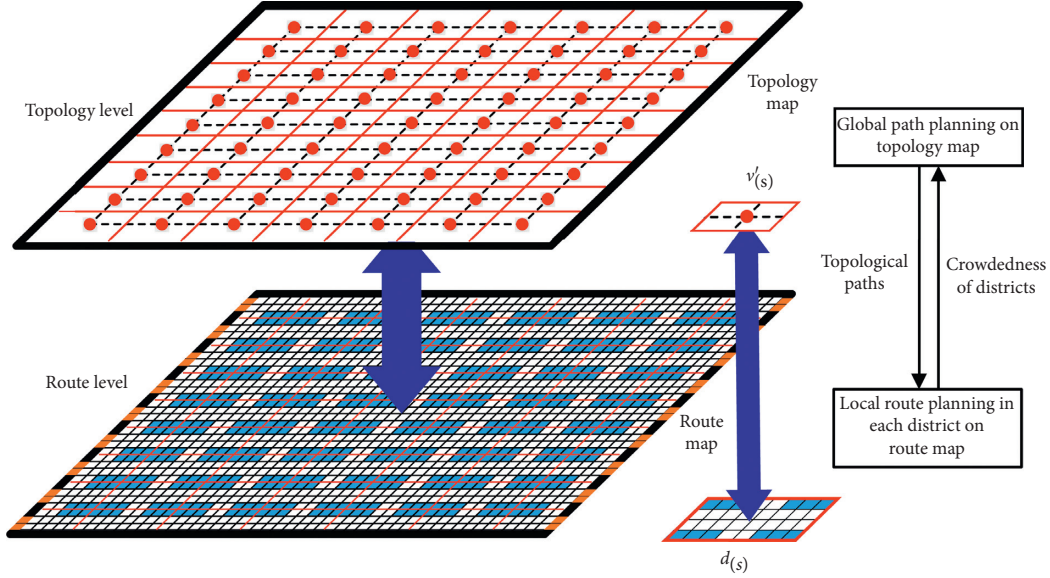


FIGURE 3: The bi-level architecture of the algorithm.

topology level, the algorithm plans the order of passing districts. In the route level, the algorithm plans paths for robots in each district separately, considering the motion uncertainty. And the introduction of map partition is as follows.

3.1. Map Partition. In this paper, the discretized map in Figure 1 is partitioned into multiple districts $D = \{d_{(1)}, \dots, d_{(M)}\}$ as shown in Figure 3. For the topology level, the generation of the topology map $G'(V', E')$ is listed as follows: (1) each vertex $v' \in V'$ corresponds to district $d_{(s)}$; (2) each edge $(v'_{(s)}, v'_{(s')}) \in E'$ represents that districts $d_{(s)}$ and $d_{(s')}$ are connected. Let the horizontal edges and vertical edges be E'_h and E'_v . For the route level, all feasible ways of each district in the discretized map of Figure 1 are shown in the route map.

3.2. Topology Level. In the topology level, the first step of the planning problem introduced in Section 3 is to plan the passing orders of districts for the robots which have been assigned tasks. At each time step, there might be multiple tasks assigned to robots. The number of these tasks is N and the robots corresponding to these tasks are $R_t = \{r_{(1)}, \dots, r_{(t)}\} (t \leq n)$. According to the task $t_{(i)}$ which has been assigned to the robot $r_{(i)}$, the districts which contain the start position and goal position of $t_{(i)}$ are

$d_{(s_{(i)})}$ and $d_{(g_{(i)})}$, respectively (here $s_{(i)}$ and $g_{(i)}$ represent the index of the corresponding district). To formally describe the plan results, the passing order of districts for a robot $r_{(i)}$ is a map $\mathbf{o}_{(i)}: N^* \rightarrow D$.

In this paper, $T (T \in N^*)$ is defined as the macroscopical time, which is the independent discrete variable in the topology level. With the increase of T , the robot will move from the start district $d_{(s_{(i)})}$ to the goal district $d_{(g_{(i)})}$. Different from the discrete time step, the macroscopic time represents the temporal space and corresponds to the district in topology level. Therefore, the change of T represents the transfer of the robot in the topology level. And if districts $\mathbf{o}_{(i)}(T)$ and $\mathbf{o}_{(i)}(T+1)$ are different, then the corresponding edge that robot $r_{(i)}$ will pass in the graph $G'(V', E')$ is defined as follows:

$$E_{(i)}(T) = (\mathbf{o}_{(i)}(T), \mathbf{o}_{(i)}(T+1)). \quad (2)$$

In this formulation, the edge $E_{(i)}(T)$ represents the transition from $\mathbf{o}_{(i)}(T)$ to $\mathbf{o}_{(i)}(T+1)$, which is the moving direction for the robot $r_{(i)}$. For the robot $r_{(i)}$ which has been assigned the task, there is the corresponding macroscopical time $T_{s_{(i)}}$ and $T_{g_{(i)}}$ when $r_{(i)}$ reaches the start district $d_{(s_{(i)})}$ and goal district $d_{(g_{(i)})}$, respectively. Then, the planning problem concerned with the districts can be formulated as follows:

$$\min \sum_{i=1}^t \sum_{T=T_{s_{(i)}}}^{T_{g_{(i)}}} L(E_{(i)}(T))(1 + C(T, E_{(i)}(T))) \text{ s.t. } \begin{cases} \mathbf{o}_{(i)}(T_{s_{(i)}}) = d_{(s_{(i)})}, & 1 \leq i \leq t, \\ \mathbf{o}_{(i)}(T_{g_{(i)}}) = d_{(g_{(i)})}, & 1 \leq i \leq t. \end{cases} \quad (3)$$

In this formulation of the optimization objective, $L(\cdot)$ is the length of the edge $E_{(i)}(T)$, which is the corresponding distance between the center of the two districts. Due to the length difference of the edges in the graph $G'(V', E')$, the durations of passing these edges are different. In our problem, the length of the horizontal edge is twice that of the

vertical edge. Therefore, the duration of passing the horizontal edge is twice that of the vertical edge. For the other environment map with different layout, the same principle applies. According to this principle, the passing order for the robot $r_{(i)}$ should satisfy the following:

$$o_{(i)}(T+1) = \begin{cases} o_{(i)}(T), & (o_{(i)}(T-1), o_{(i)}(T)) \in E'_h \text{ and } o_{(i)}(T-2) \neq o_{(i)}(T-1), \\ \text{next district,} & \text{otherwise.} \end{cases} \quad (4)$$

Based on this principle, equation (2) can be rewritten as

$$E_{(i)}(T+1) = \begin{cases} E_{(i)}(T), & E_{(i)}(T-1) \in E'_h \text{ and } E_{(i)}(T-2) \neq E_{(i)}(T-1), \\ \text{next edge,} & \text{otherwise.} \end{cases} \quad (5)$$

In the optimization objective of equation (3), $C(\cdot)$ represents the crowdedness of the edge $E_{(i)}(T)$ at macroscopic time T (here we assume that $o_{(i)}(T)$ and $o_{(i)}(T+1)$ are different). And it is defined as follows:

$$C(T, E_{(i)}(T)) = \frac{NC_{(o_{(i)}(T))}(T) + NC_{(o_{(i)}(T+1))}(T)}{2}, \quad (6)$$

where $NC_{(o_{(i)}(T))}(T)$ and $NC_{(o_{(i)}(T+1))}(T)$ are the normalized crowdedness of the district $o_{(i)}(T)$ and $o_{(i)}(T+1)$, respectively. And the crowdedness of each district $o_{(i)}(T)$ is denoted as $C_{(o_{(i)}(T))}(T)$, which represents the number of robots in each district at the macroscopic time T . It can be obtained by the passing orders for each robot. Furthermore, these values are changing dynamically since robots are moving. Then, the method of normalization can be adopted to calculate $NC_{(o_{(i)}(T))}(T)$ and $NC_{(o_{(i)}(T+1))}(T)$.

Furthermore, due to the motion uncertainty of robots themselves, robots would not arrive at the planned district precisely. Therefore, at each macroscopic time, there are multiple robots which are assigned new tasks and need new path planning in current warehouse environment plan, the normalized crowdedness in equation (6) should be recalculated according to the distribution and passing orders of all robots in topology level. To be specific, assuming that one robot is located at district $d_{(s1)}$ and will move to district $d_{(s2)}$ when there are some new tasks and the planning is needed at the macroscopic time T , if the edge corresponding to the transition between $d_{(s1)}$ and $d_{(s2)}$ belongs to E'_h , then this robot would still stay at this edge at next time $(T+1)$ in equation (6). If the edge belongs to E'_v , then this robot would move to next edge at macroscopic time $(T+1)$.

Therefore, the optimization objective in equation (3) considers the length of the edges which robots would pass and the crowdedness of the related districts, whose aim is to make sure that the paths of robots would not be too long and the districts which have been planned for these robots would not be too crowded. And this design can guarantee the balance among different districts and efficiency of the whole warehousing system.

The constraints listed in equation (3) make sure that the algorithm can plan paths for robots from their start position to the goal position in topology level.

To solve the optimization problem in equation (3), K shortest path planning algorithm [24] is adopted to obtain the multiple shortest paths for each robot which needs to be planned at the moment. According to the planned paths, the passing order can be generated by using equations (4) and (5). The optimization objective can be calculated based on the rules introduced above, and the optimal solution among all the combinations can be selected, which contains the information concerned with the passing order for the robots assigned with new tasks in the topology level.

Figure 4 shows an example of the planning results in topology level. And it can be seen that the planning results follow the analysis above; the algorithm in the topology level prefers to choose the paths which are shorter and less crowded since this can minimize the optimization objective mentioned in equation (3).

3.3. Route Level. In the bi-level architecture, when the planning of the topology level has been finished, according to the distribution of the robots, the planning of each district would be conducted for the robots in this district distributedly.

In the route level, the route map is made up of all the feasible ways in each district. And the local route planning is to plan the actions of all the robots in the district being planned.

In the planning of each district, considering the motion uncertainty of robots and the probability p_u , the planning step α is introduced that should be intuitively related to p_u (α not only refers to the planning step at each planning but also refers to the interval between each two plannings). The aim of this idea is that the planning step can be adjusted dynamically based on p_u and some redundant planning in [11, 12] would not be needed anymore. Here, we define the planning step as the follows:

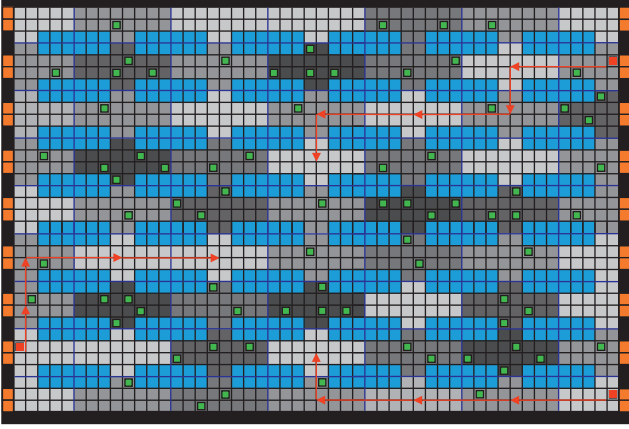


FIGURE 4: Illustration of the planning result in topology level. Red squares represent the robots which need to be planned. Green squares represent the other robots. The red line shows the planned passing order for each robot assigned with new task. According to the number of robots, the grey shade implies the crowdedness of each district. The fewer the number of robots, the lighter the color.

$$\alpha = \lfloor \frac{1}{p_u} \rfloor. \quad (7)$$

It can be seen from this definition that there is an inverse relationship between the planning step S and the probability of motion uncertainty p_u . When p_u gets larger, which means that there is higher probability that robots would not follow the previous planned path, the planning step α becomes smaller and the algorithm would plan paths for robots more conservatively. The first aim of computation for probability of motion uncertainty is to guarantee the real-time performance of the planning in route level since the traditional replanning all the paths from the current position to the goal position. However, the planning step is dependent on the probability of the motion uncertainty. Another aim of this is the planning step is adjusted based on p_u , which can prevent the situation that frequent replanning might destroy the optimized planned paths.

In the route level, the planning of each district can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\pi_{(x)}} & E(R(S_{(1)}) + \lambda \cdot R(S_{(2)}) + \dots + \lambda^{(\alpha-1)} R(S_{(\alpha)})), \\ \text{s.t.} & \begin{cases} p_{(i)}(t) \neq p_{(j)}(t+1), & \forall i, j \in [1, m], i \neq j, \\ p_{(j)}(t) \neq p_{(i)}(t+1), & \forall i, j \in [1, m], i \neq j, \\ p_{(i)}(t) \neq p_{(j)}(t), & \forall i, j \in [1, m], i \neq j. \end{cases} \end{aligned} \quad (8)$$

In the formulation of this optimization objective, $\pi_{(x)}$ represents one policy and there are k policies which are $\pi_{(1)}, \pi_{(2)}, \dots, \pi_{(k)}$. Each policy contains the information concerned with actions of the robots at each time step in the district. And for each policy, $\pi_{(x)} = (\pi_{(x),(1)}, \pi_{(x),(2)}, \dots, \pi_{(x),(\alpha)})$ where $\pi_{(x),(1)}$ indicates the policy at the first time step for all the robots in the district being planned. And $\pi_{(x),(1)} = (\pi_{(x),(1)}^{r_{(1)}}, \pi_{(x),(1)}^{r_{(2)}}, \dots, \pi_{(x),(1)}^{r_{(m)}})$ ($m \leq t$) where $\pi_{(x),(1)}^{r_{(1)}}$ indicates the policy at the first time step for the robot $r_{(1)}$ and m is the number of the robots located in the district being planned. And $\pi_{(x)}^{r_{(1)}} = (\pi_{(x),(1)}^{r_{(1)}}, \pi_{(x),(2)}^{r_{(1)}}, \dots, \pi_{(x),(\alpha)}^{r_{(1)}})$ represent the α step policy of $r_{(1)}$.

$E(\cdot)$ indicates the expectation of the reward which is similar to that of MDP. The aim of designing this objective function is that this can obtain the probabilistic optimal solution since there exists motion uncertainty of robots. λ ($0 < \lambda < 1$) is the discount factor. λ has relationship with p_u as equation (9). The aim of equation (9) is that the discount factor can be adjusted dynamically based on and when is high; the long-term reward is not reliable anymore and the path planning based on MDP is concerned more on the optimization of total short-term reward.

$\mathbf{S}_{(1)}, \dots, \mathbf{S}_{(\alpha)}$ represents the state from the start time step when the current district needs to be planned to the time step which is $(\alpha - 1)$ time steps after the start time step. And for $\mathbf{S}_{(1)}$, it contains states for all the robots in the district, which means that $\mathbf{S}_{(1)} = (s_{(1)}^{r_{(1)}}, s_{(1)}^{r_{(2)}}, \dots, s_{(1)}^{r_{(m)}})$. The state $s_{(1)}^{r_{(1)}}$ indicates the state for the robot $r_{(1)}$ after the action $\pi_{(1),(1)}^{r_{(1)}}$ is taken. $R(\cdot)$ defines the reward for each state based on the distribution of the robots in the corresponding district.

$$\lambda = 1 - p_u. \quad (9)$$

Therefore, according to the Markov decision process in [8], the optimization objective in equation (8) implies the expectation of the rewards for the state from the start time step to after $(\alpha - 1)$ time steps. And it can be rewritten as follows:

$$\min_{\pi_{(x)}} E(R(S_{(1)}) + \lambda \cdot R(S_{(2)}) + \dots + \lambda^{(\alpha-1)} R(S_{(\alpha)})) = \min_{\pi_{(x)}} \sum_{q=1}^{\alpha} \lambda^{(q-1)} \left(\sum_{S'_{(q)}} P_{S_{(q-1)\pi_{(x)},(q-1)}}(S'_{(q)}) \cdot R(S'_{(q)}) \right), \quad (10)$$

where $\mathbf{S}'_{(q)}$ indicates the possible state after the action $\pi_{(x),(q-1)}$ is taken at the state $\mathbf{S}_{(q-1)}$ since there exists the motion uncertainty of robots, $R(\mathbf{S}'_{(q)})$ represents the reward of the robots in this district at the state $\mathbf{S}'_{(q)}$, and $P_{\mathbf{S}_{(q-1)\pi_{(x),(q-1)}}}(\mathbf{S}'_{(q)})$ is the probability that the state of robots in this district would turn to $\mathbf{S}'_{(q)}$ when the action $\pi_{(x),(q-1)}$ is taken at the state $\mathbf{S}_{(q-1)}$. The probability can be calculated as follows:

$$P_{\mathbf{S}_{(q-1)\pi_{(x),(q-1)}}}(\mathbf{S}'_{(q)}) = \prod_{j=1}^m P_{s_{(q-1),\pi_{(x),(q-1)}}^{r_{(j)}}}(s_{(q)}^{r_{(j)}}), \quad (11)$$

where $s_{(q-1)}^{r_{(j)}}$ is the state of the robot $r_{(j)}$ after the action $\pi_{(x),(q-1)}$ is taken at state $s_{(q-1)}^{r_{(j)}}$ and $P_{s_{(q-1),\pi_{(x),(q-1)}}^{r_{(j)}}}(s_{(q)}^{r_{(j)}})$ is the probability that the state of the robot $r_{(j)}$ will change into $s_{(q)}^{r_{(j)}}$ after the action $\pi_{(x),(q-1)}$ is taken at state $s_{(q-1)}^{r_{(j)}}$. And this probability is dependent on the motion uncertainty and can be represented as follows:

$$P_{s_{(q-1),\pi_{(x),(q-1)}}^{r_{(j)}}}(s_{(q)}^{r_{(j)}}) = \begin{cases} p_u, & \pi_{(x),(q-1)}^{r_{(j)}} = \text{move and } r_{(j)} \text{ stays still,} \\ 1 - p_u, & \pi_{(x),(q-1)}^{r_{(j)}} = \text{move and } r_{(j)} \text{ moves,} \\ 1, & \pi_{(x),(q-1)}^{r_{(j)}} = \text{waiting.} \end{cases} \quad (12)$$

For the reward $R(\mathbf{S}'_{(q)})$ of the policy $\pi_{(x)}$, it is related to the crowdedness of the district and the distances among robots in the district, which can be defined as follows:

$$\varepsilon = \begin{cases} \infty, & p_{(i)}(t) = p_{(j)}(t+1), p_{(j)}(t) = p_{(i)}(t+1) \text{ or } p_{(i)}(t) = p_{(j)}(t), \exists i, j \in [1, m], \\ e^{1/(1-p_u)}, & p_{(i)}(t) = p_{(j)}(t), \exists i, j \in [1, m], \\ 1, & \text{otherwise.} \end{cases} \quad (16)$$

It can be seen from equation (16) that when there would be a conflict between robot $r_{(i)}$ and $r_{(j)}$, ε is set at ∞ . When a robot would move to the location at next time step where another robot is staying at for now, then ε is set at $e^{1/(1-p_u)}$. Otherwise, ε is set at 1.

According to equations (13) to (16), the first aim of this design is to prevent that each district would be too crowded since this would cause the coordination space to be too small, which means that once there happens motion uncertainty in some robots, it would be hard for the algorithm to find the solution and the system efficiency would be decreased. Another aim for the set of ε is to prevent the conflicts and make sure that one robot would not choose the action which might cause it to be next to another one. And this is beneficial for the motion coordination of robots since the motion uncertainty of one would cause it to still stay at the previous location, which increases the probability of the conflicts. Therefore, this setting can decrease the probability of the situation like that.

To solve the optimization problem in equation (8), the policy set π needs to be generated in the district and the

$$R(\mathbf{S}'_{(q)}) = Cd'_{(x)} \cdot \left(\varepsilon \cdot \sum_{h=1}^m \text{value}(s_{(q)}^{r_{(h)}}) \right), \quad (13)$$

where $Cd'_{(x)}$ is the normalized crowdedness of the district being planned when the policy $\pi_{(x)}$ is taken. And the crowdedness $Cd_{(x)}$ of the district being planned can be calculated as follows:

$$Cd_{(x)} = \frac{m}{\sum_{i=1}^m \sum_{j=1}^m \text{distance}_{(i)(j)}}, \quad (14)$$

where $\text{distance}_{(i)(j)}$ is the distance between $r_{(i)}$ and $r_{(j)}$ at state $\mathbf{S}'_{(q)}$. Then, the crowdedness $Cd_{(x)}$ needs to be normalized by using the principle of the softmax according to the following formula:

$$Cd'_{(x)} = \frac{e^{Cd_{(x)}}}{\sum_{h=1}^k e^{Cd_{(h)}}}. \quad (15)$$

In equation (13), $\text{value}(s_{(q)}^{r_{(h)}})$ indicates the value of the robot $r_{(h)}$ at state $\mathbf{S}'_{(q)}$. It is defined as the distance between the robot $r_{(h)}$ and the next district which $r_{(h)}$ heads to since the passing order has been planned in the topology level. The rule for calculating this distance is based on the Manhattan distance.

To avoid the conflicts among multiple robots, the factor ε is introduced. And ε in equation (13) is defined as follows:

policy with the minimum objective would be selected as the solution. And the principle for the candidate action set of the policies is that robots would not be far away from the next district after taking this action. There are two benefits for this principle. Firstly, the farther the distance from the next district, the larger the objective function in equation (8). Secondly, due to the introduction of this principle, the candidate actions will decrease, which can in turn lower the computational complexity. According to above principle, there are two kinds of candidate action sets when robot is in different locations of each district:

- (1) When robot is in any location of district and not coming into the boundary, the candidate action set of robot is denoted as {left, up, still}. The action left means that robot drives to the other side of two-way street at next time step, up means that robot follows current side of street, and still means that robot stays still at next time step. For example, in Figure 5, robots $r_{(1)}$ and $r_{(3)}$ are in the district and not coming into boundary at next time step. Therefore, their

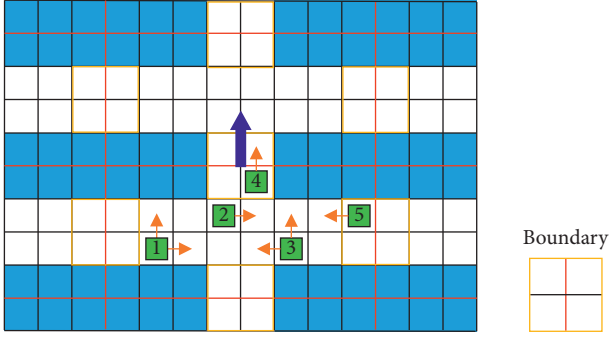


FIGURE 5: Illustration of the candidate actions. Green squares represent the robots in the district. The blue arrow indicates the next district where these robots are heading to. Therefore, the orange arrows for each robot represent the candidate actions they can take. Besides the actions represented by these orange arrows, staying still is also included in the candidate actions for each time step.

candidate action set includes left, up, and *still* as orange arrows show. And as shown in Figure 5, the boundary between two districts is defined as an area composed of four grids, and one half of the boundary belongs to one district and the other belongs to the neighboring district. Considering that path planning in the route level is conducted separately in each district, robot should only follow one direction (drive to the right) in the process of traversing the boundary to the neighboring district, for avoiding the conflict.

- (2) When robot is in the boundary of district or coming into the boundary, the candidate action set of robot is denoted as $\{\text{up, still}\}$ or $\{\text{left, still}\}$ according to the rule that robot should only drive to the right in the process of traversing the boundary. For example, robot $r_{(2)}$ in Figure 5 is coming into the boundary at next time step and its candidate action set is $\{\text{left, still}\}$ which means that it needs to drive to the right side of two-way street firstly. $r_{(4)}$ and $r_{(5)}$ are in the boundary and their candidate action set is $\{\text{up, still}\}$.

According to the above candidate action set, the generation process of robot $r_{(i)}$ and policy $\pi_{(q)}^{r_{(i)}}$ is shown in Figure 6. And it can obtain multiple policies for each robot. Then, π can be generated with the combination of multiple policies of each robot in the district.

4. Discussion and Analysis

4.1. Conflict Avoidance. In this paper, we propose a bi-level path planning algorithm. The architecture of the proposed

method is divided into topology level and route level. And in the topology level, the passing order of districts is planned for robots. There is no conflict in topology level because the districts are allowed to traverse for multiple robots at the same time. Therefore, the conflict avoidance of the proposed algorithm in the paper mainly depends on whether or not it could plan conflict-free path in route level. And we try to prove it in the following proposition.

Proposition 1. *The path planning in route level could plan conflict-free paths for all robots in the district.*

Proof. In route level of the proposed method, the path for robots in the district is planned for many times and α steps with MDP at each time. Therefore, we try to use mathematical induction to prove the conflict-free path can always be planned by the proposed method in the district at each time. And we firstly describe some symbols of some variables to facilitate the expression of proof.

For any district $d_{(s)} \in D$ with m robots, the route map can be denoted as $G_p(V_p, E_p)$. And robots in the district can be denoted as $R_p = \{r_{(1)}, \dots, r_{(m)}\}$. n_{re} represents the n_{re} th path planning in each district. In the n_{re} th planning: the beginning of time step is denoted as $t_{n_{re}}$, and the planning step is α . The solution with MDP is $\pi_{n_{re}}$ and planned path of $\pi_{n_{re}}$ for any robot $r_{(i)}$ is $[p_{(i)}(t_{n_{re}}), p_{(i)}(t_{n_{re}} + 1), \dots, p_{(i)}(t_{n_{re}} + \alpha)]$. The proof detail is as follows:

- (a) Firstly, when $n_{re} = 1$, prove that the path planning is conflict-free:

At the first time of path planning, robots are leaving the boundary in each district. These robots of each district are heading to different directions and located in the different sides of two-way street in the map. Therefore, there always exists one solution π_1 that makes sure that all robots go on with their current side of two-way street in the α time steps and no robots from the opposite direction run in the same side of two-way street. And the planned paths for robots meet equation (17). It can be concluded that reward of the above solution π_1 according to equation (16) will not be ∞ and π_1 could be feasible solution of the optimization objective, which could prove that the proposed method could plan conflict-free path for all robots when $n_{re} = 1$.

$$\begin{cases} p_{(i)}(t) \neq p_{(j)}(t+1) \text{ and } p_{(j)}(t) \neq p_{(i)}(t+1), & \forall i, j \in R_p, t_{n_{re}} \leq t \leq t_{n_{re}+1}, \\ p_{(i)}(t) \neq p_{(j)}(t), & \forall i, j \in R_p, t_{n_{re}} \leq t \leq t_{n_{re}+1}. \end{cases} \quad (17)$$

- (b) Secondly, when $n_{re} = k_1$, suppose that the planned path in the k_1 th planning is conflict-free and prove

the planned path in the $(k_1 + 1)$ th planning is also conflict-free:

Considering that the path planning in the k_1 th planning is conflict-free, the ending location $p_{(i)}(t_{k_1} + \alpha)$ of planned path for any robot $r_{(i)} \in R_p$ satisfies equation (17), which means that there does not exist the situation of direction conflict and location. On the other hand, the ending location $p_{(i)}(t_{k_1} + \alpha)$ of the k_1 th path planning is equivalent to the beginning location $p_{(i)}(t_{k_1+1})$ of the $(k_1 + 1)$ th path planning. If there exists $v1_i, v2_i \in R_p$ where $v1_i$ represents the next location from with the action of going up and $v2_i$ represents the next location from $p_{(i)}(t_{k_1+1})$ with the action of going left, equation (18)

$$\begin{cases} p_{(h)}(t_{n_{re}}) \neq v2_i, & \forall h \in R_p \text{ and } h \neq i \text{ if } p_{(j)}(t_{n_{re}}) = v1_i, \exists j \in R_p \text{ and } j \neq i, \\ p_{(h)}(t_{n_{re}}) \neq v1_i, & \forall h \in R_p \text{ and } h \neq i \text{ if } p_{(j)}(t_{n_{re}}) = v2_i, \exists j \in R_p \text{ and } j \neq i, \end{cases} \quad (18)$$

and when robots go to the boundary in the last time planning, considering that the robots always follow one direction when they are coming into boundary or traversing on the boundary, the district-distributed way of the proposed method in route level will not have conflict between robots from different districts.

To summarize, there always would be conflict-free solution for path planning in the route level with the proposed method. \square

4.2. Optimization. For a multirobot system with each robot subject to uncertainty, directly computing the optimal solution is very difficult and it must recompute many times during the running of robots. Therefore, it is hard to directly prove the optimization or near optimization. In this section, we will discuss the optimization of the proposed method. Considering that the architecture of proposed method in this paper is divided into two levels and paths are planned separately, we discuss the optimization of proposed method from following two parts:

- (1) For path planning in topology level: in the topology level, the optimal orders of districts for all robots can be obtained and the reason is as follows. Firstly, K shortest path planning algorithm [24] is adopted to obtain the multiple orders of passing districts for each robot. And combinations of orders of passing districts among robots can be generated. When K is large enough, all possible combinations of orders of passing districts can be obtained. Secondly, equation

can be concluded. It means that each robot $r_{(i)}$ can go left or up without conflict with other robots. Therefore, it can construct a candidate solution π_{k_1} that makes robots heading to the same direction run in the same side of two-way streets in the α steps of path. And when robots heading to the different direction run in the same side of two-way street, the planned path will not have conflict with other robots. Therefore, it can be proved that the planned path in the $(k_1 + 1)$ th planning is conflict-free.

(3) is put up with in the topology level, which aims to minimize the corresponding distance between the center of the two districts and total crowdedness between districts among all the robots assigned with new tasks. And above optimization objective can be calculated based on all possible combinations of orders of passing of districts for robots.

- (2) For path planning in route level: the proposed method takes path planning for all robots in each district for many times and plans paths of α steps for the robots at each time until all robots leave the district. So, it is difficult to directly prove the optimization. In this part, we compare above method with existing method in [8] to discuss the optimization of the proposed method. Similar to the proposed method, the existing method also takes path planning in each district for many times. The difference is that at each time planning, the existing method plans the path of static α_{global} steps with the MDP method and executes the first α steps of them. The optimization objective is shown in equation (19). α_{global} can make sure that the destination of planned path is the location where robots are leaving from the current district. In the existing method in [8], α is also intuitively related to p_u following equation (7). The existing method in [8] has proved the optimization.

$$\min_{\pi_{(x)}} \sum_{q=1}^{\alpha} \lambda^{(q-1)} \left(\sum_{S'_{(q)}} P_{S_{(q-1)\pi_{(x)},(q-1)}}(S'_{(q)}) \cdot R(S'_{(q)}) \right) + \sum_{q=1}^{\alpha_{\text{global}} - \alpha} \lambda^{(\alpha+q-1)} \left(\sum_{S'_{(q)}} P_{S_{(q-1)\pi_{(x)},(q-1)}}(S'_{(q)}) \cdot R(S'_{(q)}) \right). \quad (19)$$

For discussing the optimization of proposed method, we try to discuss the similarity of first α steps for solution at each

time planning compared with existing method in [8] in the following two situations:

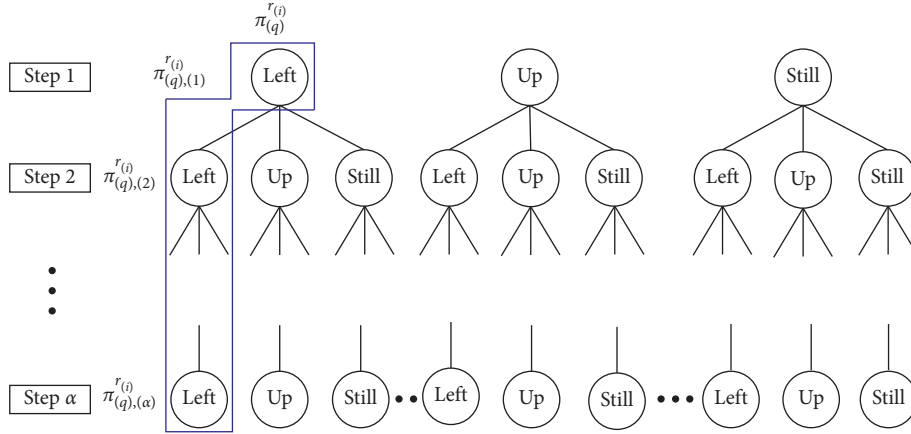


FIGURE 6: Generation of policy $\pi_{(q)}^{r(i)}$ for $r_{(i)}$ in each district. Assuming that the candidate actions in each district include left, up, and still, the generation of policy $\pi_{(q)}^{r(i)}$ for robot $r_{(i)}$ is shown in this figure. In this tree structure, different level of branches corresponds to different steps. Each circle represents the action for $r_{(i)}$ in the district at the corresponding time step. For example, the action left means that all the robots in the district should go left. A complete policy $\pi_{(q)}^{r(i)}$ from the first time step to the α th time step is illustrated in the blue box. Multiple policies for each robot can be generated from this tree structure.

- (a) When the motion certainty probability p_u is low ($p_u \leq 0.2$):

In this situation, α is close to α_{global} according to (7). And it is obvious that the first α steps of solution from the proposed method and existing method in [6] are close to the same. For example, Figure 7 shows a situation where three robots are located at different states, and Table 1 shows the different solutions of path planning for the situation in Figure 7 using above two methods when p_u is different (α_{global} is set to 10 for the existing method in [8]). As shown in Table 1, when p_u is 0.1 or 0.2, first α steps of the proposed method are the same as the existing method in [6].

- (b) When the motion certainty probability p_u is high ($p_u > 2$):

In this situation, consider that both the proposed method and existing method in [8] choose the MDP method where the discount factor λ exists in optimization objective and has relationship with p_u according to equation (9). Therefore, when p_u is high, λ^α in (19) will be lower. This means the last $(\alpha_{global} - \alpha)$ steps of solution have little influence on the computation for equation (19). And in this situation, the first α steps of the proposed method are close to the existing method in [6]. For example, in Table 1, when α is close to α_{global} according to (19), it is obvious that the solution from the proposed method is close to first α steps of solution from the existing method in [8]. Figure 7 shows a situation where three robots are located at different states, and Table 1 shows the different solutions of above two methods when p_u is different. As shown in Table 1, in the example of Figure 7, when p_u is 0.4 or 0.5, although the solution of above two methods is not exactly the same, the solution from the proposed method is

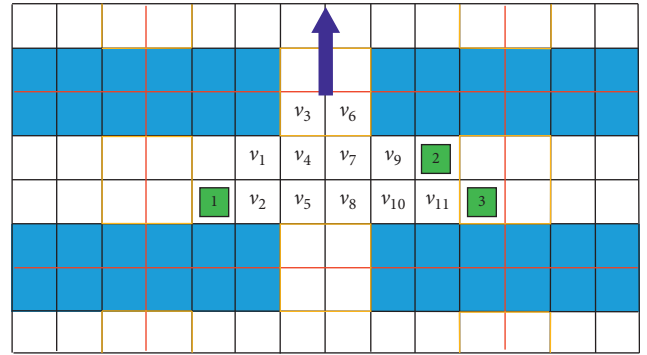


FIGURE 7: Illustration of the example of trying to plan path for three robots in the district. The original location of robots is shown in the figure. And the blue arrow indicates the next district where these three robots are heading to. $v_1 \sim v_{11}$ represent different locations in the district and v_6 is the destination where robots are leaving from the current district.

close to the first α steps of solution from the existing method in [8].

5. Simulation Result and Analysis

In order to evaluate the proposed method, simulations were implemented under the Matlab environment with Intel 3.60 GHz Core i7-4790U CPU and 8G RAM. The map of warehouse is shown in Figure 1.

The motion uncertainty p_u ranges from 0 to $(1/2)$ to simulate the situations under different motion modes. The number of robots is set at 10, 20, 30, 40, and 50. The number of tasks is 1000.

The makespan, number of planning steps, and computation time are contrasted with those of the existing method in [21, 22] under a different number of robots. And for the existing method in [21, 22], the probability corresponding to the disturbance intensity whether during the following second the robot will be prevented from moving is set to p_u and the

TABLE 1: The solution of path planning for three robots which are in different locations as shown in Figure 7.

	The solution of the proposed method	The first α -step solution of the existing method in [8]
$p_u = 0.1$ $\alpha = 10$ $\lambda = 0.9$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{left}), (v_4, \text{up})$ $(v_7, \text{left}), (v_6, \text{left})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left}), (v_6, \text{left})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left}), (v_8, \text{still})$ $(v_8, \text{left}), (v_7, \text{left}), (v_6, \text{left})$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{left}), (v_4, \text{up})$ $(v_7, \text{left}), (v_6, \text{left})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left}), (v_6, \text{left})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left}), (v_8, \text{still})$ $(v_8, \text{left}), (v_7, \text{left}), (v_6, \text{left})$
$p_u = 0.2$ $\alpha = 5$ $\lambda = 0.8$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{left}), (v_4, \text{up})$ (v_7, left) $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left}), (v_6, \text{up})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left}), (v_8, \text{still})$ (v_8, left)	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{left}), (v_4, \text{up}), (v_7, \text{left})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left}), (v_6, \text{up})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left}), (v_8, \text{still})$ (v_8, left)
$p_u = 0.3$ $\alpha = 3$ $\lambda = 0.7$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{up})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left})$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{up})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left})$
$p_u = 0.4$ $\alpha = 3$ $\lambda = 0.6$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{up})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left})$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up}), (v_5, \text{left})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up}), (v_7, \text{left})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up}), (v_{10}, \text{left})$
$p_u = 0.5$ $\alpha = 2$ $\lambda = 0.5$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up})$	$r_{(1)}: (v_0, \text{up}), (v_2, \text{up})$ $r_{(2)}: (v_{12}, \text{up}), (v_9, \text{up})$ $r_{(3)}: (v_{13}, \text{up}), (v_{11}, \text{up})$

lower bound which represents the best possible travel time is set to the shortest distance from start to end for each robot. The average of 20 repeated simulations is shown in Tables 2–4. The tasks have been generated before simulations.

5.1. Makespan. Table 2 shows the simulation results of average makespan of the existing method and the proposed method. Standard deviations are also shown in Table 2. Decrease percentage of makespan (the last column in Table 2) is calculated according to the following formula:

$$\text{decrease percentage} = \frac{\text{makespan}_1 - \text{makespan}_2}{\text{makespan}_1} \times 100\%, \quad (20)$$

where makespan_1 and makespan_2 represent the makespan of the existing method in [21, 22] and the proposed method, respectively. Graphical summaries of simulation results of makespan are shown in Figure 8.

5.1.1. Average Time. Compared to the existing method in [21, 22], it can be seen that the proposed method in this paper effectively decreases the makespan. When the number of robots is relatively small, the decrease percentage is low because the robots are located sparsely in the warehouse environment and the motion uncertainties of robots would not influence too many other normal robots, which means that most of the normal robots could still follow the previous planned paths. As the number of robots increases, the decrease percentage of makespan is greatly improved because the number of influenced robots would increase further, which means that the proposed method would replan to find more optimal paths based on the probability p_u for more robots to avoid robots which are in the motion uncertainty. Under this circumstance, by combining MDP, redundant waiting of the method in [21, 22] is not necessary. When the value of p_u increases, the

proposed method could still improve the system efficiency and has a smaller variance compared to the existing method in [21, 22]. The percentage of decrease can be up to 27.9% when the number of robots is 50 and p_u is 0.5.

5.1.2. Variance. Compared to the existing method in [21, 22], the variance of the proposed method is smaller. Furthermore, with the increase of the number of robots, the variance in the results of the proposed method increases less than that of the existing method in [21, 22]. The comparison between the results of the proposed method and the existing method in [21, 22] means that the proposed method has less randomness and is more stable. The reason is that the method in this paper would optimize paths of all the robots depending on the probability p_u .

In the proposed method, unlike the existing method in [21, 22] that would solve the problem of motion uncertainty through a waiting strategy, the proposed method to coordinate paths of robots under motion uncertainties adopts the strategy of replanning new paths to choose the probabilistic optimal solution. Therefore, when motion uncertainty happens, all the normal robots would follow the paths that could reach the destination as soon as possible from the global view instead of adopting the strategy of going forward and waiting. Therefore, compared to the existing methods in [21, 22], the decrease of the efficiency induced by the motion uncertainties could be lowered. In this way, the efficiency of the whole system could be improved when motion uncertainties happen by adopting the proposed method. Compared to the existing methods, the efficiency could be increased, which is beneficial for the overall running of the system.

5.2. Number of Planning Steps. Percentage of planning (listed in Table 2) is calculated according to the following formula:

TABLE 2: Simulation results in a warehouse environment of 50×34 4-neighbor grids: makespan (standard deviation).

p_u	Robot number	Existing method in [21, 22]	Proposed method	Decrease percentage (%)
0.1	10	19794.5 (272.4)	19022.4 (187.1)	3.9
	20	13582.6 (285.6)	12210.8 (193.5)	10.1
	30	11394.5 (294.2)	9297.9 (203.8)	18.4
	40	9019.3 (314.8)	7044.1 (209.4)	21.9
	50	7768.4 (332.9)	5818.5 (216.7)	25.1
0.2	10	22956.7 (280.4)	22061.4 (193.5)	3.9
	20	15636.6 (290.1)	14041.7 (199.4)	10.2
	30	13103.2 (301.5)	10666.1 (204.5)	18.6
	40	9019.3 (317.6)	8702.4 (210.4)	22.2
	50	9587.2 (334.9)	7123.3 (220.7)	25.7
0.3	10	25895.6 (287.5)	24859.8 (197.2)	4.0
	20	16879.3 (298.4)	15140.7 (200.1)	10.4
	30	14374.5 (309.7)	11657.7 (208.7)	18.9
	40	12287.3 (338.6)	9510.4 (218.5)	22.6
	50	10625.4 (344.8)	7820.4 (229.5)	26.4
0.4	10	28190.4 (294.1)	27034.6 (196.5)	4.1
	20	18724.5 (302.6)	16777.2 (200.4)	10.4
	30	16394.6 (315.3)	13246.8 (206.8)	19.2
	40	14274.8 (334.7)	10991.6 (218.5)	23.0
	50	12463.4 (354.7)	9085.8 (230.5)	27.1
0.5	10	31852.4 (300.6)	30514.6 (199.2)	4.2
	20	19952.6 (310.9)	17857.6 (201.5)	10.5
	30	17365.4 (325.6)	13979.1 (208.7)	19.5
	40	15284.7 (344.8)	11692.8 (221.8)	23.5
	50	13561.2 (356.3)	9777.6 (231.5)	27.9

$$\text{planning percentage} = \frac{N_{re}}{\text{makespan}} \times 100\%, \quad (21)$$

where N_{re} and makespan represent the number of planning and the makespan of corresponding method, respectively. The makespan of each method is given in Table 2. Graphical summaries of simulation results of planning percentage are shown in Figure 9. It can be seen from Table 3 that the percentage of planning of the proposed method is much more smaller than that of the method in [21, 22]. It has to plan at each time step since the method of [21, 22] is a kind of reactive method. Therefore, the percentage is 100%. On the contrary, the proposed method in the route level would only plan when robots have finished the path of planning step α or when the motion uncertainty happens. Therefore, the percentage of planning would decrease, compared with the existing methods in [21, 22].

Another conclusion we can get from Table 3 is that when the probability p_u is small, the percentage reduction compared with the method in [21, 22] is larger and when the probability p_u increases, the percentage reduction becomes smaller. The reason for this is that when p_u is small, there would not be too many time steps when motion uncertainties happen, which means that there would not be too many time steps when planning is needed. Therefore, the percentage of planning is relatively small, which causes the percentage reduction to be larger. When p_u becomes larger, the time steps when motion uncertainties happen become more, which causes more planning. Therefore, the percentage of planning increases, which causes the percentage reduction to be smaller. However, it is noteworthy that even

if the percentage of planning becomes larger when p_u increases, the percentage of the proposed method is still fewer than that of the method in [21, 22].

5.3. Computation Time. Table 4 shows the average of the computation time under different p_u . The statistics in the table demonstrate that the proposed method could satisfy the requirement of the real-time performance. The reason that the proposed method is higher than that of the existing method in [21, 22] is that the existing method only plans for the next time step while the proposed method needs to plan the whole path to the destination. Therefore, the computation of the proposed method would be larger than that of the existing method. However, as shown in Figure 10, the trend of computation time is more close to linear relationship for the proposed method and increasing speed will reduce slowly with robot number increasing. On the other hand, even if the computation time is somewhat larger, the proposed method could still plan conflict-free paths for all the robots in real time, which can be applied for the lifelong scheduling in warehouses.

The real-time performance of the bi-level probabilistic path planning algorithm for multiple robots with motion uncertainty is guaranteed by the bi-level planning architecture and the distributed planning in each district. The topology level would plan the passing orders of districts for all robots assigned with tasks and the path planning is conducted in each district for route level. The planning in each district only needs to take the actions of the current district being planned into consideration. Therefore, the

TABLE 3: Simulation results in a warehouse environment of 50×34 4-neighbor grids: number of planning (standard deviation).

p_u	Robot number	Existing method in [21, 22]	Planning percentage (%)	Proposed method	Planning percentage (%)
0.1	10	19794.3 (272.4)	100	2587.0 (102.4)	13.6
	20	13582.6 (285.6)	100	1721.7 (104.1)	14.1
	30	11394.5 (294.2)	100	1348.2 (107.3)	14.5
	40	9019.3 (314.8)	100	1056.6 (109.9)	15.0
	50	7768.4 (332.9)	100	907.7 (112.6)	15.6
0.2	10	22956.7 (280.4)	100	5405.0 (104.3)	24.5
	20	15636.6 (290.1)	100	3510.4 (106.4)	25.0
	30	13103.2 (301.5)	100	2815.9 (109.6)	26.4
	40	11185.6 (317.6)	100	2427.9 (112.1)	27.9
	50	9587.2 (334.9)	100	2108.5 (115.9)	29.6
0.3	10	25895.6 (287.5)	100	8825.2 (106.5)	35.5
	20	16879.3 (298.4)	100	5708.1 (108.5)	37.7
	30	14374.5 (309.7)	100	4558.2 (110.8)	39.1
	40	12287.3 (328.6)	100	3937.3 (113.4)	41.4
	50	10625.6 (344.8)	100	3456.6 (116.5)	44.2
0.4	10	28190.4 (294.1)	100	12544.1 (108.6)	46.4
	20	18724.5 (302.6)	100	8170.5 (110.7)	48.7
	30	16394.6 (315.3)	100	6861.8 (113.6)	51.8
	40	14274.8 (334.7)	100	6034.4 (116.8)	54.9
	50	12463.4 (354.7)	100	5324.3 (119.9)	58.6
0.5	10	31852.4 (300.6)	100	17728.9 (110.3)	58.1
	20	19952.6 (310.9)	100	10750.3 (112.9)	60.2
	30	17365.4 (325.6)	100	8764.9 (115.6)	62.7
	40	15284.7 (344.8)	100	7647.1 (118.4)	65.4
	50	13561.2 (356.3)	100	6736.8 (122.4)	68.9

TABLE 4: Simulation results in a warehouse environment of 50×34 4-neighbor grids: computation time (standard deviation).

p_u	Robot number	Existing method in [21, 22]	Proposed method
0.1	10	0.0015s (0.000154)	0.0083s (0.000356)
	20	0.0021s (0.000202)	0.1287s (0.000391)
	30	0.0026s (0.000283)	0.2176s (0.000514)
	40	0.0033s (0.000362)	0.3025s (0.000698)
	50	0.0041s (0.000451)	0.3786s (0.000824)
0.2	10	0.0016s (0.000156)	0.0041s (0.000322)
	20	0.0022s (0.000209)	0.095s (0.000371)
	30	0.0027s (0.000291)	0.1756s (0.000432)
	40	0.0034s (0.000369)	0.2476s (0.000599)
	50	0.0042s (0.000468)	0.3012s (0.000774)
0.3	10	0.0017s (0.000162)	0.0029s (0.000316)
	20	0.0023s (0.000218)	0.0523s (0.000354)
	30	0.0028s (0.000401)	0.1026s (0.000407)
	40	0.0035s (0.000379)	0.1424s (0.000558)
	50	0.0043s (0.000481)	0.1672s (0.000732)
0.4	10	0.0019s (0.000171)	0.0026s (0.000301)
	20	0.0025s (0.000226)	0.0278s (0.000334)
	30	0.0030s (0.000311)	0.0478s (0.000386)
	40	0.0037s (0.000397)	0.0623s (0.000514)
	50	0.0045s (0.000492)	0.0745s (0.000702)
0.5	10	0.0021s (0.000178)	0.0021s (0.000289)
	20	0.0027s (0.000235)	0.0312s (0.000318)
	30	0.0032s (0.000323)	0.0558s (0.000371)
	40	0.0039s (0.000398)	0.0797s (0.000501)
	50	0.0047s (0.000506)	0.0976s (0.000688)

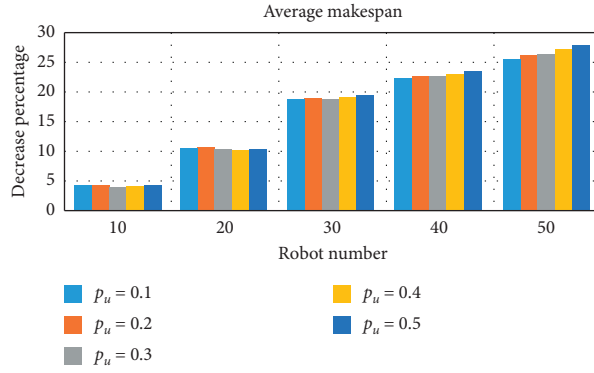


FIGURE 8: Decrease percentage of average makespan in a warehouse environment of 50×34 4-neighbor grids.

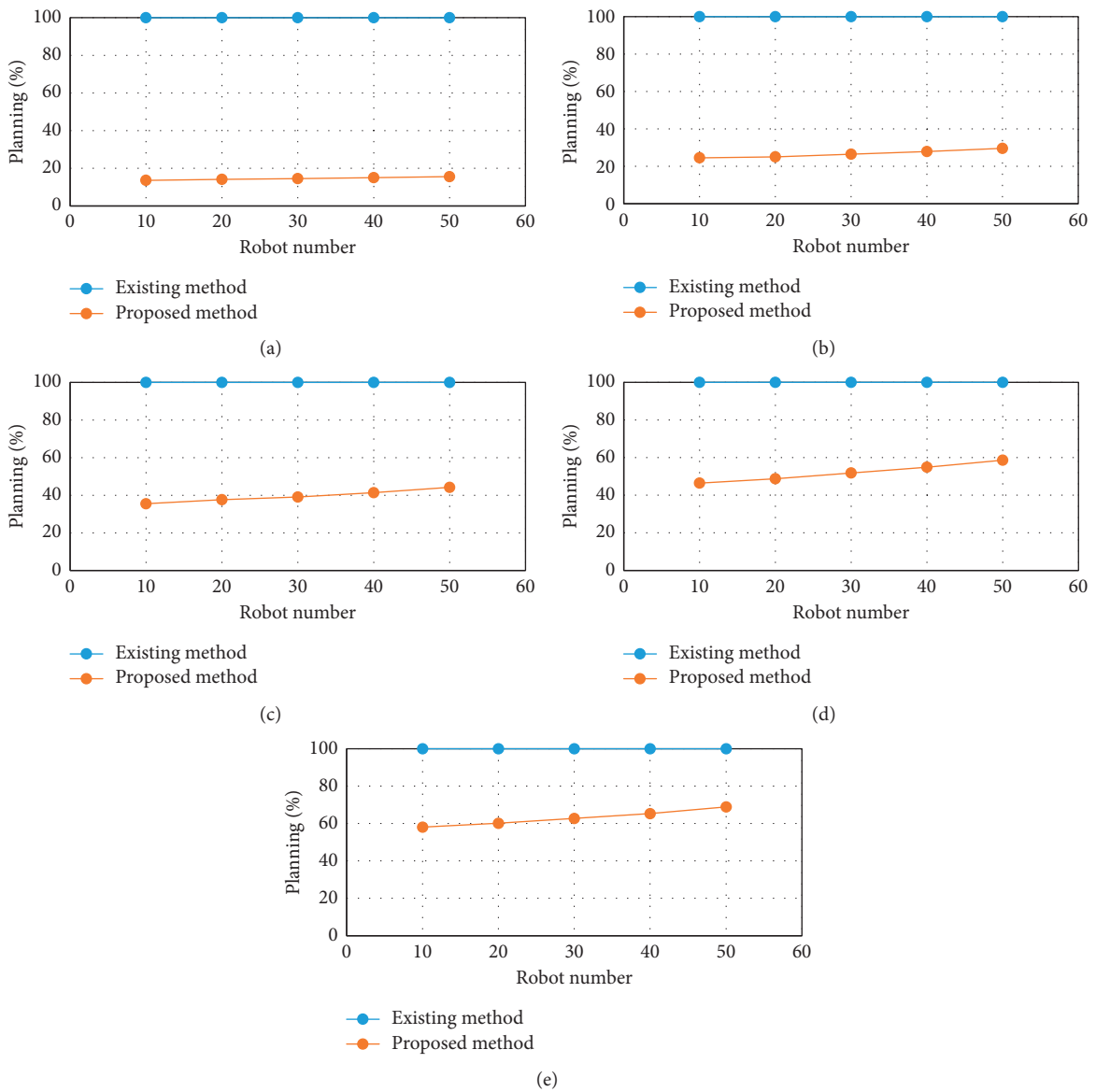


FIGURE 9: Planning percentage in a warehouse environment of 50×34 4-neighbor grids. (a) $p_u = 0.1$. (b) $p_u = 0.2$. (c) $p_u = 0.3$. (d) $p_u = 0.4$. (e) $p_u = 0.5$.

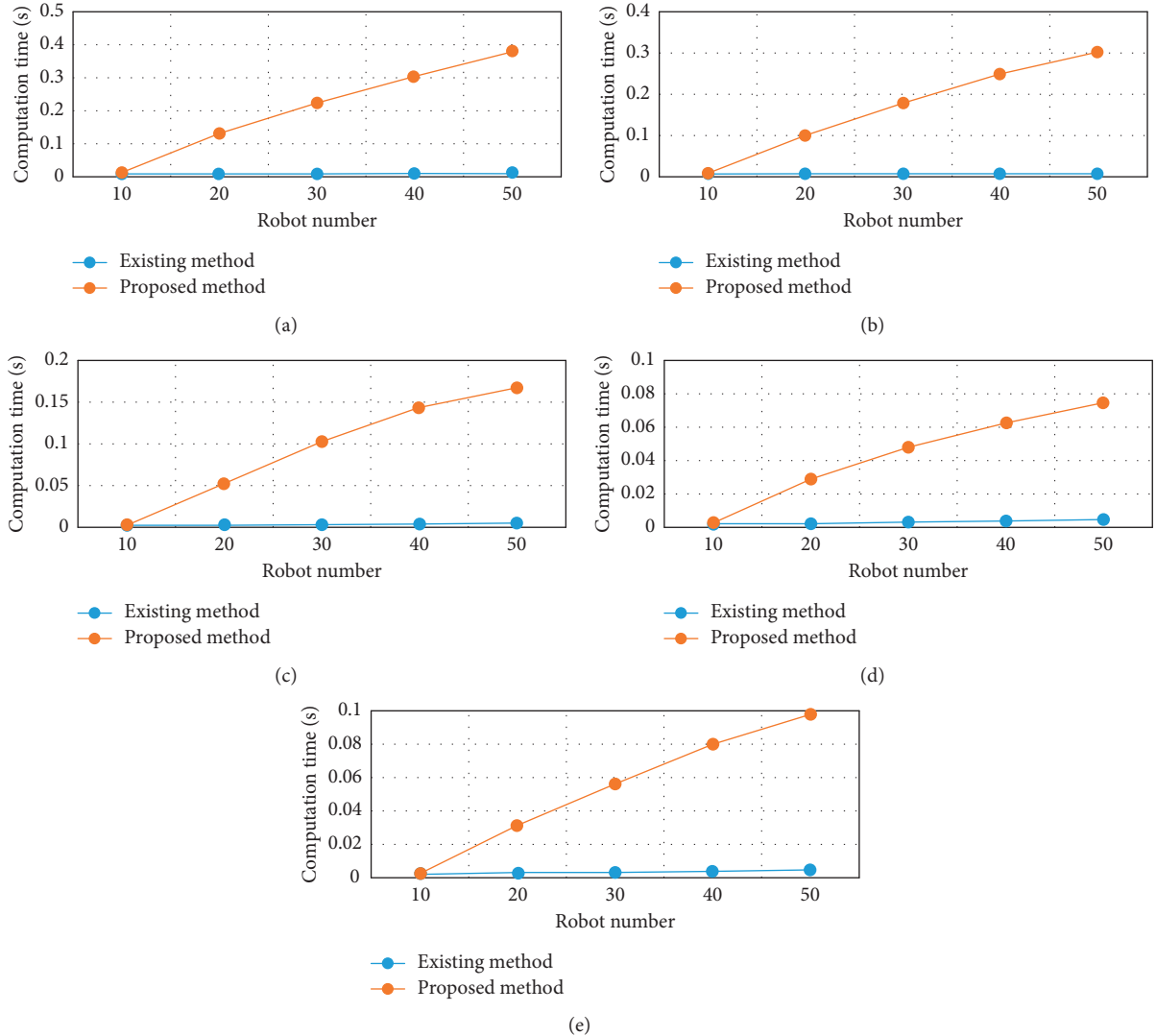


FIGURE 10: Computation time in a warehouse environment of 50×34 4-neighbor grids. (a) $p_u = 0.1$. (b) $p_u = 0.2$. (c) $p_u = 0.3$. (d) $p_u = 0.4$. (e) $p_u = 0.5$.

number of planning depends on p_u , which means that the computation is adjusted by p_u .

6. Conclusion

For improving the system efficiency when there are motion uncertainties among robots in the warehouse environment, this paper proposes a bi-level probabilistic path planning algorithm. In the proposed algorithm, the map is partitioned into multiple interconnected districts and the architecture of proposed algorithm is composed of topology level and route level generating from above map: in the topology level, the district crowdedness is taken into consideration to optimize the passing orders of districts and achieve the district equilibrium; in the route level, path planning is conducted in each district with a MDP-based probabilistic method, where the planning step is related to the probability of the motion uncertainty to improve the efficiency and guaranteeing the real-time performance.

Furthermore, the conflict avoidance of designed algorithm is proved and optimization is discussed. The simulation results validate that the designed algorithm can improve the system efficiency and the percentage can be up to 27.9% in the designed warehouses environment of this paper, which also accords with optimization discussion. It can be also seen that the number of planning has been reduced to a large extent.

Our future work will involve the further optimization of paths by taking the interaction of flow among different districts into consideration since the actions are only obtained by the distributed planning in each district. Another aspect is that the real experiments are needed to verify the effectiveness of the proposed method.

Data Availability

The source code and simulation data of the proposed method and compared method in [18, 19] are available

from the corresponding author upon request. They can also be found at https://github.com/SourceCode2020/Bi-level_Probabilistic_Path_Planning_Algorithm.

Conflicts of Interest

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

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant no. 61773261.

References

- [1] H. Ma, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with delay probabilities," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3605–3612, San Francisco, CA, USA, February 2017.
- [2] G. Li, H. P. Hildre, and H. Zhang, "Toward time-optimal trajectory planning for autonomous ship maneuvering in close-range encounters," *IEEE Journal of Oceanic Engineering*, pp. 1–16, 2019.
- [3] M. Gombolay, R. Wilcox, and J. Shah, "Fast scheduling of multi-robot teams with temporospatial constraints," in *Proceedings of the Robotics: Science and Systems Conference*, Berlin, Germany, June 2013.
- [4] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 804–818, 2011.
- [5] S. Chakravorty and S. Kumar, "Generalized sampling-based motion planners," *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 45, no. 4, pp. 647–662, 2015.
- [6] S. Kumar and S. Chakravorty, "Multi-agent generalized probabilistic RoadMaps: MAGPRM," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3747–3753, Vilamoura, Portugal, October 2012.
- [7] A. W. ter Mors, "Conflict-free route planning in dynamic environments," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2166–2171, San Francisco, CA, USA, September 2011.
- [8] L. Liu and N. Michael, "An MDP-based approximation method for goal constrained multi-MAV planning under action uncertainty," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 56–62, Stockholm, Sweden, May 2006.
- [9] H. Ma, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with delay probabilities," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.
- [10] J. Scharpf, D. M. Roijers, F. A. Oliehoek, M. T. J. Spaan, and M. M. de Weerd, "Solving transition-independent multi-agent MDPs with sparse interactions," in *Proc. 30th AAAI Conf. Artif. Intell. (AAAI), May.*, pp. 3174–3180, 2016.
- [11] Y. Hu, Y. Gao, and B. An, "Multiagent reinforcement learning with unshared value functions," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 647–662, 2015.
- [12] C. An, G. Konidaris, A. Anders, G. Cruz, J. P. How, and L. P. Kaelbling, "Policy search for multi-robot coordination under uncertainty," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1760–1778, 2016.
- [13] C. Amato, G. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling, "Planning for decentralized control of multiple robots under uncertainty," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1241–1248, Seattle, WA, USA, May. 2015.
- [14] F. L. Da Silva, R. Glatt, and A. H. R. Costa, "MOO-MDP: an object-oriented representation for cooperative multiagent reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 567–579, 2019.
- [15] L. Bu, R. Babu, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.
- [16] M. L. Koga, V. Freire, and A. H. R. Costa, "Stochastic abstract policies: generalizing knowledge to improve reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 77–88, 2015.
- [17] F. L. Silva, R. Glatt, and A. H. R. Costa, "Simultaneously learning and advising in multiagent reinforcement learning," in *Proceedings of the Sixteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1100–1108, Sao Paulo, Brazil, May 2017.
- [18] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, 2012.
- [19] M. Geist and O. Pietquin, "Algorithmic survey of parametric value function approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 845–867, 2013.
- [20] F. L. Silva and A. H. R. Costa, "Accelerating multiagent reinforcement learning through transfer learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 5034–5035, San Francisco, CA, USA, February 2017.
- [21] M. Čáp, J. Gregoire, and E. Frazzoli, "Provably safe and deadlock-free execution of multi-robot plans under delaying disturbances," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5113–5118, Daejeon, South Korea, October 2016.
- [22] J. Gregoire, M. Čáp, and E. Frazzoli, "Locally-optimal multi-robot navigation under delaying disturbances using homotopy constraints," *Autonomous Robots*, vol. 42, no. 4, pp. 895–907, 2018.
- [23] J. Gregoire, "Priority-based coordination of mobile robots," 2014, <http://arxiv.org/abs/1410.0879>.
- [24] J. Hershberger, M. Maxel, and S. Suri, "Finding the k shortest simple paths: a new algorithm and its implementation," *ACM Transactions on Algorithms*, vol. 3, no. 4, p. 45, 2007.

Research Article

Distributed Fuzzy Adaptive Control for Heterogeneous Nonlinear Multiagent Systems with Similar Composite Structure

Yongqing Fan , Tiantian Xiao, and Zhen Li

School of Automation, Xi'an Key Laboratory of Advanced Control and Intelligent Process, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Correspondence should be addressed to Yongqing Fan; fanyongqing@xupt.edu.cn

Received 3 April 2020; Accepted 7 May 2020; Published 1 June 2020

Guest Editor: Yan Wu

Copyright © 2020 Yongqing Fan 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 distributed fuzzy adaptive control with similar parameters is constructed for a class of heterogeneous multiagent systems. Unlike many existing works, the dimensions of each multiagent dynamic system are considered to be nonidentical in this paper. Firstly, similar properties for different dimensions of multiagent systems are introduced, and some similar parameters among multiagent systems are also proposed. Secondly, a distributed fuzzy adaptive control on the basis of similar parameters is designed for the consensus of leader-follower multiagent systems. Following the graph theory and Lyapunov stability approach, it is concluded that UUB (uniformly ultimately bounded) of all signals in the closed-loop system can be guaranteed, and the consensus tracking error converges to a small compact zero set. Finally, a simulation example with different dimensions is provided to illustrate the effectiveness of the proposed method.

1. Introduction

Multiagent systems have been widely utilized in various fields such as remedial actions [1, 2], social engineering systems [3], satellites engineering [4], and robots cooperative [5, 6]. More and more researchers inclined to design the fundamental collective controls for multiagent systems to make sure that the consensus or synchronization of leader-follower can be guaranteed, and many excellent controls for linear and nonlinear multiagent systems were proposed in recent years [7–10]. Generally speaking, the main work in designing controls is that all agents in the entire dynamic network must reach an agreement, and the information of each agent only can be shared locally. Unfortunately, in lots of actual engineering systems, uncertain nonlinear components existed such as electrical control systems and mechanical control systems; hence, it is a challenge to project appropriate control with limited information.

Fuzzy logic system (FLS) and neural network (NN) are two universal approximations to compensate uncertain terms in all sorts of complexity fields [11–15], and a great quantity of corresponding research works was derived by

scholars [16–21]. For example, aiming at the high-order multiagent systems with unknown nonlinearities in [22], an observer-based distributed fuzzy adaptive control was designed to deal with the unknown nonlinear functions. For a class of strict feedback form of multiagent systems, a novel event-triggered control was presented for the consensus tracking in [23]. Adaptive NN event-triggered control plan was investigated for the nonstrict feedback multiagent systems with sensor faults and input saturation in [24]. In [25], a fuzzy observer was designed to evaluate the unmeasurable states of nonlinear multiagent systems, and an event-triggered control approach was studied to make the followers synchronize with leader's trajectory. However, these existing results only researched on the identity of each agent, which means that the states of every agent have same dynamical behaviors [22–25]. In order to break this limitation, different dynamic behaviors of each agent that can be called as heterogeneous multiagent systems have been studied [26–28]. For instance, a distributed adaptive fuzzy control combining with the backstepping technique was addressed for a class of second-order heterogeneous multiagent systems in [29]. In [30], the output consensus of multiagent systems was guaranteed by using the devised

fuzzy adaptive control. A robust consensus protocol was designed for essential heterogeneous multiagent systems in [31]. In order to ensure the consensus of heterogeneous multiagent systems, a distributed proportional integral control based on sufficient conditions was derived in [32]. It should be noted that the proposed control schemes in [8, 10, 26–28] were only valid for linear multiagent systems. These abundant research achievements provided well guidance for some new design algorithm controls of heterogeneous multiagent systems. Nevertheless, the dimensions of every agent are completely congruent in these literatures [8, 26–32], and the raised control schemes will be invalidated to settle the consensus or synchronization of multiagent systems with different dimensions. Consequently, it is necessary to exploit other original control approaches to tackle the consensus of multiagent systems with distinct dimensions.

Motivated by the similar properties of large-scale systems in [33–38], the definition of similar nodes was introduced for large-scale composite systems with different dimensions, and some effective controls with similar parameters were addressed. From the viewpoint of mathematics, every agent can be defined as a series of nodes in a network; hence, the character of similar nodes in these excellent research works can be drawn to develop consensus control with similar parameters.

This paper attempts to investigate a novel consensus fuzzy adaptive control for a class of multiagent systems with different dimensions, in which the dimensions of each agent are unequal, and the similar parameters of agents are used for devising consistency control. Compared to recent existing works on the consensus of heterogeneous multiagent systems, the principal contributions are three aspects: first, the dimensions of follower systems are different with the dimensions of leader, and the similar definition among multiagent systems is explored. Second, a distributed fuzzy adaptive control methodology with similar parameters is provided. Last, the control matrix gain can be solved by the condition of proposed linear matrix inequality (LMI).

The remaining parts of this paper are organized as follows. Interaction topology, the property of similar composite structure, and FLS are displayed in Section 2. Section 3 presents the fuzzy adaptive control and stability analysis. A simulation example is given for the consensus of multiagent system with nonidentical dimension in Section 4. Finally, Section 5 summarizes conclusions.

Throughout this paper, the following notations are hired. $R^{m \times n}$ denotes the $n \times n$ dimensional Euclidean space; $\text{diag}\{A_1 \ A_2 \ \dots \ A_N\}$ expresses the block-diagonal matrix with matrices $A_1 \ A_2 \ \dots \ A_N$ on its principal diagonal; the notation $\|\cdot\|$ refers to the vector-2-norm. A^T and A^{-1} denote the transpose matrix and inverse matrix of A , respectively; I_n represents the identity matrix with n appropriate dimensions; and $P > 0$ (< 0) means that P is a positive (negative) definite matrix. The Kronecker product of matrices A and P is symbolized by $A \otimes P$; the maximum and minimum eigenvalues matrix A are denoted corresponding to $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$.

2. Preliminaries and Problem Formulation

2.1. Graph Theory. A directed digraph $G = \{V, E\}$ is utilized to describe the information exchange among each agent, where $V = [v_1 \ v_2 \ \dots \ v_N]$ stands for the nonempty set of nodes for each agent. The edge set E contains an edge (v_j, v_i) which means node v_j is able to transfer the relative state information to node v_i ; then, nodes i and j are called as the neighbors when the edge (v_j, v_i) exists. Let N_i denote the set of neighbors of node i ($i = 1, 2, \dots, N$). The directed digraph G can also be described by an adjacency matrix

$$[\alpha_{ij}], \text{ where } \alpha_{ij} = \begin{cases} 1, & (v_j, v_i) \in E \\ 0, & \text{otherwise} \end{cases}. \text{ In addition, it is assumed that there are no repeated edges and no self-loops, i.e., } \alpha_{ii} = 0. \text{ The Laplacian matrix } L = [l_{ij}] \in R^{N \times N} \text{ is defined as } l_{ij} = \begin{cases} -\alpha_{ij}, & i \neq j \\ \sum_{k \neq i, k=1}^N \alpha_{ik}, & i = j \end{cases}.$$

A digraph is said to have a spanning tree, if there exists a node that is called as the root such that the node has directed paths to all other nodes in the graph. The graph \bar{G} consists of G , node 0 (the leader), and the directed edges from the node 0 to the followers in G , and only a small percentage of the followers can receive the information from the leader. Then, we get the following lemma.

Lemma 1 (see [8]). *Let the matrix $\bar{L} = L + \text{diag}([\alpha_{10} \ \alpha_{20} \ \dots \ \alpha_{N0}])$ is positive definite with $\alpha_{i0} > 0$, the i th agent has access to the leader's state information, whereas $\alpha_{i0} = 0$ if otherwise.*

2.2. Preliminaries and Multiagent System. Consider a group of agent system with a leader and N followers labeled as 0 and $1, 2, \dots, N$, respectively. The dynamics of the leader is described as

$$\dot{x}_0(t) = A_0 x_0(t) + B_0 [u_0(t) + s(x_0, t)], \quad (1)$$

where $A_0(t) \in R^{n_0 \times n_0}$ and $B_0 \in R^{n_0 \times m_0}$ are the system matrix and input matrix of leader, respectively. $x_0(t) \in R^{n_0 \times 1}$ is the state vector of leader. $u_0(t)$ denotes the input vector of leader, and matrix $K_0 \in R^{m_0 \times n_0}$ will be given in the process of control design, which will make $A_0 + B_0 K_0$ be Hurwitz stabilized. $s(x_0, t)$ is defined as an input bounded signal and satisfies $|s(x_0, t)| \leq \bar{s}$ for all $t \geq t_0$, and \bar{s} is a known constant.

The dynamics of the followers are defined as follows:

$$\dot{x}_i(t) = A_i x_i(t) + B_i [u_i(t) + g_i(x_i)], \quad i = 1, 2, \dots, N, \quad (2)$$

where $A_i \in R^{n_i \times n_i}$ and $B_i \in R^{n_i \times m_i}$ denote the system matrix and input matrix in the i th followers system, respectively. $x_i(t) \in R^{n_i \times 1}$ and $u_i(t) \in R^{m_i \times n_i}$ are corresponding to the state vector and input vector of the i th follower, respectively. $g_i(x_i)$ represents the unknown nonlinear function.

Assumption 1 (see [38]). Consider N agent systems as given in (2), and the follower system (2) is called similar to the leader system (1), if there exists N matrices $K_i \in R^{m_i \times n_i}$,

matrix $K_0 \in R^{m_0 \times n_0}$, and N matrices $T_i \in R^{n_0 \times n_i}$ satisfying the following condition:

$$\begin{cases} T_i(A_i + B_i K_i) = (A_0 + B_0 K_0)T_i, \\ T_i B_i = B_0. \end{cases} \quad (3)$$

Definition 1. In Assumption 1, T_i and K_i and K_0 are called as similar parameters with different dimensions.

Remark 1. Assumption 1 ensures that the matrices $A_i + B_i K_i$ and $A_0 + B_0 K_0$ possess some common eigenvalues. Thus, Assumption 1 implies that the agent systems as given in (1) and (2) contain certain similar inner dynamical behavior, and these agent systems are named as similar structure agents with similar parameters.

Remark 2. From a mathematical point of view, Assumption 1 admits that the state dimensions can be different or identical in multiagent systems. Especially, if $A_i = A_0$ and $n_i = n_0$ in (1) and (2), then the agent system (1) and (2) coincides with the system in [26–32].

In order to address the unknown nonlinear function, the following fuzzy logic system (FLS) is utilized in this paper. It mainly includes four parts: fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifier. The fuzzifier is a mapping from the input space and state space to the fuzzy sets. The fuzzy rule base contains several linguistics rules. The fuzzy rules are represented as follows:

$$p\text{th: If } x_1 \text{ is } F_1^p, x_2 \text{ is } F_2^p, \dots, x_n \text{ is } F_n^p, \text{ then } y \text{ is } G_p, \quad p = 1, 2, \dots, h, \quad (4)$$

where $x = (x_1, x_2, \dots, x_n)^T$ and y are the input and output of the FLS, respectively. $\mu_{F_i^p}(x_i)$ and $\mu_{G_p}(y)$ are the membership functions of fuzzy sets F_i^p and G_p , respectively. By employing singleton fuzzifier, center average defuzzifier, and product inference, the output of FLS can be expressed as

$$y(x) = \frac{\sum_{p=1}^h \theta_p \prod_{i=1}^n \mu_{F_i^p}(x_i)}{\sum_{p=1}^h \left[\prod_{i=1}^n \mu_{F_i^p}(x_i) \right]}, \quad (5)$$

where $\theta_p = \arg \max_{y \in R} \mu_{G_p}(y)$. If we denote $\theta^T = [\theta_1, \theta_2, \dots, \theta_h]$ and $\psi(x) = [\psi_1(x), \psi_2(x), \dots, \psi_n(x)]^T$, then the fuzzy logical systems can be rewritten as

$$y(x) = \theta^T \psi(x). \quad (6)$$

Lemma 2 (see [39]). *For any given two vectors $x, y \in R^n$ and a scalar $a > 0$, the following inequality holds:*

$$2x^T y \leq ax^T x + a^{-1} y^T y. \quad (7)$$

Lemma 3 (see [11]). *For any given uncertain continuous function $g(x)$ on a compact set Ω and an arbitrary*

approximation accuracy $\varepsilon > 0$, there exists a FLS such as (4) such that the following universal approximation holds:

$$\sup_{x \in \Omega} |\theta^T \psi(x) - g(x)| \leq \varepsilon. \quad (8)$$

According to Lemma 3, we know that the unknown nonlinear function $g_i(x_i)$ in the i th follower system can be approximated by

$$g_i(x_i) = \theta_i^T \psi_i(x_i) + \varepsilon_i(t), \quad (9)$$

where $\theta_i(t) = [\theta_{i1}(t), \theta_{i2}(t), \dots, \theta_{im_i}(t)]^T$ is unknown parameter vector that will be designed by adaptive laws and $\psi_i(x_i) = [\psi_{i1}, \psi_{i2}, \dots, \psi_{im_i}]^T$ is the fuzzy basis function as shown in (6). In this paper, approximation accuracy $\varepsilon_i(t)$ is a time-varying function and satisfies $|\varepsilon_i(t)| \leq \bar{\varepsilon}_i$ for all $t \geq t_0$, where $\bar{\varepsilon}_i$ is a known constant.

Control Purpose. The aim of this paper is to design a distributed fuzzy adaptive control by using similar parameter such that the consensus errors are UUB.

3. Main Results

To solve the consensus problem of the leader-follower system (1) and (2), the following control strategy is proposed:

$$\begin{aligned} u_i(t) = cF \sum_{j=1}^n \alpha_{ij} (T_i x_i(t) - T_j x_j(t)) + K_i x_i(t) \\ + \bar{K} (T_i x_i(t) - T_0 x_0(t)) - \bar{\theta}_i^T(t) \psi_i(x_i), \end{aligned} \quad (10)$$

where $F = -0.5B_0^T P$ and the control gain matrix $\bar{K} = YX^{-1}$ can be proposed by solving the following LMI:

$$\begin{bmatrix} \Delta & X^T \\ * & -Q \end{bmatrix} \leq 0, \quad (11)$$

where $\Delta = A_0 X + X A_0^T + \bar{B}_0 X + X \bar{B}_0^T + B_0 Y + Y^T B_0^T - c \lambda B_0 B_0^T$, $X > 0$, $Q > 0$, and parameter $0 < \lambda < \lambda_{\min}(\bar{L})$.

In control (10), parameters $\bar{\theta}_i(t)$ denote the estimation values of $\theta_i(t)$ and $\tilde{\theta}_i(t)$ are their errors, and the relation between them is defined as $\tilde{\theta}_i(t) = \bar{\theta}_i(t) - \theta_i(t)$. The estimation $\bar{\theta}_i(t)$ can be designed as

$$\dot{\bar{\theta}}_i(t) = -\kappa_{\theta_i} \bar{\theta}_i(t) + \rho_{\theta_i} \psi_i(x_i) (PB_0)^T e_i(t), \quad (12)$$

where $\kappa_{\theta_i} = [\kappa_{\theta_{i1}} \ \kappa_{\theta_{i2}} \ \dots \ \kappa_{\theta_{im_i}}]$ is a vector consisting of some known positive constants given by designer, $\theta_i(t) = [\theta_{i1}^T(t), \theta_{i2}^T(t), \dots, \theta_{im_i}^T(t)]^T$, $\rho_{\theta_i} > 0$.

Theorem 1. *Suppose that Assumption 1 is satisfied, and at least one agent system in connected graph G has access to the state information of the leader system (1). With the action of control (10), the consensus error between leader system (1) and follower system (2) is UUB and belongs to the following set:*

$$D := \left\{ e(t) = \begin{bmatrix} e_1^T(t) & e_2^T(t) & \dots & e_N^T(t) \end{bmatrix}^T : \|e(t)\| \leq \sqrt{\frac{\sigma}{\lambda_{\min}(I_N \otimes P)\gamma}} \right\}, \quad (13)$$

where σ and γ will be given later.

Proof. Let consensus error as $e_i(t) = T_i x_i(t) - T_0 x_0(t)$ and $T_0 = I^{n_0 \times n_0}$ is an identity matrix. By applying Assumption 1, it becomes

$$\begin{aligned} T_i \dot{x}_i(t) &= (A_0 + B_0 K_0) T_i x_i(t) - B_0 \bar{\theta}_i^T(t) \psi_i(x_i) \\ &\quad + B_0 g_i(x) + c \sum_{j=1}^n \alpha_{ij} B_0 F(T_i x_i(t) - T_j x_j(t)) \\ &\quad + B_0 \bar{K} e_i(t). \end{aligned} \quad (14)$$

The leader system is transformed as

$$\begin{aligned} T_0 \dot{x}_0(t) &= (A_0 + B_0 K_0) T_0 x_0(t) \\ &\quad + c \alpha_{i0} B_0 F(T_i x_i(t) - T_0 x_0(t)) + B_0 s(x_0, t), \end{aligned} \quad (15)$$

and then, the error system can be transformed as

$$\begin{aligned} \dot{e}_i(t) &= (A_0 + B_0 K_0 + B_0 \bar{K}) e_i(t) \\ &\quad + B_0 \left[-\bar{\theta}_i^T(t) \psi_i(x_i) + \varepsilon_i(t) + s(x_0, t) \right] \\ &\quad + c B_0 F \left\{ \sum_{j=1}^n \alpha_{ij} [e_i(t) - e_j(t)] \right. \\ &\quad \left. - \alpha_{i0} [T_i x_i(t) - T_0 x_0(t)] \right\}. \end{aligned} \quad (16)$$

For brevity, (16) is equal to

$$\begin{aligned} \dot{e}(t) &= [I_N \otimes (A_0 + B_0 K_0 + B_0 \bar{K}) - (c\bar{L}) \otimes (B_0 F)] e(t) \\ &\quad + (I_N \otimes B_0) \left[-\bar{\theta}^T(t) \psi(x) + \varepsilon(t) + s(x_0, t) \right], \end{aligned} \quad (17)$$

where $e(t) = [e_1^T(t), e_2^T(t), \dots, e_N^T(t)]^T$, $\bar{\theta}(t) = [\bar{\theta}_1^T(t), \bar{\theta}_2^T(t), \dots, \bar{\theta}_N^T(t)]^T$, $\psi(x) = [\psi_1^T(x_1), \psi_2^T(x_2), \dots, \psi_N^T(x_N)]^T$, $\varepsilon(t) = [\varepsilon_1^T(t), \varepsilon_2^T(t), \dots, \varepsilon_N^T(t)]^T$, and $-\bar{L}e(t) = \sum_{j=1}^n \alpha_{ij} [e_i(t) - e_j(t)] - \alpha_{i0} [T_i x_i(t) - T_0 x_0(t)]$

The following candidate Lyapunov function is considered:

$$V(t) = \frac{1}{2} e^T(t) (I_N \otimes P) e(t) + \frac{1}{2} \bar{\theta}^T(t) \rho_\theta^{-1} \bar{\theta}(t). \quad (18)$$

The derivative of $V(t)$ along system (17) is

$$\begin{aligned} \dot{V}(t) &= \frac{1}{2} e^T(t) \{ I_N \otimes [P(A_0 + B_0 K_0 + B_0 \bar{K}) \\ &\quad + (A_0 + B_0 K_0 + B_0 \bar{K})^T P] - (c\bar{L}) \otimes (PB_0 B_0^T P) \} e(t) \\ &\quad + \bar{\theta}^T(t) \rho_\theta^{-1} \dot{\bar{\theta}}(t) + e^T(t) [I_N \otimes (PB_0)] \\ &\quad \cdot \left[-\bar{\theta}^T(t) \psi(x) + \varepsilon(t) + s(x_0, t) \right] \\ &\leq \frac{1}{2} e^T(t) \{ I_N \otimes [P(A_0 + B_0 K_0 + B_0 \bar{K}) \\ &\quad + (A_0 + B_0 K_0 + B_0 \bar{K})^T P] - c\lambda(PB_0 B_0^T P) \} e(t) \\ &\quad + e^T(t) [I_N \otimes (PB_0)] [\varepsilon(t) + \bar{s}] - \frac{\kappa_\theta \bar{\theta}^T(t) \bar{\theta}(t)}{\rho_\theta} \\ &= -\frac{1}{2} e^T(t) (I_N \otimes Q) e(t) - \frac{\kappa_\theta \bar{\theta}^T(t) \bar{\theta}(t)}{\rho_\theta} \\ &\quad + e^T(t) [I_N \otimes (PB_0)] [\varepsilon(t) + \bar{s}]. \end{aligned} \quad (19)$$

Based on Lemma 3, one obtains that

$$\begin{aligned} -\bar{\theta}^T(t) \bar{\theta}(t) &= -\bar{\theta}^T(t) (\bar{\theta}(t) + \theta(t)) \\ &\leq -\frac{1}{2} \bar{\theta}^T(t) \bar{\theta}(t) + \frac{1}{2} \theta^T(t) \theta(t). \end{aligned} \quad (20)$$

Combining with (19) and (20), it becomes

$$\begin{aligned} \dot{V}(t) &\leq -\frac{1}{2} e^T(t) (I_N \otimes Q) e(t) - \frac{1}{2} \frac{\kappa_\theta \bar{\theta}^T(t) \bar{\theta}(t)}{\rho_\theta} \\ &\quad + \frac{1}{2} \frac{\kappa_\theta \theta^T(t) \theta(t)}{\rho_\theta} + e^T(t) [I_N \otimes (PB_0)] [\varepsilon(t) + \bar{s}] \\ &\leq -\frac{1}{2} \lambda_{\min}(I_N \otimes Q) e^T(t) e(t) - \frac{1}{2} \frac{\kappa_\theta \bar{\theta}^T(t) \bar{\theta}(t)}{\rho_\theta} \\ &\quad + \|e(t)\| \cdot \|I_N \otimes (PB_0)\| \cdot (\bar{\varepsilon} + \bar{s}) + \frac{1}{2} \frac{\kappa_\theta \theta^T(t) \theta(t)}{\rho_\theta} \\ &\leq -\gamma V(t) + \sigma. \end{aligned} \quad (21)$$

If denoting $\gamma = \min\{\lambda_{\min}(I_N \otimes Q)/\lambda_{\max}(I_N \otimes P), \kappa_\theta\}$, $\sigma = \|e(t)\| \cdot \|I_N \otimes (PB_0)\| \cdot [\bar{\varepsilon} + \bar{s}] + (1/2)(\kappa_\theta/\rho_\theta)\theta^T(t)\theta(t)$, then it follows

$$V(t) \leq \left[V(0) - \frac{\sigma}{\gamma} \right] e^{-\gamma t} + \frac{\sigma}{\gamma}. \quad (22)$$

Inequality (22) shows that the consensus error $e(t)$ can be guaranteed to be UUB with

$$\|e(t)\| \leq \frac{1}{\sqrt{\lambda_{\min}(I_N \otimes P)}} \sqrt{\left[V(0) - \frac{\sigma}{\gamma} \right] e^{-\gamma t} + \frac{\sigma}{\gamma}}. \quad (23)$$

Accordingly, the conclusion is

$$\lim_{t \rightarrow \infty} \|e(t)\| \leq \sqrt{\frac{\sigma}{\lambda_{\min}(I_N \otimes P)\gamma}}. \quad (24)$$

(24) means that the consensus error $e(t)$ converges to the set D , which is defined in (13). This completes the proof. \square

Remark 3. The inequality $P(A_0 + B_0 K_0 + B_0 \bar{K}) + (A_0 + B_0 K_0 + B_0 \bar{K})^T P - c\lambda P B_0 B_0^T P \leq -Q$ is a nonlinear matrix inequality; through multiplying by P^{-1} on the left and right sides of this inequality and defining $B_0 K_0 = \bar{B}_0$, $P^{-1} = X$, and $Y = \bar{K}X$, the inequality (11) can be obtained.

The multiagent system with similar composite structure and proposed control scheme is explained as the block diagram in Figure 1

4. Simulation Example

In this section, a simulation example is given to prove the effectiveness of the proposed control. Six agent systems are considered including one leader labeled 0 and five followers labeled 1, 2, 3, 4, and 5. Figure 2 shows the communication between the leader and each follower, it is easy to know that only the first agent can obtain the state information of the leader.

From Figure 2, the Laplacian matrix of the follower system and the degree matrix of the leader system can be calculated as follows:

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \quad (25)$$

$$\text{diag}([\alpha_{10} \ \alpha_{20} \ \cdots \ \alpha_{50}]) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Matrices in the leader-follower system are represented by

$$A_0 = \begin{bmatrix} 1 & -7 & -3 & -1 & -1 & 2 & -2 & -4 \\ 5 & 2 & -4 & 2 & -7 & -8 & 9 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -9 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 1 & -7 & -3 & -1 & -1 & 2 & -2 \\ 2 & 7 & 3 & -2 & -8 & -10 & 2 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0 \\ 1 \\ O_{1 \times 6} \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 & -7 & -3 & -1 & -1 & 2 \\ 3 & 7 & -2 & -1 & -6 & -9 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6 \end{bmatrix}, \quad (26)$$

$$B_1 = \begin{bmatrix} 0 \\ 1 \\ O_{1 \times 5} \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & -7 & -3 & -1 & -1 \\ 5 & 3 & -5 & 2 & 3 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -3 \end{bmatrix},$$

$$A_4 = \begin{bmatrix} 1 & -7 & -3 & -1 \\ 1 & 3 & 2 & 7 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} 1 & -7 & -3 \\ 5 & 2 & 1 \\ 0 & 0 & -1 \end{bmatrix},$$

$$B_2 = [0 \ 1 \ O_{1 \times 4}]^T,$$

$$B_3 = [0 \ 1 \ O_{1 \times 3}]^T,$$

$$B_4 = [0 \ 1 \ O_{1 \times 2}]^T,$$

$$B_5 = [0 \ 1 \ 0]^T.$$

By using the similar condition in Assumption 1, the similar parameters can be obtained as

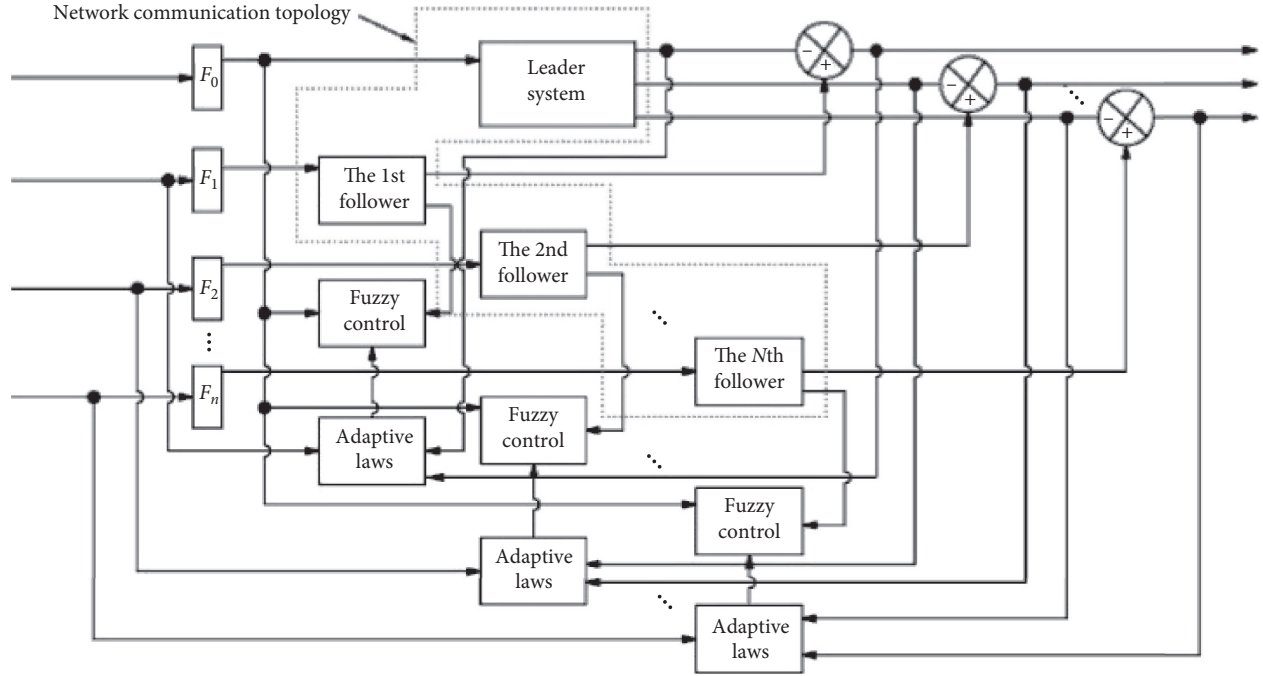


FIGURE 1: The block diagram of closed-loop multiagent systems.

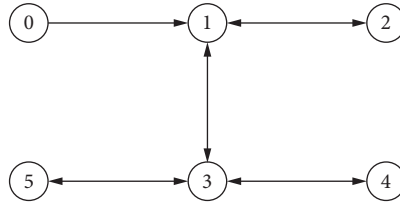


FIGURE 2: The communication topology.

$$\bar{K} = [-0.5909 \quad -0.0964 \quad -12.7679 \quad -0.9654 \quad -0.0483 \quad -0.2017 \quad -0.3305 \quad -0.2047],$$

$$P = \begin{bmatrix} 0.4057 & -0.1665 & -0.1051 & -0.0329 & -0.0324 & 0.0673 & -0.0684 & -0.1369 \\ -0.1665 & 0.4504 & 0.1632 & 0.0545 & 0.0553 & -0.1180 & 0.1225 & 0.2487 \\ -0.1051 & 0.1632 & 0.3894 & 0.0388 & 0.0379 & -0.0726 & 0.0708 & 0.1398 \\ -0.0329 & 0.0545 & 0.0388 & 0.3294 & 0.0120 & -0.0233 & 0.0229 & 0.0453 \\ -0.0324 & 0.0553 & 0.0379 & 0.0120 & 0.4195 & -0.0229 & 0.0226 & 0.0449 \\ 0.0673 & -0.1180 & -0.0726 & -0.0233 & -0.0229 & 0.8276 & -0.0457 & -0.0914 \\ -0.0684 & 0.1225 & 0.0708 & 0.0229 & 0.0226 & -0.0457 & 1.1140 & 0.0923 \\ -0.1369 & 0.2487 & 0.1398 & 0.0453 & 0.0449 & -0.0914 & 0.0923 & 1.4012 \end{bmatrix},$$

$$T_0 = I_8,$$

$$T_1 = [I_7 \quad O_{7 \times 1}]^T,$$

$$T_2 = [I_6 \quad O_{6 \times 2}]^T,$$

$$T_3 = [I_5 \quad O_{5 \times 3}]^T,$$

$$T_4 = [I_4 \quad O_{4 \times 4}]^T,$$

$$T_5 = [I_3 \quad O_{3 \times 5}]^T,$$

$$K_0 = [3 \quad -6 \quad 14 \quad -2 \quad 6 \quad 11 \quad -12 \quad -6],$$

$$K_1 = [6 \quad -11 \quad 7 \quad 2 \quad 7 \quad 13 \quad -5],$$

$$K_2 = [5 \quad -11 \quad 12 \quad 1 \quad 5 \quad 12],$$

$$K_3 = [3 \quad -7 \quad 15 \quad -2 \quad -4],$$

$$K_4 = [7 \quad -7 \quad 8 \quad -7],$$

$$K_5 = [3 \quad -6 \quad 9].$$

(27)

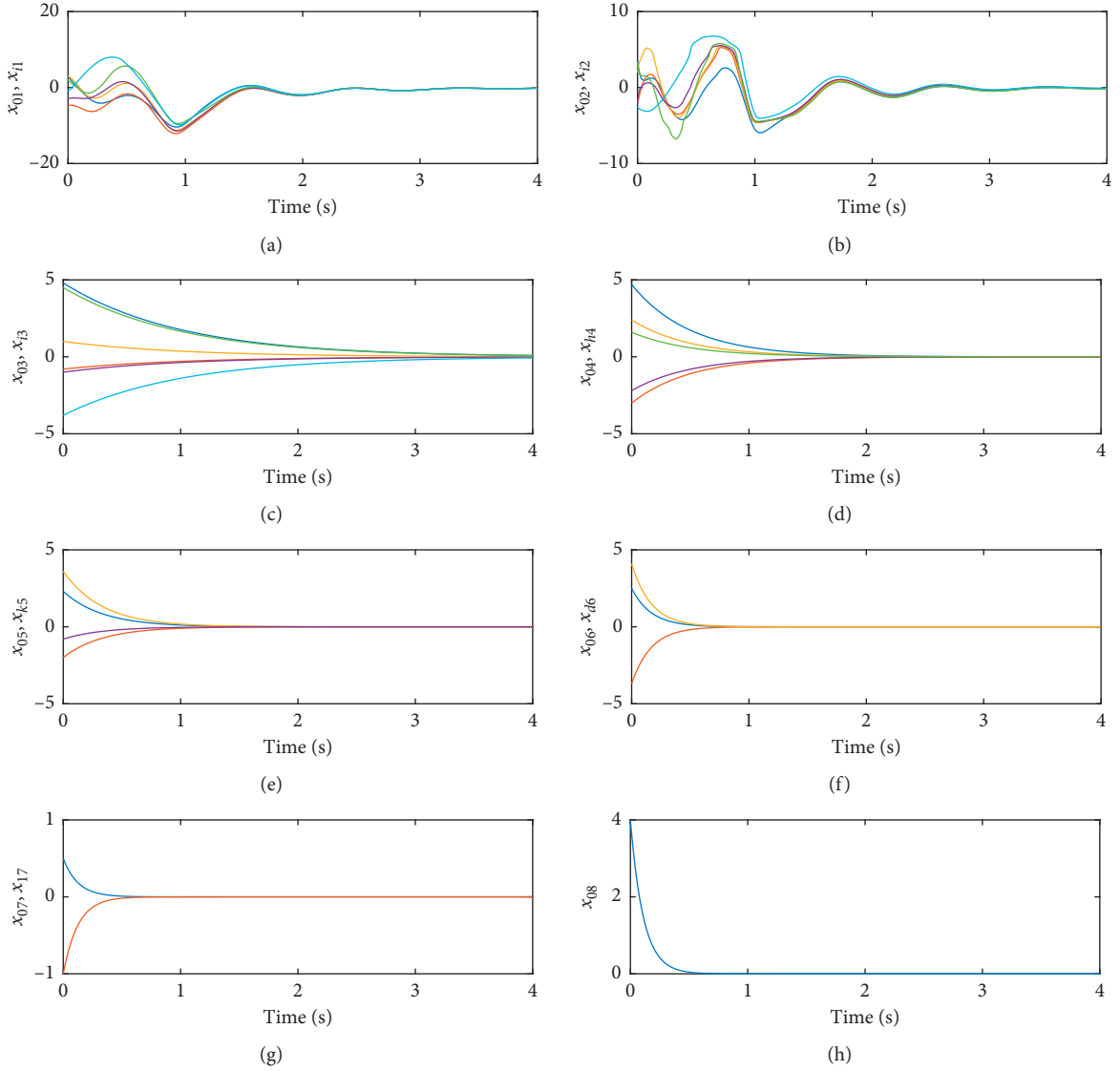


FIGURE 3: Trajectories of the states of leader x_0 and followers x_i ($i = 1, 2, 3, 4, 5$; $h = 1, 2, 3, 4$; $k = 1, 2, 3$; $d = 1, 2$).

The nonlinear functions are chosen as

$$f_i = x_{i1}^2 \sin x_{i2} \cos x_{i3} - 0.5x_{i2}^3 \sin x_{i2} - 2x_{i3}^4 \cos x_{i3}. \quad (28)$$

The solutions of the linear matrix inequality (11) are shown as matrices \bar{K} and P .

The input bounded signal can be chosen as

$$s(x_0, t) = \begin{cases} 60, & 0 < t \leq 1, \\ 0, & 1 < t \leq 2, \\ 60, & 2 < t \leq 3, \\ 0, & 3 < t \leq 4, \\ 60, & 4 < t \leq 5. \end{cases} \quad (29)$$

The initial values of the states in the leader and five followers are chosen as

$$\begin{aligned} x_0(0) &= [3.1, 3.5, 4.8, 4.7, 2.3, 2.5, 0.5, 4]^T, \\ x_1(0) &= [-4.8, -2, -0.8, -3, -2, -3.7, -1]^T, \\ x_2(0) &= [3, 2, 1, 2.4, 3.6, 4.1]^T, \\ x_3(0) &= [-3, -2, -1, -2.2, -0.8]^T, \\ x_4(0) &= [2, 3.3, 4.5, 1.6]^T, \\ x_5(0) &= [-1, -2.5, -3.8]^T. \end{aligned} \quad (30)$$

The initial values of adaptive parameters $\bar{\theta}_i(t)$ are given as

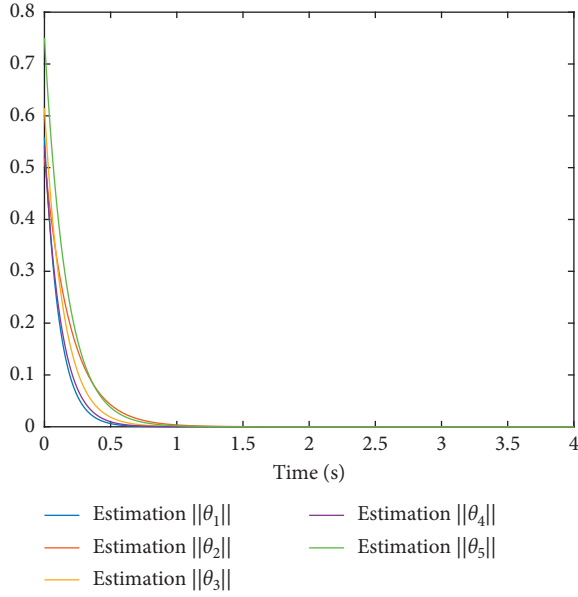


FIGURE 4: Trajectories of the adaptive estimation parameters $\bar{\theta}_i(t)$.

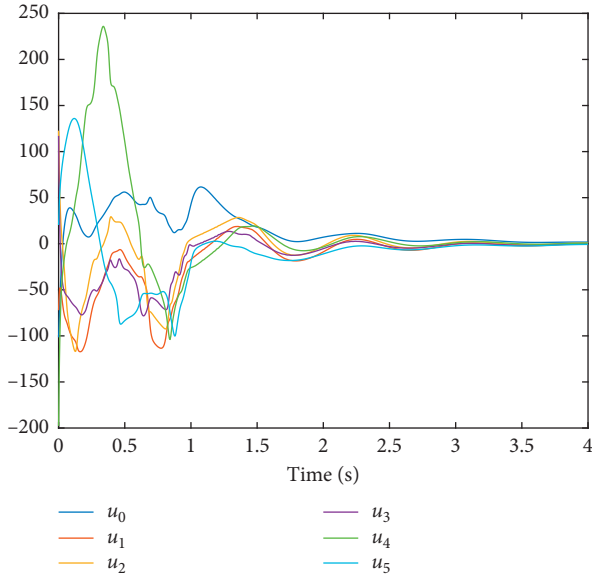


FIGURE 5: Trajectories of the control $u_0(t)$ and $u_i(t)$.

$$\begin{aligned}
 \bar{\theta}_1(0) &= [0.13, 0.22, 0.10, 0.23, 0.18, 0.39]^T, \\
 \bar{\theta}_2(0) &= [0.24, 0.26, 0.11, 0.14, 0.15, 0.30]^T, \\
 \bar{\theta}_3(0) &= [0.16, 0.29, 0.17, 0.21, 0.35, 0.27]^T, \\
 \bar{\theta}_4(0) &= [0.20, 0.21, 0.11, 0.31, 0.29, 0.13]^T, \\
 \bar{\theta}_5(0) &= [0.34, 0.29, 0.38, 0.26, 0.35, 0.17]^T.
 \end{aligned} \tag{31}$$

The parameters in the adaptive law (12) are chosen as

$$\begin{aligned}
 \kappa_{\theta_i} &= [9 \ 5 \ 7 \ 8 \ 6], \\
 \rho_{\theta_i} &= [0.003 \ 0.002 \ 0.002 \ 0.003 \ 0.001].
 \end{aligned} \tag{32}$$

The simulation results of the leader system and follower systems are shown as Figures 3–5.

As shown in Figure 3, although the dimensions of leader system and follower systems are nonidentical, the trajectories of x_i in follower systems can synchronize to the state of x_0 in leader system with the proposed distributed fuzzy adaptive control, and it can reach a consistent state in a relatively fast time. Similarly, the norm of adaptive estimated parameters is converged to a small zero field in Figure 4, which can be updated online automatically with the given adaptive laws. From Figure 5, it is shown that the time responses of corresponding control are UUB. Finally, it is concluded that UUB of all signals in the closed-loop system can be guaranteed in Figures 3–5, and the consensus of leader-follower system can be realized by the proposed distribute fuzzy adaptive control with similar parameters whether the leader system and follower systems have the identical or nonidentical dimensions.

5. Conclusion

The consensus problem of leader-follower multiagent systems with different dimensions has been considered in this paper. For the unknown nonlinear functions in systems, FLSs are applied to approximate the unknown nonlinear functions, and then a distributed fuzzy adaptive control based on similar condition is designed. With the proposed fuzzy adaptive control, the states of each follower system can stably track the states of the leader system, and it is proved that all signals in the closed-loop system are UUB. The designed method has been verified by a simulation example.

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 (Grant nos. 51875457 and 61903298), Shaanxi Provincial Department of Science and Technology Key Project in the Field of Industry (2018ZDXM-GY-039), and National Natural Science Foundation of Shaanxi (Grant no. 2019JQ-341).

References

- [1] S. D. J. McArthur, E. M. Davidson, V. M. Catterson et al., “Multi-agent systems for power engineering applications—part I: concepts, approaches, and technical challenges,” *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1743–1752, 2007.
- [2] S. D. J. McArthur, E. M. Davidson, V. M. Catterson et al., “Multi-agent systems for power engineering applications—part II: technologies, standards, and tools for building

- multi-agent systems,” *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1753–1759, 2007.
- [3] T. Li and J.-F. Zhang, “Asymptotically optimal decentralized control for large population stochastic multiagent systems,” *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1643–1660, 2008.
 - [4] Z. Peng, D. Wang, H. Zhang, and G. Sun, “Distributed neural network control for adaptive synchronization of uncertain dynamical multiagent systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1508–1519, 2014.
 - [5] A. Howard, L. E. Parker, and G. S. Sukhatme, “Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection,” *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.
 - [6] S.-J. Chung, J.-J. E. Slotine, Cooperative robot control and concurrent synchronization of Lagrangian systems,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 686–700, 2009.
 - [7] M. Rehan, A. Jameel, and C. K. Ahn, “Distributed consensus control of one-sided Lipschitz nonlinear multiagent systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 8, pp. 1297–1308, 2018.
 - [8] F. Xiao and T. Chen, “Adaptive consensus in leader-following networks of heterogeneous linear systems,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1169–1176, 2018.
 - [9] J. Fu, G. Wen, W. Yu, T. Huang, and J. Cao, “Exponential consensus of multiagent systems with Lipschitz nonlinearities using sampled-data information,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4363–4375, 2018.
 - [10] J. Qin and C. Yu, “Exponential consensus of general linear multi-agent systems under directed dynamic topology,” *Automatica*, vol. 50, no. 9, pp. 2327–2333, 2014.
 - [11] L.-X. Wang and J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least-squares learning,” *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.
 - [12] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.-Y. Su, “Personalized variable gain control with tremor attenuation for robot teleoperation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1759–1770, 2018.
 - [13] L. Kong, W. He, C. Yang, Z. Li, and C. Sun, “Adaptive fuzzy control for coordinated multiple robots with constraint using impedance learning,” *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3052–3063, 2019.
 - [14] Z. G. Hou, L. Cheng, and M. Tan, “Decentralized robust adaptive control for the multiagent system consensus problem using neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 3, pp. 636–647, 2009.
 - [15] W. Ouyang, W. Liang, C. Li, H. Zheng, Q. Ren, and P. Li, “Steering motion control of a snake robot via a biomimetic approach,” *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 1, pp. 32–44, 2019.
 - [16] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, “Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.
 - [17] W. He, L. Kong, Y. Dong, Y. Yu, C. Yang, and C. Sun, “Fuzzy tracking control for a class of uncertain MIMO nonlinear systems with state constraints,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 3, pp. 543–554, 2019.
 - [18] C. Yang, X. Wang, L. Cheng, and H. Ma, “Neural-learning-based telerobot control with guaranteed performance,” *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3148–3159, 2017.
 - [19] Y. Li, C. Yang, W. Yan, R. Cui, and A. Annamalai, “Admittance-based adaptive cooperative control for multiple manipulators with output constraints,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3621–3632, 2019.
 - [20] G. Peng, C. Yang, W. He, and C. L. P. Chen, “Force sensorless admittance control with neural learning for robots with actuator saturation,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3138–3148, 2020.
 - [21] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, “Robot learning system based on adaptive neural control and dynamic movement primitives,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
 - [22] Q. Shen, P. Shi, and Y. Shi, “Distributed adaptive fuzzy control for nonlinear multiagent systems via sliding mode observers,” *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3086–3097, 2016.
 - [23] Y.-X. Li, G.-H. Yang, and S. Tong, “Fuzzy adaptive distributed event-triggered consensus control of uncertain nonlinear multiagent systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 9, pp. 1777–1786, 2019.
 - [24] L. Cao, H. Li, G. Dong, and R. Lu, “Event-triggered control for multiagent systems with sensor faults and input saturation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2019.
 - [25] Q. Zhou, W. Wang, H. Liang, M. V. Basin, and B. Wang, “Observer-based event-triggered fuzzy adaptive bipartite containment control of multi-agent systems with input quantization,” *IEEE Transactions on Fuzzy Systems*, pp. 1–13, 2019.
 - [26] H. Kim, H. Shim, and J. H. Seo, “Output consensus of heterogeneous uncertain linear multi-agent systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 200–206, 2011.
 - [27] W. Hu, L. Liu, and G. Feng, “Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy,” *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1914–1924, 2017.
 - [28] Y. Lv, Z. Li, Z. Duan, and G. Feng, “Novel distributed robust adaptive consensus protocols for linear multi-agent systems with directed graphs and external disturbances,” *International Journal of Control*, vol. 90, no. 2, pp. 137–147, 2017.
 - [29] C. L. P. Chen, C.-E. Ren, and T. Du, “Fuzzy observed-based adaptive consensus tracking control for second-order multiagent systems with heterogeneous nonlinear dynamics,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 4, pp. 906–915, 2016.
 - [30] C.-E. Ren, L. Chen, and C. L. P. Chen, “Adaptive fuzzy leader-following consensus control for stochastic multiagent systems with heterogeneous nonlinear dynamics,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 181–190, 2017.
 - [31] Z. Li, Z. Duan, and F. L. Lewis, “Distributed robust consensus control of multi-agent systems with heterogeneous matching uncertainties,” *Automatica*, vol. 50, no. 3, pp. 883–889, 2014.
 - [32] Y. Lv, Z. Li, and Z. Duan, “Distributed PI control for consensus of heterogeneous multiagent systems over directed graphs,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1602–1609, 2020.
 - [33] C. D. Araújo and J. D. Castro, “Application of power system stabilisers in a plant with identical units,” *IEEE Proceedings C*

- (*Generation, Transmission and Distribution*), vol. 138, no. 1, pp. 11–18, 1991.
- [34] L. Bakule and J. Lunze, “Decentralized design of feedback control for large-scale systems,” *Kybernetika*, vol. 24, no. 8, pp. 1–3, 1988.
 - [35] Y.-H. Wang and S.-Y. Zhang, “Robust control for nonlinear similar composite systems with uncertain parameters,” *IEE Proceedings—Control Theory and Applications*, vol. 147, no. 1, pp. 80–86, 2000.
 - [36] X.-G. Yan and G.-Z. Dai, “Decentralized output feedback robust control for nonlinear large-scale systems,” *Automatica*, vol. 34, no. 11, pp. 1469–1472, 1998.
 - [37] X.-G. Yan, J. Lam, and G.-Z. Dai, “Decentralized stabilization for nonlinear similar composite systems with uncertainty,” *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 3444–3449, 1999.
 - [38] Y. Wang, Y. Fan, Q. Wang, and Y. Zhang, “Stabilization and synchronization of complex dynamical networks with different dynamics of nodes via decentralized controllers,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 8, pp. 1786–1795, 2012.
 - [39] P. Wang and Y. Jia, “Robust H^∞ containment control for uncertain multi-agent systems with inherent nonlinear dynamics,” *International Journal of Systems Science*, vol. 47, no. 5, pp. 1073–1083, 2016.