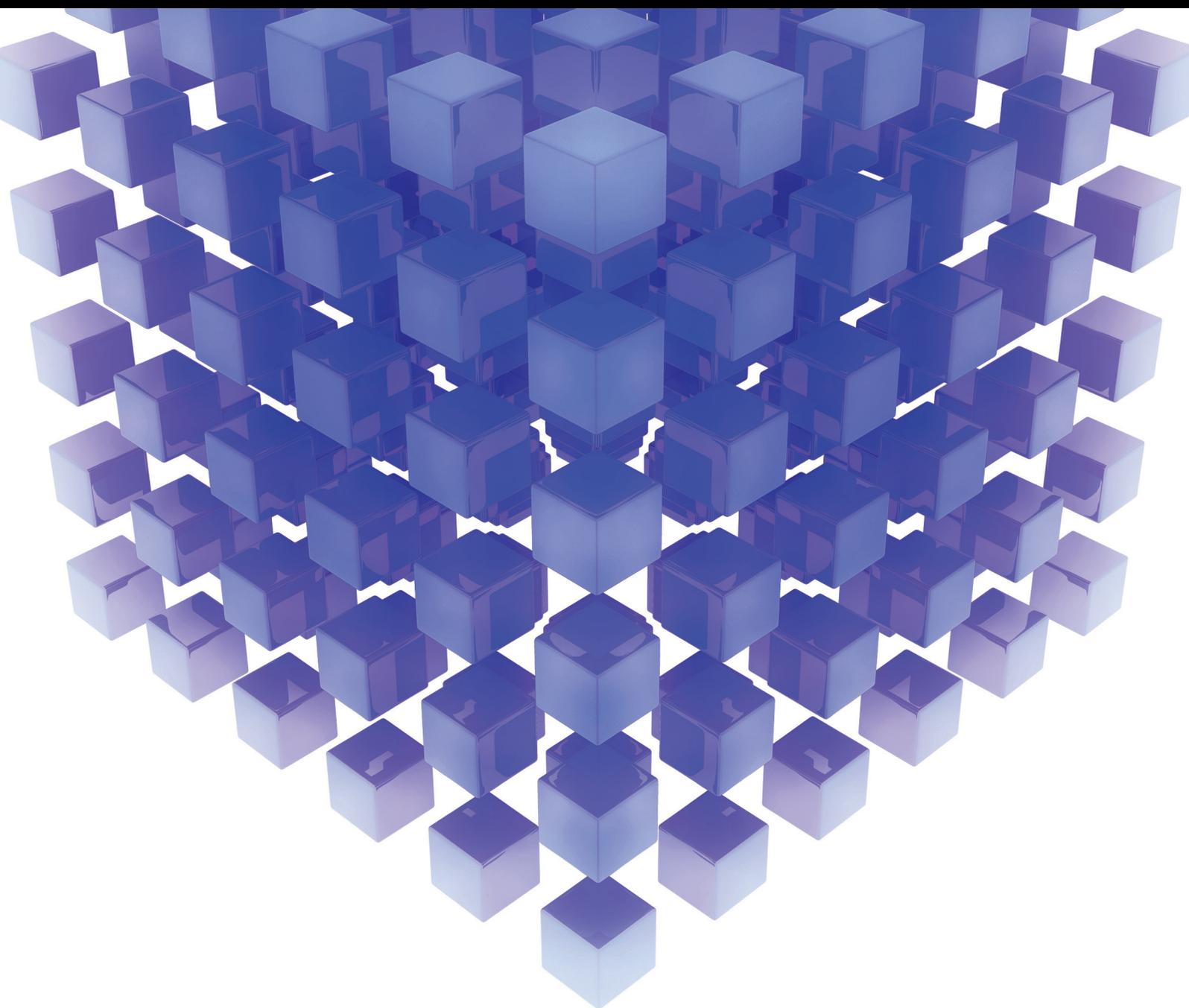


Mathematical Problems in Engineering

Applied Mathematics to Mobile Robotics and their Applications

Lead Guest Editor: Francisco Valero

Guest Editors: Marco Ceccarelli and Ashitava Ghosal





Applied Mathematics to Mobile Robotics and their Applications

Mathematical Problems in Engineering

Applied Mathematics to Mobile Robotics and their Applications

Lead Guest Editor: Francisco Valero

Guest Editors: Marco Ceccarelli



Copyright © 2017 Hindawi. All rights reserved.

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

Editorial Board

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

Elisa Francomano, Italy
Leonardo Freitas, UK
Tomonari Furukawa, USA
Mohamed Gadala, Canada
Matteo Gaeta, Italy
Mauro Gaggero, Italy
Zoran Gajic, Iraq
Erez Gal, Israel
Ugo Galvanetto, Italy
Akemi Gálvez, Spain
Rita Gamberini, Italy
Maria L. Gandarias, Spain
Arman Ganji, Canada
Zhong-Ke Gao, China
Giovanni Garcea, Italy
Jose M. Garcia-Aznar, Spain
Alessandro Gasparetto, Italy
Oleg V. Gendelman, Israel
Mergen H. Ghayesh, Australia
Agathoklis Giaralis, UK
Anna M. Gil-Lafuente, Spain
Ivan Giorgio, Italy
Alessio Gizzi, Italy
David González, Spain
Rama S. R. Gorla, USA
Oded Gottlieb, Israel
Nicolas Gourdain, France
Kannan Govindan, Denmark
Antoine Grall, France
Fabrizio Greco, Italy
Jason Gu, Canada
Federico Guarracino, Italy
José L. Guzmán, Spain
Quang Phuc Ha, Australia
Zhen-Lai Han, China
Thomas Hanne, Switzerland
Xiao-Qiao He, China
Sebastian Heidenreich, Germany
Luca Heltai, Italy
Alfredo G. Hernández-Díaz, Spain
M.I. Herreros, Spain
Eckhard Hitzer, Japan
Paul Honeine, France
Jaromir Horacek, Czech Republic
Muneo Hori, Japan
András Horváth, Italy
Gordon Huang, Canada
Sajid Hussain, Canada

Asier Ibeas, Spain
Orest V. Iftime, Netherlands
Giacomo Innocenti, Italy
Emilio Insfran Pelozo, Spain
Nazrul Islam, USA
Benoit lung, France
Benjamin Ivorra, Spain
Payman Jalali, Finland
Reza Jazar, Australia
Khalide Jbilou, France
Linni Jian, China
Bin Jiang, China
Zhongping Jiang, USA
Ningde Jin, China
Dylan F. Jones, UK
Tamas Kalmar-Nagy, Hungary
Tomasz Kapitaniak, Poland
Julius Kaplunov, UK
Haranath Kar, India
Konstantinos Karamanos, Belgium
Jean-Pierre Kenne, Canada
Chaudry M. Khalique, South Africa
Do Wan Kim, Republic of Korea
Nam-Il Kim, Republic of Korea
Manfred Krafczyk, Germany
Frederic Kratz, France
Petr Krysl, USA
Jurgen Kurths, Germany
Kyandoghere Kyamakya, Austria
Davide La Torre, Italy
Risto Lahdelma, Finland
Hak-Keung Lam, UK
Jimmy Lauber, France
Antonino Laudani, Italy
Aimé Lay-Ekuakille, Italy
Nicolas J. Leconte, France
Marek Lefik, Poland
Yaguo Lei, China
Thibault Lemaire, France
Stefano Lenci, Italy
Roman Lewandowski, Poland
Panos Liatsis, UAE
Anatoly Lisnianski, Israel
Peide Liu, China
Peter Liu, Taiwan
Wanquan Liu, Australia
Alessandro Lo Schiavo, Italy
Jean Jacques Loiseau, France

Paolo Lonetti, Italy
Sandro Longo, Italy
Sebastian López, Spain
Luis M. López-Ochoa, Spain
Vassilios C. Loukopoulos, Greece
Valentin Lychagin, Norway
Emilio Jiménez Macías, Spain
Antonio Madeo, Italy
José María Maestre, Spain
Fazal M. Mahomed, South Africa
Noureddine Manamanni, France
Didier Maquin, France
Giuseppe Carlo Marano, Italy
Damijan Markovic, France
Francesco Marotti de Sciarra, Italy
Rodrigo Martínez-Bejar, Spain
Benoit Marx, France
Franck Massa, France
Paolo Massioni, France
Alessandro Mauro, Italy
Michael Mazilu, UK
Fabio Mazza, Italy
Driss Mehdi, France
Roderick Melnik, Canada
Pasquale Memmolo, Italy
Xiangyu Meng, Canada
Jose Merodio, Spain
Alessio Merola, Italy
Luciano Mescia, Italy
Laurent Mevel, France
Yuri Vladimirovich Mikhlin, Ukraine
Aki Mikkola, Finland
Hiroyuki Mino, Japan
Pablo Mira, Spain
Vito Mocella, Italy
Roberto Montanini, Italy
Gisele Mophou, France
Rafael Morales, Spain
Marco Morandini, Italy
Simone Morganti, Italy
Aziz Moukrim, France
Emiliano Mucchi, Italy
Josefa Mula, Spain
Domenico Mundo, Italy
Jose J. Muñoz, Spain
Giuseppe Muscolino, Italy
Marco Mussetta, Italy
Hakim Naceur, France

Hassane Naji, France
Keivan Navaie, UK
Dong Ngoduy, UK
Tatsushi Nishi, Japan
Xesús Nogueira, Spain
Ben T. Nohara, Japan
Mohammed Nouari, France
Mustapha Nourelfath, Canada
Roger Ohayon, France
Mitsuhiro Okayasu, Japan
Calogero Orlando, Italy
Javier Ortega-Garcia, Spain
Alejandro Ortega-Moñux, Spain
Naohisa Otsuka, Japan
Erika Ottaviano, Italy
Arturo Pagano, Italy
Alkis S. Paipetis, Greece
Alessandro Palmeri, UK
Pasquale Palumbo, Italy
Elena Panteley, France
Achille Paolone, Italy
Xosé M. Pardo, Spain
Manuel Pastor, Spain
Pubudu N. Pathirana, Australia
Francesco Pellicano, Italy
Marcello Pellicciari, Italy
Haipeng Peng, China
Mingshu Peng, China
Zhike Peng, China
Marzio Pennisi, Italy
Maria Patrizia Pera, Italy
Matjaz Perc, Slovenia
Francesco Pesavento, Italy
Dario Piga, Switzerland
Antonina Pirrotta, Italy
Marco Pizzarelli, Italy
Vicent Pla, Spain
Javier Plaza, Spain
Sébastien Poncet, Canada
Jean-Christophe Ponsart, France
Mauro Pontani, Italy
Stanislav Potapenko, Canada
Christopher Pretty, New Zealand
Luca Pugi, Italy
Giuseppe Quaranta, Italy
Dane Quinn, USA
Vitomir Racic, Italy
Jose Ragot, France
K. R. Rajagopal, USA
Alain Rassineux, France
S.S. Ravindran, USA
Alessandro Reali, Italy
Oscar Reinoso, Spain
Nidhal Rezg, France
Ricardo Riaza, Spain
Gerasimos Rigatos, Greece
Francesco Ripamonti, Italy
Eugenio Roanes-Lozano, Spain
Bruno G. M. Robert, France
José Rodellar, Spain
Rosana Rodríguez López, Spain
Ignacio Rojas, Spain
Alessandra Romolo, Italy
Debasish Roy, India
Gianluigi Rozza, Italy
Rubén Ruiz García, Spain
Antonio Ruiz-Cortes, Spain
Ivan D. Rukhlenko, Australia
Mazen Saad, France
Kishin Sadarangani, Spain
Andrés Sáez, Spain
Mehrddad Saif, Canada
Salvatore Salamone, USA
Nunzio Salerno, Italy
Miguel A. Salido, Spain
Roque J. Salterén, Spain
Alessandro Salvini, Italy
Giuseppe Sanfilippo, Italy
Miguel A. F. Sanjuan, Spain
Vittorio Sansalone, France
José A. Sanz-Herrera, Spain
Nickolas S. Sapidis, Greece
Evangelos J. Sapountzakis, Greece
Andrey V. Savkin, Australia
Thomas Schuster, Germany
Lotfi Senhadji, France
Joan Serra-Sagrasta, Spain
Gerardo Severino, Italy
Ruben Sevilla, UK
Leonid Shaikhet, Israel
Hassan M. Shanechi, USA
Bo Shen, Germany
Suzanne M. Shontz, USA
Babak Shotorban, USA
Zhan Shu, UK
Christos H. Skiadas, Greece
Delfim Soares Jr., Brazil
Alba Sofi, Italy
Francesco Soldovieri, Italy
Raffaele Solimene, Italy
Jussi Sopanen, Finland
Marco Spadini, Italy
Ruben Specogna, Italy
Sri Sridharan, USA
Ivanka Stamova, USA
Salvatore Strano, Italy
Yakov Strelniker, Israel
Sergey A. Suslov, Australia
Thomas Svensson, Sweden
Andrzej Swierniak, Poland
Andras Szekrenyes, Hungary
Yang Tang, Germany
Alessandro Tasora, Italy
Sergio Teggi, Italy
Alexander Timokha, Norway
Gisella Tomasini, Italy
Francesco Tornabene, Italy
Antonio Tornambe, Italy
Javier Martinez Torres, Spain
George Tsiatas, Greece
Antonios Tsourdos, UK
Emilio Turco, Italy
Vladimir Turetsky, Israel
Mustafa Tutar, Spain
Ilhan Tuzcu, USA
Efstratios Tzirtzilakis, Greece
Filippo Ubertini, Italy
Francesco Ubertini, Italy
Hassan Ugail, UK
Giuseppe Vairo, Italy
Eusebio Valero, Spain
Pandian Vasant, Malaysia
Marcello Vasta, Italy
Miguel E. Vázquez-Méndez, Spain
Josep Vehi, Spain
Kalyana C. Veluvolu, Republic of Korea
Fons J. Verbeek, Netherlands
Franck J. Vernerey, USA
Georgios Veronis, USA
Anna Vila, Spain
Rafael-Jacinto Villanueva-Micó, Spain
Uchechukwu E. Vincent, UK
Francesca Vipiana, Italy
Mirko Viroli, Italy



Michael Vynnycky, Sweden
Shuming Wang, China
Yan-Wu Wang, China
Yongqi Wang, Germany
Roman Wendner, Austria
Desheng D. Wu, Sweden
Yuqiang Wu, China
Guangming Xie, China

Xuejun Xie, China
Gen Q. Xu, China
Hang Xu, China
Joseph J. Yame, France
Xinggang Yan, UK
Luis J. Yebra, Spain
Peng-Yeng Yin, Taiwan
Qin Yuming, China

Vittorio Zampoli, Italy
Ibrahim Zeid, USA
Huaguang Zhang, China
Qingling Zhang, China
Zhao Zhang, China
Jian G. Zhou, UK
Quanxin Zhu, China
Mustapha Zidi, France

Applied Mathematics to Mobile Robotics and Their Applications

Francisco Valero, Marco Ceccarelli, and Ashitava Ghosal

Volume 2017, Article ID 8706164, 2 pages

Cascade Sliding Control for Trajectory Tracking of a Nonholonomic Mobile Robot with Adaptive Neural Compensator

Flavio Capraro, Francisco Guido Rossomando, Carlos Soria, and Gustavo Scaglia

Volume 2017, Article ID 8501098, 13 pages

Influence of the Friction Coefficient on the Trajectory Performance for a Car-Like Robot

Francisco Valero, Francisco Rubio, Carlos Llopis-Albert, and Juan Ignacio Cuadrado

Volume 2017, Article ID 4562647, 9 pages

Adaptive Fuzzy Integral Terminal Sliding Mode Control of a Nonholonomic Wheeled Mobile Robot

Shuying Peng and Wuxi Shi

Volume 2017, Article ID 3671846, 12 pages

A FastSLAM Algorithm Based on Nonlinear Adaptive Square Root Unscented Kalman Filter

Yu-feng Zhang, Qi-xun Zhou, Ju-zhong Zhang, Yi Jiang, and Kai Wang

Volume 2017, Article ID 4197635, 9 pages

Kinematics, Dynamics, and Optimal Control of Pneumatic Hexapod Robot

Long Bai, Lu-han Ma, Zhifeng Dong, and Xinsheng Ge

Volume 2017, Article ID 6841972, 16 pages

Path Tracking Control of Automatic Parking Cloud Model considering the Influence of Time Delay

Yiding Hua, Haobin Jiang, Yingfeng Cai, Xupei Zhang, Shidian Ma, and Di Wu

Volume 2017, Article ID 6590383, 14 pages

Robust Visual Tracking Using the Bidirectional Scale Estimation

An Zhiyong, Guan Hao, and Li Jinjiang

Volume 2017, Article ID 3276103, 10 pages

Parameter Selection for Ant Colony Algorithm Based on Bacterial Foraging Algorithm

Peng Li and Hua Zhu

Volume 2016, Article ID 6469721, 12 pages

Editorial

Applied Mathematics to Mobile Robotics and Their Applications

Francisco Valero,¹ Marco Ceccarelli,² and Ashitava Ghosal³

¹*Centro de Investigación en Ingeniería Mecánica, Universitat Politècnica de València, València, Spain*

²*Laboratory of Robotics and Mechatronics, DICeM, University of Cassino and Southern Latium, Cassino, Italy*

³*Department of Mechanical Engineering, IIS, Bangalore, India*

Correspondence should be addressed to Francisco Valero; fvalero@mcm.upv.es

Received 29 October 2017; Accepted 30 October 2017; Published 15 November 2017

Copyright © 2017 Francisco Valero 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 topics of this special issue are related to hot problems for the scientific community working not only in robotics since mobile robots are challenging systems for many fields with increasing applications.

Mobile robots can move autonomously in a wide range of environments with or without assistance from external human operators and can be used in a great variety of applications, such as agricultural and industrial works, road transport, exploration, services, information, and orientation in large shopping malls. They can be included in many other systems such as humanoid robots, unmanned rovers, entertainment pets, and drones. A distinctive feature is their ability to move independently, with some intelligence to react and make decisions based on the perception they receive from the environment by means of sensors.

The design and developments of mobile robots involve different technological areas such as mechanics, electronics, and computer science. This multidisciplinary approach implies studies of the dynamic behavior of a mobile robot which are necessary to ensure the performance and safety not only of the mobile robot but also of the environment within which it moves. Sensor and the control systems are also needed to guarantee the result.

It is important to bear in mind that the main pillars of mobile robotics, according to many literature sources, consist of the fields of locomotion, perception, cognition, and navigation.

Locomotion problems mean the understanding of the robot mechanism and its kinematics, dynamics, and control theory as related to the system mobility. Perception implies the fields of signal analysis and specialized areas such as

computer vision and sensor technologies. Cognition and perception systems give information about the environment, the robot itself, and the relationship between robot and environment. This information is processed and then appropriate commands are sent to the actuators, which move the mechanical structure. Navigation skills are intended to the robot to move from one place to another in a known or unknown environment, taking into account the values of the sensors. The robot must rely on aspects, such as perception, localization, cognition, and motion control (path, trajectory, and tracking planning). Over the years, numerous methodologies have arisen that attempt to solve motion planning.

This special issue contains the set of the following works that are related to the above-mentioned topics.

In S. Peng et al., the uncertainties in the trajectory tracking problem for a wheeled mobile robot are approximated by a fuzzy logic system. A robust controller is employed to compensate for the lumped errors. As a result, the tracking position errors converge asymptotically to zeros with faster response than other existing controllers.

F. Valero et al. have tackled the influence of the friction coefficient on the trajectory performance for a car-like robot.

Y. Zhang et al. develop a fast-simultaneous localization and mapping (FastSLAM) algorithm for a vehicle in indoor environments. It is based on nonlinear adaptive square root unscented Kalman filter.

L. Bai et al. consider a pneumatic hexapod robot which is driven by inert gas carried by itself. Kinematics, dynamics, and optimal control are attached with design purposes. The optimal input gas pressure of leg swing and body moving in one step is obtained by pseudospectral method.

In Y. Hua et al., a path tracking control of an automatic parking cloud model is proposed by considering the influence of time delay. This paper presents a kinematic model of the automatic parking system and analyzes the kinematic constraints of the vehicle. The effectiveness and timeliness of automatic parking controller are tested in the aspect of path tracking through a real vehicle experiment.

A. Zhiyong et al. present a novel visual tracking algorithm that learns the translation and scale filters with a complementary scheme to object tracking with robust scale estimation which is a challenging task in computer vision.

In P. Li et al., the authors propose an improvement of a path planning algorithm based on Ant Colony Algorithms (ACA). This is carried out by a better parameter selection method that is based on the bacterial foraging algorithm (BFA). The results indicate that the proposed parameter selection is a superior method as being able to determine the best parameter combination rapidly, accurately, and effectively.

We hope that readers will consider this special issue useful for their work on mobile robots regarding advanced mathematical modelling, design, analysis, optimization techniques, and so forth.

Acknowledgments

Finally, we would like to thank the reviewers whose collaboration has made it possible to prepare and publish this special issue, as well as the authors who have submitted their papers.

*Francisco Valero
Marco Ceccarelli
Ashitava Ghosal*

Research Article

Cascade Sliding Control for Trajectory Tracking of a Nonholonomic Mobile Robot with Adaptive Neural Compensator

Flavio Capraro,¹ Francisco Guido Rossomando,¹ Carlos Soria,¹ and Gustavo Scaglia²

¹*Instituto de Automática-CONICET, Facultad de Ingeniería, Universidad Nacional de San Juan, Av. San Martín (Oeste) 1109, San Juan, Argentina*

²*Instituto de Ingeniería Química-CONICET, Facultad de Ingeniería, Universidad Nacional de San Juan, Av. San Martín (Oeste) 1109, San Juan, Argentina*

Correspondence should be addressed to Francisco Guido Rossomando; frosoma@inaut.unsj.edu.ar

Received 31 March 2017; Revised 12 July 2017; Accepted 24 July 2017; Published 19 September 2017

Academic Editor: Francisco Valero

Copyright © 2017 Flavio Capraro 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 design of sliding mode controllers (SMC) with adaptive capacity is presented. This control technique is formed by two cascaded SMC controllers, one of them having an adaptive neural compensator (ANC); both are put on a WMR (wheeled mobile robot). The mobile robot is divided into a kinematics and a dynamics structure; the first SMC controller acts only on the kinematic structure and the SMC with neural adaptive compensator on the other one. The dynamic SMC was designed applying an inverse dynamic controller and using the model dynamics of the WMR. The adaptive neural compensation (ANC) was used in order to reduce the control error caused by the dynamics variations but it conveys a residual approximation error, so a sliding part was designed to cancel such error. This technique allows achieving the control objective despite parameter variations and external disturbances that take place in the dynamics; on the other hand, the ANC can adjust its neural parameters to reduce the dynamics variations of the WMR and thus improve the trajectory tracking control. Problems of convergence and stability are treated and design rules based on Lyapunov's theorem are given.

1. Introduction

1.1. Previous Works. The problem of trajectory tracking is a classic control one; its importance lies in the actual applications that these algorithms have. Although some of these problems have been solved before, there is always the possibility of somehow improving the algorithms previously designed.

The biggest advantage in using SMC technique is the low sensitivity in front of changes in the process dynamics and external disturbances, which eliminates the necessity of an accurate model. Recent works about disturbance rejections and SMC applied to robots control are dealt with in papers such as [1], where a spherical robot using Omni wheels uses a cascade sliding mode control (CSMC) to control its position.

The structure of the CSMC is formed by a cascade combination of states of the dynamic model and a set

of sliding surfaces. This work shows that the problem of constant speed can be avoided in order to achieve the position control without adaptive control techniques. An application of SMC in WMR is presented in [2]; the trajectory tracking problem is investigated for a nonholonomic wheeled mobile robot (NWMR) with parameter uncertainties and external disturbances. In this strategy, combining the kinematic model with the dynamic model, the actuator voltage is employed as the control input, and the uncertainties are approximated by a fuzzy logic system. The control system is stable and the tracking position errors converge asymptotically to zeros. Simulation results demonstrate the effectiveness of the proposed strategy.

In similar form, Ashrafiuon et al. [3] presented a nonlinear sliding mode control law that is employed to stabilize the error dynamics. It is shown that the control law is uniformly asymptotically stable if unknown disturbances and modeling

uncertainties are bounded. The framework is applied to differential drive mobile robots, air vehicles operating in the vertical plane, and marine vehicles. Simulations are presented for models of in-house mobile robots and surface vessels subject to unknown disturbances.

Another design for SMC controllers is proposed by [4], who uses a controller based on linear algebra for mobile robots. The results presented show a good performance, while the oscillations in trajectory tracking are reduced.

In the work presented by [5], the tracking control of a two-wheeled mobile robot is considered to underpin the developed theoretical results based on SMC. Model-based tracking control of a wheeled mobile robot (WMR) is first transferred to a stabilization problem for the corresponding tracking error system, and then the developed theoretical results are applied to show that the tracking error system is globally asymptotically stable even in the presence of matched and mismatched uncertainties. These works [3–5] do not present adaptive capacity combined with SMC in trajectory tracking control.

In another work [6], a practical control law for wheeled mobile robots is proposed in order to improve the transient performance and decrease the tracking errors. The nominal system is governed using a controller derived under the back-stepping framework. Such a design can effectively reach the system's tracking objective and enhance robustness via properly configured parameters.

Another case of cascade-control algorithm based on a sliding mode is proposed by [7]. It is implemented for trajectory tracking control of a hydraulically driven 6-DOF (six degrees of freedom) robot manipulator. The proposed controller, which consists of two control loops, is applied to separate the hydraulic dynamics from the mechanical part so that the designed controller takes into account not only the mechanical dynamics but also the hydraulic dynamics of the manipulator. Experimental results demonstrate the satisfactory position tracking behavior of the parallel manipulator using a cascade controller based on a sliding mode.

In the meantime, a two-time-scale filter is applied to the system function to estimate the disturbances, essentially improving the system's precision. Experimental results demonstrated that the proposed control is practical for WMRs in tracking control. Instead in [8] the authors present a SMC for the whole structure of the mobile robot and the results show a good performance as well as asymptotic convergence demonstrated using Lyapunov's theory. But the control law is not trivial to implement for another kind of mobile robot. Similar research is shown in [9], where an adaptive neural network DSC (dynamic surface control) formulated on disturbance observer for the WMR (wheeled mobile robot) with parameters variations and external disturbances has been implemented. The observer is employed to estimate the unknown disturbances, and the neural approximation is used to identify the nonlinear model and all possible uncertainties of the system. Then, the Lyapunov theory is used to demonstrate the convergence of the proposed control system. Simulation and experiments results are included to demonstrate the adaptive capability and robustness of the proposed control approach.

In [10], an adaptive exponential SMC is proposed as a possible solution to minimize the chattering effect, external disturbances, and parameters variations in trajectory tracking of a wheeled mobile robot. The results obtained via experimentation show the effectiveness of the adaptive exponential SMC against disturbances and parameters variations.

Another important consideration in SMC is how to reduce the chattering effect; in [11] passivity-based adaptive and nonadaptive chattering-free sliding mode controllers are proposed assuming that the norm upper bound of the derivative of the sliding surface is available.

In [12], a method based on an integral SMC applied in the trajectory tracking of WMR is presented. The control technique is designed to solve the reaching phase problem with the elimination of matched disturbances and minimize the unmatched one. The proposed controller has two parts: first one is a high-level stabilization controller to compensate the known system and second one is a supervisory controller to improve the trajectory tracking in the presence of disturbances. The simulation results are based on the kinematics model of the nonholonomic mobile robot.

1.2. Contribution. In this work, an adaptive neural compensation (ANC) is considered. The ANC can be used by static controllers, obtaining as a result an adaptive control over the whole robotic system to be controlled. In this case, the ANC is applied over an inverse dynamic controller, and the complete control system shows a similar behavior of an adaptive controller system.

The paper [13] describes a method for the design of the discrete time dynamic compensation for a WMR with parameters variations, and the application of this method to trajectory tracking control. This method uses a single NN-RBF for dynamic compensation and is designed using Lyapunov's criterion. The stability analysis demonstrates a Uniformly Ultimately Bounded (UUB) control method.

Another work [14] shows a control technique based on a SMC and adaptive neural control applied over a WMR, but the sliding control is acting over the whole dynamic part of WMR.

This paper deals with the adaptive trajectory tracking control using SMC, and its design is based on a nominal robot dynamics. The control system has two parts. The first one uses the robot kinematics and SMC. The second one uses the WMR dynamics to design a nominal inverse control with ANC.

The main assumption of this work is that the most important parametric variations and uncertainties of the WMR appear in the dynamic part. The control technique uses a neural net based on RBF (radial basis functions) in combination with SMC and inverse dynamics. The control part based on inverse dynamics is designed on the nominal model of the WMR.

The adaptation laws of the ANC are obtained using Lyapunov's stability criterion, obtaining an asymptotic stability of the proposed control system.

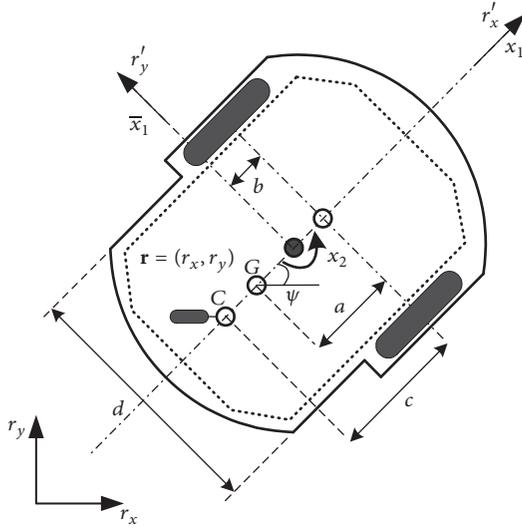


FIGURE 1: Mobile robot parameters distribution.

The combination of cascade SMC and ANC technique presents the following advantages:

- (1) The focus of this technique is the application of two sliding controllers in cascade to reduce problems on the perfect velocity matching of the WMR caused by parameters or dynamics variations. And it can be applied in WMR systems or nonlinear systems when their parameters vary or are unknown.
- (2) The ANC does not need to learn the whole robot structure; it adjusts the dynamic variations to approximate the nominal to the real dynamics of the WMR.
- (3) The ANC is easy to implement and it can be applied in different controllers without adaptive capacity, obtaining an adaptive control technique for the proposed control problem.

1.3. Organization. The paper is presented as follows. In Section 2, a description of the WMR system is given and the robot model (kinematic and dynamic parts) is proposed. Explicit expressions of the cascade SMC and the ANC compensation are derived in Sections 3 and 4, respectively. These expressions are used to obtain the neural adjustment law and prove the convergence of the proposed control system (Section 5). Several experimental results are presented in Section 6 for the evaluation of the proposed control technique and demonstrate the effectiveness of the adaptive capability. Finally, Section 7 concludes the paper.

2. Robot System Description

Figure 1 shows a typical structure of a nonholonomic mobile robot. It consists of a robot body (CPU, sensors, and motors) with two driving wheels mounted on the same axis and two motors for each one and a free rear wheel (or castor wheel). Both motors are the actuators of motion and orientation and provide the torque to the wheels.

For WMR control, the variables of interest in inertial Cartesian frame (XY plane) are r_x , r_y , ψ , x_1 , and x_2 , where

r_x , r_y are the coordinates of the point of interest \mathbf{r} , the angle ψ indicates the WMR direction, and the pair x_1 , x_2 is the linear and angular velocities, respectively, of the WMR.

The WMR parameters are described as follows:

G denotes the mass center of the WMR.

c is the distance between the free wheel (castor wheel) referred from the axis of the traction wheels.

a is the distance between the G point and the axis of the traction wheels.

The robot's model is divided taking into account the kinematics and the dynamics models, as shown in Figure 2. The kinematic and dynamic models were formulated for [15] and can be expressed as follows.

Kinematics Model

$$\begin{pmatrix} \dot{r}_x(t) \\ \dot{r}_y(t) \\ \dot{\psi}(t) \end{pmatrix} = \begin{pmatrix} \cos \psi(t) & -a \sin \psi(t) \\ \sin \psi(t) & a \cos \psi(t) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} \delta_{rx}(t) \\ \delta_{ry}(t) \\ 0 \end{pmatrix}. \quad (1)$$

Dynamic Model

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} \frac{\kappa_3}{\kappa_1} x_2^2(t) - \frac{\kappa_4}{\kappa_1} x_1(t) \\ -\frac{\kappa_5}{\kappa_2} x_1(t) x_2(t) - \frac{\kappa_6}{\kappa_2} x_2(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{\kappa_1} & 0 \\ 0 & \frac{1}{\kappa_2} \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} + \begin{pmatrix} \delta_{x1}(t) \\ \delta_{x2}(t) \end{pmatrix}. \quad (2)$$

The identified parameters vector $\boldsymbol{\kappa} = [\kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5, \kappa_6]^T$ is described in the Appendix and the model uncertainties of the WMR are the pair $(\delta_{rx}, \delta_{ry})$ and the pair $(\delta_{x1}, \delta_{x2})$, where the pair $(\delta_{rx}, \delta_{ry})$ is dependent on WMR friction and slip velocities and some of these uncertainties influences on mobile robots are described in [16]. On the other hand, the pair $(\delta_{x1}, \delta_{x2})$ is dependent on robot's parameters as load weight, mass, inertia, wheel diameter, tires properties (static and dynamic friction), electric motor, its servos, and others.

3. Sliding Mode Kinematics Controller

To obtain the kinematics controller, the robot's kinematics model is necessary to know. The robot's kinematics model (2) can be rewritten as follows:

$$\begin{aligned} \dot{\mathbf{r}}(t) &= \mathbf{L}(t) \mathbf{x}(t) + \boldsymbol{\delta}(t) \\ &= \begin{pmatrix} \cos \psi(t) & -a \sin \psi(t) \\ \sin \psi(t) & a \cos \psi(t) \end{pmatrix} \mathbf{x}(t) + \begin{pmatrix} \delta_{rx}(t) \\ \delta_{ry}(t) \end{pmatrix}. \end{aligned} \quad (3)$$

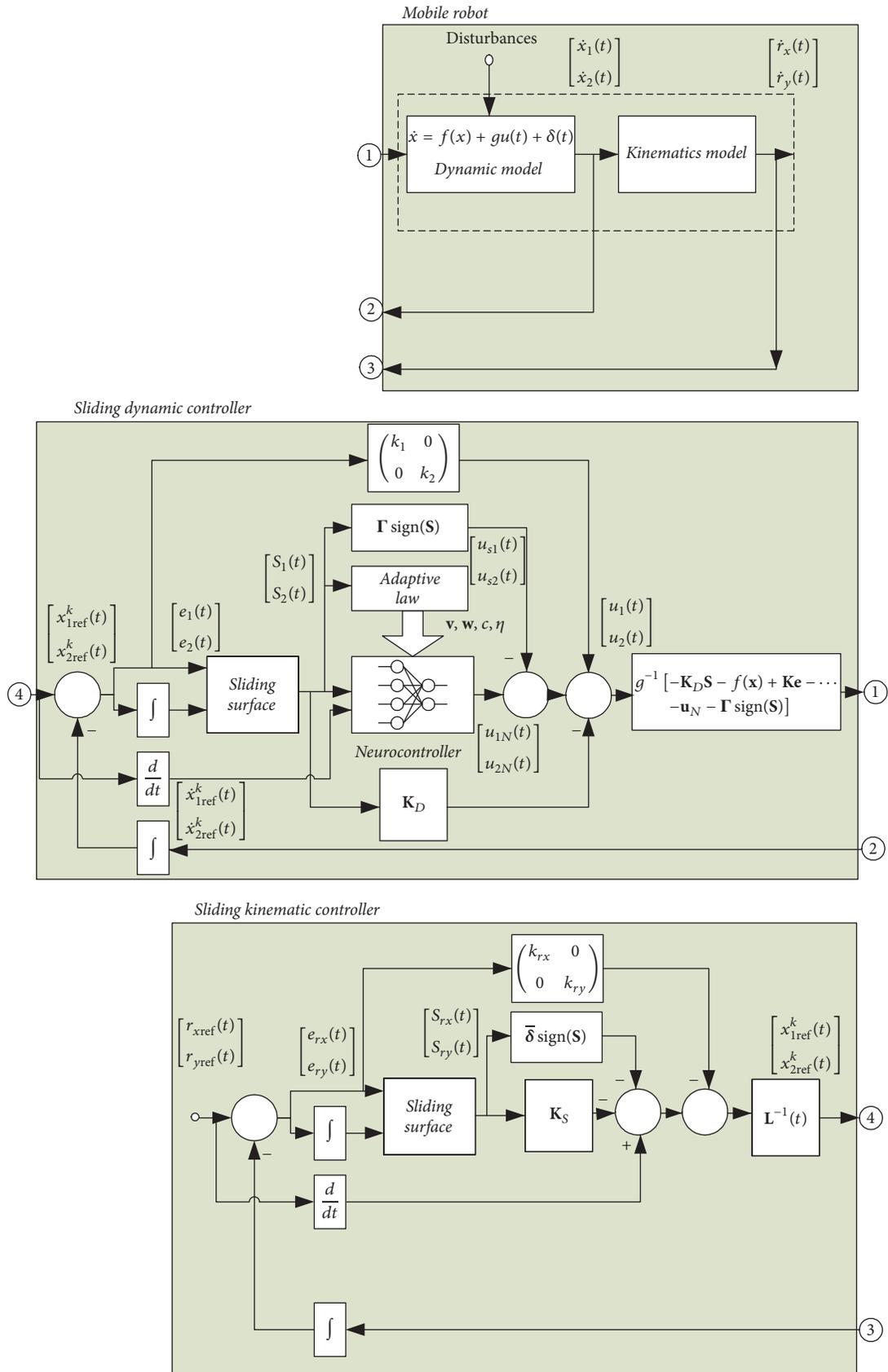


FIGURE 2: Structure control of CSMC with ANC.

For (3) to be controllable, it requires that $\mathbf{L}(t)$ must be nonsingular.

Assumption 1. $\boldsymbol{\delta}_r(t)$ is a nonmodeled kinematics vector which is bounded, where $\bar{\delta}$ is the upper bound.

$$\bar{\delta} = \sup_{t \in \mathcal{R}^+} |\boldsymbol{\delta}(t)|, \quad (4)$$

where

$$\begin{aligned} \mathbf{r}(t) &= (r_x(t) \ r_y(t))^T; \\ \mathbf{x}(t) &= (x_1(t) \ x_2(t))^T; \\ \boldsymbol{\delta}(t) &= (\delta_{rx}(t) \ \delta_{ry}(t))^T. \end{aligned} \quad (5)$$

They are the position output vector, the velocities input vector, and the vector of nonmodeled kinematics. The trajectory tracking error can be expressed as

$$\begin{aligned} \mathbf{e}_r(t) &= \mathbf{r}(t) - \mathbf{r}_{\text{ref}}(t) \\ &= (r_x(t) - r_{x\text{ref}}(t), r_y(t) - r_{y\text{ref}}(t))^T. \end{aligned} \quad (6)$$

The main objective of the control law, which can be obtained by the SMC, is the convergence of any state trajectory of nonlinear system to specified surface (defined by designer) into space state and maintain it in the subsequent time [17–19].

A sliding surface for a MIMO system can be obtained by position error (6) and defining $\mathbf{S}_r = (S_{rx}, S_{ry})^T$ as

$$\begin{aligned} S_{ri}(\mathbf{r}) &= \left(\frac{d}{dt} + k_{ri} \right) \int_0^t e_{ri}(\tau) d\tau \\ &= e_{ri}(t) + k_{ri} \int_0^t e_{ri}(\tau) d\tau \quad \text{for } i = x, y. \end{aligned} \quad (7)$$

The derivative of (7) is

$$\begin{aligned} \dot{S}_{ri}(\mathbf{r}) &= \frac{d}{dt} \left(e_{ri}(t) + k_{ri} \int_0^t e_{ri}(\tau) d\tau \right) \\ &= \dot{e}_{ri}(t) + k_{ri} e_{ri}(t). \end{aligned} \quad (8)$$

k_{ri} is a strictly positive constant and S_{ri} denotes a row of \mathbf{S}_r vector. In the design of the kinematics SMC, first, an ideal equivalent control law $\mathbf{x}_{\text{ref}}^k(t)$ must be determined. The ideal control law is obtained from (8) equaling zero as

$$\dot{\mathbf{S}}_r = \begin{pmatrix} \dot{S}_{rx} \\ \dot{S}_{ry} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (9)$$

Replacing (3) with (8), equaling zero, the following is obtained:

$$\begin{aligned} \dot{\mathbf{e}}_r(t) + \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t) \\ = (\mathbf{L}\mathbf{x}(t) + \boldsymbol{\delta}(t) - \dot{\mathbf{r}}_{\text{ref}}(t)) + \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t) \\ = 0. \end{aligned} \quad (10)$$

Similar case was studied in [17], defining the control law $\mathbf{x}_{\text{ref}}^k(t)$ that guarantees the sliding condition (9), which is composed of an ideal control $\mathbf{x}_{\text{ref}}^k(t) = \mathbf{L}^{-1}[-\text{diag}(k_{rx}, k_{ry})\mathbf{e}_r(t) - \boldsymbol{\delta}_r(t) + \dot{\mathbf{r}}_{\text{ref}}(t)]$, but the term $\boldsymbol{\delta}_r$ is unknown; taking into account Assumption 1, $\boldsymbol{\delta}_r$ is bounded. For control purposes, it can be approximated by $-\bar{\delta} \text{sign}(\mathbf{S})$. Thus, the control law is defined by

$$\begin{aligned} \mathbf{x}_{\text{ref}}^k(t) &= \mathbf{L}^{-1}(t) \left[-\mathbf{K}_S \mathbf{S}_r - \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t) \right. \\ &\quad \left. - \bar{\delta} \text{sign}(\mathbf{S}) + \dot{\mathbf{r}}_{\text{ref}}(t) \right]. \end{aligned} \quad (11)$$

Then the values of \mathbf{L}^{-1} , $\text{sign}(S_i)$, and \mathbf{K}_S are

$$\begin{aligned} \mathbf{L}^{-1} &= \begin{pmatrix} \cos \psi(t) & \sin \psi(t) \\ -\frac{1}{a} \sin \psi(t) & \frac{1}{a} \cos \psi(t) \end{pmatrix} \\ \text{sgn}(S_i) &= \begin{cases} 1 & \text{for } S_i > 0 \\ 0 & \text{for } S_i = 0 \\ -1 & \text{for } S_i < 0, \end{cases} \\ \mathbf{K}_S &= \begin{pmatrix} K_{Sx} & 0 \\ 0 & K_{Sy} \end{pmatrix}. \end{aligned} \quad (12)$$

The term $(-\mathbf{K}_S \mathbf{S})$ was included to guarantee the convergence of (11); now we define Lyapunov's function candidate like

$$V^k = \frac{\mathbf{S}_r^T \mathbf{S}_r}{2}, \quad (13)$$

where the superscript k denotes the analysis performed with respect to the kinematic controller; now differentiating (13) with respect to time and replacing (3) lead to

$$\begin{aligned} \dot{V}^k &= \mathbf{S}_r^T \dot{\mathbf{S}}_r = \mathbf{S}_r^T (\dot{\mathbf{e}}_r(t) + \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t)) \\ &= \mathbf{S}^T ((\dot{\mathbf{r}}(t) - \dot{\mathbf{r}}_{\text{ref}}(t)) + \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t)) = \dots \\ &= \mathbf{S}_r^T ((\mathbf{L}\mathbf{x}(t) + \boldsymbol{\delta}(t) - \dot{\mathbf{r}}_{\text{ref}}(t)) \\ &\quad + \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t)). \end{aligned} \quad (14)$$

Replacing (11) with (14),

$$\begin{aligned} \dot{V}^k &= \mathbf{S}_r^T \left((-\mathbf{K}_S \mathbf{S}_r + \boldsymbol{\delta}(t) - \bar{\delta} \text{sign}(\mathbf{S})) \right. \\ &\quad \left. + \text{diag}(k_{rx}, k_{ry}) \mathbf{e}_r(t) \right) \leq -\|\mathbf{K}_S\| \|\mathbf{S}_r\|^2 + \|\boldsymbol{\delta}\| \|\mathbf{S}_r\| \\ &\quad - \bar{\delta} \|\mathbf{S}_r\|. \end{aligned} \quad (15)$$

Then, the following is obtained:

$$\dot{V}^k \leq -\|\mathbf{K}_S\| \|\mathbf{S}_r\|^2 + \Delta_r \|\mathbf{S}_r\| < 0 \quad \text{for } S_i \neq 0. \quad (16)$$

From Assumption 1 $(\|\boldsymbol{\delta}_r\| - \bar{\delta}) = \Delta_r \leq 0$ and $\dot{V}^k(t)$ is negative definite; now integrating both sides of (16) results in

$$\begin{aligned} \int_0^{ts} (\|\mathbf{K}_S\| \|\mathbf{S}_r\|^2 + \Delta_r \|\mathbf{S}_r\|) dt &= - \int_0^{ts} \dot{V}^k(t) dt \\ &= V^k(0) - V^k(ts). \end{aligned} \quad (17)$$

Because $V^k(0)$ is bounded and $V^k(ts)$ is also bounded and nonincreasing, it can be obtained that

$$\lim_{ts \rightarrow \infty} \left[- \int_0^{ts} \dot{V}^k(t) dt \right] = V^k(0) - V^k(ts) < \infty. \quad (18)$$

By Barbalat's lemma [17], it can be shown that limit $\lim_{t \rightarrow \infty} [V^k(t)]$. That is, $S_r(t)$ tends to zero as t tends to infinity, being a stable control system. Therefore, the tracking error of the SMC based on WMR kinematics tends to zero as $S_r(t)$ tends to zero.

4. Sliding Mode Controller with Adaptive Neural Compensation

4.1. Problem Formulation. At this point, new solutions based on SMC and ANC for WMR considering parameter variations in its dynamics are proposed. The aim of this work is to reduce the trajectory tracking error produced by the uncertainties in the WMR model and parameter variations in the dynamics, but maintaining an adequate mechanical performance and improving the performance of the WMR during trajectory tracking.

The proposed strategy in this matter combines inverse dynamics control and adaptive neural compensation (ANC) with SMC, where the inverse dynamics controller is built with the known part of the robot's dynamics, the adaptive neural part compensates dynamic variations of the robot, and SMC eliminates the error introduced by the neural network.

This point aims to design a combination of SMC and ANC which ensure the global convergence of controlled variables for the control system for a given reference trajectory $\mathbf{x}_{\text{ref}}^k(t)$.

Now, the system model (2) is expressed in compact form as

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \Delta f(\mathbf{x}) + (g + \Delta g(\mathbf{x})) \mathbf{u}, \quad (19)$$

where

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}; \\ f(\mathbf{x}) &= \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{\kappa_3}{\kappa_1} x_2^2(t) - \frac{\kappa_4}{\kappa_1} x_1(t) \\ -\frac{\kappa_5}{\kappa_2} x_1(t) x_2(t) - \frac{\kappa_6}{\kappa_2} x_2(t) \end{pmatrix}; \\ g\mathbf{u}(t) &= \begin{pmatrix} g_1 & 0 \\ 0 & g_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\kappa_1} & 0 \\ 0 & \frac{1}{\kappa_2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\ &= \begin{pmatrix} \frac{u_1}{\kappa_1} \\ \frac{u_2}{\kappa_2} \end{pmatrix}; \end{aligned} \quad (20)$$

$$\Delta f(\mathbf{x}) + \Delta g(\mathbf{x}) \mathbf{u}(t) = \begin{pmatrix} \delta_{x1}(\mathbf{x}) \\ \delta_{x2}(\mathbf{x}) \end{pmatrix},$$

where $f_i(\cdot)$, $g_i(\cdot)$ denote row vectors of $f(\cdot)$ and $g(\cdot)$, respectively, $\mathbf{x} = [x_1, x_2]^T \in \mathfrak{R}^n$ is the state vector, and $\mathbf{u} = [u_1, u_2]^T \in \mathfrak{R}^n$ with $n = 2$ is the control action input vector.

Assumption 2. Dynamic variations in (19) produce another pair of additive functions represented by $\Delta f(\mathbf{x})$ and $\Delta g(\mathbf{x})$, which represent nonlinear unknown functions and are considered as uncertainties.

The WMR dynamics from (1) without its variation is given by

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g\mathbf{u}. \quad (21)$$

Assumption 3. The matrix \mathbf{g} is nonsingular; that is, \mathbf{g}^{-1} exists and has bounded norm; this is equivalent to assuming

$$\begin{aligned} \lambda_l(\mathbf{g}) &> \lambda_{\min} > 0, \\ \|\mathbf{g}\|_2 &= \lambda_p(\mathbf{g}) \leq \lambda_{\max} < \infty, \end{aligned} \quad (22)$$

where $\lambda_l(\mathbf{g})$ and $\lambda_p(\mathbf{g})$ are the lower and upper eigenvalues of \mathbf{g} , respectively.

4.2. Implementation of SMC with Adaptive Neural Compensation (ANC). The objective of this issue is to design a control law that combines SMC with adaptive neural compensation. And this control law must ensure the convergence of all controlled variables. The SMC with ANC which is designed based on the NN-RBF structure can deal with this kind of control problem [20].

For a good trajectory tracking, the output $\mathbf{x}(t)$ must follow the desired trajectory $\mathbf{x}_{\text{ref}}^k(t)$; therefore, the tracking error should converge to zero.

Now, we define the tracking error as

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_{\text{ref}}^k = (x_1 - x_{\text{ref}1}^k, x_2 - x_{\text{ref}2}^k)^T = (e_1, e_2)^T. \quad (23)$$

A sliding surface $\mathbf{S}(\mathbf{x}) = (S_1, S_2)^T$ for the WMR system can be written in terms of the control error,

$$\begin{aligned} S_i(\mathbf{x}) &= \left(\frac{d}{dt} + k_i \right) \int_0^t e_i(\tau) d\tau \\ &= e_i(t) + k_i \int_0^t e_i(\tau) d\tau \quad \text{for } i = 1, 2. \end{aligned} \quad (24)$$

The derivate of (24) is

$$\begin{aligned} \dot{S}_i(\mathbf{x}) &= \frac{d}{dt} \left(e_i(t) + k_i \int_0^t e_i(\tau) d\tau \right) \\ &= (\dot{x}_i - \dot{x}_{\text{ref}i}) + k_i e_i. \end{aligned} \quad (25)$$

To ensure that the state variables remain on the sliding surface, it is necessary for $\dot{S}_i(\mathbf{x})$ to equal zero,

$$\dot{S}_i(\mathbf{x}) = (f_i(\mathbf{x}) + g_i u_i - \dot{x}_{\text{ref}i}) + k_i e_i = 0. \quad (26)$$

Based on (26), the equivalent control law $\mathbf{u}(t)$ can be obtained, and in the same way of the previous section, we

included the term $K_{Di}S_i$ to guarantee the convergence, and the obtained control law is expressed as

$$u_i(t) = (g_i)^{-1} (-K_{Di}S_i + \dot{x}_{refi} - f_i(\mathbf{x}) - k_i e_i). \quad (27)$$

Now, it is necessary to define a control law for the nonlinear system (21), define a new control law \mathbf{u} that guarantees robust performance and the convergence, and also ensure the sliding condition [20]. This new control law is implemented by (27) and a discontinuous term $-\Gamma \text{sign}(\mathbf{S})$. It is defined by

$$u_i(t) = \kappa_i [-K_{Di}S_i + \dot{x}_{refi} - f_i(\mathbf{x}) - k_i e_i - \Gamma_i \text{sign}(S_i)]. \quad (28)$$

From (20), $g_i^{-1} = \kappa_i$ because g is a diagonal matrix and the term $\Gamma_i \text{sign}(S_i)$ is constituted by a strictly positive constant Γ_i and a discontinuity ($\text{sign}(S_i)$). The function $\text{sign}(\cdot)$ is defined by

$$\text{sign}(S_i) = \begin{cases} 1 & \text{for } S_i > 0 \\ 0 & \text{for } S_i = 0 \\ -1 & \text{for } S_i < 0. \end{cases} \quad (29)$$

To prove the convergence of (28), a Lyapunov's function candidate is defined as

$$V = \frac{1}{2} (\mathbf{S}^T \mathbf{S}) = \sum_{i=1}^2 \frac{1}{2} (S_i^2). \quad (30)$$

Differentiating (30) with respect to time is expressed by

$$\begin{aligned} \dot{V} &= \sum_{i=1}^2 S_i \dot{S}_i = \sum_{i=1}^2 S_i \left(\frac{d}{dt} e_i(t) + k_i e_i(t) \right) \\ &= \sum_{i=1}^2 S_i ((\dot{x}_i - \dot{x}_{refi}) + k_i e_i(t)) = \dots \\ &= \sum_{i=1}^2 S_i (f_i(\mathbf{x}) + g_i u_i - \dot{x}_{refi} + k_i e_i(t)), \end{aligned} \quad (31)$$

and replacing (28) with (31),

$$\begin{aligned} \dot{V} &= \sum_{i=1}^2 S_i \dot{S}_i = \sum_{i=1}^2 S_i (-K_{Di}S_i - \Gamma_i \text{sgn}(S_i)) \\ &\leq -\sum_{i=1}^2 (K_{Di}S_i^2 + \Gamma_i |S_i|) < 0. \end{aligned} \quad (32)$$

Equation (32) proves the convergence to zero of the designed control law (28). The next step is considering the model indicated in (19) with dynamic variations; the nonmodeled structure can be expressed as

$$\dot{x}_i = (f_i(\mathbf{x}) + \Delta f_i(\mathbf{x})) + ((g_i + \Delta g_i(\mathbf{x})) u_i). \quad (33)$$

Substituting the dynamic part of the WMR (33) into (26) and applying the proposed control action (28), the following is obtained:

$$\dot{S}_i(\mathbf{x}) = -K_{Di}S_i + \Delta f_i(\mathbf{x}) + \Delta g_i(\mathbf{x}) u_i - \Gamma_i \text{sign}(S_i). \quad (34)$$

In (34), the terms corresponding to $\Delta f_i(\mathbf{x})$ and $\Delta g_i(\mathbf{x})$ disturbances are undesired; to deal with this problem, the $u_{iN}(t)$ term with adaptive capacity is added, which would compensate $\Delta f_i(\mathbf{x})$ and $\Delta g_i(\mathbf{x})$ variations in the dynamics of the system. And the term $\Gamma_i \text{sign}(S_i)$ would eliminate the undesired errors (approximation error of NN-RBF and the uncertainties) that the adaptive term ($u_{iN}(t)$) cannot compensate.

Taking into account these considerations, from (28), the control law that combines SMC and ANC is expressed as

$$u_i(t) = \kappa_i [-K_{Di}S_i - f_i(\mathbf{x}) + \dot{x}_{refi} - k_i e_i - (u_{iN} - \Gamma_i \text{sign}(S_i))], \quad (35)$$

where u_{iN} is a compensation variable which can be approximated by the neural function and it is the main structure of the ANC; the compensation variable can be represented as function of the optimal neural parameters, indicated by

$$u_{iN}^* = \mathbf{v}_i^{*T} \boldsymbol{\varphi}^*(\boldsymbol{\zeta}, \mathbf{c}^*, \boldsymbol{\rho}^*) + \sum_{j=1}^2 (\mathbf{w}^{*T} \boldsymbol{\varphi}^*(\boldsymbol{\zeta}, \mathbf{c}^*, \boldsymbol{\rho}^*))_{ij} u_j + \varepsilon_{in} \quad i = 1, 2, \quad (36)$$

where $\mathbf{v}^* \in \mathfrak{R}^{(n \times m)}$, $\mathbf{w}^* \in \mathfrak{R}^{(n \times m)}$, and $\boldsymbol{\varphi}^* \in \mathfrak{R}^{(m \times 1)}$ are optimal parameter vectors of the weights \mathbf{v} , the input weights vector \mathbf{w} , and the radial basis functions $\boldsymbol{\varphi}$, respectively; \mathbf{c}^* and $\boldsymbol{\rho}^*$ are the optimal values of the respective vectors (centers \mathbf{c} and widths $\boldsymbol{\rho}$); and ε_n is the neural approximation error generated by the ANC, and it is bounded by $|\varepsilon_{\text{Max}}| \geq |\varepsilon_{in}| > 0$.

Other neural parameters of the ANC are defined by $\boldsymbol{\zeta}$ and m : the first one is the regressor vector and can be defined as $\boldsymbol{\zeta} = [\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{x}_{ref}^k(t), \mathbf{u}(t)]^T$ and the second one is the number of RBF neurons ($m = 5$) defined by empirical criterion [21]. The vector of RBF functions is $(\boldsymbol{\zeta}) = [\varphi_1(\boldsymbol{\zeta}), \varphi_2(\boldsymbol{\zeta}), \dots, \varphi_m(\boldsymbol{\zeta})]^T$; each vector component is defined by

$$\varphi_i^*(\boldsymbol{\zeta}, \mathbf{c}^*, \boldsymbol{\rho}^*) = \exp[-\rho_i^{*2} \|\boldsymbol{\zeta} - \mathbf{c}_i^*\|^2] \quad (37)$$

with $\mathbf{c}^* = [c_1^* \ c_2^* \ \dots \ c_n^*]^T$ in the center of the receptive field and $\boldsymbol{\rho}^*$ is the width of the Gaussian function.

The optimal parameters of (36) are constant and unknown. These parameters are used as analytical tool to estimate their real values. The estimate neural parameters are $\hat{\mathbf{v}}$, $\hat{\mathbf{w}}$, and $\hat{\boldsymbol{\varphi}}$, and they are estimated parameter vectors of \mathbf{v}^* , \mathbf{w}^* , and $\boldsymbol{\varphi}^*$, respectively; and $\hat{\mathbf{c}}$ and $\hat{\boldsymbol{\rho}}$ are the corresponding estimation of \mathbf{c}^* and $\boldsymbol{\rho}^*$, respectively.

Defining the neural parameter approximation error (weights) like $\tilde{\mathbf{v}} = \mathbf{v}^* - \hat{\mathbf{v}}$, $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and function error

like $\tilde{\varphi} = \varphi^* - \hat{\varphi}$, the neural compensation expressed in (36) can be written as

$$\begin{aligned} u_{iN} = & \tilde{v}_i^T \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}) + \tilde{v}_i^T \tilde{\varphi}(\zeta, \hat{c}, \hat{\rho}) + \tilde{v}_i^T \tilde{\varphi}(\zeta, \tilde{c}, \tilde{\rho}) \\ & + \tilde{v}_i^T \tilde{\varphi}(\zeta, \tilde{c}, \tilde{\rho}) + \dots + \sum_{j=1}^2 (\tilde{w}^T \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}))_{ij} u_j \\ & + \sum_{j=1}^2 (\tilde{w}^T \tilde{\varphi}(\zeta, \hat{c}, \hat{\rho}))_{ij} u_j \\ & + \sum_{j=1}^2 (\tilde{w}^T \tilde{\varphi}(\zeta, \tilde{c}, \tilde{\rho}))_{ij} u_j \\ & + \sum_{j=1}^2 (\tilde{w}^T \tilde{\varphi}(\zeta, \tilde{c}, \tilde{\rho}))_{ij} u_j + \varepsilon_{in}, \end{aligned} \quad (38)$$

where $\tilde{v}_i^T \hat{\varphi} + \tilde{v}_i^T \tilde{\varphi}$ and $\tilde{w}^T \hat{\varphi} + \tilde{w}^T \tilde{\varphi}$ represent the learning error of each neural term, and the terms (product of two errors) like $\tilde{v}_i^T \tilde{\varphi}$ and $\sum_{j=1}^2 (\tilde{w}^T \tilde{\varphi})_{ij} u_j$ can be considered into ε_{in} .

Taking into account [22, 23], the neural approximations can uniformly approximate any continuous function. Now, it is possible to consider that $\hat{v}_i \hat{\varphi} = \Delta f_i(x)$ and $\sum_{j=1}^2 (\tilde{w}^T \hat{\varphi})_{ij} u_j = \sum_{j=1}^2 \Delta g_{ij}(x) u_j$.

To obtain the closed loop equation, it is necessary to consider the proposed sliding surface (25), and replacing the robotic system (19) and also control law (35) with neural compensation (38), the closed loop equation becomes

$$\begin{aligned} \dot{S}_i = & -K_{Di} S_i - \tilde{v}_i^T \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}) - \tilde{v}_i^T \tilde{\varphi}(\zeta, \hat{c}, \hat{\rho}) \\ & - \sum_{j=1}^2 (\tilde{w}^T \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}))_{ij} u_j - \dots \\ & - \sum_{j=1}^2 (\tilde{w}^T \tilde{\varphi}(\zeta, \hat{c}, \hat{\rho}))_{ij} u_j - \varepsilon_{in} - \Gamma_i \text{sign}(S_i). \end{aligned} \quad (39)$$

Using a Taylor's series to approximate the function difference $\tilde{\varphi} = \varphi^*(\zeta, c^*, \rho^*) - \hat{\varphi}(\zeta, \hat{c}, \hat{\rho})$ and taking the point $c^* = \hat{c}$ and $\rho^* = \hat{\rho}$, the expansion can be described by

$$\begin{aligned} \varphi^*(\zeta, c^*, \rho^*) = & \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}) + \Theta^T \tilde{c} + \Lambda^T \tilde{\rho} \\ & + \mathbf{O}(\zeta, \tilde{c}, \tilde{\rho}), \end{aligned} \quad (40)$$

where $\mathbf{O}(\cdot)$ represents the high-order term in a Taylor's series approximation and Θ and Λ are derivatives of $\varphi^*(\zeta, c^*, \rho^*)$ with respect to c^* and ρ^* at $(\hat{c}, \hat{\rho})$. They are expressed as

$$\begin{aligned} \Theta^T = & \left(\frac{\partial \varphi(\zeta, c^*, \rho^*)}{\partial c^*} \right) \Bigg|_{\substack{c^* = \hat{c} \\ \rho^* = \hat{\rho}}}, \\ \Lambda^T = & \left(\frac{\partial \varphi(\zeta, c^*, \rho^*)}{\partial \rho^*} \right) \Bigg|_{\substack{c^* = \hat{c} \\ \rho^* = \hat{\rho}}}. \end{aligned} \quad (41)$$

From (40), the following is obtained:

$$\tilde{\varphi} = \Theta^T \tilde{c} + \Lambda^T \tilde{\rho} + \mathbf{O}(\zeta, \tilde{c}, \tilde{\rho}). \quad (42)$$

From (42), the high-order term $\mathbf{O}(\cdot)$ is bounded by

$$\begin{aligned} \|\mathbf{O}(\zeta, \tilde{c}, \tilde{\rho})\| = & \|\tilde{\varphi}(\cdot) - \Theta^T \tilde{c} - \Lambda^T \tilde{\rho}\| \leq \dots \\ \leq & \|\tilde{\varphi}(\cdot)\| + \|\Theta^T \tilde{c}\| + \|\Lambda^T \tilde{\rho}\| \\ \leq & b_1 + b_2 \|\tilde{c}\| + b_3 \|\tilde{\rho}\| \leq \mathbf{O}_{\text{Max}}, \end{aligned} \quad (43)$$

where b_1 , b_2 , and b_3 are constants used to determine the boundedness of $\mathbf{O}(\cdot)$, due to the fact that the radial basis functions (gauss bell functions) and its derivatives are bounded by b_1 , b_2 , and b_3 . Replacing (42) with (39), the following can be obtained:

$$\begin{aligned} \dot{S}_i = & -K_{Di} S_i - \tilde{v}_i^T \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}) - \tilde{v}_i^T (\Theta^T \tilde{c} + \Lambda^T \tilde{\rho} + \mathbf{O}) \\ & - \sum_{j=1}^2 (\tilde{w}^T \hat{\varphi}(\zeta, \hat{c}, \hat{\rho}))_{ij} u_j - \dots \\ & - \sum_{j=1}^2 (\tilde{w}^T (\Theta^T \tilde{c} + \Lambda^T \tilde{\rho} + \mathbf{O}))_{ij} u_j - \varepsilon_{in} \\ & - \Gamma_i \text{sign}(S_i). \end{aligned} \quad (44)$$

Now, considering all terms that contain high-order terms ($\mathbf{O}(\cdot)$) in (44) and including the neural approximation error, they can be expressed as " ε_{Max} ," and it is bounded by a constant, as shown in (45):

$$|\varepsilon_{\text{Max}}| = \left| \tilde{v}_i^T \mathbf{O} + \sum_{j=1}^2 (\tilde{w}^T \mathbf{O})_{ij} u_j + \varepsilon_{in} \right| \leq \Gamma_i \quad i = 1, 2. \quad (45)$$

5. Neural Parameters Tuning

To achieve a stable adjustment law, a Lyapunov's candidate function must be proposed and it must consider all neural parameters (error, neural weights, spreads, and centers); the proposed function is

$$V = \frac{1}{2} \left\{ \sum_{i=1}^2 \left[S_{\Omega i}^2 + \frac{\tilde{v}_i^T \tilde{v}_i}{\alpha_i} + \frac{\tilde{w}_i^T \tilde{w}_i}{\beta_i} \right] + \left(\frac{\tilde{c}^T \tilde{c}}{\gamma_1} + \frac{\tilde{\rho}^T \tilde{\rho}}{\gamma_2} \right) \right\}, \quad (46)$$

where $\alpha_i \in \mathfrak{R}^{1 \times 1}$, $\beta_i \in \mathfrak{R}^{1 \times 1}$, $\gamma_1 \in \mathfrak{R}^{1 \times 1}$, and $\gamma_2 \in \mathfrak{R}^{1 \times 1}$ are nonnegative constants, respectively. n is outputs number ($n = 2$) and m is neurons number ($m = 5$). In this analysis, to reduce any possible cause of chattering effect, a boundary layer thickness Ω_i is defined, and $S_{\Omega i}$ can be computed as follows:

$$\begin{aligned} S_{\Omega i} = & S_i - \Omega_i \text{sat} \left(\frac{S_i}{\Omega_i} \right) \\ \text{sat} \left(\frac{S_i}{\Omega_i} \right) = & \begin{cases} \text{sign}(S_i) & \text{if } |S_i| \geq \Omega_i \\ \left(\frac{S_i}{\Omega_i} \right) & \text{if } |S_i| < \Omega_i. \end{cases} \end{aligned} \quad (47)$$

In this analysis, the sign function was changed by a saturation function, in similar way as done by [14, 24]. This boundary layer (BL) smoothes out the control behavior and ensures that the system states remain within this layer

$$S_{\Omega_i} = \{e_i \in \mathbb{R} \mid S_i(e_i) \leq |\Omega_i|\}. \quad (48)$$

Now, the derivative of V (46) with respect to time is written in (49), noting that $S_{\Omega_i} = 0$ inside the BL and $\dot{S}_{\Omega_i} = \dot{S}_i$ outside. Then (49) can be expressed as

$$\begin{aligned} \frac{dV}{dt} = & \sum_{i=1}^2 \left[S_{\Omega_i} \frac{dS_i}{dt} + \alpha_i^{-1} \tilde{\mathbf{v}}_i^T \frac{d\tilde{\mathbf{v}}_i}{dt} + \beta_i^{-1} \tilde{\mathbf{w}}_i^T \frac{d\tilde{\mathbf{w}}_i}{dt} \right] \\ & + \left(\gamma_1^{-1} \frac{d\tilde{\mathbf{c}}^T}{dt} \tilde{\mathbf{c}} + \gamma_2^{-1} \frac{d\tilde{\mathbf{p}}^T}{dt} \tilde{\mathbf{p}} \right). \end{aligned} \quad (49)$$

Substituting (44) in (49), the following is obtained:

$$\begin{aligned} \frac{dV}{dt} = & \sum_{i=1}^2 \left[-K_{Di} S_{\Omega_i}^2 - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \hat{\boldsymbol{\Phi}}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \boldsymbol{\Theta}^T \tilde{\mathbf{c}} \right. \\ & - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \mathbf{O} \\ & - S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \hat{\boldsymbol{\Phi}}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}))_{ij} u_j - \dots \\ & - S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T (\boldsymbol{\Theta}^T \tilde{\mathbf{c}} + \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \mathbf{O}))_{ij} u_j - S_{\Omega_i} \varepsilon_m \\ & \left. - \Gamma_i |S_{\Omega_i}| + \alpha_i^{-1} \tilde{\mathbf{v}}_i^T \frac{d\tilde{\mathbf{v}}_i}{dt} + \beta_i^{-1} \tilde{\mathbf{w}}_i^T \frac{d\tilde{\mathbf{w}}_i}{dt} \right] + \dots \\ & + \left(\gamma_1^{-1} \frac{d\tilde{\mathbf{c}}^T}{dt} \tilde{\mathbf{c}} + \gamma_2^{-1} \frac{d\tilde{\mathbf{p}}^T}{dt} \tilde{\mathbf{p}} \right). \end{aligned} \quad (50)$$

Rearranging (50),

$$\begin{aligned} \frac{dV}{dt} = & \sum_i \left[-K_{Di} S_{\Omega_i}^2 - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \hat{\boldsymbol{\Phi}}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \boldsymbol{\Theta}^T \tilde{\mathbf{c}} \right. \\ & - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} - S_{\Omega_i} \tilde{\mathbf{v}}_i^T \mathbf{O} + \alpha_i^{-1} \tilde{\mathbf{v}}_i^T \frac{d\tilde{\mathbf{v}}_i}{dt} + \beta_i^{-1} \tilde{\mathbf{w}}_i^T \frac{d\tilde{\mathbf{w}}_i}{dt} \\ & - S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \boldsymbol{\Theta}^T \tilde{\mathbf{c}})_{ij} u_j - S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}})_{ij} u_j \\ & - S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \mathbf{O})_{ij} u_j - S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \hat{\boldsymbol{\Phi}}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}))_{ij} u_j \\ & \left. - S_{\Omega_i} \varepsilon_m - \Gamma_i |S_{\Omega_i}| \right] + \dots + \left(\gamma_1^{-1} \frac{d\tilde{\mathbf{c}}^T}{dt} \tilde{\mathbf{c}} + \gamma_2^{-1} \frac{d\tilde{\mathbf{p}}^T}{dt} \tilde{\mathbf{p}} \right). \end{aligned} \quad (51)$$

Now grouping terms in (51) leads to

$$\begin{aligned} \frac{dV}{dt} = & \sum_{i=1}^2 \left[-K_{Di} S_{\Omega_i}^2 + \tilde{\mathbf{v}}_i^T \left(-S_{\Omega_i} \hat{\boldsymbol{\Phi}}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) + \alpha_i^{-1} \frac{d\tilde{\mathbf{v}}_i}{dt} \right) \right. \\ & + \tilde{\mathbf{w}}_i^T \left(-S_{\Omega_i} \sum_{j=1}^2 \hat{\boldsymbol{\Phi}}_j(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) u_j + \beta_i^{-1} \frac{d\tilde{\mathbf{w}}_i}{dt} \right) - \dots \\ & \left. - S_{\Omega_i} \left(\tilde{\mathbf{v}}_i^T \mathbf{O} + \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \mathbf{O})_{ij} u_j + \varepsilon_m \right) - \Gamma_i |S_{\Omega_i}| \right] \\ & + \left(-\sum_{i=1}^2 \left[S_{\Omega_i} \tilde{\mathbf{v}}_i^T \boldsymbol{\Lambda}^T + S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \boldsymbol{\Lambda}^T)_{ij} u_j \right] \right. \\ & \left. + \gamma_2^{-1} \frac{d\tilde{\mathbf{p}}^T}{dt} \tilde{\mathbf{p}} + \dots \right. \\ & \left. + \left(-\sum_{i=1}^2 \left[S_{\Omega_i} \tilde{\mathbf{v}}_i^T \boldsymbol{\Theta}^T + S_{\Omega_i} \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \boldsymbol{\Theta}^T)_{ij} u_j \right] \right. \right. \\ & \left. \left. + \gamma_1^{-1} \frac{d\tilde{\mathbf{c}}^T}{dt} \tilde{\mathbf{c}} \right) \right]. \end{aligned} \quad (52)$$

Now $\dot{\tilde{\mathbf{v}}}$, $\dot{\tilde{\mathbf{w}}}$, $\dot{\tilde{\mathbf{c}}}$, and $\dot{\tilde{\mathbf{p}}}$ are selected as

$$\begin{aligned} \frac{d\tilde{\mathbf{v}}_i}{dt} &= \alpha_i S_{\Omega_i} \hat{\boldsymbol{\Phi}}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}), \\ \frac{d\tilde{\mathbf{w}}_i}{dt} &= \beta_i S_{\Omega_i} \sum_{j=1}^2 \hat{\boldsymbol{\Phi}}_j(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) u_j, \\ \frac{d\tilde{\mathbf{p}}^T}{dt} &= \gamma_2 \sum_{i=1}^2 S_{\Omega_i} \left[\tilde{\mathbf{v}}_i^T \boldsymbol{\Lambda}^T + \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \boldsymbol{\Lambda}^T)_{ij} u_j \right], \\ \frac{d\tilde{\mathbf{c}}^T}{dt} &= \gamma_1 \sum_{i=1}^2 S_{\Omega_i} \left[\tilde{\mathbf{v}}_i^T \boldsymbol{\Theta}^T + \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \boldsymbol{\Theta}^T)_{ij} u_j \right]. \end{aligned} \quad (53)$$

Considering (53) into (52), then (52) can be rewritten as

$$\begin{aligned} \frac{dV}{dt} = & \sum_{i=1}^2 \left[-K_{Di} S_{\Omega_i}^2 - \Gamma_i |S_{\Omega_i}| \right. \\ & \left. - S_{\Omega_i} \left(\tilde{\mathbf{v}}_i^T \mathbf{O} + \sum_{j=1}^2 (\tilde{\mathbf{w}}_i^T \mathbf{O})_{ij} u_j + \varepsilon_m \right) \right]. \end{aligned} \quad (54)$$

From (54), it follows that

$$\frac{dV}{dt} \leq \sum_i \left(-K_{Di} S_{\Omega_i}^2 + |\varepsilon_{\text{Max}}| |S_{\Omega_i}| - \Gamma_i |S_{\Omega_i}| \right) < 0. \quad (55)$$

TABLE 1: Mobile robot parameters.

Parameters	Pioneer 3DX	Pioneer 2DX	Units
κ_1	0.24089	0.3037	s
κ_2	0.2424	0.2768	s
κ_3	$-9.3603e^{-4}$	$-4.018e^{-4}$	s-m/rad ²
κ_4	0.99629	0.9835	
κ_5	$-3.725e^{-3}$	$-3.818e^{-3}$	s/m
κ_6	1.0915	1.0725	

We integrate both sides of (55) which is defined as

$$\begin{aligned} & \int_0^{t_s} \sum_i (K_{Di} |S_{\Omega i}|^2 + \Gamma_i |S_{\Omega i}| - |\varepsilon_{Max}| |S_{\Omega i}|) dt \\ & \leq - \int_0^{t_s} \left(\frac{dV}{dt} \right) dt = [V(0) - V(t_s)]. \end{aligned} \quad (56)$$

Because $V(0)$ is bounded, and $V(t_s)$ is also bounded and nonincreasing, the following can be obtained:

$$\lim_{t_s \rightarrow \infty} \left[- \int_0^{t_s} \left(\frac{dV}{dt} \right) dt \right] = V(0) - V(t_s) < \infty. \quad (57)$$

Therefore, by Barbalat's lemma [17], it can be shown that $\lim_{t_s \rightarrow \infty} (-dV(t)/dt) = 0$. That is, $S(t)$ tends to zero as t tends to infinity.

As a result, the proposed control is stable. Moreover, the control error of the dynamic part will converge to zero asymptotically according to $S(t) \rightarrow 0$.

From (53), considering $\dot{\mathbf{v}}_i^* = 0$, $\dot{\mathbf{w}}_i^* = 0$, $\dot{\mathbf{c}}^* = 0$, and $\dot{\mathbf{p}}^* = 0$, the neural adjusting rules are

$$\frac{d\hat{\mathbf{v}}_i}{dt} = -\alpha_i S_{\Omega i} \hat{\boldsymbol{\varphi}}(\boldsymbol{\zeta}, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}), \quad (58)$$

$$\frac{d\hat{\mathbf{w}}_i}{dt} = -\beta_i S_{\Omega i} \sum_{j=1}^2 \hat{\boldsymbol{\varphi}}_j(\boldsymbol{\zeta}, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) u_j, \quad (59)$$

$$\frac{d\hat{\boldsymbol{\rho}}^T}{dt} = -\gamma_2^{-1} \sum_{i=1}^2 S_{\Omega i} \left[\hat{\mathbf{v}}_i^T \boldsymbol{\Lambda}^T + \sum_{j=1}^2 (\hat{\mathbf{w}}_i^T \boldsymbol{\Lambda}^T)_{ij} u_j \right], \quad (60)$$

$$\frac{d\hat{\mathbf{c}}^T}{dt} = -\gamma_1^{-1} \sum_{i=1}^2 S_{\Omega i} \left[\hat{\mathbf{v}}_i^T \boldsymbol{\Theta}^T + \sum_{j=1}^2 (\hat{\mathbf{w}}_i^T \boldsymbol{\Theta}^T)_{ij} u_j \right]. \quad (61)$$

Equations (58)–(61) are addressed to the tuning online of the dynamic compensation. Equations (58) and (59) adjust the neural weights, and (60) and (61) are responsible for setting the widths and the centers of each RBF vector of the ANC.

6. Experimental Results and Discussion

This section analyzes the performance of the proposed control technique, a single experiment is executed, and the corresponding results are presented below. To implement the experiment, a pioneer 2DX is used. The robot's parameters are indicated in Table 1. The control action commands



FIGURE 3: Pioneer 2DX mobile robot.

of the WMR are linear and angular velocities. Figure 3 shows a pioneer 2DX, and it has an 800 MHz Intel-Pentium with 512 Mb of memory RAM onboard CPU in which the proposed control technique is implemented.

In the experiment, the odometric sensors are used to sense the robot position; this experiment tries to demonstrate the advantages of CSMC-AMC method with respect to other control techniques from the literature.

The experiment is performed in the following manner: the robot must follow a predetermined trajectory (in this case a pentagon), where the dynamic controller is calibrated with different parameters not corresponding to the pioneer 2DX robot on the nominal model part ($f_i(x)$ and $g_i(x)$ in (33)) (being the parameters of the 3DX pioneer robot).

The ANC (adaptive neural compensator) is activated at 50 sec. after initiating the experiment. Such compensator reduces the control error produced by the difference between the real and nominal dynamical structure, while the sliding part is used to eliminate the residual error added by the neural approximation, reducing the control error $e(t)$, and therefore the error position $e_r(t)$ too. The registered results in the experiment are presented in Figures 4–9.

The response driven by the proposed control technique shows a larger value of the trajectory error because the robot is far away from the reference trajectory in the initial moment. Since the ANC can compensate these phenomena through learning, it can be seen that after 50 secs. of the initial time, there is a transitory error due to the initial commutation of the ANC, but then the tracking errors are continuously reduced and a better tracking performance of the proposed control technique is obtained. Figure 6 shows the control actions and the WMR output speeds of the CSMC-ANC technique, while Figure 5 depicts the desired and followed trajectory by the WMR with CSMC-ANC.

In Figure 7, the trajectory errors for the experiments using the proposed CSMC-ANC technique to follow the desired reference trajectory are shown. The trajectory error is defined as the error norm of the difference between the instantaneous distance of the WMR and the desired reference. It does easily verify that the error at final time is less than the initial position error. Figures 8 and 9 show the neural weights evolution during the experiment. In both figures, it can be observed that the network starts to set the weights from 50 sec., that is, the instant when the ANC compensator is activated.

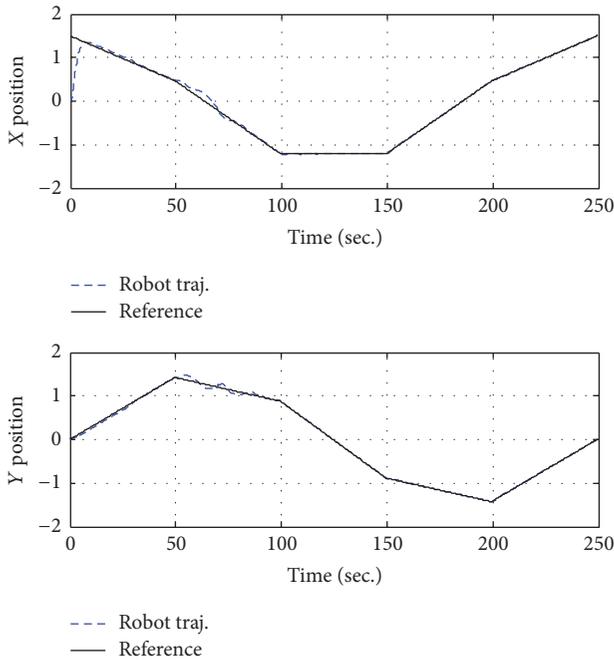


FIGURE 4: Reference signals of position and instantaneous mobile robot position.

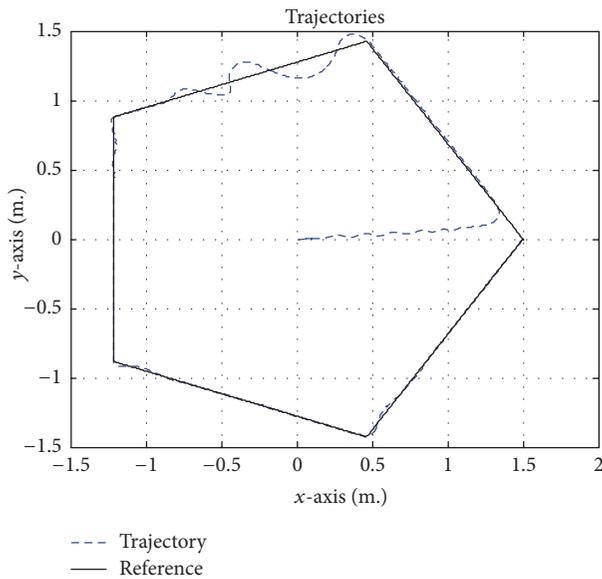


FIGURE 5: Desired and real trajectory followed by the WMR.

In this work, a control technique is presented that uses two SMC based controllers to control the trajectory of a mobile robot. The design of the proposed control technique was done to avoid the chattering effect. Therefore, the controller uses an adaptive part (ANC) to reduce the effect produced by the discontinuous function in the control action; this discontinuity is the main cause of this effect. And to ensure the reduction, the discontinuous function was replaced by the saturation function equation (47) smoothing the output response. Further, the cascade SMC with ANC is robust and efficient with respect to the dynamical variations

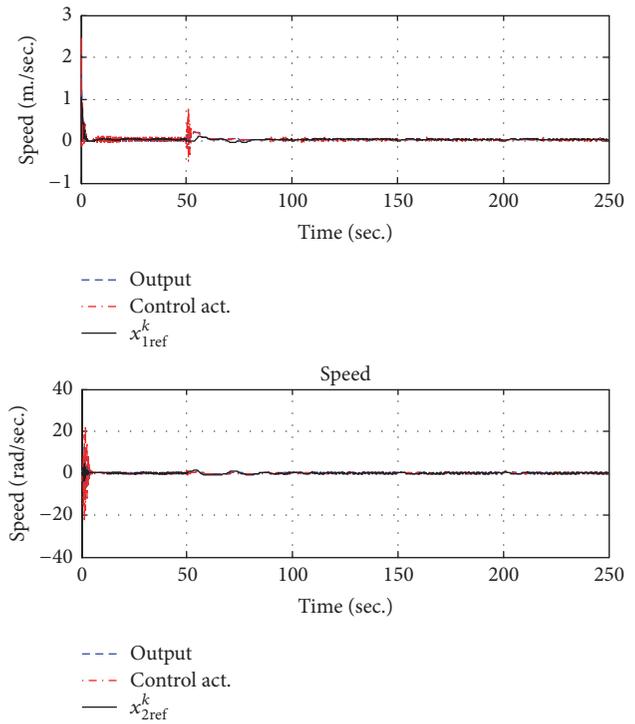


FIGURE 6: Control actions and output speeds of the WMR and by the cascade SMC with ANC.

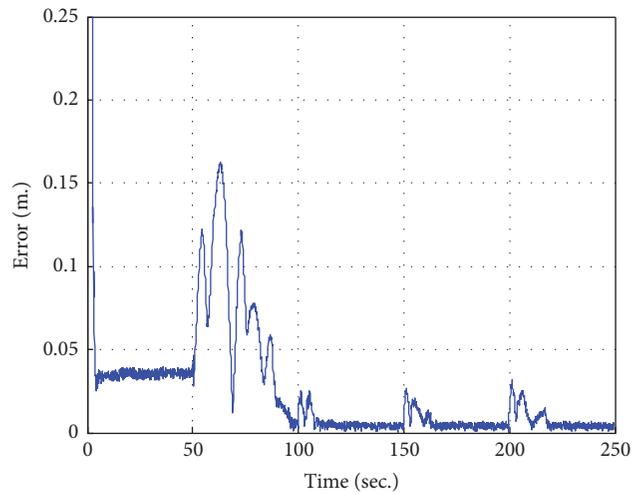


FIGURE 7: Trajectory error with cascade SMC and ANC activated after 50 sec.

of the WMR. In addition, it is effective in reducing the control error produced by the parameter variations or model uncertainties. The proposed controller designed in this research work for a WMR does not use the real dynamics for its implementation.

In the literature, there are many mathematical formulations for WMR dynamics not concerned with perfect accuracy, but the response of the cascade SMC-ANC as a control action to these dynamic variations as well as possible parameter variations of the WMR improves the system robustness.

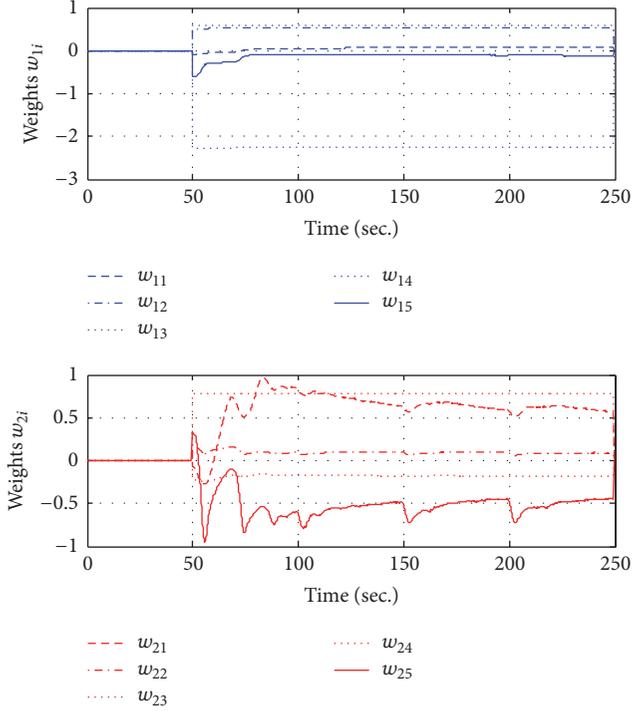


FIGURE 8: Evolution of the neural output weights (w) of the cascade SMC with ANC during the experiment.

The proposed control technique design methodology presented in this research work admits two significant characteristics, with important implications for WMR control design applications. The first one, the pair CSMC-ANC, is stable and its control error tends to zero (Section 6). And second one can be considered as a possible solution for the control of WMR and of any nonlinear systems, when the uncertainties are present in the robot dynamics or there are parameter variations in the robot.

7. Conclusions

In this research work, the problem of using two SMC controllers connected in cascaded has been considered. One of them uses an adaptive neural compensation (ANC). This technique was applied to nonlinear systems with nonmodeled uncertainties.

In the proposed technique, it is necessary to highlight that the controller design of the dynamic part was built using the combination of a feedback linearization controller and a SMC with an adaptive neural compensation (ANC). This control technique guarantees asymptotic convergence through a Lyapunov-based adaptive neural compensation.

The ANC in combination with SMC can reduce the control error caused by parameters variations of the WMR that affects the feedback linearization built with the known dynamics. This compensation modifies the control action to reduce the effects of dynamic variations of the WMR or model uncertainties. The main advantage of this technique is that the ANC does not need to learn all the dynamics

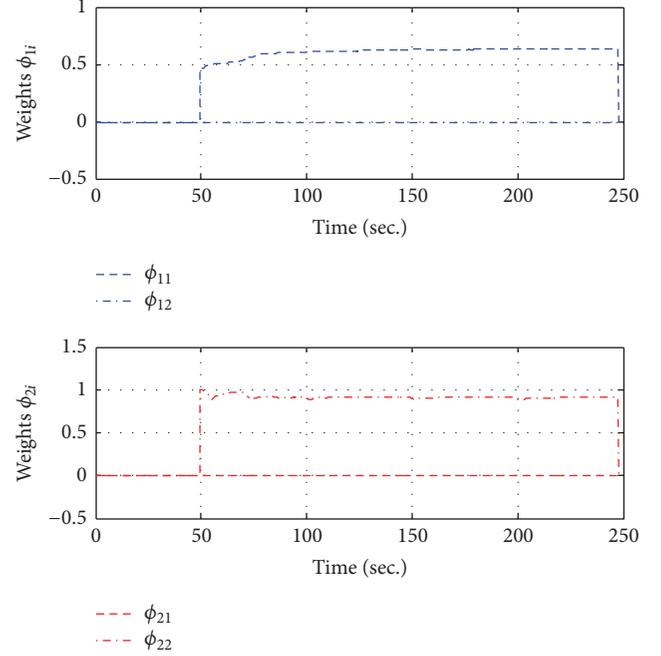


FIGURE 9: Evolution of the neural input weights (φ) of the cascade SMC with ANC during the experiment.

of the system structure, and it was designed to compensate only the dynamic variations and all possible uncertainties in the model. All other errors tend to zero by the action of the CSMC.

By experimental results, the proposed method has been shown to reduce trajectory errors more than the conventional SMC.

Appendix

Parameters Description

The identified parameters [15] can be described by

$$\begin{aligned}
 \kappa_1 &= \left(\frac{((R_a/k_a)(MR_t r + 2I_e) + 2rk_{DT})}{2rk_{PT}} \right), \\
 \kappa_2 &= \left(\frac{((R_a/k_a)(I_e d^2 + 2R_t r(I_z + Mb^2)) + 2rdk_{DR})}{2rdk_{PR}} \right), \\
 \kappa_3 &= \left(\frac{(R_a/k_a) MbR_t}{2k_{PT}} \right); \\
 \kappa_4 &= \left(\frac{(R_a/k_a)(k_a k_b/R_a + B_e)}{rk_{PT}} + 1 \right), \\
 \kappa_5 &= \left(\frac{(R_a/k_a) MbR_t}{dk_{PR}} \right); \\
 \kappa_6 &= \left(\frac{(R_a/k_a)(k_a k_b/R_a + B_e) d}{2rk_{PR}} + 1 \right).
 \end{aligned} \tag{A.1}$$

In these relations, M is the robot mass; r is the radius of the left and right wheels; k_b is equal to the electromotoric force constant multiplied by the reduction constant; R_a is the electric resistance; k_a is the constant of torque multiplied by the reduction constant; k_{PR} , k_{PT} , γ k_{DT} are positive constants; I_e and B_e are the moment of inertia and the viscous friction coefficient both belonging to the combination of motor, gearbox, and wheel; and R_t is the nominal radius of the wheel.

Conflicts of Interest

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

Acknowledgments

This work was funded by the National Council of Scientific and Technological Research (CONICET) PIO-CONICET Project no. 15020150100084CO.

References

- [1] K.-S. Huang, Y.-H. Lin, K.-B. Lin, B.-K. Lee, and C.-K. Hwang, "Cascade sliding mode control of a spherical wheel robot driven by Omni wheels," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '12)*, pp. 1607–1611, Xian, China, July 2012.
- [2] S. Peng and W. Shi, "Adaptive fuzzy integral terminal sliding mode control of a nonholonomic wheeled mobile robot," *Mathematical Problems in Engineering*, vol. 2017, Article ID 3671846, 12 pages, 2017.
- [3] H. Ashrafiuon, S. Nersesov, and G. Clayton, "Trajectory tracking control of planar underactuated vehicles," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1959–1965, 2017.
- [4] P. Proaño, L. Capito, G. Scaglia, A. Rosales, and O. Camacho, "Comparison of two linear algebra approaches for mobile robots controllers design," *International Journal of Automation and Control*, 2016.
- [5] J. Mu, X. G. Yan, S. K. Spurgeon, and Z. Mao, "Generalised regular form based SMC for nonlinear systems with application to a WMR," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6714–6723, 2017.
- [6] W. Sun, S. Tang, H. Gao, and J. Zhao, "Two time-scale tracking control of nonholonomic wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2059–2069, 2016.
- [7] H. Guo, Y. Liu, G. Liu, and H. Li, "Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode," *Control Engineering Practice*, vol. 16, no. 9, pp. 1055–1068, 2008.
- [8] F. G. Rossomando, C. Soria, and R. Carelli, "Neural network-based compensation control of mobile robots with partially known structure," *IET Control Theory and Applications*, vol. 6, no. 12, pp. 1851–1860, 2012.
- [9] S. Luo, S. Wu, Z. Liu, and H. Guan, "Wheeled mobile robot RBFNN dynamic surface control based on disturbance observer," *ISRN Applied Mathematics*, vol. 2014, Article ID 634936, 9 pages, 2014.
- [10] H. Mehrjerdi, Y. M. Zhang, and M. Saad, "Adaptive exponential sliding mode control for dynamic tracking of a nonholonomic mobile robot," *Intelligent Robotics and Applications*, vol. 7506, no. 1, pp. 643–652, 2012.
- [11] V. Parra-Vega and G. Hirzinger, "Chattering-free sliding mode control for a class of nonlinear mechanical systems," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 12, pp. 1161–1178, 2001.
- [12] A. Bessas, A. Benalia, and F. Boudjema, "Integral sliding mode control for trajectory tracking of wheeled mobile robot in presence of uncertainties," *Journal of Control Science and Engineering*, vol. 2016, Article ID 7915375, 10 pages, 2016.
- [13] F. G. Rossomando, C. Soria, and R. Carelli, "Autonomous mobile robots navigation using RBF neural compensator," *Control Engineering Practice*, vol. 19, no. 3, pp. 215–222, 2011.
- [14] F. G. Rossomando, C. Soria, and R. Carelli, "Sliding mode neuro adaptive control in trajectory tracking for mobile robots," *Journal of Intelligent Robotic Systems*, vol. 74, no. 3, pp. 931–944, 2014.
- [15] C. De La Cruz and R. Carelli, "Dynamic modeling and centralized formation control of mobile robots," in *Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics (IECON '06)*, pp. 3880–3885, Paris, France, November 2006.
- [16] F. Valero, F. Rubio, C. Llopis-Albert, and J. I. Cuadrado, "Influence of the friction coefficient on the trajectory performance for a car-like robot," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4562647, 9 pages, 2017.
- [17] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, vol. 199, Prentice-Hall, Englewood Cliffs, NJ, USA, 1991.
- [18] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory and Applications*, CRC Press, 1998.
- [19] V. I. Utkin, *Sliding Modes in Control Optimization*, Springer, Berlin, Germany, 2013.
- [20] C. H. Tsai, H. Y. Chung, and F. M. Yu, "Neuro-sliding mode control with its applications to seesaw systems," *IEEE Transactions on Neural Networks*, vol. 15, no. 1, pp. 124–134, 2004.
- [21] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Mathematical Problems in Engineering*, vol. 2013, Article ID 425740, 11 pages, 2013.
- [22] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [23] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [24] F. G. Rossomando, C. Soria, and R. Carelli, "Adaptive neural sliding mode compensator for a class of nonlinear systems with unmodeled uncertainties," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2251–2259, 2013.

Research Article

Influence of the Friction Coefficient on the Trajectory Performance for a Car-Like Robot

Francisco Valero,¹ Francisco Rubio,¹ Carlos Llopis-Albert,² and Juan Ignacio Cuadrado¹

¹Center of Technological Research in Mechanical Engineering, Universitat Politècnica de València, Building 5E, Valencia, Spain

²Department of Mechanical Engineering and M. Strength, Universitat Politècnica de València, Valencia, Spain

Correspondence should be addressed to Francisco Rubio; frubio@mcm.upv.es

Received 30 March 2017; Accepted 24 May 2017; Published 22 June 2017

Academic Editor: Rafael Morales

Copyright © 2017 Francisco Valero 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 collision-free trajectory planner for a car-like mobile robot moving in complex environments is introduced and the influence of the coefficient of friction on important working parameters is analyzed. The proposed planner takes into account not only the dynamic capabilities of the robot but also the behaviour of the tire. This planner is based on sequential quadratic programming algorithms and the normalized time method. Different values for the coefficient of friction have been taken following a normal Gaussian distribution to see its influence on the working parameters. The algorithm has been applied to several examples and the results show that computation times are compatible with real-time work, so the authors call them efficient generated trajectories as they avoid collisions. Besides, working parameters such as the minimum trajectory time, the maximum vehicle speed, computational time, and consumed energy have been monitored and some conclusions have been reached.

1. Introduction

Trajectory planning in Robotics has been a field of active work for decades, and, in recent years, its application to mobile robots and autonomous vehicles has gained particular relevance, given the importance of industrial use in the case of robots and safety in the area of vehicles [1, 2].

These uses introduce the need for nonholonomic restrictions when the robot is modelled, and in many cases this raises the issue of obtaining the trajectory as an optimization problem, as in Peng et al. [3]. In the case of autonomous vehicles, there are specific problems associated with driving on different road types (see Katrakazas et al. [4]).

Many authors have based the generation of trajectories on paths adjusted for different functions using, for example, Bezier curves [5, 6], B-splines, as in Elbanhawi et al. [7], clothoid curves (see Broggi et al. [8]), and polynomials [9, 10].

Techniques for trajectory planning seek continuous functions and their derivatives, with the consequent impact on the robot's kinematic and dynamic response, which will also be subject to specific restrictions in each case.

Effectiveness and operating capability of performing a prescribed trajectory for mobile robots, as for any other vehicle, will depend heavily on their dynamic behaviour.

In cases where the trajectory is generated along a previously obtained path without considering the dynamic constraints of the robot and its ability to transmit forces to the ground, its feasibility is not guaranteed, nor is the time required for it to be performed, so the efficiency of work done by the robot cannot be verified.

Significant simplifications are required to model the dynamics of the robot so that it can work with reduced computing times, and in four-wheel (car-type) robots it is usual to work with the "bicycle" model and different alternatives when considering tire behaviour, from neglecting them, as in Wang and Qi [11], to different approaches, as in Jeon et al. [12] and Cong et al. [13]. See also the dynamic model proposed by Staicu [14].

This paper presents a planner for obtaining trajectories for four-wheel robots while considering their dynamic properties. This planner, which is able to work efficiently, is based on solving optimization problems recursively.

The values of the coefficient of friction have been taken from a normal Gaussian distribution.

One objective of this paper is to determine the influence of the coefficient of friction on important operating parameters such as minimum trajectory time, maximum speed of the robot, computational time, and consumed energy.

The dynamic approach includes modelling the entire car-like robot, including tires, engine, brakes, and transmission system, with the aim of seeking feasible and efficient trajectories for the robot and analysing the influence of the coefficient of friction on the working parameters. This approach marks a clear difference with other planners that only include kinematic constraints, as in Simba et al. [15, 16]. Those planners either are conservative or do not guarantee the feasibility of the trajectories, as in Li and Shao [17] and Tokekar et al. [18].

Efficient computation time is achieved by simplifying the dynamic model, rapidly evaluating collisions, as explained in Rubio et al. [19], and generating a trajectory based on a path that is composed of parts which are defined by coordinates corresponding to polynomial functions of time. The efficiency of the trajectory is due to the fact that it uses the full dynamic capability of the robot. It starts from an initial minimum time trajectory that is adapted to environmental obstacles, the efficiency of which is shown in the examples analysed. A lot of examples (up to 100) have been solved by varying the coefficient of friction. The results have been monitored using the corresponding graphs in Section 5.

2. Dynamic Model of the Robotic Vehicle

In this section, we present the modelling of the dynamic behaviour of a mobile robot with tires. This model will be used in an iterative process that requires a sequence of optimization problems to be solved. Dynamic restrictions are essential to find feasible trajectories and are based on simplified but safe modelling.

Other authors have also used a simplified dynamic model, demonstrating its efficiency. For example, in Cazalilla et al. [20], the authors reduce the number of relevant dynamic parameters for the sake of simplicity of the model. Another example of simplified dynamic model can be found in Iriarte et al. [21]. To work in real time, Pastorino et al. [22] also resort to simplification in the modelling of the vehicle.

The robot used as an example has four wheels arranged symmetrically about its central axis, with the driving torque acting on the rear wheels, with braking on all wheels, and with front-wheel steering. The tire behaviour is crucial for determining the dynamic performance of the robot.

In addition, the following simplifying assumptions are considered:

- (i) There are no roll and pitch motions.
- (ii) There is no side load transfer.
- (iii) There are no aerodynamic effects.
- (iv) A “bicycle” type, planar model is used with three degrees of freedom and a restriction associated with the steering angle. The front wheels are simplified and

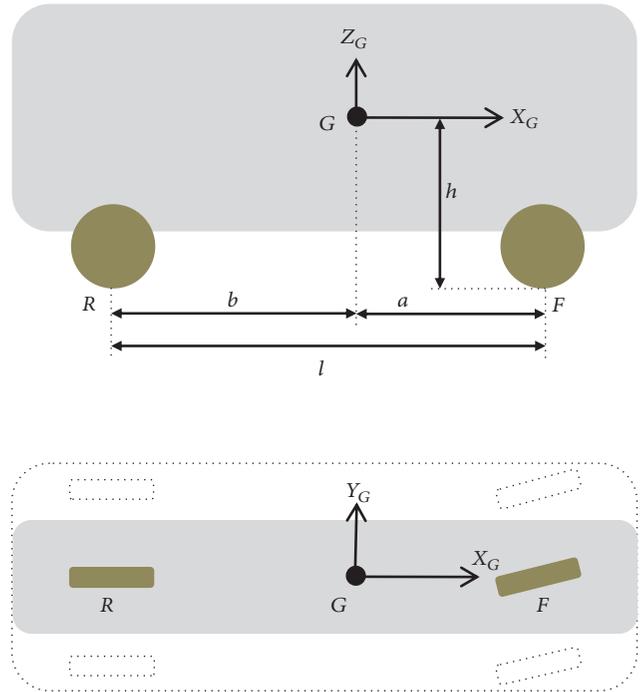


FIGURE 1: Geometry and its simplification.

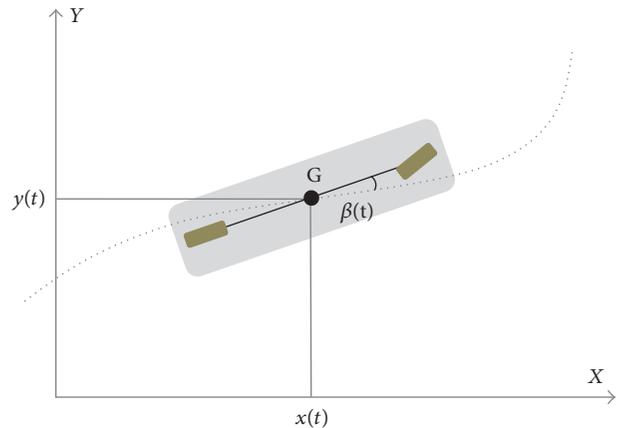


FIGURE 2: Robot configuration.

replaced by one that will account for the force exerted by the two. The same simplification applies to the rear wheels.

- (v) The steering angle is equal for each front wheel and corresponds to the steering angle of the “bicycle model.”

The geometry of the robot is shown at the top of Figure 1, in a side view, and the reduction to the “bicycle model” can be seen in the plan view.

2.1. Features of the Trajectory. A trajectory is established using time-dependent Cartesian components of the position (see Figure 2).

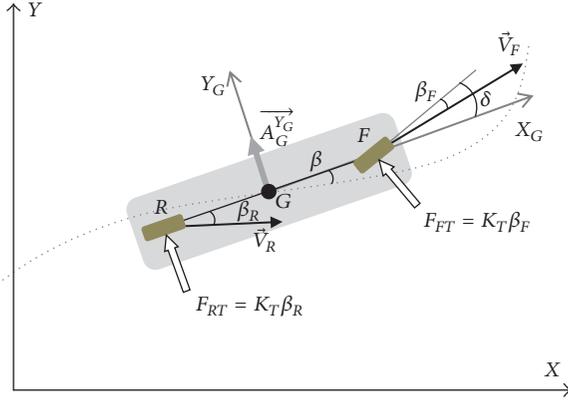


FIGURE 3: Robot kinematics.

The position (G) of the COG of the robot is

$$\begin{aligned} x &= a_x + b_x t + c_x t^2 + d_x t^3, \\ y &= a_y + b_y t + c_y t^2 + d_y t^3. \end{aligned} \quad (1)$$

By differentiating with respect to time, the COG's velocity and acceleration are obtained.

The orientation of the velocity of G is

$$\varphi(t) = \tan^{-1} \left(\frac{\dot{y}}{\dot{x}} \right). \quad (2)$$

Considering the small sideslip angles both in the vehicle and in the tires and β negligible compared to φ , the robot kinematics can be written as a function of (1) and its derivatives.

The sideslip angle in the rear wheel is obtained from the components of its velocity in local coordinates $X_G Y_G$ (see Figure 3):

$$\begin{aligned} \vec{V}_R^l &\approx (\dot{x}^2 + \dot{y}^2)^{1/2} \vec{i}_l \\ &+ \left((\dot{x}^2 + \dot{y}^2)^{1/2} \beta - b \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \right) \vec{j}_l, \end{aligned} \quad (3)$$

where

$$\beta_R \approx \beta - \frac{(\dot{x}\ddot{y} - \dot{y}\ddot{x})b}{(\dot{x}^2 + \dot{y}^2)^{3/2}}. \quad (4)$$

Proceeding in a similar way to that used to obtain (4), the sideslip angle in the front axle is

$$\beta_F \approx \delta - \beta - \frac{(\dot{x}\ddot{y} - \dot{y}\ddot{x})a}{(\dot{x}^2 + \dot{y}^2)^{3/2}}, \quad (5)$$

where δ is the steering angle.

Lateral acceleration in local coordinates (Y_G) direction is

$$A_G^{Y_G} \approx (-\ddot{x} \sin \varphi + \ddot{y} \cos \varphi) - (\ddot{x} \cos \varphi + \ddot{y} \sin \varphi) \beta. \quad (6)$$

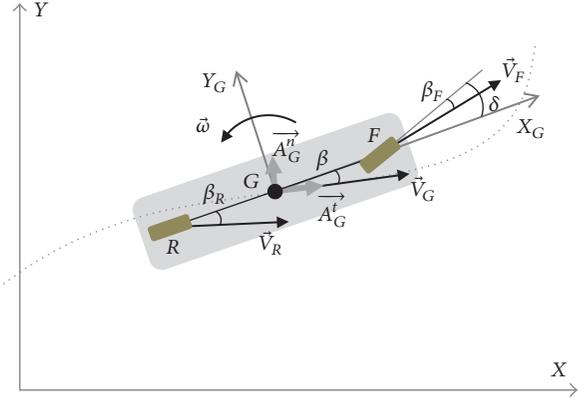


FIGURE 4: Lateral forces on tires.

Under the small sideslip hypothesis, it is usual to consider the lateral behaviour of the tires linearly, so as the front and rear tires are equal, the lateral forces are

$$F_{RT} = -K_T \beta_R \quad (7)$$

$$F_{FT} = -K_T \beta_F$$

with a direction normal to the rim and opposite to the sideslip (see Figure 4).

The equilibrium equation of the side forces of the robot considering small δ is

$$F_{RT} + F_{FT} = m A_G^{Y_G}. \quad (8)$$

The equation of moments is

$$a F_{FT} - b F_{RT} = I_z \dot{\omega}, \quad (9)$$

where I_z is the moment of inertia of the vehicle around an axis parallel to Z passing through G.

Equations (8) and (9) are a linear system that enables β and δ to be obtained as functions of the kinematics, tire's lateral stiffness, and the inertial characteristics of the vehicle. Using these values, β_R and β_F can be obtained from (4) and (5), respectively.

2.2. Forces Transmitted by the Tire. A force at the contact point of the tire with the ground is considered, so, in local coordinates,

$$\vec{F}_F = F_F^{X_G} \vec{i}_l + F_F^{Y_G} \vec{j}_l + F_F^{Z_G} \vec{k}_l, \quad (10)$$

$$\vec{F}_R = F_R^{X_G} \vec{i}_l + F_R^{Y_G} \vec{j}_l + F_R^{Z_G} \vec{k}_l.$$

In X_G direction, assuming a small steering angle, the following equilibrium equation is set:

$$F_F^{X_G} + F_R^{X_G} = m A_G^{X_G}, \quad (11)$$

where

$$A_G^{X_G} = \ddot{x} \cos(\beta + \varphi) + \ddot{y} \sin(\beta + \varphi). \quad (12)$$

In the front wheel,

$$\begin{aligned} A^{X_G} > 0 &\longrightarrow \\ F_F^{X_G} &= 0, \\ A^{X_G} \leq 0 &\longrightarrow \end{aligned} \quad (13)$$

$$F_F^{X_G} = 0.6mA^{X_G} + F_{F_r}^{X_G}$$

with losses due to rolling motion:

$$F_{F_r}^{X_G} = \mu_r F_F^{Z_G}, \quad (14)$$

where μ_r is considered constant because the velocity and sideslip angles are small.

The force on the rear wheel is

$$\begin{aligned} A^{X_G} > 0 &\longrightarrow \\ F_R^{X_G} &= mA^{X_G} + F_{R_r}^{X_G} + F_{F_r}^{X_G}, \\ A^{X_G} \leq 0 &\longrightarrow \end{aligned} \quad (15)$$

$$F_R^{X_G} = 0.4mA^{X_G} + F_{R_r}^{X_G}$$

with

$$F_{R_r}^{X_G} = \mu_r F_R^{Z_G}. \quad (16)$$

Equations (13) and (15) are in line with the torque applied to the rear wheels (for positive A^{X_G}), while the braking torque is divided between the front (60%) and rear wheels (40%) when the acceleration is negative.

Components in direction Y_G , for small sideslip angles, correspond to a linear lateral slip behaviour of the tire and are given by (7).

Components in direction Z_G , taking into account only the load transfer due to longitudinal acceleration (direction X_G), are

$$\begin{aligned} F_F^{Z_G} &= \frac{m}{l} (bg - A^{X_G}h), \\ F_R^{Z_G} &= \frac{m}{l} (ag + A^{X_G}h). \end{aligned} \quad (17)$$

A circle is set, limiting the friction force for the combined forces at each wheel, which prevents them from working with higher friction forces than those defined by μ_t .

$$\begin{aligned} \sqrt{(F_F^{X_G})^2 + (F_F^{Y_G})^2} &< \mu_t F_F^{Z_G}, \\ \sqrt{(F_R^{X_G})^2 + (F_R^{Y_G})^2} &< \mu_t F_R^{Z_G}. \end{aligned} \quad (18)$$

2.3. Driving Force. The car-like mobile robot has an electric engine with rear wheel driving and steering on the front wheels. The maximum torque that can be transmitted to the rear wheels is conditioned by the characteristics of the transmission and the limitations of the engine, so a driving force is achieved which is limited by the curve in Figure 5.

Using this restriction, both the integrity of the transmission and the possibility of achieving the best performance required of the engine transmission system are guaranteed.

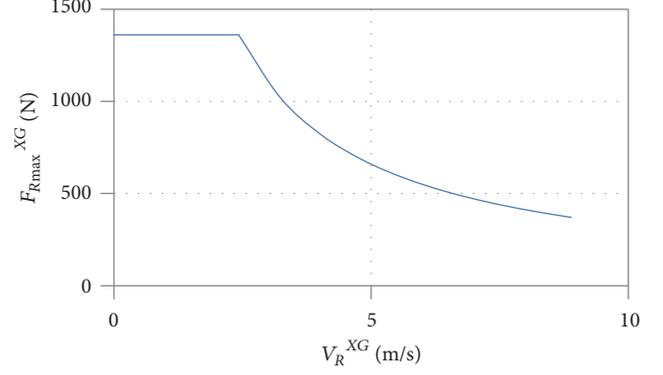


FIGURE 5: Driving force versus velocity.

3. Modelling of the Trajectory

The trajectory equations between successive configurations are

$$\begin{aligned} \forall t \in [0, t_j] \\ \downarrow \\ x_j &= a_{xj} + b_{xj}t + d_{xj}t^2 + e_{xj}t^3 \\ y_j &= a_{yj} + b_{yj}t + d_{yj}t^2 + e_{yj}t^3, \end{aligned} \quad (19)$$

where $j = 1, \dots, m-1$.

A robot configuration is given by the position and orientation of the center of gravity of the mobile robot in global coordinates.

The following conditions associated with the given configurations are considered in order to ensure continuity.

(i) *Position.* For each interval, the initial and final configurations are known, so $(4(m-1))$ equations are established:

$$\begin{aligned} x_j(0) &= x_j, \\ y_j(0) &= y_j, \\ x_j(t_j) &= x_{j+1}, \\ y_j(t_j) &= y_{j+1}. \end{aligned} \quad (20)$$

(ii) *Velocity.* The initial and final velocities of the trajectory must be zero, so four new equations are established:

$$\begin{aligned} \dot{x}_1(0) &= 0, \\ \dot{y}_1(0) &= 0, \\ \dot{x}_{m-1}(t_m) &= 0, \\ \dot{y}_{m-1}(t_m) &= 0. \end{aligned} \quad (21)$$

At each intermediate step, the final velocity of the previous interval must be equal to the start of the next, leading to $(2(m-2))$ equations:

$$\begin{aligned} \dot{x}_j(0) &= \dot{x}_{j-1}(t_{j-1}), \\ \dot{y}_j(0) &= \dot{y}_{j-1}(t_{j-1}). \end{aligned} \quad (22)$$

(iii) *Acceleration.* At each intermediate step, the final acceleration of the previous interval must be equal to the start of the next, leading to $(2(m-2))$ equations:

$$\begin{aligned} \ddot{x}_j(0) &= \ddot{x}_{j-1}(t_{j-1}), \\ \ddot{y}_j(0) &= \ddot{y}_{j-1}(t_{j-1}). \end{aligned} \quad (23)$$

Once the times needed to perform the trajectory between the different configurations for each interval are known, a linear system of equations is established, which makes it possible to obtain the coefficients of the cubic polynomials, providing an efficient solution using the normalized time method (see [23]).

3.1. Minimum Time Trajectory. The minimum time trajectory must also meet the following restrictions:

(i) The driving force cannot exceed the value determined by the limit transmitted to the wheel according to Figure 5:

$$F_R^{X_G}(t) \leq F_{R_{\max}}^{X_G} \quad \forall t \in [0, t_j] \quad \text{with } j = 1, \dots, m-1. \quad (24)$$

(ii) Forces on the tires must not exceed the limit of friction force at any point of the trajectory, so that, according to (18), the following equations are fulfilled:

$$\begin{aligned} \sqrt{(F_F^{X_G}(t))^2 + (F_F^{Y_G}(t))^2} &< \mu_t F_F^{Z_G}(t) \\ \sqrt{(F_R^{X_G}(t))^2 + (F_R^{Y_G}(t))^2} &< \mu_t F_R^{Z_G}(t) \end{aligned} \quad (25)$$

$$\forall t \in [0, t_j] \quad \text{with } j = 1, \dots, m-1.$$

(iii) The maximum velocity at any point of the trajectory is limited:

$$|\vec{V}_G(t)| < V_{\max}; \quad \forall t \in [0, t_j] \quad \text{with } j = 1, \dots, m-1, \quad (26)$$

where $\vec{V}_G(t)$ is the velocity of the center of gravity in global coordinates.

The minimum time is obtained by solving an optimization problem using the restrictions detailed above as variables of time for each interval of the trajectory, and the objective function is as follows:

$$\sum_{j=1}^{m-1} t_j = t_{\min}. \quad (27)$$

3.2. Obstacles. Patterned obstacles such as spheres, cylinders, and rectangular prisms are used (see [19]), so any geometry will be wrapped up by combinations of these basic obstacles.

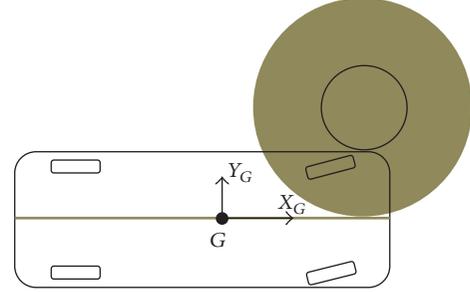


FIGURE 6: Wire-frame vehicle with grown obstacle.

3.3. Collision Detection. The wire-frame model of the robot with grown obstacle techniques (Figure 6) is used to check for collisions (see [24]).

3.4. Generation of the Trajectory without Collision. The problem posed is that of obtaining a feasible and efficient trajectory for a robot in an environment with static obstacles, which enables motion between two given configurations (c_i and c_f). An efficient trajectory is one which is close to the minimum time with a relatively low computational cost and respects the limitations of the robot dynamics. Obviously, the feasibility of the trajectory implies the absence of collisions.

The process proposed to solve the problem includes the following steps.

3.4.1. Calculation of the Initial Minimum Time Trajectory. Trajectory T_{\min} , linked to the sequence of configurations $C = \{c_i, c_f\}$, is obtained using the procedure described in Section 3.1.

3.4.2. Search for Collisions. The first configuration from T_{\min} which presents a collision c_c is identified, and then a previous and near configuration c_a is sought.

3.4.3. Generation of Adjacent Trajectories. Four configurations around c_a are generated (c_{aj} , $j = 1, \dots, 4$).

3.4.4. Generation of Offspring Trajectories. For each collision-free c_{aj} configuration obtained in the previous section, the offspring trajectory T_k is calculated and is added to the set of CT trajectories, where they are stored and ordered from smaller to longer time.

3.4.5. Trajectory Selection. Then, the minimum time trajectory T_1 of CT is selected and collisions are checked as detailed in Section 3.4.2. If T_1 has a collision, it is taken out of CT and the process returns to the step in Section 3.4.3 and is repeated until a solution is reached.

The proposed solution to the problem is an efficient trajectory without collisions which is an offspring trajectory resulting from the minimum time trajectory T_{\min} .

4. Case Study

The example shown in Figure 7 has been solved. The initial and final configurations are $C_i(x_1, y_1)$ and $C_f(x_2, y_2)$.

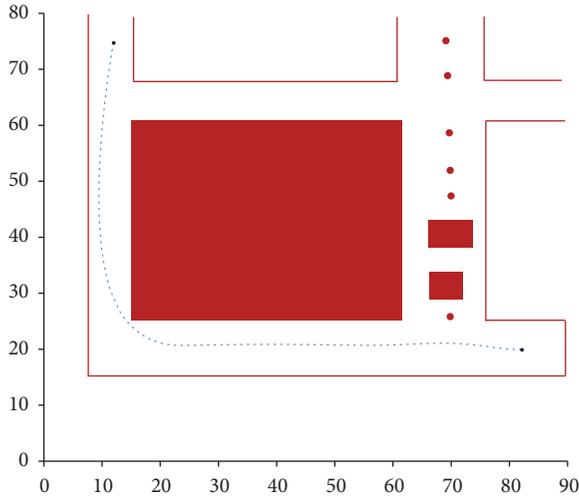


FIGURE 7: Initial and final car-like robot configuration.

The coefficient of adhesion between the ground and the tire may vary depending on several factors due to working conditions and time. These factors cover environmental conditions (such as wet, dry, or dirty road surfaces, moisture content, and temperature), technical characteristics of the tire, tire pressures, tire wear and maintenance standards, and so forth. The variations of the coefficient of adhesion have an important effect on the optimal trajectory time of the car-like autonomous robot, that is, the execution time of the robot task. Furthermore, they play a major role in how the robot performs its tasks appropriately, its productivity, and safety considerations.

This paper carries out an assessment of uncertainty based on the effects of the coefficient of adhesion regarding the optimal trajectory time by means of Monte Carlo simulations. Furthermore, the effects of the coefficient of adhesion over different variables are also analysed, which cover minimum trajectory time, maximum drive torque, maximum velocity, maximum braking torque, and consumed and dissipated energy. In addition, the computational time to generate an optimal trajectory is also provided.

4.1. Monte Carlo Simulations. The Monte Carlo simulations are based on the same case study and make use of the same car-like autonomous robot. 100 simulations have been run. All simulations share the same working conditions, initial and final configuration, and obstacles in the generated collision-free trajectories. The only changed parameter is the coefficient of adhesion. Values of the coefficient of adhesion are sampled from a Gaussian statistical distribution with a mean (m) of 0.25 and a standard deviation (s) of $m/3$, that is, 0.0833. In this way, we reproduce the actual values of the coefficient of adhesion which we find in real working conditions.

4.2. Results and Findings. Figure 8 shows the optimal trajectory time versus the coefficient of adhesion for an ensemble of a hundred simulations and also its trend line. Results show that, as expected, lower minimum times are achieved for higher values of the coefficient of adhesion due to the greater

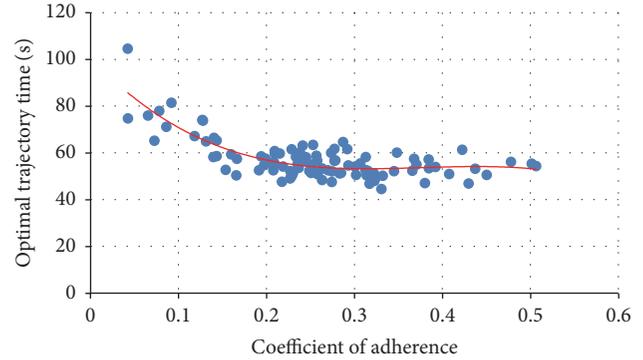


FIGURE 8: Optimal trajectory time (s) versus the coefficient of adhesion for an ensemble of a hundred simulations (blue dotted points) and also its trend line (red line).

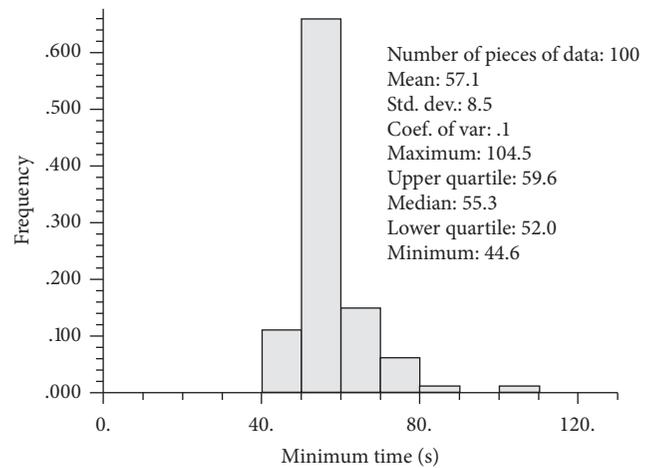


FIGURE 9: Frequency distribution and univariate statistics for the minimum trajectory time.

grip forces between tires and terrain. The minimum time is a discontinuous function due to numerical discrepancies in the coupling of the kinematic and dynamic model of the robot and the SQP method to solve the nonlinear optimization problem.

The frequency distribution and univariate statistics for the minimum trajectory time are depicted in Figure 9. The distribution has a mean of 57.1s and a standard deviation of 8.5s, where the minimum trajectory time ranges from 0.04s to 0.5s. These results show that the minimum trajectory time also follows a Gaussian distribution with a lower standard deviation in response to the Gaussian statistical distribution of the coefficient of friction. On the other hand, the results also show the importance of taking into account the uncertainty in the coefficient of adhesion and the worth of the presented methodology. This is even more critical when considering that a robot task may be repeated constantly, for instance, in an assembly line. In this case, the minimum time has a great influence in production and economic considerations. The uncertainty assessment carried out may help in defining an efficient scheduling process of the robot's tasks.

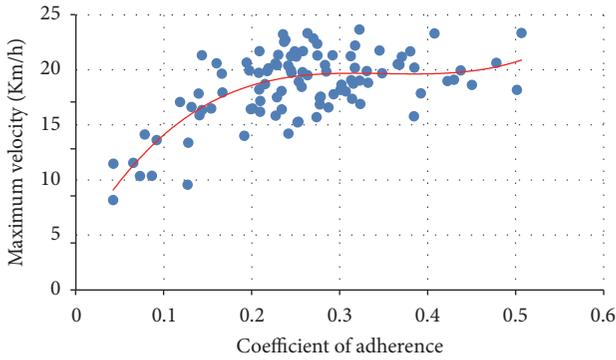


FIGURE 10: Maximum velocity (km/m) versus the coefficient of adherence for an ensemble of a hundred simulations (blue dotted points) and also its trend line (red line).

It means that we can use the Gaussian distribution of the coefficient of friction to predict, if necessary, the optimal trajectory times for the car-like robot to travel or do a task.

Figure 10 presents the maximum velocity versus the coefficient of adherence. It can be seen that higher velocities are achieved for greater grip forces, that is, for greater coefficient of adherence values. These results are in line with those obtained for the minimum trajectory time. Again, a wide dispersion in the velocity is obtained for the different simulations. The followed path will not always be the same and small radius of curvature limits the robot’s speed. Also note that the velocities values stabilize from the mean value of the Gaussian distribution of the coefficient of adherence (0.25). It is advisable to work with values of velocities as high as possible. Also bear in mind that velocities range from 8 km/h to 24 km/h. The maximum value is also limited by the driving torque and the kinematics and dynamics characteristics of the car-like robot.

Figure 11 represents the maximum driving torque on the rear wheels versus the coefficient of adherence, which shows that the driving torque is limited by the value of the coefficient of adherence. With small values of the coefficient of adherence, the driving torque cannot be bigger than the friction force in tires to avoid slippage. The maximum values are also limited by the maximum torque provided by the engine and the followed path which can be very sinuous. Initially, the path is sinuous and the engine cannot give the maximum driving torque. Also, at higher velocities, the driving torque decreases.

The maximum braking torque in the front wheels also strongly depends on the coefficient of adherence (Figure 12). It is shown that the higher the grip force is the higher the braking torque obtained is.

Figure 13 shows the computational time for the different simulations. They present low computational cost for all simulations with a mean of 1.8s, very small value that shows the efficiency of the algorithm. It can also be seen that the trend line slightly decreases for higher values of the coefficient of adherence due to better behaviour of those values in the nonlinear optimization procedure to obtain minimum time trajectories.

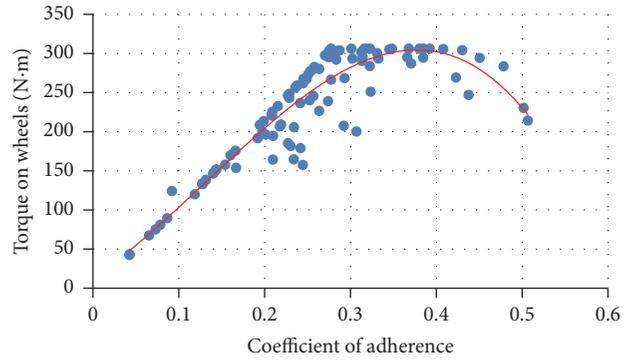


FIGURE 11: Maximum driving torque on rear wheels (N-m) versus the coefficient of adherence for an ensemble of a hundred simulations (blue dotted points) and also its trend line (red line).

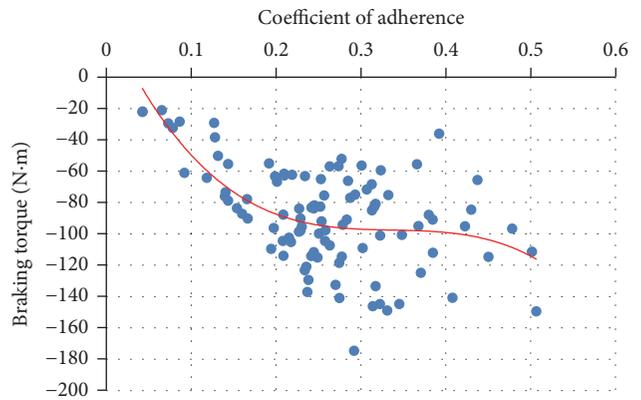


FIGURE 12: Maximum braking torque in the front wheels (N-m) with regard to the coefficient of adherence for an ensemble of a hundred simulations (blue dotted points) and also its trend line (red line).

Finally, Figure 14 presents the energy consumed and dissipated for the autonomous robot for the different simulations. It shows that the energy balance is equal to the energy consumed minus the energy dissipated. It also shows that the energy dissipated increases with the coefficient of adherence, while the energy balance remains almost constant for all simulations, with a mean value of 6678 (J).

5. Conclusions

The simplified dynamic model of the robotic vehicle makes it possible to achieve safe, collision-free, feasible trajectories while fulfilling the dynamic capabilities of the robot, with moderate computation times.

Efficient trajectories are obtained for complex environments with scattered obstacles, always respecting the limitations of the robot dynamics.

The methodology allows determining all the important working variables in the calculation of an optimal robot trajectory (i.e., minimum trajectory time, maximum velocity, torque on wheels, braking torque, computational cost, consumed and dissipated energy, and energy balance). Furthermore, the methodology allows dealing with uncertain environments and a wide range of real case studies by

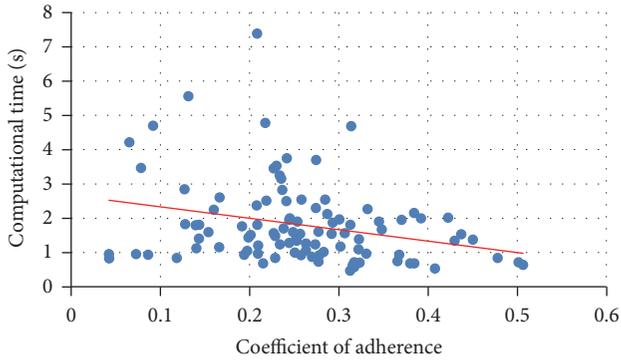


FIGURE 13: Computing time for the different simulations.

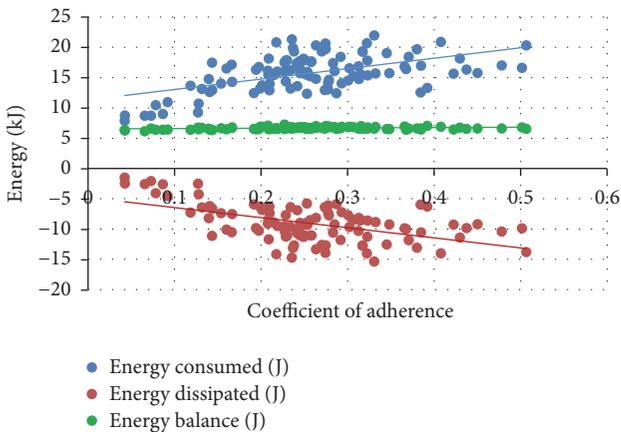


FIGURE 14: Consumed and dissipated energy for the autonomous robot and energy balance (J).

taking in consideration the kinematics and dynamics of the autonomous robot and providing optimal and collision-free trajectories.

From the graphs, it can be concluded that a relatively large coefficient of friction is good for most of the working parameters except for the consumed and dissipated energy. Because the energy is not recovered, high consumption is not recommended. Because we are using as objective function the time needed to perform the trajectory, this has an energetic cost.

Finally, based on this algorithm, a trajectory planner for robotic vehicles equipped with instrumentation to detect moving obstacles can be developed, since the computation times would allow trajectories to be recalculated according to the obstacles' motion. This is a topic that could be further developed in future work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness, which has funded the DPI2013-44227-R project.

References

- [1] D. Adninaite, R. Allsop, and G. Jost, "Ranking EU progress on road safety. 9th road safety performance index report," Tech. Rep., European Transport Safety Council (ETSC), 2015.
- [2] R. Siegwart and R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, MIT press, 2004.
- [3] J. Peng, W. Luo, W. Liu, W. Yu, and J. Wang, "A suboptimal and analytical solution to mobile robot trajectory generation amidst moving obstacles," *Autonomous Robots*, vol. 39, pp. 1–23, 2015.
- [4] C. Katrakazas, M. Quddus, W. H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research*, vol. 60, pp. 416–442, 2015.
- [5] J. Perez, J. Godoy, J. Villagni, and E. Onieva, "Trajectory generator for autonomous vehicles in urban environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [6] K. Kawabata, L. Ma, J. Xue, and N. Zhemg, "A path generation for automated vehicle based on Bezier curve and via-points," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2013.
- [7] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using B-spline curves," *Journal of Intelligent Robotic Systems*, vol. 80, pp. 23–56, 2015.
- [8] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli, "Autonomous vehicles control in the VisLab intercontinental autonomous challenge," *Annual Reviews in Control*, vol. 36, no. 1, pp. 161–171, 2012.
- [9] B. Nagy and A. Kelly, "Trajectory generation for car-like robots using cubic curvature polynomials," *In Field and Service Robots*, 2001.
- [10] M. Wang, T. Ganjineh, and R. Rojas, "Action Annotated Trajectory Generation for Autonomous Maneuvers on Structured Road Networks," in *Proceedings of the 5th International Conference on Automation, Robotics and Applications*, 2011.
- [11] D. Wang and F. Qi, "Trajectory planning for a four-wheel-steering vehicle," in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation Seoul*, 2001.
- [12] J. H. Jeon, R. V. Cowlagi, S. C. Peters et al., "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proceedings of the American Control Conference (ACC)*, 2013.
- [13] Y. Cong, O. Sawodny, H. Chen, J. Zimmermann, and A. Lutz, "Motion planning for an autonomous vehicle driving on motorways by using flatness properties," in *Proceedings of the 2010 IEEE International Conference on Control Applications part of 2010 IEEE Multi-Conference on Systems and Control*, IEEE, Yokohama, Japan, September 2010.
- [14] S. Staicu, "Dynamics equations of a mobile robot provided with caster wheel," *Nonlinear Dynamics*, vol. 58, no. 1-2, pp. 237–248, 2009.
- [15] K. R. Simba, N. Uchiyama, and S. Sano, "Real-time obstacle-avoidance motion planning for autonomous mobile robots," in *the Australian Control Conference*, 2014.
- [16] K. R. Simba, N. Uchiyama, and S. Sano, "Real-time smooth trajectory generation for nonholonomic mobile robots using Bézier curves," *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 31–42, 2016.
- [17] B. Li and Z. Shao, "Simultaneous dynamic optimization: a trajectory planning method for nonholonomic car-like robots," *Advances in Engineering Software*, vol. 87, pp. 30–42, 2015.

- [18] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, vol. 37, pp. 279–300, 2014.
- [19] F. Rubio, F. Valero, J. L. Sunyer, and A. Garrido, "The simultaneous algorithm and the best interpolation function for trajectory planning," *Industrial Robot*, vol. 37, no. 5, pp. 441–451, 2010.
- [20] J. Cazahilla, M. Vallés, V. Mata, M. Díaz-Rodríguez, and A. Valera, "Adaptive control of a 3-DOF parallel manipulator considering payload handling and relevant parameter models," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 5, pp. 468–477, 2014.
- [21] X. Iriarte, J. Ros, V. Mata, and A. Plaza, "Multibody model reduction by parameter elimination," in *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*, 2015.
- [22] R. Pastorino, E. Sanjurjo, A. Luaces, M. A. Naya, W. Desmet, and J. Cuadrado, "Validation of a Real-Time Multibody Model for an X-by-Wire Vehicle Prototype Through Field Testing," *Journal of Computational and Nonlinear Dynamics*, vol. 10, no. 3, Article ID 031006, 2015.
- [23] J. L. Suñer, F. Valero, J. J. Ródenas, and A. Besa, *Comparación Entre Procedimientos de Solución de La Interpolación Por Funciones Splines Para La Planificación de Trayectorias de Robots Industriales*, 2007.
- [24] F. Rubio, F. Valero, J. Sunyer, and V. Mata, "Direct step-by-step method for industrial robot path planning," *Industrial Robot: An International Journal*, vol. 36, no. 6, pp. 594–607, 2009.

Research Article

Adaptive Fuzzy Integral Terminal Sliding Mode Control of a Nonholonomic Wheeled Mobile Robot

Shuying Peng^{1,2} and Wuxi Shi^{2,3}

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

²Tianjin Key Laboratory of Advanced Technology of Electrical Engineering and Energy, Tianjin 300387, China

³School of Electrical Engineering and Automation, Tianjin Polytechnic University, Tianjin 300387, China

Correspondence should be addressed to Shuying Peng; psy19982002@163.com

Received 5 December 2016; Revised 13 March 2017; Accepted 16 March 2017; Published 5 April 2017

Academic Editor: Ashitava Ghosal

Copyright © 2017 Shuying Peng and Wuxi Shi. 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, the trajectory tracking problem is investigated for a nonholonomic wheeled mobile robot with parameter uncertainties and external disturbances. In this strategy, combining the kinematic model with the dynamic model, the actuator voltage is employed as the control input, and the uncertainties are approximated by a fuzzy logic system. An auxiliary velocity controller is integrated with an adaptive fuzzy integral terminal sliding mode controller, and a robust controller is employed to compensate for the lumped errors. It is proved that all the signals in the closed system are bounded and the auxiliary velocity tracking errors can converge to a small neighborhood of the origin in finite time. As a result, the tracking position errors converge asymptotically to zeros with faster response than other existing controllers. Simulation results demonstrate the effectiveness of the proposed strategy.

1. Introduction

A wheeled mobile robot (WMR) is an uncertain nonlinear MIMO dynamic system. When the WMR constrains the wheel's "pure rolling without slipping," it is also a typical kind of nonholonomic systems characterized by kinematic constraints. Such constraints are not integrable and can not be eliminated from the model equations. Given so many characteristics that are hard to handle, there has been tremendous research on the nonholonomic WMR (NWMMR) in the past few decades.

The trajectory tracking problem is one of the most popular problems on the WMR. With an assumption of "perfect velocity tracking," the initial kinematic controller for the NWMMR was designed in [1, 2]. However, such an assumption is difficult to hold in practice for the dynamic model of the NWMMR is neglected. Considering the kinematic model and the dynamic model of the NWMMR together, based on backstepping technique, Fierro and Lewis [3] presented a dynamical extension that combines a kinematic controller with a torque controller. In this method, it is assumed that the dynamic structure of the NWMMR and the parameters

are completely known. However, in practical WMRs, there exist parameter uncertainties and external disturbances. In addition, wheel skidding and slipping may happen. To overcome these difficulties, the torus shaped rear wheels were used for three WMRs in [4, 5]. The modeling and analysis were investigated to design the controller for the WMR in [6]. Meanwhile, a variety of nonlinear control techniques have been used by many researchers, such as adaptive control [7–11], robust adaptive control [12–14], adaptive fuzzy logic control [15–18], adaptive neural network control [19, 20], and sliding mode control [21–23], and several kinds of the aforementioned methodologies are integrated to solve this problem [24, 25].

One idea of some proposed literatures related to the trajectory tracking problem of the NWMMR is that an auxiliary velocity controller is designed for the kinematic model of the NWMMR to make the tracking position errors converge asymptotically to zeros, and a dynamic controller is designed for the dynamic model of the NWMMR to make the auxiliary velocity tracking errors as small as possible. Meanwhile, a robust controller is employed to compensate the total uncertainties. For instance, by virtue of the universal

approximation property of the fuzzy logic system (FLS) [26–31], a control structure combining a kinematic controller with a dynamic controller plus a fuzzy compensator was proposed in [15]. A complete control law based on a kinematic controller and an adaptive fuzzy sliding mode controller was developed for a NWMR in the presence of dynamic uncertainties as well in [25]. These dynamic controllers share a common idea of choosing the wheel torque as the control input. However, as stated in [17], the wheel is driven by the actuator in reality. Hence, the resulting electrically driven mobile robot (the robot kinematics, robot dynamics, and wheel actuator dynamics) is represented as a third-order system. So most of existing torque controllers designed with respect to the second-order, that is, the wheel actuator dynamics, have been neglected and might degrade the performance of the tracking control. Therefore, it is more reasonable to use the actuator voltage as the control input. For realizing the trajectory tracking of the NWMR with high performance, the wheel actuator dynamics are combining with the dynamics of the NWMR and the actuator voltage is employed as the control input in [8, 16–18]. All these dynamic controllers can guarantee that the auxiliary velocity tracking errors converge to an adjustable neighborhood of the origin as time goes to infinity. However, the finite time convergence of the auxiliary velocity tracking errors can not be guaranteed.

The terminal sliding mode control (TSMC), which was first proposed in [32, 33], is an effective scheme to guarantee the finite time convergence of the auxiliary velocity tracking errors. However, the initial TSMC may cause the singularity problem around the equilibrium [34], which would result in an unbounded control signal. In order to avoid this problem, a nonsingular terminal sliding mode control (NTSMC) was developed in [35–38]. The continuous nonsingular terminal sliding mode [36] has been extended into a class of MIMO nonlinear systems [39]. Furthermore, using integral operation, an integral terminal sliding mode control (ITSMC) was presented in [40, 41] for a class of first-order systems. Apart from finite time convergence and nonsingularity, in the ITSMC design, the system can also start on the integral terminal sliding mode surface from the initial time instant. Therefore, the reaching time to the sliding mode surface is eliminated.

Based on the previous results, this paper addresses the trajectory tracking problem for the NWMR with parameter uncertainties and external disturbances. Combining the kinematic model with the dynamic model, a control strategy is proposed which integrates an auxiliary velocity controller with an adaptive fuzzy integral terminal sliding mode controller. In this control strategy, using the universal approximation property of the FLS, the uncertainties are approximated by a fuzzy logic system and a robust controller is employed to compensate for the lumped errors. Meanwhile, instead of the wheel torque, the actuator voltage is employed as the control input. The main originality of the proposed control strategy is that the adaptive fuzzy integral terminal sliding mode controller can guarantee the finite time convergence of the auxiliary velocity tracking errors. It is proved that all the signals in the closed system are bounded and the auxiliary velocity tracking errors converge to a small

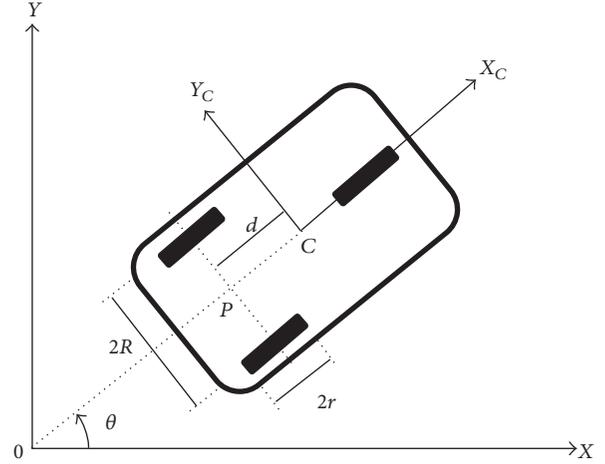


FIGURE 1: A wheeled mobile robot.

neighborhood of the origin in finite time. Therefore, the tracking position errors converge asymptotically to zeros with faster response than other existing controllers. Simulation results demonstrate the effectiveness of the proposed strategy.

The remainder of this paper is organized as follows. Section 2 reviews some basics of the model of the NWMR, the ITSMC, and the FLS. By use of the ITSMC and the FLS, a control strategy is proposed which integrates an auxiliary velocity controller with an adaptive fuzzy integral terminal sliding mode controller in Section 3. Section 4 gives simulation results to illustrate our results. Conclusions are given in Section 5.

2. Preliminaries

In this section, we will review some basics of the model of the NWMR, the ITSMC, and the FLS briefly.

2.1. Model of the Nonholonomic Wheeled Mobile Robot. We consider a typical example of the WMR, which is called Type (2,0) WMR in [6]. Such a WMR is composed of two deriving wheels and one passive wheel. The two deriving wheels are controlled independently by two actuators to achieve the motion and orientation, and the passive wheel prevents the robot from tipping over as it moves on a plane. Figure 1 describes the posture of the WMR in Cartesian coordinates. Both driving wheels with the same radius r are mounted on the same axis and separated by $2R$. The center of mass of the WMR is located at C , and P is located in the midpoint of the two driving wheels of the WMR. The distance between P and C is d . When the electrical part of the actuator is taken into account, the kinematic equation and the dynamic equation of the NWMR can be written as follows from [16, 19]:

$$\dot{q} = s(q) \vartheta, \quad (1)$$

$$\begin{aligned} \overline{M}(q) \dot{\vartheta} + \overline{V}(q, \dot{q}) \vartheta + \overline{F}(\vartheta) + \overline{\tau}_d \\ = \frac{NK_T}{R_a} \overline{B}u - \frac{N^2 K_T K_b}{R_a} \overline{B}X \vartheta, \end{aligned} \quad (2)$$

where

$$\begin{aligned} S(q) &= \begin{bmatrix} \cos \theta & -d \sin \theta \\ \sin \theta & d \cos \theta \\ 0 & 1 \end{bmatrix}, \\ \bar{M}(q) &= \begin{bmatrix} m & 0 \\ 0 & I - md^2 \end{bmatrix}, \\ \bar{V}(q, \dot{q}) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\ \bar{B} &= \frac{1}{r} \begin{bmatrix} 1 & 1 \\ R & -R \end{bmatrix}, \\ X &= \bar{B}^T. \end{aligned} \quad (3)$$

$q = [x, y, \theta]^T$, $C(x, y)$ is the coordinate of C in the global coordinate frame XOY , and θ is the orientation of the local coordinate frame $X_C Y_C$ attached on the WMR platform measured from X axis and is also called the heading angle of the WMR. $\vartheta = [\nu, \omega]^T$, where ν and ω are the linear velocity of the point P along the robot axis and angle velocity, respectively. $\bar{M}(q)$ is the inertia matrix, $\bar{V}(q, \dot{q})$ is the centripetal and Coriolis matrix, $\bar{F}(\vartheta) \in R^{2 \times 1}$ is the surface friction, and $\bar{\tau}_d \in R^{2 \times 1}$ denotes bounded unknown disturbances including unstructured unmodeled dynamics. N is the gear ration, K_T is the motor torque constant, K_b is the counter electromotive force coefficient, and R_a is the electric resistance. $u = [u_1, u_2]^T$ is the actuator voltage input vector.

Several properties of the NWMR are given as follows [19].

Property 1. The matrix $\bar{M}(q)$ is symmetric and positive definite.

Property 2. The matrix $\bar{M}(q)$ is bounded; that is, there exist positive constants m_1 and m_2 satisfying $m_1 \|x\|^2 \leq x^T \bar{M}(q)x \leq m_2 \|x\|^2$, for all $x \in R^2$.

Property 3. The matrix $\bar{M}(q) - 2\bar{V}(q, \dot{q})$ is skew symmetric resulting in the following characteristic: $x^T (\bar{M}(q) - 2\bar{V}(q, \dot{q}))x = 0$ for all $x \in R^2$.

In view of the dynamic model of the NWMR, (2) is a first-order system; the ITSM can be utilized so that the finite time convergence of the auxiliary velocity tracking errors of the NWMR is obtained.

2.2. Integral Terminal Sliding Mode. Now, a new form of the integral terminal sliding mode is defined as

$$s = x(t) - x(0) + \beta \int_0^t |x(\tau)|^\gamma \text{sign}(x(\tau)) d\tau, \quad (4)$$

where $x \in R$ is the system state variable, $\beta > 0$, and $0 < \gamma < 1$, which generalizes the integral terminal sliding mode [41]

$$s = x(t) + \beta \int_0^t x^{q/p}(\tau) d\tau, \quad (5)$$

where $\beta > 0$ and p and q are odd integers satisfying $p > q > 0$.

Remark 1. It is worthwhile to notice that the range of the power γ is larger than that of the power q/p . Meanwhile, by means of the basic theorem of differential and integral calculus [42], the integral terminal sliding mode (4) is continuous and differentiable although the absolute and signum operators are involved. Besides these properties, from (4), it is obvious that $s(0) = 0$ without the prior knowledge of the parameter β . This implies that the system starts on the integral terminal sliding mode surface (4) from the initial time instant much easily.

Furthermore, on the sliding surface, $s = 0$, which results in

$$\dot{x}(t) + \beta |x(t)|^\gamma \text{sign}(x(t)) = 0. \quad (6)$$

The finite time t_s that is taken from $x(0) \neq 0$ to $x(t_s) = 0$ is given by

$$t_s = \frac{1}{\beta(1-\gamma)} |x(0)|^{1-\gamma}. \quad (7)$$

As we stated previously, there exist parameter uncertainties and unknown disturbances in practical WMR. Taking these factors into account, an unknown nonlinear function is contained in the model of the NWMR. We will use the FLS to approximate this function.

2.3. Fuzzy Logic Systems. In this section, the FLS is discussed briefly. The basic configuration of an FLS consists of four components: fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifier. The fuzzy rule base is a collection of IF-THEN rules and the l th fuzzy rule is written as

$$R^l: \text{IF } x_1 \text{ is } F_1^l \text{ and } \dots \text{ and } x_n \text{ is } F_n^l, \text{ THEN } y \text{ is } G^l,$$

where F_i^l and G^l are fuzzy sets, associating with fuzzy membership functions $\mu_{F_i^l}(x_i)$ and $\mu_{G^l}(y)$, respectively, $i = 1, \dots, n$, $l = 1, \dots, m$, m is the number of rules.

Based on these fuzzy IF-THEN rules, the FLS performs a mapping from an input vector $x = [x_1, \dots, x_n]^T \in R^n$ to an output variable $y \in R$. If we use the strategy of singleton fuzzifier, product inference, and center-average defuzzifier, the output of the FLS can be defined as follows:

$$y(x) = \frac{\sum_{l=1}^m y^l \left(\prod_{i=1}^n \mu_{F_i^l}(x_i) \right)}{\sum_{l=1}^m \prod_{i=1}^n \mu_{F_i^l}(x_i)}, \quad (8)$$

where y^l is the point in G^l at which $\mu_{G^l}(y)$ obtains its maximum value 1.

For simplicity, $y(x)$ can be written in the following compact form:

$$y(x) = \hat{\theta}^T \xi(x) := \hat{f}(x | \hat{\theta}), \quad (9)$$

where $\hat{\theta} = [y^1, \dots, y^m]^T$ is called the unknown parameter vector which is to be updated and $\xi(x) = [\xi^1(x), \dots, \xi^m(x)]^T$ is called the fuzzy basis function vector, $\xi^l(x) = \prod_{i=1}^n \mu_{F_i^l}(x_i) / \sum_{l=1}^m \prod_{i=1}^n \mu_{F_i^l}(x_i)$, $l = 1, 2, \dots, m$.

Lemma 2 (see [31]). Let $f(x)$ be a continuous function defined on a compact set Ω . Then, for any constant $\varepsilon > 0$, there exists a fuzzy system (9) such that $\sup_{x \in \Omega} |f(x) - \hat{f}(x | \hat{\theta})| \leq \varepsilon$.

3. Controller Design

It is easy to see that the posture of the NWMR $q(t) = [x(t), y(t), \theta(t)]^T$ satisfies the following equations from (1):

$$\begin{aligned} \dot{x} &= v \cos \theta - \omega d \sin \theta, \\ \dot{y} &= v \sin \theta + \omega d \cos \theta, \\ \dot{\theta} &= \omega. \end{aligned} \quad (10)$$

It is assumed that the reference trajectory $q_r(t) = [x_r(t), y_r(t), \theta_r(t)]^T$ is generated by a reference NWMR with the kinematic equation as (10):

$$\begin{aligned} \dot{x}_r &= v_r \cos \theta_r - \omega_r d \sin \theta_r, \\ \dot{y}_r &= v_r \sin \theta_r + \omega_r d \cos \theta_r, \\ \dot{\theta}_r &= \omega_r. \end{aligned} \quad (11)$$

The objective of the trajectory tracking control is to design a strategy such that $q(t)$ converges asymptotically to $q_r(t)$, while all signals in the derived closed-loop system remain bounded. In this study, an auxiliary velocity controller ϑ_c is designed for the kinematic model (1) to meet the control objective. Then, the actuator voltage control input u is designed for the dynamic model (2) such that ϑ converges to ϑ_c which is designed at the first step in finite time.

Remark 3. As pointed out in [18], the classical auxiliary velocity controller [1] adopted in [3, 16, 18, 19, 25] can only guarantee that $q(t)$ converges asymptotically to $q_r(t)$ when d equals zero. However, d does not equal zero in general. Therefore, the reference point of the practical NWMR is not in accordance with the desired point of the reference NWMR, which results in incomplete tracking of the posture. In this paper, we modify the kinematic model of the reference NWMR as (11) and adopt another auxiliary velocity controller [43].

3.1. An Auxiliary Velocity Controller Design. We define the tracking position errors as the difference between the center of mass C of the NWMR and the desired point of the reference NWMR as follows [16, 19]:

$$e_p = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \quad (12)$$

The first derivative of the error yields

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 - d \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} v_r \cos e_3 - \omega_r d \sin e_3 \\ v_r \sin e_3 + \omega_r d \cos e_3 \\ \omega_r \end{bmatrix}. \quad (13)$$

Therefore, the objective of this study becomes the design of an auxiliary velocity controller to make the tracking position errors asymptotically converge to zeros. In this study, according to [43], the auxiliary velocity controller is designed as

$$\vartheta_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 (e_1 + d - d \cos e_3) \\ \omega_r + k_3 v_r (e_2 - d \sin e_3) + k_2 \sin e_3 \end{bmatrix}, \quad (14)$$

where $k_1, k_2, k_3 > 0$ are design parameters.

Substituting (14) into (13), the closed-loop kinematic equation can be written as

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 - d \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_r \cos e_3 + k_1 (e_1 + d - d \cos e_3) \\ \omega_r + k_3 v_r (e_2 - d \sin e_3) + k_2 \sin e_3 \end{bmatrix} + \begin{bmatrix} v_r \cos e_3 - \omega_r d \sin e_3 \\ v_r \sin e_3 + \omega_r d \cos e_3 \\ \omega_r \end{bmatrix}. \quad (15)$$

Assumption 4 (see [8]). The reference velocities $\vartheta_r = [v_r, \omega_r]^T$ and \dot{v}_r are bounded.

Lemma 5. For the kinematic model (1) of the NWMR satisfying Assumption 4, the auxiliary velocity controller (14) will ensure that the tracking position errors converge asymptotically to zeros.

Proof. Consider the following Lyapunov function candidate

$$V_1 = \frac{1}{2} (e_1 + d - d \cos e_3)^2 + \frac{1}{2} (e_2 - d \sin e_3)^2 + \frac{1 - \cos e_3}{k_3}. \quad (16)$$

Differentiating V_1 with respect to time, we have

$$\begin{aligned} \dot{V}_1 &= (e_1 + d - d \cos e_3) (\dot{e}_1 + d \dot{e}_3 \sin e_3) \\ &\quad + (e_2 - d \sin e_3) (\dot{e}_2 - d \dot{e}_3 \cos e_3) + \frac{1}{k_3} \dot{e}_3 \sin e_3. \end{aligned} \quad (17)$$

Replacing (15) into (17) and after some manipulations, one obtains

$$\dot{V}_1 = -k_1 (e_1 + d - d \cos e_3)^2 - \frac{k_2}{k_3} \sin^2 e_3 \leq 0. \quad (18)$$

Therefore, the tracking position error $e_p = [e_1, e_2, e_3]^T$ is bounded. With Assumption 4, ϑ_c and $\dot{e}_p = [\dot{e}_1, \dot{e}_2, \dot{e}_3]^T$ are bounded. So \dot{V}_1 is bounded and V_1 is uniformly continuous accordingly. By Barbalat's lemma [44], $\dot{V}_1 \rightarrow 0$ as $t \rightarrow \infty$, which implies that $e_1 \rightarrow 0$ and $e_3 \rightarrow 0$ as $t \rightarrow \infty$.

From (15), one obtains

$$\dot{e}_3 = -k_3 v_r (e_2 - d \sin e_3) - k_2 \sin e_3. \quad (19)$$

Using Barbalat's lemma again, $\dot{e}_3 \rightarrow 0$ as $t \rightarrow \infty$, which implies that $e_2 \rightarrow 0$ as $t \rightarrow \infty$.

Hence, $e_p \rightarrow 0$ as $t \rightarrow \infty$; that is, the tracking position errors converge asymptotically to zeros. \square

Now, it remains to design the actuator voltage control input so that the desired velocities ϑ_c can be obtained in finite time.

3.2. Adaptive Integral Terminal Sliding Mode Controller Design. In this study, the auxiliary velocity tracking error is defined as

$$e_{\vartheta} = [e_{\vartheta 1}, e_{\vartheta 2}]^T = \vartheta_c - \vartheta. \quad (20)$$

Consequently, the dynamic equation (2) of the NWMR can be rewritten as

$$\begin{aligned} \overline{M}(q) \dot{e}_{\vartheta} &= -\overline{V}(q, \dot{q}) e_{\vartheta} + \frac{N^2 K_T K_b}{R_a} \overline{B} X \vartheta + \overline{M}(q) \dot{\vartheta}_c \\ &+ \overline{V}(q, \dot{q}) \vartheta_c + \overline{F}(\vartheta) + \overline{\tau}_d - \frac{N K_T}{R_a} \overline{B} u. \end{aligned} \quad (21)$$

A continuous nonsingular integral terminal sliding mode is defined as in the form (4):

$$\begin{aligned} S &= \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \\ &= \begin{bmatrix} e_{\vartheta 1} - e_{\vartheta 1}(0) + \beta_1 \int_0^t |e_{\vartheta 1}|^{\gamma_1} \text{sign}(e_{\vartheta 1}) d\tau \\ e_{\vartheta 2} - e_{\vartheta 2}(0) + \beta_2 \int_0^t |e_{\vartheta 2}|^{\gamma_2} \text{sign}(e_{\vartheta 2}) d\tau \end{bmatrix}, \end{aligned} \quad (22)$$

where $\beta_i > 0$, $0 < \gamma_i < 1$, $i = 1, 2$.

Denote

$$\Lambda = \text{diag}(\beta_1, \beta_2), \quad (23)$$

$$\text{sig}(e_{\vartheta})^{\gamma} = [|e_{\vartheta 1}|^{\gamma_1} \text{sign}(e_{\vartheta 1}), |e_{\vartheta 2}|^{\gamma_2} \text{sign}(e_{\vartheta 2})]^T.$$

S can be rewritten as follows:

$$S = e_{\vartheta} - e_{\vartheta}(0) + \Lambda \int_0^t \text{sig}(e_{\vartheta})^{\gamma} d\tau. \quad (24)$$

Utilizing S and its derivative with respect to time, (21) can be arranged as follows:

$$\begin{aligned} \overline{M}(q) \dot{S} &= -\overline{V}(q, \dot{q}) S + \frac{N^2 K_T K_b}{R_a} \overline{B} X \vartheta + f(x) \\ &- \frac{N K_T}{R_a} \overline{B} u, \end{aligned} \quad (25)$$

where

$$\begin{aligned} f(x) &= [f_1(x), f_2(x)]^T \\ &= \overline{M}(q) \dot{\vartheta}_c + \overline{M}(q) \Lambda \text{sig}(e_{\vartheta})^{\gamma} + \overline{V}(q, \dot{q}) \vartheta_c \\ &- \overline{V}(q, \dot{q}) e_{\vartheta}(0) + \overline{V}(q, \dot{q}) \Lambda \int_0^t \text{sig}(e_{\vartheta})^{\gamma} d\tau \\ &+ \overline{F}(\vartheta) + \overline{\tau}_d \end{aligned} \quad (26)$$

and $x = [\vartheta_c^T, \dot{\vartheta}_c^T, \vartheta^T]^T$.

If $f(x)$ is known, let the actuator voltage control input

$$\begin{aligned} u &= \frac{R_a}{N K_T} \overline{B}^{-1} \left(\frac{N^2 K_T K_b}{R_a} \overline{B} X \vartheta + f(x) + K_1 S \right. \\ &\left. + K_2 \text{sig}(S)^{\rho} \right), \end{aligned} \quad (27)$$

where $K_1 = \text{diag}(k_{11}, k_{12})$, $K_2 = \text{diag}(k_{21}, k_{22})$, $k_{1i} > 0$, $k_{2i} > 0$, $i = 1, 2$, and $0 < \rho < 1$.

Substituting (27) into (25), the closed-loop dynamic equation can be written as

$$\overline{M}(q) \dot{S} = -\overline{V}(q, \dot{q}) S - K_1 S - K_2 \text{sig}(S)^{\rho}. \quad (28)$$

Multiplying S^T by (28) yields

$$S^T \overline{M}(q) \dot{S} = -S^T \overline{V}(q, \dot{q}) S - S^T K_1 S - S^T K_2 \text{sig}(S)^{\rho}. \quad (29)$$

Theorem 6. For the dynamic model (2) of the NWMR with a known function $f(x)$ (26), if the integral terminal sliding mode is chosen as (22) and the actuator voltage control input is designed as (27), then the integral terminal sliding mode S and the auxiliary velocity tracking error e_{ϑ} will converge to zeros in finite time.

To prove Theorem 6, we introduce two lemmas.

Lemma 7 (see [36]). Suppose a_1, a_2, \dots, a_n and $0 < p < 2$ are all positive numbers, then the following inequality holds:

$$(a_1^2 + a_2^2 + \dots + a_n^2)^p \leq (a_1^p + a_2^p + \dots + a_n^p)^2. \quad (30)$$

Lemma 8 (see [36]). An extended Lyapunov description of finite time stability can be given with the form of fast terminal sliding mode as

$$\dot{V}(x) + \alpha V(x) + \beta V^{\gamma}(x) \leq 0, \quad \alpha, \beta > 0, \quad 0 < \gamma < 1, \quad (31)$$

and the settling time can be given by

$$t_r \leq \frac{1}{\alpha(1-\gamma)} \ln \frac{\alpha V^{1-\gamma}(x(0)) + \beta}{\beta}. \quad (32)$$

Proof of Theorem 6. Consider the following Lyapunov function candidate:

$$V_2 = \frac{1}{2} S^T \bar{M}(q) S. \quad (33)$$

Differentiating V_2 with respect to time and using (29) yields

$$\begin{aligned} \dot{V}_2 &= S^T \dot{\bar{M}}(q) \dot{S} + \frac{1}{2} S^T \ddot{\bar{M}}(q) S \\ &= \frac{1}{2} S^T \left(\dot{\bar{M}}(q) - 2\bar{V}(q, \dot{q}) \right) S - S^T K_1 S \\ &\quad - S^T K_2 \text{sig}(S)^\rho. \end{aligned} \quad (34)$$

From Property 3, which makes the first term zero, \dot{V}_2 becomes

$$\dot{V}_2 = -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho. \quad (35)$$

Denote $\lambda_i = \min\{k_{i1}, k_{i2}\}$, $i = 1, 2$; the following inequality holds:

$$\dot{V}_2 \leq -\lambda_1 \sum_{i=1}^2 s_i^2 - \lambda_2 \sum_{i=1}^2 |s_i|^{\rho+1}. \quad (36)$$

Applying Lemma 7 into (36), one obtains

$$\begin{aligned} \dot{V}_2 &\leq -\lambda_1 \frac{2}{m_2} \sum_{i=1}^2 \frac{m_2}{2} s_i^2 \\ &\quad - \lambda_2 \left(\frac{2}{m_2} \right)^{(\rho+1)/2} \left[\sum_{i=1}^2 \frac{m_2 s_i^2}{2} \right]^{(\rho+1)/2}. \end{aligned} \quad (37)$$

Utilizing Property 2, we have

$$\dot{V}_2 \leq -\lambda_1 \frac{2}{m_2} V_2 - \lambda_2 \left(\frac{2}{m_2} \right)^{(\rho+1)/2} V_2^{(\rho+1)/2}, \quad (38)$$

or

$$\dot{V}_2 + \lambda_1 \frac{2}{m_2} V_2 + \lambda_2 \left(\frac{2}{m_2} \right)^{(\rho+1)/2} V_2^{(\rho+1)/2} \leq 0. \quad (39)$$

From Lemma 8, it follows that S will converge to zero in finite time

$$\begin{aligned} t_r &\leq \frac{1}{\lambda_1 (2/m_2) (1 - (\rho+1)/2)} \\ &\quad \cdot \ln \frac{\lambda_1 (2/m_2) V_2^{1-(\rho+1)/2}(0) + \lambda_2 (2/m_2)^{(\rho+1)/2}}{\lambda_2 (2/m_2)^{(\rho+1)/2}}. \end{aligned} \quad (40)$$

Moreover, on the sliding mode surface, according to (7),

$$t_{si} = \frac{1}{\beta_i (1 - \gamma_i)} |e_{\theta_i}(0)|^{1-\gamma_i}, \quad i = 1, 2. \quad (41)$$

Therefore, the auxiliary velocity tracking error e_θ will converge to zero in finite time $t = t_r + \max\{t_{s1}, t_{s2}\}$. \square

Due to the fact that $f(x)$ contains all the mobile robot parameters (such as mass, moment of inertia, and friction coefficients) and external disturbances, in the following, we assume that $f_i(x)$, $i = 1, 2$, can be approximated by the following FLS:

$$\hat{f}_i(x | \hat{\theta}_{fi}) = \hat{\theta}_{fi}^T \xi_{fi}(x), \quad (42)$$

where $\xi_{fi}(x)$ is the fuzzy basis function vector and $\hat{\theta}_{fi}$ is the parameter vector of each fuzzy system designed later.

Define the optimal approximation parameters $\hat{\theta}_{fi}^*$ as follows:

$$\hat{\theta}_{fi}^* = \underset{\hat{\theta}_{fi} \in \Omega_{fi}}{\text{argmin}} \left[\sup_{x \in U} |f_i(x) - \hat{f}_i(x | \hat{\theta}_{fi})| \right], \quad (43)$$

where Ω_{fi} is the compact set of allowable controller parameters. Moreover, the parameter error and the minimum approximation error are defined as $\tilde{\theta}_{fi} = \hat{\theta}_{fi} - \hat{\theta}_{fi}^*$ and $\omega_{fi}(x) = f_i(x) - \hat{f}_i(x | \hat{\theta}_{fi}^*)$, respectively.

Assumption 9. For $i = 1, 2$, ω_{fi} is bounded. That is, there exists an unknown constant $\bar{\omega}_{fi}$ such that $|\omega_{fi}(x)| < \bar{\omega}_{fi}$.

Denote

$$\hat{f}(x | \hat{\theta}_f) = [\hat{f}_1(x | \hat{\theta}_{f1}), \hat{f}_2(x | \hat{\theta}_{f2})]^T, \quad (44)$$

$$\hat{f}(x | \theta_f^*) = [\hat{f}_1(x | \theta_{f1}^*), \hat{f}_2(x | \theta_{f2}^*)]^T.$$

By using the fuzzy approximation $\hat{f}(x | \hat{\theta}_f)$ instead of $f(x)$, the following control law from (27) is obtained:

$$\begin{aligned} u &= \frac{R_a}{NK_T} \bar{B}^{-1} \left(\frac{N^2 K_T K_b}{R_a} \bar{B} X \vartheta + \hat{f}(x | \hat{\theta}_f) + K_1 s \right. \\ &\quad \left. + K_2 \text{sig}(s)^\rho + u_r \right), \end{aligned} \quad (45)$$

where $u_r = [u_{r1}, u_{r2}]^T$ is a robust controller, which is designed as

$$u_r = \begin{cases} \frac{(\hat{\varepsilon}_f + \sigma) s}{\|s\|}, & s \neq 0, \\ 0, & s = 0, \end{cases} \quad (46)$$

$\hat{\varepsilon}_f$ is the estimate of $\sum_{i=1}^2 \bar{\omega}_{fi}$, $\tilde{\varepsilon}_f = \hat{\varepsilon}_f - \sum_{i=1}^2 \bar{\omega}_{fi}$, and σ is a positive constant.

Remark 10. It is noticed that the robust controller u_r (46) is similar to that in [37]. However, in [37], σ is required no less than the estimation error of the unknown function. Whereas, in practice, it is difficult to determine such an estimation error. In this paper, σ is relaxed to be an arbitrary positive number.

Substituting (45) into (25), the closed-loop dynamic equation can be rewritten as

$$\begin{aligned} \bar{M}(q) \dot{S} &= -\bar{V}(q, \dot{q}) S - \hat{f}(x | \hat{\theta}) - K_1 S - K_2 \text{sig}(S)^\rho \\ &\quad - u_r + f(x). \end{aligned} \quad (47)$$

Multiplying S^T to (47) and after some manipulations, we can get

$$\begin{aligned} S^T \overline{M}(q) \dot{S} = & -S^T \overline{V}(q, \dot{q}) S - S^T K_1 S - S^T K_2 \text{sig}(S)^\rho \\ & - S^T u_r + \sum_{i=1}^2 s_i \omega_{fi}(x) + \sum_{i=1}^2 s_i \tilde{\theta}_{fi}^T \xi_{fi}(x). \end{aligned} \quad (48)$$

We use the following adaptation laws to adjust the unknown parameters $\hat{\theta}_{fi}$ and $\hat{\varepsilon}_f$:

$$\begin{aligned} \dot{\hat{\theta}}_{fi} = & -\mu_{fi} s_i \xi_{fi}(x), \\ \dot{\hat{\varepsilon}}_f = & \eta \|s\|, \end{aligned} \quad (49)$$

where $\mu_{fi} > 0$, $i = 1, 2$, and $\eta > 0$.

The properties of the proposed adaptive fuzzy ITSMC law is summarized by the following theorem.

Theorem 11. *For the dynamic model (2) of the NWMR with an unknown function $f(x)$ (26), if the integral terminal sliding mode is chosen as (22), the actuator voltage control input with dynamic robust controller u_r (46) is designed as (45), and the adaptation laws are (49); then*

- (1) all the signals in the closed system are bounded;
- (2) the sliding variable S will converge to the neighborhood of the integral terminal sliding mode $S = 0$ as $\|S\| \leq \delta = \min\{\delta_1, \delta_2\}$ in finite time, where

$$\begin{aligned} \delta_1 = & \frac{\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)\| + \|\tilde{\varepsilon}_f\|}{\lambda_1}, \\ \delta_2 = & \left[\frac{\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)\| + \|\tilde{\varepsilon}_f\|}{\lambda_2} \right]^{1/\rho}. \end{aligned} \quad (50)$$

Moreover, the auxiliary velocity tracking error e_{vi} will converge to the region $|e_{vi}| \leq 2\delta + |e_{vi}(0)|$, $i = 1, 2$ in finite time.

Proof. Consider the following Lyapunov function candidate:

$$V_3 = V_{31} + V_{32}, \quad (51)$$

where

$$\begin{aligned} V_{31} = & \frac{1}{2} S^T \overline{M}(q) S, \\ V_{32} = & \frac{1}{2} \left(\sum_{i=1}^2 \frac{1}{\mu_{fi}} \tilde{\theta}_{fi}^T \tilde{\theta}_{fi} + \frac{1}{\eta} \tilde{\varepsilon}_f^2 \right). \end{aligned} \quad (52)$$

(1) Differentiating V_{31} with respect to time, using (48) and Property 3, we have

$$\begin{aligned} \dot{V}_{31} = & -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho - S^T u_r + \sum_{i=1}^2 s_i \omega_{fi}(x) \\ & + \sum_{i=1}^2 s_i \tilde{\theta}_{fi}^T \xi_{fi}(x). \end{aligned} \quad (53)$$

Note that

$$\sum_{i=1}^2 s_i \omega_{fi}(x) \leq \|S\| \sum_{i=1}^2 \bar{\omega}_i. \quad (54)$$

Substituting (46) and (54) into (53), the following inequality holds:

$$\begin{aligned} \dot{V}_{31} \leq & -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho - \|S\| (\tilde{\varepsilon}_f + \sigma) \\ & + \|S\| \sum_{i=1}^2 \bar{\omega}_i + \sum_{i=1}^2 s_i \tilde{\theta}_{fi}^T \xi_{fi}(x) \\ = & -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho - \|S\| \sigma - \|S\| \tilde{\varepsilon}_f \\ & + \sum_{i=1}^2 s_i \tilde{\theta}_{fi}^T \xi_{fi}(x). \end{aligned} \quad (55)$$

There results

$$\begin{aligned} \dot{V}_{31} \leq & -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho - \|S\| \tilde{\varepsilon}_f \\ & + \sum_{i=1}^2 s_i \tilde{\theta}_{fi}^T \xi_{fi}(x). \end{aligned} \quad (56)$$

Differentiating V_{32} with respect to time yields

$$\dot{V}_{32} = \sum_{i=1}^2 \frac{1}{\mu_{fi}} \tilde{\theta}_{fi}^T \dot{\tilde{\theta}}_{fi} + \frac{1}{\eta} \tilde{\varepsilon}_f \dot{\tilde{\varepsilon}}_f. \quad (57)$$

Combining (56) with (57), we can get

$$\begin{aligned} \dot{V}_{31} + \dot{V}_{32} \leq & -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho \\ & + \sum_{i=1}^2 \tilde{\theta}_{fi}^T \left(s_i \xi_{fi}(x) + \frac{1}{\mu_{fi}} \dot{\tilde{\theta}}_{fi} \right) \\ & + \tilde{\varepsilon}_f \left(\frac{1}{\eta} \dot{\tilde{\varepsilon}}_f - \|S\| \right). \end{aligned} \quad (58)$$

That is,

$$\begin{aligned} \dot{V}_{31} + \dot{V}_{32} \leq & -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho \\ & + \sum_{i=1}^2 \tilde{\theta}_{fi}^T \left(s_i \xi_{fi}(x) - \frac{1}{\mu_{fi}} \dot{\tilde{\theta}}_{fi} \right) \\ & + \tilde{\varepsilon}_f \left(\frac{1}{\eta} \dot{\tilde{\varepsilon}}_f - \|S\| \right). \end{aligned} \quad (59)$$

Applying the adaptation laws (49) into (59), one has

$$\dot{V}_{31} + \dot{V}_{32} \leq -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho. \quad (60)$$

Clearly, $\dot{V}_3 = \dot{V}_{31} + \dot{V}_{32} \leq 0$, it is concluded that all the signals s_i , e_{vi} , $\hat{\theta}_{fi}$, and $\hat{\varepsilon}_f$ are bounded.

(2) According to (56), one gets

$$\begin{aligned} \dot{V}_{31} &\leq -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho - \|S\| \tilde{\varepsilon}_f \\ &\quad + S^T [\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)] \\ &= -S^T K_1 S - S^T K_2 \text{sig}(S)^\rho \\ &\quad + S^T \left[\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) - \frac{\tilde{\varepsilon}_f}{\|S\|} S \right], \end{aligned} \quad (61)$$

which can be further changed into the following two forms:

$$\begin{aligned} \dot{V}_{31} &< -S^T \left\{ K_1 - \text{diag} \left[\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) \right. \right. \\ &\quad \left. \left. - \frac{\tilde{\varepsilon}_f}{\|S\|} S \right] \text{diag}^{-1}(S) \right\} S - S^T K_2 \text{sig}(S)^\rho, \end{aligned} \quad (62)$$

$$\begin{aligned} \dot{V}_{31} &< -S^T K_1 S - S^T \left\{ K_2 - \text{diag} \left[\hat{f}(x | \theta_f^*) \right. \right. \\ &\quad \left. \left. - \hat{f}(x | \hat{\theta}_f) - \frac{\tilde{\varepsilon}_f}{\|S\|} S \right] \text{diag}^{-1}[\text{sig}(S)^\rho] \right\} \\ &\quad \cdot \text{sig}(S)^\rho. \end{aligned} \quad (63)$$

Denote

$$\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) - \frac{\tilde{\varepsilon}_f}{\|S\|} S = [\tilde{f}_1, \tilde{f}_2]^T. \quad (64)$$

For (62), if $\lambda_1 > |\tilde{f}_i|/|s_i|$, $i = 1, 2$, which means the matrix $K_1 - \text{diag}[\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) - (\tilde{\varepsilon}_f/\|S\|)S] \text{diag}^{-1}(S)$ is positive definite, the similar structure as (35) is kept. Hence, finite time stability is guaranteed. Otherwise, $|s_i| \leq |\tilde{f}_i|/|\lambda_1|$, $i = 1, 2$. We can conclude that

$$\begin{aligned} \|S\|^2 &= s_1^2 + s_2^2 \leq \frac{\tilde{f}_1^2 + \tilde{f}_2^2}{\lambda_1^2} \\ &= \frac{\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) - (\tilde{\varepsilon}_f/\|S\|)S\|^2}{\lambda_1^2} \\ &\leq \frac{(\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)\| + \|\tilde{\varepsilon}_f\|)^2}{\lambda_1^2}, \end{aligned} \quad (65)$$

that is,

$$\|S\| \leq \frac{\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)\| + \|\tilde{\varepsilon}_f\|}{\lambda_1} = \delta_1. \quad (66)$$

Therefore, the region $\|S\| \leq \delta_1$ can be reached in finite time.

For (63), if $\lambda_2 > |\tilde{f}_i|/|s_i|^\rho$, $i = 1, 2$, which means the matrix $K_2 - \text{diag}[\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) - (\tilde{\varepsilon}_f/\|S\|)S] \text{diag}^{-1}(S^\rho)$ is positive definite, the similar structure as (35) is kept.

Hence, finite time stability is guaranteed. Otherwise, $|s_i|^\rho \leq |\tilde{f}_i|/|\lambda_2|$, $i = 1, 2$. We can conclude from Lemma 7 that

$$\begin{aligned} \|S\|^{4\rho} &= (s_1^2 + s_2^2)^{2\rho} \leq (s_1^{2\rho} + s_2^{2\rho})^2 \leq \left(\frac{\tilde{f}_1^2 + \tilde{f}_2^2}{\lambda_2^2} \right)^2 \\ &= \frac{\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f) - (\tilde{\varepsilon}_f/\|S\|)S\|^4}{\lambda_2^4} \\ &\leq \frac{(\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)\| + \|\tilde{\varepsilon}_f\|)^4}{\lambda_2^4}; \end{aligned} \quad (67)$$

that is,

$$\|S\| \leq \left[\frac{\|\hat{f}(x | \theta_f^*) - \hat{f}(x | \hat{\theta}_f)\| + \|\tilde{\varepsilon}_f\|}{\lambda_2} \right]^{1/\rho} = \delta_2. \quad (68)$$

Therefore, the region $\|S\| \leq \delta_2$ can be reached in finite time.

By virtue of (66) and (68), the region $\|S\| \leq \delta = \min\{\delta_1, \delta_2\}$ can be reached in finite time.

When $\|S\| \leq \delta$, for $i = 1, 2$, $|s_i| \leq \delta$. The integral terminal sliding mode (22) can be changed into the following form:

$$\begin{aligned} s_i &= e_{vi} - e_{vi}(0) + \beta_i \int_0^t |e_{vi}|^{\gamma_i} \text{sign}(e_{vi}) d\tau = \phi_i, \\ &|\phi_i| \leq \delta, \end{aligned} \quad (69)$$

or the equivalent form

$$\begin{aligned} e_{vi} - e_{vi}(0) &+ \left[\beta_i - \frac{\phi_i}{\int_0^t |e_{vi}|^{\gamma_i} \text{sign}(e_{vi}) d\tau} \right] \int_0^t |e_{vi}|^{\gamma_i} \text{sign}(e_{vi}) d\tau \\ &= 0. \end{aligned} \quad (70)$$

If $\beta_i > |\phi_i|/\int_0^t |e_{vi}|^{\gamma_i} \text{sign}(e_{vi}) d\tau$, (70) is kept in the form of the integral terminal sliding mode. Hence, finite time convergence is guaranteed. Otherwise, $|\int_0^t |e_{vi}|^{\gamma_i} \text{sign}(e_{vi}) d\tau| \leq |\phi_i|/\beta_i \leq \delta/\beta_i$; the region

$$\begin{aligned} |e_{vi}| &\leq \beta_i \left| \int_0^t |e_{vi}|^{\gamma_i} \text{sign}(e_{vi}) d\tau \right| + \delta + |e_{vi}(0)| \\ &\leq 2\delta + |e_{vi}(0)|, \end{aligned} \quad (71)$$

can be reached in finite time. \square

Remark 12. Both the control law (27) and the adaptive fuzzy control law (45) contain a nonlinear term $K_1 S + K_2 \text{sig}(S)^\rho$ with the form of fast terminal sliding mode, which assures the boundedness of the signals in the closed system and the finite time convergence of the auxiliary velocity tracking error.

Remark 13. According to (66) and (68), the parameters λ_1 and λ_2 can be chosen large enough to make the boundary δ small. However, increasing the parameters λ_1 and λ_2 will increase the level of control input and will cause implementation problem.

Remark 14. The relationship between the auxiliary velocity tracking error e_{v_i} and the width of the boundary layer δ surrounding the integral terminal sliding mode surface $S = 0$ is given by (69) and (71).

4. Simulation Results

In this section, a simulation will be provided to show the effectiveness of the proposed control strategy.

Referring to [18], the parameters of the NWMR and its actuators are chosen as $m = 10$ kg, $I = 5$ kg·m², $R = 0.12$ m, $r = 0.067$ m, $d = 0.3$ m, $N = 50$, $K_b = 0.026$, $K_T = 0.026$ Nm/A, and $R_a = 3.5$ Ω . The surface friction and the external disturbance are generated by $\bar{F}(\vartheta) + \bar{\tau}_d = [5 \sin(2t), 5 \sin(2t)]^T$. In this simulation, the initial posture and velocity of the practical NWMR are taken as $q(0) = [0.1, 0.2, 0]^T$ and $\vartheta = [\nu, \omega]^T = [0, 0]^T$, respectively.

The reference linear velocity and angular velocity are defined as $v_r(t) = 1$ and $\omega_r(t) = 1$. The trajectory of the reference NWMR is defined as

$$\begin{aligned}\dot{x}_r &= v_r \cos \theta_r - \omega_r d \sin \theta_r, \\ \dot{y}_r &= v_r \sin \theta_r + \omega_r d \cos \theta_r, \\ \dot{\theta}_r &= \omega_r.\end{aligned}\quad (72)$$

The initial posture of the reference NWMR is taken as $q_r(0) = [0, 0, 0]^T$.

The objective of the trajectory tracking control is to design a strategy such that $q(t)$ converges asymptotically to $q_r(t)$, while all signals in the derived closed-loop system are able to remain bounded. In the proposed control strategy, an auxiliary velocity controller ϑ_c is designed for the kinematic model to meet the control objective. Then, the actuator voltage control input u is designed for the dynamic model such that ϑ converges to ϑ_c which is designed at the first step in finite time.

In the actuator voltage control input u , the nonlinear function

$$\begin{aligned}f(x) &= [f_1(x), f_2(x)]^T \\ &= \bar{M}(q) \dot{\vartheta}_c + \bar{M}(q) \Lambda \text{sig}(e_\vartheta)^y + \bar{V}(q, \dot{q}) \vartheta_c \\ &\quad - \bar{V}(q, \dot{q}) e_\vartheta(0) + \bar{V}(q, \dot{q}) \Lambda \int_0^t \text{sig}(e_\vartheta)^y d\tau \\ &\quad + \bar{F}(\vartheta) + \bar{\tau}_d\end{aligned}\quad (73)$$

is contained, where $x = [\vartheta_c^T, \dot{\vartheta}_c^T, \vartheta^T]^T$.

We suppose that there is no prior information of the robot parameters such as mass, moment of inertial, friction coefficients, and the external disturbance; that is, the nonlinear function $f_i(x)$, $i = 1, 2$, is assumed to be completely unknown. Two fuzzy systems in the form of (9) are used to approximate $f_1(x)$ and $f_2(x)$. The fuzzy systems have $x_1 = v_c$, $x_2 = \omega_c$, $x_3 = \dot{v}_c$, $x_4 = \dot{\omega}_c$, $x_5 = v$, and $x_6 = \omega$ as inputs;

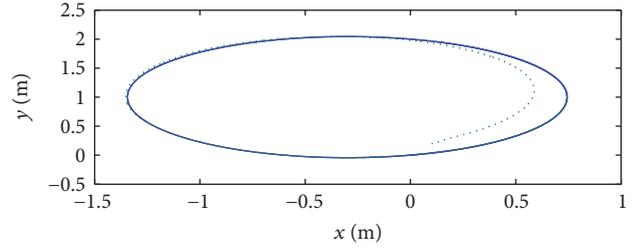


FIGURE 2: Reference trajectory (-) and actual trajectory (···).

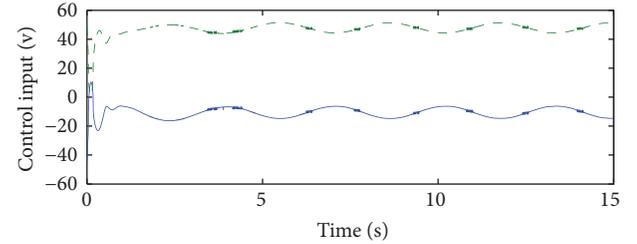


FIGURE 3: Actuator voltages u_1 (-) and u_2 (- -).

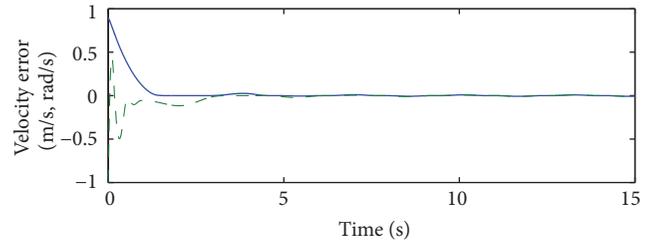


FIGURE 4: Auxiliary velocity tracking errors e_{g1} (-) and e_{g2} (- -).

the fuzzy membership functions for each variable x_i , $i = 1, 2, \dots, 6$, are chosen as

$$\begin{aligned}\mu_{F_1^1}(x_i) &= \exp\left[-\frac{1}{2}\left(\frac{x_i + 1.25}{0.6}\right)^2\right], \\ \mu_{F_1^2}(x_i) &= \exp\left[-\frac{1}{2}\left(\frac{x_i}{0.6}\right)^2\right], \\ \mu_{F_1^3}(x_i) &= \exp\left[-\frac{1}{2}\left(\frac{x_i - 1.25}{0.6}\right)^2\right].\end{aligned}\quad (74)$$

The initial values of the estimated parameters $\hat{\theta}_{f_i}(0)$, $i = 1, 2$, and $\hat{\varepsilon}_f(0)$ are all set to 0.01.

Referring to [1, 31, 36], the parameters of the control law are chosen as $k_1 = 1$, $k_2 = 20$, $k_3 = 10$, $\beta_1 = \beta_2 = 1$, $\gamma_1 = \gamma_2 = 0.5$, $k_{11} = k_{12} = k_{21} = k_{22} = 20$, $\rho = 0.3$, $\mu_{f_1} = \mu_{f_2} = 0.5$, $\eta = 0.001$, and $\sigma = 0.1$.

Using our control strategy to control the NWMR, the simulation results are shown in Figures 2–5. Figure 2 is the trajectory tracking process in X-Y plane of the NWMR, Figure 3 is the actuator voltage control input, Figure 4 is the auxiliary velocity tracking errors, and Figure 5 is the tracking position errors, respectively. From Figure 4, it can be observed that the auxiliary velocity tracking error e_{g1}

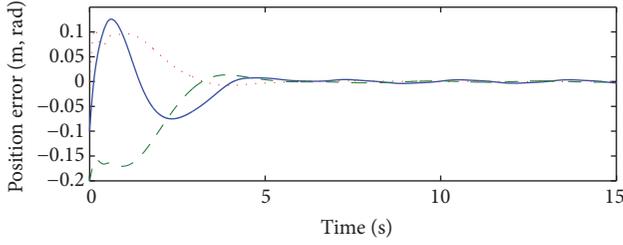


FIGURE 5: Tracking position errors e_1 (-), e_2 (- -), and e_3 (\cdots).

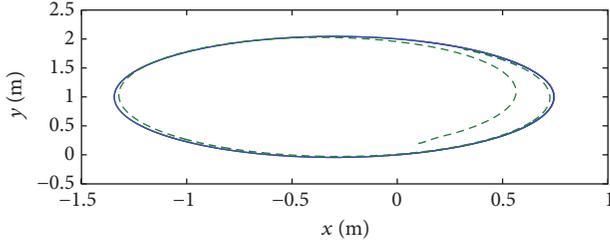


FIGURE 6: Reference trajectory (-) and actual trajectory (- -).

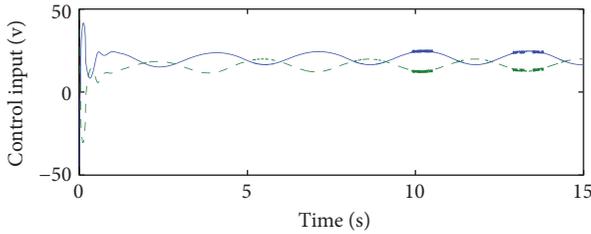


FIGURE 7: Actuator voltages u_1 (-) and u_2 (- -).

converges to $|e_{\theta_1}| \leq 5 \times 10^{-3}$ in finite time $t = 1.389$ s and the auxiliary velocity tracking error e_{θ_2} converges to $|e_{\theta_2}| \leq 7 \times 10^{-3}$ in finite time $t = 3.052$ s, respectively.

In order to compare the proposed integral terminal sliding mode with the integral sliding mode adopted in [25], we use integral sliding mode instead of the integral terminal sliding mode in our control strategy while other design parameters are the same as the corresponding design parameters used in the above simulation. The corresponding simulation results are shown in Figures 6–9. From Figure 8, it can be observed that the auxiliary velocity tracking error e_{θ_1} converges to $|e_{\theta_1}| \leq 3 \times 10^{-2}$ in finite time $t = 13.47$ s and the auxiliary velocity tracking error e_{θ_2} converges to $|e_{\theta_2}| \leq 4 \times 10^{-2}$ in finite time $t = 14.05$ s, respectively.

It is observed that the actual velocity can track the auxiliary velocity in less time using the control strategy proposed in this paper from Figures 4 and 8. As a result, the practical NWMR can track the reference NWMR asymptotically with faster response from Figures 2 and 6. Meanwhile, this favorable performance was obtained with no prior information of the robot parameters such as mass, moment of inertial, friction coefficients, and the external disturbance.

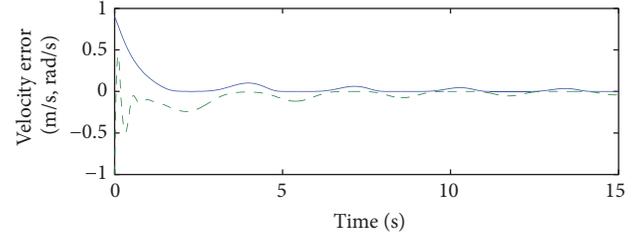


FIGURE 8: Auxiliary velocity tracking errors e_{θ_1} (-) and e_{θ_2} (- -).

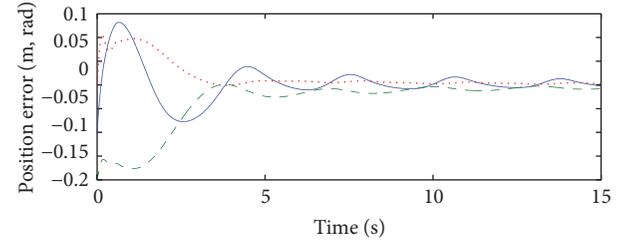


FIGURE 9: Tracking position errors e_1 (-), e_2 (- -), and e_3 (\cdots).

5. Conclusions

In this paper, a control strategy has been proposed for the trajectory tracking problem of the NWMR with parameter uncertainties and external disturbances. In this study, we take the wheel actuator dynamics into system dynamics and choose the actuator voltage as the control input. The FLS is adopted to estimate the unknown function coming from parameter uncertainties and external disturbances. An adaptive fuzzy integral terminal sliding mode controller is integrated with an auxiliary velocity controller. It has been shown that all the signals in the closed system are bounded and the auxiliary velocity tracking errors converge to a small neighborhood of the origin in finite time. Hence, the tracking position errors converge asymptotically to zeros with faster response than other existing controllers. Simulation results have been provided to show the feasibility of the proposed control strategy. However, wheel skidding and slipping are unavoidable due to tire deformation and other reasons in real environments. In the future, we will extend our results to the trajectory tracking control of the WMR with wheel skidding and slipping.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Tianjin, China, under Grants 15JCYBJC47800.

References

- [1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in

- Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 384–389, May 1990.
- [2] T. H. Bui, T. T. Nguyen, T. L. Chung, and S. B. Kim, “A simple nonlinear control of a two-wheeled welding mobile robot,” *International Journal of Control, Automation and Systems*, vol. 1, no. 1, pp. 35–42, 2003.
 - [3] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics,” in *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 3805–3810, December 1995.
 - [4] T. Appala and A. Ghosal, “Tip over stability analysis of a three-wheeled mobile robot capable of traversing uneven terrains without slip,” *Applied Mechanics and Materials*, vol. 110–116, pp. 2940–2947, 2012.
 - [5] T. Appala and A. Ghosal, “A mobile robot with a two-degree-of-freedom suspension for traversing uneven terrain with minimal slip: modeling, simulation and experiments,” *Mechanism and Machine Theory*, vol. 93, pp. 83–97, 2015.
 - [6] D. Wang and C. B. Low, “Modeling and analysis of skidding and slipping in wheeled mobile robots: control design perspective,” *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 676–687, 2008.
 - [7] T. Fukao, H. Nakagawa, and N. Adachi, “Adaptive tracking control of a nonholonomic mobile robot,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, 2000.
 - [8] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, “Adaptive output-feedback control for trajectory tracking of electrically driven non-holonomic mobile robots,” *IET Control Theory and Applications*, vol. 5, no. 6, pp. 830–838, 2011.
 - [9] J. Huang, C. Wen, W. Wang, and Z.-P. Jiang, “Adaptive output feedback tracking control of a nonholonomic mobile robot,” *Automatica*, vol. 50, no. 3, pp. 821–831, 2014.
 - [10] S. J. Yoo, “Approximation-based adaptive control for a class of mobile robots with unknown skidding and slipping,” *International Journal of Control, Automation and Systems*, vol. 10, no. 4, pp. 703–710, 2012.
 - [11] Y. Koubaa, M. Boukattaya, and T. Dammak, “An adaptive control for uncertain mobile robot considering skidding and slipping effects,” in *Proceedings of the 5th International Conference on Systems and Control (ICSC '16)*, pp. 13–19, May 2016.
 - [12] J. Wu, G. Xu, and Z. Yin, “Robust adaptive control for a non-holonomic mobile robot with unknown parameters,” *Journal of Control Theory and Applications*, vol. 7, no. 2, pp. 212–218, 2009.
 - [13] L. Xin, Q. Wang, J. She, and Y. Li, “Robust adaptive tracking control of wheeled mobile robot,” *Robotics and Autonomous Systems*, vol. 78, pp. 36–48, 2016.
 - [14] M. Chen, “Disturbance attenuation tracking control for wheeled mobile robots with skidding and slipping,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3359–3368, 2017.
 - [15] A. Zou, Z. Hou, M. Tan, and Z. Zhao, “Tracking control of a nonholonomic mobile robot using a fuzzy-based approach,” in *Fuzzy Systems and Knowledge Discovery*, vol. 4223 of *Lecture Notes in Computer Science*, pp. 826–835, Springer, Berlin, Germany, 2006.
 - [16] T. Das and I. N. Kar, “Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots,” *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 501–510, 2006.
 - [17] Z.-G. Hou, A.-M. Zou, L. Cheng, and M. Tan, “Adaptive control of an electrically driven nonholonomic mobile robot via backstepping and fuzzy approach,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 803–815, 2009.
 - [18] T. Q. Vinh, N. H. Giap, T.-W. Kim, M.-G. Jeong, J.-H. Shin, and W.-H. Kim, “Adaptive robust fuzzy control and implementation for path tracking of a mobile robot,” in *Proceedings of the IEEE International Symposium on Industrial Electronics (IEEE ISIE '09)*, pp. 1943–1949, July 2009.
 - [19] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot using neural networks,” *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 589–600, 1998.
 - [20] W. Zeng, Q. Wang, F. Liu, and Y. Wang, “Learning from adaptive neural network output feedback control of a unicycle-type mobile robot,” *ISA Transactions*, vol. 61, pp. 337–347, 2016.
 - [21] J.-M. Yang and J.-H. Kim, “Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 578–587, 1999.
 - [22] D. Chwa, “Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 637–644, 2004.
 - [23] J. K. Lee, Y. H. Choi, and J. B. Park, “Sliding mode tracking control of mobile robots with approach angle in cartesian coordinates,” *International Journal of Control, Automation and Systems*, vol. 13, no. 3, pp. 718–724, 2015.
 - [24] Y. Hu and S. X. Yang, “A fuzzy neural dynamics based tracking controller for a nonholonomic mobile robot,” in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '03)*, pp. 205–210, IEEE, July 2003.
 - [25] M. Yue, S. Wang, and Y. Zhang, “Adaptive fuzzy logic-based sliding mode control for a nonholonomic mobile robot in the presence of dynamic uncertainties,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 229, no. 11, pp. 1979–1988, 2015.
 - [26] S. Tong, B. Huo, and Y. Li, “Observer-based adaptive decentralized fuzzy fault-tolerant control of nonlinear large-scale systems with actuator failures,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 1, pp. 1–15, 2014.
 - [27] S. Tong, T. Wang, and Y. Li, “Fuzzy adaptive actuator failure compensation control of uncertain stochastic nonlinear systems with unmodeled dynamics,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 563–574, 2014.
 - [28] Y. Li, S. Tong, and T. Li, “Observer-based adaptive fuzzy tracking control of MIMO stochastic nonlinear systems with unknown control directions and unknown dead zones,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 1228–1241, 2015.
 - [29] Y. Li, S. Tong, and T. Li, “Hybrid fuzzy adaptive output feedback control design for uncertain MIMO nonlinear systems with time-varying delays and input saturation,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 4, pp. 841–853, 2016.
 - [30] S. Tong, S. Sui, and Y. Li, “Fuzzy adaptive output feedback control of MIMO nonlinear systems with partial tracking errors constrained,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 729–742, 2015.
 - [31] W. Shi, “Adaptive fuzzy control for multi-input multi-output nonlinear systems with unknown dead-zone inputs,” *Applied Soft Computing Journal*, vol. 30, pp. 36–47, 2015.
 - [32] S. T. Venkataraman and S. Gulati, “Control of nonlinear systems using terminal sliding modes,” in *Proceedings of the American Control Conference*, pp. 891–893, June 1992.
 - [33] M. Zhihong, A. P. Paplinski, and H. R. Wu, “A robust mimo terminal sliding mode control scheme for rigid robotic manipulators,” *IEEE Transactions on Automatic Control*, vol. 39, no. 12, pp. 2464–2469, 1994.

- [34] K.-B. Park and J.-J. Lee, "Comments on 'a robust MIMO terminal sliding mode control scheme for rigid robotic manipulators,'" *IEEE Transactions on Automatic Control*, vol. 41, no. 5, pp. 761–762, 1996.
- [35] Y. Feng, X. Yu, and Z. Man, "Non-singular terminal sliding mode control of rigid manipulators," *Automatica*, vol. 38, no. 12, pp. 2159–2167, 2002.
- [36] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," *Automatica*, vol. 41, no. 11, pp. 1957–1964, 2005.
- [37] L. Wang, T. Chai, and L. Zhai, "Neural-network-based terminal sliding-mode control of robotic manipulators including actuator dynamics," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3296–3304, 2009.
- [38] T. Madani, B. Daachi, and K. Djouani, "Non-singular terminal sliding mode controller: application to an actuated exoskeleton," *Mechatronics*, vol. 33, pp. 136–145, 2016.
- [39] V. Nekoukar and A. Erfanian, "Adaptive fuzzy terminal sliding mode control for a class of MIMO uncertain nonlinear systems," *Fuzzy Sets and Systems*, vol. 179, no. 1, pp. 34–49, 2011.
- [40] S. Khoo, L. Xie, and Z. Man, "Integral terminal sliding mode cooperative control of multi-robot networks," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '09)*, pp. 969–973, IEEE, Singapore, July 2009.
- [41] C.-S. Chiu, "Derivative and integral terminal sliding mode control for a class of MIMO nonlinear systems," *Automatica*, vol. 48, no. 2, pp. 316–326, 2012.
- [42] A. Vladimir, R. Zorich, and R. Cooke, *Mathematical Analysis I*, Springer, 2008.
- [43] D. Huang, J. Zhai, W. Ai, and S. Fei, "Disturbance observer-based robust control for trajectory tracking of wheeled mobile robots," *Neurocomputing*, vol. 198, pp. 74–79, 2016.
- [44] J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

Research Article

A FastSLAM Algorithm Based on Nonlinear Adaptive Square Root Unscented Kalman Filter

Yu-feng Zhang,¹ Qi-xun Zhou,² Ju-zhong Zhang,³ Yi Jiang,³ and Kai Wang⁴

¹School of Electrical and Control Engineering, Xi'an University of Science and Technology, Shaanxi, Xi'an 710054, China

²Key Laboratory of Embedded System and Service Computing, Tongji University, Ministry of Education, Shanghai 201804, China

³713th Institute of China Shipbuilding Industry Corporation, Zhengzhou, China

⁴Naval Representative Office in Zhengzhou Region, Zhengzhou, China

Correspondence should be addressed to Yu-feng Zhang; xkdzhangyufeng@163.com

Received 26 August 2016; Revised 15 December 2016; Accepted 6 February 2017; Published 15 March 2017

Academic Editor: Francisco Valero

Copyright © 2017 Yu-feng 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.

For fast simultaneous localization and mapping (FastSLAM) problem, to solve the problems of particle degradation, the error introduced by linearization and inconsistency of traditional algorithm, an improved algorithm is described in the paper. In order to improve the accuracy and reliability of algorithm which is applied in the system with lower measurement frequency, a new decomposition strategy is adopted for a posteriori estimation. In proposed decomposition strategy, the problem of solving a 3-dimensional state vector and N 2-dimensional state vectors in traditional FastSLAM algorithm is transformed to the problem of solving N 5-dimensional state vectors. Furthermore, a nonlinear adaptive square root unscented Kalman filter (NASRUKF) is used to replace the particle filter and Kalman filter employed by traditional algorithm to reduce the model linearization error and avoid solving Jacobian matrices. Finally, the proposed algorithm is experimentally verified by vehicle in indoor environment. The results prove that the positioning accuracy of proposed FastSLAM algorithm is less than 1 cm and the azimuth angle error is 0.5 degrees.

1. Introduction

In simultaneous localization and mapping (SLAM), vehicle uses the carried sensors to sense surroundings and uses the sensed information to create environment map on one hand. On the other hand, vehicle uses created map to locate and guide.

There are several methods to deal with SLAM problem, in which EKF-SLAM and FastSLAM are the two most popular methods. EKF-SLAM has some obvious limitations: inconsistency due to errors accumulation introduced by linearization, complex computation to deal with high-dimensional joint covariance, lack of robustness to incorrect data association, and so on [1–3]. FastSLAM employs the Rao-Blackwellized particle filter (RBPF) to estimate position and extended Kalman filter (EKF) to estimate map features. Compared with EKF-SLAM, FastSLAM has a lower complexity and

is more robust regarding the data association problem. Nevertheless, the FastSLAM based on RBPF also suffers from some drawbacks such as particle degeneration, Jacobian Matrix solving and linear processing of nonlinear function [4–6]. To deal with these problems, the SLAM based on square root unscented Kalman filter (SRUKF) is proposed. Instead of approximating the state and measurement transition functions by Taylor series expansion, the unscented transformation employed by SLAM is used to update nonlinear state and measurement functions and the accuracy of the state estimation has been significantly improved [7–9].

The filtering approaches used in all aforementioned SLAM methods can achieve good performance under certain assumptions. However, the assumptions are typically not entirely satisfied in practical applications. Thus, the performance of SLAM algorithm may be downgraded from the theoretical performance, which can potentially lead to

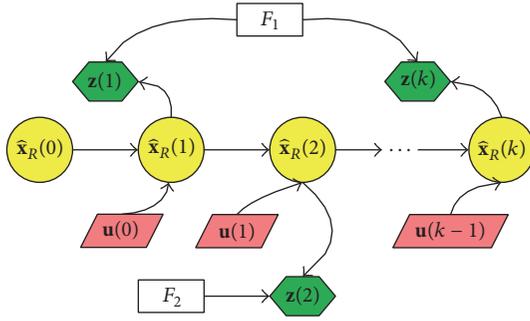


FIGURE 1: Schematic diagram of SLAM problem.

divergence. To prevent divergence and to improve the practicability of SLAM algorithm, the so-called adaptive filtering approach has been used in SLAM algorithm to dynamically adjust the parameters of the supposedly optimum filter based on estimating the unknown parameters for online estimation of motion and by estimating the signal and noise statistics from the available data. The adaptive filter can be realized to adjust important coefficients through fuzzy logic algorithm or neural network algorithm [10, 11]. Another popular method to realize the adaptive filtering is to use estimators [12–14].

To overcome the shortcomings of traditional adaptive filter, we developed a new nonlinear adaptive square root unscented Kalman filter (NASRUKF) which can be used in nonlinear or linear system for multisensory data fusion with uncertain process noise [15, 16]. In [17], the prototype of FastSLAM algorithm based on NASRUKF is described, and the validity is demonstrated by simple preliminary experiment.

In this paper, the improved FastSLAM algorithm based on NASRUKF is described in detail. To verify the accuracy of FastSLAM under the condition of low measurement frequency, the algorithms based on SRUKF and ASRUKF are compared and analyzed. And a complete experiment is designed for further validation.

2. FastSLAM Framework

The idea of FastSLAM algorithm comes from an analysis result of Dynamic Bayesian Network (DBN): if the vehicle pose is fully determined, then the map features are mutually independent. As shown in Figure 1, if the vehicle pose estimation $(\hat{\mathbf{x}}_R(0), \hat{\mathbf{x}}_R(1), \dots, \hat{\mathbf{x}}_R(k))$ is known, the estimations of map features F_1 and F_2 are independent from each other. The input information of FastSLAM includes the control information $\mathbf{u}(0 : k - 1)$ and the measurement information $\mathbf{z}(1 : k)$.

From a probabilistic point of view, SLAM problem is to solve the posterior probability distribution of system state vectors composed of vehicle pose vector \mathbf{x}_R and map feature vector \mathbf{x}_F , which can be expressed as follows [18]:

$$p(\mathbf{x}(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \mathbf{x}_R(0)). \quad (1)$$

From the aforementioned DBA analysis result and Bayesian formula, formula (1) can be expressed as follows:

$$\begin{aligned} & p(\mathbf{x}(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \mathbf{x}_R(0)) \\ &= p(\mathbf{x}_R(1 : k), \mathbf{x}_F(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \\ & \mathbf{x}_R(0)) = p(\mathbf{x}_R(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \mathbf{x}_R(0)) \quad (2) \\ & \cdot \prod_{i=1}^N p(\mathbf{x}_{F_i}(1 : k) | \mathbf{x}_R(1 : k), \mathbf{z}(1 : k)). \end{aligned}$$

Formula (2) shows the idea of FastSLAM algorithm that the joint SLAM state estimation can be factored into two independent estimations: vehicle pose component and map feature component. Thus, the problem of solving $(2N + 3)$ dimension state vectors is converted into two parts: a 3-dimension state vector solving of vehicle pose and N 2-dimension state vectors solving of map features.

For traditional FastSLAM algorithm, estimation of vehicle pose is achieved by particle filtering. Each map feature, in each particle, can be estimated using Extended Kalman Filters conditioned on the robot pose of the particle. And the factor weight of particles is calculated to determine the probability that a certain particle enters the final set.

The implementation process of FastSLAM algorithm is as follows. (1) The posterior probability distribution of vehicle pose is estimated using the control information and the motion model. (2) Map features of particles are associated with measurement information via maximum likelihood estimation. (3) According to correlated measurement information of each particle, update the estimations of each map feature and vehicle pose.

3. Optimization of FastSLAM

As the performance of FastSLAM depends on measurement information, the accuracy and reliability of traditional algorithm are downgraded when measurement data are limited by detecting frequency of sensors. In addition, the linearization method increases the estimation error for the reason that the actual system has strong nonlinearity and uncertain noise.

In [17], the basic idea of an modified FastSLAM was introduced simply. In this section, the modified FastSLAM introduced in [17] will be described in detail which is especially appropriate for low speed vehicle system (such as the wheelchair used for the aged). Considering low scanning speed sensors, a new decomposition strategy is designed for the posterior probability distribution. Moreover, an improved a posteriori estimation method is proposed which simultaneously estimates the joint state vectors comprised of vehicle pose and map features.

3.1. Posterior Probability Distribution Decomposition Strategy. The proposed posterior probability distribution decomposition strategy of state vectors is shown as

$$\begin{aligned}
 p(\mathbf{x}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)) &= p(\mathbf{x}_R(1:k), \mathbf{x}_F(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)) \\
 &= \begin{cases} \prod_{i=1}^N p(\mathbf{x}_{Ri}(k), \mathbf{x}_{Fi}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & (k=1) \\ \prod_{i=1}^N p((\mathbf{x}_R(1:k-1), \mathbf{x}_{Ri}(k), \mathbf{x}_{Fi}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & (k>1). \end{cases} \quad (3)
 \end{aligned}$$

Assuming that there are N particle state estimations $\widehat{\mathbf{X}}_1, \widehat{\mathbf{X}}_2, \dots, \widehat{\mathbf{X}}_n$ and relevant covariance matrices of estimation error $\mathbf{P}_{11}, \mathbf{P}_{22}, \dots, \mathbf{P}_{mm}$, the particle state estimations are irrelevant; that is, $\mathbf{P}_{ij} = 0$ ($i \neq j$). Then optimal estimation of the joint state vectors can be expressed as

$$\widehat{\mathbf{X}}_g = \frac{\sum_{i=1}^N \mathbf{P}_{ii}^{-1} \widehat{\mathbf{X}}_i}{\sum_{i=1}^N \mathbf{P}_{ii}^{-1}}. \quad (4)$$

The physical meaning of formula (4) is that if the estimation accuracy of $\widehat{\mathbf{X}}_i$ is low, its global contribution $\mathbf{P}_{ii}^{-1} \widehat{\mathbf{X}}_i$ will be low.

It can be drawn from (3) and (4) that

$$\widehat{\mathbf{x}}_R(k) = \frac{\sum_{i=1}^N \mathbf{P}_{Ri}^{-1}(k) \widehat{\mathbf{x}}_{Ri}(k)}{\sum_{i=1}^N \mathbf{P}_{Ri}^{-1}(k)}. \quad (5)$$

As shown in (3) and (5), the proposed FastSLAM algorithm converts the high-dimensional joint state estimation into several independent low dimensional joint state estimations. That is, the state vectors solving of $(2N+3)$ dimensions is converted into N 5-dimension state vectors solving.

Compared with the traditional FastSLAM algorithms, the correlation between vehicle pose and map features in low dimension state vectors is considered which is helpful to improve the accuracy and reliability of FastSLAM, especially when one frame data of the sensor can be used to observe at most one map feature which cannot create a new full map feature. This often appears in a vehicle system with lower measuring frequency sensors, such as sonar or infrared sensor, which can only obtain one map feature in a sampling cycle.

Equations (3) and (5) can be simplified as

$$\begin{aligned}
 p(\mathbf{x}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)) &= \begin{cases} p(\mathbf{x}_R(1:k), \mathbf{x}_{Foi}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & \mathbf{z} \text{ from Fo} \\ p(\mathbf{x}_R(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & \text{others} \end{cases} \quad (6) \\
 & \quad (i = 1, 2, \dots, N),
 \end{aligned}$$

where Fo represents the created map feature; \mathbf{x}_{Foi} represents the i th state vector with created feature. The meaning of formula (6) is that if the k th measurement data is from created feature, then joint estimation of vehicle pose and the created feature is carried out. If not, the vehicle pose is estimated only and the k th measurement data is saved to create a new map feature.

3.2. NASRUKF. For a nonlinear and discrete system, stochastic sequence is described by

$$\begin{aligned}
 x(k) &= f(x(k-1), u(k-1)) + w(k-1), \\
 z(k) &= h(x(k)) + v(k), \end{aligned} \quad (7)$$

where $f(x)$ and $h(x)$ are nonlinear functions of the system; $x(k)$ is the state vector of system; $u(k-1)$ is the control vector applied at time $k-1$; $w(k-1)$ is process noise; $z(k)$ is observation vector; $v(k)$ is the measurement noise.

The sigma points are calculated as

$$\begin{aligned}
 \chi(k-1) &= [\widehat{\mathbf{x}}(k-1) \quad \widehat{\mathbf{x}}(k-1) + \gamma \mathbf{S}(k-1) \quad \widehat{\mathbf{x}}(k-1) - \gamma \mathbf{S}(k-1)]. \end{aligned} \quad (8)$$

The time update equations are the following:

$$\begin{aligned}
 \widehat{\chi}^*(k) &= \mathbf{f}(\chi(k-1), \mathbf{u}(k-1)), \\
 \widehat{\mathbf{x}}^-(k) &= \sum_{i=0}^{2L} W_i^{(m)} \widehat{\chi}_i^*(k), \\
 \mathbf{S}^-(k) &= \mathbf{q}r \left\{ \left[\sqrt{W_1^{(c)}} (\widehat{\chi}_{1:2L}^*(k) - \widehat{\mathbf{x}}^-(k)) \quad \sqrt{\widehat{\mathbf{Q}}(k-1)} \right] \right\}, \\
 \mathbf{S}^-(k) &= \text{cholupdate} \left\{ \left[\mathbf{S}^-(k), \widehat{\chi}_0^*(k) - \widehat{\mathbf{x}}^-(k), W_0^{(c)} \right] \right\}. \end{aligned} \quad (9)$$

Augment sigma points are

$$\hat{\mathbf{x}}^-(k) = [\hat{\mathbf{x}}^-(k) \quad \hat{\mathbf{x}}^-(k) + \gamma \mathbf{S}^-(k) \quad \hat{\mathbf{x}}^-(k) - \gamma \mathbf{S}^-(k)]. \quad (10)$$

Estimate the square root of the measurement noise matrix:

$$\begin{aligned} \hat{\mathbf{Z}}^-(k) &= \mathbf{h}(\hat{\mathbf{x}}^-(k)), \\ \hat{\mathbf{z}}^-(k) &= \sum_{i=0}^{2L} W_i^{(m)} \hat{\mathbf{Z}}_i^-(k), \\ \tilde{\mathbf{z}}(k) &= \mathbf{z}(k) - \hat{\mathbf{z}}^-(k), \\ \sqrt{\mathbf{R}^{**}} &= \text{cholupdate} \left\{ \sqrt{1-d(k)} \sqrt{\hat{\mathbf{R}}(k-1)}, |\tilde{\mathbf{z}}(k)|, \right. \\ &\quad \left. d(k) \right\}, \end{aligned} \quad (11)$$

$$\sqrt{\mathbf{R}^*} = \text{cholupdate} \left\{ \sqrt{\mathbf{R}^{**}}, \hat{\mathbf{Z}}_{0:2L}^-(k) - \hat{\mathbf{z}}^-(k), -d(k) \cdot W_i^{(c)} \right\},$$

$$\sqrt{\hat{\mathbf{R}}(k)} = \text{diag} \left\{ \sqrt{\text{diag}(\sqrt{\mathbf{R}^*} \sqrt{\mathbf{R}^{*T}})} \right\}.$$

The measurement update equations are as follows:

$$\begin{aligned} \mathbf{P}_{\mathbf{z}\mathbf{z}}(k) &= \sum_{i=0}^{2L} W_i^{(c)} (\hat{\mathbf{x}}_i^-(k) - \hat{\mathbf{x}}^-(k)) (\hat{\mathbf{Z}}_i^-(k) - \hat{\mathbf{z}}^-(k))^T, \\ \mathbf{S}_{\mathbf{z}}(k) &= qr \left\{ \left[\sqrt{W_1^{(c)}} (\hat{\mathbf{Z}}_{1:2L}^-(k) - \hat{\mathbf{z}}^-(k)) \quad \sqrt{\hat{\mathbf{R}}(k)} \right] \right\}, \\ \mathbf{S}_{\mathbf{z}}(k) &= \text{cholupdate} \left\{ \left[\mathbf{S}_{\mathbf{z}}(k), \hat{\mathbf{Z}}_0^-(k) - \hat{\mathbf{z}}^-(k), W_0^{(c)} \right] \right\}, \quad (12) \\ \mathbf{K}(k) &= \frac{(\mathbf{P}_{\mathbf{z}\mathbf{z}}(k) / \mathbf{S}_{\mathbf{z}}(k)^T)}{\mathbf{S}_{\mathbf{z}}(k)}, \end{aligned}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k) \tilde{\mathbf{z}}(k),$$

$$\mathbf{U} = \mathbf{K}(k) \mathbf{S}_{\mathbf{z}}(k),$$

$$\mathbf{S}(k) = \text{cholupdate} \{ \mathbf{S}^-(k), \mathbf{U}, -1 \},$$

where $\gamma = \sqrt{L+\lambda}$; $d(k) = (1-b)/(1-b^{k+1})$ and b is the forgetting factor, typically $0 < b < 1$; the weights ($W_i^{(m)}$ and $W_i^{(c)}$) of the mean and covariance are given by

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{L+\lambda}, \\ W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta, \quad (13) \\ W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, \quad i = 1, \dots, 2L, \end{aligned}$$

where $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter. The constant α determines the spread of the sigma points around the mean,

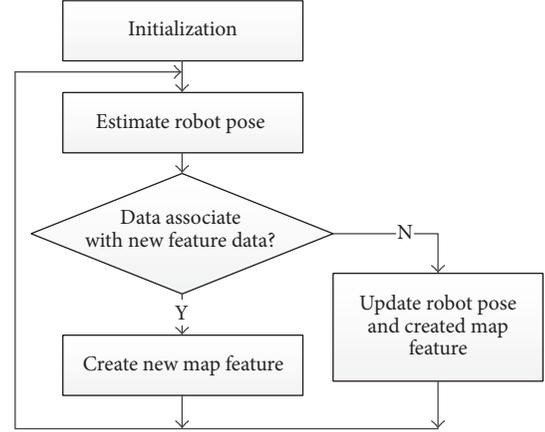


FIGURE 2: Implementation process of FastSLAM algorithm.

which is typically set to a small, positive value. The constant $\kappa \geq 0$ is a secondary scaling parameter. $\beta \geq 0$ is used to incorporate the prior knowledge of the distribution (for Gaussian distributions, $\beta = 2$ is optimal).

4. Implementation of New FastSLAM Algorithm

The proposed FastSLAM algorithm implementation process is shown as in Figure 2.

Initialization. During initialization, the initial pose of vehicle is obtained by the static state estimation method based on the measurement data of sensors. In addition, the initial feature of the map is created using the feature extraction method.

After initialization, the vehicle location and mapping will be calculated and updated based on proposed FastSLAM algorithm in the following steps.

Vehicle Pose Estimation. The vehicle pose is estimated using ASRUKF.

Data Association. According to (14), (15), and (16), data association of the measurement data at time k is dealt with to determine whether it is a created map feature. If the measurement data is a created map feature, algorithm turns to vehicle pose and map feature updating; otherwise, the data shall be stored as new feature data, and algorithm turns to new map feature creating.

Vehicle Pose and Map Feature Updating. According to [17], this step consists of two cases. The first one is that the measurement data ($\rho(k), \varphi(k)$) matches the created linear feature. The second case is that the measurement data matches the created arc feature.

(1) The Measurement Data Matches the Created Linear Feature. If the line segment is not parallel to Y_I -axis, that is, $y = a_j x + b_j$, it can be defined that $\mathbf{x}_j(k) = [x(k), y(k), \theta(k), a_j(k), b_j(k)]^T$ is the state variable of the nonlinear discrete system described as

$$\begin{aligned}
\mathbf{x}_j(k) &= \mathbf{x}_j(k-1) + \mathbf{w}_j(k-1), \\
z_j(k) &= \rho(k) = \frac{d_S}{|\cos \theta_j|} = \frac{d_S}{|\cos(\alpha_j - \theta(k) - \varphi(k))|} \\
&= \left| \frac{a_j(k) \cdot (x(k) - Ls \cdot \cos \theta(k)) - (y(k) - Ls \cdot \sin \theta(k)) + b_j(k)}{-a_j(k) \cos(\theta(k) + \varphi(k)) + \sin(\theta(k) + \varphi(k))} \right| + v_j(k).
\end{aligned} \tag{14}$$

If the line segment is parallel to Y_I -axis, that is, $x = c_j$, $\mathbf{x}_j(k) = [x(k), y(k), \theta(k), c_j(k)]^T$ can be defined as the state variable of the nonlinear discrete system, which is described by

$$\begin{aligned}
\mathbf{x}_j(k) &= \mathbf{x}_j(k-1) + \mathbf{w}_j(k-1), \\
z_j(k) &= \rho(k) \\
&= \left| \frac{(x(k) - Ls \cdot \cos \theta(k)) - c_j(k)}{\cos(\theta(k) + \varphi(k))} \right| + v_j(k),
\end{aligned} \tag{15}$$

where \mathbf{w}_j and v_j have the statistical property of zero mean, mutual independence, and Gaussian distribution. The covariance matrix of \mathbf{w}_j and v_j is \mathbf{Q}_j and R_j , respectively.

(2) *The Measurement Data Matches the Created Arc Feature.* It can be defined that $\mathbf{x}_{c_j}(k) = [x(k), y(k), \theta(k), x_{c_j}(k), y_{c_j}(k), R_{c_j}(k)]^T$ is the state variable of the nonlinear discrete system described as

$$\begin{aligned}
\mathbf{x}_{c_j}(k) &= \mathbf{x}_{c_j}(k-1) + \mathbf{w}_{c_j}(k-1), \\
z_{c_j}(k) &= \rho(k) \\
&= d_S |\cos \theta_j| \pm \sqrt{d_S^2 \cos^2 \theta_j - (d_S^2 - R_{c_j}^2(k))} \\
&\quad + v_{c_j}(k),
\end{aligned} \tag{16}$$

where

$$d_S = \sqrt{(x(k) - Ls \cos \theta(k) - x_{c_j}(k))^2 + (y(k) - Ls \sin \theta(k) - y_{c_j}(k))^2}. \tag{17}$$

New Map Feature Creating. If the number of new feature data exceeds threshold value N_{new} , a new map feature should be created.

5. Experimental Results

In order to evaluate the performance of proposed FastSLAM algorithm, experimental results are shown in this section. The test environment is built as Figure 3. The main parameters are depicted as follows: $Ls = 180$ mm, the length threshold of linear segment is $L_{\min} = 500$ mm, the distance threshold between point and line is $d_{\max} = 80$ mm, and the threshold of average vector distance is $md_{\max} = 30$ mm.

As shown in Figure 3, CPU Motion Controller implements the proposed FastSLAM algorithm and uploads the results to the computer via the receiving node R_1 and the transmitting node T_1 for real-time display.

The whole experiment process is divided into the static state estimation and the dynamic update.

5.1. The Static State Estimation. The surroundings are scanned by the sensors of vehicle under stationary state. The scanned data set is clustered by adaptive breakpoint detector.

Then the scanned data set is segmented:

- (1) First, the point C is detected as the maximum distance from line AB which exceeds the threshold. Thus, the continuous data set AB is divided into two subsets of AC and CB. Then, the point D is detected as the maximum distance from line AC which exceeds the threshold and the data set AC is divided into two subsets of AD and DC. Similarly, the data set CB is divided into two subsets of CE and EB and the data set EB is divided into two subsets of EF and FB, as shown in Figure 4(a). At last, the distance between point C and line DE is detected to be less than the threshold. Thus, the subsets DC and CE are combined into one data set DE.
- (2) Second, the lengths of AD, DE, EF, and FB are calculated, which exceed the line threshold. Thus, the subsets AD, DE, EF, and FB are processed as line segment. Then the subsets FB are analyzed by most similar algorithm which belongs to the arc segment, as shown in Figure 4(b).

5.2. Dynamic Update

(1) *Evaluation of Vehicle Pose Estimation Accuracy.* The real-time pose is estimated using ASRKF and SRKF, respectively, when the vehicle travels along the desired trajectory. The

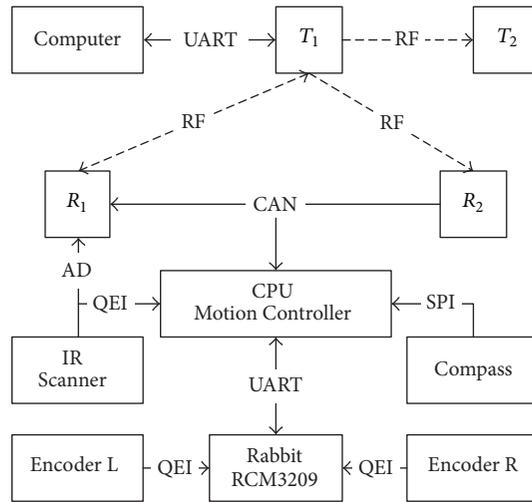


FIGURE 3: Schematic diagram of test data transmission.

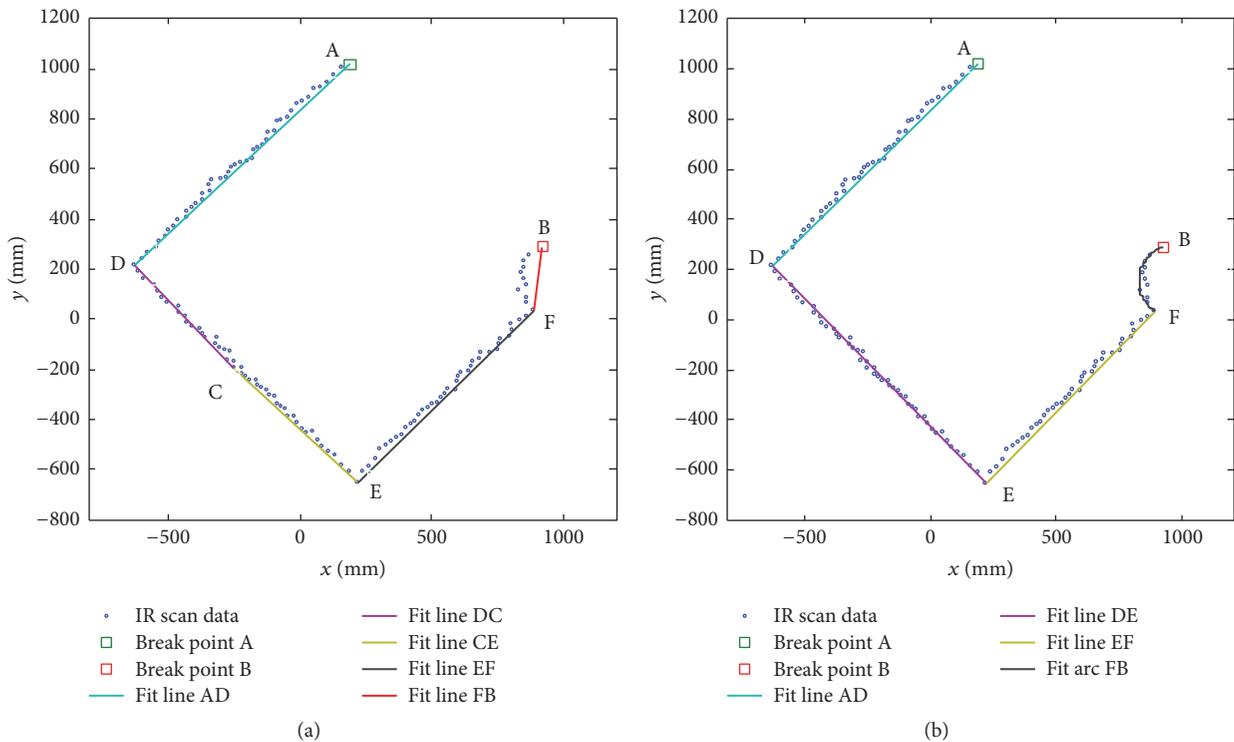


FIGURE 4: The static state estimation of map features.

statistic characteristic of process noise is unknown. The data measurement period is 25 ms.

The result of experiment is shown as in Figure 5. It is observed from Figure 5 that the errors of vehicle state (xy and θ) estimated by NASRUKF are significantly smaller than the errors estimated by SRUKF. After 4 seconds, as the identification of process noise statistical feature, the conclusions are more obvious. (1) From Figure 5(a), the estimated error of x - y plane calculated by ASRUKF is within 1 cm, while the estimated error calculated by SRUKF is within 5 cm. (2)

From Figure 5(b), the estimated error of azimuth calculated by ASRUKF is within 1.15° (0.02 rad), while the estimated error calculated by SRUKF is almost 5.16° (0.09 rad).

(2) *Evaluation of Map Creating Accuracy.* When vehicle slowly moves along the line, the vehicle track and environmental map feature are dynamically updated. The test results are shown in Figure 6 and Table 1.

Note. Full lines in the figure (real map, real track, and real theta) denote the true value of actual state. Short dash

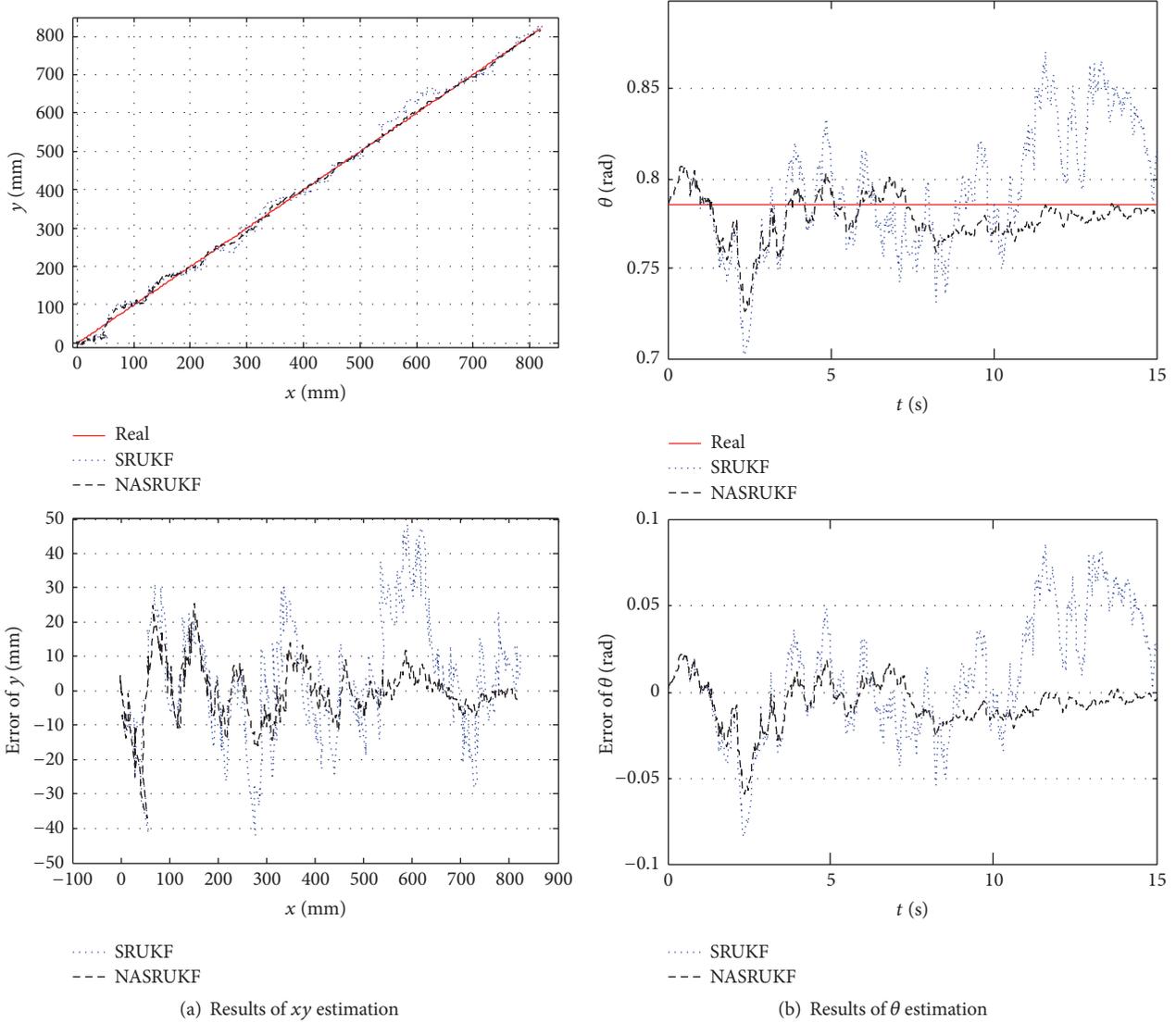


FIGURE 5: Results of vehicle pose estimation.

TABLE 1: Error in created environmental features.

(a)

Feature	Line L_{AD}		Line L_{DE}		Line L_{EF}		Arc FB	
	Slope	Intercept	Slope	Intercept	Slope	Intercept	Center	Radius
Error	0.0616	-10.6518 mm	-0.0292	7.8651 mm	0.0247	-15.5775 mm	15.5822 mm	-13.3833 mm

(b)

Feature	Break point A	Corner point D	Corner point E	Corner point F
Error	21.7072 mm	20.3451 mm	18.5317 mm	31.0601 mm

lines (estimate map) denote the map feature estimated with the improved FastSLAM algorithm. Long dash lines (estimate track and estimate theta) denote the state estimation or estimation error using the improved FastSLAM algorithm.

As shown in Figure 6 and Table 1, the localization accuracy based on the improved FastSLAM algorithm is within 1 cm and the azimuthal error is approximately 0.5° . The position error of the corner point in the map feature is approximately 3 cm. The errors in both the center position

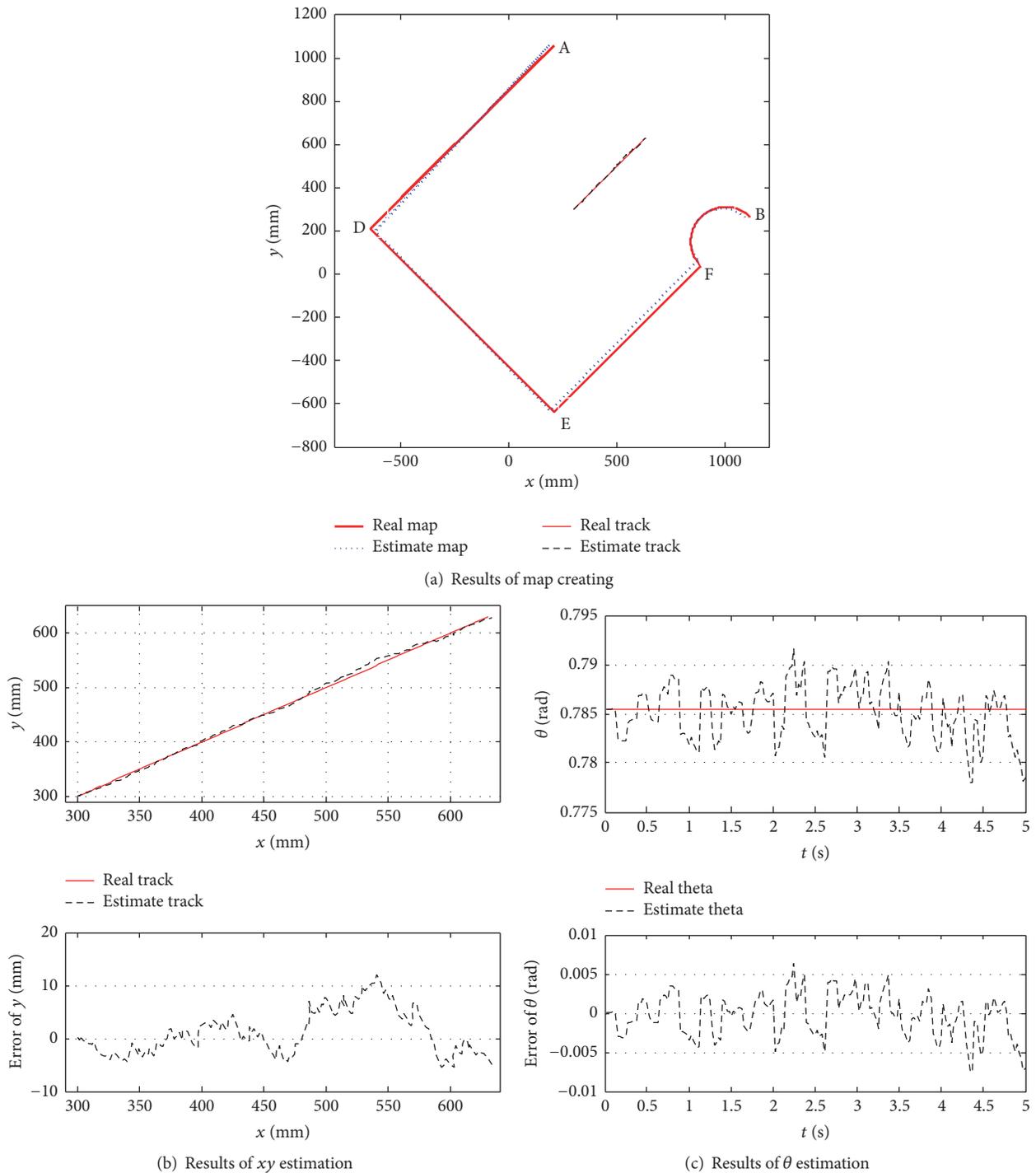


FIGURE 6: Results of SLAM dynamic update.

and radius of the arc are within 2 cm. The slope error of line segment is within 0.07. The intercept error is within 2 cm. It is demonstrated by the test results that the proposed FastSLAM algorithm is effective.

6. Conclusions

An improved FastSLAM algorithm based on NASRUKF is proposed in paper. To improve the accuracy and reliability

of the FastSLAM limited by measuring frequency, a new posterior probability distribution decomposition strategy is proposed. And the vehicle pose and map features are estimated using NASRUKF algorithm, which overcomes the limitations of the traditional FastSLAM algorithm (e.g., solutions to Jacobian matrix are needed; failure to meet consistency conditions and occurrence of “degeneration of particle set”). The adaptive ability of filters is improved while SLAM accuracy is maintained, which provides an

effective way to solve SLAM problem. The effectiveness of the proposed FastSLAM algorithm is verified by indoor experiment results.

Conflicts of Interest

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

Acknowledgments

The authors would like to express their great appreciation to the National Natural Science Foundation of China (no. 51307137), the doctoral scientific research foundation of XUST (no. 2016018), the Fundamental Research Funds of Xi'an University of Science & Technology (no. 201317), and the open fund project of the key laboratory for embedded system and service computing (Tongji University) of the education ministry of China (no. ESSCKF 2016-05).

References

- [1] M. Cugliari and F. Martinelli, "A FastSLAM algorithm based on the unscented filtering with adaptive selective resampling," in *Field and Service Robotics*, vol. 42 of *Springer Tracts in Advanced Robotics*, pp. 359–368, Springer, Berlin, Germany, 2008.
- [2] D. Scaramuzza, R. Siegwart, and A. Martinelli, "A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics," *International Journal of Robotics Research*, vol. 28, no. 2, pp. 149–171, 2009.
- [3] Y. Song, Y. Song, and Q. Li, "Robust iterated sigma point FastSLAM algorithm for mobile robot simultaneous localization and mapping," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 24, no. 4, pp. 693–700, 2011.
- [4] Z. Kurt-Yavuz and S. Yavuz, "A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms," in *Proceedings of the IEEE 16th International Conference on Intelligent Engineering Systems (INES '12)*, pp. 37–43, June 2012.
- [5] A. Monjazez, J. Z. Sasiadek, and D. Neculescu, "Autonomous navigation among large number of nearby landmarks using FastSLAM and EKF-SLAM—a comparative study," in *Proceedings of the 16th International Conference on Methods and Models in Automation and Robotics (MMAR '11)*, pp. 369–374, IEEE, Międzyzdroje, Poland, August 2011.
- [6] D. Trivun, E. Salaka, D. Osmankovic, J. Velagic, and N. Osmic, "Active SLAM-based algorithm for autonomous exploration with mobile robot," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '15)*, pp. 74–79, March 2015.
- [7] Y. Song, Q. Li, Y. Kang, and D. Yan, "Effective cubature FastSLAM: SLAM with Rao-Blackwellized particle filter and cubature rule for Gaussian weighted integral," *Advanced Robotics*, vol. 27, no. 17, pp. 1301–1312, 2013.
- [8] H. Ankishan, E. O. Tartan, and F. Ari, "Square root unscented based FastSlam optimized by particle swarm optimization passive congregation," in *Proceedings of the 10th IEEE International Conference on Mechatronics and Automation (ICMA '13)*, pp. 469–475, August 2013.
- [9] R. Havangi, H. D. Taghirad, M. A. Nekoui, and M. Teshnehlab, "A square root unscented fastSLAM with improved proposal distribution and resampling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2334–2345, 2014.
- [10] H. Du, Y. Hao, and Y. Zhao, "SLAM algorithm based on fuzzy adaptive Kalman filter," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 40, no. 1, pp. 58–62, 2012.
- [11] Q.-L. Li, Y. Song, and Z.-G. Hou, "Neural network based FastSLAM for autonomous robots in unknown environments," *Neurocomputing*, vol. 165, pp. 99–110, 2015.
- [12] H. Wang, G. Fu, J. Li, Z. Yan, and X. Bian, "An adaptive UKF based SLAM method for unmanned underwater vehicle," *Mathematical Problems in Engineering*, vol. 2013, Article ID 605981, 12 pages, 2013.
- [13] A. Torres-González, J. M. Ramiro-de Dios, and A. Ollero, "An adaptive scheme for robot localization and mapping with dynamically configurable inter-beacon range measurements," *Sensors (Switzerland)*, vol. 14, no. 5, pp. 7684–7710, 2014.
- [14] D. Liu, J.-M. Duan, and H.-X. Yu, "FastSLAM algorithm based on adaptive fading extended Kalman filter," *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, vol. 38, no. 3, pp. 644–651, 2016.
- [15] Z. Yong, Z. Yufeng, and Z. Juzhong, "A new adaptive square-root unscented kalman filter for nonlinear systems," *Applied Mechanics and Materials*, vol. 300–301, pp. 623–626, 2013.
- [16] Y. Zhou, C. Zhang, Y. Zhang, and J. Zhang, "A new adaptive square-root unscented kalman filter for nonlinear systems with additive noise," *International Journal of Aerospace Engineering*, vol. 2015, Article ID 381478, 9 pages, 2015.
- [17] J. Zhang, Y. Jiang, and K. Wang, "A modified FastSLAM for an autonomous mobile robot," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 1755–1759, Heilongjiang, China, August 2016.
- [18] J. C. Prieto, A. R. Jimenez, J. Guevara et al., "Performance evaluation of 3D-LOCUS advanced acoustic LPS," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2385–2395, 2009.

Research Article

Kinematics, Dynamics, and Optimal Control of Pneumatic Hexapod Robot

Long Bai,¹ Lu-han Ma,¹ Zhifeng Dong,¹ and Xinsheng Ge²

¹*School of Mechanical Electronic & Information Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China*

²*Mechanical & Electrical Engineering School, Beijing Information Science & Technology University, Beijing 100192, China*

Correspondence should be addressed to Long Bai; bailong0316jn@126.com

Received 8 August 2016; Revised 16 January 2017; Accepted 12 February 2017; Published 9 March 2017

Academic Editor: Francisco Valero

Copyright © 2017 Long Bai 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 hexapod robot is driven by inert gas carried by itself, which has board application prospect in rescue operation of disaster conditions containing flammable gas. Cruising ability is main constraint for practical engineering application which is influenced by kinematics and dynamics character. The matrix operators and pseudospectral method are used to solve dynamics modeling and numerical calculation problem of robot under straight line walking. Kinematics model is numerically solved and relationship of body, joints, and drive cylinders is obtained. With dynamics model and kinematics boundary conditions, the optimal input gas pressure of leg swing and body moving in one step is obtained by pseudospectral method. According to action character of magnetic valve, calculation results of control inputs satisfy engineering design requirements, and cruising ability under finite gas is obtained.

1. Introduction

In recent years, more and more robots are used in industrial accident's detect and rescue operation. The commonly used robots include motor and hydraulic drive types, but they are not suited for some close space accident environment which fills with flammable gas, such as gas explosion accident of coal mine, because electric devices of them may lead to secondary explosion. The pneumatic robot is driven by inert gas and is convenient to be controlled, which is widely used in industrial and medical domains. Verrelst et al. [1] designed a pneumatic biped robot, which verifies feasibility of using pneumatic system as power source, Lavoie and Desbiens [2] designed a cockroach type pneumatic hexapod robot, Morimoto et al. [3] designed a rehabilitation used soft touch manipulator by soft cylinder and obtained a high working accuracy, Qiu et al. [4] designed a pipe inspection robot by soft cylinders, Diez et al. [5] designed a neural rehabilitation pneumatic robot, Low et al. [6] explored a soft pneumatic massager used in joints auxiliary motion, and Ramsauer et al. [7] explored an error detection using pneumatic Stewart platform.

With these backgrounds, a natural antiexplosion pneumatic hexapod robot (PHR) which is driven by inert gas is designed in this exploration. However, cruising ability is a big influence in robot's engineer application, for the carried gas's volume is limited by self-weight of robot. The cruising ability of PHR is measured by straight line walking distance limited by product of volume and pressure of carried gas. During straight line walking, the same characters of each gait decide they have same gas consumption, so cruising ability problem changes to be calculation of distance and gas consumption of one step. Gas consumption of one step is defined as product of cylinder's volume and drive pressure. The cylinder volume is known, and pressure is influenced by dynamics character of robot. The optimal control method is used to calculate minimum drive pressure.

In the last few years, there are many explorations on optimal control problems of hexapod robots. Sliva and Machado [8] reviewed optimization method used in legged robots; energy/power optimal control objective functions are listed out; Sanz-Merodio et al. [9] explored energy consumption of mammal and insect type robots and concluded that leg

dynamics accounts for most energy consumption; Chen et al. [10] designed an insect type hexapod robot and leg has a series mechanism type; the optimal control of leg swing is solved by pseudospectral method; Roy et al. [11–14] explored kinematic dynamics and optimal control problems of hexapod robot; the hexapod robot is driven by electric motor, so it has a series mechanism type; Luneckas et al. [15] analyzed hexapod robot's energy consumption by motion of body and step height; Deng et al. [16] explored energy reducing problem of hexapod robot by kinematics analysis; Gonzalez de Santos et al. [17] explored minimization of hexapod robot in irregular terrain; the optimal analysis is based on statically stable gait; Jin et al. [18] explored hexapod walking robot's power consumption optimization problem by torque distribution algorithm and parameters include duty factor, stride length, bogy height, and foot trajectory lateral offset; Zhu et al. [19] explored optimal design of hexapod robot with kinematic model.

Fundamentally, PHR is a parallel mechanism, most of optimal control explorations of it are static, or simplify it as serial mechanism, so the real dynamics character cannot faithfully represent it. The first reason is that complex dynamics character of parallel mechanism makes it difficult to use triangle functions to calculate it, and complex triangle and antitriangle transformations will lead to unsolvable model. Secondly, complex nonlinear characters need optimal control algorithm that has high calculation accuracy and stability, but classic algorithms such as Runge-Kutta method do not satisfy these two characters.

According to references of dynamics modeling by Lie group [20, 21] and optimal control with pseudospectral method [22, 23], the matrix and vector operators can avoid triangle and antitriangle transformations which makes dynamics modeling easier. Pseudospectral method is a global numerical method which has high stability and is widely used in many domains, many engineering problems are solved successfully [24, 25]. So, in this exploration, matrix and vector operators are used as units for dynamic modeling, and optimal control problem is solved by pseudospectral method. The control inputs curves which satisfy pneumatic control characters are obtained, and then cruising ability calculation method is built at last, which offers a reference for the improvement of robot.

2. The Mechanism and Gait of Pneumatic Hexapod Robot

PHR is a biorobot, so it has two design schemes, insect type in Figure 1 and mammal type in Figure 2. Many people like insect type, but it has some problems. Firstly, it walks along X direction by triangle gait of hip; joints bear great yawing forces which will lead to joints' rapid abrasion and to not being suited for engineering application. Secondly, it has a big width, so crosswise passing ability is restricted. Thirdly, the realization of straight line walking needs combination motion of three joints, which is more difficult to realize by pneumatic system and has high gas consumption. The mammal type PHR does not have these problems. The nitrogen gas bottle is in trunk; the maximum pressure can reach 15 MPa. The

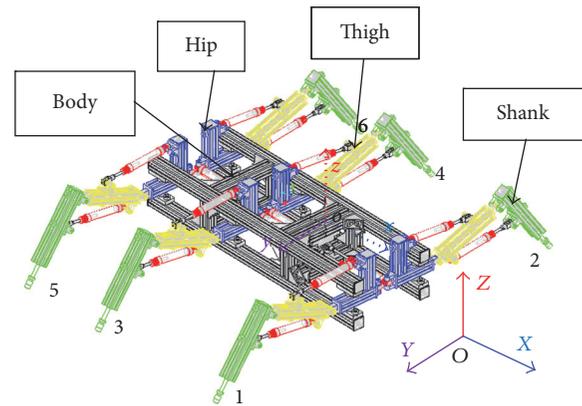


FIGURE 1: The insect type hexapod robot.

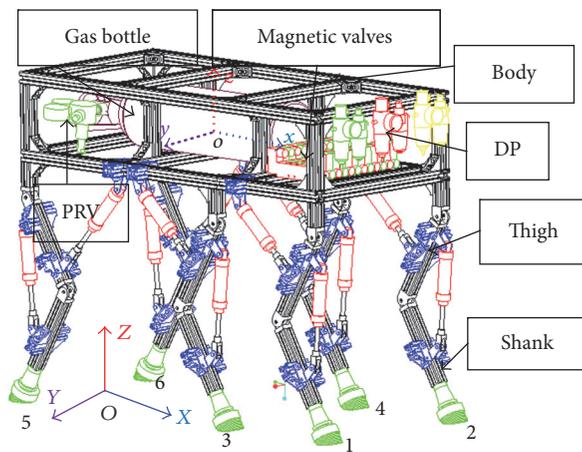


FIGURE 2: The mammal type hexapod robot.

high pressure gas is decompressed to 1 MPa by PRV (pressure reducing valve), and then gas can be decomposed to different low pressures from 0.15 MPa to 0.8 MPa by DP (duplex pieces). All the magnetic valves and control devices can be packaged in box which is convenient for antiexplosion design. Each leg is composed of shank and thigh which are driven by cylinder.

According to comparison between Figures 1 and 2, each leg of mammal type hexapod robot has two joints which is less than the insect type. Based on this mechanism, the PHR can realize straight line walking as gait in Figure 3. The gait can be divided into four actions. Firstly, the shank drive cylinder shrinks to make foot tip separate from ground; secondly, the thigh drive cylinder stretches out to make the leg step forward; thirdly, the shank drive cylinder stretches out to make foot contact with ground again; fourthly, the thigh drive cylinder shrinks to make body move forward. According to the above gait analysis, the two joints work at different time and have no coincides. So the swing of thigh and shank can be treated as a parallel pendulum. When body moves forward by support of legs, PHR and ground form a close loop. According to straight line walking gait, the movement of body is only decided by motion of thigh. So the movement

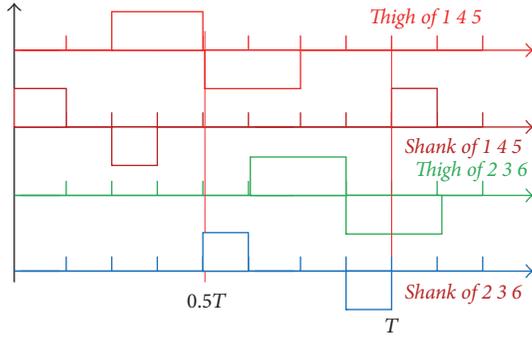


FIGURE 3: The straight line walking gait of PHR.

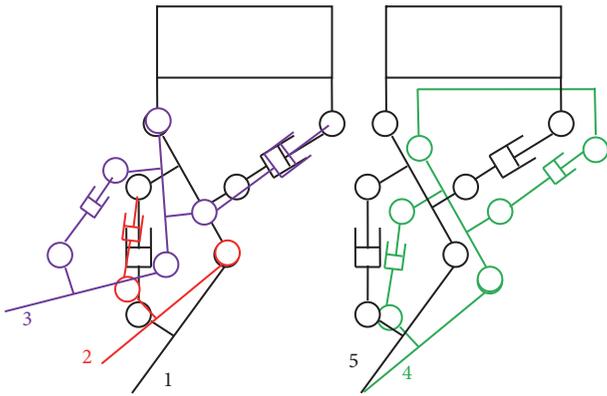


FIGURE 4: The leg's motion process under straight walking gait.

process can be expressed only by one parameter. The motion process of straight line walking can be expressed as in Figure 4.

3. Kinematics Modeling of PHR

In this part, the pneumatic hexapod robot's whole straight line walking kinematic model is built with matrix and vector operators. The parallel pendulum kinematic model that corresponds to leg swing process is built at first. Then the kinematics model of whole machine under straight line walking process is derived as follows.

3.1. The Kinematic Model of Leg Swing. The mechanism and parameters of thigh and shank are given in Figure 5. According to Figure 5, thigh and shank have same mechanism which consists of rocker (O_1O_3) and push rod O_2O_3 . So thigh and shank can be expressed by same kinematic model. Suppose rotate angle of rocker along O_1 is θ_1 , rotate angle of push rod along O_2 is θ_2 , and length of push rod is l , rotation matrix of rocker and push rod can be written as

$$\mathbf{R}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}, \quad i = 1, 2. \quad (1)$$

The rotation matrix satisfies $\mathbf{R}\mathbf{R}^T = \mathbf{I}_{2 \times 2}$, $\det \mathbf{R} = 1$. Suppose position vector of point O_3 on rocker is $\mathbf{r}_1 = (x_a, y_a)$,

so the coordinate of O_3 in inertial frame is $\mathbf{R}_1\mathbf{r}_1$. Suppose position vector of O_2 in inertial frame is $\mathbf{r}_2 = (x_b, y_b)$. According to translate process as $O_1 \rightarrow O_2 \rightarrow O_3$, the coordinate of O_3 in inertial frame is $\mathbf{r}_2 + l\mathbf{R}_2\mathbf{e}_1$. According to coordinates of O_2 , O_3 , and length between them, the relation can be expressed as

$$\|\mathbf{R}_1\mathbf{r}_1 - \mathbf{r}_2\|^2 = l^2. \quad (2)$$

Equation (2) can be unfolded as

$$\mathbf{r}_1^T\mathbf{r}_1 - 2\mathbf{r}_2^T\mathbf{R}_1\mathbf{r}_1 + \mathbf{r}_2^T\mathbf{r}_2 = l^2. \quad (3)$$

Equation (1) can be decomposed by $\mathbf{R}_i = p_i\mathbf{I} + q_i\mathbf{S}(1)$. Bring it into (3), and p_i , q_i can be solved by nonlinear equations as (4). $p_i^2 + q_i^2 = 1$ is the constraint between p_i and q_i .

$$q_iA + Bp_i = C \quad (4)$$

$$p_i^2 + q_i^2 = 1.$$

In (4), $A = \mathbf{r}_2^T\mathbf{S}(1)\mathbf{r}_1$, $B = \mathbf{r}_2^T\mathbf{r}_1$, $C = (\mathbf{r}_1^T\mathbf{r}_1 + \mathbf{r}_2^T\mathbf{r}_2 - l^2)/2$. According to (4), \mathbf{R}_1 can be calculated. The relation of push rod and rocker's attitudes and length of push rod is

$$l\mathbf{R}_2\mathbf{e}_1 = \mathbf{R}_1\mathbf{r}_1 - \mathbf{r}_2. \quad (5)$$

The attitude matrix satisfies the differential relation as $\dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\omega)$; then, differentiating (3), the velocity relation can be derived as

$$\omega_1\mathbf{r}_1^T\mathbf{S}(1)\mathbf{R}_1^T\mathbf{r}_2 = l\dot{v}. \quad (6)$$

In (6), $\dot{l} = v$. After transposition of terms, the relation between angular velocity of rocker and velocity of push rod is written as

$$\omega_1 = \frac{l}{\mathbf{r}_1^T\mathbf{S}(1)\mathbf{R}_1^T\mathbf{r}_2}v. \quad (7)$$

Differentiating (3), the relation of pose, attitude, velocity, and angular velocity is written as

$$\omega_2 l \mathbf{R}_2 \mathbf{S}(1) \mathbf{e}_1 = \omega_1 \mathbf{R}_1 \mathbf{S}(1) \mathbf{r}_1 - v \mathbf{R}_2 \mathbf{e}_1. \quad (8)$$

Systemizing (8), the relation between angular velocity of push rod and rocker is written as

$$\omega_2 = \omega_1 \frac{\mathbf{e}_2^T \mathbf{R}_2^T \mathbf{R}_1 \mathbf{S}(1) \mathbf{r}_1}{l}. \quad (9)$$

Differentiating (6), and then systemizing it, the angular acceleration of rocker is written as

$$\dot{\omega}_1 = \frac{v^2 + la + \omega_1^2 \mathbf{r}_1^T \mathbf{R}_1 \mathbf{r}_2}{\mathbf{r}_1^T \mathbf{S}(1) \mathbf{R}_1^T \mathbf{r}_2}. \quad (10)$$

In (10), $a = \dot{v}$. Differentiating (8), the angular acceleration of push rod is written as

$$\begin{aligned} \dot{\omega}_2 = & \omega_1^2 \frac{\mathbf{e}_1^T \mathbf{S}(1) \mathbf{R}_2^T \mathbf{R}_1 \mathbf{r}_1}{l} - 2 \frac{\omega_2 v}{l} \\ & - \dot{\omega}_1 \frac{\mathbf{e}_1^T \mathbf{S}(1) \mathbf{R}_2^T \mathbf{R}_1 \mathbf{S}(1) \mathbf{r}_1}{l}. \end{aligned} \quad (11)$$

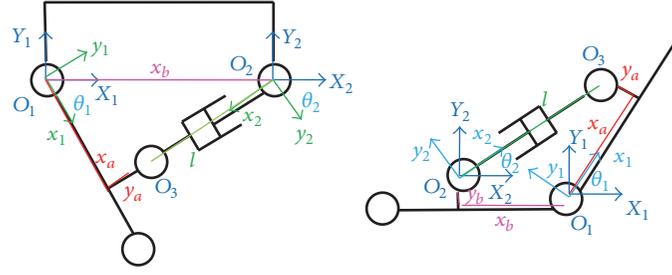


FIGURE 5: The parallel pendulum diagram of thigh and shank.

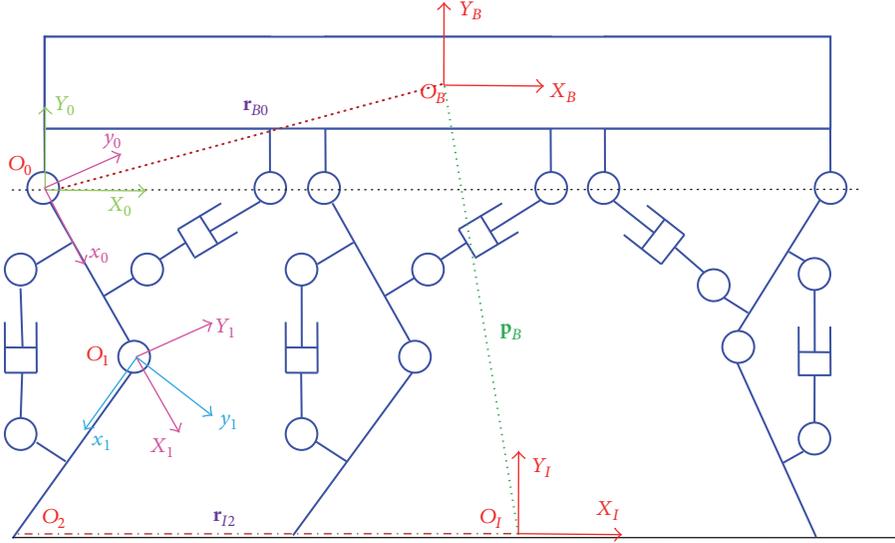


FIGURE 6: The mechanisms schematic diagram of pneumatic hexapod robot under straight line walking.

3.2. Kinematic Model of Body Moving. The movement of body is expressed by pose of body frame's origin O_B relative to inertial frame $O_I X_I Y_I$ as in Figure 6. Define \mathbf{p}_B and \mathbf{r}_{I2} as position vectors of body and foot tip in inertial frame $O_I X_I Y_I$, respectively, \mathbf{r}_{B0} is position vector of thigh joint O_0 in frame $O_B X_B Y_B$, \mathbf{r}_{01} and \mathbf{r}_{12} are position vectors of shank joint O_1 in thigh's body frame $O_0 x_0 y_0$ and foot tip O_2 in shank's body frame $O_1 x_1 y_1$, respectively. Supposing that rotation matrixes of thigh and shank are \mathbf{R}_0 and \mathbf{R}_1 , respectively, so the close loop relation of body, thigh, shank, and ground is derived as

$$\mathbf{r}_{I2} - \mathbf{r}_{B0} - \mathbf{R}_0 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) = \mathbf{p}_B. \quad (12)$$

According to (12) and straight line walking gait, the position of body is decided by rotation of thigh for shank joint keeps still during body moves. The velocity relation of leg and body is derived out by differential calculation of (12) and the result is

$$-\dot{\mathbf{R}}_0 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) - \mathbf{R}_0 \dot{\mathbf{R}}_1 \mathbf{r}_{12} = \dot{\mathbf{p}}_B. \quad (13)$$

Equation (13) can be written as an expansion type as

$$-\omega_0 \mathbf{R}_0 \mathbf{S}_1 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) - \omega_1 \mathbf{R}_0 \mathbf{R}_1 \mathbf{S}_1 \mathbf{r}_{12} = \mathbf{v}_B. \quad (14)$$

The acceleration relation of leg and body is derived by differentiating (14), and the result is

$$\begin{aligned} & -\dot{\omega}_0 \mathbf{R}_0 \mathbf{S}_1 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) - \omega_0 \dot{\mathbf{R}}_0 \mathbf{S}_1 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) \\ & - \omega_0 \mathbf{R}_0 \mathbf{S}_1 \dot{\mathbf{R}}_1 \mathbf{r}_{12} - \dot{\omega}_1 \mathbf{R}_0 \mathbf{R}_1 \mathbf{S}_1 \mathbf{r}_{12} - \omega_1 \dot{\mathbf{R}}_0 \mathbf{R}_1 \mathbf{S}_1 \mathbf{r}_{12} \\ & - \omega_1 \mathbf{R}_0 \dot{\mathbf{R}}_1 \mathbf{S}_1 \mathbf{r}_{12} = \mathbf{a}_B. \end{aligned} \quad (15)$$

The neat type of (15) is

$$\begin{aligned} & -\alpha_0 \mathbf{R}_0 \mathbf{S}_1 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) - \alpha_1 \mathbf{R}_0 \mathbf{R}_1 \mathbf{S}_1 \mathbf{r}_{12} \\ & + \omega_0^2 \mathbf{R}_0 (\mathbf{r}_{01} + \mathbf{R}_1 \mathbf{r}_{12}) + (2\omega_0 \omega_1 + \omega_1^2) \mathbf{R}_0 \mathbf{R}_1 \mathbf{r}_{12} \\ & = \mathbf{a}_B. \end{aligned} \quad (16)$$

4. Dynamics Modeling of PHR

4.1. Dynamics Model of Leg Swing. With kinematics model, the dynamics model of parallel single pendulum can be built by Lagrange theory. For the control objective is rocker, so rotation angle θ_1 of rocker is chosen as generalized coordinate of dynamics system. The complete expression form of

rotation matrix of push rod \mathbf{R}_2 can be derived by multiplying \mathbf{e}_1^T and \mathbf{e}_2^T on both sides of (5), respectively. It shows in

$$\mathbf{R}_2 = \frac{n_1 \mathbf{I} + n_2 S(1)}{l}. \quad (17)$$

In (17), $n_1 = \mathbf{e}_1^T \mathbf{R}_1 \mathbf{r}_1 - \mathbf{e}_1^T \mathbf{r}_2$ and $n_2 = \mathbf{e}_2^T \mathbf{R}_1 \mathbf{r}_1 - \mathbf{e}_2^T \mathbf{r}_2$; then bring (17) into (9) which is the expression of ω_2 , so it has a new type as (13).

$$\omega_2 = \frac{n_3}{l} \omega_1. \quad (18)$$

In (18), $n_3 = n_1 \mathbf{e}_2^T \mathbf{R}_1 S(1) \mathbf{r}_1 - n_2 \mathbf{e}_1^T \mathbf{R}_1 S(1) \mathbf{r}_1$. Based on (6), v can be written as

$$v = \frac{\omega_1 \mathbf{r}_1^T S(1) \mathbf{R}_1^T \mathbf{r}_2}{l}. \quad (19)$$

According to system's pose-attitude relation, the kinetic energy of system can be written as

$$T_1 = \frac{1}{2} J_1 \omega_1^2 + \frac{1}{2} (J_2 + J_3) \omega_2^2 + \frac{1}{2} m_3 v^2. \quad (20)$$

The potential energy of system can be written as

$$V_1 = m_1 g \mathbf{e}_2^T \mathbf{R}_1 \mathbf{r}_{m_1} + (m_2 + m_3) g \mathbf{e}_2^T \mathbf{r}_2 + \frac{(m_2 g l_{m_2} + m_3 g (l - l_{m_3}))}{l} n_2. \quad (21)$$

Bringing (18) and (19) into (20) and (21), energy formula based on \mathbf{R}_1 and ω_1 can be obtained. According to Lagrange theory, dynamics equation of conservation system can be derived out by

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \omega_1} \right) - \left(\frac{\partial L}{\partial \mathbf{R}_1} \right) = 0. \quad (22)$$

After expanding dynamics equation, the complete equation of dynamics system can be expressed as

$$\begin{aligned} \dot{\omega}_1 = & -n_{16} \omega_1^2 + \frac{J_a \omega_2}{n_j} (n_{12} - n_{10}) - m_3 \frac{\omega_1^2 n_{14} n_9}{l^2 n_j} \\ & - n_{15} n_{13} - \frac{m_3 \omega_1 n_{4a}}{l n_j} (n_5 - n_{11}) - \frac{m_3 g}{n_j} n_4 n_m \\ & - \frac{m_1 g}{n_j} n_{17}. \end{aligned} \quad (23)$$

In (23), the expressions of parameters are shown as follows.

$$\begin{aligned} l_a &= \mathbf{r}_1^T \mathbf{r}_1 + \mathbf{r}_2^T \mathbf{r}_2, \\ l_b &= \mathbf{r}_2^T \mathbf{R}_1 \mathbf{r}_1, \\ n_{1a} &= \mathbf{e}_1^T \mathbf{R}_1 \mathbf{r}_1, \\ l_b &= \mathbf{r}_2^T \mathbf{R}_1 \mathbf{r}_1, \\ n_{1a} &= \mathbf{e}_1^T \mathbf{R}_1 \mathbf{r}_1, \\ n_{2a} &= \mathbf{e}_2^T \mathbf{R}_1 \mathbf{r}_1, \\ n_{3a} &= \mathbf{e}_2^T \mathbf{R}_1 S(1) \mathbf{r}_1, \\ n_{3b} &= \mathbf{e}_1^T \mathbf{R}_1 S(1) \mathbf{r}_1 n_{4a} = \mathbf{r}_2^T \mathbf{R}_1 S(1) \mathbf{r}_1, \\ n_{5a} &= \mathbf{r}_1^T \mathbf{R}_1^T \mathbf{r}_2, \\ n_{17} &= \mathbf{e}_2^T \mathbf{R}_1 S(1) \mathbf{r}_{m_1} l = (l_a - 2l_b)^{1/2}, \\ n_4 &= -\frac{n_{4a}}{l}, \\ n_1 &= n_{1a} - \mathbf{e}_1^T \mathbf{r}_2, \\ n_2 &= n_{2a} - \mathbf{e}_2^T \mathbf{r}_2 n_3 = n_1 n_{3a} - n_2 n_{3b}, \\ n_m &= \frac{n_2}{l}, \\ n_5 &= \omega_1 \frac{n_{5a} l + n_4 n_{4a}}{l^2}, \\ J_a &= J_2 + J_3 \\ n_9 &= -\frac{n_{4a}}{l}, \\ n_8 &= n_{3b} n_{3a} - n_1 n_{2a} - n_{3a} n_{3b} + n_2 n_{1a}, \\ n_j &= \left(J_1 + J_a \frac{n_3^2}{l^2} \right) + m_3 \left(\frac{n_{4a}}{l} \right)^2, \\ n_{10} &= \frac{\omega_1}{l^2} (n_8 l - n_4 n_3) \\ n_{11} &= \frac{\omega_1}{l^2} (n_{5a} l + n_4 n_{4a}), \\ n_{12} &= \omega_1 \frac{n_8 l - n_4 n_3}{l^2}, \\ n_{13} &= \frac{(n_{3a} l - n_2 n_4)}{l^2}, \\ n_{14} &= n_{5a} l + n_4 n_{4a}, \\ n_{15} &= \frac{(m_2 g l_{m_2} + m_3 g (l - l_{m_3}))}{n_j}, \\ n_{16} &= \frac{J_a n_3 (n_8 l - n_4 n_3)}{l^3 n_j}. \end{aligned} \quad (24)$$

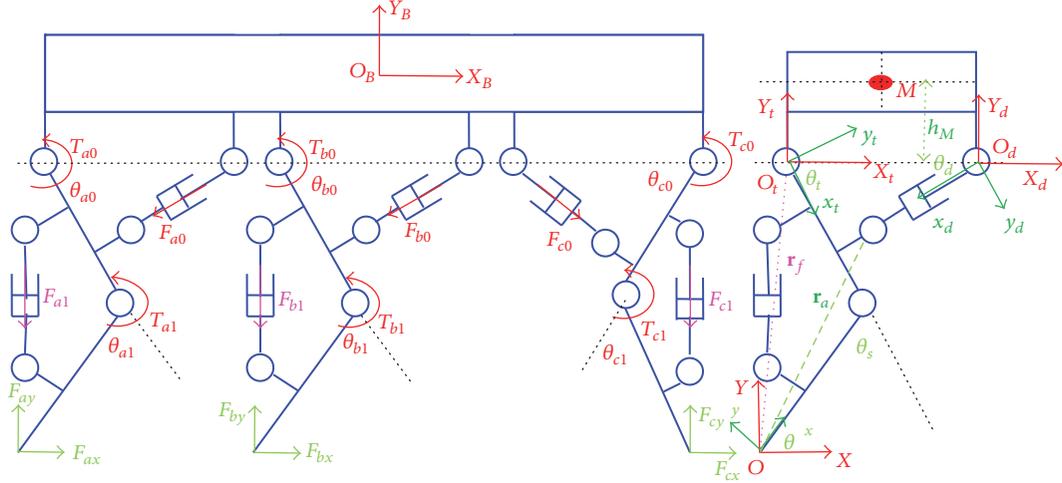


FIGURE 7: The mechanisms of PHR's whole machine and the equivalent type under straight line gait.

4.2. Dynamics Model of Body Moving. The straight line walking of PHR is realized by rotation of thigh joints, and shank joints keep still during motion process. According to Figure 7, PHR is supported by three legs, contact forces are on feet tips which are $\mathbf{F}_a = [F_{ax}; F_{ay}]$, $\mathbf{F}_b = [F_{bx}; F_{by}]$, $\mathbf{F}_c = [F_{cx}; F_{cy}]$, respectively. Thigh joints are driven by T_{a0}, T_{b0}, T_{c0} ; shank joints are driven by T_{a1}, T_{b1}, T_{c1} . The torque corresponds to drive forces of cylinders. When shank joints keep still, shank thigh and shank drive cylinder can be handled as one unit. So drive forces of shank have no influence on the walking. The six legs of PHR have same mechanism character, and body has no rotation during straight line walking, so rotation angles of thighs are equal, which means $\theta_{a0} = \theta_{b0} = \theta_{c0}$, $\theta_{a1} = \theta_{b1} = \theta_{c1} = C$ in Figure 7. Supposing that feet tips have no motion relative to ground, so the mechanism of the whole machine can be equivalent to the mechanism type as the right part in Figure 7. In this mechanism, displacement of body is equal to displacement of thigh joint in inertial frame, and dynamics character of whole machine can be expressed by one parameter.

Define m_b as mass of body; mass center is defined at M . For body has no rotation during straight line walking, so potential energy is only related to vertical distance between mass center and hinge joint; m_1 is gather mass of shank, thigh, and drive cylinder of shank. m_2, m_3 are masses of cylinder tube and rod, respectively. Define J_1 as inertia moment of leg unit which rotates along foot tip O ; J_2, J_3 are inertia moments of drive cylinder's tube and rod along O_d and O_a , respectively. Suppose \mathbf{r}_f is position vector of point O_t in body frame Oxy of leg unit, and position vector of O_a in frame Oxy is \mathbf{r}_a . So kinetic energy and potential energy of leg unit are written as

$$T_1 = \frac{1}{2} J_1 \dot{\theta}^2 \quad (25)$$

$$V_1 = m_1 g \mathbf{e}_2^T \mathbf{R}(\theta) \mathbf{r}_{m1}.$$

In formula (25), $\theta = \pi - \theta_t - \theta_s$, so $\dot{\theta} = -\dot{\theta}_t = -\omega_t$, ω_t is angular velocity of thigh, and \mathbf{r}_{m1} is position vector of mass center. The position of O_t in frame OXY is written as

$$\begin{aligned} \mathbf{r}_t &= \mathbf{R}(\theta) \mathbf{r}_f, \\ \mathbf{v}_t &= -\omega_t \mathbf{R}(\theta) \mathbf{s}_1 \mathbf{r}_f \\ \mathbf{r}_a &= \mathbf{R}(\theta) \mathbf{r}_a, \\ \mathbf{v}_a &= -\omega_t \mathbf{R}(\theta) \mathbf{s}_1 \mathbf{r}_a. \end{aligned} \quad (26)$$

So kinetic energy and potential energy of body is written as

$$\begin{aligned} T_2 &= \frac{1}{2} m_b \mathbf{v}_t^2 = \frac{1}{2} m_b \omega_t^2 (\mathbf{R}(\theta) \mathbf{s}_1 \mathbf{r}_f)^T (\mathbf{R}(\theta) \mathbf{s}_1 \mathbf{r}_f) \\ &= \frac{1}{2} m_b \omega_t^2 \|\mathbf{r}_f\|^2 \end{aligned} \quad (27)$$

$$V_2 = m_b g \mathbf{e}_2^T (\mathbf{R}(\theta) \mathbf{r}_f + h_M \mathbf{e}_2).$$

The motion of drive cylinder is analyzed as follows. The motion of drive cylinder consists of motion of cylinder tube and rod. As in Figure 7, the tube connects with thigh and rod connects with body, so the motion of tube is combination of displacement of point O_a and rotation along O_a . The motion of rod is combination of displacement of point O_d and rotation along O_d . So kinetic and potential energy of cylinder tube are written as (28), and the rod's are written as (29).

$$T_3 = \frac{1}{2} m_2 \mathbf{v}_a^2 + \frac{1}{2} J_2 \omega_d^2 = \frac{1}{2} (m_2 \|\mathbf{r}_a\|^2 \omega_t^2 + J_2 \omega_d^2) \quad (28)$$

$$V_3 = m_2 g \mathbf{e}_2^T (\mathbf{R}(\theta) \mathbf{r}_a + l_{m2} \mathbf{R}(\theta_d) \mathbf{e}_1)$$

$$T_4 = \frac{1}{2}m_3\mathbf{v}_t^2 + \frac{1}{2}J_3\omega_d^2 = \frac{1}{2}\left(m_2\|\mathbf{r}_f\|^2\omega_t^2 + J_2\omega_d^2\right) \quad (29)$$

$$V_4 = m_3g\mathbf{e}_2^T\left(\mathbf{R}(\theta)\mathbf{r}_f + \mathbf{r}_d + l_{m3}\mathbf{R}(\pi - \theta_d)\mathbf{e}_1\right).$$

For straight line walking is realized by three legs' motion, so Lagrange function of whole system is written as

$$\begin{aligned} L = T - V = & (3T_1 + T_2 + 3T_3 + 3T_4) - (3V_1 + V_2 \\ & + 3V_3 + 3V_4) = \frac{1}{2}\left(3J_1 + m_b\|\mathbf{r}_f\|^2 + 3m_2\|\mathbf{r}_a\|^2\right. \\ & + 3m_3\|\mathbf{r}_f\|^2\left.)\omega_t^2 + \frac{3}{2}(J_2 + J_3)\omega_d^2\right. \\ & - \left[3m_1g\mathbf{e}_2^T\mathbf{R}(\theta)\mathbf{r}_{m1} + m_bg\mathbf{e}_2^T\mathbf{R}(\theta)\mathbf{r}_f\right. \\ & + 3m_3g\mathbf{e}_2^T\mathbf{R}(\theta)\mathbf{r}_f + 3m_2g\mathbf{e}_2^T\mathbf{R}(\theta)\mathbf{r}_a] \\ & + 3m_3gl_{m3}\mathbf{e}_2^T\mathbf{R}^T(\theta_d)\mathbf{e}_1 - 3m_2gl_{m2}\mathbf{e}_2^T\mathbf{R}(\theta_d)\mathbf{e}_1 \\ & \left. - m_bgh_M - 3m_3g\mathbf{e}_2^T\mathbf{r}_d.\right. \end{aligned} \quad (30)$$

According to kinematic character of parallel pendulum, $\mathbf{R}(\theta_d)$ can be expressed by $\mathbf{R}(\theta_t)$, and ω_d can be expressed by ω_t and $\mathbf{R}(\theta_t)$. The concrete expressions are in

$$\begin{aligned} \mathbf{R}_d &= \frac{n_{d1}\mathbf{I} + n_{d2}S(1)}{l} \\ \omega_d &= \frac{n_{d3}}{l^2}\omega_t \\ n_{d1} &= -\mathbf{e}_1^T\mathbf{R}_t\mathbf{r}_1 + \mathbf{e}_1^T\mathbf{r}_2 \\ n_{d2} &= \mathbf{e}_2^T\mathbf{R}_t\mathbf{r}_1 - \mathbf{e}_2^T\mathbf{r}_2 \\ n_{d3} &= \\ &= \left(-\mathbf{e}_1^T\mathbf{R}_t\mathbf{r}_1\mathbf{e}_2 + \mathbf{e}_1^T\mathbf{r}_2\mathbf{e}_2 - \mathbf{e}_2^T\mathbf{R}_t\mathbf{r}_1\mathbf{e}_1 + \mathbf{e}_2^T\mathbf{r}_2\mathbf{e}_1\right)^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1 \\ &\mathbf{r}_1^T\mathbf{r}_1 - 2\mathbf{r}_2^T\mathbf{R}_t\mathbf{r}_1 + \mathbf{r}_2^T\mathbf{r}_2 = l^2. \end{aligned} \quad (31)$$

In (31) $\mathbf{R}_t = \mathbf{R}(\theta_t)$, $\mathbf{R}_d = \mathbf{R}(\theta_d)$; Lagrange function can be written as (32) with (31).

$$\begin{aligned} L = L(\omega_t, \mathbf{R}_t) &= \frac{1}{2}\left(3J_1 + m_b\|\mathbf{r}_f\|^2 + 3m_2\|\mathbf{r}_a\|^2 + 3m_3\|\mathbf{r}_f\|^2\right)\omega_t^2 \\ &+ \frac{3}{2}(J_2 + J_3)\left(\frac{n_{d3}}{l^2}\right)^2\omega_t^2 + (\mathbf{R}_{ts}\mathbf{e}_2)^T \\ &\cdot \left[3m_1g\mathbf{r}_{m1} + m_bg\mathbf{r}_f + 3m_3g\mathbf{r}_f + 3m_2g\mathbf{r}_a\right] \\ &- (3m_3gl_{m3} + 3m_2gl_{m2})\frac{n_{d2}}{l} - m_bgh_M \\ &- 3m_3g\mathbf{e}_2^T\mathbf{r}_d. \end{aligned} \quad (32)$$

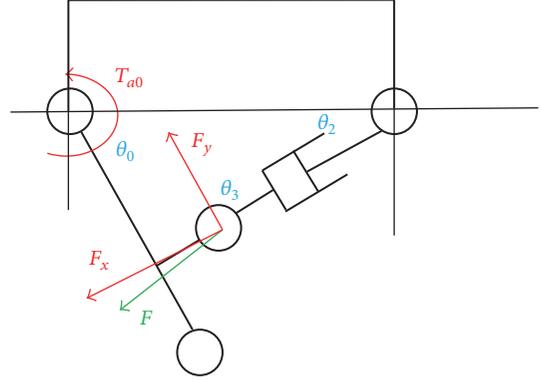


FIGURE 8: The force of the single joint.

In (32), $\mathbf{R}_{ts} = \mathbf{R}(\theta_t + \theta_s)$. According to Lagrange theory, the dynamics equation of system is obtained as

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \omega_t} - \frac{\partial L}{\partial \theta_t} &= k_1\dot{\omega}_t + k_2\omega_t^2 + k_3 \\ k_1 &= \left[\left(3J_1 + m_b\|\mathbf{r}_f\|^2 + 3m_2\|\mathbf{r}_a\|^2 + 3m_3\|\mathbf{r}_f\|^2\right)\right. \\ &\quad \left.+ 3(J_2 + J_3)\left(\frac{n_{d3}}{l^2}\right)^2\right] \\ k_2 &= \left[6(J_2 + J_3)\left(\frac{n_{d3}}{l^2}\right)\left(\frac{\partial n_{d3}}{\partial \theta_t}\frac{1}{l^2} + \frac{2\mathbf{r}_2^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1n_{d3}}{l^4}\right)\right] \\ &\quad - 3(J_2 + J_3)\left(\frac{n_{d3}}{l^2}\right)\left(\frac{\partial n_{d3}}{\partial \theta_t}\frac{1}{l^2} + \frac{2\mathbf{r}_2^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1n_{d3}}{l^4}\right) \\ k_3 &= (3m_3gl_{m3} + 3m_2gl_{m2})\left(\frac{\partial n_{d2}}{\partial \theta_t}\frac{1}{l} + \frac{\mathbf{r}_2^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1}{l^3}\right) \\ &\quad - (\mathbf{R}_{ts}\mathbf{s}_1\mathbf{e}_2)^T\left[3m_1g\mathbf{r}_{m1} + m_bg\mathbf{r}_f + 3m_3g\mathbf{r}_f\right. \\ &\quad \left.+ 3m_2g\mathbf{r}_a\right]. \end{aligned} \quad (33)$$

In (33) $\partial n_{d3}/\partial \theta_t = (-\mathbf{e}_1^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1\mathbf{e}_2 - \mathbf{e}_2^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1\mathbf{e}_1)^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1 - (\mathbf{e}_2^T\mathbf{r}_2\mathbf{e}_1 + \mathbf{e}_1^T\mathbf{r}_2\mathbf{e}_2 - \mathbf{e}_1^T\mathbf{R}_t\mathbf{r}_1\mathbf{e}_2 - \mathbf{e}_2^T\mathbf{R}_t\mathbf{r}_1\mathbf{e}_1)^T\mathbf{R}_t\mathbf{r}_1$, $\partial n_{d2}/\partial \theta_t = \mathbf{e}_2^T\mathbf{R}_t\mathbf{s}_1\mathbf{r}_1 - \mathbf{e}_2^T\mathbf{r}_2$, $\mathbf{r}_1^T\mathbf{r}_1 - 2\mathbf{r}_2^T\mathbf{R}_t\mathbf{r}_1 + \mathbf{r}_2^T\mathbf{r}_2 = l^2$. So (33) is dynamics equation of PHR under straight line walking.

4.3. The Force Analysis. The dynamics models of leg swing and straight line walking are built in Sections 3.1 and 3.2, respectively. According to dynamics models, rotation angles of joints are chosen as generalized coordinates. Actually, the joints are driven by cylinders, so the relation between joint drive torques and cylinder drive forces should be constructed.

In order to construct the relation between joint torque and cylinder drive force, the parallel pendulum structure of thigh joint is used as example in Figure 8. Supposing the cylinder's push force is F , the angle between push rod and thigh is θ_3 . According to triangle character $\theta_3 = \pi - \theta_0 - \theta_2$, two

components of cylinder drive force on orthogonal directions of thigh are

$$\begin{aligned} F_x &= \sin(\theta_0 + \theta_2) F = (\sin \theta_0 \cos \theta_2 + \cos \theta_0 \sin \theta_2) F \\ F_y &= -\cos(\theta_0 + \theta_2) F \\ &= -(\cos \theta_0 \cos \theta_2 - \sin \theta_0 \sin \theta_2) F. \end{aligned} \quad (34)$$

So the torque on joint is

$$\begin{aligned} T &= F_x r_x + F_y r_y = F [(\sin \theta_0 \cos \theta_2 + \cos \theta_0 \sin \theta_2) r_x \\ &\quad - (\cos \theta_0 \cos \theta_2 - \sin \theta_0 \sin \theta_2) r_y]. \end{aligned} \quad (35)$$

According to rotation relation, suppose that \mathbf{R}_0 and \mathbf{R}_2 are

$$\begin{aligned} \mathbf{R}_0 &= \begin{bmatrix} c(-\theta_0) & -s(-\theta_0) \\ s(-\theta_0) & c(-\theta_0) \end{bmatrix} = \begin{bmatrix} c\theta_0 & s\theta_0 \\ -s\theta_0 & c\theta_0 \end{bmatrix}, \\ \mathbf{R}_2 &= \begin{bmatrix} c[-(\pi - \theta_2)] & -s[-(\pi - \theta_2)] \\ s[-(\pi - \theta_2)] & c[-(\pi - \theta_2)] \end{bmatrix} \\ &= \begin{bmatrix} -c\theta_2 & s\theta_2 \\ -s\theta_2 & -c\theta_2 \end{bmatrix}. \end{aligned} \quad (36)$$

So the projections of hinge joint on thigh and drive force vector are as in

$$\begin{aligned} \mathbf{R}_0 \mathbf{r}_1 &= \begin{bmatrix} c\theta_0 & s\theta_0 \\ -s\theta_0 & c\theta_0 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} r_x c\theta_0 + s\theta_0 r_y \\ -r_x s\theta_0 + r_y c\theta_0 \end{bmatrix} \\ F \mathbf{R}_2 \mathbf{e}_1 &= l \begin{bmatrix} -c\theta_2 & s\theta_2 \\ -s\theta_2 & -c\theta_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = F \begin{bmatrix} -c\theta_2 \\ -s\theta_2 \end{bmatrix}. \end{aligned} \quad (37)$$

The cross product of (37) is

$$\begin{aligned} T &= F \begin{bmatrix} r_x c\theta_0 + s\theta_0 r_y \\ -r_x s\theta_0 + r_y c\theta_0 \end{bmatrix}^T \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -c\theta_2 \\ -s\theta_2 \end{bmatrix} \\ &= F [r_x (c\theta_0 s\theta_2 + s\theta_0 c\theta_2) - r_y (c\theta_0 c\theta_2 - s\theta_0 s\theta_2)]. \end{aligned} \quad (38)$$

According to analysis on geometry, the joint torque is cross product of leg's hinge point's position vector and cylinder direction vector, which can be expressed as matrix type as in

$$\tau = (\mathbf{R}_0 \mathbf{r}_1)^T \mathbf{s}_1 (F \mathbf{R}_2 \mathbf{e}_1). \quad (39)$$

According to the relation between \mathbf{R}_2 and \mathbf{R}_0 , (39) can be written as

$$\begin{aligned} \tau &= \frac{F}{l} (\mathbf{R}_0 \mathbf{r}_1)^T \mathbf{s}_1 (n_1 \mathbf{e}_1 + n_2 \mathbf{s}_1 \mathbf{e}_1) \\ &= \frac{F}{l} (\mathbf{R}_0 \mathbf{r}_1)^T (n_1 \mathbf{e}_2 - n_2 \mathbf{e}_1). \end{aligned} \quad (40)$$

Furthermore, the drive force is generated by gas, and the relation is $F = \pi d^2 p / 4$, d is cylinder bore (mm), and p is gas pressure (MPa). When cylinder is confirmed, the gas consumption is only influenced by pressure p .

5. The Optimal Control with Pseudospectral Method

The main character of pseudospectral method is that the state and control variables of ordinary differential equations are discrete on Legendre-Gauss points. The discrete points are used as nodes to construct Lagrange interpolating polynomial which is used to approximate state variables and control variables. The derivatives of state variables are approximated by differentiating the overall interpolating polynomial so that differential equation constraints are changed to be algebra constraints. The integral part of performance index is calculated by Gauss integral. From above transformations, the optimal control problem is translated to be a nonlinear programming problem with a series of algebra constraints.

5.1. The Problem Description. According to (23) and (40), state equation of parallel pendulum which represents leg swing is written as (41). According to (23) and (33), state equation of robot during straight line walking is as in (42).

$$\begin{aligned} \dot{\theta}_1 &= \omega_1 \\ \dot{\omega}_1 &= -n_{16} \omega_1^2 + \frac{J_a \omega_2}{n_j} (n_{12} - n_{10}) - m_3 \frac{n_{14} n_9}{l^2 n_j} \omega_1^2 \\ &\quad - n_{15} n_{13} - \frac{m_3 \omega_1 n_{4a}}{l n_j} (n_5 - n_{11}) - \frac{m_3 g}{n_j} n_4 n_m \\ &\quad - \frac{m_1 g}{n_j} n_{17} + \frac{\pi d^2 p}{4 l n_j} (\mathbf{R}_1 \mathbf{r}_1)^T (n_1 \mathbf{e}_2 - n_2 \mathbf{e}_1) \end{aligned} \quad (41)$$

$$\begin{aligned} \dot{\theta}_t &= \omega_t \\ \dot{\omega}_t &= \frac{F}{k_1 l} (\mathbf{R}_t \mathbf{r}_1)^T (n_{d1} \mathbf{e}_2 - n_{d2} \mathbf{e}_1) - \frac{k_2}{k_1} \omega_t^2 - \frac{k_3}{k_1}. \end{aligned} \quad (42)$$

The common optimal control problem can be described as searching the control variable $\mathbf{u}(t)$ which satisfies minimum objective function. In (41) and (42), state variables are $\mathbf{x}(t) = [\theta_1; \omega_1]$ and $\mathbf{x}(t) = [\theta_t; \omega_t]$, respectively; control input is gas pressure p , so $\mathbf{u}(t) = [0; p]$. The minimum objective function is

$$J = \Phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t)) dt. \quad (43)$$

In (43), state variable $\mathbf{x}(t)$, initial time t_0 , and end time t_f satisfy dynamics equation as (44), which represents ordinary differential equations as (41) and (42).

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (44)$$

The boundary conditions are $\phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0$. In this exploration, boundary conditions include initial, terminal, and boundary values of rotation angle θ and angular velocities ω . The control constraint is written as $\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0$. In this exploration, the constraint is variation boundaries of input gas pressure.

5.2. *The Time Domain Transformation.* Before using Gauss-pseudospectral method, the time interval of optimal control should be transformed from $t \in [t_0, t_f]$ to $\tau \in [-1, 1]$ first. The process is shown as

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0}. \quad (45)$$

The transformation process for minimum performance index is written as

$$J = \Phi(\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^1 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau. \quad (46)$$

The dynamics differential equation constraints can be transformed to be

$$\dot{\mathbf{x}}(\tau) = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau), \quad \tau \in [-1, 1]. \quad (47)$$

The boundary condition: $\phi(\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f) = 0$.

The path constraints: $\mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \leq 0$.

5.3. *The State and Control Variables Approximated by the Overall Interpolating Polynomial.* Gauss-pseudospectral method uses n Legendre-Gauss points and $\tau_0 = -1$ as nodes, which forms $n+1$ Lagrange interpolating polynomials $L_i(\tau)$, $i = 0, \dots, n$ as primary function to approximate the state variables, as in

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^n L_i(\tau) \mathbf{x}(\tau_i). \quad (48)$$

In (48), base function of Lagrange interpolating polynomials can be expressed as (49), which makes approximate state on nodes equal to virtual conditions, as $\mathbf{x}(\tau_i) \approx \mathbf{X}(\tau_i)$, $i = 0, \dots, n$,

$$L_i(\tau) = \prod_{j=0, j \neq i}^n \frac{\tau - \tau_j}{\tau_i - \tau_j}. \quad (49)$$

The Lagrange interpolating polynomials are used as basis function for approximate control variables as

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^n L_i(\tau) \mathbf{U}(\tau_i). \quad (50)$$

In above equations, τ_i , $i = 1, \dots, n$ are Legendre-Gauss points.

5.4. *The Transformation of Differential Constraints to Algebra Constraints.* Differentiating state variable, dynamics differential equation constraints can be transformed to be algebra constraints as

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=0}^n \dot{L}_i(\tau_k) \mathbf{x}(\tau_i) = \sum_{i=0}^n \mathbf{D}_{ki}(\tau_k) \mathbf{x}(\tau_i). \quad (51)$$

The expression of differential matrix is written as

$$\mathbf{D}_{ki} = \dot{L}_i(\tau_k) = \begin{cases} \frac{(1 + \tau_k) \dot{P}_n(\tau_k) + P_n(\tau_k)}{(\tau_k - \tau_i) [(1 + \tau_i) \dot{P}_n(\tau_i) + P_n(\tau_i)]}, & i \neq k \\ \frac{(1 + \tau_i) \ddot{P}_n(\tau_i) + 2\dot{P}_n(\tau_i)}{2 [(1 + \tau_i) \dot{P}_n(\tau_i) + P_n(\tau_i)]}, & i = k. \end{cases} \quad (52)$$

In (52), $k = 1, \dots, n$, $i = 0, \dots, n$. From the above transformations, dynamics differential constraints are translated to be algebra constraints.

$$\sum_{i=0}^n \mathbf{D}_{ki}(\tau_k) \mathbf{X}(\tau_i) - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k, t_0, t_f) = 0. \quad (53)$$

5.5. *The Terminal State Constraints under Discrete Condition.* For nodes of Gauss-pseudospectral method excludes end point $\tau_f = 1$, so the terminal state \mathbf{X}_f is not definite in dynamics differential equation constraints. The terminal state should satisfy dynamics constraints as

$$\mathbf{x}(\tau_f) = \mathbf{x}(\tau_0) + \int_{-1}^1 \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau. \quad (54)$$

The terminal constraints are discretized and approximated by the Gauss integral method, which can be written as (55); terminal constraint is written as

$$\mathbf{X}(\tau_f) = \mathbf{X}(\tau_0) + \frac{t_f - t_0}{2} \sum_{k=1}^n w_k \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau, t_0, t_f). \quad (55)$$

In (55), $w_k = \int_{-1}^1 L_i(\tau) d\tau$ is Gauss weight; τ_k is Legendre-Gauss points.

5.6. *The Performance Index under Discrete Condition.* Integral parts of performance index are approximated by Gauss integral, and performance index with pseudospectral type can be obtained as

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^n w_k g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f). \quad (56)$$

Therefore, the continuous optimal control problem is transformed to be a nonlinear programming problem with discrete work of pseudospectral method. Then discrete control and state variables can obtain a minimum performance index which satisfies state constraints, terminal constraints, boundary conditions $\phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = 0$, and path constraints $\mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \leq 0$.

5.7. The Optimal Control of PHR. The gas consumption optimal control of PHR under straight line walking has two parts. The first part is gas consumption optimal control of leg swing when foot has no contact with ground, which corresponds to state equation (41). The second part is gas consumption optimal control of body moving by support of legs in which feet contact with ground. This part corresponds to state equation (42). The sum of these optimal control results is gas consumption of one step. For the goal of optimal control is minimum of gas consumption, so objective function is $J = \int_{t_0}^{t_f} p dt$. The optimal control of PHR can be expressed as searching control input p to make system move from initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$ to terminal condition $\mathbf{x}(t_f) = \mathbf{x}_f$ under minimum energy consumption and satisfy a certain of constraints in a given time interval. The process can be written as follows.

The Functional Extreme Value Problem of Optimal Control

The performance index: $J = \int_{t_0}^{t_f} p^2 dt$.

The constraints of initial value: $\mathbf{x}(t_0) = \mathbf{x}_0$.

The state equation: $\dot{\mathbf{x}} = \mathbf{f}[\mathbf{x}, \mathbf{u}, t]$.

The constraints of control: $\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$.

The constraints of states: $\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}$.

The boundary conditions: $t \leq t_f, \mathbf{x}(t_f) \leq \mathbf{x}_f$.

6. The Kinematics Analysis

The kinematic process of PHR's straight line walking is analyzed in this part. The structure parameters of PHR are as follows. The parameters are obtained from the 3D model of PHR as Figure 2. In order to verify the correctness of mathematical model, the 3D model is kinematic, simulated by ADAMS, and the simulation results are used as criterions for the correctness of numerical results of mathematical model. The numerical solution path is designed as follows.

The Numerical Solution Path

The initialization of the variables: $l = l_0, v = v_0, a = a_0$.

For loop:

solve the following formulas as sequence: (3), (4), (6), (8), (9), and (10);

calculate the following parameters: $\mathbf{R}_1, \mathbf{R}_2, \omega_1, \omega_2, \dot{\omega}_1, \dot{\omega}_2$;

the initial value update is as follows: $l = l(t), v = v(t), a = a(t)$;

End.

The structure parameters of thigh are $\mathbf{r}_1 = [111.5; 58.25]$, $\mathbf{r}_2 = [272.5; 31.75]$; the length of thigh is 250 mm. The structure parameters of shank are $\mathbf{r}_1 = [155.5; -58.25]$, $\mathbf{r}_2 = [-197.5; -58.25]$ and $\mathbf{r}_f = [463; 216.5]$; the length of shank is 338 mm. The initial length of thigh drive cylinder

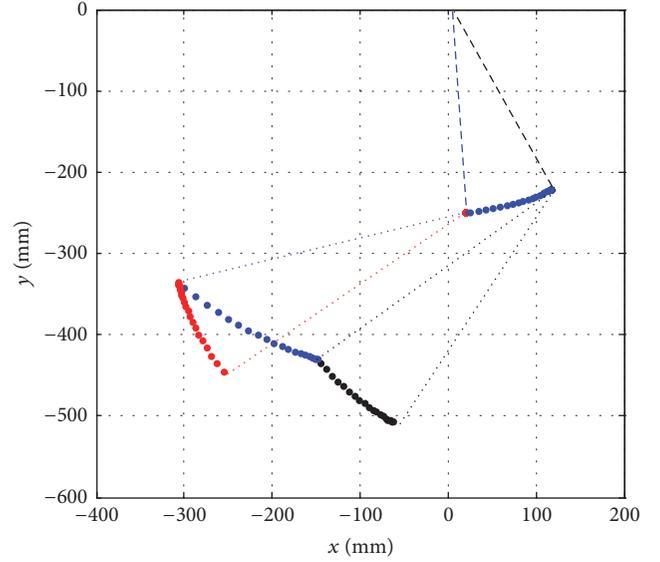


FIGURE 9: The pose-attitude variation of leg.

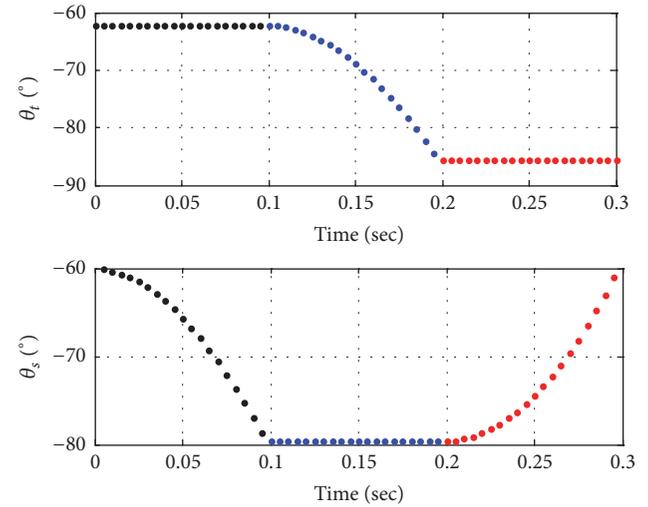


FIGURE 10: The joints' rotation angle variation.

is 198 mm, and shank's is 248 mm. The cylinder's stroke is 50 mm. For the pneumatic experiment has not proceeded, the acceleration of cylinder motion is supposed as 10000 mm/s^2 in simulation. For magnetic valve's minimum action time is 0.1s, so the action time of cylinder is supposed to be 0.1s. According to the above parameters and motion relations, the variation curves of leg's kinematics parameters are as in Figures 9–15.

The track of foot tip and shank joint is expressed in Figure 9. The black, blue, and red curves are the tracks of foot swing process. The coordinates of initial and terminal points are $[-65; 506]$ and $[-255.8; -445.5]$, respectively. These two coordinates indicate that the forward distance of one step is 190 mm, and the difference of coordinates on y direction indicates that body has y direction motion during foot transformation which is 60 mm.

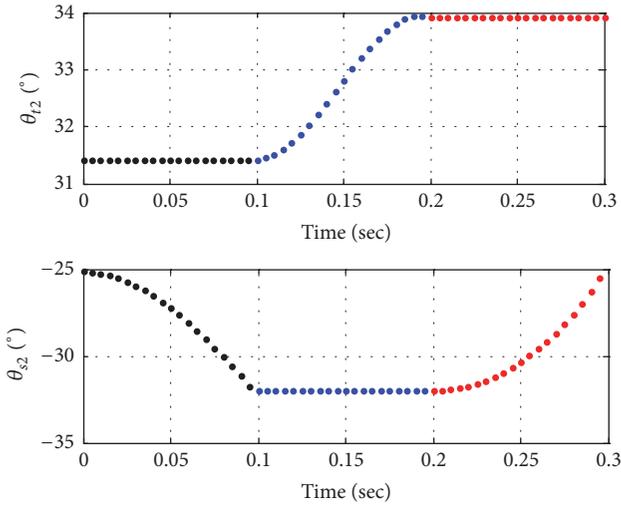


FIGURE 11: Rotation angle of drive cylinder.

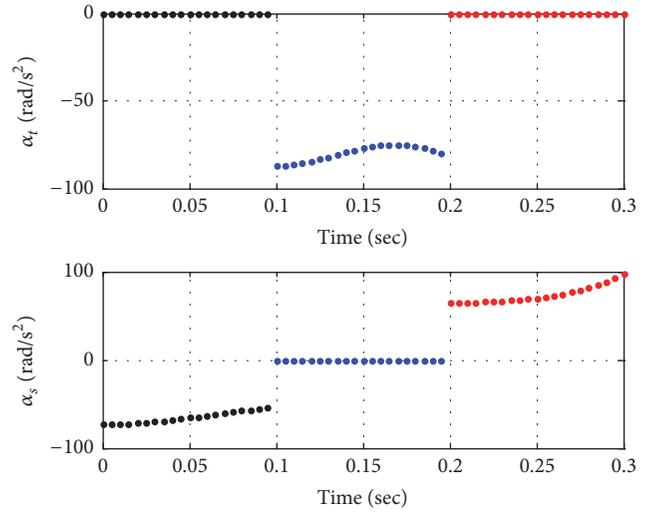


FIGURE 14: Angular acceleration of joints.

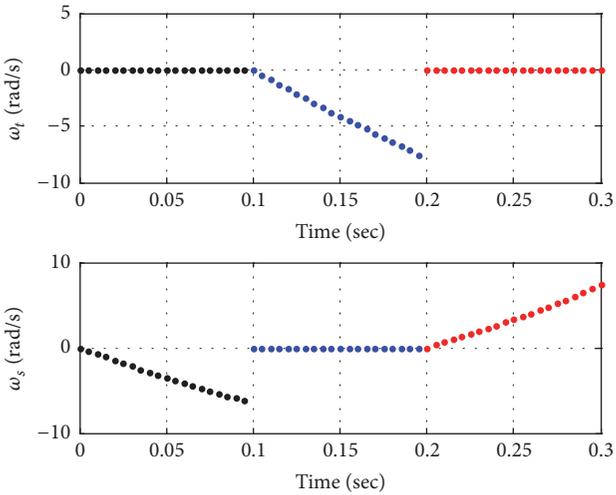


FIGURE 12: Angular velocity of thigh and shank joints.

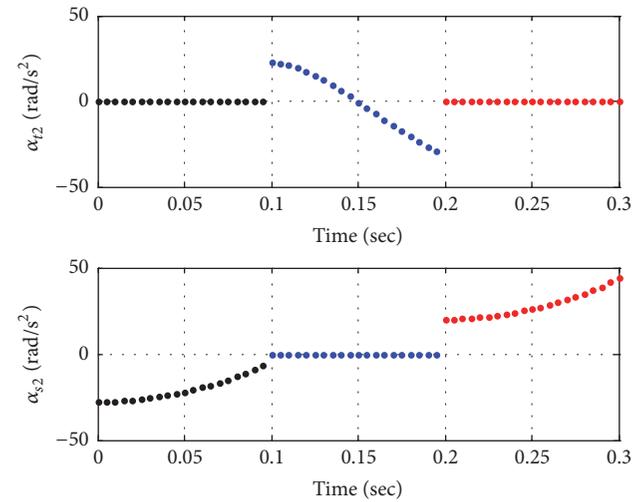


FIGURE 15: Angular acceleration of drive cylinders.

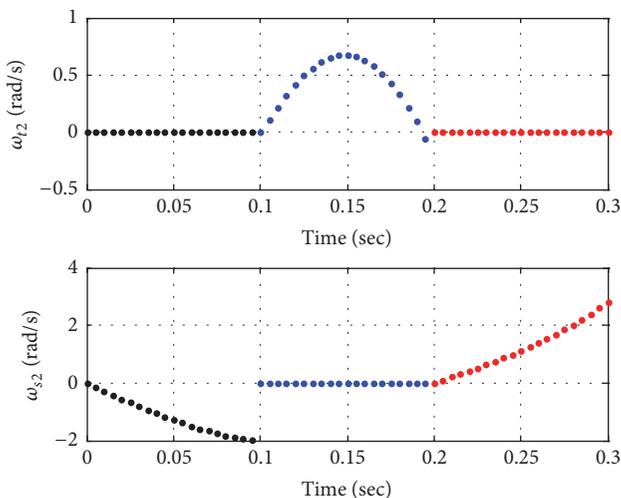


FIGURE 13: Angular velocity of drive cylinder.

The rotation angles variation curves of thigh and shank are expressed in Figure 10. According to Figure 10, the initial and terminal angles of thigh joint are -62.2° and -85.56° , the variation range of thigh joint is 23.36° , and the three key values of shank joint are -60° , -79.64° , and 19.64° , respectively. The curves are smooth and have parabola character, which means that the rotation of thigh and shank joints have stable acceleration, which is identical to the motion character of drive cylinders.

The rotation angles variation curves of drive cylinders of thigh and shank are expressed in Figure 11. According to Figure 11, the initial and terminal rotation angles of thigh joint's drive cylinder are 31.42° and 33.92° , the variation range of it is 2.5° , and the three key values of shank joint's drive cylinder are -25.15° , -32.03° , and 6.87° , respectively. The variation ranges are far smaller than main joints.

The angular velocities variation curves of thigh and shank joints are expressed in Figure 12. The variation range of

thigh joint's angular velocity is 7.552 rad/s and shank joint's variation ranges are 6.898 rad/s and 6.192 rad/s on different rotation directions. The curves are smooth and variation tendencies are similar to a straight line, which means the gradients of angular velocity are constant.

The angular velocities variation curves of drive cylinders are expressed in Figure 13. According to Figure 13, the variation curve of thigh drive cylinder's angular velocity has a parabola character, and the maximum value is 0.6729 rad/s; the variation ranges of shank drive cylinder's angular velocities are 1.958 and 2.063 rad/s. The values and variation ranges of drive cylinders angular velocities are far smaller than joints expressed in Figure 12.

The angular acceleration variation curves of thigh and shank joints are expressed in Figure 14. According to Figure 14, the variation range of thigh joint's angular acceleration is between -86.99 rad/s^2 and -80.26 rad/s^2 , which has a small variation range. The variation ranges of shank joint drive cylinder's angular acceleration on two different motion directions which are $[-73.17, -54.33] \text{ rad/s}^2$ and $[64.88, 92.57] \text{ rad/s}^2$, respectively. The variation curves are smooth and continuous, which means the motion of thigh and shank joints is second-order continuous.

The angular acceleration of drive cylinders is shown in Figure 15. The variation range of thigh joint drive cylinder's angular acceleration is $[22.45, -29.38] \text{ rad/s}^2$, and shank joint's is $[-27.79, 0] \text{ rad/s}^2$ and $[20.22, 41.47] \text{ rad/s}^2$ on two directions, respectively. This means that although rotation angles and angular velocities of the drive cylinders are small, the variation processes are rapid.

The displacement, velocities, and acceleration variation curves of body which moves by support of legs are expressed in Figures 16, 17, and 18, respectively. According to Figure 16, the initial and terminal positions of thigh joint relative to foot tip are $[258.5, 440.9] \text{ mm}$ and $[52.4, 508.4] \text{ mm}$, respectively. The variation curve is a smooth arc. According to two coordinates, the forward displacement of one step is 206.1 mm, and the displacement on vertical direction is 675 mm. According to Figure 17, the velocity's maximum value on x direction is 4.48 m/s, and on y direction is -0.461 m/s ; the velocity on x direction is 10 times the velocity on y direction. According to Figure 18, the maximum values of acceleration on x and y directions are 72.98 m/s^2 and 32.37 m/s^2 , respectively, and maximum values are occurring at the end of motion process.

According to kinematics analysis, the conclusions can be summarized as follows.

- (1) The variation curves of angles, angular velocities, and angular acceleration are all smooth, which means the straight line walking process has second-order smooth character.
- (2) The rotation angles, velocities, and acceleration values of thigh and shank joints are much larger than that of drive cylinders. This means that the mechanism of leg can be improved to be the type where drive cylinders have no rotation during motion process in the future.
- (3) The maximum values of angular velocities and angular acceleration appear at the end of motion, so in

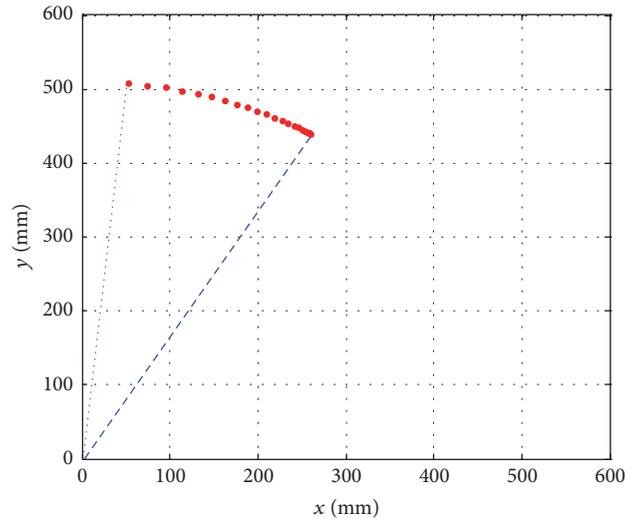


FIGURE 16: The track of thigh joint with body moving.

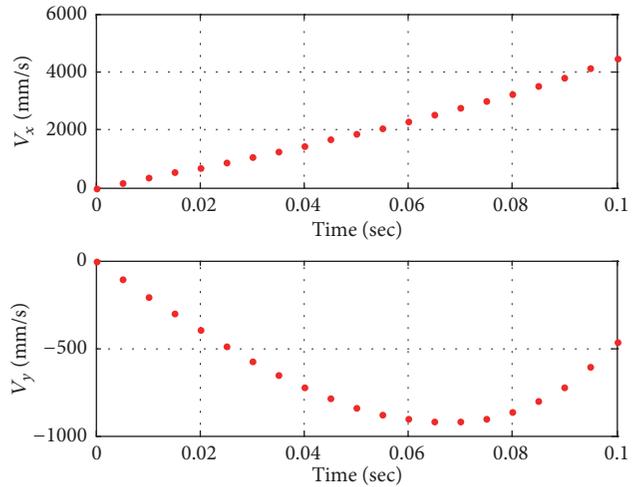


FIGURE 17: The velocity of mass center.

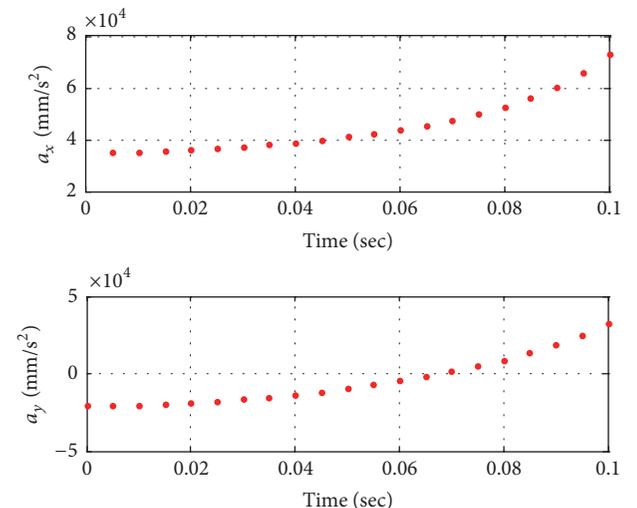


FIGURE 18: The acceleration of mass center.

the actual engineering design, the accelerate motion at the end of cylinders stroke should be avoided.

- (4) The straight line walking distance of one step is 0.2 m, and the process needs 4 cylinder strokes. The gas consumption volumes of each stroke are product of cylinder inner area and stroke distance. The front 3 strokes only need to drive a small mass, so they only need a low pressure, the 4th stroke needs to drive the mass of the whole machine, so it needs a high pressure. The pressure values of each stroke are calculated in the following part with optimal control method.

7. The Optimal Control Analysis

In order to analyze dynamics and control of PHR, the pseudospectral optimal control method is used to solve dynamics equations of straight line walking as (36) and (37). With this calculation, the variation curves of motion parameters as rotation angles, angular velocities, and angular acceleration of straight line will be obtained, and control input pressures curves of each action will also be obtained. With these curves, the dynamics and control of PHR can be synthetically analyzed.

According to straight line gait, the first step is uplift process of shank. The mass and inertia of shank are $m_s = 0.8154$ kg and $J_s = 0.0492$ kg·m² and the mass center's position vector is $\mathbf{r}_{ms} = [0.223; 0]$ m. According to the kinematics analysis results, the initial length of cylinder is 248 mm, so initial rotation angle of shank joint is 60°, and terminal length of cylinder is 198 mm, so terminal angle of shank joint is 79.64°; initial and terminal values of (36) are $\theta_1^{\text{initial}} = \pi/3$, $\omega_1^{\text{initial}} = 0$; $\theta_1^{\text{end}} = 0.44\pi$, $\omega_1^{\text{end}} = 0$; variation ranges of θ_1 and ω_1 are $\pi/3 \leq \theta_1 \leq 0.44\pi$ and $-10 \leq \omega_1 \leq 10$, respectively. The control input pressure range is $0.15 \text{ MPa} \leq p \leq 0.8 \text{ MPa}$ for minimum and maximum valid pressures of magnetic valve are 0.15 MPa and 0.8 MPa.

The second step is rotation of thigh joint. In this step shank joint keeps still, so thigh and shank can seem as a whole. The mass and inertia of this whole part are $m_{ts} = 1.576$ kg and $J_{ts} = 0.1276$ kg·m² and the mass center's position vector is $\mathbf{r}_{ms} = [0.2; 0.133]$ m. The initial length of cylinder is 198 mm, so initial rotation angle of shank joint is 62.2°, and terminal length of cylinder is 248 mm, so terminal angle of shank joint is 85.56°; the initial and terminal values of (36) are $\theta_1^{\text{initial}} = 0.346\pi$, $\omega_1^{\text{initial}} = 0$; $\theta_1^{\text{end}} = 0.475\pi$, $\omega_1^{\text{end}} = 0$; the variation ranges of θ_1 and ω_1 are $0.34\pi \leq \theta_1 \leq 0.48\pi$ and $-10 \leq \omega_1 \leq 10$, respectively. Control input pressure range is as the first step.

The third step is rotation of shank joint. The parameters of this step are identical to the first step, the initial and terminal values of (36) are $\theta_1^{\text{initial}} = 0.44\pi$, $\omega_1^{\text{initial}} = 0$; $\theta_1^{\text{end}} = \pi/3$, $\omega_1^{\text{end}} = 0$.

The fourth step is body moving with rotation of thigh joint. In this step, foot tip contacts ground and shank keeps still. The mass of body is $m_b = 30$ kg, the mass of leg is $m_l = 1.576$ kg, the moment of inertia along the foot tip is

$J_1 = 0.1247$ kg·m², the mass center position is $\mathbf{r}_{m1} = [0.227; 0.059]$ m, the initial and terminal values of (37) are $\theta_t^{\text{initial}} = 0.475\pi$, $\omega_t^{\text{initial}} = 0$; $\theta_t^{\text{end}} = 0.346\pi$, $\omega_t^{\text{end}} = 0$, and variation ranges of θ_t and ω_t are $0.34\pi \leq \theta_t \leq 0.48\pi$ and $-10 \leq \omega_t \leq 10$, respectively.

The drive cylinder two parts' mass, moment of inertia, and distance of mass center to hinge joint are as follows. $m_1 = 0.257$ kg, $J_1 = 5.3 \times 10^{-4}$ kgm², $l_{m1} = 0.069$ m; $m_2 = 0.043$ kg, $J_2 = 2.15 \times 10^{-4}$ kgm², $l_{m2} = 0.058$ m.

The 4 steps of one straight line walking gait are solved by pseudospectral method, and simulation results are expressed in Figures 19–22. According to Figure 19, the uplift process of shank only needs a low pressure as 0.15 MPa, but the control time only needs 0.08 s, as the left graph of Figure 9. If the control time is 0.1 s, the control input pressure only needs 0.1 MPa. On point of engineering, the magnetic valve will not act if the pressure is lower than 0.15 MPa, so the first optimal control result as in Figure 19 meets the need of engineering application. However, the lowest action time of magnetic valve is 0.1 s, so impact may occur during the experiment. According to Figure 20, thigh joint has a stable motion process when input pressure is 0.15 MPa and control time is 0.1 s.

The optimal control results of third step are shown in Figure 21, which are similar to results as in Figure 19, and control input pressure is also 0.15 MPa. The optimal control results of fourth step are shown in Figure 22. When the input pressure is 0.6 MPa and control time is 0.1 s, variation of rotation angle is not smooth, and when control input pressure is 0.5 MPa, and control time is 0.3 s, the variation of rotation angle and angular velocity are smooth, so control input is 0.5 MPa with control time 0.3 s being the best choice.

According to the above optimal control calculation results, the conclusions can be summarized as follows.

- (1) The leg swing only needs a low gas pressure as 0.15 MPa, and body moves by support of legs need pressure of 0.5 MPa with mass of whole machine being 30 kg.
- (2) The optimal control results indicate that it needs at least two gas pressure stages to be designed in practical engineering design.
- (3) The gas consumption of one gait is $0.15 \text{ MPa} \times (\pi/4)d^2 \times 3 \times 3 + 0.5 \text{ MPa} \times (\pi/4)d^2 = 1.85 \text{ MPa} \times (\pi/4)d^2 = 0.04 \text{ MPa} \cdot \text{L}$ with mass of whole machine being 30 kg and inner diameter of cylinder being 32 mm. In this exploration, the volume and gas pressure of high pressure bottle are 8 L \times 15 MPa, so the carried high pressure gas can support the PHR to walk 3000 steps straightly. According to kinematic analysis, the forward distance of one step is 0.2 m, so the cruising ability of PHR is 600 m.

8. Conclusion

In this paper, kinematic, dynamics, and optimal control problem of PHR are explored, and cruising ability of designed

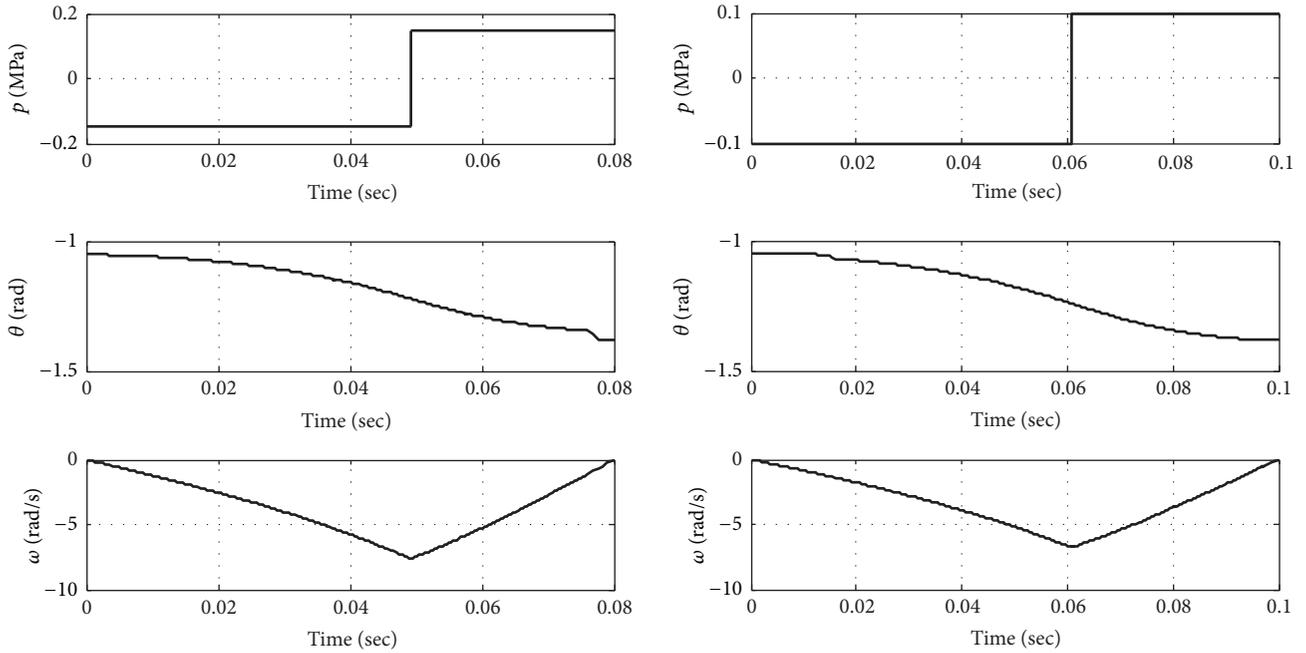


FIGURE 19: The optimal control results of shank joint on the first step.

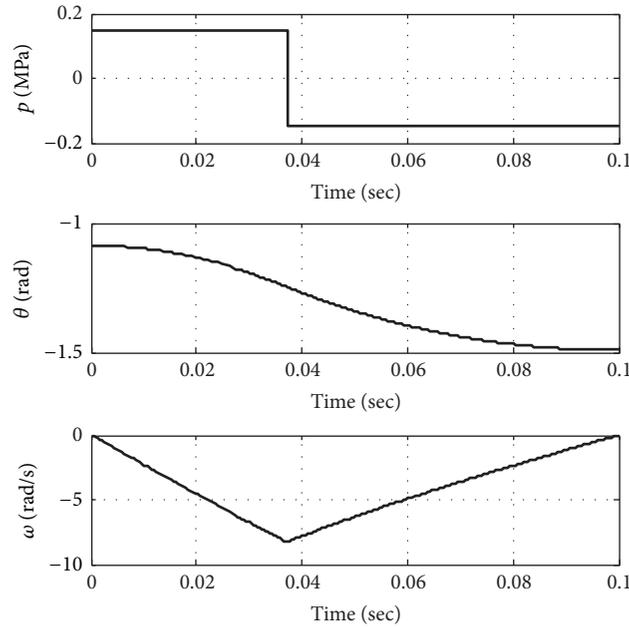


FIGURE 20: The optimal control results of thigh joint on the second step.

PHR is analyzed. According to this exploration, the conclusions can be summarized as follows.

- (1) Matrix and vector operator are a good modeling method that can replace triangle method, which makes kinematic and dynamic modeling of complex

parallel mechanism easier. In this exploration, kinematic and dynamic model of PHR are built by matrix and vector operators which are successfully solved.

- (2) Pseudospectral method is convenient to solve optimal control problems of nonlinear dynamics systems

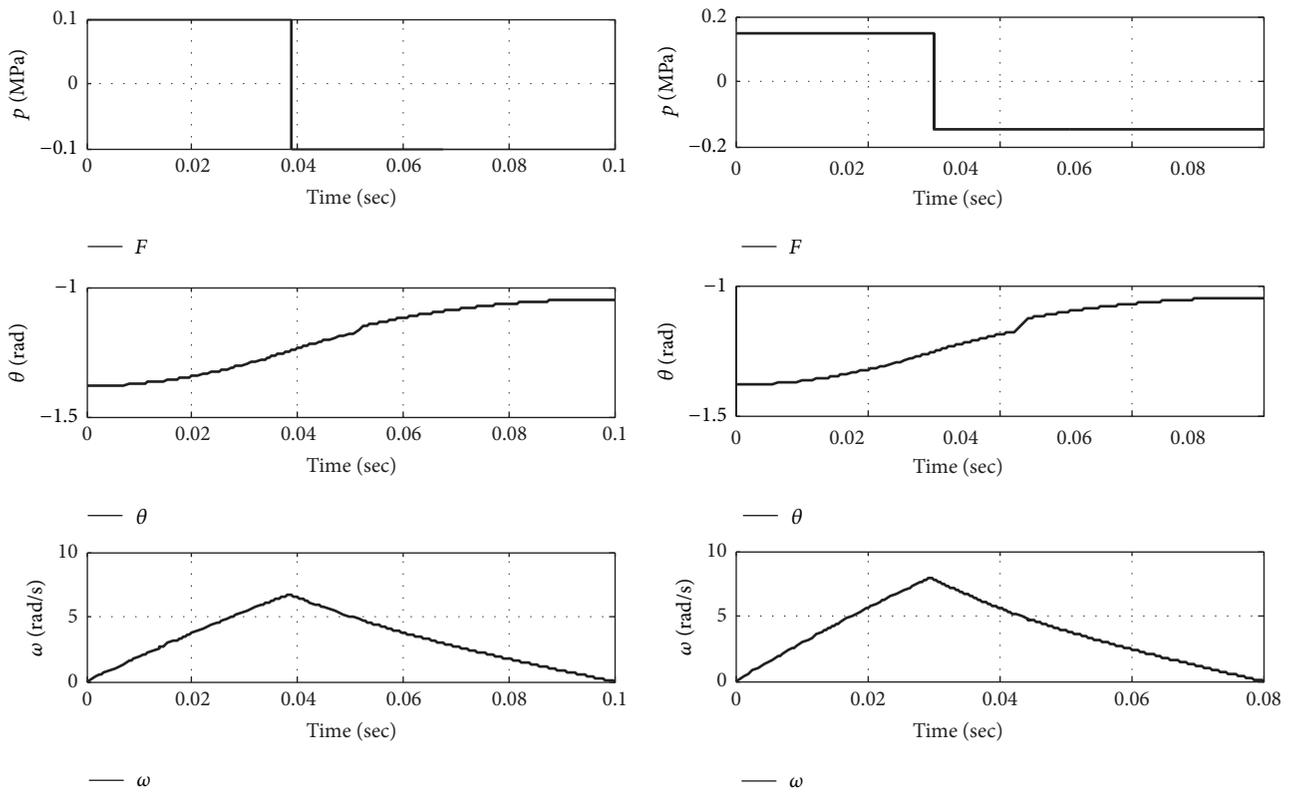


FIGURE 21: The optimal control results of shank joint on the third step.

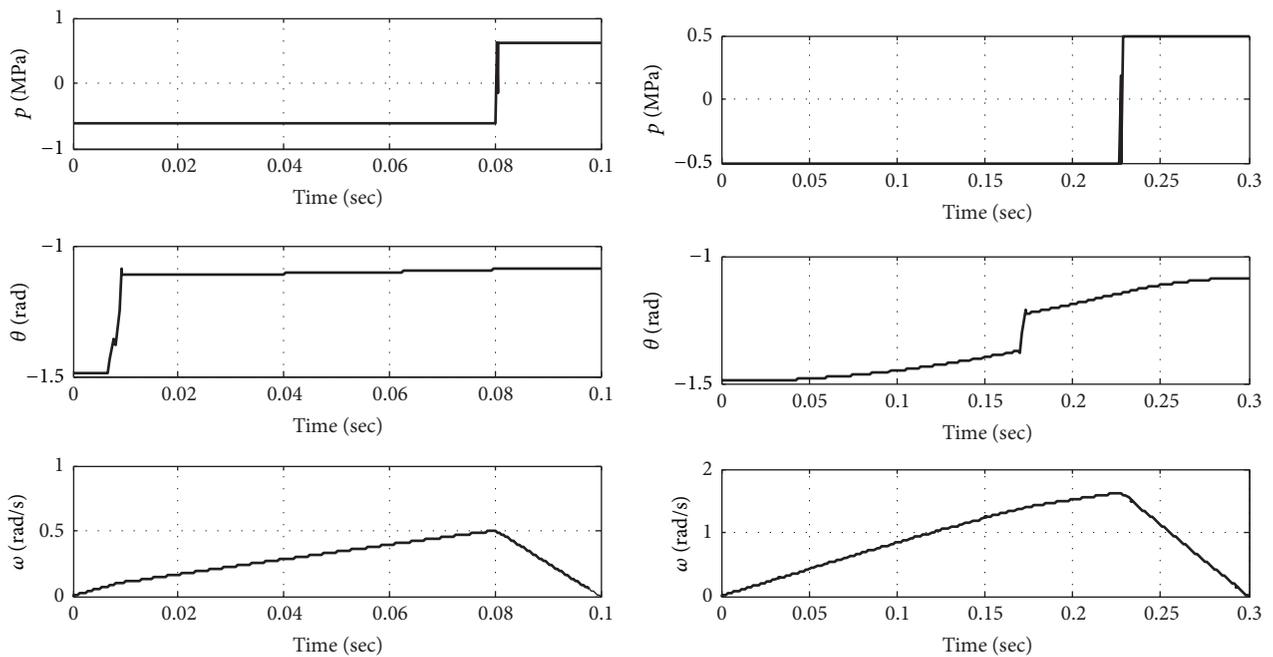


FIGURE 22: The optimal control result of thigh joint on the fourth step.

for its high accuracy. In this exploration, dynamics equation of PHR is successfully solved, and optimal input gas pressures of different actions in one gait are obtained, which offers guidance for engineering design.

- (3) Cruising ability of PHR is influenced by kinematic and dynamics characters synthetically. Forward distance of one step is obtained by kinematics analysis and gas pressures are obtained by dynamics analysis. The results indicate that cruising ability of designed PHR satisfies engineering needs.

Competing Interests

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

Acknowledgments

The exploration is supported by the Natural Science Foundation of China (11472058).

References

- [1] B. Verrelst, B. Vanderborght, J. Vermeulen, R. V. Ham, J. Naudet, and D. Lefeber, "Control architecture for the pneumatically actuated dynamic walking biped 'lucy,'" *Mechatronics*, vol. 15, no. 6, pp. 703–729, 2005.
- [2] M. Lavoie and A. L. Desbiens, "Design of a cockroach-like running robot for the 2004 SAE walking machine challenge," in *Climbing and Walking Robots*, pp. 311–318, Springer, 2006.
- [3] T. Morimoto, M. Aliff, T. Akagi, and S. Dohta, "Development of flexible haptic robot arm using flexible pneumatic cylinders with backdrivability for bilateral control," in *Proceedings of the 3rd International Conference on Intelligent Technologies and Engineering Systems (ICITES '14)*, vol. 345 of *Lecture Notes in Electrical Engineering*, pp. 231–237, Springer, 2016.
- [4] H. Qiu, S. Dohta, T. Akagi, S. Shimooka, and S. Fujimoto, "Analytical model of pipe inspection robot using flexible pneumatic cylinder," in *Proceedings of the 3rd International Conference on Intelligent Technologies and Engineering Systems (ICITES '14)*, vol. 345 of *Lecture Notes in Electrical Engineering*, pp. 325–334, Springer International Publishing, Cham, 2016.
- [5] J. A. Diez, F. J. Badesa, L. D. Lledó et al., "Design and development of a pneumatic robot for neurorehabilitation therapies," in *Robot 2015: Second Iberian Robotics Conference*, vol. 418 of *Advances in Intelligent Systems and Computing*, pp. 315–326, Springer, 2016.
- [6] F.-Z. Low, H. H. Tan, J. H. Lim, and C.-H. Yeow, "Development of a soft pneumatic sock for robot-assisted ankle exercise," *Journal of Medical Device*, vol. 10, no. 1, Article ID 014503, 2016.
- [7] M. Ramsauer, M. Kastner, P. Ferrara, R. Naderer, and H. Gattringer, "A pneumatically driven stewart platform used as fault detection device," *Applied Mechanics and Materials*, vol. 186, pp. 227–233, 2012.
- [8] M. F. Sliva and J. A. T. Machado, "A literature review on the optimization of legged robots," *Journal of Vibration and Control*, vol. 18, no. 12, pp. 1753–1761, 2011.
- [9] D. Sanz-Merodio, E. Garcia, and P. Gonzalez-De-Santos, "Analyzing energy-efficient configurations in hexapod robots for demining applications," *Industrial Robot*, vol. 39, no. 4, pp. 357–364, 2012.
- [10] J. Chen, Y. Liu, J. Zhao, H. Zhang, and H. Jin, "Biomimetic design and optimal swing of a hexapod robot leg," *Journal of Bionic Engineering*, vol. 11, no. 1, pp. 26–35, 2014.
- [11] S. S. Roy and D. K. Pratihari, "Dynamic modeling of energy efficient crab walking of hexapod robot," *Applied Mechanics and Materials*, vol. 110–116, pp. 2730–2739, 2012.
- [12] S. S. Roy, P. S. Choudhury, and D. K. Pratihari, "Dynamic modeling of energy efficient hexapod robot's locomotion over gradient terrains," in *Trends in Intelligent Robotics*, vol. 103, pp. 138–145, Springer, 2010.
- [13] S. S. Roy and D. K. Pratihari, "Effects of turning gait parameters on energy consumption and stability of a six-legged walking robot," *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 72–82, 2012.
- [14] S. S. Roy and D. K. Pratihari, "Kinematics, dynamics and power consumption analyses for turning motion of a six-legged robot," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3–4, pp. 663–688, 2014.
- [15] M. Luneckas, T. Luneckas, D. Udris, and N. M. F. Ferreira, "Hexapod robot energy consumption dependence on body elevation and step height," *Elektronika ir Elektrotechnika*, vol. 20, no. 7, pp. 7–10, 2014.
- [16] Z. Deng, Y. Liu, L. Ding, H. Gao, H. Yu, and Z. Liu, "Motion planning and simulation verification of a hydraulic hexapod robot based on reducing energy/flow consumption," *Journal of Mechanical Science and Technology*, vol. 29, no. 10, pp. 4427–4436, 2015.
- [17] P. Gonzalez de Santos, E. Garcia, R. Ponticelli, and M. Armada, "Minimizing energy consumption in hexapod robots," *Advanced Robotics*, vol. 23, no. 6, pp. 681–704, 2009.
- [18] B. Jin, C. Chen, and W. Li, "Power consumption optimization for a hexapod walking robot," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 71, no. 2, pp. 195–209, 2013.
- [19] Y. Zhu, B. Jin, W. Li, and S. Li, "Optimal design of hexapod walking robot leg structure based on energy consumption and workspace," *Transactions of Canadian Society for Mechanical Engineering*, vol. 38, no. 3, pp. 305–317, 2014.
- [20] T. Lee, *Computational Geometric Mechanics and Control of Rigid Bodies*, University of Michigan, Ann Arbor, Mich, USA, 2008.
- [21] Z. Terze, A. Muller, and D. Zlatar, "Lie-group integration method for constrained multibody systems in state space," *Multibody System Dynamics*, vol. 34, no. 3, pp. 275–305, 2015.
- [22] S.-B. Xu, S.-B. Li, and B. Cheng, "Theory and application of Legendre pseudo-spectral method for solving optimal control problem," *Control and Decision*, vol. 29, no. 12, pp. 2113–2120, 2014.
- [23] K. Tong, J. Zhou, and L. He, "Legendre-gauss pseudospectral method for solving optimal control problem," *Acta Aeronautica et Astronautica Sinica*, vol. 29, no. 6, pp. 1531–1537, 2008.
- [24] Y. Sun, M. R. Zhang, and X. L. Liang, "Improved Gauss pseudo-spectral method for solving a nonlinear optimal control problem with complex constraints," *Acta Automatica Sinica*, vol. 39, no. 5, pp. 672–678, 2013.
- [25] Y. Liu, Y. Zhao, J. Xu, and W. Liu, "Vehicle handling inverse dynamics based on Gauss pseudo-spectral method while encountering emergency collision avoidance," *Journal of Mechanical Engineering*, vol. 48, no. 22, pp. 127–132, 2012.

Research Article

Path Tracking Control of Automatic Parking Cloud Model considering the Influence of Time Delay

Yiding Hua,¹ Haobin Jiang,^{1,2} Yingfeng Cai,² Xupei Zhang,¹ Shidian Ma,² and Di Wu¹

¹School of Automobile and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China

²Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China

Correspondence should be addressed to Haobin Jiang; jianghb@ujs.edu.cn

Received 25 October 2016; Revised 6 December 2016; Accepted 19 December 2016; Published 14 February 2017

Academic Editor: Francisco Valero

Copyright © 2017 Yiding Hua 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 establishes the kinematic model of the automatic parking system and analyzes the kinematic constraints of the vehicle. Furthermore, it solves the problem where the traditional automatic parking system model fails to take into account the time delay. Firstly, based on simulating calculation, the influence of time delay on the dynamic trajectory of a vehicle in the automatic parking system is analyzed under the transverse distance D_{lateral} between different target spaces. Secondly, on the basis of cloud model, this paper utilizes the tracking control of an intelligent path closer to human intelligent behavior to further study the Cloud Generator-based parking path tracking control method and construct a vehicle path tracking control model. Moreover, tracking and steering control effects of the model are verified through simulation analysis. Finally, the effectiveness and timeliness of automatic parking controller in the aspect of path tracking are tested through a real vehicle experiment.

1. Introduction

In recent years, the “parking difficulty” problem in modern cities has become more and more notable. Parking operation in a parking area near crowded and narrow urban roads and community roads is not easy for many drivers [1, 2]. Therefore, assisted parking driving technology has become one of the research hotspots in the car engineering field and involves knowledge about electromagnetism, environment-aware sensors, signal processing, information fusion, model identification, automatic control and electric power steering direction, and automotive electronics [3–5]. Car makers pay much attention to assisted parking system and successively launched their own assisted parking driving system.

Automatic parking and its control have been studied widely in the domestic and foreign automobile industry, which has yielded fruitful achievements. However, some key technologies in the research of automatic parking have not been researched, such as time delay of automatic parking path tracking and control system.

At present, some research studies have been carried out on the parking path tracking control. Some intelligent control methods are used more extensively, such as fuzzy logic

controller, neural network controller, and genetic algorithms (GAs). Yasunobu and Murai [6] proposed a human experience based on fuzzy logic control theory. A fuzzy logic control algorithm was employed to design the parking controller [7] based on a model car test or simulation work. Jenkins and Yuhas [8] introduced a simplified neural network controller trained on the basis of kinematics data. Daxwanger and Schmidt [9] employed fuzzy and neural network schemes to develop a visually based model car backward-parking controller with online autoparking steering angle command. Tayebi and Rachid [10] designed a robust time-varying state feedback parking controller by using the Lyapunov stability rule for a wheel mobile robot. Nevertheless, the fuzzy or neural network controller must be designed on the basis of expertise and trial-and-error work or on a complicated vehicle dynamics model. This is not convenient for practical application. At the same time, neither one of the papers mentioned above addressed the issue about time delay of automatic parking path tracking.

Time delay of automatic parking path tracking mainly includes the following aspects: (1) transmission time delay of measurement signal from the sensor to the control computer; (2) time delay caused by the calculation of control law; (3)

transmission time delay of control sign from computer to actuator; (4) time delay of actuator; (5) time required to establish control [11]. Time delay greatly affects the performance of the system [12]. It might even cause instability of feedback control system, and so the vehicle will fail to track the ideal parking path. This leads to the failure of the automatic parking system and even collision accidents, which seriously affect the practicability and accuracy of the automatic parking system.

The influence of time delay on the automatic parking system has been reflected in some existing research; for example, Gutjahr and Werling [13] designed time optimal trajectories for braking based on the prediction of the future vehicle motion to compensate for actuator time delay. Choi and Song [14] designed a fault detection and handling method for automatic valet parking to resolve packet loss and time delay of communication. Song et al. [15] designed a lateral controller to be robust enough to compensate for the noise and time delay. However, the uncertainty of time delays was not considered in the papers mentioned above [16–18].

Cloud model is an uncertain transformation model between the qualitative concept proposed by Academician Li Deyi and its quantitative value. Characterized by the coexistence of the uncertainty and certainty as well as the stability during the course of knowledge representation, the cloud model has reflected the fundamental principle of the biological evolution in nature [19–21]. By virtue of its characteristics such as easy implementation of derivation process, simple rules, and strong robustness, it has been widely utilized in the field of intelligent control and subjective evaluation [22]. Normal cloud model is one of the most important among cloud models. Due to its excellent mathematical characteristics, it can describe a large number of uncertain phenomena in different disciplines [23].

Thus, this paper proposes a novel path tracking controller based on cloud model in order to solve the shortcomings of the existing the fuzzy and neural network path tracking controller and to consider the uncertainties of time-delay problems.

The paper is organized as follows. In the next section, a vehicle kinematics model is built and kinematic constraints are analyzed. Then, the influence of time delay on the parking trajectory is simulated with MATLAB. In order to meet the complex control requirements, one-dimensional and two-dimensional single rules are organically combined to form the multirule reasoning model. Subsequently, control strategy is verified with straight path and circular path tracking. At last, a real vehicle experiment is presented to verify the effectiveness and timeliness of automatic parking path tracking controller based on cloud model. The paper closes with the conclusions and references.

2. Analysis of the Vehicle Kinematics Model and Kinematic Constraints

This paper establishes a kinematic model of vehicles, based on which the path planning method of automatic parking system is analyzed. As shown in Figure 1, (x_r, y_r) and (x_f, y_f) are the

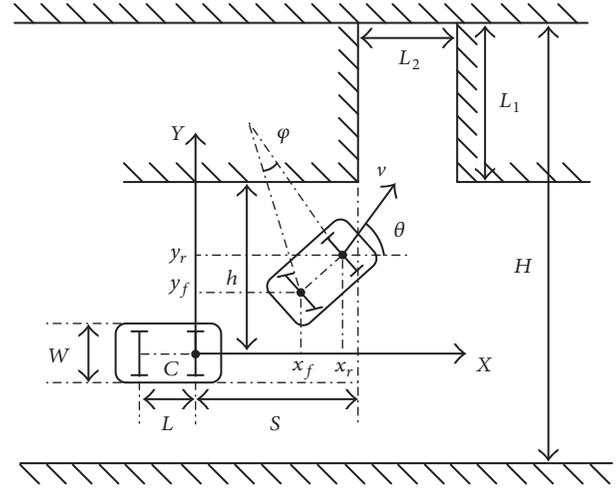


FIGURE 1: Simplified car model for parking.

midpoint coordinates of rear and front axles of the vehicle, respectively, W is indicated as the wheel tread, H is the width of the road, L_1 and L_2 are separately the width and length of the target parking space, h is the distance between the midpoint of the rear axle and the lateral barrier, S represents the distance between the midpoint of the rear axle and the tail end of the barrier at the front of the target parking space, θ denotes the course angle of the vehicle, φ is the Ackerman angle, and clockwise direction is positive.

The rear-wheel lateral velocity (vertical to wheel direction) is zero, and the vehicle motion equation in the vertical direction can be obtained:

$$\dot{x}_r \cdot \sin \theta - \dot{y}_r \cdot \cos \theta = 0. \quad (1)$$

According to Ackerman steering geometry principle, Ackerman angle φ in the process of vehicle steering approximates to the steering angle of midpoint of vehicle front axle. The central steering angle of the front axle of the vehicle is linearly proportional to steering wheel angle γ . Hence,

$$\gamma = K \cdot \varphi, \quad (2)$$

where K is proportional constant.

The midpoint of the rear axle of the parked vehicle is the origin of coordinates, and the coordinate system is established as Figure 1. The positional relationship between the midpoints of front and rear axles of the vehicle can be obtained. Hence,

$$\begin{aligned} x_r &= x_f + L \cdot \cos \theta, \\ y_r &= y_f + L \cdot \sin \theta. \end{aligned} \quad (3)$$

Formula (3) is differentiated to obtain the velocity relationship between the midpoints of front and rear axles of the vehicle. Therefore,

$$\begin{aligned} \dot{x}_r &= \dot{x}_f - L\dot{\theta} \cdot \sin \theta, \\ \dot{y}_r &= \dot{y}_f + L\dot{\theta} \cdot \cos \theta. \end{aligned} \quad (4)$$

Substituting (4) into (1), we can obtain the vehicle kinematics relation:

$$\dot{x}_f \cdot \sin \theta - \dot{y}_f \cdot \cos \theta - L \cdot \dot{\theta} = 0. \quad (5)$$

In addition, at a certain moment of parking in a garage, the velocity of midpoint of the front axle along the direction of axis is as follows:

$$\begin{aligned} \dot{x}_f &= v \cdot \cos(\theta + \varphi), \\ \dot{y}_f &= v \cdot \sin(\theta + \varphi). \end{aligned} \quad (6)$$

Substituting (6) into (5), we can obtain

$$\dot{\theta} = -\frac{v \cdot \sin \varphi}{L}. \quad (7)$$

Substituting (6) and (7) into (5), we can obtain the velocity of midpoint of front axle along the direction of axis:

$$\begin{aligned} \dot{x}_r &= v \cdot \cos \theta \cos \varphi, \\ \dot{y}_r &= v \cdot \sin \theta \cos \varphi. \end{aligned} \quad (8)$$

Formulas (7) and (8) are denoted as the kinematic equation of vehicle. Hence,

$$\begin{aligned} \dot{x}_r &= v \cdot \cos \theta \cos \varphi, \\ \dot{y}_r &= v \cdot \sin \theta \cos \varphi, \\ \dot{\theta} &= -\frac{v \cdot \sin \varphi}{L}. \end{aligned} \quad (9)$$

Formula (9) is utilized to integrate time t , and the obtained movement locus equations of midpoint of rear axle are

$$\begin{aligned} x_r(t) &= -L \cdot \cot \theta \cdot \sin\left(\frac{v \cdot \sin \theta}{L} \cdot t\right), \\ y_r(t) &= L \cdot \cot \theta \cdot \sin\left(\frac{v \cdot \sin \theta}{L} \cdot t\right) - L \cdot \cot \theta, \\ x_r^2 + (y_r + L \cdot \cot \theta)^2 &= (L \cdot \cot \theta)^2. \end{aligned} \quad (10)$$

According to the geometric relation between parameters of the vehicle and positions of each coordinate in Figure 1, movement locus equations of four vehicle wheels and envelope points can be obtained. Thereby, the actual movement locus of vehicle in the whole process from starting point to terminal point of parking can be calculated.

Parking movement process should not only meet the vehicle kinematic and geometric characteristics, but also give full consideration to external factors such as security, accuracy, and efficiency of parking. In this section, the influence of barrier in parking environment on the process of parking is analyzed.

Based on the planned parking path, namely, the rear parking locus function, the theoretical curvature ρ of vehicle

at arbitrary point in the process of parking in a garage can be given as

$$\rho = \frac{y''}{[1 + (y')^2]^{3/2}}. \quad (11)$$

According to the relation of Ackerman angle,

$$\tan \varphi = \frac{L}{R}, \quad (12)$$

where L is wheel base, R is the radius of turning circle, and $R = 1/\rho$.

According to formulas (11) and (12), Ackerman angle of the vehicle at arbitrary point is

$$\varphi = \arctan\left(\frac{L \cdot y''}{[1 + (y')^2]^{3/2}}\right). \quad (13)$$

The movement locus of A, B, C, and D can be obtained based on coordinates of midpoints of rear axle and their mutual relationship. Hence,

$$\begin{aligned} x_A &= x + (L_f + L) \cdot \cos \theta + \frac{W}{2} \cdot \sin \theta, \\ y_A &= y - (L_f + L) \cdot \sin \theta + \frac{W}{2} \cdot \cos \theta, \\ x_B &= x + (L_f + L) \cdot \cos \theta - \frac{W}{2} \cdot \sin \theta, \\ y_B &= y - (L_f + L) \cdot \sin \theta - \frac{W}{2} \cdot \cos \theta, \\ x_C &= x - L_r \cdot \cos \theta - \frac{W}{2} \cdot \sin \theta, \\ y_C &= y + L_r \cdot \sin \theta - \frac{W}{2} \cdot \cos \theta, \\ x_D &= x - L_r \cdot \cos \theta + \frac{W}{2} \cdot \sin \theta, \\ y_D &= y + L_r \cdot \sin \theta + \frac{W}{2} \cdot \cos \theta. \end{aligned} \quad (14)$$

On the basis of the parking kinematic model established in the above section, there are four positions where the danger exists in the process of parallel parking (as shown in Figure 2): (a) collision between point C and the left boundary of the carriageway, or the collision between the boundary beyond the current carriageway and vehicle running on the left carriageway in the process of parking; (b) collision between tail of vehicle which has not been parked and the barrier or vehicle at the front of right available parking space; (c) collision between point B of the vehicle and the barrier after which the vehicle is parked in the space; (d) collision between point D as well as point A of the vehicle and the barrier after which the vehicle is parked in the space.

In order to ensure the safety in the parking process and reduce the risk of collision between the vehicle and

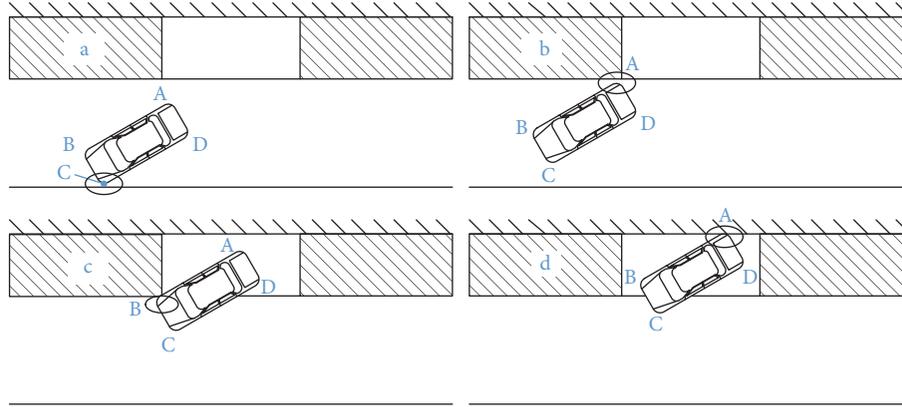


FIGURE 2: Parallel parking possible collisions.

the barrier, locus function needs to meet the following conditions:

$$\begin{aligned}
 &\text{When } x_A \in [0, s_o - L_2], \quad y_A < h_0; \\
 &\text{When } x_A \in [0, s_o - L_r], \quad y_A < h_0 + L_1; \\
 &\text{When } x_B \in [0, s_o - L_2], \quad y_B < h_0; \\
 &\text{When } x_B \in [0, s_o - L_r], \quad y_B < h_0 + L_1; \quad (15) \\
 &\text{When } x \in [0, s_o - L_r], \quad y_C < h_0 + L_1 - h; \\
 &\text{When } x = s_o - L_r, \quad y_C > h_0; \\
 &\text{When } x = s_o - L_r, \quad y_D > h_0.
 \end{aligned}$$

The analysis of the constraints of kinematics in the process of parking lays the foundation for vehicle trajectory planning and path tracking in the process of parking and ensures the safety of the running vehicle during parking.

3. Influence of Time Delay on the Parking Trajectory

Due to the variation of arc length of vehicle wheels in each process generated from the time delay, this system can calculate the passing arc length of wheels at the three stages of parking on the basis of analysis of the difference in distance D_{lateral} between the vehicle and target parking space.

In actual parking process, there are time errors in each stage, especially at the points between any two stages. ΔL_1 , ΔL_2 , and ΔL_3 are assumed as the arc length deviations at each stage under the average velocity speed of 2 km/h. When D_{lateral} is 0.5 m, 1 m, and 1.5 m, respectively, the influence of time delay on the deviation of parking path is analyzed.

By analyzing Figures 3–5, it can be known that even a short time delay can affect the parking trajectory. The red trajectory in the figure indicates the theoretically ideal parking trajectory obtained from simulation, and the blue trajectory represents the actual parking trajectory affected by time error. The time delay most significantly influences the trajectory between the first and the second stages, and the influence between the second and the third stages is

less significant. Furthermore, we can realize that the greater the time delay, the more obvious the trajectory error. If the deviation is excessive, the tested vehicle cannot be accurately parked in the target parking space theoretically. At present, only the influence of the time delay between every two stages on the whole parking trajectory deviation is considered. However, the effect of superposition of time delay on parking trajectory has not been analyzed.

4. Control Strategy

This paper constructs the tracking control model and adjusts the characteristic parameters of the cloud on the basis of uncertain cloud reasoning model. The controller can enable the vehicle to rapidly and accurately track the expected path.

4.1. Basic Principles of Cloud Model. The cloud is a model using the linguistic value to represent the uncertainty conversion between a qualitative concept and its quantitative representation. Suppose U is a quantitative domain expressed in precise values and A is a qualitative concept in U . If a quantitative value $x \in U$ is a random realization of the qualitative concept A and the membership of x to A , $\mu(x) \in [0, 1]$, is a random number with a stable tendency: $\mu: U \rightarrow [0, 1], \forall x \in U, x \rightarrow \mu(x)$, then the distribution of x on domain U is called the cloud and each x is called a cloud droplet [24]. Particularly, it is assumed that $R_1(E_1, E_2)$ can be expressed as a random function obeying normal distribution, where E_1 is expected value and E_2 is standard deviation. If $x (x \in U)$ and $\mu(x)$ satisfy the equations, which can be expressed as

$$\begin{aligned}
 x &= R(E_x, E_n), \\
 p &= R(E_n, H_e), \\
 \mu &= e^{-(1/2)((x-E_x)/p)^2},
 \end{aligned} \quad (16)$$

then the distribution of x on domain U is called the normal cloud [25]. μ is the degree of membership cloud for input variable x . Cloud numerical features can be utilized to reflect the overall characteristics of the qualitative concept expressed

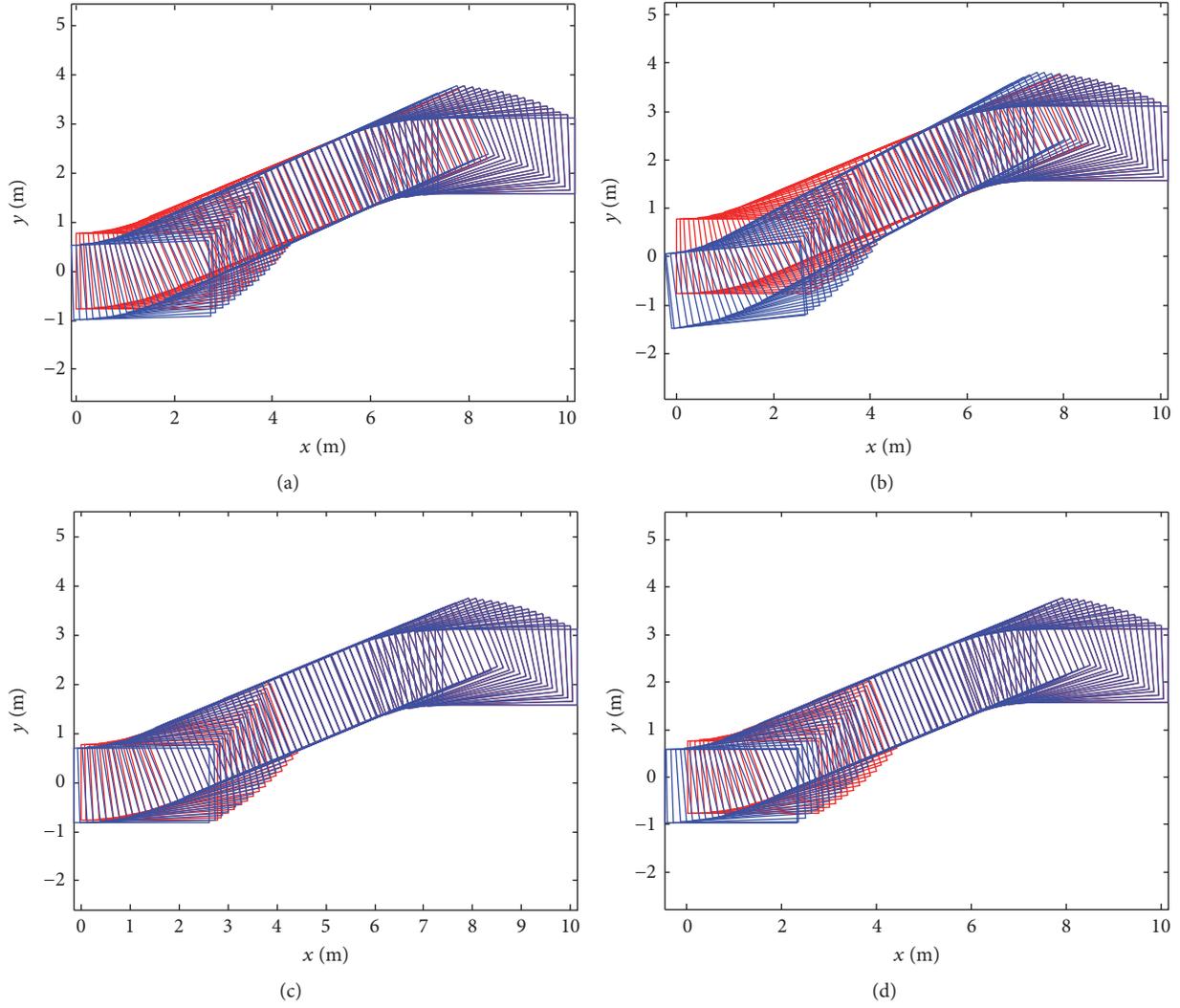


FIGURE 3: Effect of time delay on parking trajectory when $D_{\text{lateral}} = 0.5$ m. (a) The time delay is 0.3 s between the first and the second stages. (b) The time delay is 0.9 s between the first and the second stages. (c) The time delay is 0.3 s between the second and the third stages. (d) The time delay is 0.9 s between the second and the third stages.

by the cloud model. The cloud droplets have three numerical features. In (16), E_x , E_n , and H_e denote the expectation, entropy, and hyperentropy, respectively, which are used to describe the numerical characteristics of cloud. E_x is the expectation of cloud droplets in the distribution of the domain and is the most typical point that represents this qualitative concept. E_n is the uncertain measurement of the qualitative concept and reflects the relevance of fuzziness and randomness. H_e is the uncertain measurement of entropy and is determined by the fuzziness and randomness.

R_2 is assumed to be a two-dimensional random function obeying normal distribution, where E_1 and E_2 are expected value and E_3 and E_4 are standard deviation. Hence,

$$\begin{aligned} (x, y) &= R_2(E_1, E_2, E_3, E_4), \\ (p_x, p_y) &= R_2(E_3, E_4, H_{e_x}, H_{e_y}), \\ \mu' &= e^{-(1/2)[(x-E_1)^2/p_x^2 + ((y-E_2)^2/p_y^2)]}. \end{aligned} \quad (17)$$

The cloud model constructed by data meeting formula (17) on $\text{drop}(x, y, \mu')$ is a two-dimensional normal cloud model, which is abbreviated as two-dimensional normal cloud. The data constructing this cloud model for $\text{drop}(x, y, \mu')$ is called two-dimensional cloud droplets [26].

Three normal clouds with different characteristics are shown in Figure 6. Compared with the three clouds, it can be found that the smaller the value of E_n is, the more divergent the cloud will be. And it can also be found that the bigger the value of H_e is, the more divergent the cloud will be. Normal cloud is composed of some cloud droplets, which can reflect the fuzziness. Cloud model is not described through certain functions, therefore, to enhance the processing capacity for uncertainty.

The process of reasoning about the uncertain rules of the cloud model is the target rules inferred and calculated through the known conditions in a certain environment [27]. Cloud Generator (CG) is the algorithm of cloud model. The

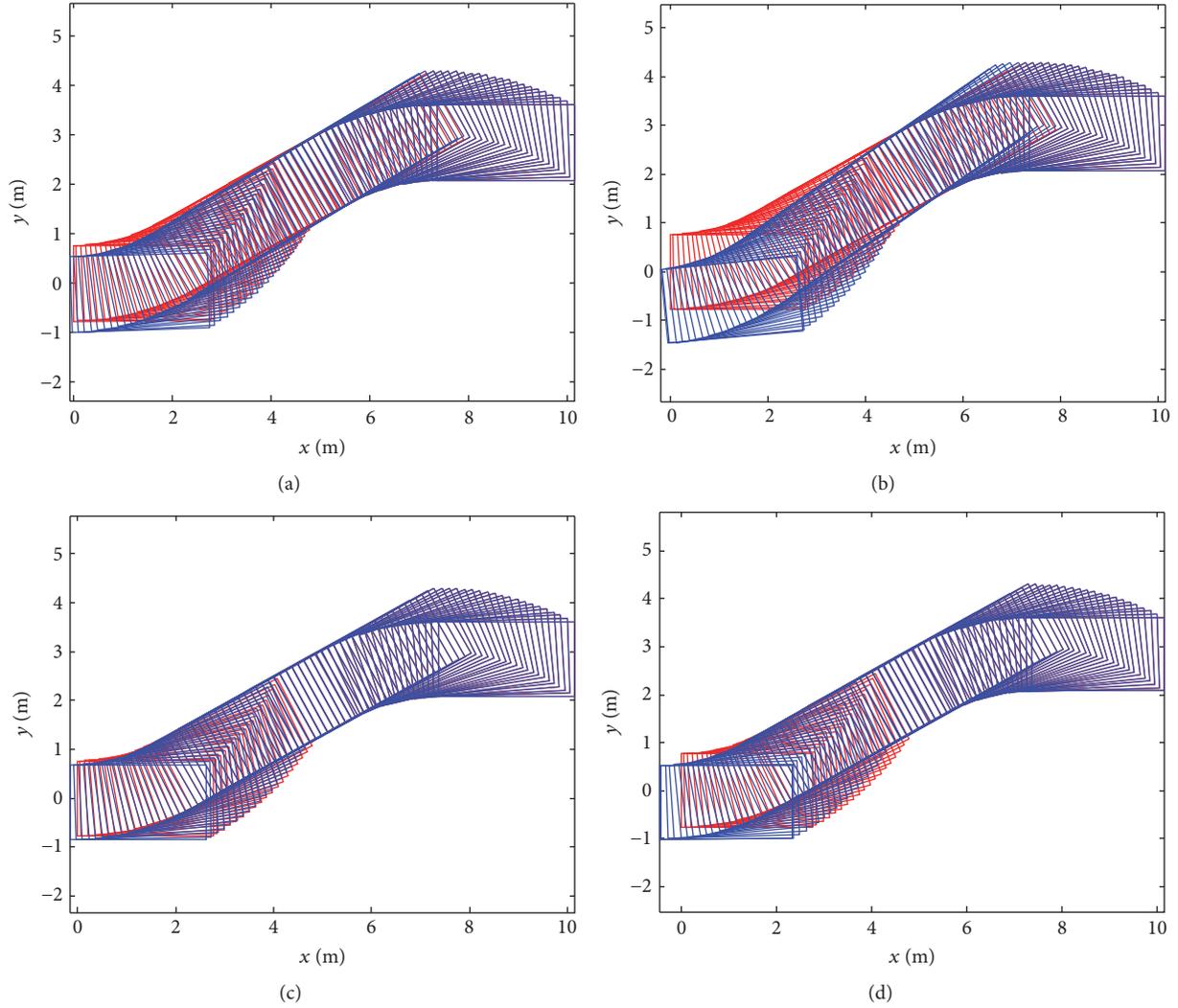


FIGURE 4: Effect of time delay on parking trajectory when $D_{\text{lateral}} = 1.0$ m. (a) The time delay is 0.3 s between the first and the second stages. (b) The time delay is 0.9 s between the first and the second stages. (c) The time delay is 0.3 s between the second and the third stages. (d) The time delay is 0.9 s between the second and the third stages.

inputs of the generator are the three numerical characteristics. The outputs are cloud droplets. CG can realize the mapping from qualitative data to quantitative data. There are many CGs such as Forward Cloud Generator, Backward Cloud Generator, X Condition Cloud Generator, and Y Condition Cloud Generator.

Consider the one-dimensional cloud model of single-rule reasoning: if x , then y . Its structure is shown in Figure 7(a), where x condition cloud model is (E_x, E_{nx}, H_{ex}) and y condition cloud model is (E_y, E_{ny}, H_{ey}) . When the rule is activated more than once by the quantitative input value x , CG_X produces a set of μ_i values, while μ_i generates a set of y_i values through CG_Y . CG_X can be presented by formula (16). CG_Y can be expressed as follows:

$$P = R(E_{ny}, H_{ey}) \quad (18)$$

$$y_i = E_y \pm \sqrt{-2 \ln(\mu_i)} \cdot P.$$

Both X and Y CGs are the basis of the construction of cloud model uncertainty reasoning, which form a single-rule generator after being connected. The two-dimensional cloud model of single-rule reasoning is shown in Figure 7(b).

4.2. Path Tracking Control Model. The basic idea of the parking path tracking control model is as follows: the vehicle runs at a certain speed. According to the known path and the running parameters of the vehicle, proper steering angle control parameters are calculated. The driving direction of the vehicle is changed, and it runs along the preplanned path. In this process, the steering control parameters are obtained through relevant inference based on the given running parameters. The mapping relation of individual one-dimensional or two-dimensional single-rule reasoning between input and output is simple and can hardly meet the complex control requirements. In this paper, the above-mentioned one-dimensional and two-dimensional single

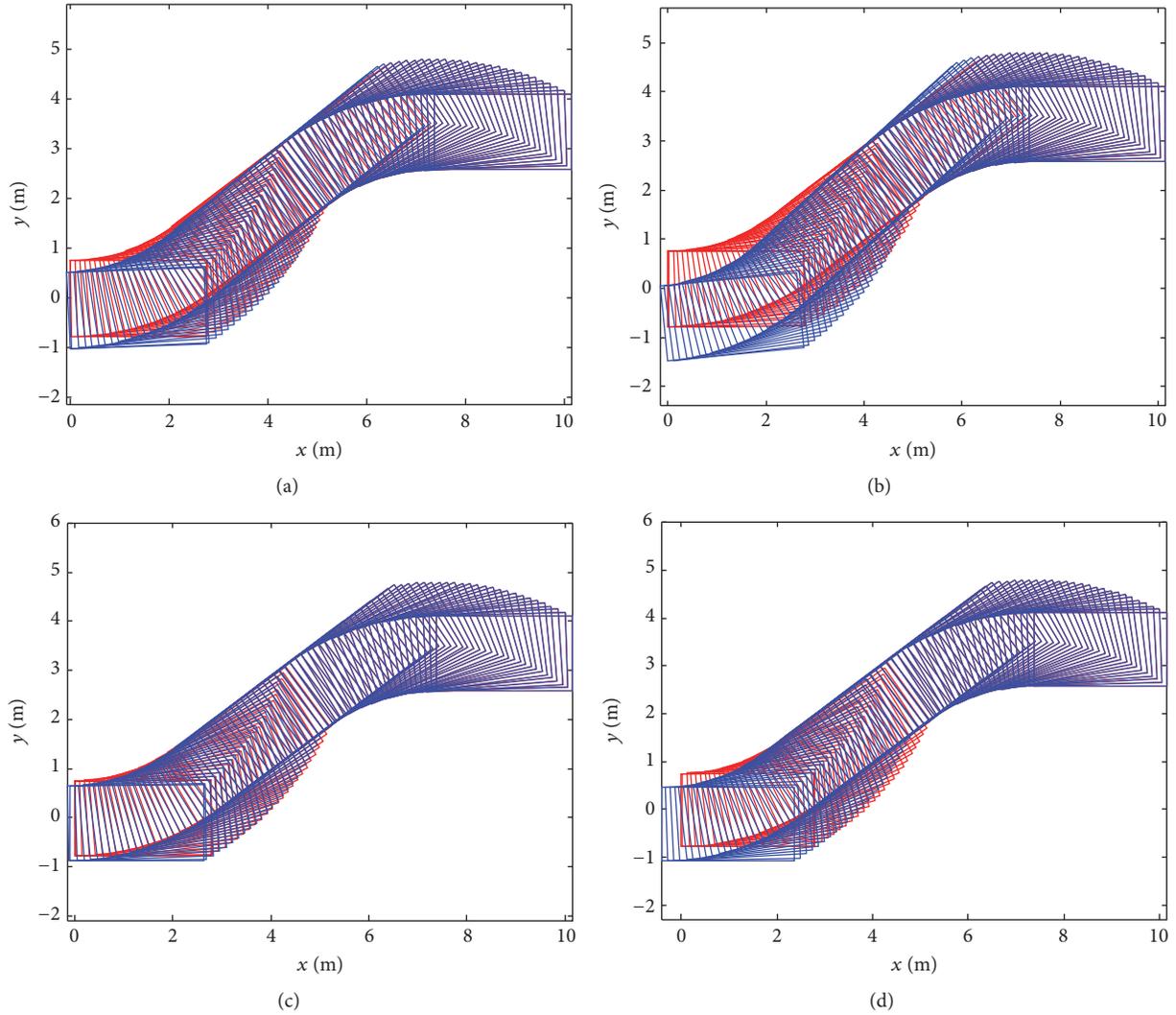


FIGURE 5: Effect of time delay on parking trajectory when $D_{\text{lateral}} = 1.5$ m. (a) The time delay is 0.3 s between the first and the second stages. (b) The time delay is 0.9 s between the first and the second stages. (c) The time delay is 0.3 s between the second and the third stages. (d) The time delay is 0.9 s between the second and the third stages.

rules are organically combined to form the multirule reasoning model shown in Figure 8. This model can complete both linear and nonlinear reasoning requirements. A more difficult prediction at a higher level can be carried out by multirule deduction.

PID controller has the advantages of simple structure, convenient adjustment, and parameter tuning contact in engineering, but the choice of traditional PID regulator parameters mainly depends on repeated tests and experience, when the state of the object changes. PID controller consists of the proportion of unit P, integral unit I, and differential unit D. In this paper, in order to facilitate the comparison, the PID controller is designed. PID controller of automatic parking is applied to adjust actual steering wheel angle output by setting k_p , k_i , and k_d .

The working process of the model is as follows: the controlled quantity relevant parameters x_1 , x_2 , and x_3 are assumed as the input of the multirule generator, according

to which the multirule derivation is carried out. The results are applied to the adjustment of constant coefficients k_{PD} and k_i so as to adjust and control the output parameter range. Finally, the weighted average output steering electrical machine is controlled in real time.

Input variables x_1 , x_2 , and x_3 activate different X conditional cloud PCG_{A_i, B_i} and CG_{a_j} , generating different μ_i and μ_j , respectively. The cloud droplet is generated through Y conditional cloud CG_{c_i} and CG_{b_j} . Through the weighted average after adjustment of constant coefficient, the inferential results could be obtained, and the generator process is completed.

Inference rule of rule base of multidimensional cloud model is formulated.

If A_i , B_j , and a_k , then R_{ijk} , where the value ranges of i , j , and k are 1~5. The corresponding rules of the front and rear part are constructed in Figure 9. The consequents of the rules library are listed in Table 1.

TABLE I: The consequents of the rules library.

$R_{11k} = (C_1, b_k)$	$R_{12k} = (C_1, b_k)$	$R_{13k} = (C_1, b_k)$	$R_{14k} = (C_2, b_k)$	$R_{15k} = (C_2, b_k)$
$R_{21k} = (C_1, b_k)$	$R_{22k} = (C_1, b_k)$	$R_{23k} = (C_2, b_k)$	$R_{24k} = (C_3, b_k)$	$R_{25k} = (C_3, b_k)$
$R_{31k} = (C_2, b_k)$	$R_{32k} = (C_2, b_k)$	$R_{33k} = (C_3, b_k)$	$R_{34k} = (C_4, b_k)$	$R_{35k} = (C_4, b_k)$
$R_{41k} = (C_3, b_k)$	$R_{42k} = (C_4, b_k)$	$R_{43k} = (C_4, b_k)$	$R_{44k} = (C_5, b_k)$	$R_{45k} = (C_5, b_k)$
$R_{51k} = (C_4, b_k)$	$R_{52k} = (C_5, b_k)$	$R_{53k} = (C_5, b_k)$	$R_{54k} = (C_5, b_k)$	$R_{55k} = (C_5, b_k)$

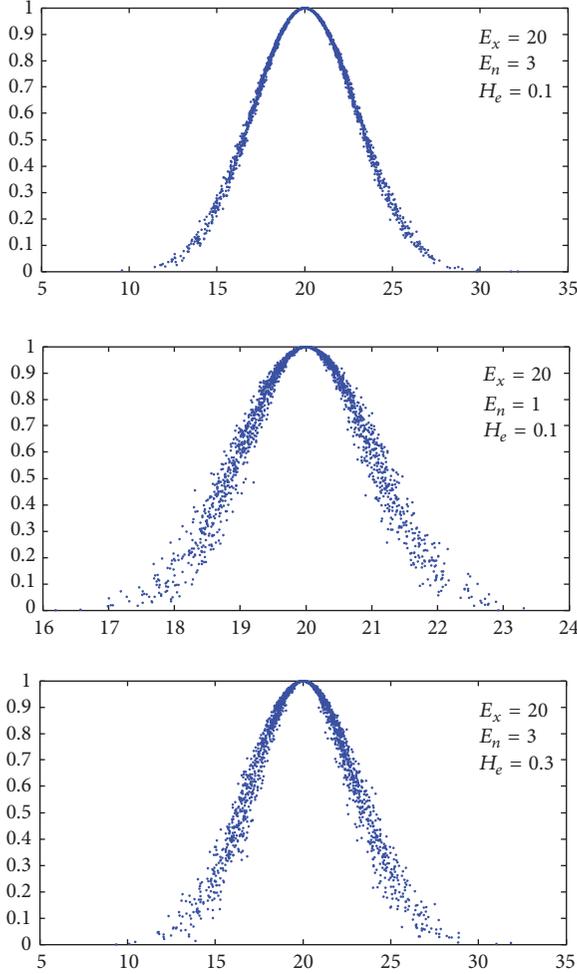


FIGURE 6: Three examples of the normal cloud.

A_i means deviation of actual steering wheel angle and theoretical steering wheel, B_j means deviation differential of actual steering wheel angle and theoretical steering wheel, a_k means deviation integral of actual steering wheel angle and theoretical steering wheel, C_i means control outputs of PD type two-dimensional cloud model mapper, and b_j means control outputs of I type two-dimensional cloud model mapper.

According to the contents discussed above, it is known that if parking path is given, Ackerman angle of the vehicle at arbitrary point on the path could be obtained. Parking kinematic model is taken as the controlled object of trajectory tracking control model. In order to control the vehicle

tracking path through control model, the deviation μ between Ackerman angle φ of the vehicle and the theoretical value φ_0 is taken as the input of the control model, among which φ_0 can be calculated through formula (13). By inputting the deviation μ , deviation integral μ_i , and deviation differential μ_c into the cloud model generator, the output controlled variable can be rapidly adjusted based on variation of deviation, so that the vehicle can be driven tracking the path.

In actual operation, in order to facilitate the measurement, the variable input of the control model is converted into angle deviation value of steering wheel according to formula (13). To avoid the adverse effects of three aspects of time delay, that is, (1) time delay in calculation of steering control law, (2) transmission time delay of control sign from microprocessor to steering motor, and (3) time required to establish control algorithm, the range of deviation μ is selected as $(-10, 10)$. -10 , -5 , 0 , 5 , and 10 are, respectively, taken as the five expected values of the front five rules. The numerical characteristic value of cloud is input. The algorithm can reduce the negative influence of time delay on the basis of guaranteeing higher control precision. Hence,

$$\begin{aligned}
 A_1 &= B_1 = a_1 = (-10, 5, 0.1); \\
 A_2 &= B_2 = a_2 = (-5, 3, 0.1); \\
 A_3 &= B_3 = a_3 = (0, 3, 0.1); \\
 A_4 &= B_4 = a_4 = (5, 3, 0.1); \\
 A_5 &= B_5 = a_5 = (10, 5, 0.1).
 \end{aligned} \tag{19}$$

Similarly, the output of the control model is the adjustment amount of the steering angle, and the numerical characteristic value is output. Therefore,

$$\begin{aligned}
 C_1 &= b_1 = (-12, 5, 0.1); \\
 C_2 &= b_2 = (-6, 3, 0.1); \\
 C_3 &= b_3 = (0, 3, 0.1); \\
 C_4 &= b_4 = (6, 3, 0.1); \\
 C_5 &= b_5 = (12, 5, 0.1).
 \end{aligned} \tag{20}$$

Figure 10 is the structural diagram of path tracking control model constructed in MATLAB/Simulink in this paper. Its running process is as follows: the current state parameters (x_c , y_c , θ_c , and φ) of the midpoint of the rear axle are obtained through a vehicle-mounted sensor. Vector quantity (x_c , y_c) is input into path tracking module, through which the ideal steering angle φ_i of the current reference

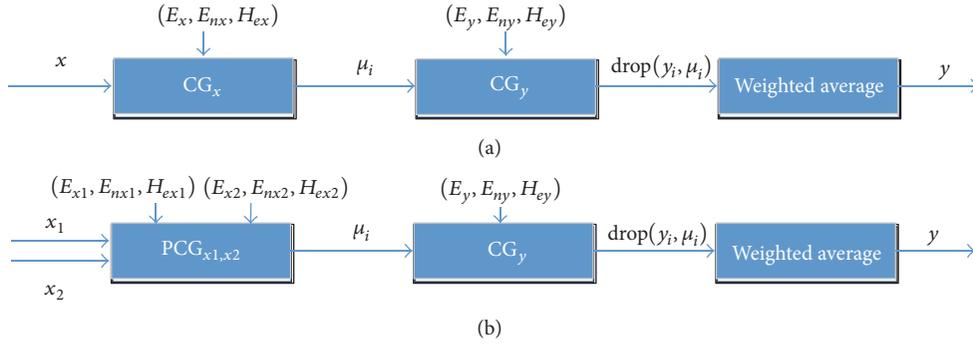


FIGURE 7: Single positive cloud rule generator.

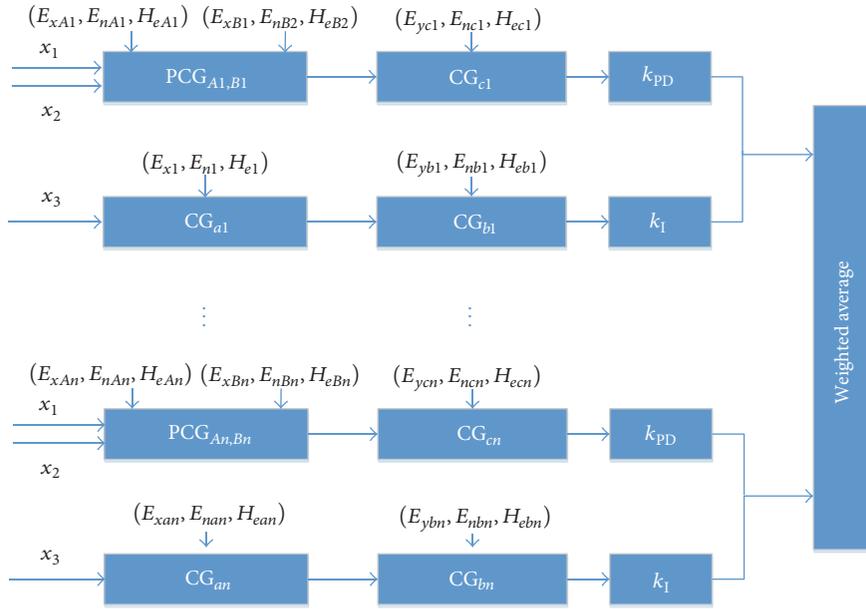


FIGURE 8: Multiple rule reasoning model.

point could be calculated. Furthermore, $\Delta\varphi$ is obtained by comparing with the actual steering angle φ and taken as the input of multirule generator to obtain the adjustment amount of the steering angle in the parking process. With the movement of the target point in the planning path, $\Delta\varphi$ is repetitively calculated until the tracking is completed.

Parking path tracking control model is established on MATLAB/Simulink simulation platform. It is assumed that the vehicle is running at the speed of 2 m/s, and the tracking simulations on different paths are carried out, respectively.

The trajectory is assigned as $y = x$ and $x \in (0, 4)$. The starting point of the path is $(0, 0)$, and direction angle is 0° . The results of tracking simulation are shown in Figure 11, where the full line is the prescribed path and the dotted line is the tracking trajectory. Ten sample points are selected to calculate the standard deviation, which is 0.15.

The trajectory is specified as $x^2 + y^2 = 1$. The starting point of the path tracking is $(-2, 0)$, and the direction angle is 0° . The simulation results are shown in Figure 12. Twenty-five

sample points are selected to calculate the standard deviation, which is about 0.1.

According to Figures 11 and 12, as for the straight line tracking, since the steering angle variation is 0, after the rapid convergence at the starting point in tracking, the fluctuation of the tracking points is less. As for the circular trajectory tracking, the steering angle significantly changes. The tracking model can achieve rapid convergence, and the tracking error is within the reasonable range.

5. Experiment and Result Analysis

Real vehicle experiment of automatic parking consists of the following components: automatic parking control system, parking space scene detection equipment, WAVEBOOK512H portable dynamic data acquisition system (including a notebook computer, and DASlab8.0 data analysis software), Agilent U1620A portable oscilloscope, tape, connecting wire, and Haima Familia test vehicle.

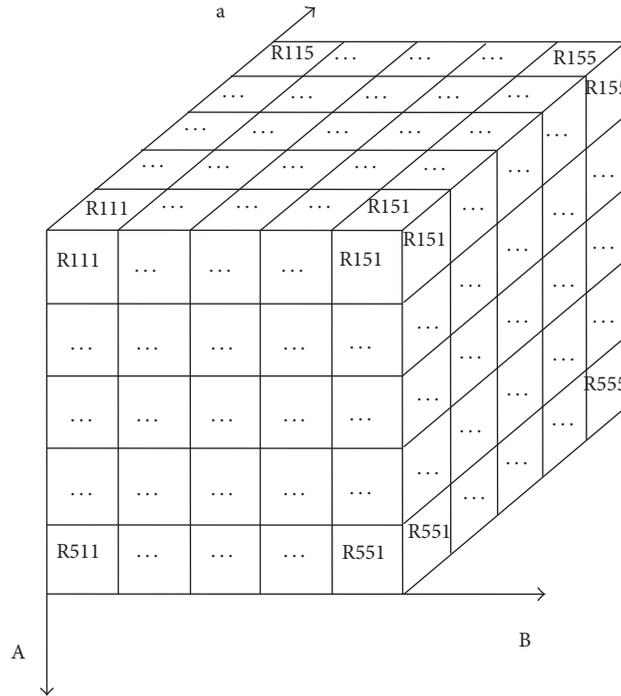


FIGURE 9: Rule library.

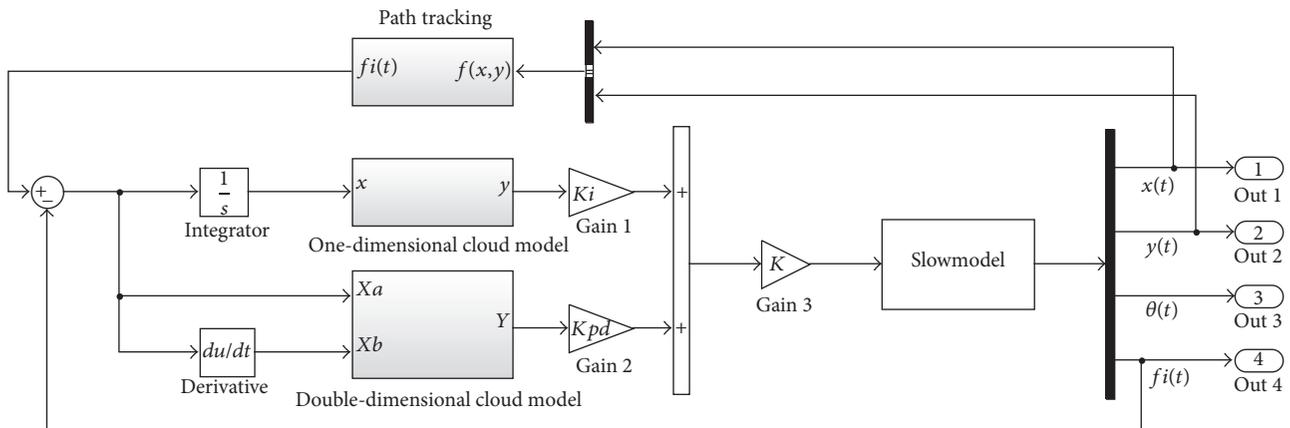


FIGURE 10: Path tracking control model.

Hardware circuit board of the automatic parking controller is developed according to the circuit principle diagram. The parking controller is loaded on the vehicle to carry out the test. Experimental diagram of parking controller hardware circuit board and parking controller loading test is shown in Figure 13.

The ultrasonic sensor of the system is mainly arranged in the front and rear, as well as the right side of the vehicle. The front and rear sensors are mainly short range radar, and the right one is mainly long distance sensor, which is shown in Figure 14.

In the parking test field, the test site is arranged according to the size of target parking space of the tested vehicle determined in the second section. The system is loaded on

the tested vehicle with adaptability adjustment. The automatic parking system is started for actual parking experiment, which is shown in Figure 15.

At the starting point of the vehicle parking, the geometric method is utilized to plan the ideal parking trajectory. The segmented curve is fit to the parking path. Path tracking control model is used to track the path. The starting point is (7.1, 2.3) and direction angle is 0° , when $D_{lateral} = 0.5$ m.

Three different lateral distances ($D_{lateral}$) of the model car are planned to execute the roadside parallel autoparking experiments. Experimental results and pictures for a fixed vehicle speed of 5 km/s are shown here for explanation. Experimental results are listed in Table 2 for comparison with the data from the PID controller and cloud model controller

TABLE 2: Path tracking control performance comparison.

Control scheme	Experimental data for the following lateral distances ($D_{lateral}$)					
	$D_{lateral}, 0.5\text{ m}$		$D_{lateral}, 1.0\text{ m}$		$D_{lateral}, 1.5\text{ m}$	
	Minimum of Y error (m)	Maximum of Y error (m)	Minimum of Y error (m)	Maximum of Y error (m)	Minimum of Y error (m)	Maximum of Y error (m)
No control	-1.03	-0.01	-0.98	-0.02	-1.02	-0.04
PID control	-0.1	0.2	-0.09	0.15	-0.12	0.18
Cloud model control	-0.01	0.06	-0.02	0.08	-0.02	0.1

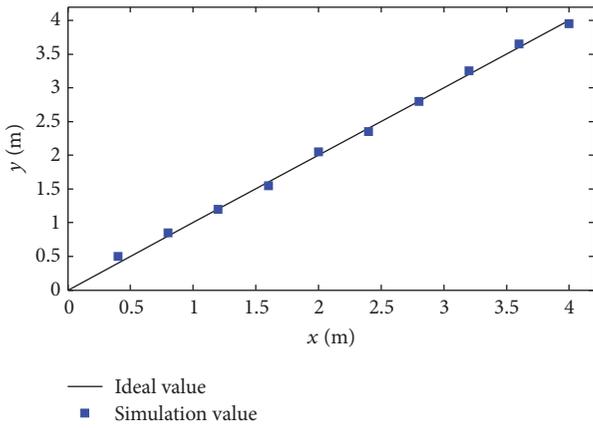


FIGURE 11: Straight path tracking.

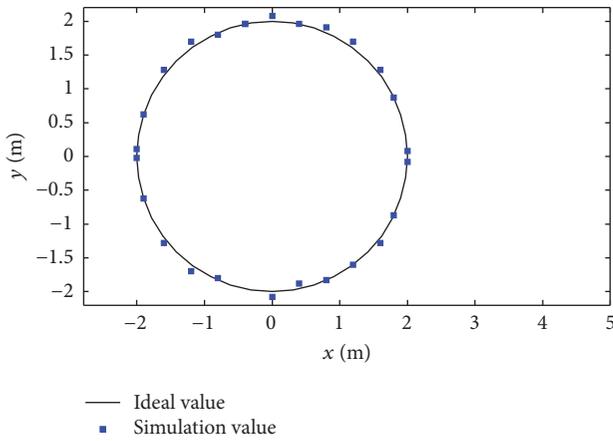


FIGURE 12: Circular path tracking.

and without controller. Four instantaneous motion pictures are shown in Figure 15.

Experimental process without controller: the wheel speed signal outputted by the wheel speed sensor in the ABS (antilock brake system) system is collected. The number of each square wave pulse signal outputted by the wheel is fixed in every circular; that is, the travelled distance (arc length) is certain. So, the number of pulses acquired can be used to calculate the distance the wheel travels (arc length). In the absence of control, there is a corresponding relationship between the distance travelled (arc length) by the wheel and

the steering angle when $D_{lateral}$ is fixed. Therefore, once the arc length of each stage is reached, the microcontroller outputs a fixed steering wheel angle to achieve the parking process. At this time, the trajectory can be calculated without controller.

The trajectory of the vehicle's motion, the tracking error of minimum of Y error, and the tracking error of maximum of Y error are shown in Figure 16. It can be observed that the model vehicle follows the specified autoparking path backwards to the target position with a small tracking error. The minimum and maximum of Y error tracking of cloud model controller are $-0.01, 0.06$; $-0.02, 0.08$; and $-0.02, 0.1$, respectively. It can also be observed from Table 2 that the dynamic control performance of the cloud model controller is better than that of PID controller. The trajectory without controller reflects the adverse impact of time delay on automatic parking. Simultaneously, it can be observed that the cloud model controller can be better to effectively reduce the adverse impact of time delay.

Because the control parameters of the PID controller are fixed, it is impossible to adjust quickly and rationally for the uncertainty of the time delay. The situation of lag and overshoot in the control process can be found. In the case of no controller, the problem of delay will further be aggravated in the process of automatic parking path tracking. The reason is that all the time delays effect will be superimposed, resulting in increasing error and even leading to parking accident finally. However, under the cloud model controller, because of the establishment of cloud model reasoning rule base and the combination of the one-dimensional and two-dimensional cloud model mapper can solve the adverse effect of time delay and improve the robustness of the system. Furthermore, cloud model controller can guarantee the stability of the system work effectively.

6. Conclusions

In this paper, the adverse impact of time delay on automatic parking is analyzed. In order to solve the uncertainty of time delays, uncertain cloud reasoning model is proposed. The precision of the position of the mapping point of midpoint at the rear axle on the ground is taken as the controlled target, and the deviation between the ideal steering wheel angle and actual steering angle is taken as the input. The single-rule reasoning of multidimensional cloud model is organically integrated to design a multirule tracking control model. According to the known path, the simulation experiment

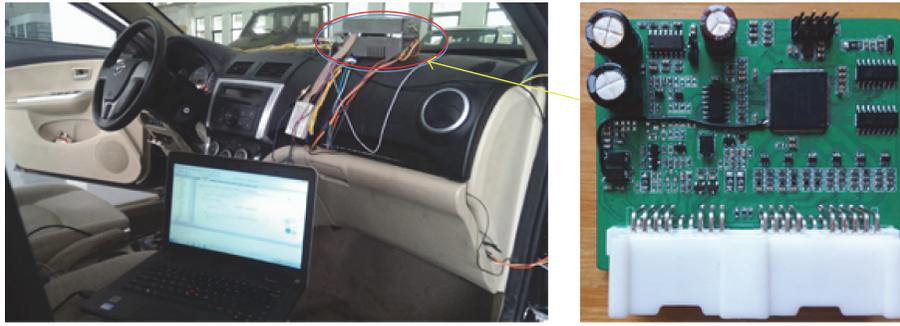


FIGURE 13: Parking controller hardware circuit board and integrated controller loading test chart.



FIGURE 14: Ultrasonic sensor placement in test vehicle. (a) Front of ultrasonic radar sensor layout. (b) Rear ultrasonic radar sensor layout. (c) Side ultrasonic radar sensor layout.

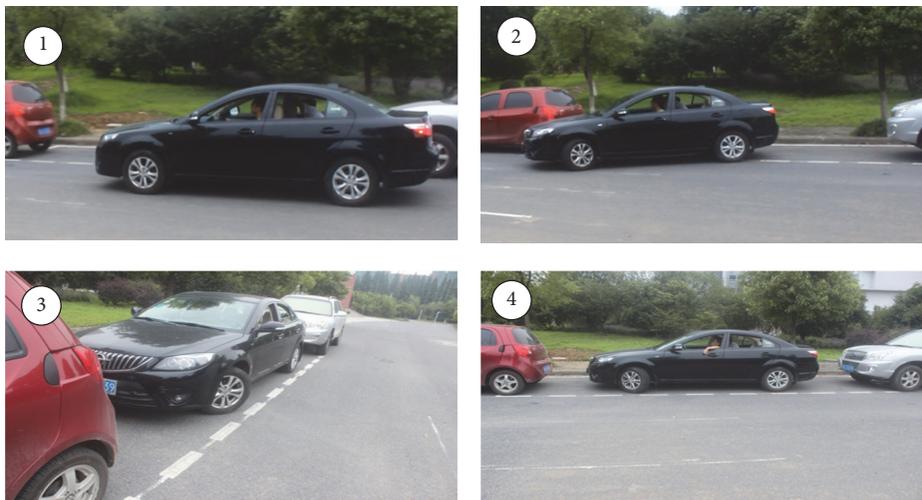


FIGURE 15: Actual test of the automatic parking system.

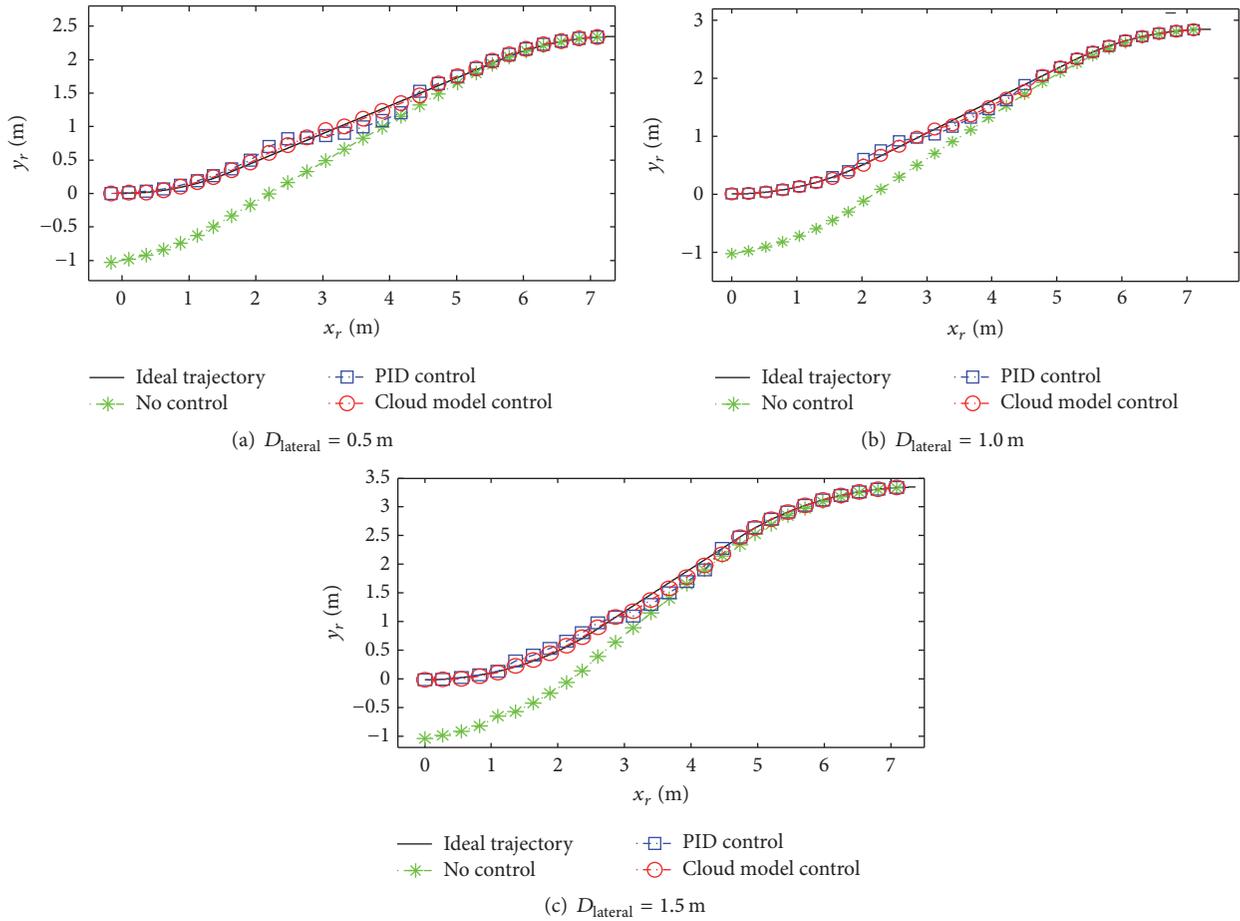


FIGURE 16: The test results of path tracking.

of the parking tracking control model is carried out under constant velocity. The results show that the tracking control model has a good tracking control performance and can quickly track the specified path. The real vehicle experimental results show that cloud model controller can achieve reasonable tracking control accuracy for this automatic parking system with uncertainty of time delays. Generally, the experimental results are better than traditional PID controller. Path tracking control of automatic parking system with variable speed will be the topic of future work.

Competing Interests

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

Acknowledgments

This work is financially supported by the National Natural Science Fund (nos. U1564201 and 51675235) and the CEEUSRO Innovative Capital Project in Science-Tech Bureau of Jiangsu Province (no. BY2012173).

References

- [1] B. Wang, C. Shao, J. Li, D. Zhao, and M. Meng, “Investigating the interaction between the parking choice and holiday travel behavior,” *Advances in Mechanical Engineering*, vol. 7, no. 6, pp. 1–11, 2015.
- [2] M. Roca-Riu, E. Fernández, and M. Estrada, “Parking slot assignment for urban distribution: models and formulations,” *Omega—International Journal of Management Science*, vol. 57, pp. 157–175, 2015.
- [3] X. Sun, L. Chen, H. Jiang, Z. Yang, J. Chen, and W. Zhang, “High-performance control for a bearingless permanent-magnet synchronous motor using neural network inverse scheme plus internal model controllers,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3479–3488, 2016.
- [4] X. Sun, L. Chen, and Z. Yang, “Overview of bearingless permanent-magnet synchronous motors,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5528–5538, 2013.
- [5] X. Sun, L. Chen, Z. Yang, and H. Zhu, “Speed-sensorless vector control of a bearingless induction motor with artificial neural network inverse speed observer,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1357–1366, 2013.
- [6] S. Yasunobu and Y. Murai, “Parking control based on predictive fuzzy control,” in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, pp. 1338–1341, June 1994.

- [7] M. Sugeno, T. Murofushi, T. Mori, T. Tatematsu, and J. Tanaka, "Fuzzy algorithmic control of a model car by oral instructions," *Fuzzy Sets and Systems*, vol. 32, no. 2, pp. 207–219, 1989.
- [8] R. E. Jenkins and B. P. Yuhua, "A simplified neural network solution through problem decomposition: the case of the truck backer-upper," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 718–720, 1993.
- [9] W. A. Daxwanger and G. K. Schmidt, "Skill-based visual parking control using neural and fuzzy networks," in *In Proceedings of the IEEE International Conference on Systems, Man, And Cybernetics 1995*, pp. 1659–1664, ancouver, Canada, October 1995.
- [10] A. Tayebi and A. Rachid, "Time-varying-based robust control for the parking problem of a wheeled mobile robot," in *Proceedings of the 13th IEEE International Conference on Robotics and Automation*, pp. 3099–3104, IEEE, Minneapolis, Minn, USA, April 1996.
- [11] M.-C. Pai, "RBF-based discrete sliding mode control for robust tracking of uncertain time-delay systems with input nonlinearity," *Complexity*, vol. 21, no. 6, pp. 194–201, 2016.
- [12] H. Wu, "Decentralised robust stabilisation of uncertain large-scale interconnected time-delay systems with unknown upper bounds of uncertainties," *International Journal of Systems Science*, vol. 47, no. 12, pp. 2816–2826, 2016.
- [13] B. Gutjahr and M. Werling, "Automatic collision avoidance during parking and maneuvering—an optimal control approach," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium (IV '14)*, pp. 636–641, Dearborn, Mich, USA, June 2014.
- [14] H. Choi and B. Song, "Fault detection and handling for sensor and communication faults of an automatic valet parking system," in *Proceedings of the 13th International Conference on Control, Automation and Systems (ICCAS '13)*, pp. 1338–1341, IEEE, Gwangju, Republic of Korea, October 2013.
- [15] B. Song, D. Kim, and H. Choi, "Cooperative lateral control for automatic valet parking," in *Proceedings of the 11th International Conference on Control, Automation and Systems (ICCAS '11)*, pp. 567–570, Gyeonggi-do, Korea, October 2011.
- [16] X. Miao and L. Li, "Adaptive observer-based control for uncertain nonlinear stochastic systems with time-delay," *Journal of the Franklin Institute*, vol. 353, no. 14, pp. 3595–3609, 2016.
- [17] Y. Batmani and H. Khaloozadeh, "On the design of suboptimal sliding manifold for a class of nonlinear uncertain time-delay systems," *International Journal of Systems Science*, vol. 47, no. 11, pp. 2543–2552, 2016.
- [18] H. Lv, Q. Zhang, and J. Ren, "Delay-dependent H_∞ control for a class of uncertain time-delay singular Markovian jump systems via hybrid impulsive control," *International Journal of Control, Automation and Systems*, vol. 14, no. 4, pp. 939–947, 2016.
- [19] Q. Zhang and P.-C. Li, "Adaptive grouping chaotic cloud model shuffled frog leaping algorithm for continuous space optimization problems," *Kongzhi yu Juece/Control and Decision*, vol. 30, no. 5, pp. 923–928, 2015.
- [20] B. Fu, D. G. Li, and M. K. Wang, "Review and prospect on research of cloud model," *Application Research of Computers*, vol. 28, no. 2, pp. 420–426, 2011.
- [21] M.-Z. Liu, X. Zhang, M.-X. Zhang, M.-G. Ge, and J. Hu, "Rescheduling decision method of manufacturing shop based on profit-loss cloud model," *Control and Decision*, vol. 29, no. 8, pp. 1458–1464, 2014.
- [22] Y.-J. Zhang, S.-F. Shao, and N. Julius, "Cloud hypermutation particle swarm optimization algorithm based on cloud model," *Pattern Recognition and Artificial Intelligence*, vol. 24, no. 1, pp. 90–96, 2011.
- [23] G.-W. Zhang, R. He, Y. Liu, D.-Y. Li, and G.-S. Chen, "Evolutionary algorithm based on cloud model," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 31, no. 7, pp. 1082–1090, 2008.
- [24] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—The business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.
- [25] C. Guang, B. Xiaoying, H. Xiaofei et al., "Cloud performance trend prediction by moving averages," *Journal of Frontiers of Computer Science and Technology*, vol. 6, no. 6, pp. 495–503, 2012.
- [26] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107–1117, 2013.
- [27] D. Xu, S. Yang, and H. Luo, "A fusion model for CPU load prediction in cloud computing," *Journal of Networks*, vol. 8, no. 11, pp. 2506–2511, 2013.

Research Article

Robust Visual Tracking Using the Bidirectional Scale Estimation

An Zhiyong,^{1,2} Guan Hao,¹ and Li Jinjiang²

¹Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 201203, China

²Key Laboratory of Intelligent Information Processing in Universities of Shandong, Shandong Institute of Business and Technology, Yantai 264005, China

Correspondence should be addressed to An Zhiyong; azytyut@163.com

Received 21 August 2016; Accepted 18 December 2016; Published 19 January 2017

Academic Editor: Francisco Valero

Copyright © 2017 An Zhiyong 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.

Object tracking with robust scale estimation is a challenging task in computer vision. This paper presents a novel tracking algorithm that learns the translation and scale filters with a complementary scheme. The translation filter is constructed using the ridge regression and multidimensional features. A robust scale filter is constructed by the bidirectional scale estimation, including the forward scale and backward scale. Firstly, we learn the scale filter using the forward tracking information. Then the forward scale and backward scale can be estimated using the respective scale filter. Secondly, a conservative strategy is adopted to compromise the forward and backward scales. Finally, the scale filter is updated based on the final scale estimation. It is effective to update scale filter since the stable scale estimation can improve the performance of scale filter. To reveal the effectiveness of our tracker, experiments are performed on 32 sequences with significant scale variation and on the benchmark dataset with 50 challenging videos. Our results show that the proposed tracker outperforms several state-of-the-art trackers in terms of robustness and accuracy.

1. Introduction

Visual tracking has drawn significant attentions in computer vision with various applications such as activity analysis, video surveillance, and auto control systems. Despite significant progress in recent years, it is still difficult due to baffling factors in complicated situations such as scale variations, partial occlusion, background clutter, deformation, and fast motion.

In recent years, many object tracking algorithms have been proposed. Among those trackers, the tracking-by-detection algorithms have achieved excellent performance by learning a discriminative classifier. Bolme et al. [1] presented a minimum output sum of squared error (MOSSE) filter on the gray images when initialized using a single frame. The MOSSE tracker is robust to variations in lighting, pose, and nonrigid deformations while running with a speed reaching several hundred frames per second. The MOSSE tracker can be performed efficiently because of using the fast Fourier transform (FFT). Recently, many new algorithms [2–5] based

on the MOSSE filters have been proposed in researches in field of object tracking.

Unlike tracking-by-detection algorithm, the TLD tracking framework [6] was presented that decomposed the tracking task into three subtasks: tracking, learning, and detection. The TLD tracker used a P-N learning algorithm to improve classification performance in long-term tracking. In recent years, sparse representation has been successfully applied to visual tracking. Zhang et al. [7] proposed the multitask tracking framework based on the particle filters and the multitask sparse representation. Wang et al. [8] proposed the WLCS tracker based on the particle filters and weighted local cosine similarity which measures the similarities between the target template and candidates. Guo et al. [9] proposed the max confidence boosting tracker that allows ambiguity in the tracking and effectively alleviates the drift problem. Zhang and Song [10] proposed the weighted multiple instance learning tracker that considers the sample importance into the tracker framework. Wang et al. [11] proposed the structure constrained grouping based on the

Bayesian inference framework which casts visual tracking as foreground superpixels grouping problem. However, those algorithms cannot handle the scale changes well in object tracking.

In this paper, we decomposed the tracking task into two subtasks: translation filter and scale filter with a complementary scheme. The translation filter relies on a temporal context regression model. As for scale filter, our key idea is using both the forward and backward tracking information to construct the scale filter. The information from the last frame to the current frame can be defined as the forward tracking information, while the information from the current frame to the last frame can be regarded as the backward tracking information. Therefore, the scale filter can be estimated by the bidirectional scale estimation, including the forward scale and the backward scale. The proposed approach achieves state-of-the-art performance on both scale variation dataset and the benchmark dataset.

2. Related Work

Our tracking algorithm is based on the correlation filter model. The classic correlation filter algorithm is the MOSSE tracker. The MOSSE tracker takes randomly affine-transformed ground truths as training set when initializing its correlation filter. However, in MOSSE tracker, the correlation filter is only used to detect the position of target.

Based on the correlation filter, Henriques et al. [2] proposed the CSK tracker algorithm that used the circulant matrices theory to learn a kernelized least-squares classifier. The CSK tracker runs at hundreds of frames per second. Furthermore, Henriques et al. [3] proposed the kernelized correlation filter (KCF) using the multichannel HOG features based on the CSK tracker. However, the CSK and KCF trackers imply the low accuracy in the scale variations sequences because the algorithms are limited to target translation. Li and Zhu [12] proposed the SAMF tracker that extends KCF to handle scale changes by sampling with several predefined scale perturbations. In fact, by sampling predefined scale variations, the SAMF tracker is not flexible enough to deal with fast and abrupt scale changes.

Zhang et al. [4] proposed a kernelized correlation filter to predict the target scale variation that often produces the inaccurate scale estimation in a complicated environment. Danelljan et al. [5] proposed the discriminative scale space tracker (DSST) that learnt the translation filter and scale filter for the tracking target. The scale filter in DSST only uses the forward tracking information to search the reasonable scale while neglecting the backward information. This makes the scale unstable usually in complex scenes.

3. Our Approach

Here we describe our approach. Section 3.1 presents translation tracking filter. In Section 3.2 we describe scale tracking filter using the bidirectional scale estimation.

3.1. Translation Tracking Filter. The MOSSE tracker only uses the image intensity features to design the translation filter.

Since the HOG feature is a strong local analysis feature, it has been widely used in visual tracking. So we use both the HOG feature and intensity feature to build the translation filter. Here we build the translation filter using the ridge regression similar to the DSST and MOSSE. Given n features $f = \{f_1, f_2, \dots, f_i, f_n\}$, the translation tracking filter h^i is obtained by minimizing the sum of squared errors with the regularization term

$$\min \left\| \sum_{i=1}^n f_i * h^i - g \right\|^2 + \lambda \sum_{i=1}^n \|h^i\|^2, \quad (1)$$

where g is the desired correlation outputs. Here g can be designed as the 2D Gaussian shaped peak centered on the target in training image. In this paper, the upper case variables F_i, G and the filter H^i denote the Fourier transform of their lower case counterparts separately. The solution to (1) in the Fourier transform domain is [5]

$$H^i = \frac{\bar{G} \odot F_i}{\sum_{i=1}^n F_i \odot \bar{F}_i + \lambda}, \quad (2)$$

where \odot is the element-wise product and the bar \bar{G} represents complex conjugation. To obtain a robust approximation, the translation filter H_t^{trans} can be updated by the numerator $H_{t,1}^i$ and denominator $H_{t,2}$.

$$H_{t,1}^i = \eta H_{t-1,1}^i + (1 - \eta) \bar{G} \odot F_i \quad (3)$$

$$H_{t,2} = \eta H_{t-1,2} + (1 - \eta) \sum_{i=1}^n F_i \odot \bar{F}_i, \quad (4)$$

where t is the index of frame and η is the learning rate. Given the features F^Z of an image patch Z in the new frame, the confidence score can be calculated as

$$y = \frac{\sum_{i=1}^n \bar{H}_{t,1}^i \odot F^Z}{H_{t,2} + \lambda}. \quad (5)$$

So the new target location can be estimated by searching for the location of the maximum confidence score.

3.2. Scale Tracking Filter. In this section, a 1-dimensional correlation filter is constructed to estimate the target scale. The purpose of scale tracking filter is to locate the appropriate scale for the target in the current frame. Let $W \times R$ be the target size and N is the number of scales $S = \{a^n \mid n \in [-(N-1)/2], \dots, [(N-1)/2]\}$, where a denotes the scale factor. For each $a^n \in S$, we extract an image patch of size $a^n W \times a^n R$ centered around the estimated location. At the same time, we use the HOG features to construct the scale pyramid.

Here the training features at each scale level are set to a k -dimensional feature descriptor. Given the training samples $f^S = \{f_1^S, \dots, f_j^S, \dots, f_k^S\}$, the scale filter h^j is obtained by minimizing the sum of squared errors with the regularization term

$$\min \left\| \sum_{j=1}^k f_j^S * h^j - g \right\|^2 + \lambda \sum \|h^j\|^2, \quad (6)$$

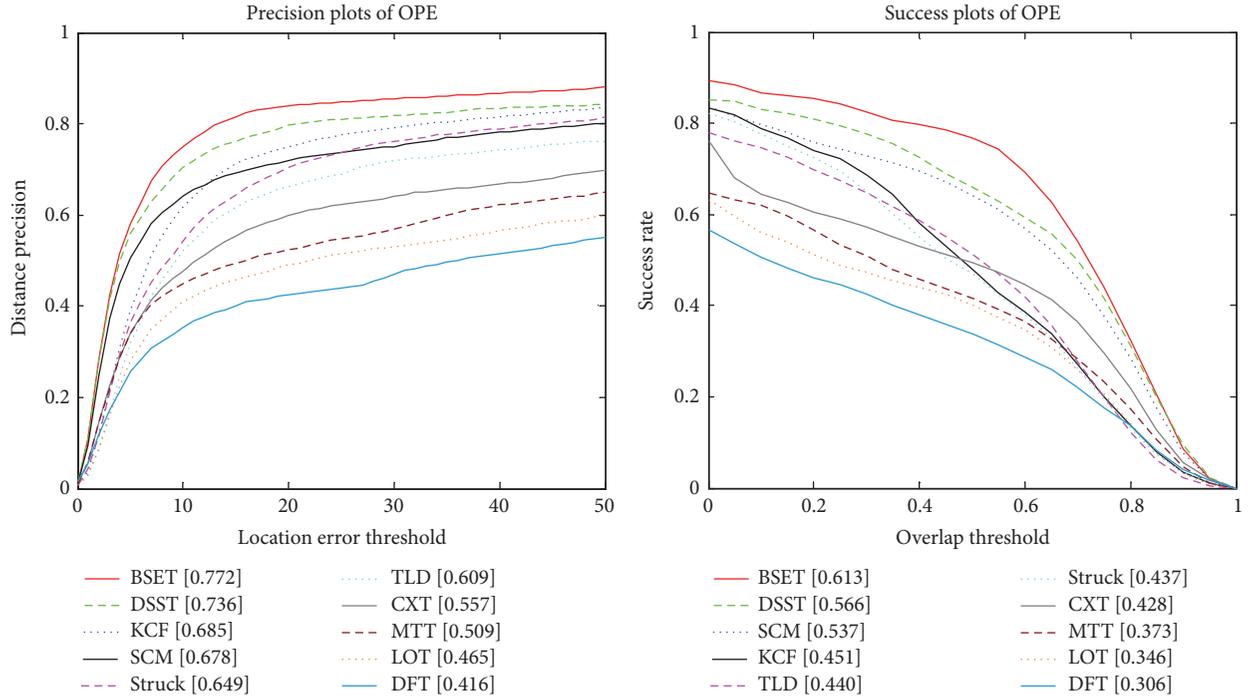


FIGURE 1: Precision and success plots over all the 32 sequences.

where g is as the 1-dimensional Gaussian shaped peak centered around the current scale. Then the solution to (6) in the Fourier transform domain is

$$H^j = \frac{\overline{G} \odot F_j^S}{\sum_{j=1}^k F_j^S \odot \overline{F_j^S} + \lambda}, \quad (7)$$

where the upper case variables F_j^S , \overline{G} and the filter H^j denotes the Fourier transform of their lower case counterparts separately. Assume the scale filter H_{t-1}^{scale} in the last frame be expressed by the numerator A_{t-1}^j and denominator B_{t-1} . Given the features F^t of an image patch in the current frame, the maximum scale confidence score can be calculated as

$$y_t^F = \max \left(\frac{\sum_{j=1}^k \overline{A_{t-1}^j} \odot F^t}{B_{t-1} + \lambda} \right), \quad (8)$$

where t is the index of frame. So the new target scale can be estimated by searching for the scale of the maximum confidence score. Let $S^F = c \times S_{t-1}$ denote the corresponding scale to y_t^F . Here S_{t-1} denotes the target scale in the last frame and c denotes the relevant factor between the new target scale in the current frame and the target scale in the last frame. Then the scale S^F can be set to the forward scale for the new target since it uses the last frame's filter H_{t-1}^{scale} to estimate the target scale in the current frame.

In order to improve the performance, we compute a backward scale for the same target. Based on the new target location and the forward scale S^F , we can obtain a new scale

filter H_{s^F} that includes the numerator A^j and denominator B . It is noted that the scale filter H_{s^F} only uses the new target information in the current frame. Given the features F^{t-1} of an image patch in the last frame, the max confidence score can be calculated as

$$y_{t-1}^I = \max \left(\frac{\sum_{j=1}^k \overline{A^j} \odot F^{t-1}}{B + \lambda} \right). \quad (9)$$

Let $S_{t-1}^I = d \times S^F$ denote the corresponding scale to y_{t-1}^I in the last frame and let d denote the backward relevant factor. Then the new target scale can be expressed by $S^I = S_{t-1}^I / d$. The scale S^I can be set to the backward scale estimation since it uses the current frame's filter H_{s^F} to estimate scale.

In most cases, the backward scale S^I is equal to forward scale S^F . However, there are always estimating errors for the backward and forward scales in practical situation. Here the trend of scale change can be defined as two types: increase and decrease. When the scale value is more than 1, the scale trend is regarded as the increase. While the scale value is less than 1, we can assume that the scale trend is the decrease. The forward scale's trend is often inconsistent with the backward scale's trend in some situations. The main reason is that the scale filter H_{t-1}^{scale} includes the history information while the scale filter H_{s^F} only uses the current scale information. In fact, the scale filter H_{t-1}^{scale} with the history information can produce inaccurate trend estimates for scale. Thus, we should set the backward scale to the final scale when the forward scale's trend is inconsistent with the backward scale's trend.



FIGURE 2: A visualization of the tracking results of five visual trackers on challenging sequences: car24, carscale, human5, RedTeam, and skating1.

So a conservative strategy is adopted to integrate two above scales

$$S_t = \begin{cases} S^I, & \text{if } S^F < 1 \ \&\& S^I \geq 1 \\ S^I, & \text{if } S^F \geq 1 \ \&\& S^I < 1 \\ S^F, & \text{else.} \end{cases} \quad (10)$$

When the forward scale's trend is identical to the backward scale's trend, we can set the forward scale to the final scale

since the filter H_{t-1}^{scale} includes abundant scale information. So the scale filter can be updated by the numerator A_t^j and denominator B_t

$$A_t^j = \eta A_{t-1}^j + (1 - \eta) F_j^t \odot \bar{G} \quad (11)$$

$$B_t = \eta B_{t-1} + (1 - \eta) \sum_{j=1}^k F_j^t \odot \bar{F}_j^t, \quad (12)$$

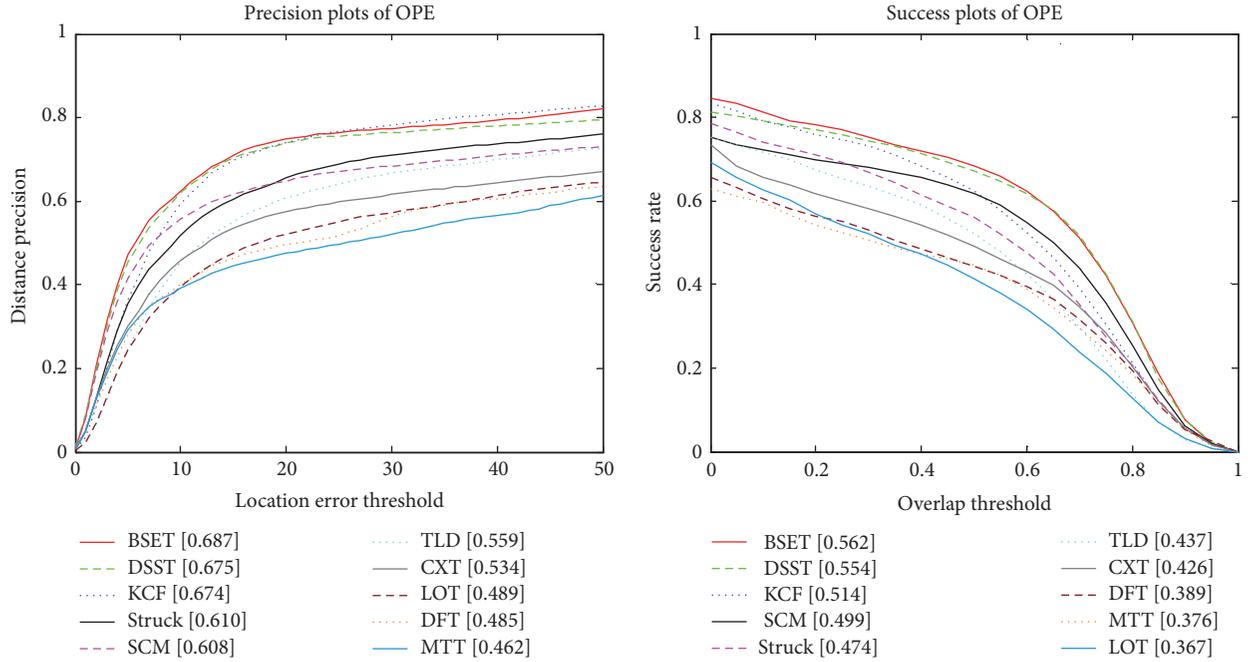


FIGURE 3: Evaluation on OTB-50 dataset.

where η is the learning rate. It is noted that the features F_j^t in (11) and (12) are based on the final scale S_t . It is effective to update the scale filter since the accurate scale S_t can improve the performance of scale filter to a great extent. We present an outline of our method in Algorithm 1.

Algorithm 1 (proposed tracking algorithm).

Input.

Image I_t

Previous target position and scale $(x_{t-1}, y_{t-1}, S_{t-1})$.

Translation model H_{t-1}^{trans} : the numerator $H_{t-1,1}^i$ and denominator $H_{t-1,2}$.

Scale model H_{t-1}^{scale} : the numerator A_{t-1}^j and denominator B_{t-1} .

Output.

Estimated target position and scale (x_t, y_t, S_t) .

Updated translation model $H_t^{\text{trans}}(H_{t,1}^i, H_{t,2})$ and scale model $H_t^{\text{scale}}(A_t^j, B_t)$.

Translation Estimation.

(1) Extract the target features at the previous target position (x_{t-1}, y_{t-1}) and scale S_{t-1} from I_t .

(2) Compute the translation filter H^{trans} using (2).

(3) Set (x_t, y_t) to the target position that maximizes the confidence score in (5).

Scale Estimation.

(4) Extract the target features around from I_t at target position (x_t, y_t) and scale S_{t-1} .

(5) Compute the forward scale S_t^F using the scale filter H_{t-1}^{scale} .

(6) Compute the scale filter H_{s^F} and the backward scale S_t^I .

(7) Integrate the forward scale S_t^F and the backward scale S_t^I as the final scale S_t using (10).

Model Update.

(8) Extract the target features from I_t at the target position (x_t, y_t) and scale S_t .

(9) Update the translation model H_t^{trans} : updated by the numerator $H_{t,1}^i$ in (3) and denominator $H_{t,2}$ in (4).

(10) Update the scale model H_t^{scale} : updated by the numerator A_t^j in (11) and denominator B_t in (12).

4. Experiments

We name our proposed tracker ‘‘BSET’’ (*Bidirectional Scale Estimation Tracker*). All trackers in this paper are implemented in Matlab 2013 on an Intel I5-3210 2.50 GHz CPU with 4 GB RAM. The regularization parameter λ is set to 0.01 and the learning rate η is set to 0.025 in formulas. The number of scales is set to 33 with a self-adaptive scale factor. Given a target of size $M \times N$, the self-adaptive scale factor can be set to

$$a = \begin{cases} 1.04, & \text{if } \min(M, N) \leq 20 \\ 1.02, & \text{if } \max(M, N) \geq 100 \\ 1.03, & \text{else.} \end{cases} \quad (13)$$

TABLE 1: Per-video overlap precision (OP) (%) on the 32 sequences. “*” indicates the best performance, “**” indicates the second best ones, and “#” indicates the third best ones.

	BSET	DSST	KCF	Struck	SCM	TLD	MTT	LOT	DFT	CXT
<i>biker</i>	47.2*	26.8	25.4	25.4	44.4#	31.0	31.0	43.0	25.4	46.5**
<i>blurCar2</i>	100.0*	100.0*	94.7	93.8	14.4	99.8**	13.3	28.7	17.4	95.0#
<i>boy</i>	100.0*	100.0*	99.2**	97.5#	43.9	93.5	48.7	65.0	48.3	49.7
<i>car24</i>	100.0*	17.3	17.3	17.0	100.0*	48.3**	100.0*	37.2#	7.2	100.0*
<i>car4</i>	100.0*	100.0*	36.4	39.8	97.3**	79.2#	31.1	4.9	25.8	29.9
<i>carScale</i>	88.5*	84.5**	44.4	43.3	65.1	43.7	56.7	46.4	44.8	78.2#
<i>couple</i>	10.7	10.7	24.3	54.3	10.7	100.0*	61.4**	56.4#	8.6	56.4#
<i>crossing</i>	100.0*	100.0*	95.0**	94.2#	100.0*	51.7	22.5	30.8	64.2	34.2
<i>dog1</i>	100.0*	100.0*	65.1	65.3	84.7	67.3	78.5	98.6#	52.1	99.8**
<i>doll</i>	99.7*	99.7*	55.2	68.8	98.7**	62.4	50.4	87.1	35.0	97.5#
<i>dudek</i>	99.0*	98.1**	97.6	98.0#	97.6	84.2	92.9	61.8	80.1	92.4
<i>fleetface</i>	70.0**	66.5	66.9#	66.6	70.6*	56.7	54.6	57.9	55.6	64.6
<i>freeman1</i>	77.3**	35.3	16.3	21.5	80.7*	21.2	16.0	18.7	17.8	25.8#
<i>freeman3</i>	60.0#	31.3	27.8	20.0	93.0**	58.0	47.8	7.4	33.0	93.7*
<i>freeman4</i>	16.6	41.7*	18.4	15.5	24.4#	26.9**	18.0	13.4	18.0	18.0
<i>girl</i>	87.4	30.6	74.2	98.0*	88.2#	76.4	93.2**	54.6	25.2	64.2
<i>Human5</i>	99.7*	24.3	23.6	34.1	41.5	51.2**	35.1	42.5#	7.6	28.1
<i>Human8</i>	100.0*	100.0*	30.5#	13.3	100.0*	13.3	11.7	92.2**	13.3	10.9
<i>ironman</i>	34.9*	13.3#	15.1**	4.8	13.3#	6.6	8.4	8.4	3.6	3.0
<i>liquor</i>	40.9	40.9	98.1*	40.6	32.1	58.2#	20.1	96.3**	22.9	21.0
<i>matrix</i>	18.0#	18.0#	13.0	12.0	30.0*	7.0	29.0**	4.0	6.0	4.0
<i>RedTeam</i>	96.2*	70.0**	37.6	39.8	39.1	28.3	51.4#	10.1	27.0	45.3
<i>shaking</i>	100.0*	100.0*	1.4	16.7	89.6**	40.0	1.1	7.7	82.5#	10.7
<i>singer1</i>	27.6	100.0*	27.6	29.9	100.0*	99.1**	34.5#	24.2	27.6	32.2
<i>skating1</i>	98.0*	52.3**	36.3	37.0	42.3#	22.8	13.5	28.0	16.3	12.0
<i>soccer</i>	14.5	38.8**	39.3*	15.6	23.7#	12.2	18.1	21.7	21.9	12.8
<i>surfer</i>	89.4#	29.0	39.9	15.7	40.7	89.9**	5.6	37.2	3.7	96.3*
<i>toy</i>	97.0**	90.0#	43.2	49.1	22.1	73.4	53.1	21.4	46.9	97.4*
<i>trellis</i>	96.8**	97.7*	84.0	78.4	85.4#	47.3	19.3	31.6	51.8	80.8
<i>walking</i>	99.8*	99.8*	51.5	56.6	95.9#	38.3	95.9#	96.8**	55.1	21.8
<i>walking2</i>	100.0*	100.0*	38.0	43.4#	100.0*	34.0	99.2**	39.0	38.2	39.8
<i>woman</i>	88.3	93.3#	93.6*	93.5**	85.8	16.6	19.8	8.4	93.5**	20.6
<i>Average OP</i>	76.8*	65.9**	47.8	46.8	64.2#	51.2	41.6	40.0	33.6	49.5

The strategy can adjust the scale factor with the size of target adaptively and overcomes the problem of scale slow increase for the small target. When the size of target is large, we should set a small value to the scale factor since this can adapt the scale change responsively. To assess the overall performance of BSET, a large benchmark dataset (OTB-50) [13] is adopted that contains 50 videos with many challenging attributes such as scale variation, in-plane rotation, low resolution, and background clutter.

We adopt the 32 sequences annotated with “scale variation” as the scale variation dataset. Two criteria, the center location error (CLE) as well as the overlap precision, are employed to the scale dataset in our paper. The CLE can be defined as the average Euclidean distance between the

ground truth and the estimated center location of the target. Overlap precision (OP) [5] is defined as the percentage of frames where the bounding box overlap surpasses a threshold $t \in [0, 1]$. We compare our method with the nine state-of-the-art trackers: DSST [5], KCF [3], Struck [14], SCM [15], TLD [6], MTT [7], LOT [16], DFT [17], and CXT [18], which have been shown to provide excellent performance in literatures. In addition, we provide two kinds of plots: precision plot and success plot [13] to evaluate all trackers, where trackers are ranked using the area under curve (AUC).

4.1. Experiment 1: Robust Scale Estimation. The scale variation dataset includes the 32 sequences and those sequences

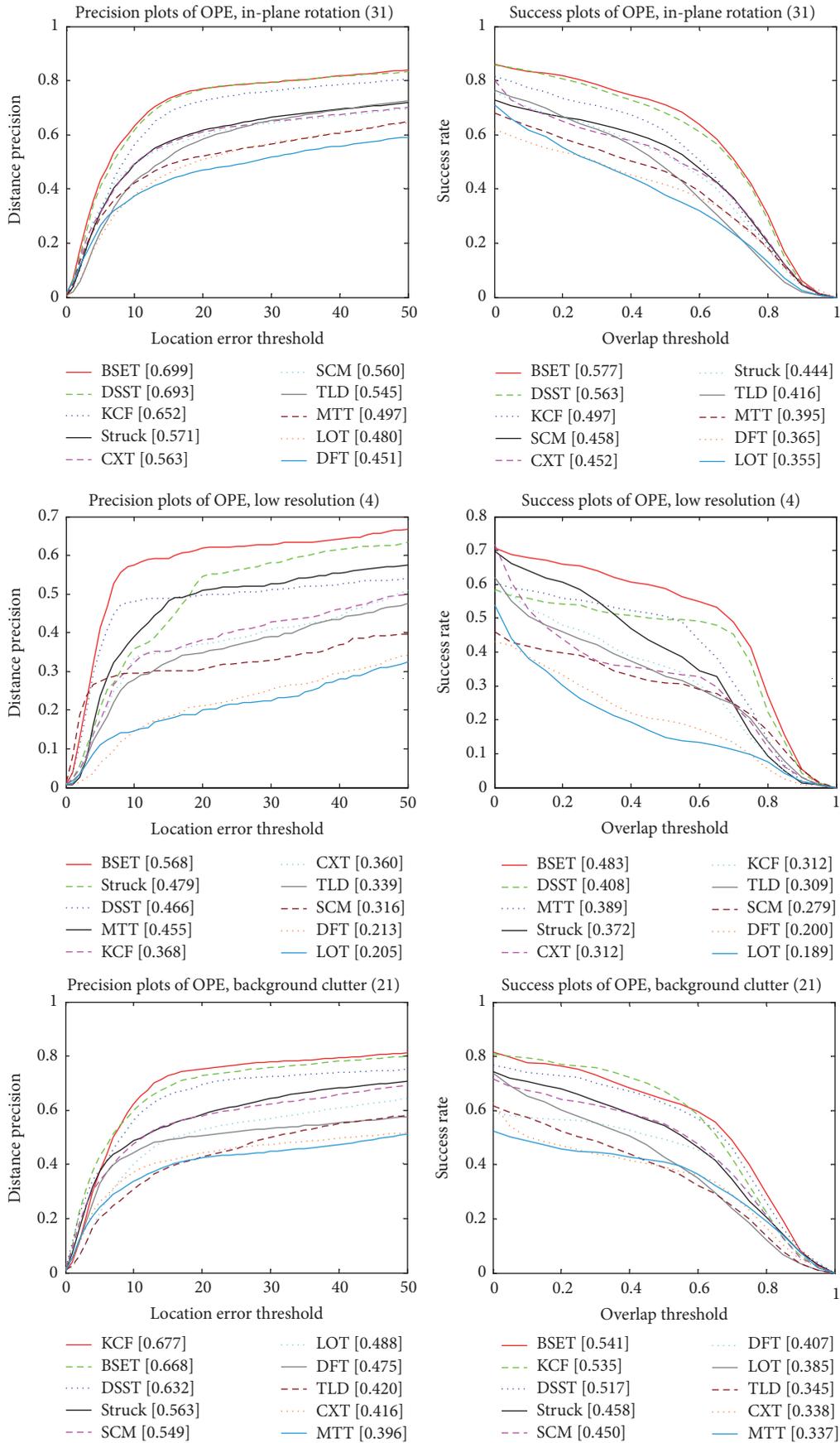


FIGURE 4: Evaluation on three tracking challenges of in-plane rotation, low resolution, and background clutter.

TABLE 2: Per-video CLE (in pixel) on the 32 sequences. “*” indicates the best performance, “**” indicates the second best ones, and “#” indicates the third best ones.

	BSET	DSST	KCF	Struck	SCM	TLD	MTT	LOT	DFT	CXT
biker	35.4[#]	68.9	71.2	25.0^{**}	72.6	66.4	76.8	69.3	74.7	20.0[*]
blurCar2	2.8[*]	2.9^{**}	6.8	10.4	125.6	6.5[#]	141.8	53.6	152.4	7.2
boy	1.9[*]	2.1^{**}	2.9[#]	3.8	51.0	4.5	12.8	66.0	106.3	7.4
car24	1.7^{**}	1.6[*]	4.1	119.7	1.9^{**}	3.0	2.6	100.0	165.6	3.5
car4	1.7[*]	1.9^{**}	9.9	8.7	4.3^{**}	12.8	22.3	167.3	61.9	58.1
carScale	18.5^{**}	19.1[#]	16.1[*]	36.4	33.4	22.6	87.6	101.2	75.8	24.5
couple	129.9	125.1	47.6	11.3^{**}	109.6	2.5[*]	27.8[#]	37.1	108.6	41.8
crossing	1.3[*]	1.5^{**}	2.2	2.8	1.6[#]	24.3	57.1	36.7	22.3	23.4
dog1	3.9[*]	4.3[#]	4.2^{**}	5.7	7.0	4.2^{**}	4.3[#]	4.6	41.2	4.9
doll	2.4[*]	3.0^{**}	8.4	8.9	3.4[#]	6.0	110.3	6.3	59.5	4.7
dudek	13.4	13.3	11.4^{**}	11.4^{**}	10.8[*]	18.1	14.1	85.1	18.7	12.8[#]
fleetface	27.6[#]	33.6	26.4^{**}	23.0[*]	27.6[#]	41.2	69.1	33.7	68.0	45.1
freeman1	9.1[#]	7.6^{**}	94.9	14.3	6.9[*]	39.7	117.8	86.9	10.4	26.8
freeman3	5.3[#]	16.6	19.3	16.8	3.2[*]	29.3	15.6	40.5	32.6	3.6^{**}
freeman4	57.4	20.0[*]	27.1[#]	48.7	37.7	39.2	23.5^{**}	38.6	57.5	65.6
girl	5.0[#]	11.0	11.9	2.6[*]	2.6[*]	9.8	4.3^{**}	22.8	24.0	11.0
human5	5.0[*]	302.5	175.5	6.9[#]	9.3	5.3^{**}	8.3	90.3	259.1	200.9
human8	2.2^{**}	2.4[#]	3.8	63.8	1.9[*]	66.0	76.0	4.0	73.8	67.3
ironman	53.8[*]	205.9	194.9	127.7	163.5	93.2^{**}	215.3	98.7[#]	239.7	162.7
liquor	98.4	98.5	5.3[*]	91.0	99.2	37.6[#]	543.4	8.5^{**}	221.1	131.8
matrix	83.0	70.2[#]	76.4	194.5	48.2[*]	57.2^{**}	75.4	73.5	105.8	151.6
RedTeam	2.5[*]	2.8[#]	3.8	4.3	4.1	35.5	2.7^{**}	71.9	50.3	16.8
shaking	7.6[*]	8.2^{**}	112.5	30.7	11.0[#]	37.1	97.9	82.6	26.3	129.2
singer1	8.5	3.5^{**}	12.8	14.5	2.7[*]	8.0[#]	36.2	141.4	18.8	11.4
skating1	6.9[*]	8.4[#]	7.7^{**}	82.9	16.4	145.5	293.3	110.5	174.2	129.8
soccer	13.0[*]	20.3[#]	15.4^{**}	71.4	77.6	77.1	84.2	42.2	139.5	89.2
surfer	4.0[#]	20.1	8.7	9.0	14.8	3.8^{**}	37.0	26.1	150.9	3.1[*]
toy	8.4	7.9[#]	7.8^{**}	11.4	50.8	8.9	12.8	53.0	31.9	6.1[*]
trellis	2.7^{**}	2.6[*]	7.8	6.9[#]	7.0	31.1	68.8	47.7	44.9	7.0
walking	2.5[#]	1.7[*]	4.0	4.6	2.5[#]	10.2	3.5	2.4^{**}	5.9	205.7
walking2	3.0^{**}	3.2[#]	29.0	11.2	1.7[*]	44.6	4.0	64.9	29.1	34.7
woman	10.6	9.7	10.1	4.2[*]	7.9^{**}	139.9	137.3	117.1	8.5[#]	72.5
Average CLE	19.7[*]	34.4	32.5[#]	33.9	31.8^{**}	35.3	77.6	62.0	83.1	55.6

also have challenging problems such as illumination variation, motion blur, background clutter, and occlusion. Table 1 shows the Per-video overlap precision at a threshold 0.5 compared with 9 state-of-the-art trackers. From Table 1, we find that the BSET algorithm provides better or similar performance on 18 out of the 32 sequences. The BSET algorithm performs well with an average OP of 0.768, which outperforms the DSST algorithm by 10.9%. Table 2 shows Per-video CLE compared with 9 state-of-the-art trackers. From Table 2, we find that the BSET algorithm provides better or similar performance on 12 out of the 32 sequences. The SCM tracker performs well with average CLE of 31.8 pixels. But the BSET tracker achieves lower average CLE of 19.7 pixels.

In addition, we report precision plots and success plots of OPE in Figure 1. In precision and success plots, our approach tracks scale more accurately on the scale variation dataset than other state-of-the-art trackers. This also indicates that using bidirectional scale estimation can improve the accuracy of scale estimation. In the precision plot, the precision score of the BSET algorithm is 0.772, which outperforms the DSST algorithm by 3.6%. Meanwhile, in the success plot, the proposed BSET algorithm achieves the AUC of 0.613, which outperforms the DSST algorithm by 4.7%. Figure 2 shows a visualization of the tracking results of our approach and the state-of-the-art visual trackers DSST [5], KCF [3], Struck [10], and TLD [6] on five challenging sequences: car24,

carscale, human5, RedTeam, and skating1. Our algorithm provides promising results compared to existing trackers on these sequences. Our algorithm performs at 13.7 frames per second on scale variation dataset that basically meets the requirement of engineering applications.

4.2. Experiment 2: Evaluation on Benchmark Dataset. Here we evaluate all the trackers on the OTB-50 dataset that includes 50 sequences (including 51 tracking targets). Resultant precision plots and success plots are shown in Figure 3, which shows our tracker is superior comparing to state-of-the-art trackers on the benchmark dataset. Among the trackers, the DSST method achieves the precision score of 0.675 and success score of 0.554 while the BSET algorithm achieves the precision score of 0.687 and success score of 0.562.

In addition, we report results for three challenging attributes in Figure 4. In the precision plot, the proposed BSET algorithm outperforms the DSST and KCF algorithms in in-plane rotation and low resolution. Meanwhile, in background clutter, the precision score of the BSET algorithm is 0.668 which is close to the KCF 0.677. In the success plot, the DSST method performs with overall success in in-plane rotation (0.563) and low resolution (0.408) while the BSET algorithm achieves success rate of 0.577 and 0.483, respectively. The KCF method performs well with overall success in background clutter (0.535) while the BSET algorithm achieves the success rate of 0.541. However, the BSET algorithm cannot handle well the fully occluded (e.g., freeman4). The main reason is that the accuracy of the backward scale will be poor in the fully occluded. In summary, the BSET algorithm adapts to scale variation, in-plane rotation, low resolution, and background clutter.

5. Conclusion

In this paper, we propose an effective tracking algorithm using the bidirectional scale estimation in a tracking-by-detection framework. We decomposed the tracking task into two subtasks: translation filter and scale filter to estimate the target objects. The translation filter is constructed using the ridge regression and multidimensional features. The scale filter is constructed by the bidirectional scale estimation, including the forward and the backward scales. Experimental results show that our proposed tracker performs favorably against several state-of-the-art trackers on the scale variation dataset.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors are grateful to the support by National Natural Science Foundation of China (61373079, 61472227) and National Natural Science Foundation of Shandong Province (ZR2013FL018, ZR2015FL019, and ZR2014FL007).

References

- [1] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2544–2550, San Francisco, Calif, USA, June 2010.
- [2] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision (ECCV '12)*, pp. 702–715, Florence, Italy, October 2012.
- [3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.
- [4] L. Zhang, Y. Wang, H. Sun, Z. Yao, and S. He, "Robust visual correlation tracking," *Mathematical Problems in Engineering*, vol. 2015, Article ID 238971, 13 pages, 2015.
- [5] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the 25th British Machine Vision Conference (BMVC '14)*, pp. 1–11, Linköping, Sweden, September 2014.
- [6] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [7] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 367–383, 2013.
- [8] D. Wang, H. Lu, and C. Bo, "Visual tracking via weighted local cosine similarity," *IEEE Transaction on Systems, Man, and Cybernetics Part B*, vol. 45, no. 9, pp. 1838–1850, 2015.
- [9] W. Guo, L. Cao, T. X. Han, S. Yan, and C. Xu, "Max-confidence boosting with uncertainty for visual tracking," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1650–1659, 2015.
- [10] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.
- [11] L. Wang, H. Lu, and D. Wang, "Visual tracking via structure constrained grouping," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 794–798, 2015.
- [12] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision—ECCV 2014 Workshops*, vol. 8926 of *Lecture Notes in Computer Science*, pp. 254–265, Springer, 2015.
- [13] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [14] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, Barcelona, Spain, November 2011.
- [15] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1838–1845, Providence, RI, USA, June 2012.
- [16] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR '12), pp. 1940–1947, June 2012.

- [17] L. Sevilla-Lara and E. Learned-Miller, “Distribution fields for tracking,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1910–1917, Providence, RI, USA, June 2012.
- [18] T. B. Dinh, N. Vo, and G. Medioni, “Context tracker: exploring supporters and distracters in unconstrained environments,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1177–1184, Colorado Springs, Colo, USA, June 2011.

Research Article

Parameter Selection for Ant Colony Algorithm Based on Bacterial Foraging Algorithm

Peng Li and Hua Zhu

School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Hua Zhu; zhuhua83591917@163.com

Received 8 October 2016; Revised 28 November 2016; Accepted 6 December 2016

Academic Editor: Francisco Valero

Copyright © 2016 P. Li and H. Zhu. 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 optimal performance of the ant colony algorithm (ACA) mainly depends on suitable parameters; therefore, parameter selection for ACA is important. We propose a parameter selection method for ACA based on the bacterial foraging algorithm (BFA), considering the effects of coupling between different parameters. Firstly, parameters for ACA are mapped into a multidimensional space, using a chemotactic operator to ensure that each parameter group approaches the optimal value, speeding up the convergence for each parameter set. Secondly, the operation speed for optimizing the entire parameter set is accelerated using a reproduction operator. Finally, the elimination-dispersal operator is used to strengthen the global optimization of the parameters, which avoids falling into a local optimal solution. In order to validate the effectiveness of this method, the results were compared with those using a genetic algorithm (GA) and a particle swarm optimization (PSO), and simulations were conducted using different grid maps for robot path planning. The results indicated that parameter selection for ACA based on BFA was the superior method, able to determine the best parameter combination rapidly, accurately, and effectively.

1. Introduction

In the 1990s, Dorigo et al. [1] were inspired by the foraging behavior of ants and proposed an ant colony algorithm (ACA) that had the characteristics of strong robustness, a high degree of parallelism, and positive feedback. The ACA demonstrates high effectiveness and superiority in global optimization and in solving the traveling salesman [2, 3], shop scheduling [4–7], and robot path planning [8–10] problems.

However, the optimal performance and efficiency of the ACA are closely related to the chosen parameters that include the information heuristic factor α , the expectation heuristic factor β , the pheromone evaporation factor ρ , the pheromone strength Q , and the number of ants M [11]. Parameter selection differs for different types of optimization problems. Moreover, even for the same type of optimization problem, parameters may change due to the different scales of the problem. Traditional parameter selection methods for the ACA include the empirical selection method, the trial and error method, and the orthogonal experimental design method [12]. The empirical selection method requires a priori knowledge of the problem in question, and the

method requires the need to conduct multiple experiments, consuming considerable time and effort, rendering it difficult to obtain the optimum parameters. Hei and Du [13] applied the trial and error method, which required a large number of digital simulations to obtain the parameters, and the authors determined that it was difficult to guarantee that the obtained parameters were optimal. Gan and Li [14] studied parameter selection for ACA based on the orthogonal experimental design method, which ignores coupling between the parameters. This method is computationally intensive, which is a disadvantage. Traditional parameter selection methods for the ACA are mainly based on a large number of replicated experiments, which is inefficient and time consuming. Moreover, traditional parameter selection methods ignore the coupling between the parameters, making it difficult to obtain the optimal parameters.

In the ACA, the parameters are coupled with each other, and it is difficult to achieve the best performance for the algorithm by adjusting only a single parameter. Therefore, it is common to adjust multiple parameters simultaneously to determine the optimal parameter combination. Recently, scholars who have considered the parameter selection in

the ant colony algorithm as an optimization problem have applied intelligent algorithms to solve the parameter selection problem in the ACA. Feng [15] proposed a parameter selection method based on a GA. Although this method is effective with regard to global optimization, it had the disadvantage of slow convergence speed. Li et al. [16] applied a PSO, which can avoid subjectivity in parameter selection and obtain optimum parameters rapidly. In spite of that, this method is prone to premature convergence and falling into a local optimal solution. Although GA and PSO are suitable for obtaining ACA parameters quickly and accurately, due to the disadvantages of the methods, the parameter selection can be improved upon. Therefore, a suitable intelligent algorithm that can optimize globally is required to obtain the optimal parameters for the ACA.

The bacterial foraging algorithm (BFA) [17] has a fast convergence speed and searches objects in parallel. The algorithm includes a chemotaxis operator, a reproduction operator, and an elimination-dispersal operator. In addition, the BFA is able to jump out of the local optimal solution. Therefore, the BFA has shown good adaptability for solving the problems of job shop scheduling [18], robot path planning [19], image processing [20], and high dimensional optimization [21]. In this study, ACA parameters are mapped into a multidimensional space, and a chemotactic operator is used to enable each group of parameters to approach the optimal value and speed up the convergence of each set of parameters. A reproduction operator is used to accelerate the optimization for the entire set of parameters. Lastly, the elimination-dispersal operator is utilized to strengthen the global optimization ability of the parameters to avoid falling into a local optimal solution. The BFA is commonly applied to solve the problem of parameter selection for ACA and is able to take into account the coupling between parameters, which allows for adjusting the parameters automatically with little prior knowledge. Therefore, it is appropriate to apply the BFA to obtain the ACA parameters.

2. Model and Parametric Analysis of the Ant Colony Algorithm

The ACA described in this paper will be applied to robot path planning. Generally, environmental modeling is an important aspect of path planning.

2.1. Environmental Modeling. By the direct encoding format for the data, this study uses the grid method [22] for environmental modeling. XY is described as a convex polygon, an area of limited motion that contains several static obstacles in a two-dimensional plane. A Cartesian coordinate system is established for XY . Figure 1 shows a grid sequence diagram with a size of 10×10 , and the black grids in the figure represent obstacles. v is an arbitrary grid within XY , V is the sum of all grids, and, therefore, $v \in V$. $v(x, y)$ are the coordinates of a grid, where its center point coordinates are x and y . Suppose that $N = \{1, 2, \dots, n\}$ is a set of grid sequence number, and this study regards a grid i ($i \in N$) as $v_i(x_i, y_i)$.

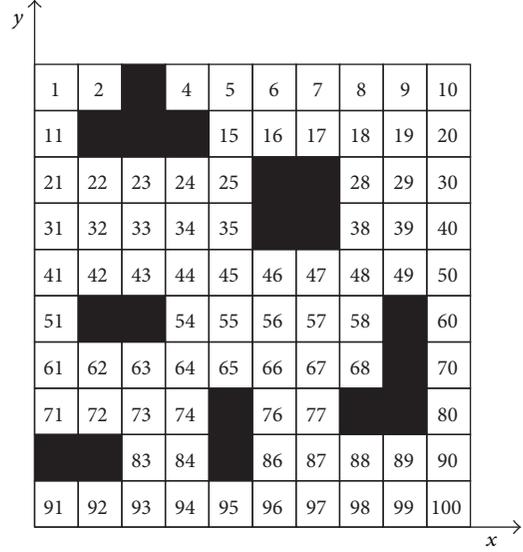


FIGURE 1: Grid sequence diagram.

2.2. Model of Ant Colony Algorithm

2.2.1. Rules of Path Selection. $p_{ij}^m(t)$ is defined as the transition probability that ant m walks forward from v_i to v_j , which can be expressed as follows [23]:

$$p_{ij}^m(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{v_m \in V_{\text{allowed}}^m} [\tau_{im}(t)]^\alpha \cdot [\eta_{im}(t)]^\beta}, & \text{if } v_j \in V_{\text{allowed}}^m, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where V_{allowed}^m is a set of the next feasible grid when ant m arrives at v_i ; $\tau_{ij}(t)$ is the residual pheromones between v_i and v_j at t time; α represents the information elicitation factor, which shows the relative importance of $\tau_{ij}(t)$; $\eta_{ij}(t)$ is the expectation heuristic function between v_i and v_j at t time, which can be defined as the reciprocal of the distance d_{ij} between v_i and v_j , namely, $\eta_{ij}(t) = 1/d_{ij}$; β is the expected heuristic factor, which shows the relative importance of $\eta_{ij}(t)$.

2.2.2. Rules of Pheromone Updating. Ants will leave pheromones in their path, and the pheromones will evaporate over time to avoid the masking of heuristic information due to excessive residual pheromones. Supposing that ρ ($0 \leq \rho \leq 1$) is the pheromone evaporation coefficient, the rules of pheromone updating between v_i and v_j at $t + \Delta t$ time can be expressed as follows [23]:

$$\tau_{ij}(t + \Delta t) = (1 - \rho) \cdot \tau(t) + \Delta\tau_{ij}(t), \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{m=1}^M \Delta\tau_{ij}^m(t),$$

where $\Delta\tau_{ij}^m(t)$ is the pheromone that ant m leaves between v_i and v_j , and this paper applies the Ant-Cycle model [1] proposed by Dorigo et al., which can be expressed as follows:

$$\Delta\tau_{ij}^m(t) = \begin{cases} \frac{Q}{L_m}, & v_i, v_j \in \text{Path}_m, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where Q is the pheromone intensity; L_m is the total length of path traversed by ant m ; Path_m is the set of grids traversed by ant m .

2.3. Parametric Analyses of the Ant Colony Algorithm. The parameters for ACA are extremely important to the performance of the algorithm, and appropriate parameters can enhance the global search ability of the algorithm and improve the convergence speed considerably [24]. From the above formulas, it is evident that the parameters with a significant impact on the path selection of the ant m are α , β , ρ , and Q . We will elaborate further on the relationship between the four parameters and the performance of the ACA.

2.3.1. Information Elicitation Factor α . The information elicitation factor α , which represents the relative importance of the pheromone, reflects the importance of the accumulation of the pheromone with regard to the ants' path selection. If α is very large, the ants tend to choose the same path that the preceding ants have chosen, resulting in stronger cooperation among the ants. Although the convergence speed of ACA will be accelerated, it is likely for the algorithm to fall into the local optimal solution and reduce the global search ability. Conversely, if α is too small, the convergence speed of the ACA is slowed down, regardless of the fact that the global search ability of the algorithm can be improved.

2.3.2. Expected Heuristic Factor β . The expected heuristic factor β , which represents the relative importance of the visibility, reflects the importance of the heuristic information with regard to the ants' path selection. If the value is very large, the probability of a state transition is close to that of a greedy algorithm. If β is too small, the heuristic information has virtually no effect on path selection, which leads ACA to fall into stagnation or a local optimum.

2.3.3. Pheromone Evaporation Coefficient ρ . The pheromone evaporation coefficient ρ , which represents the degree of pheromone evaporation, reflects the degree of mutual influence among ants. Generally, the value of ρ is $[0, 1]$, which prevents the infinite accumulation of pheromone effectively. If ρ is too small, the global search ability of ACA will be reduced. Otherwise, if ρ is too large, it will improve the global

search ability of ACA; however, the convergence speed will be slow.

2.3.4. Pheromone Intensity Q . The pheromone intensity Q , which represents the total pheromone, affects the convergence speed of the ACA to a certain extent. If Q is large, the pheromone concentration will be highly concentrated, which leads the algorithm to fall into a local optimum. Furthermore, a small Q results in a slow optimization speed.

3. Parameters Selection of ACA Based on BFA

In 2002, Passino put forward a bacterial foraging algorithm [17], based on the foraging behavior of *Escherichia coli* in the human intestinal tract. This algorithm includes a chemotaxis operator, a reproduction operator, and an elimination-dispersal operator. The algorithm is able to search in parallel and jump out of the local minima easily. In this paper, we will transfer any set of parameters $(\alpha_i, \beta_i, \rho_i, Q_i)$ to four-dimensional arrays, which are seen as an individual bacteria p_i , where $p_i = (\alpha_i, \beta_i, \rho_i, Q_i)^T$. P is a bacterial population, and $P = \{p_1, p_2, \dots, p_n\}$. Because the application of the algorithms discussed in this paper is aimed at the problem of robot path planning in a grid map environment, the objective of this study is the selection of optimal parameters for the ACA. path_i is the path for a robot under the individual bacteria p_i , and $\text{path}_i \in \text{PATH}$. Therefore, the fitness function of BFA can be expressed as follows:

$$f(p_i) = \text{Length}(\text{path}_i), \quad (4)$$

where $\text{Length}(\ast)$ is adopted to calculate the length of a path.

3.1. Chemotaxis Operator T_c^B . The chemotaxis operator represents the core of the BFA, and it determines changes in the location of the bacteria that are searching for food. Therefore, the chemotaxis operator plays a decisive role in finding a food source for the bacteria, which has an important influence on the performance and convergence of the algorithm. $P(m)$ is the bacterial population, and $P(m) = [p_1(m) \ p_2(m) \ \dots \ p_n(m)]^T$. And the chemotaxis operator can be expressed as follows:

$$\begin{aligned} P_c^B(m) &= T_c^B(P(m)) \\ &= [p_1^{Bc}(m) \ p_2^{Bc}(m) \ \dots \ p_n^{Bc}(m)]^T \end{aligned} \quad (5)$$

$\forall i \in [1, n], \ p_i^{Bc}(m) = T_c^B(p_i(m))$.

Supposing that $p_i(m) = p_i(m, j, k, l)$ and $p_i^{Bc}(m) = p_i(m, j + 1, k, l)$, this paper adjusts the position of a bacteria in accordance with the formula (6), which can be expressed as follows [25]:

$$p_i(m, j + 1, k, l) = p_i(m, j, k, l) + \text{Step} \times \varphi(i), \quad (6)$$

$$\varphi(i) = \frac{p_i(m, j, k, l) - p_{\text{rand}}(m, j, k, l)}{\text{sqrt}((p_i(m, j, k, l) - p_{\text{rand}}(m, j, k, l))^T \times (p_i(m, j, k, l) - p_{\text{rand}}(m, j, k, l)))}, \quad (7)$$

where $p_i(m, j, k, l)$ represents the current location of the bacterial individual i ; j is the number of the chemotaxis operator; k is the number of the reproduction operator; l is the number of the elimination-dispersal operator; Step represents the step that the bacteria moves forward each time; $\varphi(i)$ is the direction of random tumbling; $p_{\text{rand}}(m, j, k, l)$ represents a random position in the neighborhood of the current individual.

In the chemotaxis operator, the movement patterns of the bacteria include flipping and moving. The bacteria move forward in any direction with a unit step, which is defined as the flipping operator. After $p_i(m)$ executes the flipping operator, namely, $p_i^{\text{Bc}}(m) = T_c^{\text{B}}(p_i(m))$, and the fitness value of $p_i(m)$ is not improved, namely, $f(p_i^{\text{Bc}}(m)) > f(p_i(m))$, $p_i(m)$ jumps out of the loop. If the fitness value of $p_i(m)$ is improved, namely, $f(p_i^{\text{Bc}}(m)) < f(p_i(m))$, $p_i(m)$ keeps moving in the same direction until the fitness value cannot be improved further, or the algorithm achieves the maximum number of flipping, which is defined as the moving operator.

3.2. Reproduction Operator T_r^{B} . The reproduction operator ensures the improvement in the performance of the bacterial population, which encourages the population to move towards the optimal direction. The reproduction operator is conducive to achieving the global optimal, which can be expressed as follows:

$$\begin{aligned} P_r^{\text{B}}(m) &= T_r^{\text{B}}(P_c^{\text{B}}(m)) \\ &= [p_1^{\text{Br}}(m) \ p_2^{\text{Br}}(m) \ \cdots \ p_n^{\text{Br}}(m)]^T. \end{aligned} \quad (8)$$

Supposing that $P_c^{\text{B}}(m)_{\text{value}}$ is the fitness value of the bacterial population after performing the chemotaxis operator, which can be expressed as follows:

$$P_c^{\text{B}}(m)_{\text{value}} = f(P_c^{\text{B}}(m)) = \text{Length}(\text{PATH}_c^{\text{B}}(m)). \quad (9)$$

With the standard of $P_c^{\text{B}}(m)_{\text{value}}$, this paper gets $P_c^{\text{B}}(m)_{\text{bad}}$ that is half of the bacterial population with a bad fitness value and $P_c^{\text{B}}(m)_{\text{good}}$ that represents half of the bacterial population with a good fitness value. Therefore, $P_c^{\text{B}}(m) = P_c^{\text{B}}(m)_{\text{bad}} \cup P_c^{\text{B}}(m)_{\text{good}}$. Then, this paper replaces $P_c^{\text{B}}(m)_{\text{bad}}$ with $P_c^{\text{B}}(m)_{\text{good}}$ to get $P_r^{\text{B}}(m)$, and $P_r^{\text{B}}(m) = [P_c^{\text{B}}(m)_{\text{good}} \ P_c^{\text{B}}(m)_{\text{good}}]$.

3.3. Elimination-Dispersal Operator T_e^{B} . The elimination-dispersal operator generates a new individual with a certain probability, which has a different location compared to the dead individual. The operator has a promoting effect on the algorithm, because the new individual may be closer to the food source, which is more conducive to jumping out of the local optimal solution and finding the global optimal solution. The elimination-dispersal operator can be expressed as follows:

$$\begin{aligned} P_e^{\text{B}}(m) &= T_e^{\text{B}}(P_r^{\text{B}}(m)) \\ &= [p_1^{\text{Be}}(m) \ p_2^{\text{Be}}(m) \ \cdots \ p_n^{\text{Be}}(m)]^T \end{aligned} \quad (10)$$

$\forall i \in [1, n]$, $p_i^{\text{Be}}(m) = T_e^{\text{B}}(p_i^{\text{Br}}(m))$. In this paper, some bacteria individuals pass away in a certain probability. Supposing that p_e is the probability of death for the individual. If $\text{rand} < p_e$, $p_i^{\text{Br}}(m)$ passes away, and $p_i^{\text{Be}}(m) = p_{\text{rand}}(m)$, where $p_{\text{rand}}(m)$ is a new bacterial individual generated randomly.

3.4. Steps of the Algorithm. See Algorithm 1.

4. Experimental Analysis

In order to analyze the application performance of the parameter selection for the ACA based on the BFA, we ran a simulation for robot path planning in a grid map environment.

4.1. Establishing Grid Maps. Figure 2 shows six grid maps, where the white grid represents an accessible area and the black grid is a barrier. The ACA based on the parameters obtained using the BFA is used to determine the robot's optimal path in the map without running into obstacles. The upper left corner grid is the starting point, and the lower right corner grid is the end point. Beginning at the starting point, the robot has the potential to move into 8 different directions, including the front, rear, left, right, left front, left rear, right front, and right rear. Excluding the grids with obstacles or already gone, the robot moves one grid at a time. Details of the grid maps are shown in Table 1, including the size, the number of grids, the number of obstacles, and the coverage rate. Maps 1–4 are of the same size, but the complexity is different for each map. Although Map 5 is small, its complexity is high. Map 6 is not only large, but also highly complex. In general, the larger and the more complex the grid map is, the more difficult robot path planning becomes.

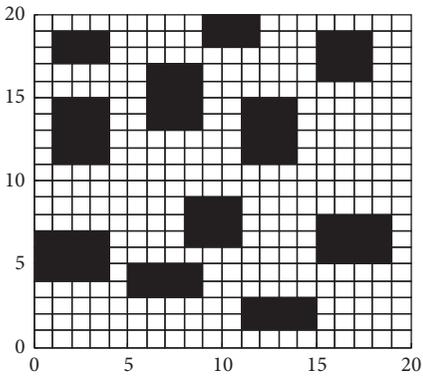
4.2. Simulation. To analyze further the application performance of the parameters obtained by using the BFA for the ACA, we conducted the tests for the different grid maps using MATLAB and compared the results with the GA and the PSO. The following parameters were used: parameter initialization of ACA: $m = 40$; parameter initialization of GA: size of population: 20, probability of selection: 0.08, probability of crossover: 0.3, probability of mutation: 0.1, maximum generation: 32; parameter initialization of PSO: number of particles: 20, inertia weight: 0.5, parameter of speed adjusting: 1.9 and 0.8; parameter initialization of BFA: $P = 20$, $N_c = 4$, $N_s = 4$, Step = 1, $N_r = 4$, $N_e = 2$, and $P_e = 0.1$. Prior to using the different intelligent algorithms to determine ACA parameter selection, we initialized α , β , ρ , and Q: $\alpha \in [0, 10]$, $\beta \in [0, 20]$, $\rho \in [0, 1]$, and Q $\in [50, 150]$.

Formula (4) was used as the fitness function for BFA, which was used to compare the parameter performance. In order to ensure the rigor of the experiment, formula (4) was also used as the fitness function of GA and PSO. Length(*) was adopted to calculate the length of the route path_{*i*} of the bacterial individual p_i . Therefore, ACA was used for path planning to obtain the different route path under different map environments for the GA, PSO, and BFA. Due to the randomness of ACA, a fixed mathematical formula for ACA could not be used for path planning. As long as

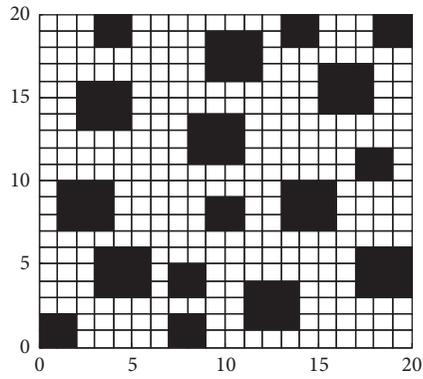
```

Begin
  Initialize: the bacterial population  $P$ ; times of chemokines  $N_c$ ; maximum
  steps of chemokines  $N_s$ ; step Step; times of reproduction  $N_r$ ; times of
  elimination-dispersal  $N_e$ ; probability of elimination-dispersal  $P_e$ ;
  maximum generations  $G_{\max}$ .
   $m = 1$ ;
  While  $m \leq G_{\max}$ 
    For  $j = 1: N_e$ 
      For  $k = 1: N_r$ 
        For  $l = 1: N_c$ 
          While  $i \leq n$ 
             $P_c^B(m) = T_c^B(P(m)) = [p_1^{Bc}(m) \ p_2^{Bc}(m) \ \dots \ p_n^{Bc}(m)]^T$ ;
             $i = i + 1$ ;
          End
        End
         $P_r^B(m) = T_r^B(P_c^B(m)) = [p_1^{Br}(m) \ p_2^{Br}(m) \ \dots \ p_n^{Br}(m)]^T$ ;
      End
      If  $\text{rand} < P_e$ 
         $P_e^B(m) = T_e^B(P_r^B(m)) = [p_1^{Be}(m) \ p_2^{Be}(m) \ \dots \ p_n^{Be}(m)]^T$ ;
      End
    End
     $m = m + 1$ ;
  End
End
  
```

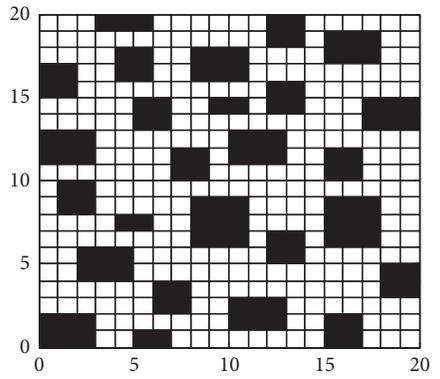
ALGORITHM 1



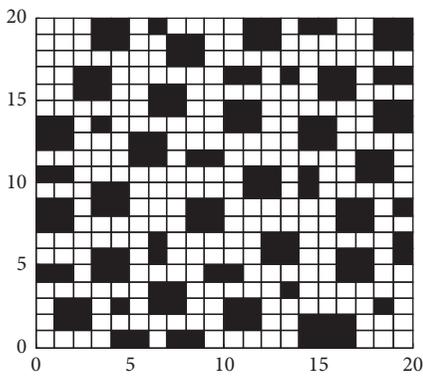
(a) Map 1



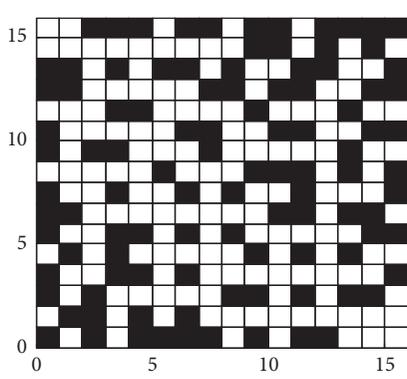
(b) Map 2



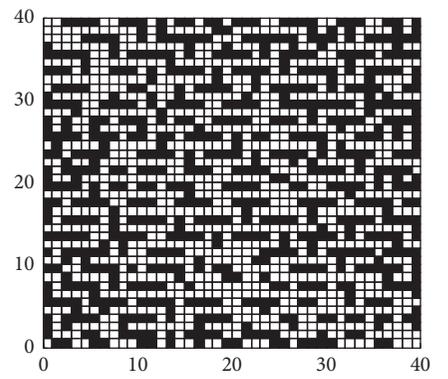
(c) Map 3



(d) Map 4



(e) Map 5



(f) Map 6

FIGURE 2: Six grid maps.

TABLE 1: Details of grid maps.

	Size	Number of grids	Number of obstacles	Coverage rate
Map 1	20 × 20	400	106	26.5%
Map 2	20 × 20	400	113	28.25%
Map 3	20 × 20	400	123	30.75%
Map 4	20 × 20	400	133	33.25%
Map 5	16 × 16	256	103	40.23%
Map 6	40 × 40	1600	663	41.44%

TABLE 2: Performance comparison of GA, PSO, and BFA.

	Optimization times	Number of fitness function calls	Simulation time/s	Minimum distance	Average distance
GA	10	274	142.1787	78.1838	101.9677
	20	532	298.3970	75.4975	101.1249
	30	788	449.1179	72.6690	101.1712
PSO	10	220	118.1516	76.7696	107.8844
	20	420	217.0521	75.4975	107.5190
	30	620	334.9856	72.9411	106.6085
BFA	10	505	249.2963	72.6690	96.1448
	20	1018	526.5445	71.2548	91.8834
	30	1570	810.1920	71.0122	91.1293

TABLE 3: Test results of GA, PSO, and BFA.

		Map 1	Map 2	Map 3	Map 4	Map 5	Map 6
GA	Maximum distance	82.6690	80.6690	90.0833	85.4975	56.5269	169.9239
	Minimum distance	28.6274	28.6274	28.6427	28.6427	22.3848	72.6690
	Average distance	34.9960	34.7249	36.0491	35.4845	24.8728	101.5615
	Standard deviation	2.1426	2.1856	3.1071	2.3975	1.4661	3.0223
PSO	Maximum distance	45.4558	47.1127	41.1127	43.7990	28.7279	191.5807
	Minimum distance	28.0416	28.0416	28.0416	28.6427	22.3848	74.4264
	Average distance	32.2744	31.4533	33.2238	32.7348	24.2252	100.0113
	Standard deviation	0.4886	0.5327	0.3746	0.5969	0.1976	6.2984
BFA	Maximum distance	39.7990	38.8701	39.4558	39.7990	29.2132	119.3970
	Minimum distance	28.0416	28.0416	28.0416	28.0416	22.3848	69.0122
	Average distance	30.9162	31.2931	31.5497	31.6602	23.3873	88.9554
	Standard deviation	0.7087	0.6806	0.5672	0.6367	0.7376	2.7994

GA, PSO, and BFA call the fitness function, these algorithms have to use ACA for path planning, which increases the complexity and simulation time for the algorithms, but cannot be avoided. For example, for Map 6, Table 2 shows the performance comparison of GA, PSO, and BFA. It is evident that the fitness function is invoked multiple times for either algorithm in Table 2. The chemotaxis operator and reproduction operator of the BFA are based on the fitness value of individual bacteria for the evolution of the evaluation criteria. As a result, BFA had the highest number for invoking the fitness function, resulting in the longest simulation time for BFA. However, this study was mainly concerned with path planning based on global maps, which did not require rapid real-time performance. Therefore, it was acceptable to use the BFA for ACA parameter selection. In Table 2, the performance for parameter selection for GA and PSO for 30 optimizations was inferior compared to the performance of

BFA for 10 optimizations, which demonstrated that BFA had fast convergence speed and superior performance.

The test results for the use of GA, PSO, and BFA are shown in Table 3. The maximum, minimum, and average values were the best for BFA, indicating that the BFA algorithm was the preferred method for determining the combination of optimal parameters. The chemotactic operator that adjusts the parameters in an adaptive manner ensures that each group of parameters approaches the optimal value in order to speed up the convergence. The reproduction operator accelerates the speed of optimization for the entire set of parameters. Therefore, the parameters obtained using BFA were the most appropriate to achieve the shortest path for the robot using ACA. The standard deviation for BFA was slightly larger than for PSO due to the elimination-dispersal operator, which introduced new parameter combinations. The new parameter combinations with unknown performance increased the

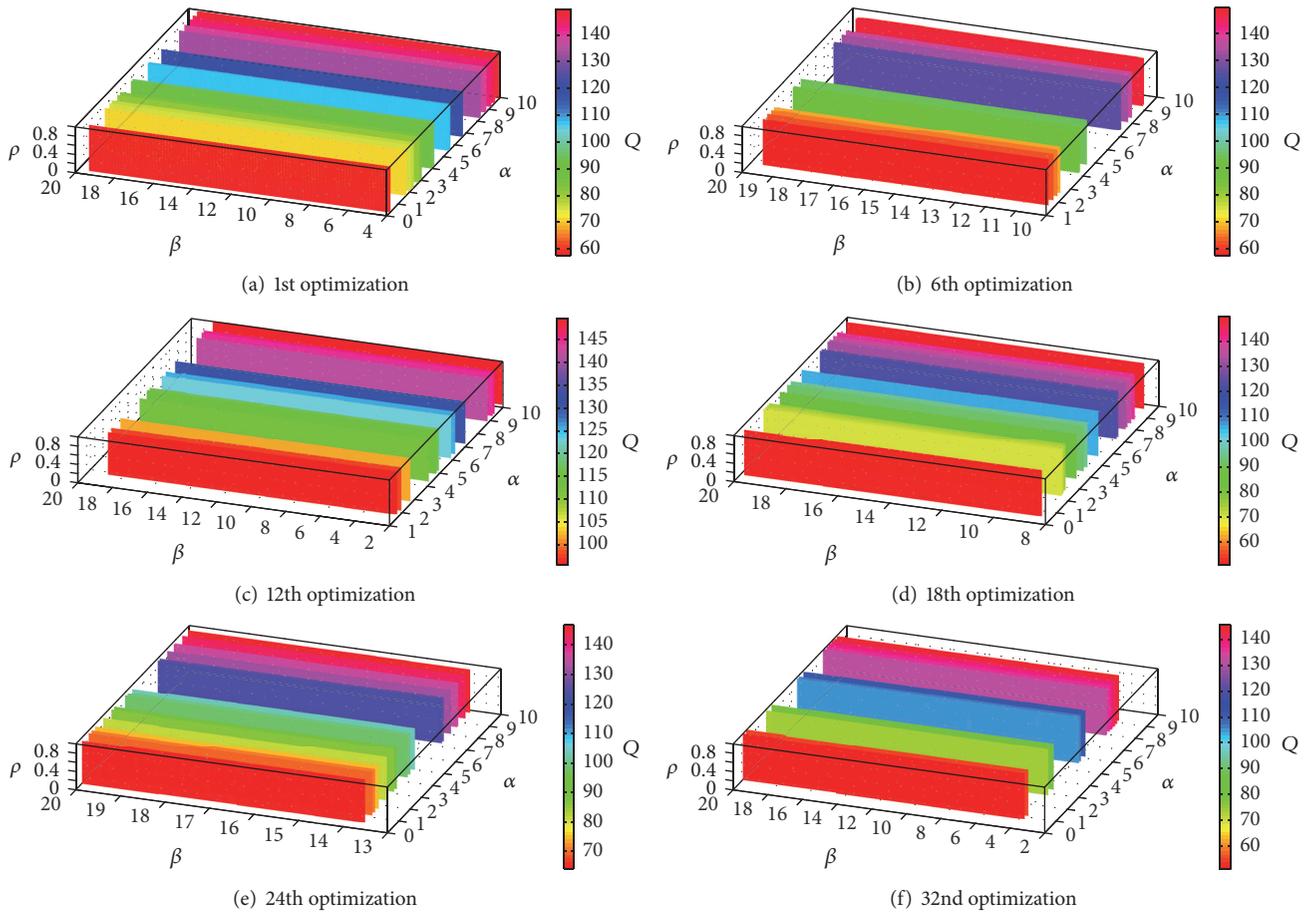


FIGURE 3: Distribution of parameters obtained by GA.

diversity of the parameter combinations, which improved the global optimization and avoided the algorithm to fall into a local optima. Furthermore, the initial path of the robot was relatively long. The test results shown in Map 1–5 demonstrate that the performance based on GA was the worst, that of BFA was the best, and that of PSO was close to BFA. The test results shown in Map 6 indicate that BFA performed best, demonstrating that BFA was most suitable for parameter optimization in a complex environment. Therefore, the test results proved that parameter selection for ACA based on BFA was highly effective and superior to the other algorithms.

The initial number of parameter combinations was 20 for each algorithm. Using Map 6 as an example, Figures 3, 4, and 5 show the distribution of parameters under different optimal generation for the three algorithms. In Figure 3, the distribution of parameters underwent a process of aggregating diverging and aggregating. Therefore, it was difficult for the parameters to approach the optimal value continuously. The distribution of parameters obtained using PSO is illustrated in Figure 4, and it shows that the position and velocity of a particle are updated constantly based on the population information and individual experience in order to obtain the optimal parameters. The distribution of the 20-parameter combinations exhibited a gradual trend

to aggregate. The parameters aggregated quickly from the 1st optimization to the 12th optimization. However, after the 18th optimization, the aggregating trend was almost unchanged, indicating that PSO fell into a local optimal solution and underwent premature convergence. Figure 5 depicts the parameter distribution obtained using BFA and shows that the 20 parameter combinations aggregate continuously due to the chemotactic operator and reproduction operator. These results demonstrated the effectiveness and superiority of parameter selection for ACA using BFA.

Figure 6 shows the evolution curves of the distances using the three algorithms in the environment of Map 6. The average evolution curves indicate that the convergence speed was highest for BFA. The average evolution curves for GA and PSO exhibited relatively large fluctuations. The convergence speed of GA and PSO was slow, indicating that GA and PSO fell into a local optimal solution. The optimal evolution curves show that BFA had the strongest optimization ability. Moreover, the optimization process for BFA was the most stable. The optimization processes for GA and PSO had large fluctuations.

The parameters of ACA for the shortest path are illustrated in Table 4, where the shortest path represents the minimum value shown in Table 3. The results indicated that

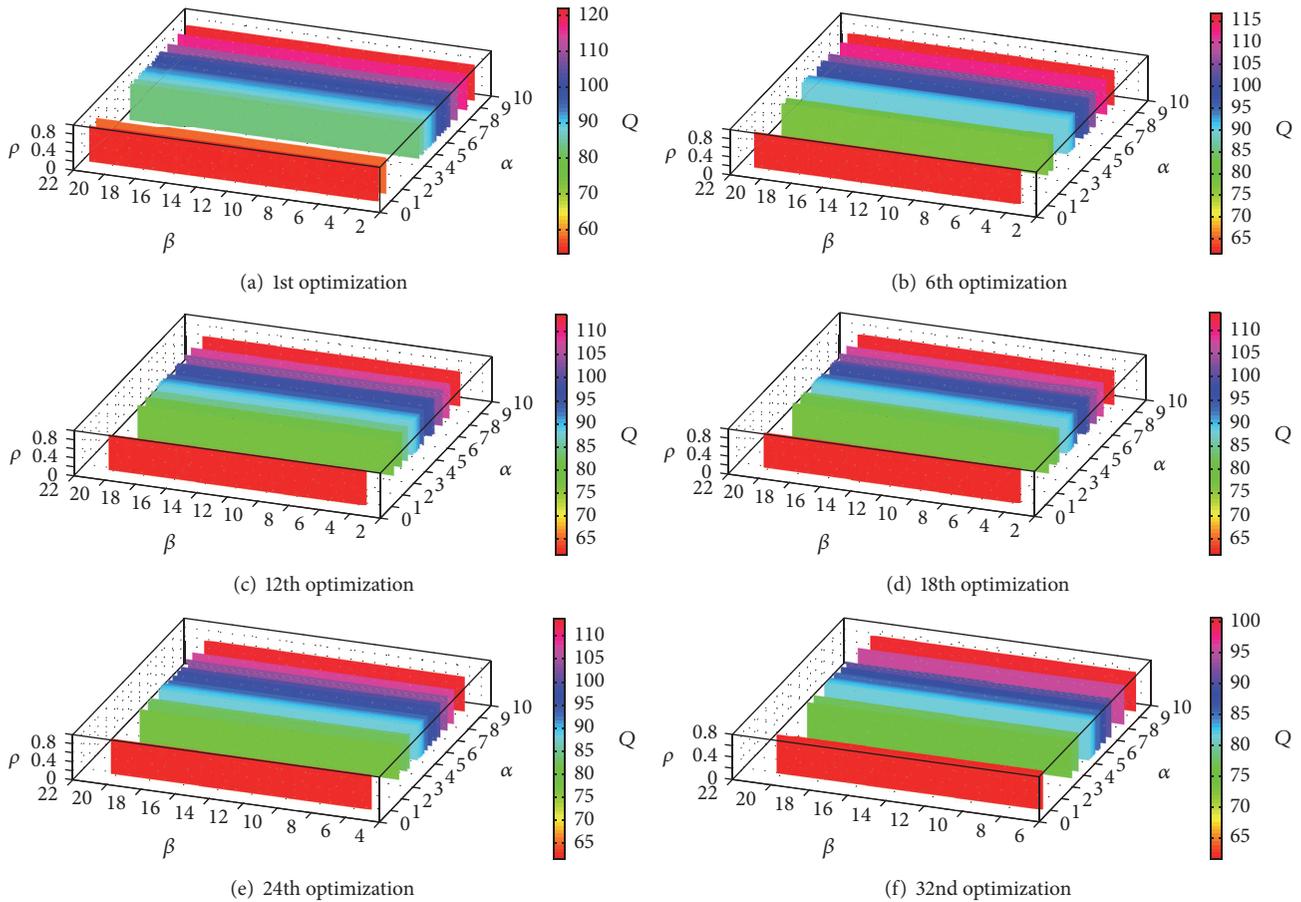


FIGURE 4: Distribution of parameters obtained by PSO.

TABLE 4: Parameters of ACA in the shortest path.

		Map 1	Map 2	Map 3	Map 4	Map 5	Map 6
GA	α	2.1260	9.2126	8.1890	2.2835	7.1654	8.7402
	β	20	19.451	17.2549	18.9804	12.8627	18.6667
	ρ	1	0.8	0.4667	0.6667	0.4667	0.8000
PSO	Q	54.3988	116.8622	78.1525	122.5318	51.7595	146.6764
	α	9.1656	7.2622	6.2604	4.4029	1.7695	5.0572
	β	19.0578	22.7214	14.6075	16.6394	15.0183	18.1021
	ρ	0.4479	0.7653	0.5728	0.4764	0.3806	0.5898
BFA	Q	94.8379	133.7824	91.8428	129.7067	132.0365	66.2206
	α	7.7052	10.0753	1.1790	4.8531	9.6174	1.3500
	β	19.9652	18.8933	18.5790	19.3417	19.2342	19.5401
	ρ	0.2760	0.0199	0.5992	0.3716	0.1962	0.8699
	Q	51.6965	132.6316	100.8766	139.1583	147.0755	70.7550

the shortest distance obtained by the different algorithms was the same for the same grid map. Nevertheless, the corresponding parameter combinations were different, which proved that the ACA parameters were coupled with each other. Therefore, to find the relationship between the parameters and the performance of ACA, some researchers use the

control variable method and change a single parameter for an experiment, which is considered not rigorous.

The performance of the parameter combinations in Table 4 was determined using the fitness function. The fitness function performs path planning for the current grid map, which means that the performance is based on chance and

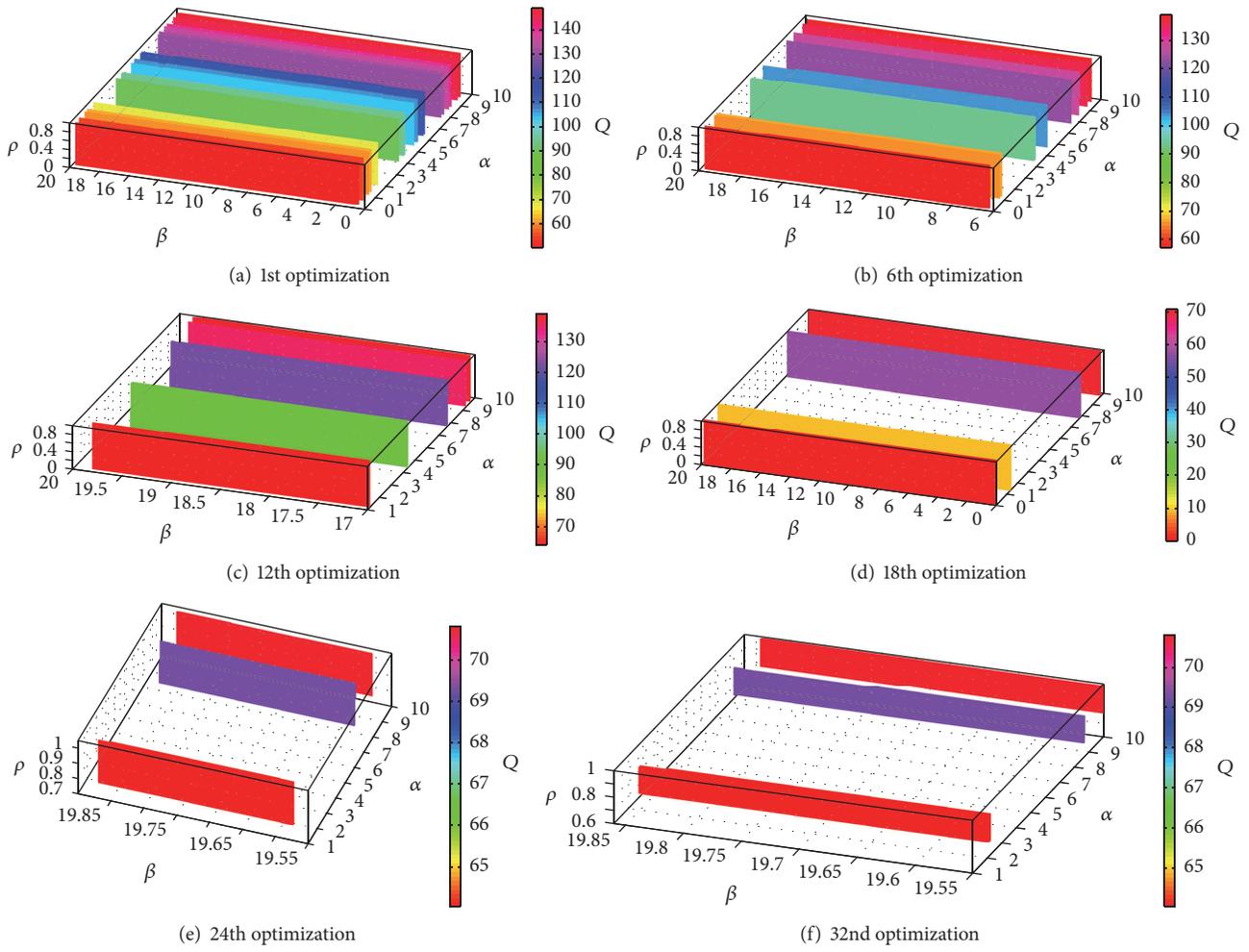


FIGURE 5: Distribution of parameters obtained by BFA.

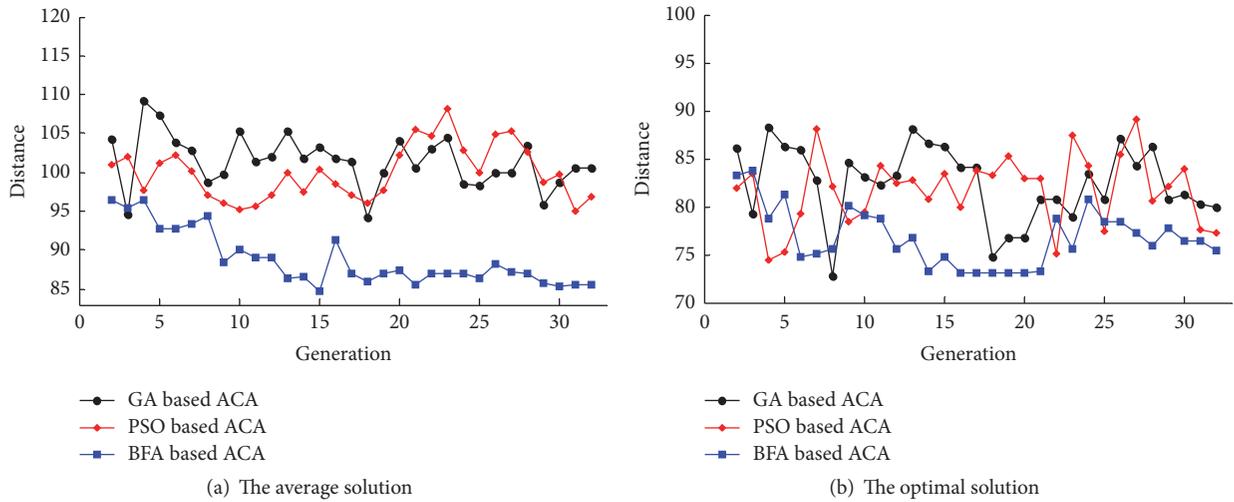


FIGURE 6: Evolution curves of the distance by the three algorithms.

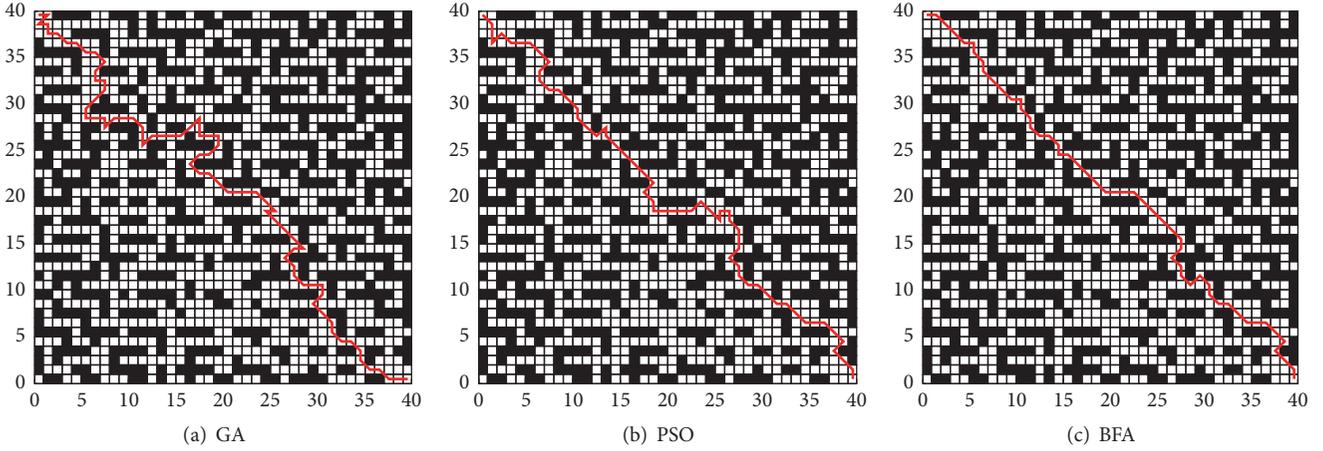


FIGURE 7: Optimal paths under different parameter combinations.

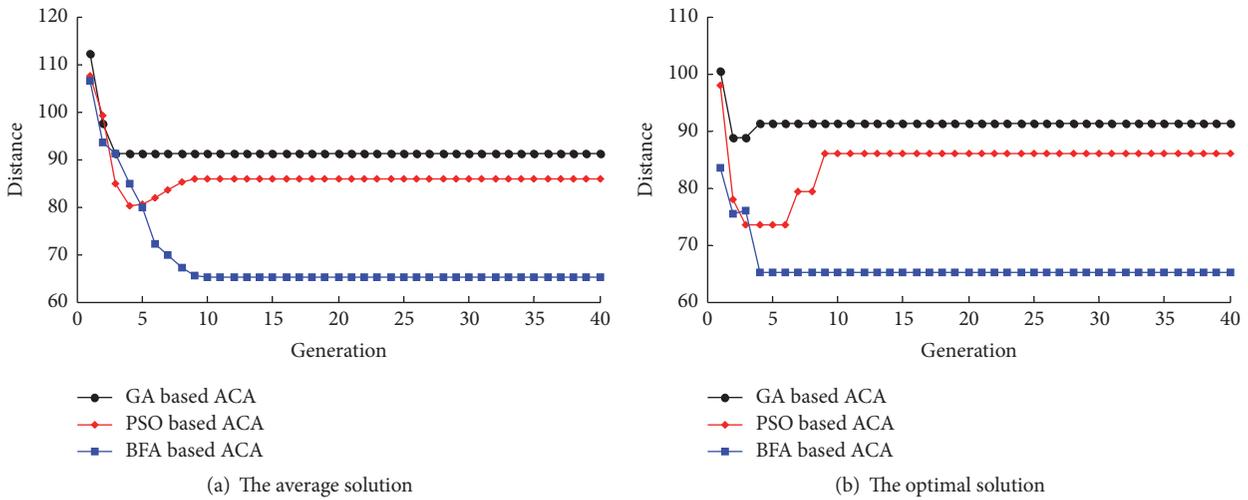


FIGURE 8: Evolution curves of the distance under different parameter combinations.

TABLE 5: Test results under different parameter combinations.

	GA ($\alpha = 8.7402, \beta = 18.666, \rho = 0.8000, Q = 146.6764$)	PSO ($\alpha = 5.0572, \beta = 18.1021, \rho = 0.5898, Q = 66.2206$)	BFA ($\alpha = 1.3500, \beta = 19.5401, \rho = 0.8699, Q = 70.7550$)
Maximum distance	139.6396	128.2254	129.7817
Minimum distance	88.669	73.4975	65.2548
Average distance	91.41	85.5504	66.7729
Standard deviation	2.1076	3.033	6.1966

randomness. To validate further the performance of the parameter combinations, tests were conducted based on ACA with different parameter combinations and using Map 6 for robot path planning. It was assumed that the number of ants was 40 and the cycle time was 40.

The results of this test are shown in Table 5. Based on the maximum, minimum, and average values, it was evident that the use of BFA resulted in the best performance regarding parameter combination. The initial path selection of the ants was poor. As the cycle increased, the path became shorter. The

ants found the best route based on the parameters obtained by BFA, and the path was much shorter than the initial path, resulting in a large standard deviation. To some extent, the large standard deviation proved the effectiveness and superiority of BFA used for parameters selection for ACA.

The optimal paths using different parameter combinations are shown in Figure 7. The use of ACA based on the parameters obtained using BFA, resulted in the shortest path, indicating that BFA was able to obtain the most appropriate parameters. Figure 8 illustrates the evolution curves of the

distance under different parameter combinations, and it is evident that the parameters obtained by GA or PSO led to ACA falling into the local optimal solution and to an apparent premature convergence. However, using ACA based on parameters obtained by BFA resulted in the optimal path. The test results in this paper showed that the parameters had a large impact on the performance of ACA. Choosing suitable parameters enabled ACA to achieve an optimal performance. However, unsuitable parameters resulted in ACA falling into a local optima. The test results showed that the use of BFA could determine optimal parameters for ACA to achieve the best performance.

5. Conclusion

Optimal performance of ACA mainly depended on suitable parameters. However, the parameters were coupled with each other, and the potential number of parameters was large, presenting a considerable challenge for ACA parameter selection. Therefore, multiple parameter selection for ACA based on BFA was proposed. The parameters for ACA were mapped into a multidimensional space, and the optimal parameters were obtained automatically using BFA, demonstrating the superiority of the method. Moreover, the four parameters α , β , ρ , and Q could be obtained simultaneously. The parameter optimization process took into account the coupling between the parameters, which ensured the rationality of the parameters. The test results showed that the parameters obtained using BFA for ACA were the most suitable. Compared with GA and PSO, BFA had a better performance for parameter selection of ACA with regard to convergence speed, optimization capability, and stability, which demonstrated the effectiveness and superiority of parameter selection for ACA based on BFA.

Competing Interests

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

Acknowledgments

The project is supported by the National High Technology Research and Development Program of China (863 Program) (Grant no. 2012AA041504) and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

- [1] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [2] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.
- [3] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [4] K.-L. Huang and C.-J. Liao, "Ant colony optimization combined with taboo search for the job shop scheduling problem," *Computers and Operations Research*, vol. 35, no. 4, pp. 1030–1046, 2008.
- [5] C. Blum and M. Sampels, "An ant colony optimization algorithm for shop scheduling problems," *Journal of Mathematical Modelling and Algorithms*, vol. 3, no. 3, pp. 285–308, 2004.
- [6] Q. Xu, J. Mao, and Z. Jin, "Simulated annealing-based ant colony algorithm for tugboat scheduling optimization," *Mathematical Problems in Engineering*, vol. 2012, Article ID 246978, 22 pages, 2012.
- [7] Y. Jiang, Z. Xu, X. Xu, Z. Liao, and Y. Luo, "A schedule optimization model on multirunway based on ant colony algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 368208, 11 pages, 2014.
- [8] H. Mei, Y. Tian, and L. Zu, "A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment," *International Journal of Information Technology*, vol. 12, no. 3, pp. 78–88, 2006.
- [9] G.-Z. Tan, H. He, and A. Sloman, "Ant colony system algorithm for real-time globally optimal path planning of mobile robots," *Acta Automatica Sinica*, vol. 33, no. 3, pp. 279–285, 2007.
- [10] J. Zhao, D. Cheng, and C. Hao, "An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle," *Mathematical Problems in Engineering*, vol. 2016, Article ID 7672839, 10 pages, 2016.
- [11] Y. Dai, L. Liu, and S. Wang, "Parameter optimization of ant colony algorithm based on particle swarm optimization," in *Proceedings of the International Conference*, pp. 1266–1269, University of Electronic Science and Technology of China, China, December 2007.
- [12] X. S. Wang, J. Pan, and Y. H. Cheng, "Parameter selection for ant colony algorithms based on graph knowledge transfer," *Control and Decision*, vol. 26, no. 12, pp. 1855–1860, 2011.
- [13] Y. Hei and P. Du, "Optimal choice of the parameters of ant colony algorithm," *Journal of Convergence Information Technology*, vol. 6, no. 9, pp. 96–104, 2011.
- [14] Y. Gan and S. Li, "Studies on parameters configuration for microhabitat ant colony optimization algorithm," *Manufacturing Automation*, vol. 33, no. 3, pp. 66–69, 2011.
- [15] Y. Feng, "Parameters optimization research of ant colony optimization based on genetic algorithm," *Journal of Guiyang College Natural Sciences*, vol. 9, no. 4, pp. 25–27, 2014.
- [16] Q. Li, C. Zhang, P. Chen, and Y.-X. Yin, "Improved ant colony optimization algorithm based on particle swarm optimization," *Control and Decision*, vol. 28, no. 6, pp. 873–883, 2013.
- [17] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.
- [18] F. Zhao, Y. Liu, Z. Shao, X. Jiang, C. Zhang, and J. Wang, "A chaotic local search based bacterial foraging algorithm and its application to a permutation flow-shop scheduling problem," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 9, pp. 962–981, 2016.
- [19] X.-D. Liang, L.-Y. Li, J.-G. Wu, and H.-N. Chen, "Mobile robot path planning based on adaptive bacterial foraging algorithm," *Journal of Central South University*, vol. 20, no. 12, pp. 3391–3400, 2013.
- [20] O. P. Verma, M. Hanmandlu, A. K. Sultania, and A. S. Parihar, "A novel fuzzy system for edge detection in noisy image using bacterial foraging," *Multidimensional Systems and Signal Processing*, vol. 24, no. 1, pp. 181–198, 2013.

- [21] S. Dasgupta, A. Biswas, S. Das, B. K. Panigrahi, and A. Abraham, "A micro-bacterial foraging algorithm for high-dimensional optimization," in *2009 IEEE Congress on Evolutionary Computation (CEC)*, pp. 785–792, Trondheim, Norway, May 2009.
- [22] P. Li and H. Zhu, "Immune optimization of path planning for coal mine rescue robot," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, vol. 11, no. 1, pp. 91–98, 2016.
- [23] Q. Yang, L. Fang, and X. Duan, "RMACO: a randomly matched parallel ant colony optimization," *World Wide Web*, vol. 19, no. 6, pp. 1009–1022, 2016.
- [24] L. Liu, Y. Dai, and L. Wang, "Ant colony algorithm parameters optimization," *Computer Engineering*, vol. 34, no. 11, pp. 208–210, 2008.
- [25] C. Yang, J. Ji, J. Liu, J. Liu, and B. Yin, "Structural learning of Bayesian networks by bacterial foraging optimization," *International Journal of Approximate Reasoning*, vol. 69, pp. 147–167, 2016.