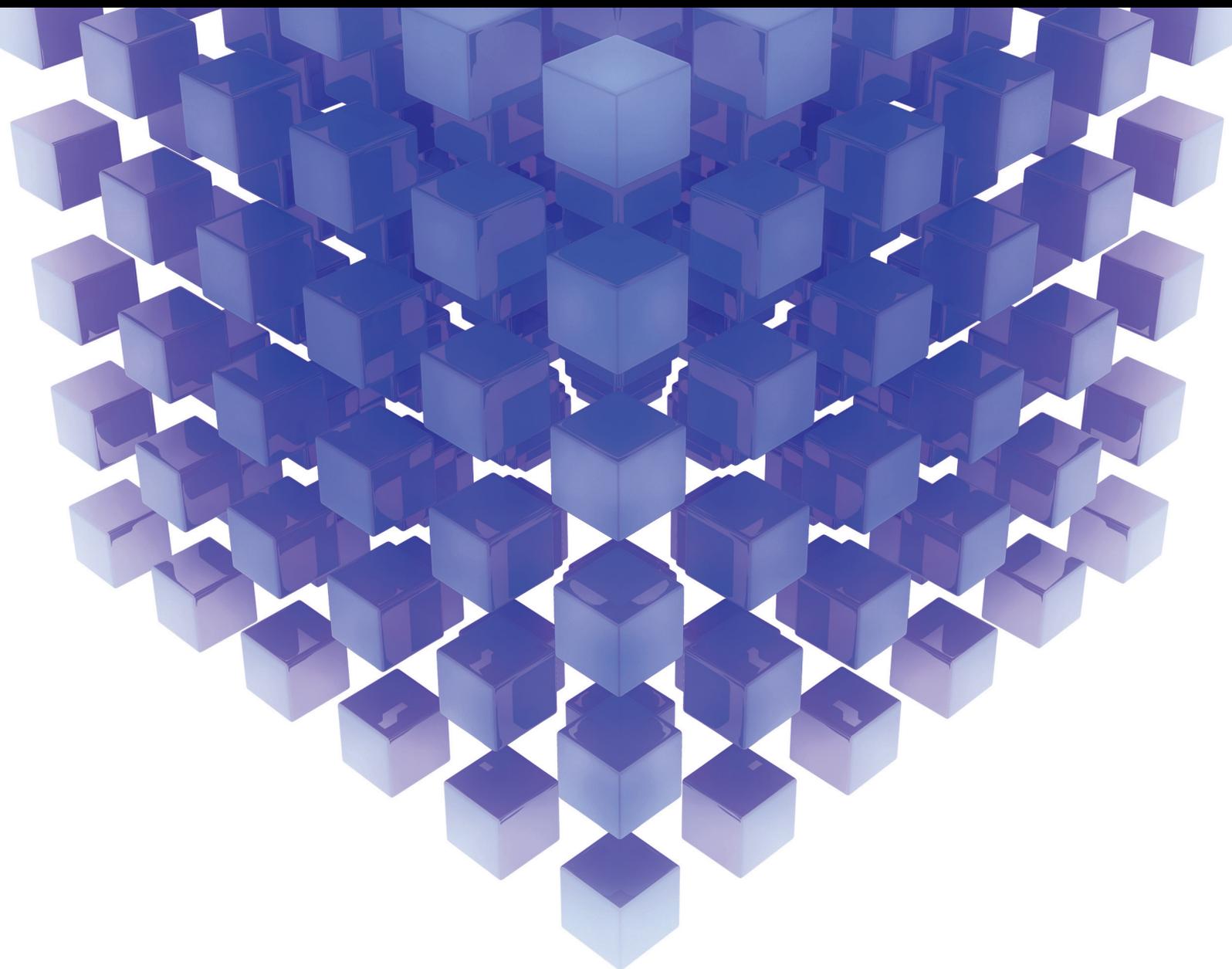


Mathematical Problems in Engineering

Robotics and Control Engineering of Wave and Tidal Energy-Recovering Systems

Special Issue Editor in Chief: Rafael M. Herrera

Guest Editors: José A. Somolinos, Carlos E. Ugalde-Loo, and José Gaspar





Robotics and Control Engineering of Wave and Tidal Energy-Recovering Systems

Mathematical Problems in Engineering

Robotics and Control Engineering of Wave and Tidal Energy-Recovering Systems

Special Issue Editor in Chief: Rafael M. Herrera

Guest Editors: José A. Somolinos, Carlos E. Ugalde-Loo,
and José Gaspar



Copyright © 2018 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
AITOUCHE Abdelouhab, France
Leonardo Acho, Spain
José A. Acosta, Spain
Paolo Adesso, Italy
Claudia Adduce, Italy
Ramesh Agarwal, USA
Francesco Aggogeri, Italy
Juan C. Agüero, Australia
R Aguilar-López, Mexico
Tarek Ahmed-Ali, France
Elias Aifantis, USA
Muhammad N. Akram, Norway
Guido Ala, Italy
Andrea Alaimo, Italy
Reza Alam, USA
Nicholas Alexander, UK
Salvatore Alfonzetti, Italy
Mohammad D. Aliyu, Canada
Juan A. Almendral, Spain
José Domingo Álvarez, Spain
Cláudio Alves, Portugal
J. P. Amezcua-Sanchez, Mexico
Lionel Amodeo, France
Sebastian Anita, Romania
Renata Archetti, Italy
Felice Arena, Italy
Sabri Arik, Turkey
Francesco Aristodemo, Italy
Fausto Arpino, Italy
Alessandro Arsie, USA
Edoardo Artioli, Italy
Fumihiko Ashida, Japan
Farhad Aslani, Australia
Mohsen Asle Zaem, USA
Romain Aubry, USA
Matteo Aureli, USA
Richard I. Avery, USA
Viktor Avrutin, Germany
Francesco Aymerich, Italy
Sajad Azizi, Belgium
Michele Baccocchi, Italy
Seungik Baek, USA
Adil Bagirov, Australia
Khaled Bahlali, France
Laurent Bako, France
Pedro Balaguer, Spain
Stefan Balint, Romania
Ines Tejado Balsera, Spain
Alfonso Banos, Spain
Jerzy Baranowski, Poland
Roberto Baratti, Italy
Andrzej Bartoszewicz, Poland
David Bassir, France
Chiara Bedon, Italy
Azeddine Beghdadi, France
Denis Benasciutti, Italy
Ivano Benedetti, Italy
Rosa M. Benito, Spain
Elena Benvenuti, Italy
Giovanni Berselli, Italy
Michele Betti, Italy
Jean-Charles Beugnot, France
Pietro Bia, Italy
Carlo Bianca, France
Simone Bianco, Italy
Vincenzo Bianco, Italy
Vittorio Bianco, Italy
Gennaro N. Bifulco, Italy
David Bigaud, France
Antonio Bilotta, Italy
Paul Bogdan, USA
Guido Bolognesi, UK
Rodolfo Bontempo, Italy
Alberto Borboni, Italy
Paolo Boscaroli, Italy
Daniela Boso, Italy
Guillermo Botella-Juan, Spain
Boulaïd Boulkroune, Belgium
Fabio Bovenga, Italy
Francesco Braghin, Italy
Ricardo Branco, Portugal
Maurizio Brocchini, Italy
Julien Bruchon, France
Matteo Bruggi, Italy
Michele Brun, Italy
Vasilis Burganos, Greece
Tito Busani, USA
Raquel Caballero-Águila, Spain
Filippo Cacace, Italy
Pierfrancesco Cacciola, UK
Salvatore Caddemi, Italy
Roberto Caldelli, Italy
Alberto Campagnolo, Italy
Eric Campos-Canton, Mexico
Marko Canadija, Croatia
Salvatore Cannella, Italy
Francesco Cannizzaro, Italy
Javier Cara, Spain
Ana Carpio, Spain
Caterina Casavola, Italy
Sara Casciati, Italy
Federica Caselli, Italy
Carmen Castillo, Spain
Inmaculada T. Castro, Spain
Miguel Castro, Portugal
Giuseppe Catalanotti, UK
Nicola Caterino, Italy
Alberto Cavallo, Italy
Gabriele Cazzulani, Italy
Luis Cea, Spain
Miguel Cerrolaza, Venezuela
M. Chadli, France
Gregory Chagnon, France
Ludovic Chamoin, France
Ching-Ter Chang, Taiwan
Qing Chang, USA
Michael J. Chappell, UK
Kacem Chehdi, France
Peter N. Cheimets, USA
Xinkai Chen, Japan
Luca Chiapponi, Italy
Francisco Chicano, Spain
Nicholas Chileshe, Australia
Adrian Chmielewski, Poland
Ioannis T. Christou, Greece
Hung-Yuan Chung, Taiwan
Simone Cinquemani, Italy
Roberto G. Citarella, Italy
Joaquim Ciurana, Spain
John D. Clayton, USA
Francesco Clementi, Italy

Piero Colajanni, Italy
Giuseppina Colicchio, Italy
Enrico Conte, Italy
Francesco Conte, Italy
Alessandro Contento, USA
Mario Cools, Belgium
José A. Correia, Portugal
Jean-Pierre Corriou, France
J.-C. Cortés, Spain
Carlo Cosentino, Italy
Paolo Crippa, Italy
Andrea Crivellini, Italy
Frederico R. B. Cruz, Brazil
Erik Cuevas, Mexico
M. C. Cunha, Portugal
Peter Dabnichki, Australia
Luca D'Acierno, Italy
Weizhong Dai, USA
Andrea Dall'Asta, Italy
Purushothaman Damodaran, USA
Farhang Daneshmand, Canada
Giuseppe D'Aniello, Italy
Sergey N. Dashkovskiy, Germany
Fabio De Angelis, Italy
Samuele De Bartolo, Italy
Pietro De Lellis, Italy
Luciano R. O. de Lima, Brazil
Alessandro De Luca, Italy
Stefano de Miranda, Italy
Filippo de Monte, Italy
Maria do Rosário de Pinho, Portugal
Michael Defoort, France
Alessandro Della Corte, Italy
Xavier Delorme, France
Laurent Dewasme, Belgium
Angelo Di Egidio, Italy
Roberta Di Pace, Italy
Ramón I. Diego, Spain
Yannis Dimakopoulos, Greece
Zhengtao Ding, UK
M. Djemai, France
Alexandre B. Dolgui, France
Georgios Dounias, Greece
Florent Duchaine, France
George S. Dulikravich, USA
Bogdan Dumitrescu, Romania
Horst Ecker, Austria
Saeed Eftekhari Azam, USA
Ahmed El Hajjaji, France
Antonio Elipe, Spain
Fouad Erchiqui, Canada
Anders Eriksson, Sweden
R. Emre Erkmen, Australia
Gilberto Espinosa-Paredes, Italy
Andrea L. Facci, Italy
Giacomo Falcucci, Italy
Giovanni Falsone, Italy
Hua Fan, China
Nicholas Fantuzzi, Italy
Yann Favennec, France
Fiorenzo A. Fazzolari, UK
Giuseppe Fedele, Italy
Roberto Fedele, Italy
Arturo J. Fernández, Spain
Jesus M. Fernandez Oro, Spain
Massimiliano Ferrara, Italy
Francesco Ferrise, Italy
Eric Feulvarch, France
Barak Fishbain, Israel
S. Douwe Flapper, Netherlands
Thierry Floquet, France
Eric Florentin, France
Alessandro Formisano, Italy
Francesco Franco, Italy
Elisa Francomano, Italy
Tomonari Furukawa, USA
Juan C. G. Prada, Spain
Mohamed Gadala, Canada
Matteo Gaeta, Italy
Mauro Gaggero, Italy
Zoran Gajic, USA
Erez Gal, Israel
Jaime Gallardo-Alvarado, Mexico
Ugo Galvanetto, Italy
Akemi Gálvez, Spain
Rita Gamberini, Italy
Maria L. Gandarias, Spain
Arman Ganji, Canada
Zhiwei Gao, UK
Zhong-Ke Gao, China
Giovanni Garcea, Italy
Luis Rodolfo Garcia Carrillo, USA
Jose M. Garcia-Aznar, Spain
Akhil Garg, China
Alessandro Gasparetto, Italy
Oleg V. Gendelman, Israel
Stylianos Georgantzinos, Greece
Fotios Georgiades, UK
Mergen H. Ghayesh, Australia
Georgios I. Giannopoulos, Greece
Agathoklis Giaralis, UK
Pablo Gil, Spain
Anna M. Gil-Lafuente, Spain
Ivan Giorgio, Italy
Gaetano Giunta, Luxembourg
Alessio Gizzi, Italy
Jefferson L.M.A. Gomes, UK
Emilio Gómez-Déniz, Spain
Antonio M. Gonçalves de Lima, Brazil
David González, Spain
Chris Goodrich, USA
Rama S. R. Gorla, USA
Nicolas Gourdain, France
Kannan Govindan, Denmark
Antoine Grall, France
George A. Gravvanis, Greece
Fabrizio Greco, Italy
Simonetta Grilli, Italy
Jason Gu, Canada
Federico Guarracino, Italy
Michele Guida, Italy
José L. Guzmán, Spain
Quang Phuc Ha, Australia
Petr Hájek, Czech Republic
Zhen-Lai Han, China
Thomas Hanne, Switzerland
Mohammad A. Hariri-Ardebili, USA
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
S. Hassan Hosseinnia, Netherlands
Mengqi Hu, USA
Gordon Huang, Canada
Sajid Hussain, Canada

Asier Ibeas, Spain
 Orest V. Iftime, Netherlands
 Przemyslaw Ignaciuk, Poland
 Giacomo Innocenti, Italy
 Emilio Insfran Pelozo, Spain
 Alessio Ishizaka, UK
 Nazrul Islam, USA
 Benoit Jung, France
 Benjamin Ivorra, Spain
 Payman Jalali, Finland
 Mahdi Jalili, Australia
 Łukasz Jankowski, Poland
 Samuel N. Jator, USA
 Juan C. Jauregui-Correa, Mexico
 Reza Jazar, Australia
 Khalide Jbilou, France
 Piotr Jędrzejowicz, Poland
 Isabel S. Jesus, Portugal
 Linni Jian, China
 Bin Jiang, China
 Zhongping Jiang, USA
 Emilio Jiménez Macías, Spain
 Ningde Jin, China
 Xiaoliang Jin, USA
 Liang Jing, Canada
 Dylan F. Jones, UK
 Palle E. Jorgensen, USA
 Vyacheslav Kalashnikov, Mexico
 Tamas Kalmar-Nagy, Hungary
 Tomasz Kapitaniak, Poland
 Julius Kaplunov, UK
 Haranath Kar, India
 Konstantinos Karamanos, Belgium
 Krzysztof Kecik, Poland
 Jean-Pierre Kenne, Canada
 Chaudry M. Khaliq, South Africa
 Do Wan Kim, Republic of Korea
 Nam-Il Kim, Republic of Korea
 Jan Koci, Czech Republic
 Ioannis Kostavelis, Greece
 Sotiris B. Kotsiantis, Greece
 Manfred Krafczyk, Germany
 Frederic Kratz, France
 Petr Krysl, USA
 Krzysztof S. Kulpa, Poland
 Shailesh I. Kundalwal, India
 Jurgen Kurths, Germany
 Kyandoghere Kyamakya, Austria
 Davide La Torre, Italy
 Risto Lahdelma, Finland
 Hak-Keung Lam, UK
 Giovanni Lancioni, Italy
 Jimmy Lauber, France
 Antonino Laudani, Italy
 Hervé Laurent, France
 Aimé Lay-Ekuakille, Italy
 Nicolas J. Leconte, France
 Dimitri Lefebvre, France
 Eric Lefevre, France
 Marek Lefik, Poland
 Yaguo Lei, China
 Kauko Leiviskä, Finland
 Thibault Lemaire, France
 Roman Lewandowski, Poland
 Chen-Feng Li, China
 Jian Li, China
 En-Qiang Lin, USA
 Zhiyun Lin, China
 Peide Liu, China
 Peter Liu, Taiwan
 Wanquan Liu, Australia
 Bonifacio Llamazares, Spain
 Alessandro Lo Schiavo, Italy
 Jean Jacques Loiseau, France
 Francesco Lolli, Italy
 Paolo Lonetti, Italy
 Sandro Longo, Italy
 Sebastian López, Spain
 Pablo Lopez-Crespo, Spain
 Luis M. López-Ochoa, Spain
 Ezequiel López-Rubio, Spain
 Vassilios C. Loukopoulos, Greece
 Jose A. Lozano-Galant, Spain
 Helen Lu, Australia
 Gabriel Luque, Spain
 Valentin Lychagin, Norway
 Antonio Madeo, Italy
 José María Maestre, Spain
 Alessandro Magnani, Italy
 Fazal M. Mahomed, South Africa
 Noureddine Manamanni, France
 Paolo Manfredi, Italy
 Didier Maquin, France
 Giuseppe Carlo Marano, Italy
 Damijan Markovic, France
 Francesco Marotti de Sciarra, Italy
 Rui Cunha Marques, Portugal
 Luis Martínez, Spain
 Rodrigo Martinez-Bejar, Spain
 Guiomar Martín-Herrán, Spain
 Denizar Cruz Martins, Brazil
 Benoit Marx, France
 Elio Masciari, Italy
 Franck Massa, France
 Paolo Massioni, France
 Alessandro Mauro, Italy
 Fabio Mazza, Italy
 Laura Mazzola, Italy
 Driss Mehdi, France
 Roderick Melnik, Canada
 Pasquale Memmolo, Italy
 Xiangyu Meng, USA
 Jose Merodio, Spain
 Alessio Merola, Italy
 Mahmoud Mesbah, Iran
 Luciano Mescia, Italy
 Laurent Mevel, France
 Mariusz Michta, Poland
 Leandro F. F. MIGUEL, Brazil
 Aki Mikkola, Finland
 Giovanni Minafò, Italy
 Hiroyuki Mino, Japan
 Pablo Mira, Spain
 Dimitrios Mitsotakis, New Zealand
 Vito Mocella, Italy
 Sara Montagna, Italy
 Roberto Montanini, Italy
 Francisco J. Montáns, Spain
 Luiz H. A. Monteiro, Brazil
 Gisele Mophou, France
 Rafael Morales, Spain
 Marco Morandini, Italy
 Javier Moreno-Valenzuela, Mexico
 Simone Morganti, Italy
 Caroline Mota, Brazil
 Aziz Moukrim, France
 Dimitris Mourtzis, Greece
 Emiliano Mucchi, Italy
 Josefa Mula, Spain
 Jose J. Muñoz, Spain
 Giuseppe Muscolino, Italy

Marco Mussetta, Italy
Hakim Naceur, France
Alessandro Naddeo, Italy
Hassane Naji, France
Mariko Nakano-Miyatake, Mexico
Keivan Navaie, UK
AMA Neves, Portugal
Luís C. Neves, UK
Dong Ngoduy, New Zealand
Nhon Nguyen-Thanh, Singapore
Tatsushi Nishi, Japan
Xesús Nogueira, Spain
Ben T. Nohara, Japan
Mohammed Nouari, France
Mustapha Nourelfath, Canada
Włodzimierz Ogryczak, Poland
Roger Ohayon, France
Krzysztof Okarma, Poland
Mitsuhiro Okayasu, Japan
Alberto Olivares, Spain
Enrique Onieva, Spain
Calogero Orlando, Italy
Alejandro Ortega-Moñux, Spain
Sergio Ortobelli, Italy
Naohisa Otsuka, Japan
Erika Ottaviano, Italy
Pawel Packo, Poland
Arturo Pagano, Italy
Alkis S. Paipetis, Greece
Roberto Palma, Spain
Alessandro Palmeri, UK
Pasquale Palumbo, Italy
Jürgen Pannek, Germany
Elena Panteley, France
Achille Paolone, Italy
George A. Papakostas, Greece
Xosé M. Pardo, Spain
Vicente Parra-Vega, Mexico
Manuel Parra, Spain
Petr Páta, Czech Republic
Pubudu N. Pathirana, Australia
Surajit K. Paul, India
Sitek Paweł, Poland
Luis Payá, Spain
Alexander Paz, USA
Igor Pažanin, Croatia
Libor Pekař, Czech Republic

Francesco Pellicano, Italy
Marcello Pellicciari, Italy
Haipeng Peng, China
Mingshu Peng, China
Zhengbiao Peng, Australia
Zhi-ke Peng, China
Marzio Pennisi, Italy
Maria Patrizia Pera, Italy
Matjaz Perc, Slovenia
A. M. Bastos Pereira, Portugal
Ricardo Perera, Spain
Francesco Pesavento, Italy
Ivo Petras, Slovakia
Francesco Petrini, Italy
Lukasz Pieczonka, Poland
Dario Piga, Switzerland
Paulo M. Pimenta, Brazil
Antonina Pirrotta, Italy
Marco Pizzarelli, Italy
Vicent Pla, Spain
Javier Plaza, Spain
Dragan Poljak, Croatia
Jorge Pomares, Spain
Sébastien Poncet, Canada
Volodymyr Ponomaryov, Mexico
Jean-Christophe Ponsart, France
Mauro Pontani, Italy
Cornelio Posadas-Castillo, Mexico
Francesc Pozo, Spain
Christopher Pretty, New Zealand
Luca Pugi, Italy
Krzysztof Puszynski, Poland
Giuseppe Quaranta, Italy
Vitomir Racic, Italy
Jose Ragot, France
Carlo Rainieri, Italy
K. Ramamani Rajagopal, USA
Ali Ramazani, USA
Higinio Ramos, Spain
Alain Rassineux, France
S.S. Ravindran, USA
Alessandro Reali, Italy
Jose A. Reinoso, Spain
Oscar Reinoso, Spain
Fabrizio Renno, Italy
Nidhal Rezg, France
Ricardo Rianza, Spain

Francesco Riganti-Fulginei, Italy
Gerasimos Rigatos, Greece
Francesco Ripamonti, Italy
Jorge Rivera, Mexico
Eugenio Roanes-Lozano, Spain
Bruno G. M. Robert, France
Ana Maria A. C. Rocha, Portugal
José Rodellar, Spain
Luigi Rodino, Italy
Rosana Rodríguez López, Spain
Ignacio Rojas, Spain
Alessandra Romolo, Italy
Debasish Roy, India
Gianluigi Rozza, Italy
Rubén Ruiz, Spain
Antonio Ruiz-Cortes, Spain
Ivan D. Rukhlenko, Australia
Mazen Saad, France
Kishin Sadarangani, Spain
Andrés Sáez, Spain
Mehrddad Saif, Canada
John S. Sakellariou, Greece
Salvatore Salamone, USA
Vicente Salas, Spain
Jose V. Salcedo, Spain
Nunzio Salerno, Italy
Miguel A. Salido, Spain
Roque J. Saltarén, Spain
Alessandro Salvini, Italy
Sylwester Samborski, Poland
Ramon Sancibrian, Spain
Giuseppe Sanfilippo, Italy
Miguel A. F. Sanjuan, Spain
Vittorio Sansalona, France
José A. Sanz-Herrera, Spain
Nickolas S. Sapidis, Greece
Evangelos J. Sapountzakis, Greece
Luis Saucedo-Mora, Spain
Marcelo A. Savi, Brazil
Andrey V. Savkin, Australia
Roberta Sburlati, Italy
Gustavo Scaglia, Argentina
Thomas Schuster, Germany
Oliver Schütze, Mexico
Lotfi Senhadji, France
Junwon Seo, USA
Joan Serra-Sagrasta, Spain

Gerardo Severino, Italy
Ruben Sevilla, UK
Stefano Sfarra, Italy
Mohamed Shaat, Egypt
Mostafa S. Shadloo, France
Leonid Shaikhet, Israel
Hassan M. Shanechi, USA
Bo Shen, Germany
Suzanne M. Shontz, USA
Babak Shotorban, USA
Zhan Shu, UK
Nuno Simões, Portugal
Christos H. Skiadas, Greece
Konstantina Skouri, Greece
Neale R. Smith, Mexico
Bogdan Smolka, Poland
Delfim Soares Jr., Brazil
Alba Sofi, Italy
Giovanni Solari, Italy
Francesco Soldovieri, Italy
Raffaale Solimene, Italy
Jussi Sopanen, Finland
Marco Spadini, Italy
Bernardo Spagnolo, Italy
Paolo Spagnolo, Italy
Ruben Specogna, Italy
Vasilios Spitas, Greece
Sri Sridharan, USA
Ivanka Stamova, USA
Rafał Stanisławski, Poland
Florin Stoican, Romania
Salvatore Strano, Italy
Yakov Strelniker, Israel
Guang-Yong Sun, China
Sergey A. Suslov, Australia
Thomas Svensson, Sweden
Andrzej Swierniak, Poland
Andras Szekrenyes, Hungary
Kumar K. Tamma, USA
Yang Tang, Germany
Hafez Tari, USA
Alessandro Tasora, Italy
Sergio Teggi, Italy
Ana C. Teodoro, Portugal

Tai Thai, Australia
Alexander Timokha, Norway
Gisella Tomasini, Italy
Francesco Tornabene, Italy
Antonio Tornambe, Italy
Javier Martinez Torres, Spain
Mariano Torrisi, Italy
George Tsiatas, Greece
Antonios Tsourdos, UK
Federica Tubino, Italy
Nerio Tullini, Italy
Andrea Tundis, Italy
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
Carlos-Renato Vázquez, Mexico
Miguel E. Vázquez-Méndez, Spain
Josep Vehi, Spain
Martin Velasco Villa, Mexico
K. C. Veluvolu, Republic of Korea
Fons J. Verbeek, Netherlands
Franck J. Vernerey, USA
Georgios Veronis, USA
Vincenzo Vespri, Italy
Renato Vidoni, Italy
V. Vijayaraghavan, Australia
Anna Vila, Spain
Rafael J. Villanueva, Spain
Francisco R. Villatoro, Spain
Uchechukwu E. Vincent, UK
Gareth A. Vio, Australia
Francesca Vipiana, Italy
Stanislav Vitek, Czech Republic
Thuc P. Vo, UK
Jan Vorel, Czech Republic

Michael Vynnycky, Sweden
Hao Wang, USA
Liliang Wang, UK
Shuming Wang, China
Yongqi Wang, Germany
Roman Wan-Wendner, Austria
Jaroslaw Wąs, Poland
P.H. Wen, UK
Waldemar T. Wójcik, Poland
Changzhi Wu, Australia
Desheng D. Wu, Sweden
Hong-Yu Wu, USA
Yuqiang Wu, China
Michalis Xenos, Greece
Guangming Xie, China
Xue-Jun Xie, China
Gen Q. Xu, China
Hang Xu, China
Joseph J. Yame, France
Xinggang Yan, UK
Mijia Yang, USA
Yongheng Yang, Denmark
Luis J. Yebra, Spain
Peng-Yeng Yin, Taiwan
Qin Yuming, China
Elena Zaitseva, Slovakia
Arkadiusz Żak, Poland
Daniel Zaldivar, Mexico
Francesco Zammori, Italy
Vittorio Zampoli, Italy
Rafał Zdunek, Poland
Ibrahim Zeid, USA
Huaguang Zhang, China
Kai Zhang, China
Qingling Zhang, China
Xian-Ming Zhang, Australia
Xuping Zhang, Denmark
Zhao Zhang, China
Yifan Zhao, UK
Jian G. Zhou, UK
Quanxin Zhu, China
Mustapha Zidi, France
Gaetano Zizzo, Italy

Contents

Robotics and Control Engineering of Wave and Tidal Energy-Recovering Systems

Rafael Morales , José A. Somolinos, Carlos E. Ugalde-Loo, and José Gaspar
Editorial (2 pages), Article ID 6594517, Volume 2018 (2018)

A Multiagent Architecture for Mobile Robot Navigation Using Hierarchical Fuzzy and Sliding Mode Controllers

Dorra Ayedi , Maïssa Boujelben, and Chokri Rekik
Research Article (11 pages), Article ID 9315925, Volume 2018 (2018)

Proportional Load Sharing and Stability of DC Microgrid with Distributed Architecture Using SM Controller

Muhammad Rashad , Uzair Raof, Muhammad Ashraf, and Bilal Ashfaq Ahmed
Research Article (16 pages), Article ID 2717129, Volume 2018 (2018)

Design, Analysis, and Experiment for Rescue Robot with Wheel-Legged Structure

Meng Ning, Bilun Xue, Zefeng Ma, Changhong Zhu, Zihao Liu, Cuncai Zhang, Yao Wang, and QiuJu Zhang
Research Article (16 pages), Article ID 5719381, Volume 2017 (2018)

Fuzzy Logic Controller Design for Intelligent Robots

Ching-Han Chen, Chien-Chun Wang, Yi Tun Wang, and Po Tung Wang
Research Article (12 pages), Article ID 8984713, Volume 2017 (2018)

Inverse Kinematic Solutions of Dual Redundant Camera Robot Based on Genetic Algorithm

Libo Zhang, Jingjie He, and Su Wang
Research Article (10 pages), Article ID 7486178, Volume 2017 (2018)

Robust Course Keeping Control of a Fully Submerged Hydrofoil Vessel without Velocity Measurement: An Iterative Learning Approach

Sheng Liu, Changkui Xu, and Lanyong Zhang
Research Article (14 pages), Article ID 7979438, Volume 2017 (2018)

Editorial

Robotics and Control Engineering of Wave and Tidal Energy-Recovering Systems

Rafael Morales ¹, José A. Somolinos,² Carlos E. Ugalde-Loo,³ and José Gaspar⁴

¹Escuela de Ingenieros Industriales de Albacete, Universidad de Castilla-La Mancha, 02071 Albacete, Spain

²Escuela de Ingenieros Navales, Universidad Politécnica de Madrid, Madrid, Spain

³Cardiff School of Engineering, Cardiff, UK

⁴Centre for Marine Technology and Ocean Engineering (CENTEC), Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

Correspondence should be addressed to Rafael Morales; rafael.morales@uclm.es

Received 5 February 2018; Accepted 5 February 2018; Published 13 May 2018

Copyright © 2018 Rafael Morales 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.

Energy is a vital element for human life, and if modern societies are to be sustained, then obtaining a secure, sufficient, and accessible energy supply is fundamental. The demand for the provision of energy is rapidly increasing throughout the world and this trend is likely to continue in the future. The growing recognition that global warming exists has led more governments, research centers, and corporations to commit resources to the advancement of renewable energy technologies.

Renewable energy generation could, in addition to providing a means to substantially reduce CO₂ emissions, help reduce national dependencies on imported energy, thus increasing energy security and diminishing domestic supplies of fossil fuels. One challenge as regards energy is the ability to move to a new low carbon economy in which energy demands can be met while the levels of CO₂ emitted are reduced. In order to meet this challenge, it will be necessary to exploit other forms of renewable energy that are currently less developed but which have a high potential, such as marine renewable energy. This had led to an incipient interest in the successful development of these technologies and in bringing them onto the market.

The sea is a huge collector, accumulator, and transformer of clean and inexhaustible energy and is virtually unexploited, despite covering more than 70% of the Earth's surface. Marine currents are, together with waves, the most promising sources of marine renewable energies. Some of the opportunities and benefits that could be achieved with the exploitation of wave and tidal energy-recovering systems include energy independence, decarbonization, or the creation of jobs. The

energy that could be extracted from sea waves and currents is estimated as being more than 8,000 TWh/year.

In spite of the opportunities that these marine renewable energies could provide, there are still some challenges that must be met if an awareness of marine technologies is to be promoted and their current potential increased. The difficulties related to this medium have led to the conception, design, and operation of devices with exceptional features, such as stability, redundancy, seakeeping, or survival in hostile conditions. Meeting these challenges is essential if governments, industry, maritime and oceanic services, research institutions, and universities are to obtain a unified and coordinated approach to achieve robust, feasible, and cost-effective marine technologies. Obtaining these will help the acceleration and sustainability of marine systems. In this respect, the development of new and innovative solutions in robotics and control engineering systems applied to wave and tidal energy-recovering systems has received a great deal of attention in the last years from a considerable number of researchers and from the industrial community.

The objective of this special issue is to show the latest research achievements, findings, and ideas in the fields of robotics and control engineering as regards wave and tidal energy-recovering systems. The topics dealt with in this special issue are the following:

- (i) *Sensor systems*: sensors and sensor networks, intelligent sensors, sensor uncertainty for fault tolerant control, and distributed and multimodality sensor

network for control and automation, to name but a few.

- (ii) *Control*: adaptive control, robust control, active disturbance rejection control, identification and estimation, delay systems, precision motion control, and so forth.
- (iii) *Mathematical modeling*: modeling, identification, and simulation of wave and tidal energy-recovering systems.
- (iv) *Robotics*: modeling and identification, mobile robotics, mobile sensor networks, perception systems, visual servoing, robot sensing and data fusion, and autonomous and remotely operated (surface and underwater) systems.
- (v) *Industrial informatics*: embedded systems for monitoring and controlling wave and tidal energy-recovering systems.

Acknowledgments

We would like to thank all the authors for their excellent contributions and also the reviewers for their valuable help.

Rafael Morales
José A. Somolinos
Carlos E. Ugalde-Loo
José Gaspar

Research Article

A Multiagent Architecture for Mobile Robot Navigation Using Hierarchical Fuzzy and Sliding Mode Controllers

Dorra Ayedi ¹, Maïssa Boujelben,² and Chokri Reki²

¹Control and Energy Management Lab (CEM Lab), Sousse Engineering School, University of Sousse, BP 264, Sousse Erriadh, 4023 Sousse, Tunisia

²Control and Energy Management Lab (CEM Lab), Sfax Engineering School, University of Sfax, BP W, 3038 Sfax, Tunisia

Correspondence should be addressed to Dorra Ayedi; dorra.ayedii@yahoo.fr

Received 29 September 2017; Revised 18 December 2017; Accepted 16 January 2018; Published 13 February 2018

Academic Editor: Rafael Morales

Copyright © 2018 Dorra Ayedi 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 challenge of this work is to implement an algorithm which enables the robot to achieve independent activities in the purpose of achieving a common goal, which consists in autonomous navigation in a partially unknown environment. The use of multiagent system is convenient for such a problem. Hence, we have designed a structure composed of four agents dedicated to perception, navigation, static, and dynamic obstacle avoidance. Those agents interact through a coordination system.

1. Introduction

In recent years, multiagent system becomes a very well-known field used in many applications such as E-commerce, E-health applications, network intrusion detection systems, telematic and transport systems [1], and robotic system [2, 3], which is the topic of this paper.

Indeed, several researchers have used multiagent system in the development of robotic system [4]. Multiagent system is very useful in the case of multirobots which execute different tasks, which should be coordinated with each other [5].

Furthermore, such architecture can be used in the case of a single robot. According to the principle of the multiagent system, which consists in a number of agents coordinated and dedicated to achieve common objectives, each robot activity is considered as an agent. Those agents are interacting with each other through a specific system of coordination.

The interest in the multiagent system has increased, since it has several properties compared to other architectures [6]. Among those properties, we mention decentralized control, which permits the continuity of the system even when some parts fail, coordinability, predictability, and adaptability [2, 3].

Therefore, the structure of multiagent system may be established as a first step. Then, the activity of each agent

is defined. Finally, a coordination system between agents is presented. Those steps are applied to implement the multiagent system on an intelligent wheelchair in [7]. Five agents were designed by Busquets et al. for robot navigation [8]. Innocenti et al. have proposed in [2] a multiagent structure containing four agents, combined with each other in order to obtain the desired goal. Boujelben et al. have simplified this architecture and reduced the number of agents to three, dedicated to perception, navigation, and obstacle avoidance [3].

Among the agents defined in the case of robotic field, we cite navigation agent and obstacle avoidance agent. In the literature, the design of controllers for mobile robot navigation is based on two basic approaches: the reactive and the deliberative approach [9].

The deliberative method is applied when the situation needs a previous knowledge of the environment [10]. Path planning is the principle of this approach. Researchers have developed methods of control based on deliberative approach such as PID control [11] and sliding mode [12].

For an unknown environment, the deliberative approach is not sufficient. Therefore, the reactive method is required, which is based on sensor's measurements and does not require information about the environment [9]. In the literature, neural network [13], potential field [14], and

fuzzy logic system [15] are the most popular developed methods.

In this work, we have treated the problem of an environment containing both static and dynamic obstacles. Our mobile robot should find a safe path to reach the desired target. For that purpose, we have proposed to apply the multiagent system to control the robot. Our suggested structure contains four agents:

- (i) Perception agent, which obtains the necessary information measured by the sensors of the robot and affords them to the system
- (ii) Navigation agent, responsible for guiding the robot to the target
- (iii) Static obstacle avoidance, which keeps the robot away from static obstacles
- (iv) Dynamic obstacle avoidance, which keeps the robot away from dynamic obstacles

Those agents are related through a cooperative system which allows them to achieve common goals.

Here, the navigation agent uses a simple fuzzy controller. Indeed, Chia-Feng Juang has tested the efficiency of fuzzy logic system in the field of robot navigation [16], comparing the performance of this method, while introducing some optimization approaches. In this work, fuzzy logic system was implemented in the navigation agent for its high robustness and good performances, without applying optimization methods.

For the static obstacle avoidance agents, a deliberative method is applied, since the position of obstacles is known. Sliding mode is chosen to be implemented on this agent thanks to its fast response and its robustness against variations. For the dynamic obstacles, the position of obstacles is unknown and unexpected. The deliberative approach is not sufficient, due to the complicated computation of the following path in the presence of dynamic obstacle. Fuzzy logic system is applied, since it is considered as a reactive method.

However, a large number of inputs are obtained in the case of applying the standard fuzzy logic system, due to the high quantity of information given by all robot's sensors. To simplify the problem, we have proposed to use the hierarchical fuzzy system which consists in dividing the fuzzy system into subsystems [15].

This paper is organized as follows. In Section 2, the model of the used robot is presented. Section 3 introduces the adopted multiagent architecture, detailing the role of each agent. Coordination between agents is mentioned in Section 4. Finally, Section 5 presents the simulation results to test the validity of this work.

2. Modeling of the Mobile Robot

The platform used in this work is a Khepera II robot. This mobile robot is convenient for our experiments, due to its equipment consisting on two independent wheels and eight sensors attached to its contour which ensure a best detection to the obstacles, as it is shown in Figure 1 [17].

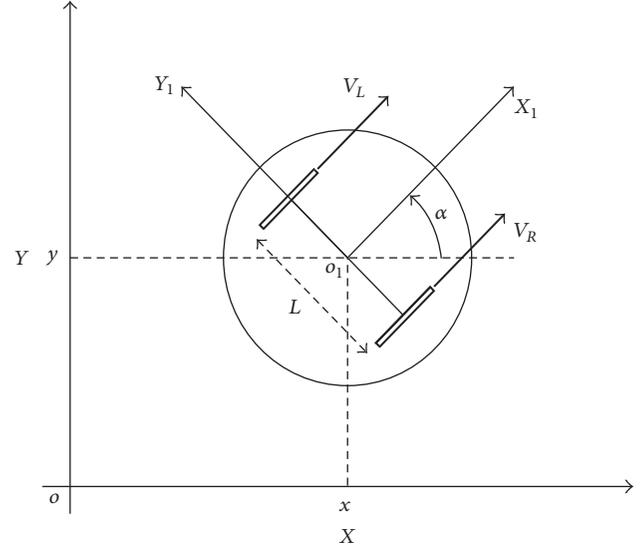


FIGURE 1: The schematic model of the Khepera II robot.

The linear right and left wheels velocities V_R and V_L as well as the angle α between the robot direction and the x -axis are responsible for guiding the robot, using the following kinematic model:

$$\begin{aligned}\dot{x} &= \frac{V_R + V_L}{2} \cos \alpha, \\ \dot{y} &= \frac{V_R + V_L}{2} \sin \alpha, \\ \dot{\alpha} &= \frac{V_R - V_L}{L},\end{aligned}\quad (1)$$

where L is the distance between the right and the left wheels.

The adopted strategy is to control the robot to a final destination in an environment containing both static and dynamic obstacles using the multiagent system.

3. Multiagent Architecture Model

The basic structure of the multiagent model is presented in Figure 2.

It is composed of four agents which interact with each other through a cooperative system. This system of coordination will be detailed in Section 5.

The information provided by the sensor's measurements is collected in the perception agent. According to those data, one of the three other agents (robot navigation, static obstacle avoidance, and dynamic obstacle avoidance) is chosen to control the robot. If the environment is safe, the robot navigation agent is activated, which is responsible for guiding the robot from an initial position to a final target. If an obstacle is located on the path of the robot, one of the obstacle avoidance agents is activated. Those two agents are in charge of moving away from obstacles, one of them applying a reactive method and the second using a deliberative approach [18]. The choice between those two agents depends on type of obstacle faced (static, dynamic).

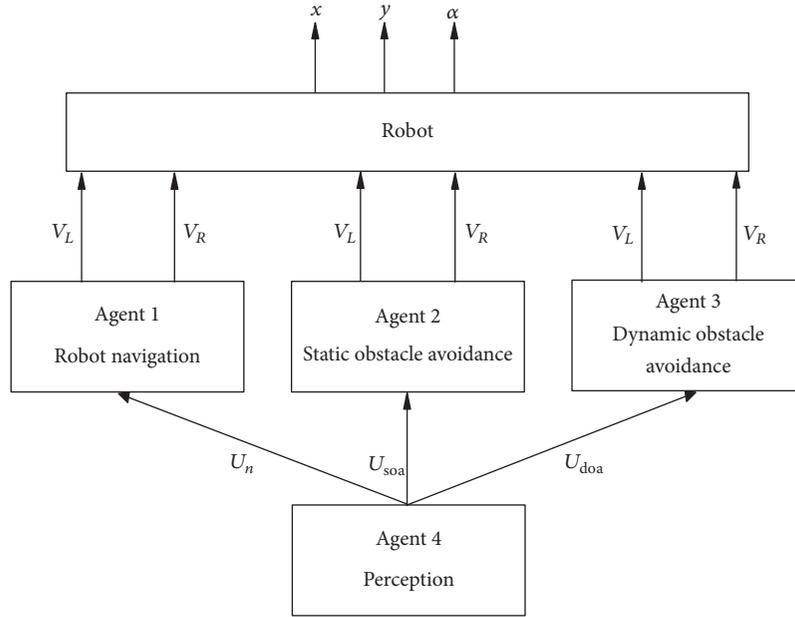
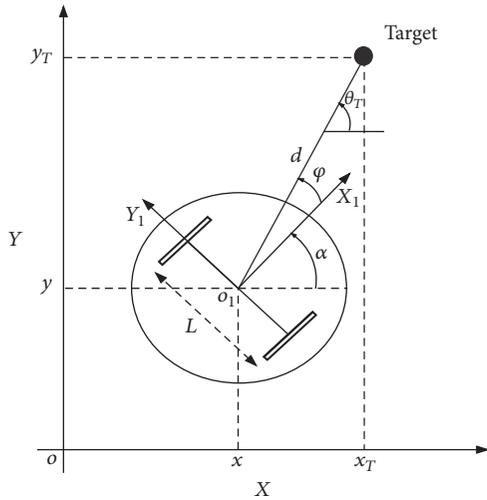


FIGURE 2: The multiagent system architecture.


 FIGURE 3: Representation of the parameters d and φ .

3.1. Navigation Agent. The role of the navigation agent, based on fuzzy logic control, is to bring the robot to a final target when the environment is supposed to be free from obstacles [3]. This task is achieved by calculating the distance d separating the robot center from the goal and the angle φ which separates the distance to the target and the robot orientation (see Figure 3), then giving them as inputs to a fuzzy logic controller [19]. This controller is responsible for the function of navigation.

The distance d and the angle φ are calculated using those following expressions:

$$d = \sqrt{(x_T - x)^2 + (y_T - y)^2}, \quad (2)$$

$$\varphi = \theta_T - \alpha_R \quad (3)$$

with

$$\theta_T = \arctan \frac{(y_T - y)}{(x_T - x)}. \quad (4)$$

Fuzzy logic system is a well-known technique recommended for robust controller's conception [20]. Its advantage consists in considering the variation field of input variables, unlike Boolean logic in which the values of the variables are 0 or 1 [3, 21].

Figures 4 and 5 show the fuzzy partition of the input variables d and φ . The distance d varies in the range [0, 700] mm, whereas the interval of the angle φ is defined between $-\pi/2$ and $\pi/2$. Five membership functions are used for the distance, and seven memberships functions are considered for the angle. Therefore, 35 rules are obtained. More details and inference tables are provided in [15, 21].

An example of a fuzzy rule which represents a link between fuzzy inputs variables (d and φ) and fuzzy outputs variables (V_L and V_R) is expressed as follows.

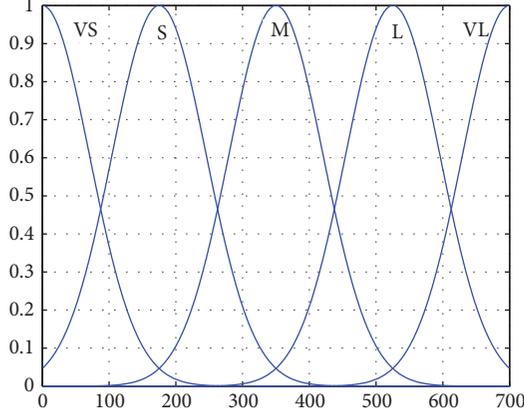
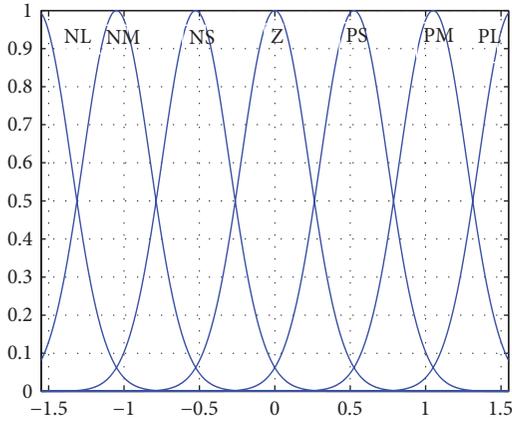
If d is A_i and φ is B_j , then $V_R = y_i$ and $V_L = z_i$, where α_i and β_j are the level activation of the rule i .

Therefore, the expressions of the output variables are

$$V_R = \frac{\sum_{i=1}^r \alpha_i y_i}{\sum_{j=1}^r \alpha_j}, \quad (5)$$

$$V_L = \frac{\sum_{i=1}^r \beta_i z_i}{\sum_{j=1}^r \beta_j}.$$

3.2. Static Obstacle Avoiding Agent. Now, we study the case of the existence of a static obstacle in the path of the robot. To avoid it, a deliberative approach is applied, which consists in defining a trajectory that the robot should follow [22, 23]. The proposed approach is based on following a specific trajectory

FIGURE 4: Membership functions for d (in mm).FIGURE 5: Membership functions for φ (in rad).

generated by limit cycle equations applying the sliding mode control [22]. Some details about the limit cycle trajectory and the sliding mode control are presented in the following subsections.

3.2.1. Limit Cycle Trajectory. First of all, we define a convergence circle around each static obstacle. According to previous works [22, 24], this circle corresponds to a dangerous zone. Its radius is defined as $R_c = R_r + R_{os} + \delta$ (see Figure 6), where

- (i) d is the distance between the robot center and the target,
- (ii) d_{os} is the distance between the robot center and the static obstacle,
- (iii) R_r is robot's radius,
- (iv) R_{os} is static obstacle's radius,
- (v) δ is a safety margin for collision avoidance.

Now we consider the following differential equation, which defines the limit cycle trajectory of the robot [25]. This system of equation permits us to obtain the position (x_r, y_r)

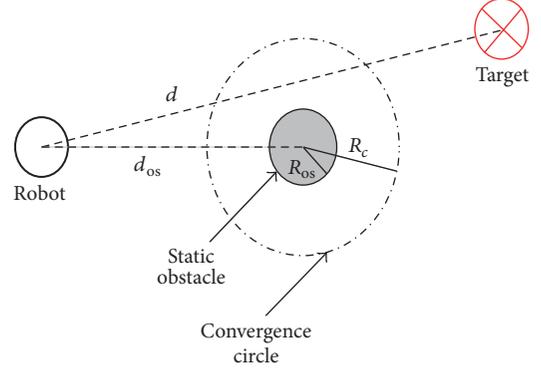


FIGURE 6: The convergence circle.

of the robot enabling it to track the convergence cycle defined previously [22]:

$$\begin{aligned} \dot{x}_r &= ay_r + x_r (R_c^2 - x_r^2 - y_r^2), \\ \dot{y}_r &= -ax_r + y_r (R_c^2 - x_r^2 - y_r^2), \end{aligned} \quad (6)$$

where a is a scalar which specifies the orbital path direction (clockwise or counterclockwise), according to the position of the robot with respect to the obstacle. If $a = 1$, the rotational direction is clockwise. If $a = -1$, this direction is counterclockwise. Figure 7 shows both cases with different initial conditions.

In the purpose of determining the appropriate direction, a new reference frame is defined $S = (C, X_o, Y_o)$ centered on the obstacle, as it is presented in Figure 8.

Applying this reference, the new coordinates of the robot are obtained (x_{or}, y_{or}) [24]. Then the sign of y_{or} determines the motion direction (clockwise if $y_{or} > 0$ and counterclockwise otherwise).

3.2.2. Sliding Mode Control. Once the reference trajectory has been determined, the approach proposed for tracking this path is the sliding mode control. This method ensures fast response, robustness, and good results in tracking the desired trajectory [26].

This method is based on choosing the appropriate sliding surface denoted by s along which the sliding motion occurs [12, 26].

To illustrate the principle of the sliding mode control, we consider Figure 9, where the tracking error $p_e = (x_e, y_e, \alpha_e)^T$ between the reference portion $p_r = (x_r, y_r, \alpha_r)^T$ and the current position $p = (x, y, \alpha)^T$ of the robot is represented and expressed by

$$\begin{bmatrix} x_e \\ y_e \\ \alpha_e \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_r - x \\ y_r - y \\ \alpha_r - \alpha \end{bmatrix}. \quad (7)$$

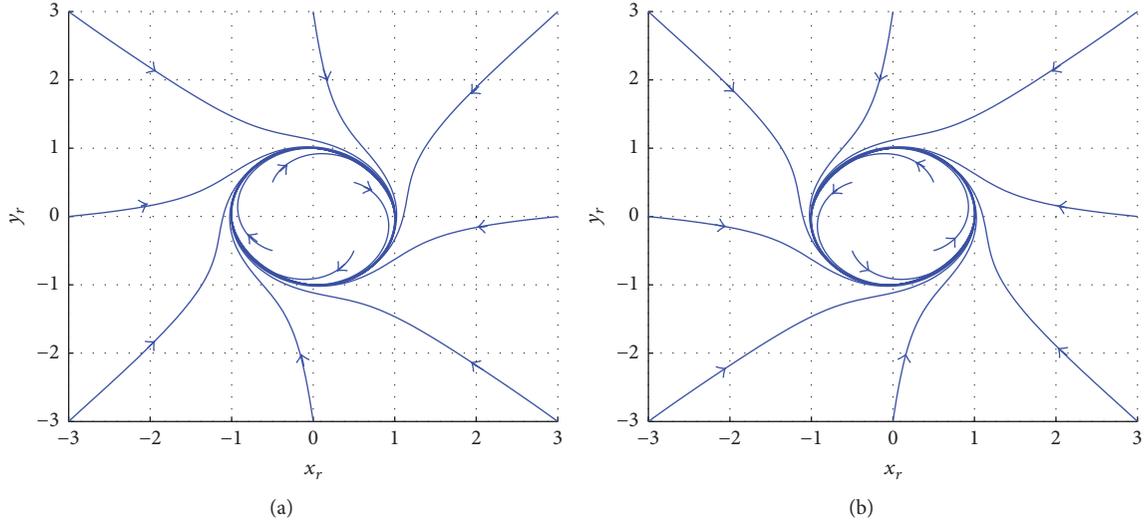
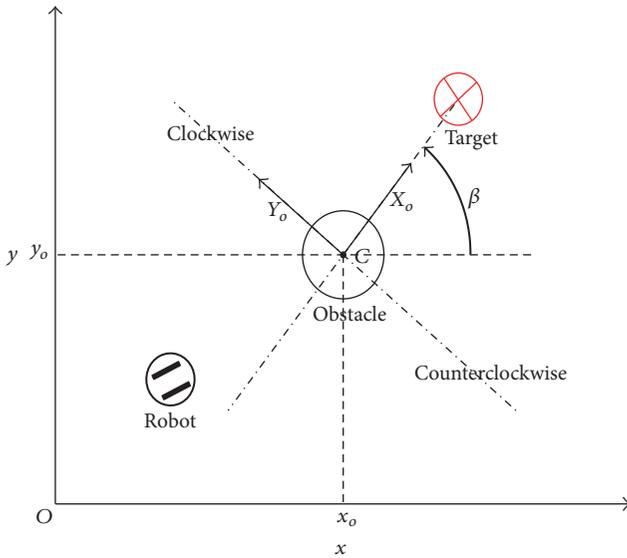


FIGURE 7: The shape of the limit cycles, (a) clockwise and (b) counterclockwise.


 FIGURE 8: The new reference frame $S = (C, X_o, Y_o)$.

By deriving this system of equations, we obtain

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\alpha}_e \end{bmatrix} = \begin{bmatrix} y_e w - v + v_r \cos \alpha_e \\ -x_e w + v_r \sin \alpha_e \\ w_r - w \end{bmatrix}. \quad (8)$$

The error portion p_e must converge to 0, which means that p converges to p_r . To achieve this purpose, the adequate linear and angular velocities v and w should be found.

The design of the switching function is a hard task due to the robot model which is a multiple-input nonlinear system [22]. Lee et al. have proposed in [12] to consider $x_e = 0$. As a consequence, the Lyapunov equation is chosen as

$$V = \frac{1}{2} y_e^2. \quad (9)$$

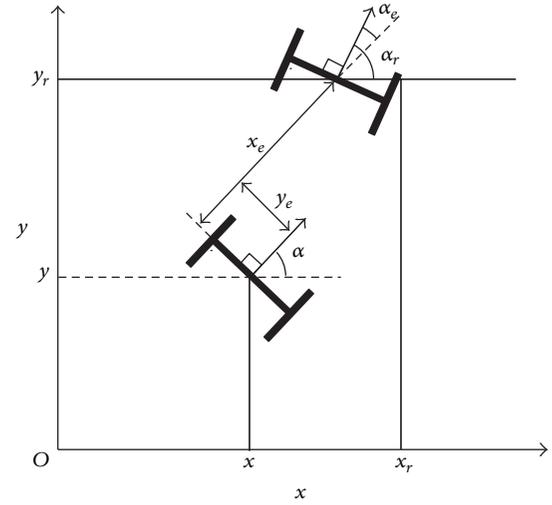


FIGURE 9: Tracking error.

By deriving V , we obtain

$$\begin{aligned} \dot{V} &= y_e \dot{y}_e = y_e (-x_e w + v_r \sin(\alpha_e)) \\ &= -x_e y_e w - v_r y_e \sin(\arctan(v_r y_e)), \end{aligned} \quad (10)$$

where $\alpha_e = -\arctan(v_r y_e)$ is the switching function condition.

α_e is always positive. Therefore, the global reaching condition given as $\dot{V} \leq 0$ is always satisfied [22]. Then we define the vector of sliding surfaces as

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_e \\ \alpha_e + \arctan(v_r y_e) \end{bmatrix}. \quad (11)$$

If we force s to converge to 0, consequently $x_e \rightarrow 0$ and $\alpha_e \rightarrow -\arctan(v_r y_e)$, which leads to $y_e \rightarrow 0$.

To reduce the problem of chattering [12], we replace the switching function by a saturation function defined as [27]

$$\dot{s} = -k \text{sat}(s). \quad (12)$$

By substituting $\arctan(v_r y_e)$ with γ , we obtain

$$\begin{aligned} \dot{s} &= \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} = \begin{bmatrix} -k_1 \text{sat}(s_1) \\ -k_2 \text{sat}(s_2) \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ \dot{\alpha}_e + \frac{\partial \gamma}{\partial v_r} \dot{v}_r + \frac{\partial \gamma}{\partial y_e} \dot{y}_e \end{bmatrix} \\ &= \begin{bmatrix} y_e \omega - v + v_r \cos \alpha_e \\ w_r - w + \frac{\partial \gamma}{\partial v_r} \dot{v}_r + \frac{\partial \gamma}{\partial y_e} (-x_e \omega + v_r \sin \alpha_e) \end{bmatrix}. \end{aligned} \quad (13)$$

The obtained control law is

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} y_e \omega + v_r \cos \alpha_e + k_1 \text{sat}(s_1) \\ w_r + \frac{\partial \gamma}{\partial v_r} \dot{v}_r + \frac{\partial \gamma}{\partial y_e} (v_r \sin \alpha_e) + k_2 \text{sat}(s_2) \end{bmatrix} \quad (14)$$

with $\partial \gamma / \partial v_r = y_e / (1 + (v_r y_e)^2)$ and $\partial \gamma / \partial y_e = v_r / (1 + (v_r y_e)^2)$.

3.3. Dynamic Obstacle Avoiding Agent. In the following section, we consider that the robot faces on his path dynamic obstacles. For its safety, the robot must know at every instance the position of the obstacle to avoid it [15]. The idea is to calculate at each moment the distance d expressed in (2). If this distance is lower than 50 mm, we apply the hierarchical fuzzy system as a reactive approach for obstacle avoidance.

We have chosen to work with the hierarchical fuzzy system rather than the standard fuzzy system due to the huge number of rules obtained if we apply one fuzzy controller [17, 21]. The principle of this method consists in decomposing the fuzzy system into subsystems having lower size and related to each other in a hierarchical way. This structure is presented in Figure 10.

According to this block diagram, each controller from the first layer is related to a sensor of the Khepera II robot. This sensor is required to send the distance d_i and the angle φ_i ($i = 1, \dots, 8$) to an i th controller, which gives as inputs the left and right wheels velocities V_{Li} and V_{Ri} , and an index denoted by I_0^i . This index indicates the degree of collision between the obstacle and the robot [21, 28]. This degree is expressed as follows:

$$I_0^i = \frac{\sum_{j=1}^{35} \mu_j(d_i, \varphi_i) I_j}{\sum_{k=1}^{35} \mu_k(d_i, \varphi_i)}. \quad (15)$$

The expressions of V_{Li} and V_{Ri} are given as

$$\begin{aligned} V_{Ri} &= \frac{\sum_{j=1}^r \mu_j(d_i, \varphi_i) y_j}{\sum_{k=1}^r \mu_k(d_i, \varphi_i)}, \\ V_{Li} &= \frac{\sum_{j=1}^r \mu_j(d_i, \varphi_i) z_j}{\sum_{k=1}^r \mu_k(d_i, \varphi_i)}, \end{aligned} \quad (16)$$

where I_j is the consequence of the j th rule and μ_j is the activation level of the j th rule.

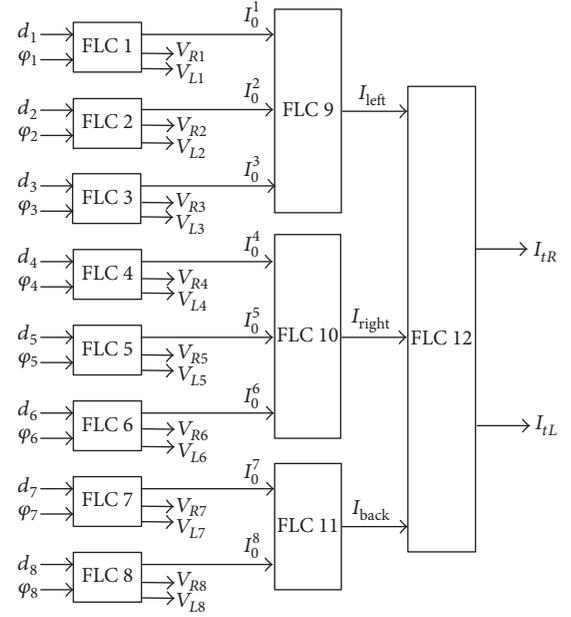


FIGURE 10: Block diagram of the hierarchical fuzzy system.

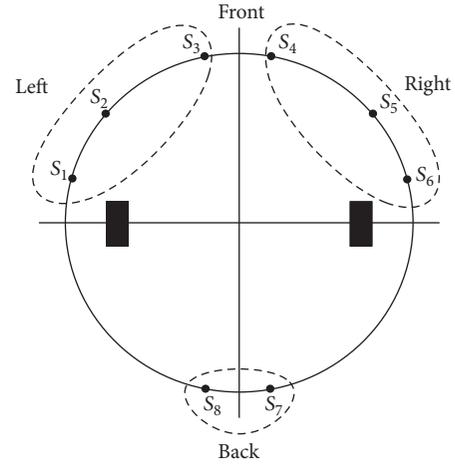


FIGURE 11: Position of the eight sensors of the robot.

Now the controllers are regrouped into three groups as the eight sensors of the Khepera II robot (see Figure 11). This decomposition is made in order to detect the obstacles existing on the right, left, and back sides.

The controllers related to each group of sensors are associated with a fuzzy controller. Therefore, we obtain three controllers which give the indexes I_{left} , I_{right} , and I_{back} . Those new indexes indicate the collision degree between the robot and the obstacle existing on the left, the right, and back sides, respectively.

Finally, I_{left} , I_{right} , and I_{back} are associated with a last fuzzy controller. The outputs of this controller are I_{IR} and I_{IL} . Unlike the previous indexes, those two last indicators give information about the absence of obstacle around the robot. I_{IR} and I_{IL} are expressed as follows:

$$I_{tR} = \frac{\sum_{j=1}^3 \sum_{i=1}^3 \mu_{ij} I_{ij}^R}{\sum_{l=1}^3 \sum_{k=1}^3 \mu_{kl}}, \quad (17)$$

$$I_{tL} = \frac{\sum_{j=1}^3 \sum_{i=1}^3 \mu_{ij} I_{ij}^L}{\sum_{l=1}^3 \sum_{k=1}^3 \mu_{kl}}.$$

The inference tables of all the indexes mentioned previously are given in [3, 28].

At least, the velocities expressions are written as

$$V_R = I_{tR} V_{tR} + \sum_{i=1}^n I_0^i V_{Li}, \quad (18)$$

$$V_L = I_{tL} V_{tL} + \sum_{i=1}^n I_0^i V_{Ri}$$

with

$$V_{Ri} = \frac{\sum_{j=1}^r \mu_j(d_i, \varphi_i) y_j}{\sum_{k=1}^r \mu_k(d_i, \varphi_i)},$$

$$V_{Li} = \frac{\sum_{j=1}^r \mu_j(d_i, \varphi_i) z_j}{\sum_{k=1}^r \mu_k(d_i, \varphi_i)}, \quad (19)$$

$$V_{tR} = \frac{\sum_{j=1}^r \mu_j(d, \varphi) y_j}{\sum_{k=1}^r \mu_k(d, \varphi)},$$

$$V_{tL} = \frac{\sum_{j=1}^r \mu_j(d, \varphi) z_j}{\sum_{k=1}^r \mu_k(d, \varphi)}.$$

4. Cooperative System between Agents: Utility Function

On this section, the cooperative control system is detailed. In fact, the coordination between the agents is necessary to avoid contradictory actions [2].

For this purpose, the utility function is proposed [2, 3]. The idea consists in giving each agent an utility function. According to the value obtained from this function, which varies between 0 and 1, the choice of the agent which controls the robot is made.

We begin by introducing the navigation utility function U_n presented in Figure 12.

This function is drawn according to the system defined as below:

$$U_n = \begin{cases} 1 & \text{if } d \leq 100 \\ \alpha * d + \beta & \text{if } 100 < d < 500 \\ 0.2 & \text{if } d \geq 500, \end{cases} \quad (20)$$

where α is the line slope and β is the ordinate at the origin.

The numerical values of this system are chosen so that the robot adopts the navigation agent when the distance from the target is less than 100 mm (U_n is maximal). When the distance varies between 100 and 50 mm, the utility function decreases linearly with d . When the distance becomes more

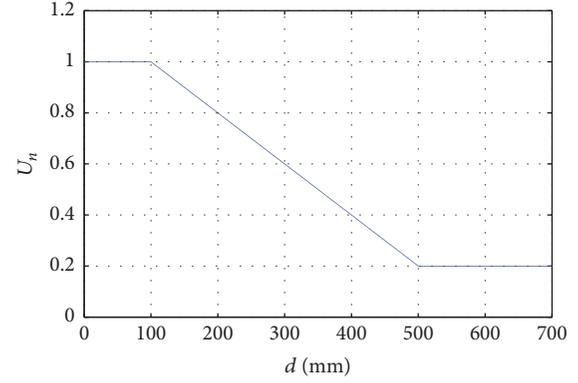


FIGURE 12: The utility function of the navigation agent.

than 500 mm, the value of U_n is minimal, which means that the navigation agent becomes disabled.

The utility function related to the static and dynamic obstacle avoidance agents, respectively, U_{soa} and U_{doa} is combination between the distances which separate the robot from the static and dynamic obstacles, respectively, d_{so} and d_{do} and the orientation of the robot with respect to the static and dynamic obstacles, respectively, φ_{so} and φ_{do} . They are defined as

$$U_{soa} = U_{d_{so}} \times U_{\varphi_{so}}, \quad (21)$$

$$U_{doa} = U_{d_{do}} \times U_{\varphi_{do}}$$

being

$$U_{d_{so}} = \begin{cases} 1 & \text{if } d_{so} \leq R_c \\ \alpha_1 * d_{so} + \beta_1 & \text{if } R_c < d_{so} < 100 \\ 0.2 & \text{if } d_{so} \geq 100, \end{cases} \quad (22)$$

$$U_{d_{do}} = \begin{cases} 1 & \text{if } d_{do} \leq 50 \\ \alpha_2 * d_{do} + \beta_2 & \text{if } 50 < d_{do} < 100 \\ 0.2 & \text{if } d_{do} \geq 100. \end{cases}$$

The same principle of the navigation agent is adopted here. For the dynamic obstacle, the robot enters to the dangerous zone when the distance from the obstacle is less than 50 mm. While for the static obstacle, the dangerous zone is defined as the convergence circle, which means that the robot should move far away when the distance from the obstacle is less than R_c , which is the radius of the convergence circle calculated previously (see Figures 13 and 14).

On the other hand, φ_{so} and φ_{do} are represented by the same function U_{φ_o} :

$$U_{\varphi_o} = \begin{cases} \alpha_3 * \varphi_o + \beta_3 & \text{if } -\frac{\pi}{2} \leq \varphi_o \leq -\frac{\pi}{6} \\ 1 & \text{if } -\frac{\pi}{6} \leq \varphi_o \leq \frac{\pi}{6} \\ -\alpha_4 * \varphi_o + \beta_4 & \text{if } \frac{\pi}{6} \leq \varphi_o \leq \frac{\pi}{2}. \end{cases} \quad (23)$$

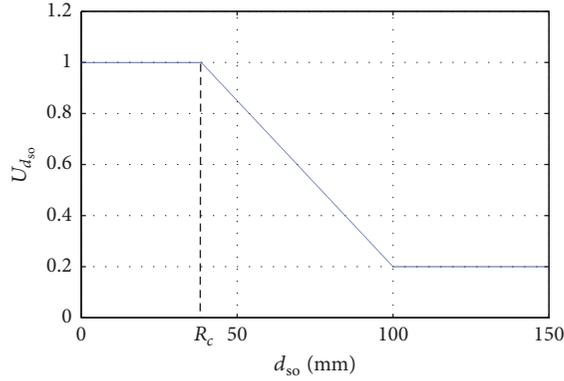


FIGURE 13: The utility function related to the distance of the static obstacle avoidance agent.

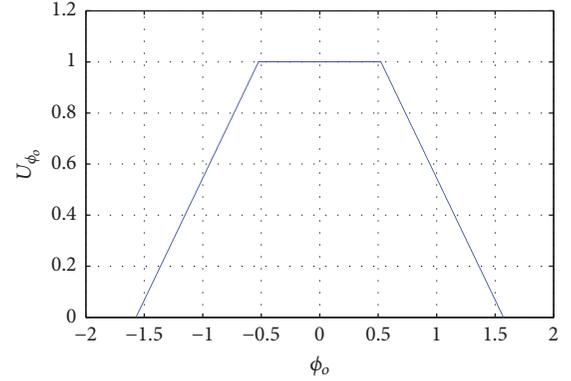


FIGURE 15: The utility function related to the angle of the dynamic obstacle avoidance agent.

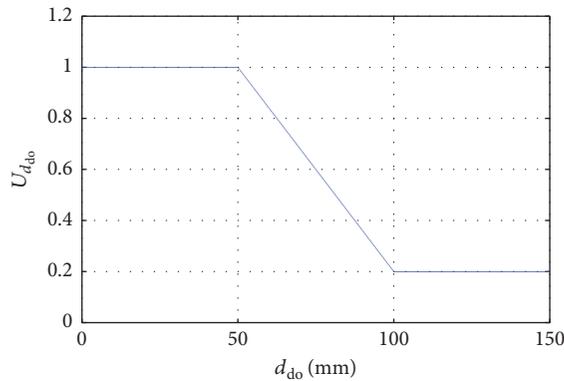


FIGURE 14: The utility function related to the distance of dynamic obstacle avoidance agent.

This utility function is illustrated in Figure 15.

α_i ($i = 1, \dots, 4$) is the line slope, and β_i is the ordinate at the origin.

Once the values of the utility functions are obtained, a negotiation is made to choose which agent the robot selects. Actually several ways of action selection exist [29]. Here, the actual agent sends its value to the agent in conflict. The one having the higher value is the one who controls the robot.

5. Simulation Results

This section is dedicated to test the efficiency of the proposed method. So different configurations were tested. We begin with the simple one. We consider an environment containing a static obstacle, and we must prove that the robot must go around the obstacle in the case of clockwise or counterclockwise motion direction. For that purpose, we have proposed two configurations (see Figures 16 and 17). In both configurations, the robot starts from the initial position $(X, Y) = (0, 0)$ with two different initial directions ($\alpha = \pi/2$ and $\alpha = 0$) and reaches the goal $(X_t, Y_t) = (300, 300)$. The robot is able to avoid the static obstacle applying the sliding mode approach to track the cycle limit trajectory regardless of the robot motion direction (clockwise or counterclockwise).

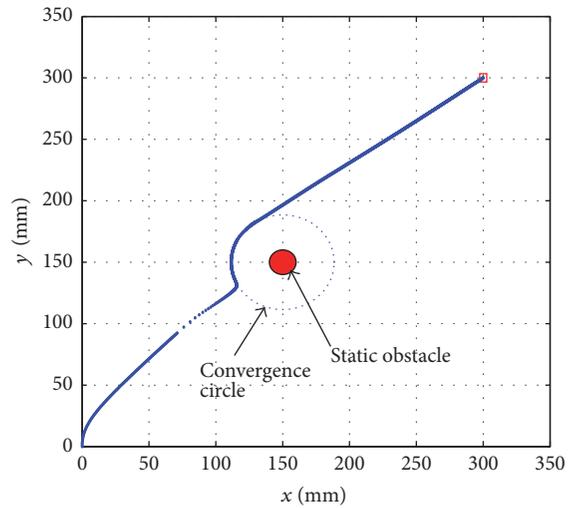


FIGURE 16: Simulation results with one static obstacle ($\alpha = \pi/2$).

Now, we treat the case of environments containing both static and dynamic obstacles.

In Figure 18, the coordination between the agents is verified in an environment containing a static obstacle and a dynamic obstacle which moves horizontally from an initial position $(X_{do}, Y_{do}) = (200, 100)$ to the left.

Figures 19 and 20 deal with the problem of cluttered environment. In Figure 19, the dynamic obstacle moves vertically from the point $(X_{do}, Y_{do}) = (75, 225)$ to the down. The configuration shows that the system can be adaptable to any change of the obstacle's position or number.

Figure 20 presents the case of highly complicated environments with several static and dynamic obstacles with different trajectories. The first dynamic obstacle starts from the position $(X_{do1}, Y_{do1}) = (300, 50)$, and the second obstacle moves from the point $(X_{do1}, Y_{do1}) = (50, 500)$. The robot detects any static obstacle that comes on its way and avoids it by tracking its convergence circle. For the dynamic obstacles, the robot is able to avoid them and reach the target, generating a smooth trajectory.

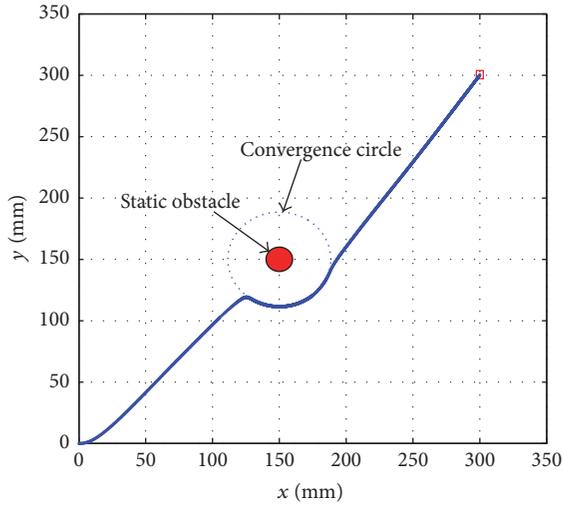


FIGURE 17: Simulation results with one static obstacle ($\alpha = 0$).

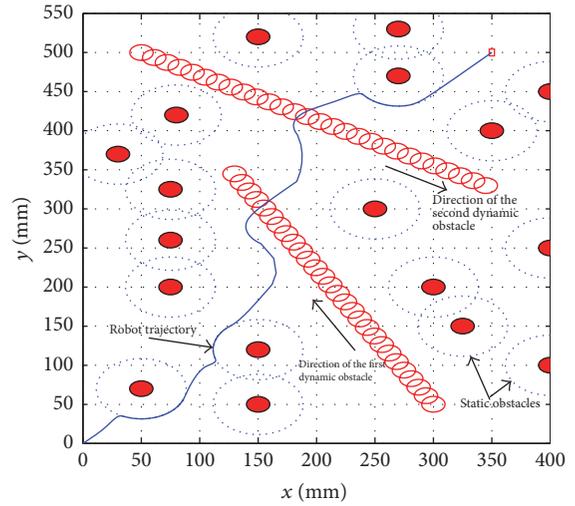


FIGURE 20: Simulation results in a cluttered environment.

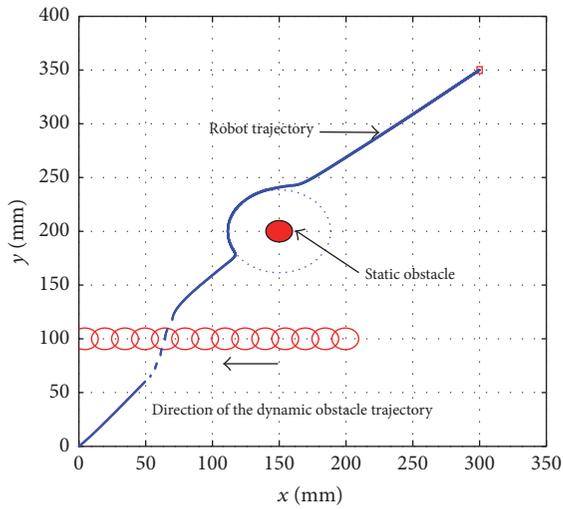


FIGURE 18: Simulation results for one dynamic obstacle and one static obstacle.

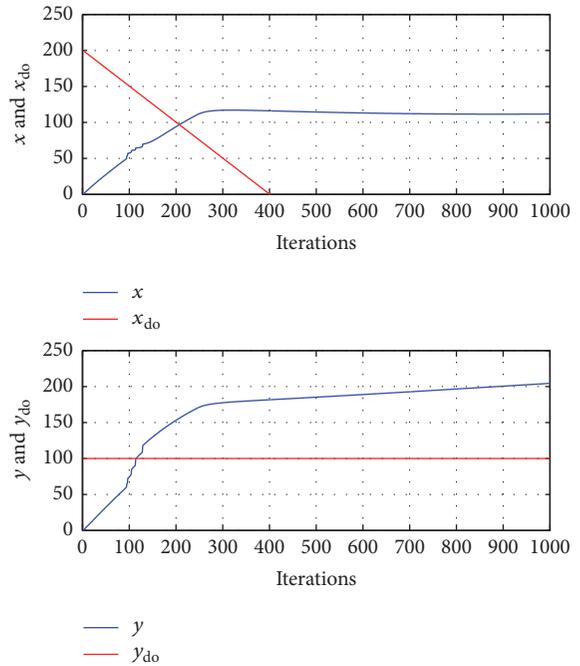


FIGURE 21: Coordinates of the robot and the dynamic obstacle according to time in the case of a horizontal mobile obstacle trajectory.

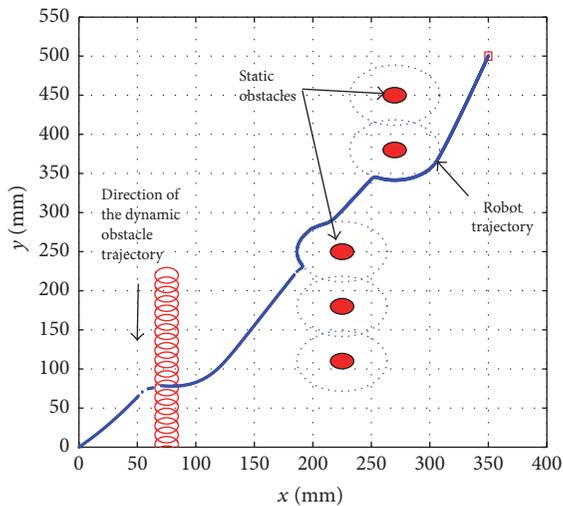


FIGURE 19: Simulation results in a cluttered environment.

According to Figures 21, 22, and 23, there is no collision between the robot and the dynamic obstacle in the three previous simulations. The coordinates of the robot and the dynamic obstacles are not equal at the same time collision t , which means that the robot can move away perfectly from the dynamic obstacles.

The simulations presented in this section illustrate the efficiency of the proposed method and the ability of the robot to transit properly from an agent to another using the suggested utility functions and to reach the final destination safely.

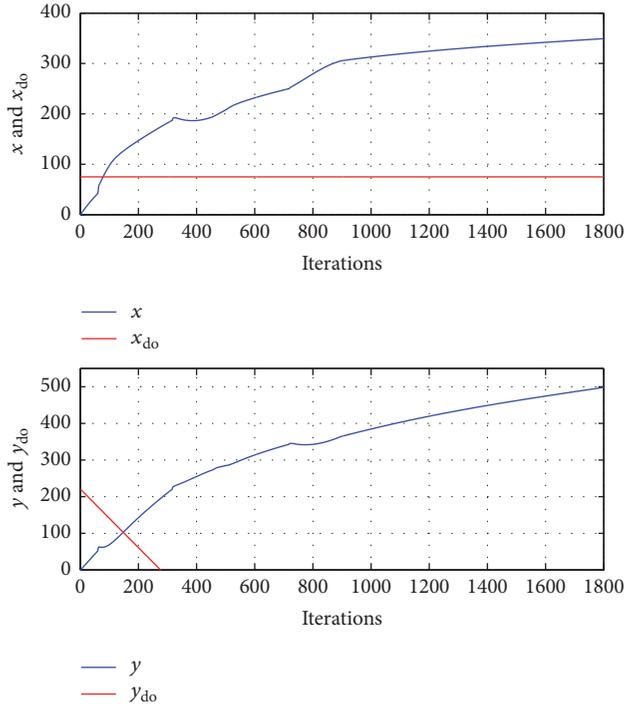


FIGURE 22: Coordinates of the robot and the dynamic obstacle according to time in the case of a vertical mobile obstacle trajectory.

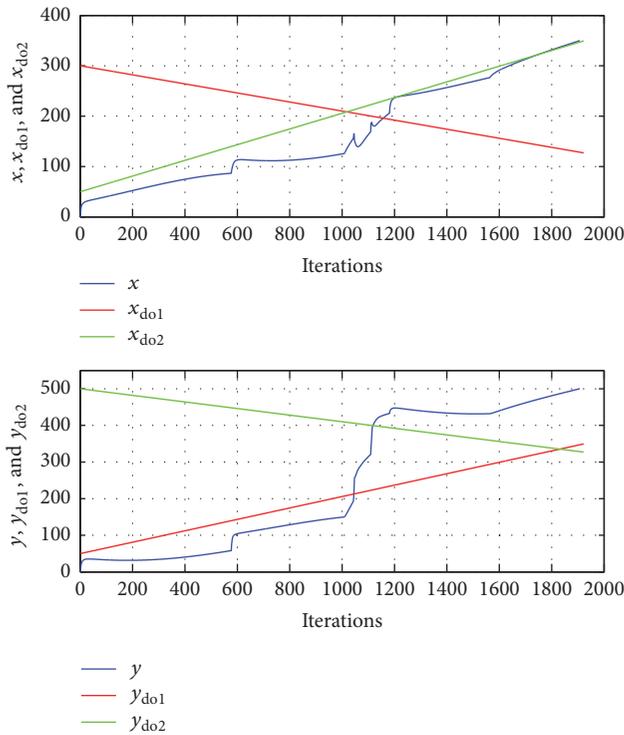


FIGURE 23: Coordinates of the robot and the dynamic obstacles according to time in the case of two different mobile obstacles trajectories.

6. Conclusion

On this work, a multiagent system is implemented for autonomous navigation of mobile robot in an environment containing static and dynamic obstacles. According to the data sent to the perception agent, a coordination system based on utility function is applied. Therefore, the suitable agent is chosen: navigation agent based on simple fuzzy system and dedicated to bring the robot to the target, static obstacle avoidance responsible for leading the robot away from static obstacles tracking an orbital trajectory with the sliding mode control, and dynamic obstacle avoidance which has a role in avoiding dynamic obstacles applying hierarchical fuzzy system. Some tests are made and presented to test the validity of the proposed method.

Conflicts of Interest

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

References

- [1] Z. A. Baig, "Multi-agent systems for protecting critical infrastructures: a survey," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1151–1161, 2012.
- [2] B. Innocenti, B. López, and J. Salvi, "A multi-agent architecture with cooperative fuzzy control for a mobile robot," *Robotics and Autonomous Systems*, vol. 55, no. 12, pp. 881–891, 2007.
- [3] M. Boujelben, C. Rekik, and N. Derbel, "A multi-agent architecture with hierarchical fuzzy controller for a mobile robot," *International Journal of Robotics and Automation*, vol. 30, no. 3, pp. 289–298, 2015.
- [4] C. G. Cena, P. F. Cardenas, R. S. Pazmino, L. Puglisi, and R. A. Santonja, "A cooperative multi-agent robotics system: Design and modelling," *Expert Systems with Applications*, vol. 40, no. 12, pp. 4737–4748, 2013.
- [5] A. Hentout, B. Bouzouia, and Z. Toukal, "Multi-agent architecture model for driving mobile manipulator robots," *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 257–268, 2008.
- [6] M. Kolp, P. Giorgini, and J. Mylopoulos, "Multi-agent architectures as organizational structures," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 1, pp. 3–25, 2006.
- [7] Y. Ono, H. Uchiyama, and W. Potter, "A mobile robot for corridor navigation: A multi-agent approach," in *Proceedings of the 42nd Annual Southeast Regional Conference, ACM-SE 42*, pp. 379–384, USA, April 2004.
- [8] D. Busquets, C. Sierra, and R. López de Mántaras, "A multiagent approach to qualitative landmark-based navigation," *Autonomous Robots*, vol. 15, no. 2, pp. 129–154, 2003.
- [9] A. Benzerrouk, L. Adouane, and P. Martinet, "Stable navigation in formation for a multi-robot system based on a constrained virtual structure," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1806–1815, 2014.
- [10] N. Hacene and B. Mendil, "Toward safety navigation in cluttered dynamic environment: A robot neural-based hybrid autonomous navigation and obstacle avoidance with moving target tracking," in *Proceedings of the 3rd International Conference on Control, Engineering and Information Technology, CEIT 2015*, dza, May 2015.

- [11] R. Rojas and A. G. Förster, "Holonomic control of a robot with an omnidirectional drive," *KI-Künstliche Intelligenz*, vol. 20, no. 2, pp. 12–17, 2006.
- [12] J. H. Lee, C. Lin, H. Lim, and J. M. Lee, "Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space," *International Journal of Control, Automation, and Systems*, vol. 7, no. 3, pp. 429–435, 2009.
- [13] I. Engedy and G. Horváth, "Artificial neural network based mobile robot navigation," in *Proceedings of the WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing*, pp. 241–246, Hungary, August 2009.
- [14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [15] M. Boujelben, C. Rekik, and N. Derbel, "Hierarchical fuzzy controller to avoid mobile obstacle for a mobile robot," in *Proceedings of the 2013 10th International Multi-Conference on Systems, Signals and Devices, SSD 2013*, tun, March 2013.
- [16] C.-F. Juang and Y.-C. Chang, "Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 379–392, 2011.
- [17] D. Ayedi, M. Boujelben, and C. Rekik, "Interval type-2 fuzzy approach for autonomous mobile robot control in presence of uncertainties," in *Proceedings of the Systems, Signals & Devices (SSD), 2017 14th International Multi-Conference*, pp. 280–287, 2017.
- [18] D. Nakhaeinia, S. H. Tang, S. B. Mohd Noor, and O. Motlagh, "A review of control architectures for autonomous navigation of mobile robots," *International Journal of Physical Sciences*, vol. 6, no. 2, pp. 169–174, 2011.
- [19] C.-W. Chen, "Interconnected TS fuzzy technique for nonlinear time-delay structural systems," *Nonlinear Dynamics*, vol. 76, no. 1, pp. 13–22, 2014.
- [20] C. Rekik, M. Djemel, and N. Derbel, "A discussion on the optimal control of a robot manipulator by a hybrid genetic-fuzzy controller," *International Journal of Robotics and Automation*, vol. 23, no. 3, pp. 150–159, 2008.
- [21] M. Boujelben, C. Rekik, and N. Derbel, "Hierarchical fuzzy controller for a nonholonomic mobile robot," in *Proceedings of the 2012 20th Mediterranean Conference on Control and Automation, MED 2012*, pp. 341–347, esp, July 2012.
- [22] M. Boujelben, C. Rekik, and N. Derbel, "A hybrid fuzzy-sliding mode controller for a mobile robot," *International Journal of Modelling, Identification and Control*, vol. 25, no. 3, pp. 155–164, 2016.
- [23] G. Antonelli and S. Chiaverini, "Experiments of fuzzy lane following for mobile robots," in *Proceedings of the 2004 American Control Conference (AAC)*, pp. 1079–1084, usa, July 2004.
- [24] L. Adouane, "Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 2009.
- [25] D.-H. Kim and J.-H. Kim, "A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer," *Robotics and Autonomous Systems*, vol. 42, no. 1, pp. 17–30, 2003.
- [26] R. Solea, A. Filipescu, and U. Nunes, "Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties," in *Proceedings of the 7th Asian Control Conference (ASCC '09)*, pp. 1701–1706, Hong Kong, August 2009.
- [27] I. González, S. Salazar, and R. Lozano, "Chattering-free sliding mode altitude control for a quad-rotor aircraft: Real-time application," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 137–155, 2014.
- [28] M. Boujelben, C. Rekik, and N. Derbel, "Hierarchical neural network control for a mobile robot," in *Proceedings of the 2nd International Conference on Automation, Control, Engineering and Computer Science (ASECS)*, 2015.
- [29] P. Pirjanian, "Behavior coordination mechanisms-state-of-the-art," Technical Report IRIS-99-375, University of Southern California, Institute for Robotics and Intelligent Systems, 1999.

Research Article

Proportional Load Sharing and Stability of DC Microgrid with Distributed Architecture Using SM Controller

Muhammad Rashad ¹, Uzair Raouf,² Muhammad Ashraf,¹ and Bilal Ashfaq Ahmed²

¹Department of Electrical Engineering, Capital University of Science and Technology (CUST), Kahuta Road, Express Highway, Islamabad 44000, Pakistan

²Department of Electrical Engineering, The University of Lahore (UOL), Defense Road, Lahore 54000, Pakistan

Correspondence should be addressed to Muhammad Rashad; muhammad.rashad@ee.uol.edu.pk

Received 7 September 2017; Revised 15 November 2017; Accepted 29 November 2017; Published 8 January 2018

Academic Editor: Rafael Morales

Copyright © 2018 Muhammad Rashad 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.

DC microgrids look attractive in distribution systems due to their high reliability, high efficiency, and easy integration with renewable energy sources. The key objectives of the DC microgrid include proportional load sharing and precise voltage regulation. Droop controllers are based on decentralized control architectures which are not effective in achieving these objectives simultaneously due to the voltage error and load power variation. A centralized controller can achieve these objectives using a high speed communication link. However, it loses reliability due to the single point failure. Additionally, these controllers are realized through proportional integral (PI) controllers which cannot ensure load sharing and stability in all operating conditions. To address limitations, a distributed architecture using sliding mode (SM) controller utilizing low bandwidth communication is proposed for DC microgrids in this paper. The main advantages are high reliability, load power sharing, and precise voltage regulation. Further, the SM controller shows high robustness, fast dynamic response, and good stability for large load variations. To analyze the stability and dynamic performance, a system model is developed and its transversality, reachability, and equivalent control conditions are verified. Furthermore, the dynamic behavior of the modeled system is investigated for underdamped and critically damped responses. Detailed simulations are carried out to show the effectiveness of the proposed controller.

1. Introduction

Microgrids are a modern form of distribution systems which can function autonomously or in combination with a main supply grid. Microgrids can operate in a low or medium voltage range and generate their own power, that is, renewable sources along with energy storage, nonrenewable sources, and power electronic (PE) controlled loads [1]. The unique property of microgrids is that they can work in islanded mode under faulty conditions, which increases the reliability of the power supply [2–4]. In AC microgrids, the distribution system is AC. AC microgrids operation and control have been exhaustively discussed in the literature [5, 6]. DC microgrids are paid attention from researchers due to the following advantages:

(1) Most of the renewable sources are inherently DC or DC friendly. So, the integration of these sources with a DC

microgrid is easy as there is no requirement of frequency synchronization circuits.

(2) In DC microgrids, reactive power compensators are not required. Hence, their control will be relaxed as power flow controllers are not required.

(3) As DC electronic loads dominate today, the unnecessary AC-to-DC converters are not required in DC microgrids. This will directly affect system cost and efficiency.

(4) Skin effect problems are absent in DC microgrids.

A general DC microgrid arrangement [7, 8] connecting different sources and loads is shown in Figure 1. To integrate dissimilar elements, PE converters (DC-to-DC, AC-to-DC, etc.) are included among sources, energy storage, and microgrids.

Distributed generation can be connected to the DC microgrid through power PE converters in a parallel configuration. It is required to find efficient control to coordinate

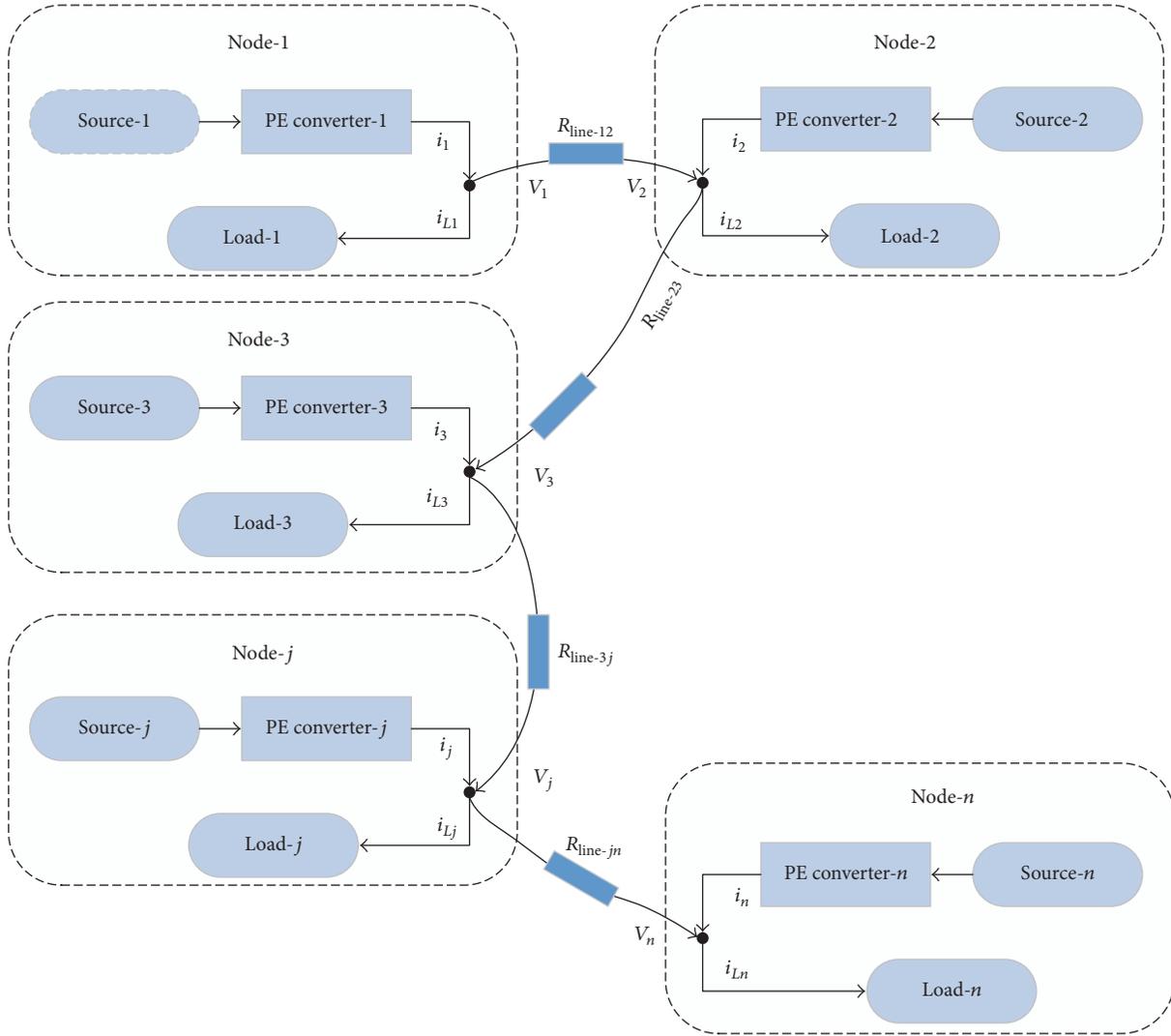


FIGURE 1: General DC microgrid arrangement [8].

among various sources, loads, and energy storage. The key concerns of the parallel connected converter are as follows: the first concern is the stability of the DC microgrid as electronic loads are very sensitive to voltage deviation. Another concern is load sharing among various sources [7–9]. However, it is the nature of sources that only one unit can establish the voltage level in a paralleling system. The reason is, the output resistances of the power sources are extremely low. Thus, even a small difference in the output voltage between the paralleled sources will cause the one that is a few mV higher to hog all the current. The lower the output voltage of the module is, the more severe this problem is [10].

To address the aforementioned challenges, different control schemes for DC microgrids are reported in [7, 10–16]. Commonly, these control schemes can be categorized as centralized and decentralized control [10–21]. In centralized control, the controller collects system data using a high bandwidth communication link, schedules the tasks based on the collected information, and directs control decisions [12, 16–18]. However, if the communication link fails, this

will degrade the system performance and reliability. Communication failure problems can be avoided in decentralized control (droop control) [13, 14, 19, 22]. In this type of control, PE converters operate on physical measured quantities. But the improvement is achieved at the cost of partial stability and losing optimum operation due to the lack of operational information and status of the other converters [23, 24]. In [25–29], droop control for AC microgrids is reported. Their extensive use in conventional AC systems made them appealing to be used in DC microgrids [30–32].

Linear proportional integral (PI) controllers are used to realize the abovementioned control schemes for proportional load sharing and stability of DC microgrids [30–32]. Despite the easy implementation of these controllers, they suffer poor sharing. In many cases, the stability of PI controllers cannot be ensured [33, 34]. Since PI controllers are linear type controllers, the control parameters of these controllers cannot be optimized as they are tuned for specific load conditions. Further, they also exhibit a slower dynamic response [35]. Hence, using these controllers for load sharing is not desirable.

Sliding mode control (SMC) for proportional load sharing and stability of DC microgrids is proposed in this paper. SMC for microgrid type systems is reported in [9, 36–40]. SM is a controller which is used for variable structure systems [36]. It is a nonlinear controller which shows robust behavior to the matched uncertainties. The SM controller can be easily implemented using integrated circuits. Hence, the SM controller is proposed for proportional load sharing and stability of DC microgrids.

Section 2 deals with the load sharing control schemes used in DC microgrids. Further, a system model and its analysis through the SM controller are presented. Furthermore, this section deals with system stability and dynamic behavior design. Detailed simulation studies are performed in Section 3. Section 4 concludes this paper.

2. Load Sharing Control in DC Microgrids

The objective of the control in DC microgrids is to achieve load sharing and maintain precise voltage regulation among the sources. The sharing control can be categorized into the following types.

2.1. Centralized Control. In this type of control, the controller collects system data using a communication link and directs control decisions. This type of control scheme is reported in [12, 16–18] and is shown in Figure 2. The PE converter of each source contains primary control and inner voltage and current control. Centralized control gives directions and control decisions to the other primary controllers. The voltage of the DC microgrid is communicated to the central controller, where it is compared with the reference voltage. The error produced is transferred to the PI controller whose output is communicated to the primary controller of each source as shown in Figure 2. However, if a single point failure occurs, this will degrade the system performance and reliability. Hence, using centralized control for load sharing and voltage regulation in DC microgrids is not attractive.

2.2. Decentralized Control. Single point failure problems can be avoided in decentralized control (droop control) [13, 14, 19, 22]. In this type of control, PE converters operate on local physical measured quantities. This does not require a separate central controller. Decentralized control sets the values for droop control which derives the inner voltage and current control of the PE converters as shown in Figure 3.

Droop control cannot achieve load sharing and voltage regulation simultaneously. Limitations of the droop control are examined below.

2.2.1. Current Sharing Inaccuracy. Current sharing will be affected due to the voltage error in parallel connected DC-to-DC converters. This problem becomes challenging due to the extra voltage drop across the line connecting parallel sources or when the characteristics of different sources are not the same. Hence, current sharing among various sources is deteriorated. To analyze this problem, a DC microgrid with two sources is shown in Figure 4, where each source is modeled by its Thevenin equivalent circuit.

The droop control used in DC systems can be expressed as

$$v_{sj} = v_s - i_{dcj}R_{dj} \quad \text{where } j = 1, 2, \quad (1)$$

where v_{sj} , v_s , i_{dcj} , and R_{dj} are node voltage, source voltage, source current, and virtual resistance of each source, respectively. The virtual resistance defined in (1) is equal to the output resistance and the output voltage of each source is equal to v_s , as shown in Figure 4. Consider the load is drawing rated current and the system has reached a steady state. The following can be derived from Figure 4:

$$v_{load} = v_s - i_{dc1}R_{d1} - idc_1R_{line1} \quad (2a)$$

$$v_{load} = v_s - i_{dc2}R_{d2} - idc_2R_{line2}. \quad (2b)$$

After simplification, these equations can be written as

$$\frac{i_{dc1}}{i_{dc2}} = \frac{R_{d2}}{R_{d1}} + \frac{R_{line2} - (R_{d2}/R_{d1})R_{line1}}{R_{d1} + R_{line1}}. \quad (3)$$

The above equation shows that, in droop controlled DC grids, the current of both sources is inversely proportional to their virtual resistance. Usually, it is assumed that DC microgrids are small-scale grids and the connecting lines will contain resistance of a small value. Hence, virtual resistance R_{dj} can be selected large. Since $R_{dj} \gg R_{line}$, the above expression can be written as

$$\frac{i_{dc1}}{i_{dc2}} = \frac{R_{d2} + R_{line2}}{R_{d1} + R_{line1}} \approx \frac{R_{d2}}{R_{d1}}. \quad (4)$$

But the abovementioned assumption is suitable for large R_{dj} . For small R_{dj} , precise current sharing cannot be ensured. Meanwhile, voltage regulation cannot be ensured with large virtual resistance. This is graphically shown in Figure 5.

2.2.2. Output Voltage Deviation. Node voltage deviation can be written as

$$\Delta v_j = v_s - v_{sj} = i_{dcj}R_{dj} \quad \text{where } j = 1, 2. \quad (5)$$

Figure 6 shows the voltage deviation with different virtual resistances. Voltage deviation is of zero value when the sources operate in an open circuit mode (source currents are zero) as shown in Figure 6. When the current by the sources is not zero, voltage deviation appears, and its value varies with the variation in load. To limit the output voltage deviation within acceptable levels, the droop coefficient R_{dj} should be limited as

$$R_{dj} \leq \frac{\Delta v_{max}}{i_{nj}}, \quad (6)$$

where i_{nj} is full-load current of source- j .

2.3. Distributive Control. Disadvantages associated with the decentralized (droop control) and centralized control can be adjusted using distributed control which is an alternative solution to achieve efficient load sharing. As a substitute of

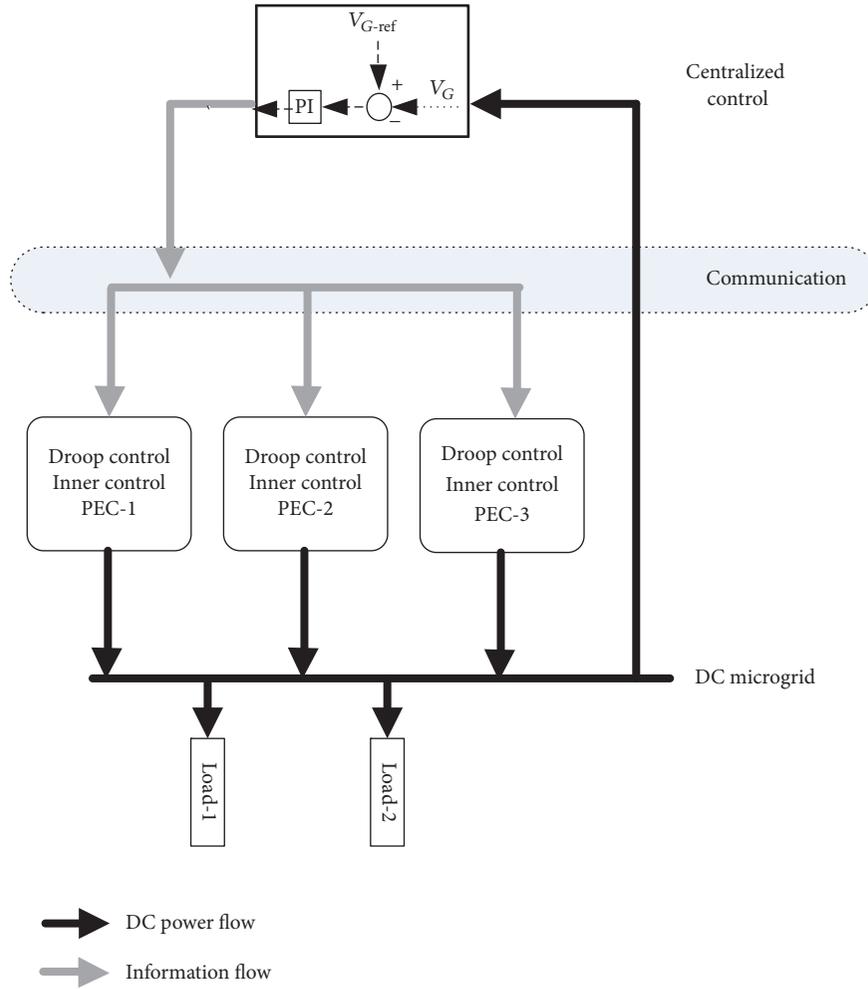


FIGURE 2: Centralized control in a DC microgrid.

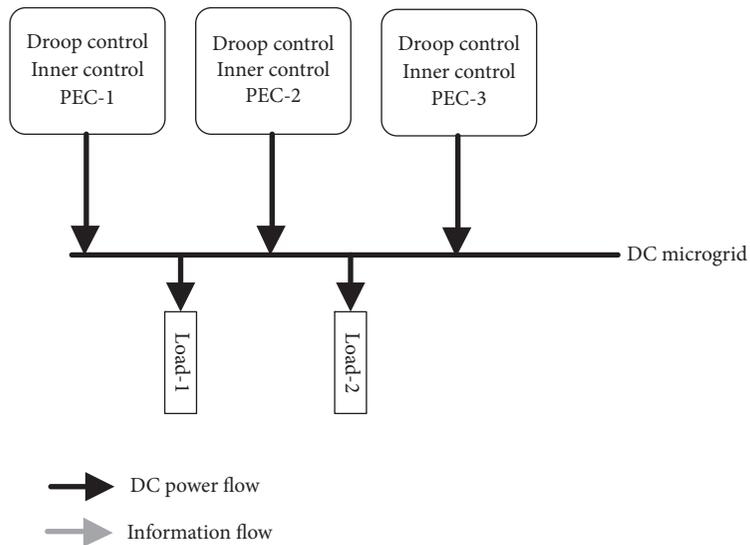


FIGURE 3: Decentralized control.

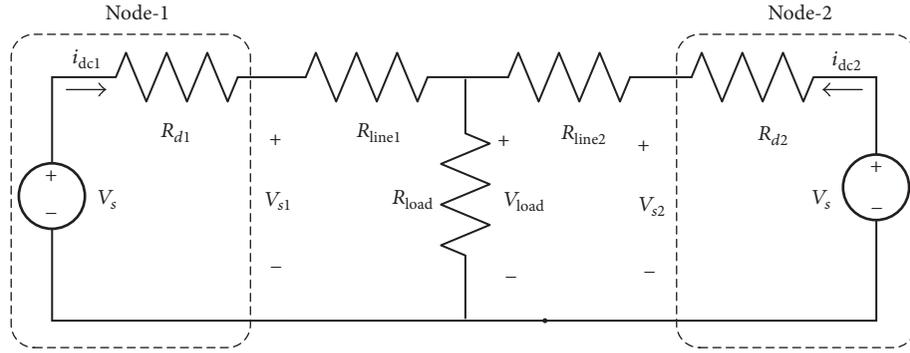


FIGURE 4: Thevenin equivalent circuit of a DC microgrid with two sources sharing the same load.

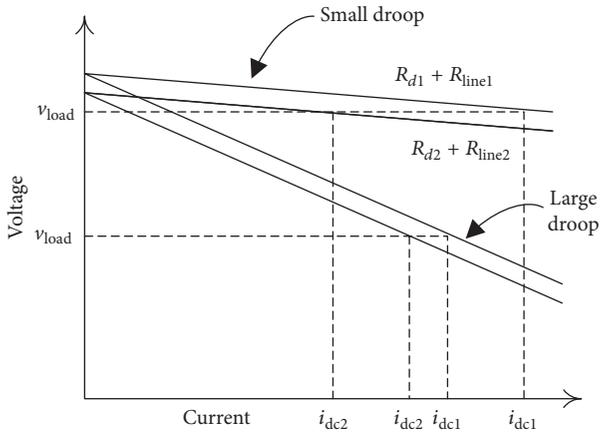


FIGURE 5: Current sharing inaccuracy using droop control [7].

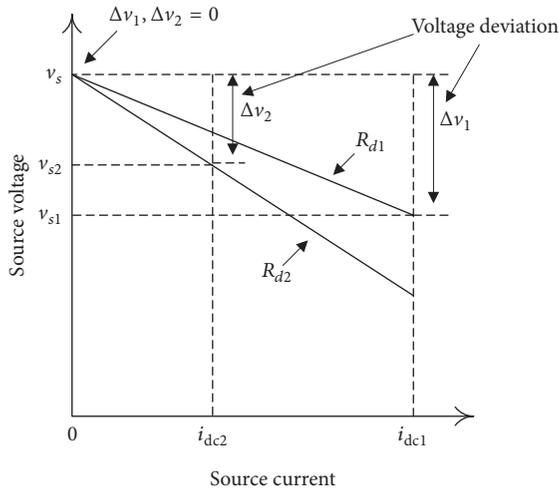


FIGURE 6: Voltage deviation with a variation in droop resistances [8].

the single central controller, distributive control is distributed among every PE converter. In [24], DC bus signaling (DBS) is proposed for distributed control in which DC voltage is used to communicate the decision about the converters' operation. In DBS, every entity in the system senses the DC voltage level for operation and decisions, which limits the

sources, loads, and storage because DC bus voltage cannot be allocated unlimitedly. Furthermore, some extreme situations (e.g., overvoltage/current and fully charged/undercharged battery) are not addressed in DBS. In [41], a current sharing line is proposed and distributed, in which the average current is communicated among converters for operation. In DC microgrids, sources are displaced from each other over a region. Thus, the current sharing bus needs to be distributed over the region with the power lines. This may inject substantial noise which can degrade the system performance. In [42], a distributive secondary control using power line signaling (PLS) is presented. The major problem in PLS is that it has slow communication. Further, electromagnetic compatibility issues need to be addressed when using it with electronic devices.

In this paper, a distributive control for DC microgrids using low bandwidth communication is proposed and is shown in Figure 7.

To determine the value of current to be shared by each source, the controller of each source communicates among the other sources and informs about the per-unit (p.u.) current supplied by it. This information is used by each controller to determine the total current supplied by all sources, which is given as

$$i_T^{\text{pu}} = \sum_{m=1}^n i_j^{\text{pu}}, \quad (7)$$

where i_j^{pu} is the p.u. current of source- j and n is the number of sources. The reference current of each source i_j^{ref} is calculated as

$$i_j^{\text{ref}} = k_j i_j^{\text{rated}} i_T^{\text{pu}}, \quad (8)$$

where k_j and i_j^{rated} are the load sharing factor and rated current of source- j , respectively.

In the architecture shown in Figure 7, the p.u. current of each source is shared. Hence, the data transmitted over the communication link by each source is 2 bytes and the total data transmitted is $2n$ bytes. Data read by each source is $2(n-1)$ bytes. Hence, the technique used for communication has to manage small data, and thus a low bandwidth communication is feasible. So, computer area network (CAN) based communication is used for DC microgrids in this paper.

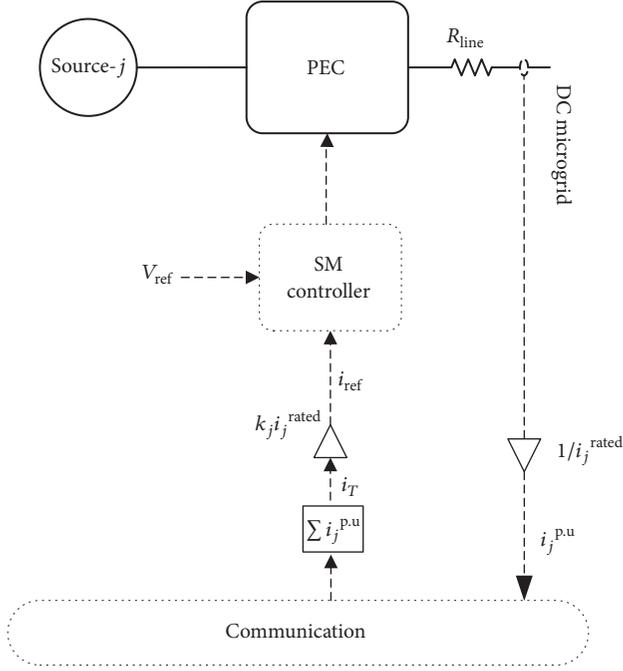


FIGURE 7: Distributive control architecture.

2.3.1. Sliding Mode Control. Each source in a DC microgrid consists of a PE converter. Linear controllers (PI, PID, and lead-lag) are being used to control the PE converters for the load sharing problem in DC microgrids [30–32]. These controllers require a linearized model of the system, which makes it difficult for them to show good power sharing performance and stability in all operating conditions [33–35]. So, an SM controller is alternatively proposed in [36–40, 43] which ensures stability in all operating conditions. Hence, in this paper, an SM controller technique is proposed for proportional load sharing and stability of DC microgrids.

(1) Modeling. A generalized DC microgrid architecture is shown in Figure 1. An equivalent model of one source with DC microgrid through DC-to-DC PE converter is shown in Figure 8. The source is modeled as voltage source v_s with current i_s , whereas the DC bus of the microgrid is modeled through capacitor C and its associated connecting line current i_{dc} . The source and DC microgrid are interfaced through the DC-to-DC PE converter. The differential equations describing the system dynamics of this model are expressed as follows:

$$\frac{di}{dt} = \frac{-v_{dc} + uv_s}{L}, \quad (9)$$

where $v_{dc} = v_{line} + v_G$, and

$$\frac{dv_{dc}}{dt} = \frac{i - i_{dc}}{C}, \quad (10)$$

where i , v_{dc} , v_{line} , v_G , i_{dc} , L , and C are inductor current, bus voltage (capacitor voltage), connecting line voltage, grid voltage, line current, inductance, and capacitance, respectively,

whereas u defines the switching state of the MOSFET switch which can be expressed as

$$u = \begin{cases} 1 & \text{switch is "ON"} \\ 0 & \text{switch is "OFF"}. \end{cases} \quad (11)$$

(2) Sliding Mode Controller Analysis. In sliding mode, most of the controllers include error of one or multiple states of the system in the sliding surface (e.g., inductor current or capacitor voltage) [44, 45]. Furthermore, some controllers include error and both of the time derivative and the integral of the error in the sliding surface to stabilize the system [38]. In this case, the sliding surface can be represented as a second-order differential equation for which extensive mathematical analysis is required to guarantee system stability. Another surface is defined in [46] for the improvement in the steady-state error and settling time, which includes voltage error and square of the capacitor current of the system.

This paper proposes an SM controller which is designed to achieve both proportional power sharing and dynamic stability of DC microgrids. The sliding surface is selected to ensure load sharing and precise voltage regulation. Thus, it is formed using the bus voltage error, current error, and integral of the bus voltage error. This way, the SM controller can detect and minimize the voltage and current errors. Further, the integral action is included to reduce the steady-state voltage error. The proposed sliding surface Ψ is given in

$$\Psi = \alpha e_v + \beta e_i + \int e_v dt, \quad (12)$$

where

$$\begin{aligned} e_v &= v_{dc} - v_{dc}^{\text{ref}}, \\ e_i &= i - i^{\text{ref}}, \end{aligned} \quad (13)$$

whereas e_v , e_i , v_{dc}^{ref} , and i^{ref} are voltage error, current error, reference bus voltage, and reference inductor current, respectively, while α and β are parameters of the sliding surface. Figure 9 shows the block diagram of the SM control system.

The derivative of the sliding surface is used to ensure the existence of SM which is given as

$$\frac{d\Psi}{dt} = \alpha \frac{dv_{dc}}{dt} + \beta \frac{di}{dt} + e_v. \quad (14)$$

Substituting (9) and (10) into (14) leads to

$$\frac{d\Psi}{dt} = \frac{\alpha}{C} (i - i_{dc}) - \frac{\beta}{L} (v_{dc} - uv_s) + e_v. \quad (15)$$

For the existence of SM, the conditions described in (16) need to be fulfilled [47], which ensures that, in steady-state condition, $v_{dc} = v_{dc}^{\text{ref}}$ and $i = i^{\text{ref}}$. Hence, voltage and current achieve the desired reference.

$$\Psi = 0,$$

$$\frac{d\Psi}{dt} = 0. \quad (16)$$

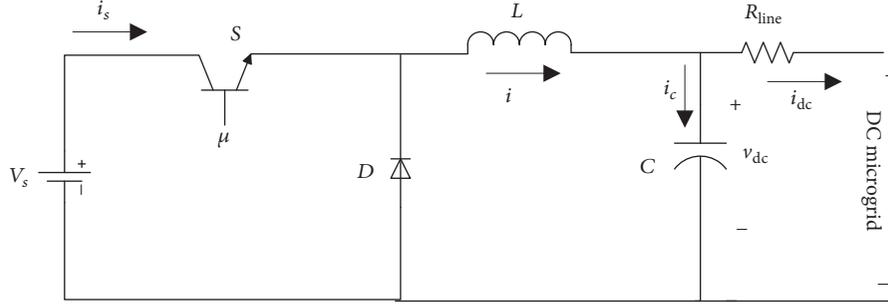


FIGURE 8: Equivalent model of one source with a DC microgrid.

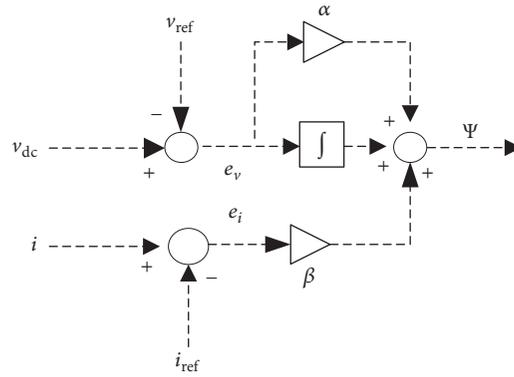


FIGURE 9: Block diagram of the SM controller.

To ensure the existence of SM (i.e., guaranteeing (16)), the transversality, reachability, and equivalent control conditions must be guaranteed.

(i) *Transversality Condition.* The transversality condition describes the system controllability. Thus, this condition must be satisfied to allow the system dynamics to be affected by the SM controller [47]. It ensures that the control variable is present in the derivative of sliding surface. It can be expressed as

$$\frac{d}{du} \left(\frac{d\Psi}{dt} \right) \neq 0. \quad (17)$$

Substituting (15) into (17) results in

$$\frac{d}{du} \left(\frac{d\Psi}{dt} \right) = \frac{\beta}{L} (v_s) \neq 0. \quad (18)$$

Equation (18) depends on the value of β . Section 3.3 determines that the value of β must be positive to guarantee a stable system behavior. Moreover, v_s and L are positive quantities. Hence, (17) will be fulfilled and the transversality condition of the system is satisfied.

(ii) *Reachability Condition.* The reachability condition describes the system's ability to reach the sliding surface [47, 48]. Thus, the reachability condition ensures that the system will

always be directed towards the sliding manifold. Mathematically, it is described in

$$\begin{aligned} \lim_{\Psi \rightarrow 0^-} \frac{d\Psi}{dt} \Big|_{u=1} &> 0, \\ \lim_{\Psi \rightarrow 0^+} \frac{d\Psi}{dt} \Big|_{u=0} &< 0. \end{aligned} \quad (19)$$

Substituting (15) into (19), it can be expressed as

$$\begin{aligned} \frac{\alpha}{C} (i - i_{dc}) + \frac{\beta}{L} (v_s - v_{dc}) + e_v &> 0 \\ -\frac{\alpha}{C} (i_{dc} - i) - \frac{\beta}{L} v_{dc} + e_v &< 0. \end{aligned} \quad (20)$$

The conditions in (20) must be fulfilled to guarantee that the SM will exist, and the system will be derived to the desired operating condition.

(iii) *Equivalent Control Condition.* The equivalent control condition defines the local stability of the system and enables that system to remain trapped inside the sliding surface [47, 48]. Mathematically, the equivalent control condition can be expressed as in

$$\begin{aligned} \frac{d\Psi}{dt} \Big|_{u=u_{eq}} &= 0 \longrightarrow \\ 0 &< u_{eq} < 1. \end{aligned} \quad (21)$$

Substituting $u = u_{\text{eq}}$ in (15) and equating to zero result in the following expression:

$$u_{\text{eq}} = \frac{v_{\text{dc}}}{v_s} - \frac{\alpha}{\beta} \left(\frac{L}{C} \right) \left(\frac{i - i_{\text{dc}}}{v_s} \right) - \frac{L}{\beta v_s} e_v. \quad (22)$$

Fulfilling the reachability condition also verifies the equivalent control condition.

(3) *Sliding Mode Dynamics.* The inequalities in (20) generally describe the existence of SM. These inequalities do not give information about the selection of the sliding parameters α and β . This section deals with the selection of sliding parameters based on the desired dynamic behavior. Thus, the stability of the system is fulfilled. For this purpose, closed loop dynamics in time domain of the system are achieved by putting sliding surface $\Psi = 0$ in (12). The dynamics are given in

$$e_i = \frac{\alpha}{\beta} e_v + \frac{1}{\beta} \int e_v dt. \quad (23)$$

The closed loop dynamics in Laplace domain are given in

$$\frac{e_i(s)}{e_v(s)} = \frac{1 + \alpha s}{\beta s}. \quad (24)$$

Formerly, the voltage dynamics of the DC bus defined in (10) are imposed by (24). Further, to find complete closed loop dynamics of the system, (10) can be expressed in Laplace domain as

$$C s v_{\text{dc}}(s) = i(s) + i_{\text{dc}}(s). \quad (25)$$

Finally, combining (24) and (25), the complete closed loop dynamics of the DC bus voltage can be expressed as

$$\begin{aligned} v_{\text{dc}}(s) = & -\frac{\beta s}{C\beta s^2 + \alpha s + 1} i_{\text{dc}}(s) \\ & + \frac{1 + \alpha s}{C\beta s^2 + \alpha s + 1} v_{\text{dc}}^{\text{ref}}(s) \\ & + \frac{\beta s}{C\beta s^2 + \alpha s + 1} i^{\text{ref}}(s). \end{aligned} \quad (26)$$

It is shown in (26) that closed loop dynamics depend on the perturbation introduced by the DC bus current and the reference. Since the reference is a constant value and the DC bus current depends on the source and load power requirements, (26) can be expressed as (27) which is used to design sliding parameters α and β of the SM controller.

$$\frac{v_{\text{dc}}(s)}{i_{\text{dc}}(s)} = -\frac{s}{C s^2 + (\alpha/\beta) s + 1/\beta}. \quad (27)$$

Finally, it is established in (27) that both sliding parameters α and β must have a positive value to ensure stable SM dynamics; otherwise, the system will show an unstable behavior.

(4) *Design of Sliding Mode Dynamic Behavior.* The dynamics shown in (27) demonstrate that the SM controller will

compensate any perturbation produced in the bus current i_{dc} (i.e., $\lim_{\Psi \rightarrow \infty} (v_{\text{dc}}(s)/i_{\text{dc}}(s)) = 0$). However, large undershoots and overshoots in bus voltage can turn off or destroy the load. Therefore, the dynamics of the bus voltage should be controlled. The characteristic polynomial in the denominator of (27) shows that it is a second-order system of the form given in

$$s^2 + 2\zeta\omega_0 s + \omega_0^2 = 0, \quad (28)$$

where ζ and ω_0 are the damping ratio and undamped natural frequency, respectively. Comparing (28) with the denominator of (27), ζ and ω_0 can be calculated as

$$\begin{aligned} \omega_0 &= \sqrt{\frac{1}{\beta C}} \\ \zeta &= \frac{\alpha}{2} \sqrt{\frac{1}{\beta C}}. \end{aligned} \quad (29)$$

ζ and ω_0 can control the response of the system. Two types of responses are considered in this paper: underdamped and critically damped response. The characteristics of these are discussed below.

(i) *Underdamped Response.* In this type, the controller enables the system response to reach the desired voltage faster. But this faster response is achieved at the expense of oscillation around the desired voltage. The loads that are sensitive to the voltage drops but not sensitive to the oscillations are suitable to be controlled using underdamped response (e.g., microprocessors) [49].

(ii) *Critically Damped Response.* In this type, the controller enables the system response to avoid oscillations. But the response will experience a longer delay reaching the desired voltage level. The loads that are sensitive to the oscillation but show tolerance to the voltage drops are suitable to be controlled through critically damped response (e.g., variable frequency drive motors) [50].

An overdamped response is not considered in this paper because it does not achieve any improvement over the critically damped response.

(a) *Underdamped Response.* The underdamped time domain response of the system presented in (27) is given in (30) for the values of α and β that will lead to the underdamped response. The condition to guarantee the underdamped response is given in (31). Hence,

$$\begin{aligned} v_{\text{dc}}(t) = & \frac{-\Delta i_{\text{dc}}}{C \left(\sqrt{(\alpha/2\beta C)^2 - (1/\beta C)} \right)} e^{((\alpha/2\beta C)t)} \\ & \cdot \sin \left(\sqrt{\left(\frac{\alpha}{2\beta C} \right)^2 - \left(\frac{1}{\beta C} \right)} \right) \end{aligned} \quad (30)$$

$$\frac{\alpha^2}{4C} < \beta, \quad (31)$$

where Δi_{dc} represents the step change in the DC bus current. Percentage overshoot for underdamped response is given in

$$\% \text{ OS} = 100e^{(-\zeta\pi/\sqrt{1-\zeta^2})}. \quad (32)$$

Conversely, the damping ratio ζ for a specific percentage overshoot is given in

$$\zeta = \sqrt{\frac{\ln(\% \text{ OS}/100)}{\pi^2 + (\ln(\% \text{ OS}/100))^2}}. \quad (33)$$

The settling time of the underdamped response is given in

$$T_s = \frac{\ln(\text{tolerance fraction})}{\zeta\omega_0}. \quad (34)$$

Equations (32) to (34) can be used to design the SM controller for specific design parameters. Finally, the constraints in (20) must be satisfied for the existence of the SM. Moreover, the selected values of α and β must satisfy the constraint in (31) which is the condition for underdamped response.

(b) *Critically Damped Response.* Critically damped time domain response is given in (35) while ensuring that the values of α and β will lead to the critically damped response. The condition for this response is given in (36). Hence,

$$v_{dc}(t) = \frac{-\Delta i_{dc}}{C}(t) e^{((\alpha/2\beta C)t)} \quad (35)$$

$$\beta = \frac{\alpha^2}{4C}. \quad (36)$$

Finally, the constraints defined in (20) must be satisfied for the existence of SM. Further, (36) must be satisfied for the critically damped response.

(5) *Sliding Mode Hysteresis Control.* In an ideal situation, the SM controller will switch the DC-to-DC converter at infinite frequency with system trajectories moving along the sliding surface when the system enters the SM operation. This condition is shown in Figure 10(a). However, the practical switch of the DC-to-DC converter will experience some switching imperfections and time delays. This will produce a dynamic behavior in the locality of the sliding surface which is identified as chattering as shown in Figure 10(b) [47, 48].

If the chattering produced in the sliding surface is left uncontrolled, the converter will start self-oscillating at a high frequency. This behavior of the converter is not desirable due to the high switching losses. Further, the exact switching frequency in the produced chattering cannot be predicted. Therefore, the converter design and component selection will turn out to be difficult. To solve these issues, the control law u is redefined as

$$u = \begin{cases} 0 = \text{"OFF"} & \text{when } \Psi > k \\ 1 = \text{"ON"} & \text{when } \Psi < -k \\ \text{unchanged} & \text{otherwise,} \end{cases} \quad (37)$$

TABLE 1: DC-to-DC converter parameters.

Parameters	Value
Desired voltage	48 V
Switching frequency	10 kHz
Inductor, L	100 μ H
Capacitor, C	4000 μ F

TABLE 2: Node parameters of DC microgrid.

Parameters	Node-1	Node-2
Desired voltage	48 V	
Source rated power	250 W	500 W
Load resistance	6	
Voltage and current regulation required	$\leq \pm 5\%$	

TABLE 3: Connecting cable parameters of DC microgrid.

Parameters	Branch-12	Branch-23
Current rating	20 A	
Cable resistance	205 m Ω	2 m Ω

where k is a positive number. Usually in SMC, a hysteresis band is introduced to tackle the chattering problem. With this alteration, the converter switch will turn on when $\Psi < -k$ and turn off when $\Psi > k$. In the region $-k \leq \Psi \leq k$, the converter switch remains unchanged and maintains its former state. Therefore, introducing a region $-k \leq \Psi \leq k$ in which no switching occurs, the switching frequency can be controlled by varying the magnitude of k .

3. Results and Discussion

To examine the load sharing performance among parallel connected sources, a DC microgrid with two sources connected in a parallel configuration to the load through connecting lines is proposed and shown in Figure 11. This type of configuration can be easily extendable for more sources and microgrids in parallel configuration. This type of system is attractive for remote areas where the national grid cannot be easily extendable due to the high cost associated with the installation of new transmission lines. Thus, a two-source DC microgrid system for load sharing is simulated using MATLAB/Simulink. Each source consists of a DC-to-DC converter. The parameters of the DC-to-DC converter are selected to support maximum voltage and current levels equal to 50 V and 10 A, respectively. Therefore, the converter supports a maximum power of 500 W. The parameters for the DC-to-DC converter are given in Table 1.

3.1. Results Using Droop Control. The details of nodes and connecting lines are given in Tables 2 and 3, respectively. Each source using droop control is shown in Figure 12.

A two-source DC microgrid shown in Figure 11 is simulated using droop control. To observe the steady-state behavior, assume that the load is drawing rated current and

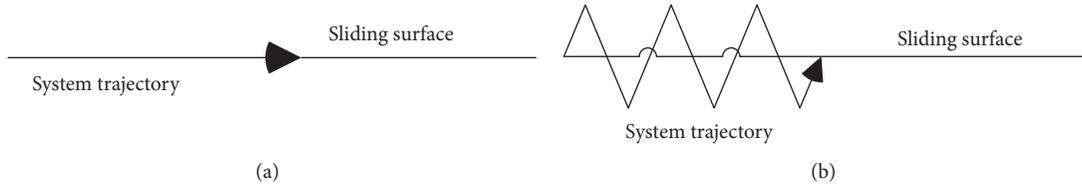


FIGURE 10: System trajectory with (a) ideal SM and (b) practical SM.

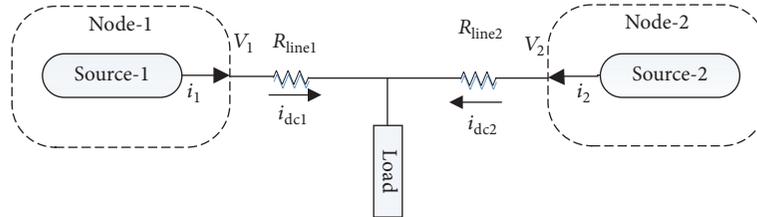


FIGURE 11: A two-source DC microgrid.

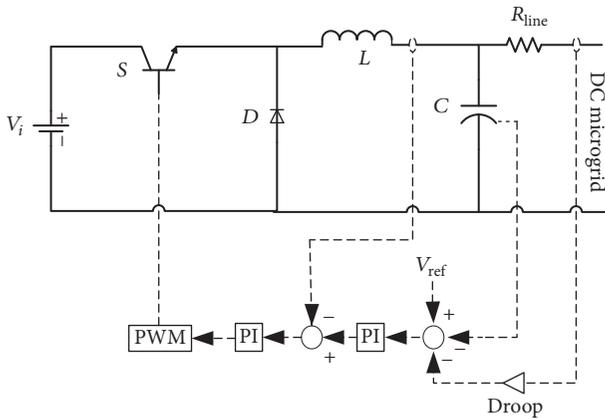


FIGURE 12: DC-to-DC buck converter using droop control.

the steady state is reached. For equal load sharing, source-1 and source-2 are simulated for the same power rating. Thus, droop gains $R_{d1} = R_{d2} = 0.2 \Omega$ are selected for equal load sharing. Node voltage and current by each source are shown in Figure 13.

For droop gain 0.2Ω , the steady-state current supplied by source-1 and source-2 is 2.9 and 4.9 A, respectively, as shown in Figure 13. For equal load sharing, the desired current to be supplied by each source is 4 A. The maximum deviation observed in the supplied current is 27.5%. Steady-state node voltages at source-1 and source-2 are 47.4 and 46.8 V, respectively. The deviation observed in node voltages at no load and full load is 2.5%. These results show that small droop gains ensure decent voltage regulation, but the load sharing performance is not acceptable. For a large droop gain of 1.9Ω , the current supplied by source-1 and source-2 is 3.8 and 4.2 A, respectively. The observed deviation in the supplied currents is 5%, which is acceptable and lower than in the earlier case. But the deviation in node voltages has increased to 16% which cannot be acceptable for the loads.

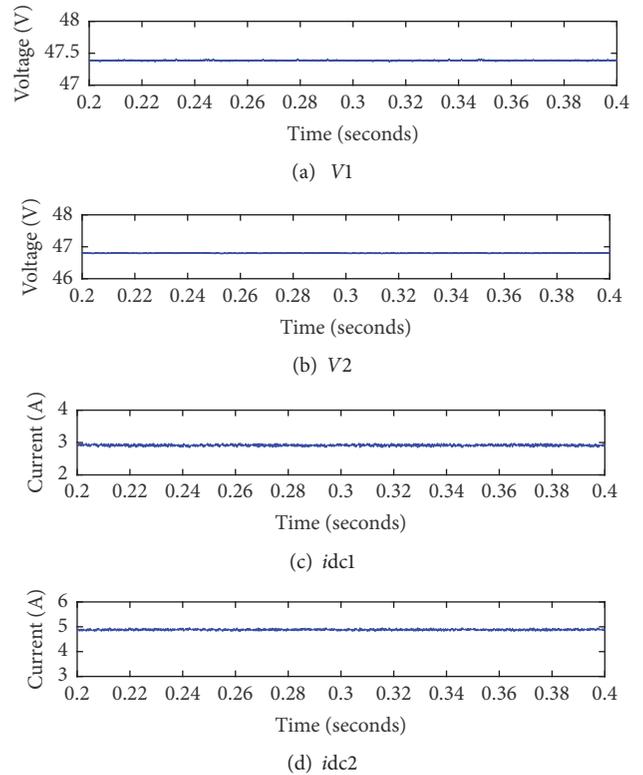


FIGURE 13: Node voltages and source currents with droop gain 0.2Ω .

3.2. Results Using Sliding Mode Control. Figure 14 shows each source with distributed control using the SM controller. The p.u. value of each source current is communicated to the other sources every 10 ms. The total communication delay is around 0.1 ms.

To observe the steady-state behavior with distributed architecture using SM controller, the two-source DC microgrid shown in Figure 11 is simulated and the results are shown in Figure 15. The current supplied by source-1 and source-2 is

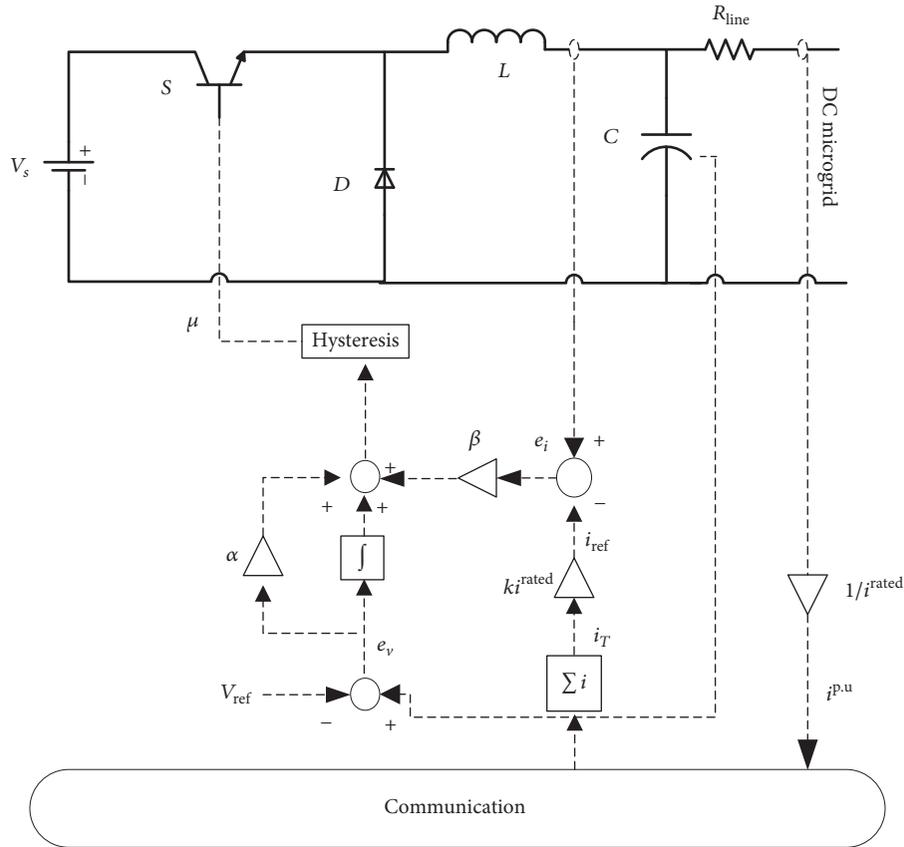


FIGURE 14: DC-to-DC buck converter with a distributed architecture using SM controller.

3.91 and 3.92 A, respectively. The observed deviation in the supplied currents is 2.08% which is a significantly low value compared to the droop controlled DC microgrid. Further, the node voltages at source-1 and source-2 are 47.78 and 47 V, respectively. The deviation observed in the node voltages is 2.25%. This confirms the steady-state load sharing and voltage regulation performance of the proposed distributed SM controller.

Further, Figure 16 shows that source-1 is sharing 25% and source-2 is sharing 75% of the rated load current. The current deviation observed is 4.1% which shows the effectiveness of the proposed architecture using the SM controller. In addition, simulations are carried out to see the effect of connecting line resistance and the results are summarized in Table 4. Each column represents the fixed value of connecting line resistance of source-1, R_{line1} , and each row represents the fixed value of connecting line resistance of source-2, R_{line2} . Each entity in Table 4 represents voltage and current sharing deviation. It can be observed that connecting line resistance affects the load sharing between sources.

To observe the transient condition, a step load of 3 Ω is applied at 0.5 s when the system is operating in steady state at the rated load, as shown in Figure 17. At the instant when a step change in load is applied, node voltages drop shortly as shown in Figure 17. However, within 25 ms, node voltages of source-1 and source-2 settle down to 49.29 and 46.83 V, respectively. This corresponds to a voltage deviation of 2.68%.

The currents supplied are 11.96 and 11.45 A. The deviation in the supplied currents is 4.5%.

Furthermore, the dynamic behavior of node voltage is investigated for underdamped and critically damped responses. The sliding parameters α and β are selected positive according to the conditions defined in (31) and (36). The values of α and β are listed in Table 5. Figure 18 shows node voltage when the load resistance is changed from 6 to 3 Ω at 0.5 s with sliding parameters that show underdamped and critically damped responses. The results of settling time are summarized in Table 5. It can be observed that as the value of ζ is increased from 0.1 to 0.6, the underdamped response is improved with smaller settling time. This response is in good agreement with the presented theory. Additionally, for $\zeta = 1$, the response is critically damped with further improved settling time as presented in theory. These results show the good dynamic performance of the SM controller with distributed architecture.

3.3. Fail-Safe Performance of Distributed Control Architecture.

A significant improvement in the distributed architecture is that it provides high reliability. To prove this claim, a three-source DC microgrid shown in Figure 19 is simulated for fault condition and shown in Figure 20. The parameters of each source are the same as given in Table 1. Source-1 and source-2 are connected to the load through connecting line resistances. But source-3 is directly connected to the load. At steady state,

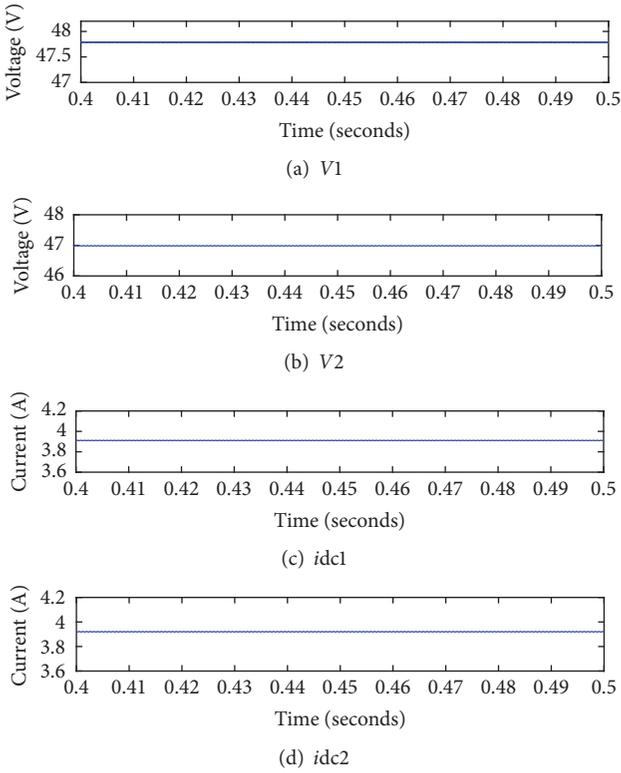


FIGURE 15: Node voltages and source currents with a distributed architecture using SM controller.

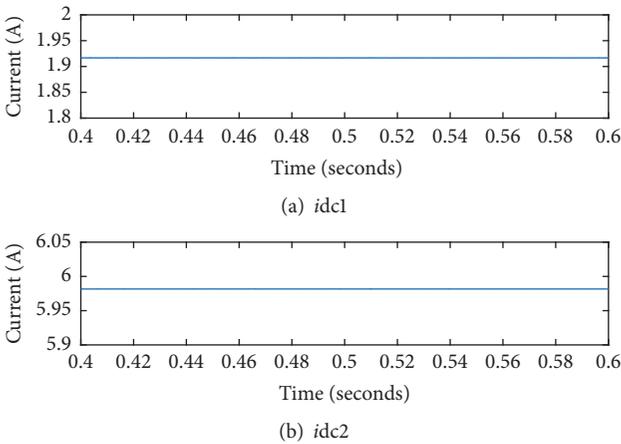


FIGURE 16: Source 1 with 25% and source 2 with 75% load sharing of the rated load.

the voltage across the load is 47.04 V. The three sources supply 3.915, 3.91, and 3.93 A. The maximum deviation observed in the supplied currents is 2.25%.

If one source becomes faulty, the capacity of the other two sources is enough to satisfy the load. Failure of source-2 is simulated by removing its power supply at 0.5 s. Under this fault, the voltage across loads and the supplied currents by sources are shown in Figure 20. After this fault, the system reaches the steady state in about 25 ms. The voltage across loads is maintained at 47.35 V. The supplied current

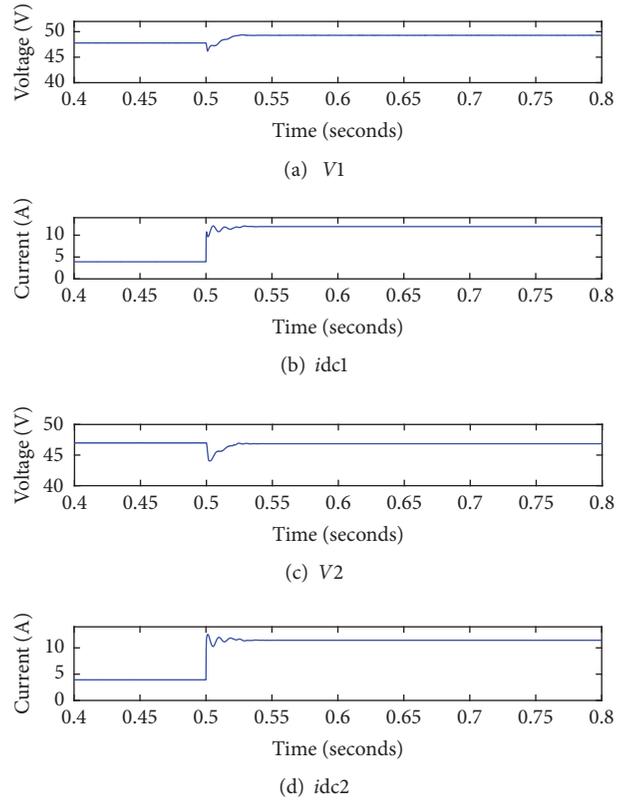


FIGURE 17: Transient response when a step load of 3Ω is applied at 0.5 seconds.

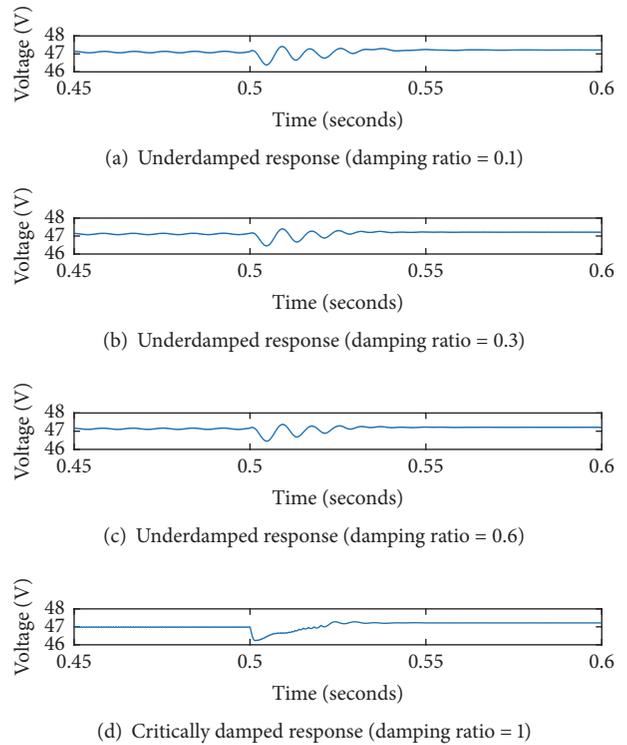


FIGURE 18: Voltage response of a source when load resistance is changed from 6 to 3Ω at 0.5 seconds.

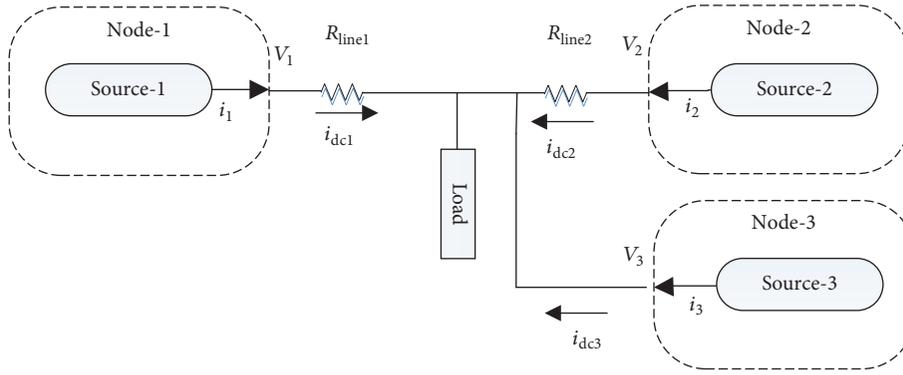


FIGURE 19: A three-source DC microgrid.

TABLE 4: Connecting cable parameters of DC microgrid.

	$R_{line1} = 0.5 \times 205 \text{ m}\Omega$	$R_{line1} = 1 \times 205 \text{ m}\Omega$	$R_{line1} = 2 \times 205 \text{ m}\Omega$
$R_{line2} = 0.5 \times 2 \text{ m}\Omega$	1.92% V/V, 1.9% A/A	2.05% V/V, 2.24% A/A	2.26% V/V, 2.52% A/A
$R_{line2} = 1 \times 2 \text{ m}\Omega$	1.95% V/V, 1.925% A/A	2.08% V/V, 2.25% A/A	2.27% V/V, 1.875% A/A
$R_{line2} = 2 \times 2 \text{ m}\Omega$	1.99% V/V, 1.95% A/A	2.12% V/V, 2.75% A/A	2.27% V/V, 1.925% A/A

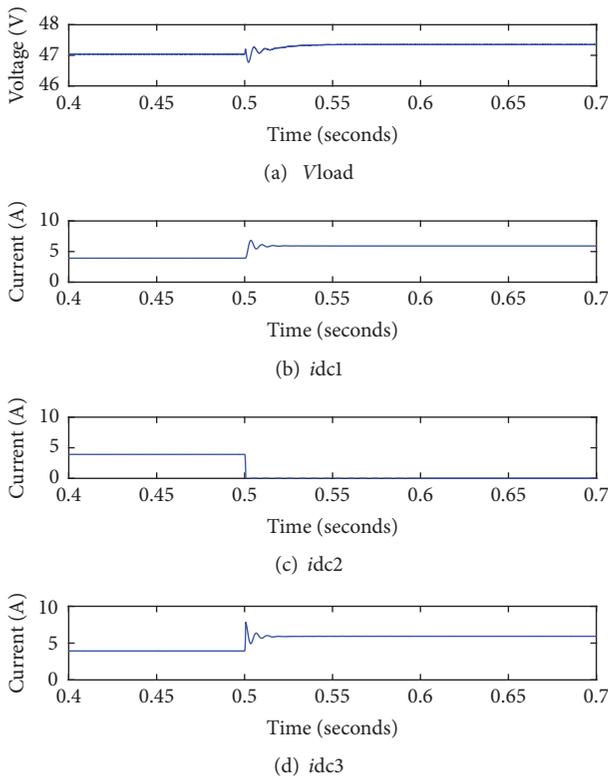


FIGURE 20: Transient response during fault on source-2 at 0.5 seconds.

by source-1 and source-3 is 5.91 and 5.93 A, respectively. This corresponds to 1.5% deviation in the supplied current. This confirms the performance of the distributed architecture using SM controller during source failure. To show the effect of chattering produced, the sliding surface of a source is simulated and shown in Figure 21. Figure 21(a) shows the

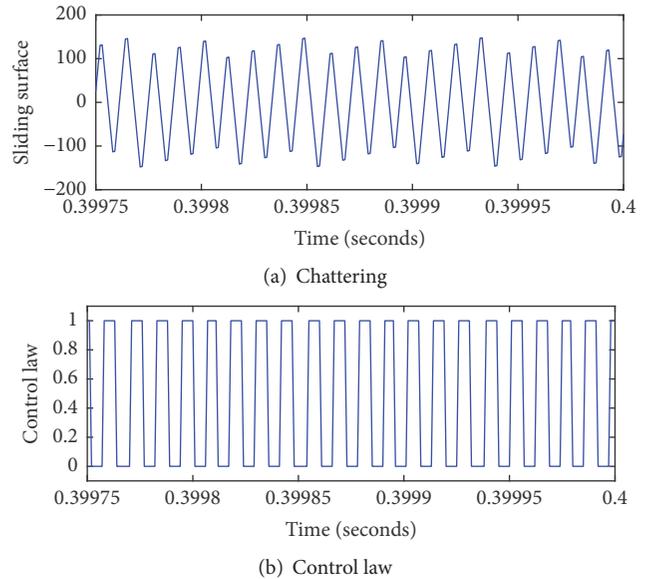


FIGURE 21: Sliding surface with chattering and control law.

produced chattering which is controlled using a hysteresis band as defined in (37). The value of k is selected as 100 based on the criteria reported in [51]. Figure 21(b) shows the control signal which is used to operate the converter switch.

3.4. Problems in Distributive Sliding Mode Application. The delays in the communication channel will directly affect the transient behavior of the system, while the steady-state response is not affected by the communication delays. Further, the SM controller is realized through analog integrated circuits. The following are some design issues in analog implementation.

TABLE 5: Voltage dynamic behavior for different values of ζ .

Damping ratio ζ	Sliding parameter (α, β)	Settling time	Response type
$\zeta = 0.1$	(0.28, 100)	0.05 s (50 ms)	Underdamped
$\zeta = 0.3$	(0.85, 100)	0.04 s (40 ms)	Underdamped
$\zeta = 0.6$	(1.7, 100)	0.03 s (30 ms)	Underdamped
$\zeta = 1$	(2.85, 100)	0.02 s (20 ms)	Critically damped

- (i) Selecting the variables is a serious concern because selecting more variables results in more computations and sensing required which will increase the complexity of the system.
- (ii) Selecting the integral and derivative of the variables involves noise sensitivity. These variables are desired to be indirectly controlled; for example, dV_c/dt can be achieved through sensing capacitor current.
- (iii) The produced chattering is a big challenge in SM. It produces excessive switching losses and limits the selection of the switching device.
- (iv) Restrictions of the analog components (e.g., slew rate, bandwidth, and saturation limits) need to be carefully considered for the proper control operation.

4. Conclusion

A distributed architecture using an SM controller is proposed for proportional load sharing and stability of DC microgrids. DC microgrids are a reliable method to provide efficient power to the consumer in the presence of renewable sources. Droop controllers which are local controllers can achieve good load sharing at the cost of voltage regulation. Further, voltages at different nodes of the DC microgrid are not the same. So, it is difficult to achieve load sharing when the connecting line resistances among the sources are considerable. A centralized controller can achieve these objectives using a high speed communication link. However, it loses reliability due to the single point failure. Additionally, these controllers are realized through proportional integral (PI) controllers which cannot ensure load sharing and stability in all operating conditions. To address limitations, a distributed architecture using an SM controller utilizing low bandwidth communication is proposed for DC microgrids in this paper. A system model is developed and its transversality, reachability, and equivalent control condition are verified. Furthermore, the dynamic behavior of the modeled system is investigated for underdamped and critically damped responses. Detailed simulation results showed good performance of the proposed controller.

Conflicts of Interest

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

References

- [1] K.-T. Mok, M.-H. Wang, S.-C. Tan, and S. Y. R. Hui, "DC electric springs—a technology for stabilizing DC power distribution systems," *IEEE Transactions on Power Electronics*, vol. 32, no. 2, pp. 1088–1105, 2017.
- [2] K. A. Saleh, A. Hooshyar, and E. F. El-Saadany, "Hybrid passive-overcurrent relay for detection of faults in low-voltage DC grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1129–1138, 2017.
- [3] L. Meng, Q. Shafiee, G. Ferrari Trecate et al., "Review on control of DC microgrids," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 99, 1 page, 2017.
- [4] X. Li, L. Guo, S. Zhang et al., "Observer-based DC voltage droop and current feed-forward control of a DC microgrid," *IEEE Transactions on Smart Grid*, vol. 99, 1 page, 2017.
- [5] T. V. Vu, S. Paran, F. Diaz-Franco, T. El-Mezyani, and C. S. Edrington, "An alternative distributed control architecture for improvement in the transient response of DC microgrids," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 574–584, 2017.
- [6] T. Dragicevic, X. Lu, J. C. Vasquez, and J. M. Guerrero, "DC microgrids—part I: a review of control strategies and stabilization techniques," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 4876–4891, 2016.
- [7] S. Anand, B. G. Fernandes, and J. M. Guerrero, "Distributed control to ensure proportional load sharing and improve voltage regulation in low-voltage DC microgrids," *IEEE Transactions on Power Electronics*, vol. 28, no. 4, pp. 1900–1913, 2013.
- [8] M. Rashad, M. Ashraf, A. I. Bhatti, D. M. Minhas, and B. A. Ahmed, "Mathematical modeling and stability analysis of DC microgrid using sm hysteresis controller," *International Journal of Electrical Power & Energy Systems*, vol. 95, pp. 507–522, 2018 (Arabic).
- [9] A. Esmaeli, "Stability analysis and control of microgrids by sliding mode control," *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 22–28, 2016.
- [10] P. Wang, X. Lu, X. Yang, W. Wang, and D. Xu, "An improved distributed secondary control method for DC microgrids with enhanced dynamic current sharing performance," *IEEE Transactions on Power Electronics*, vol. 31, no. 9, pp. 6658–6673, 2016.
- [11] S. D. Tavakoli, M. Mahdavyfakhr, M. Hamzeh, K. Sheshyekani, and E. Afjei, "A unified control strategy for power sharing and voltage balancing in bipolar DC microgrids," *Sustainable Energy, Grids and Networks*, vol. 11, pp. 58–68, 2017.
- [12] S. Peyghami, H. Mokhtari, P. C. Loh, P. Davari, and F. Blaabjerg, "Distributed primary and secondary power sharing in a droop-controlled LVDC microgrid with merged AC and DC characteristics," *IEEE Transactions on Smart Grid*, vol. 99, p. 1, 2016.
- [13] S. Peyghami, P. Davari, H. Mokhtari, P. C. Loh, and F. Blaabjerg, "Synchronverter-enabled DC power sharing approach for LVDC microgrids," *IEEE Transactions on Power Electronics*, vol. 32, no. 10, pp. 8089–8099, 2017.
- [14] Shivam and R. Dahiya, "Robust decentralized control for effective load sharing and bus voltage regulation of DC microgrid based on optimal droop parameters," *Journal of Renewable and Sustainable Energy*, vol. 9, no. 4, p. 045301, 2017.

- [15] Z. Liu, Y. Luo, R. Zhuo, and X. Jin, "Distributed reinforcement learning to coordinate current sharing and voltage restoration for islanded DC microgrid," *Journal of Modern Power Systems and Clean Energy*, pp. 1–11, 2017.
- [16] S. Peyghami, H. Mokhtari, and F. Blaabjerg, "Hierarchical power sharing control in DC microgrids," in *Microgrid, Advanced Control Methods and Renewable Energy System Integration*, vol. 3, pp. 63–100, Elsevier, 1 edition, 2016.
- [17] C. Jin, P. Wang, J. Xiao, Y. Tang, and F. H. Choo, "Implementation of hierarchical control in DC microgrids," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 8, pp. 4032–4042, 2014.
- [18] X. Liu, P. Wang, and P. C. Loh, "A hybrid AC/DC microgrid and its coordination control," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 278–286, 2011.
- [19] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuña, and M. Castilla, "Hierarchical control of droop-controlled AC and DC microgrids: a general approach toward standardization," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 1, pp. 158–172, 2011.
- [20] X. Lu, J. M. Guerrero, K. Sun, J. C. Vasquez, R. Teodorescu, and L. Huang, "Hierarchical control of parallel AC-DC converter interfaces for hybrid microgrids," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 683–692, 2014.
- [21] J. He and Y. W. Li, "An enhanced microgrid load demand sharing strategy," *IEEE Transactions on Power Electronics*, vol. 27, no. 9, pp. 3984–3995, 2012.
- [22] J. M. Guerrero, M. Chandorkar, T.-L. Lee, and P. C. Loh, "Advanced control architectures for intelligent microgrids—part I: decentralized and hierarchical control," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 4, pp. 1254–1262, 2013.
- [23] H. Liang, B. J. Choi, W. Zhuang, and X. Shen, "Stability enhancement of decentralized inverter control through wireless communications in microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 321–331, 2013.
- [24] J. Schönberger, R. Duke, and S. D. Round, "DC-bus signaling: a distributed control strategy for a hybrid renewable nanogrid," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1453–1460, 2006.
- [25] M. C. Chandorkar, D. M. Divan, and R. Adapa, "Control of parallel connected inverters in standalone AC supply systems," *IEEE Transactions on Industry Applications*, vol. 29, no. 1, pp. 136–143, 1993.
- [26] S. V. Iyer, M. N. Belur, and M. C. Chandorkar, "A generalized computational method to determine stability of a multi-inverter microgrid," *IEEE Transactions on Power Electronics*, vol. 25, no. 9, pp. 2420–2432, 2010.
- [27] J. M. Guerrero, J. Matas, L. G. De Vicuña, M. Castilla, and J. Miret, "Wireless-control strategy for parallel operation of distributed-generation inverters," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1461–1470, 2006.
- [28] H. J. Avelar, W. A. Parreira, J. B. Vieira Jr., L. C. G. De Freitas, and E. A. A. Coelho, "A state equation model of a single-phase grid-connected inverter using a droop control scheme with extra phase shift control action," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1527–1537, 2012.
- [29] W. Yao, M. Chen, J. Matas, J. M. Guerrero, and Z.-M. Qian, "Design and analysis of the droop control method for parallel inverters considering the impact of the complex impedance on the power sharing," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 576–588, 2011.
- [30] P. Karlsson and J. Svensson, "DC bus voltage control for a distributed power system," *IEEE Transactions on Power Electronics*, vol. 18, no. 6, pp. 1405–1412, 2003.
- [31] M. Mahmoodi, G. B. Gharehpetian, M. Abedi, and R. Noroozian, "Control systems for independent operation of parallel DG units in DC distribution systems," in *Proceedings of the 1st International Power and Energy Conference, PECon*, pp. 220–224, 28–29 Nov. 2006.
- [32] M. Mahmoodi, G. B. Gharehpetian, M. Abedi, and R. Noroozian, "A suitable control strategy for source converters and a Novel Load- Generation Voltage Control scheme for DC voltage determination in DC distribution systems," in *Proceedings of the 1st International Power and Energy Conference, PECon*, pp. 363–367, Malaysia, November 2006.
- [33] H.-H. Park and G.-H. Cho, "A DC-DC converter for a fully integrated PID compensator with a single capacitor," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 8, pp. 629–633, 2014.
- [34] M. M. Peretz and S. Ben-Yaakov, "Time-domain design of digital compensators for PWM DC-DC converters," *IEEE Transactions on Power Electronics*, vol. 27, no. 1, pp. 284–293, 2012.
- [35] A. Kwasinski, *A microgrid architecture with multiple-input dc/dc converters: applications, reliability, system operation, and control*, Univ. Illinois Urbana-Champaign, Urbana, Ill, USA, 2007.
- [36] S. Singh, D. Fulwani, and V. Kumar, "Robust sliding-mode control of dc/dc boost converter feeding a constant power load," *IET Power Electronics*, vol. 8, no. 7, pp. 1230–1237, 2015.
- [37] S. I. Serna-Garcés, D. G. Montoya, and C. A. Ramos-Paja, "Sliding-mode control of a charger/discharger DC/DC converter for DC-bus regulation in renewable power systems," *Energies*, vol. 9, no. 4, article no. 245, 2016.
- [38] S. I. Serna-Garcés, R. E. Jiménez, and C. A. Ramos-Paja, "Sliding-mode control of a Dc/Dc postfilter for ripple reduction and efficiency improvement in POL applications," *Journal of Applied Mathematics*, vol. 2013, Article ID 915146, 10 pages, 2013.
- [39] N. Vázquez, Y. Azaf, I. Cervantes, E. Vázquez, and C. Hernández, "Maximum power point tracking based on sliding mode control," *International Journal of Photoenergy*, vol. 2015, Article ID 380684, 8 pages, 2015.
- [40] J. Cables-Magsino, R. C. Guevara, and M. Escoto, "Implementation of sliding mode control for current sharing in fixed frequency voltage regulator modules," in *Proceedings of the IEEE Region 10 Conference: Sustainable Development Through Humanitarian Technology, TENCON 2012*, Philippines, November 2012.
- [41] W. Qiu and Z. Liang, "Practical design considerations of current sharing control for parallel VRM applications," in *Proceedings of the 20th Annual IEEE Applied Power Electronics Conference and Exposition, APEC*, pp. 281–286, USA, March 2005.
- [42] T. Dragičević, J. M. Guerrero, and J. C. Vasquez, "A distributed control strategy for coordination of an autonomous LVDC microgrid based on power-line signaling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3313–3326, 2014.
- [43] H. Valderrama-Blavi, J. M. Bosque, F. Guinjoan, L. Marroyo, and L. Martínez-Salamero, "Power adaptor device for domestic DC microgrids based on commercial MPPT inverters," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 1191–1203, 2013.
- [44] R. Silva-Ortigoza, V. Hernández-Guzmán, M. Antonio-Cruz, and D. Muñoz-Carrillo, "DC/DC Buck power converter as a

- smooth starter for a DC motor based on a hierarchical control,” *IEEE Transactions on Power Electronics*, vol. 30, no. 2, pp. 1076–1084, 2015.
- [45] Y. Zhao, W. Qiao, and D. Ha, “A sliding-mode duty-ratio controller for DC/DC buck converters with constant power loads,” *IEEE Transactions on Industry Applications*, vol. 50, no. 2, pp. 1448–1458, 2014.
- [46] K. K. S. Leung and H. S. H. Chung, “Derivation of a second-order switching surface in the boundary control of buck converters,” *IEEE Power Electronics Letters*, vol. 2, no. 2, pp. 63–67, 2004.
- [47] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electromechanical Systems*, Taylor and Francis, London, U.K, 1999.
- [48] J. J. E. Slotine and W. Li, *Sliding control in Applied Non-linear Control*, Prentice-Hall, Inc, Englewood Cliffs, NJ, USA, 1991.
- [49] Intel. Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1—Design Guidelines, Intel Corporation: Santa Clara, CA, USA, 2009.
- [50] T. Wildi, *Electrical Machines, Drives and Power Systems*, Prentice Hall, Columbus, OH, USA, 6 edition, 2006, Electrical Machines, Drives and Power Systems.
- [51] B. Shrestha, N. R. Bhuyan, and B. N. Sinha, *Sliding Mode Control of Switching Power Converters Technique and Implementation*, vol. 4, CRC press, Taylor & Francis Group, Boca Raton, NY, USA, 1st edition, 2012.

Research Article

Design, Analysis, and Experiment for Rescue Robot with Wheel-Legged Structure

Meng Ning,^{1,2} Bilun Xue,¹ Zefeng Ma,³ Changhong Zhu,¹ Zihao Liu,¹
Cuncai Zhang,¹ Yao Wang,¹ and QiuJu Zhang^{1,2}

¹School of Mechanical Engineering, Jiangnan University, Wuxi 214122, China

²Jiangsu Key Laboratory of Advanced Food Manufacturing Equipment and Technology, Jiangnan University, Wuxi 214122, China

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

Correspondence should be addressed to Meng Ning; ningmeng0608@126.com

Received 5 June 2017; Revised 26 September 2017; Accepted 12 October 2017; Published 6 November 2017

Academic Editor: Carlos E. Ugalde-Loo

Copyright © 2017 Meng Ning 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 wheel-legged rescue robot design with strong environmental adaptability is proposed. The design presented is aimed at helping rescue workers complete their missions, such as environmental and personnel search, quickly and accurately. So it has broad application prospects. In order to achieve the advantages of simple structure, easy control, small occupation space, and wide motion range, a wheel-legged rescue robot is designed in this paper, and the robot can realize three kinds of motion states, which include wheel state, rotation center lifting process, and leg state. Then the motion states are analyzed in detail, which provides a reference for motion control. Considering the wheel state and leg state share the same structure to contact with the ground, the effect of the stiffness of wheel-legged structure to the motion performance is analyzed. Then the experiment is carried out to prove the feasibility of the structure design. This study offers a design and quantitative analysis for wheel-legged rescue robot. Furthermore, a basis for future control research and engineering applications is established.

1. Introduction

The complex and dangerous environment after the disaster can cause serious danger to the life safety of rescue workers. Rescue robot can perform many tasks instead of rescue workers, such as environmental monitoring and personnel search, and it can improve the rescue efficiency and information accuracy. So the rescue robot has broad application prospect [1–4]. Because the postdisaster environment is very complex, the adaptability of complex environment is the basic function of the rescue robot.

According to the different movement forms, rescue robots can be divided into four categories: tracked rescue robot [5–8], wheeled rescue robot [9], multilegged rescue robot [10, 11], and snake-like rescue robot [12, 13]. However, the movement environment and function of the above four kinds of robots have some limitations. In order to solve the above problems, many researchers have studied the wheel-legged robot connecting the high obstacle capability

of multilegged robot and high mechanical efficiency of wheeled robot. According to the different structure form, wheel-legged structure of the robot can be divided into series structure, similar parallel structure, and simplified structure. The leg of mammals and insects has three parts: thigh (femur), calf (tibia), and foot (tarsus); thus, the leg can be considered as a series mechanism [14]. Many wheel-legged structures were designed by mimicking biological leg structure and have the advantages of high flexibility. For example, California Institute of Technology designed a six-wheel-legged mobile robot ATHLETE in 2008. Each leg of this robot has six DOF (degrees of freedom), and each wheel-legged can change posture flexibly to adapt to different terrain or across obstacles [15, 16]. Amar et al. [17] designed a wheel-legged robot, and each leg has three DOF. An electrical actuator with a ball screw is mounted on each sliding joint. Besides, many scientific research institutions [18–21] have carried out in-depth research on the structure design and performance analysis of this type of wheel-legged robots.

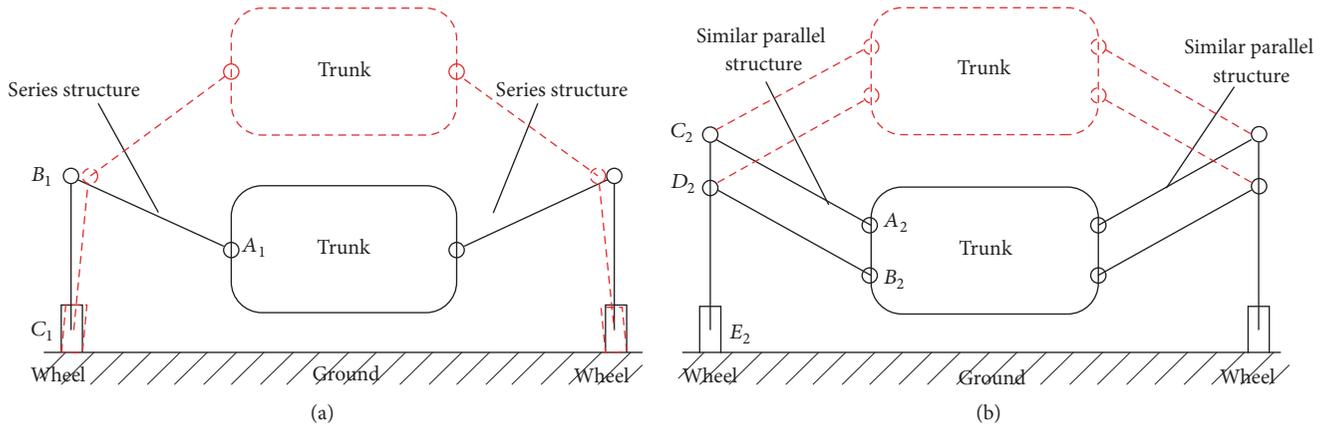


FIGURE 1: Common wheel-legged structures. (a) Series wheel-legged structure. (b) Four-bar wheel-legged structure.

Similar parallel structure has the advantages of high load capacity and rigidity, and some research institutions also have studied this type of robot. For example, Wang et al. [22] designed a wheel-legged rescue robot, and the leg is a (2-VPS+U)R series-parallel mechanism. Alamdari et al. [23] explored the use of various candidate articulated leg-wheel subsystem designs (based on the four-bar mechanism) to enhance locomotion capabilities of land-based vehicles. Besides, Luo et al. [24] and Siegwart et al. [25] also studied this type of wheel-legged robots. For some wheel-legged robots, wheel structure can directly change into leg structure, which is totally different from the biological legs. For example, Kim et al. [26] designed a wheel-leg hybrid robot which utilizes a novel transformable wheel that combines the advantages of both circular and legged wheels. Tadakuma et al. [27] and Chen et al. [28] also have studied this type of robot.

Although many robots with different wheel-legged structures have been designed and the movement performances have been analyzed, the wheel-legged robots still have the disadvantages of complex structure, difficult control, large space required for wheel-legged translation, and limited movement environment. This limits the practical application of the wheel-legged rescue robots. In order to solve the above problems, a new wheel-legged rescue robot is designed and the motion states are analyzed to provide a reference for motion control. Considering the wheel state and leg state share the same structure to contact with the ground, the elastic deformation of the wheel-legged structure is also analyzed. The experimental results prove the feasibility of the structural design and lay the foundation for further research on control methods.

2. Structure Design

2.1. Structure Design of Wheel-Legged Rescue Robot. The common series wheel-legged structure and similar parallel structure are shown in Figure 1. In Figure 1(a), link A_1B_1 and link B_1C_1 are connected by hinge, and the wheels are added at the end of the leg. In Figure 1(b), links A_1B_1 , B_1D_1 , D_1C_1 , and C_1A_1 consist of four-bar wheel-legged structure. In Figure 1, the solid line and dotted line represent the wheel state and leg

state of the robot, respectively. It can be known from Figure 1 that the robot needs a large space when the motion state is converted between the wheel state and leg state, and this limits the application of rescue robot in narrow space.

Considering the arc leg, which is a part of circle, also can make multilegged robot walk, the transformation process from wheel state to leg state is shown in Figure 2. When the left half arc structure and the right half arc structure form a circle, the robot is in the wheel state. And when the left half arc structure rotates around the rotation axis and fits in the right half arc structure, the robot is in the leg state. This kind of wheel-legged structure is simpler than other wheel-legged structure, and the transformation process between wheel state and leg state takes up small space. In particular, when the motion process of the robot changes from wheel state to leg state, the rotational center needs to be raised. So the wheel-legged robot has three kinds of motion states: wheel state, rotation center lifting process, and leg state.

The drive schematic of wheel-legged structure is shown in Figure 3. Motor 1 is fixedly connected with gear 1, and motor 2 is fixedly connected with gear 3. Gear 1 rotates under the drive of motor 1. Because gear 2 is engaged with gear 1, gear 2 is driven to rotate by gear 1. Gear 2 is connected with the fixing link through connecting plate, and the fixed link fixedly connected with the wheel. So the robot can realize the wheel movement by the driving of motor 1. Motor 2 is fixedly connected with gear 3, and gear 3 is engaged with the rack. At this time, the position of the rotation center can be lifted by the driving of motor 2. In particular, the fixed link and rack are fixed to the base plate. So when the robot is in the wheel state, the fixed link rotates, and the rack also rotates. At this time, gear 3 rotates inevitably. Motor 2 should match the motion of motor 1 when the robot is in wheel state. After the rotation center being lifted to specified location, the upper contact point contacts with the lower contact point. The circuit of the motor 3 is connected. Then the left half arc structure rotates by the driving of motor 3 until two half arc structures are fit together. When the rescue robot is in the leg state, the legs swing under the action of motor 1 and motor 2. The overall structure diagram of wheel-legged rescue robot is shown in Figure 4.

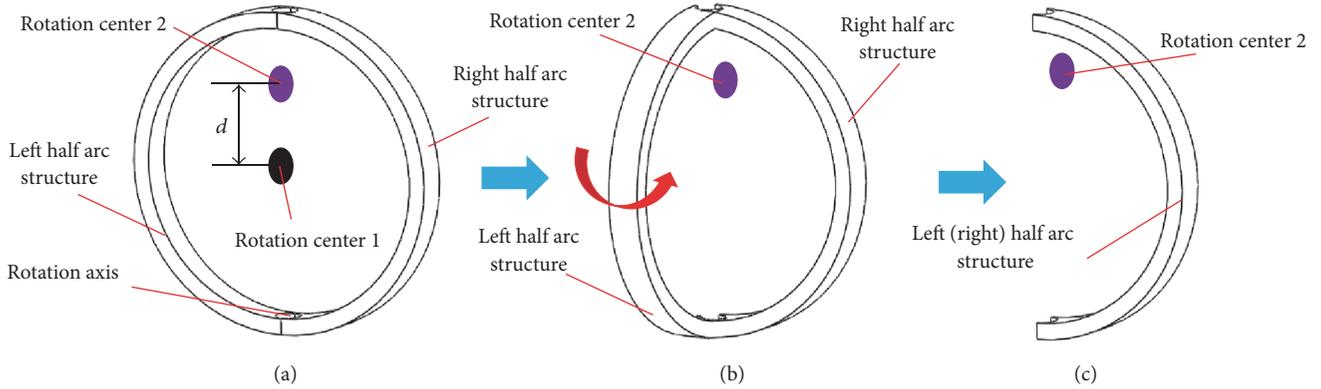


FIGURE 2: Transformation process from wheel state to leg state. (a) Wheel state. (b) Transformation process. (c) Leg state.

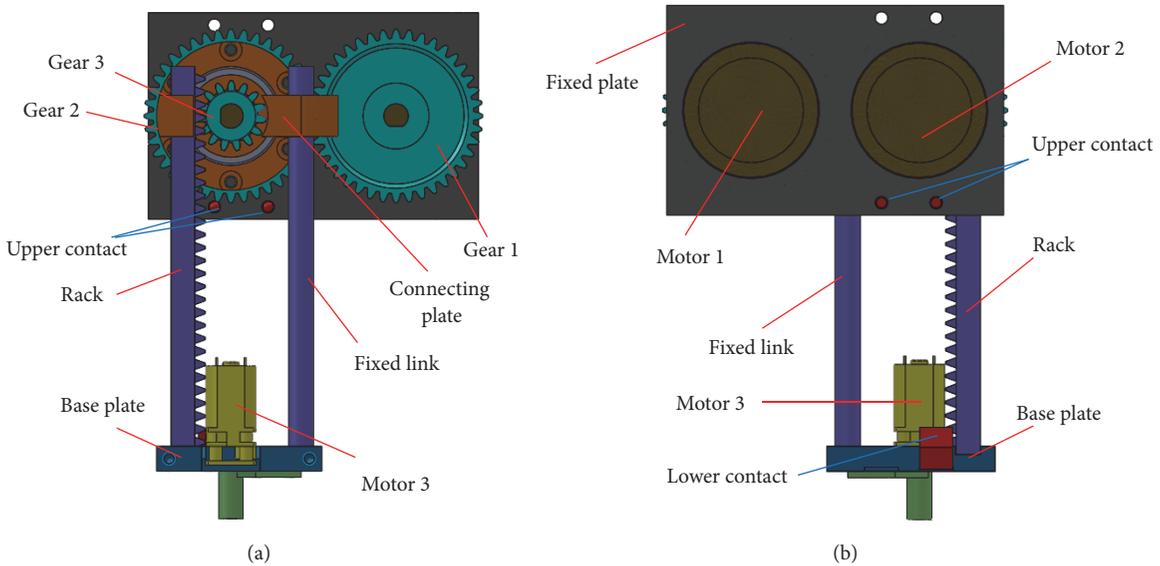


FIGURE 3: Drive schematic of wheel-legged structure. (a) Main view of main structure. (b) Postview of main structure.

2.2. *Application Range of Wheel-Legged Rescue Robot.* One of the biggest characteristics of the wheel-legged robot with semicircle legs is that it can not only move on the rugged road (Figure 5(a)), but also move in water (Figure 5(b)). This is what other wheel-legged robots, whether the series wheel-legged structure or similar parallel wheel-legged structure, do not have. When the wheel-legged robot moves on land in leg state, the wheel-legged structure can either turn clockwise or counterclockwise. And when the robot moves in the water, the wheel-legged structure can only turn counterclockwise. At this time, the wheel-legged structures are equivalent to the oars. This has greatly broadened the applications scope of robot.

3. Motion States Analysis

3.1. *Wheel State.* The rescue robot needs to overcome all kinds of obstacles, which include road resistance, which is caused by unevenness of the road surface and slope

of the ground, and inertia resistance, which is caused by overcoming the inertia of the robot.

Road resistance can be shown as [29]

$$R_R = R_f + R_i = \mu_1 \cdot mg + \mu_2 \cdot \sin \theta \cdot mg, \quad (1)$$

where R_f is the rolling resistance, which is related to the type of road, the velocity, and the structure of the tire. μ_1 is the rolling resistance coefficient. R_i is the slope resistance. θ is the angle of slope. μ_2 is the slope resistance coefficient, and it is greater than 1.

Inertia resistance includes the inertia force and inertia moment. The robot is composed of many parts, and the inertia force and inertia moment of each part are not the same. For easy calculation, the inertia resistance can be shown as

$$R_I = \mu_3 \cdot ma, \quad (2)$$

where m and a are the total mass and the acceleration of the center of mass, respectively. μ_3 is the equivalent coefficient,

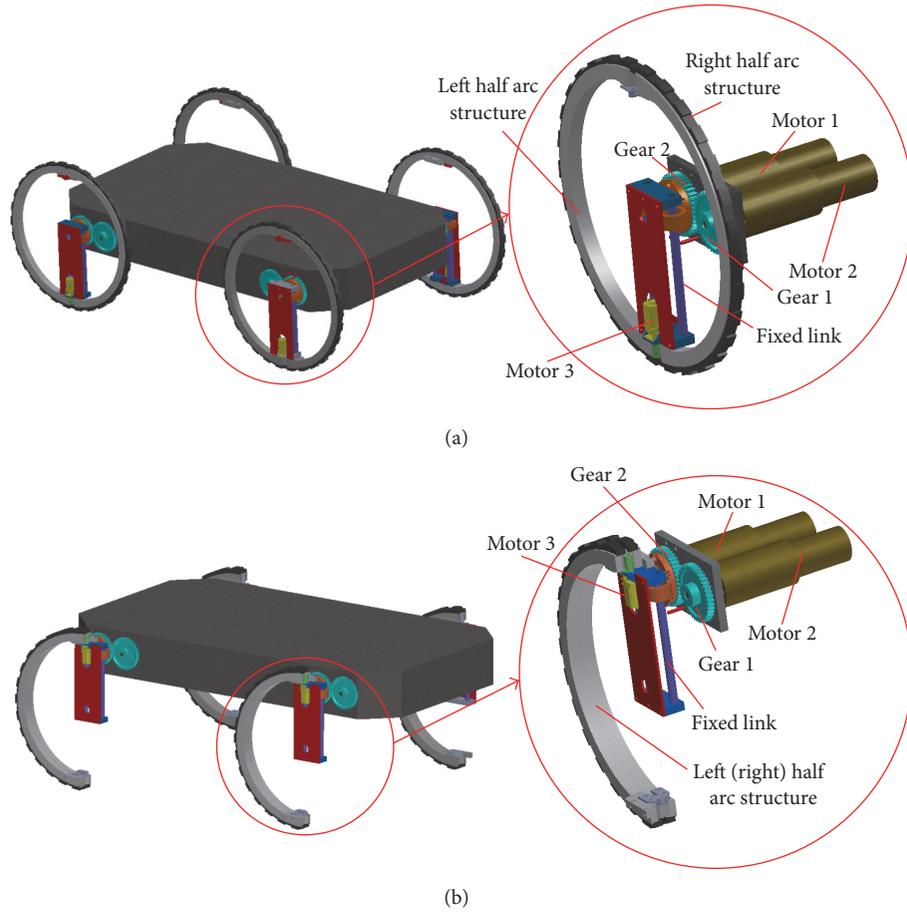


FIGURE 4: Overall structure diagram of wheel-legged rescue robot. (a) Robot in wheel state. (b) Robot in leg state.

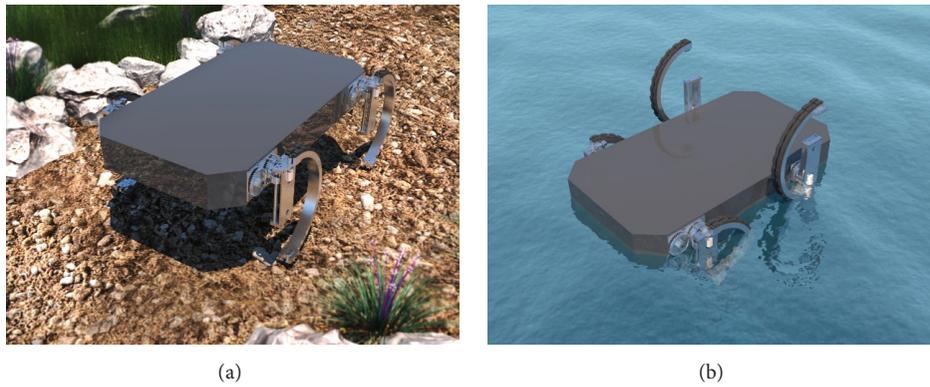


FIGURE 5: Movement environment of wheel-legged rescue robot. (a) Moving on rough terrain. (b) Moving in the water.

and it is greater than 1. At this time, the resistance of the wheel-legged rescue robot in wheel state can be shown as

$$R = R_R + R_I. \quad (3)$$

According to the structural characteristics, the torque of motor 1 can be shown as

$$M_1 = \mu_4 \cdot R \cdot \frac{D}{2} \cdot \frac{i_1}{i_2}, \quad (4)$$

where D is the diameter of the wheel, i_1 is the number of teeth of gear 1, and i_2 is the number of teeth of gear 2. μ_4 is the safety factor, and it is also greater than 1.

When the rescue robot is in the wheel state, in addition to motor 1, motor 2 also should work at the same time. So the speed of the two motors should be matched. The rotation relationship between motor 1 and motor 2 can be shown in Figure 6. If gear 3 does not rotate and only the rack rotates from position 1 to position 2, the relative position of gear

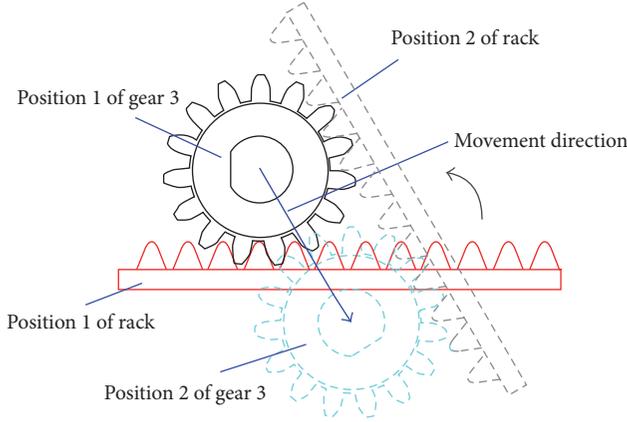


FIGURE 6: Rotation relationship between motor 1 and motor 2.

and rack changes, which can lead to the rotation center of wheel-legged changes. At this time, the wheel turns into a “cam.” According to geometric relationship, the relationship between the speeds of the two motors can be expressed as

$$n_3 = n_1 \cdot \frac{i_1}{i_2}, \quad (5)$$

where n_1 is the speed of motor 1 and n_3 is the speed of motor 3. The rotation directions of the two motors are the same.

3.2. Rotation Center Changing Process. The rotation center can be changed under the action of motor 2, and it is also the change process of movement state between wheel state and leg state. When the trunk is lifted, the torque can be shown as

$$M_{l2} = \mu_5 \cdot \frac{1}{4} m g \cdot r_{g3}, \quad (6)$$

where m is the mass of robot; r_{g3} is the radius of gear 3. μ_5 is the safety factor, and it is greater than 1. When the trunk moves downwards, motor 2 plays a limiting role. Besides, when the left half arc structure rotates, motor 3 is mainly to overcome the inertia resistance, and the torque of the motor 3 can be calculated according to (3).

3.3. Leg State. There are two walking ways when the robot is in wheel state: arcuate part of the wheel-legged structure touches the ground first and endpoint of the wheel-legged structure touches the ground first. For the former, the motion sequence can be shown in Figure 7(a). When the robot walks on the horizontal ground, the midpoint of the semicircle arc lands first, and then the structure rotates clockwise until the end point contacts with the ground. This movement mode is suitable for the robot to move on a relatively flat ground, and it has the advantage of high efficiency. For the latter, the motion sequence can be shown in Figure 7(b). The endpoint of the semicircle arc lands first, and then the structure rotates around the landing point until the diameter is perpendicular to the ground. This movement mode is suitable for the stepped road because the endpoint of the semicircle arc is not easy to slide.

In order to analyze the driving torque of the wheel-legged structure when the robot is in the walking process, force analysis should be conducted. For the first walking way, the moment balance equation is

$$\mathbf{M} + \mathbf{F}_{S1} \times \mathbf{d}_{S1} + m_l \mathbf{g} \times \mathbf{d}_{S2} + \mathbf{F}_{c1} \times \mathbf{d}_{c1} + \mathbf{M}_{c1} = 0, \quad (7)$$

where \mathbf{M} is the driving torque, \mathbf{F}_{S1} is the force of the trunk to the leg, and m_l is the mass of the wheel-legged structure. \mathbf{F}_{c1} and \mathbf{M}_{c1} are inertia force and inertia moment, respectively. \mathbf{d} is the vector from the point of force to the contact point between the wheel-legged structure and the ground. To simplify the calculation process, consider the speed of the leg in the swing phase is large and the speed of the leg in support phase is small, regardless of the inertia force and inertia moment. And safety factor can be introduced. According to the geometric relationship, (7) can be simplified as

$$M = \mu_6 \left[F_{S1} \cdot r \cos \alpha + m_l g \cdot l_{S1} \cdot \cos \left(\frac{\pi}{2} - \beta \right) \right], \quad (8)$$

where $l_{S1} = r\sqrt{2-2\cos\alpha}$; $\beta = \sin\alpha/\sqrt{2-2\cos\alpha}$. r is the radius of the wheel-legged structure, and α is rotation angle of leg. μ_6 is the safety factor.

For the second walking way, the moment balance equation is the same as (7). In particular, gravity always does negative work for the first walking way, and gravity does positive work and then does negative work for the second walking way. The driving torque of the robot with the second walking way can be shown as

$$M = F_{S1} \cdot 2r \cos \alpha + m_s g \cdot \sqrt{2}r \cdot \sin \left(\frac{\pi}{4} - \alpha \right). \quad (9)$$

The initial angle can be expressed as

$$\alpha_0 = \arcsin \frac{h}{r}, \quad (10)$$

where h is the height of step. Suppose the mass of trunk of robot is 40 Kg, $m_l = 0.5$ Kg, $h = 0.1$ m, and $r = 0.2$ m. The variation law of driving force is shown in Figure 8. Horizontal axis represents the rotation angle, and vertical coordinate represents torque. It can be known from Figure 8 that average torque of the robot in the second walking way is greater than average torque in the first walking way.

After knowing the driving torque, the motion speed of the robot also can be obtained. For the first walking way, the relationship between motor speed and motion speed of robot is

$$\begin{aligned} v_x &= r\dot{\theta} \sin \theta + r \\ v_y &= r\dot{\theta} \cos \theta. \end{aligned} \quad (11)$$

For the second walking way, the relationship between motor speed and motion speed of robot can be shown as

$$\begin{aligned} v_x &= 2r\dot{\theta} \sin \theta \\ v_y &= 2r\dot{\theta} \cos \theta. \end{aligned} \quad (12)$$

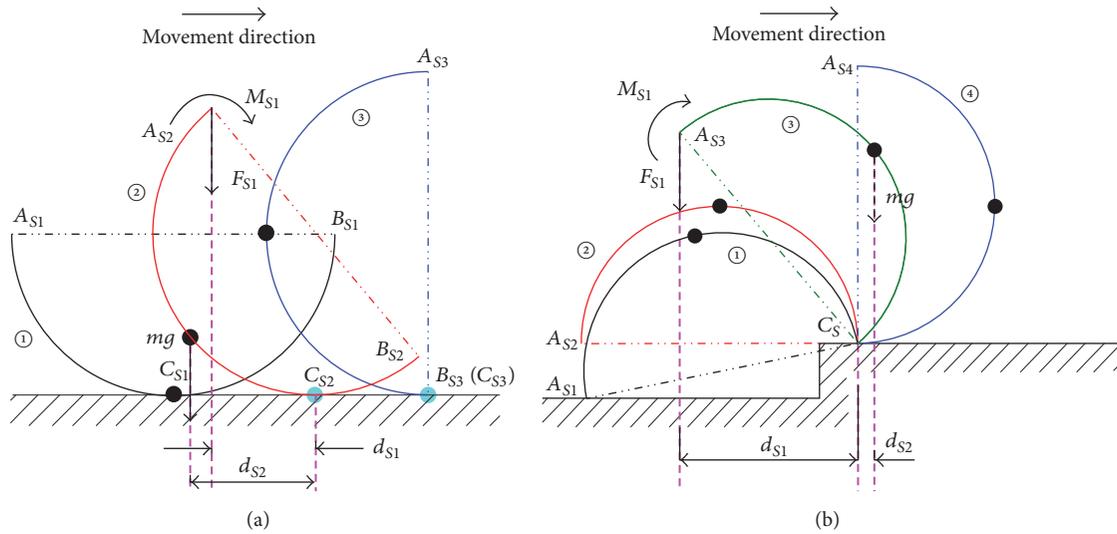


FIGURE 7: Walking ways of the rescue robot. (a) Arcuate part of the wheel-legged structure lands first. (b) Endpoint of the wheel-legged structure lands first.

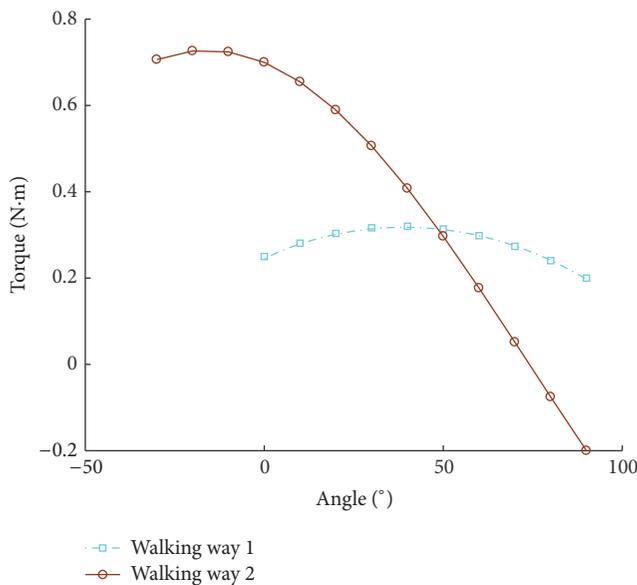


FIGURE 8: Variation law of driving torque in leg state.

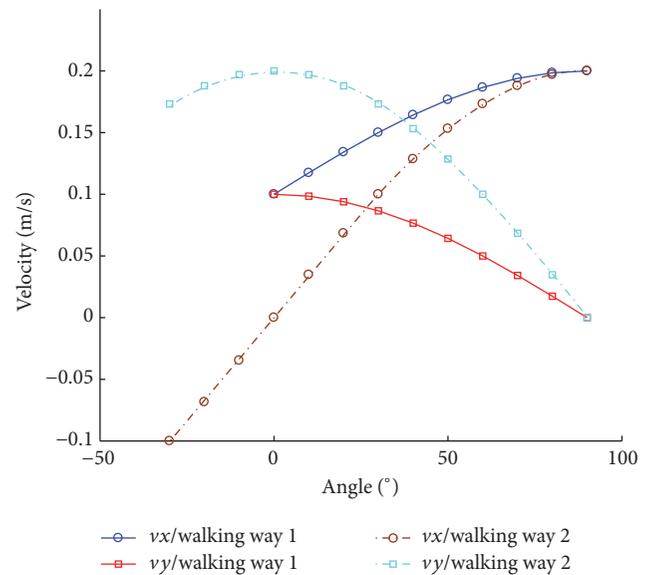


FIGURE 9: Variation law of speed in leg state.

When the speed of motor is 1 rad/s, the relationship between the rotation angle of the wheel-legged structure and the speed of the robot can be shown in Figure 9. In the leg state, the robot has both the speed along the horizontal direction and the speed along the vertical direction. When the robot moves in the second walking way, the speed along the vertical direction is greater, and this also shows that the robot is more suitable for ladder-type pavement with the second walking way.

Based on the analysis results of the wheel state, rotation center changing process, and leg state, motion law of wheel-legged structure and motor torques can be obtained. In different motion states, the required motor torque is also

different. After a comprehensive consideration, the model of motor 1 and motor 2 is maxon RE35, and the rated torque is 2.5 Nm. The model of motor 3 is GA12YN20, and the rated torque is 0.7 Nm.

In particular, when the robot is in leg state, it also can move in the water. At this time, each leg is affected by the force of water on it. The force balance equation can be shown as

$$\mathbf{F}_d + \mathbf{F}_{cd} + \mathbf{F}_t = 0 \quad (13)$$

$$\mathbf{I}_{d\perp} \times \mathbf{F}_d + \mathbf{M}_{cd} + \mathbf{M}_d = 0,$$

where \mathbf{M}_d is motor torque, \mathbf{M}_{cd} and \mathbf{F}_{cd} are inertia force and inertia moment, respectively, \mathbf{F}_d is the force of water on

wheel-legged structure, F_t is the force of trunk on wheel-legged structure, and $l_{d\perp}$ is acting arm. For the wheel-legged rescue robot, it suffers friction, pressure difference resistance, wave resistance, and air resistance. Because the velocity is not great, the friction of water and air resistance is very small. The pressure difference resistance is proportional to the square of the velocity of robot. The wave resistance is introduced due to the wave of water. In order to simplify the calculation, it can be considered proportional to the speed of the robot. The force balance equation of robot can be written as

$$(m_{d1} + m_{d2}) \frac{dv_d}{dt} = -av_d - kv_d^2 + F_d, \quad (14)$$

where m_{d1} is the mass of robot, m_{d2} is the mass of load, and v_d is the velocity of robot. a and k are coefficients of wave resistance and pressure difference resistance, respectively. F_d is the force of the water to the robot. The calculation method of F_d is different when the underwater environment is different. For example, the wave force in the regular wave can be written as [30]

$$F_d = \rho V \left[1 + K_{33} \frac{\omega_e}{\omega} \right] \omega^2 k_1 e^{(-\omega^2/g)H} \zeta_a e^{i\omega_e t}, \quad (15)$$

where ρ is the density of water, V is the drainage volume, and K_{33} is the vertical and longitudinal additional mass coefficients. ω_e and ω are encounter frequency and wave circle frequency, respectively. k_1 is reduction coefficient, which represents the influence of the ratio of the length of the underwater robot and the effective wavelength on the wave force. ζ_a is amplitude of regular wave.

4. Structural Stiffness Analysis

Because the disaster environment is very complex, the rescue robot should have good buffering performance. Otherwise, whether the robot is in the wheel state or in the leg state, the wheel-legged structures of the robot may have rigid collision with the ground frequently and further lead to a series of problems. Considering the wheel state and leg state share the same structure to contact with the ground, the effect of structural stiffness of wheel-legged structure in wheel state and leg state on motion performance is analyzed.

4.1. Structural Stiffness in Wheel State. The wheel should be elastic wheel, which means that hub should be able to achieve deformation under external force. In addition to the elastic deformation of the wheel, the tire can also achieve elastic deformation, so the motion of the robot on the rough road surface can be simplified to a two-DOF vibration system in the vertical direction.

According to Lagrange equation, motion equations of equivalent vibration system can be expressed as [31, 32]

$$\begin{aligned} m_2 \ddot{z}_2 &= k(z_1 - z_2) + c(\dot{z}_1 - \dot{z}_2) \\ m_1 \ddot{z}_1 &= c(\dot{z}_2 - \dot{z}_1) + k(z_2 - z_1) - k_t(z_1 - q), \end{aligned} \quad (16)$$

where z_1 and z_2 are vertical displacement of wheel and body, m_2 is the mass of robot, k is the equivalent stiffness coefficient

of wheel, c is the damping of wheel, m_1 is the mass of wheel, and k_t is the equivalent stiffness coefficient of tyre. q is the input of road roughness, and it is harmonic exciting force. The output can be written as $z_1 = z_{10} e^{i\omega t + \varphi}$ and $z_2 = z_{20} e^{i\omega t + \varphi}$. Inserting them into (16), it can be written as

$$\begin{aligned} z_2 (-\omega^2 m_2 + i\omega c + k) &= z_1 (i\omega c + k) \\ z_1 (-\omega^2 m_1 + i\omega c + k + k_t) &= z_2 (i\omega c + k) + q k_t. \end{aligned} \quad (17)$$

At this time, the frequency response function can be shown as

$$\begin{aligned} \frac{z_2}{z_1} &= \frac{i\omega c + k}{-\omega^2 m_2 + i\omega c + k} \\ \frac{z_1}{q} &= \frac{A_2 k_t}{A_3 A_2 - A_1^2} \\ \frac{z_2}{q} &= \frac{z_2}{z_1} \cdot \frac{z_1}{q} = \frac{A_1 k_t}{A_3 A_2 - A_1^2}, \end{aligned} \quad (18)$$

where $A_1 = i\omega c + k$; $A_2 = -\omega^2 m_2 + i\omega c + k$; $A_3 = -\omega^2 m_1 + i\omega c + k + k_t$. The amplitude frequency characteristics of the robot relative to road surface are obtained:

$$\left| \frac{z_2}{q} \right| = \gamma \sqrt{\frac{(1 - \lambda^2)^2 + (2\xi\lambda)^2}{\Delta}}, \quad (19)$$

where $\Delta = [(1 - \lambda^2)(1 + \gamma - \lambda^2/\mu)^2 - 1] + (2\xi\lambda)^2 [\gamma - (1/\mu + 1)\lambda^2]^2$. λ is the frequency ratio, μ is mass ratio of body and wheel, and ξ is damping ratio. According to the calculation results of the three-dimensional software (Solidworks), $m_1 = 0.5$ Kg, $m_2 = 40$ Kg. According to the vibration model of the car, $k_t = 200000$ N/m [33]. The amplitude frequency characteristics of robot body with different k can be shown in Figure 10(a). As can be seen from Figure 10(a), the reduction of the spring stiffness coefficient can effectively reduce the resonance peak. But the increase of the spring stiffness coefficient can shock absorption. Amplitude frequency characteristics of body acceleration to road surface velocity can be written as

$$\left| \frac{\ddot{z}_2}{\dot{q}} \right| = \left| \frac{z_2 \omega^2}{q \omega} \right| = \omega \left| \frac{z_2}{q} \right|. \quad (20)$$

The amplitude frequency characteristics of relative dynamic load on road surface velocity can be shown as

$$\frac{F_d}{G\dot{q}} = \frac{m_2 \ddot{z}_2}{m_2 g \dot{q}} = \frac{1}{g} \frac{\ddot{z}_2}{\dot{q}}. \quad (21)$$

The amplitude frequency characteristics of body acceleration to road surface velocity and relative dynamic load on road surface velocity can be shown in Figures 10(b) and 10(c), respectively. The trends are almost the same. In low-frequency section and resonance section, the bigger k is, the better the ride comfort is. And in high frequency section, the smaller k is, the better the ride comfort is.

It can be known from Figure 10 that the decrease of stiffness coefficient can effectively reduce the resonance peak.

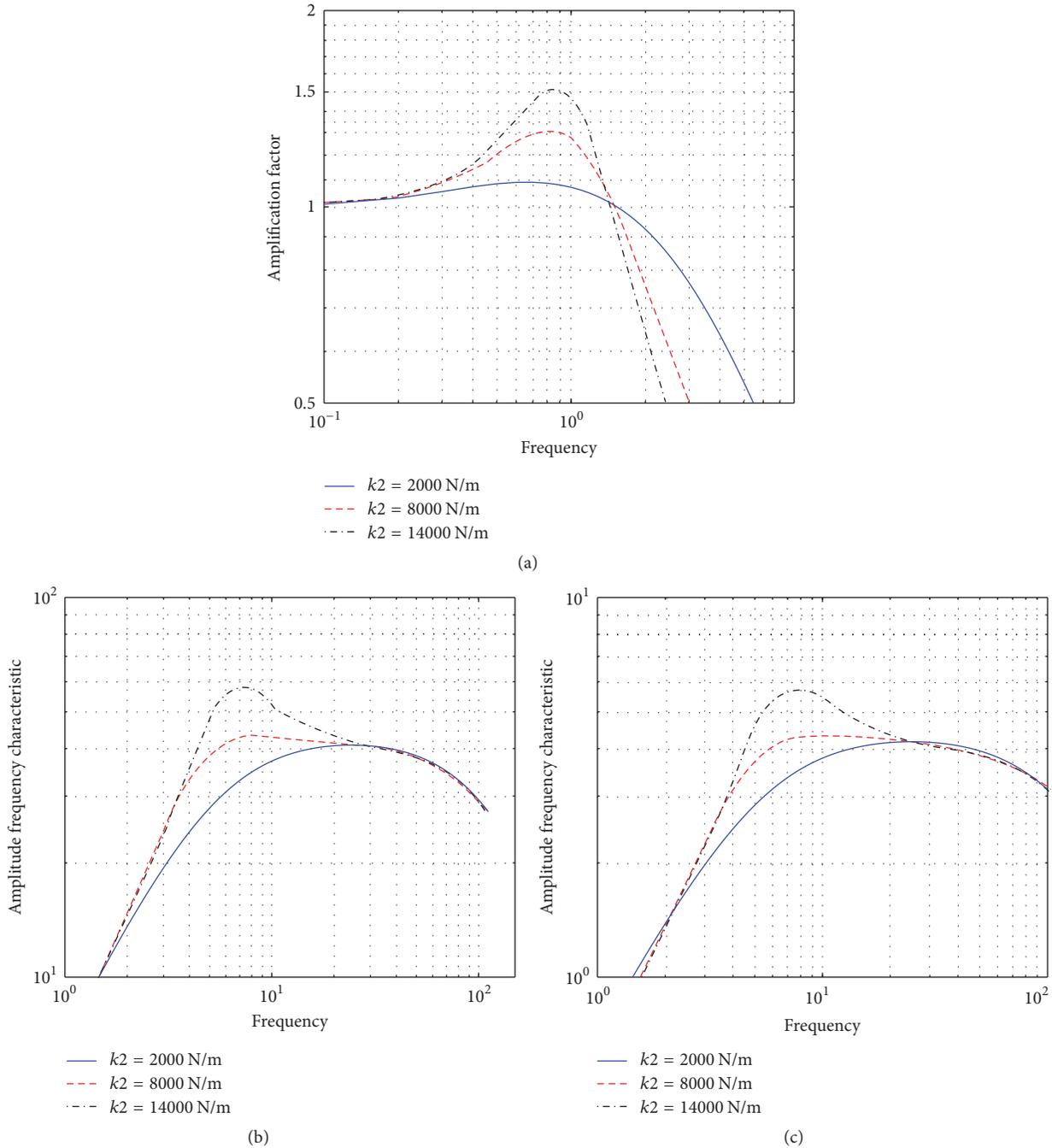


FIGURE 10: Amplitude frequency characteristics of equivalent vibration system. (a) Amplitude frequency characteristics of displacement input and displacement output. (b) Amplitude frequency characteristics of body acceleration to road surface velocity. (c) Amplitude frequency characteristics of relative dynamic load on road surface velocity.

But it may reduce shock absorption effect. Besides, the smaller stiffness may produce greater friction because the wheel deformation can make the contact of the wheel with the ground change from point contact to line contact.

4.2. Structural Stiffness in Leg State. When the rescue robot is in leg state, the buffering performance is also important. The arc leg is a compliant mechanism, and it is hard to analyze

the buffering performance because the arc leg introduces geometric nonlinearities. The pseudo-rigid-body model used in this study can simplify large-deflection analysis [34–36]. For the first walking way (the midpoint of the arc leg lands first), the leg structure which plays a role in buffering can be considered to be 1/4 circles. At this time, the 2R model is used, and the pseudo-rigid-body model is shown in Figure 11(a). The equivalent rotation pair at A_{L2} is R_{L11} , and equivalent

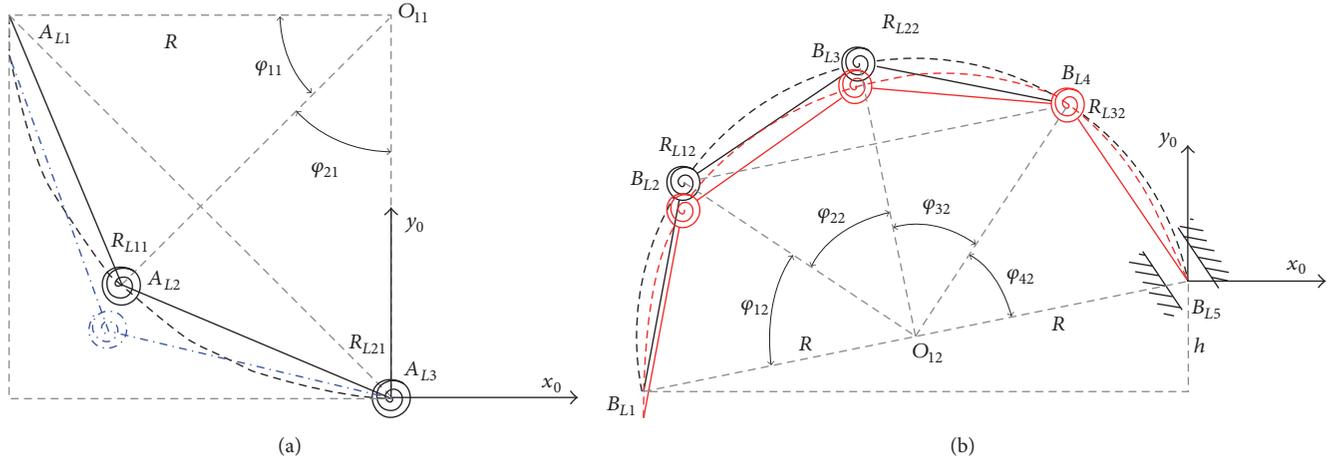


FIGURE 11: Pseudo-rigid-body model of arc leg. (a) First walking way. (b) Second walking way.

rotation pair at A_{L3} is R_{L21} . A_{L1} is the connection point between the trunk and the leg, and A_{L3} is the landing point. So the arc leg can be equivalent to 2-DOF serial mechanism, and torsion spring is used in each rotation pair. The coordinate origin of the fixed coordinate system $o_0-x_0y_0$ coincides with the landing point. The direction of the x_0 -axis is horizontal to the right, and the direction of the y_0 -axis is perpendicular to the ground and is upward.

Suppose the trunk moves only in the vertical direction. The coordinate of A_{L1} is $(-R, R-h_1)$. In triangle $A_{L1}A_{L2}A_{L3}$, the length of $A_{L1}A_{L3}$ can be shown as

$$s = \sqrt{(R-h_1)^2 + R^2}. \quad (22)$$

The rotation angles of link $A_{L1}A_{L2}$ relative to link $A_{L2}A_{L3}$ can be shown as

$$\gamma_1 = \pi - \arccos \frac{l_{11}^2 + l_{21}^2 - s^2}{2 \cdot l_{11} \cdot l_{21}} - \gamma_{10}. \quad (23)$$

The rotation angles of link $A_{L2}A_{L3}$ relative to horizontal direction can be written as

$$\gamma_2 = \arctan \frac{R-h_1}{R} - \arccos \frac{l_{21}^2 + s^2 - l_{11}^2}{2 \cdot l_{21} \cdot s} - \gamma_{20}, \quad (24)$$

where h_1 is the trunk movement distance, R is the radius of wheel-legged structure, and l_{11} and l_{21} are the length of link $A_{L1}A_{L2}$ and link $A_{L2}A_{L3}$. γ_{10} and γ_{20} are initial angles. In order to obtain the mechanical properties of buffering process, the force analysis also can be conducted. The force and torque balance equation of each link can be shown as

$$\mathbf{F}_i - \mathbf{F}_{i+1} + \mathbf{F}_{ci} + \mathbf{m}_i \mathbf{g} = 0 \quad (25)$$

$$\begin{aligned} \mathbf{M}_i - \mathbf{M}_{i+1} - \mathbf{r}_{i+1} \times \mathbf{F}_{i+1} + \mathbf{r}_i \times \mathbf{F}_i + \mathbf{r}_{ci} \times (\mathbf{F}_{ci} + \mathbf{m}_i \mathbf{g}) \\ = 0, \end{aligned} \quad (26)$$

where \mathbf{r} is the vector of the coordinate system origin to the point of force. \mathbf{F}_i and \mathbf{M}_i are the force and moment of link l_{i-1}

to l_i , and \mathbf{F}_{i+1} and \mathbf{M}_{i+1} are the force and moment of link l_{i+1} to l_i . \mathbf{F}_{ci} and \mathbf{M}_{ci} are the inertia force and inertia moment, respectively. When $i = 2$ or 3 , \mathbf{M}_i represents the torque generated by the torsion springs. When $i = 3$, \mathbf{F}_3 represents the force of the ground to the leg. To simplify the calculation process, regardless of the mass of the legs, the mass of the robot remains concentrated in the trunk. Spring damping is also not considered. At this time, the support force of the ground to the wheel-legged structure can be written as

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} -l_1 \cdot \sin \gamma_1 & l_1 \cdot \cos \gamma_1 \\ l_2 \cdot \sin \gamma_2 & l_2 \cdot \cos \gamma_2 \end{bmatrix}^{-1} \begin{bmatrix} M_1 - M_2 \\ M_2 + M_3 \end{bmatrix}, \quad (27)$$

where $M_2 = k_{11} \cdot \gamma_{11}$, $M_3 = k_{21} \cdot \gamma_{21} \cdot k_{i1}$ is the torsional stiffness coefficient.

For the second walking way, the pseudo-rigid-body model is shown in Figure 11(b). The equivalent rotation pairs at B_{L2} , B_{L3} , and B_{L4} are R_{L12} , R_{L22} , and R_{L32} , respectively. For the 1/4 circles, it is still the 2-DOF serial mechanism. In particular, considering the complexity of the environment, the end point of the leg is likely to fall into the soft soil or crevice. So B_{L4} can be seen as fixed point. The attitude of link $B_{L1}B_{L2}$ is unchanged in buffering process.

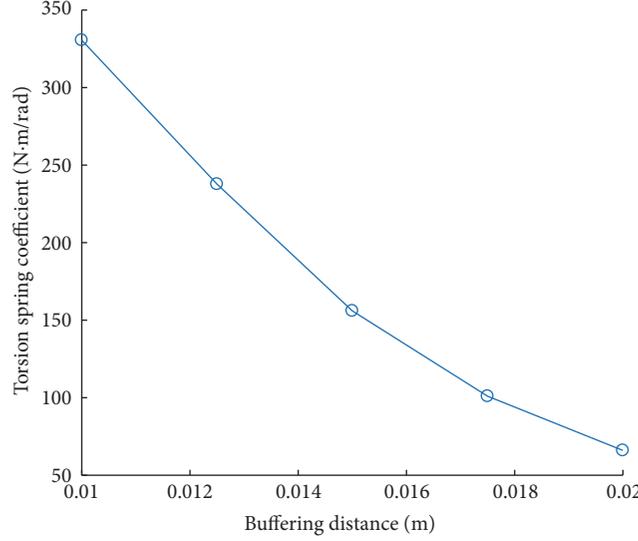
In triangle $B_{L2}B_{L3}B_{L4}$, the length of $B_{L2}B_{L4}$ can be shown as

$$s = \sqrt{T_1^2 + T_2^2}, \quad (28)$$

where $T_1 = l_4 \cdot \sin \gamma_{40} - l_1 \cdot \sin \gamma_{10} - h - h_1$; $T_2 = 2R \cdot \cos \varphi - l_4 \cdot \cos \gamma_{40} - l_1 \cdot \cos \gamma_{10}$. l_i is the length of $B_{Li}B_{L(i+1)}$. θ_{10} is the angle between link $B_{L1}B_{L2}$ and x_0 -axis, and θ_{40} is the angle between link $B_{L4}B_{L5}$ and x_0 -axis. φ is the initial inclination angle of wheel-legged structure.

At this time, the rotation angles of link $l_{(i-1)1}$ relative to link l_{i1} can be shown as

$$\gamma_2 = \pi - \arccos \frac{l_{22}^2 + l_{32}^2 - s^2}{2 \cdot l_{22} \cdot l_{32}} - \gamma_{20} \quad (29)$$

FIGURE 12: Relationship between h and k_1 .

$$\gamma_3 = \pi - \left(\arccos \frac{l_{32}^2 + s^2 - l_{22}^2}{2 \cdot l_{32} \cdot s} + \arctan \frac{T_2}{T_1} + \frac{\pi}{2} - \theta_{40} \right) - \gamma_{30} \quad (30)$$

$$\gamma_1 = \arccos \frac{l_{12}^2 + l_{22}^2 - ss^2}{2 \cdot l_{12} \cdot l_{22}} - \gamma_{10} \quad (31)$$

where

$$ss = \sqrt{(l_4 \cdot \sin \theta_{40} + l_3 \cdot \sin \theta_3 + h + h_1)^2 + (2R \cdot \cos \varphi - l_4 \cdot \cos \theta_{40} - l_3 \cdot \cos \theta_3)^2} \quad (32)$$

θ_3 is the angle between link $B_{L3}B_{L4}$ and x_0 axis.

According to (25)-(26), the support force of the ground to the leg can be obtained.

In order to analyze the mechanical properties, the stiffness coefficient of equivalent torsion springs should be further determined. When the external forces the wheel-legged structure suffers are the same and the deformation is also the same, the relationship between equivalent torsion spring stiffness coefficient and equivalent stiffness coefficient of leg can be determined according to experiment.

If the landing ground is rigid, B_{L4} can be seen as fixed point. It can be simplified as a revolute pair. So pseudo-rigid-body model has four DOF. At this time, the kinematics can be analyzed according to the same method. Because the equivalent model has four DOF and only 3 constraints can be contained, including the position and posture of end link, the angle of each link cannot be determined directly, and kinematic relationship can be written as

$$(\gamma_1 \ \gamma_2 \ \gamma_3) = f_i(\gamma_4). \quad (33)$$

Equation (33) shows that γ_1 , γ_2 , and γ_3 can be expressed with γ_4 . In order to further solve γ_4 , the force balance equation can be added according to (25)-(26). Considering kinematics

and dynamics equations synthetically, the equations can be simplified as

$$f(\gamma_4) = 0. \quad (34)$$

Then γ_4 can be obtained, and other unknown numbers also can be obtained. Suppose $R = 0.1$ m. The relationship between h and k can be shown in Figure 12. It can be known from Figure 12 that the torsional spring coefficient decreases with the increase of deformation.

The forces of the wheel-legged structure to the trunk of the robot with different maximum buffer distance can be shown in Figure 13. Figures 13(a) and 13(b) show the force along x_0 -axis and y_0 -axis with the first walking way, and Figures 13(c) and 13(d) show the force along x_0 -axis and y_0 -axis with the second walking way. It can be known from Figure 13 that the force the trunk suffers gradually increased with the increase of deformation of wheel-legged structure. At the same time, larger torsional spring coefficients correspond to greater forces. For the different walking way, the force the trunk suffers along x_0 -axis is larger with the second walking way, but the force the trunk suffers along y_0 -axis is smaller with the second walking way.

It can be known from the above analysis that smaller stiffness of the wheel-legged structure can make the change trend

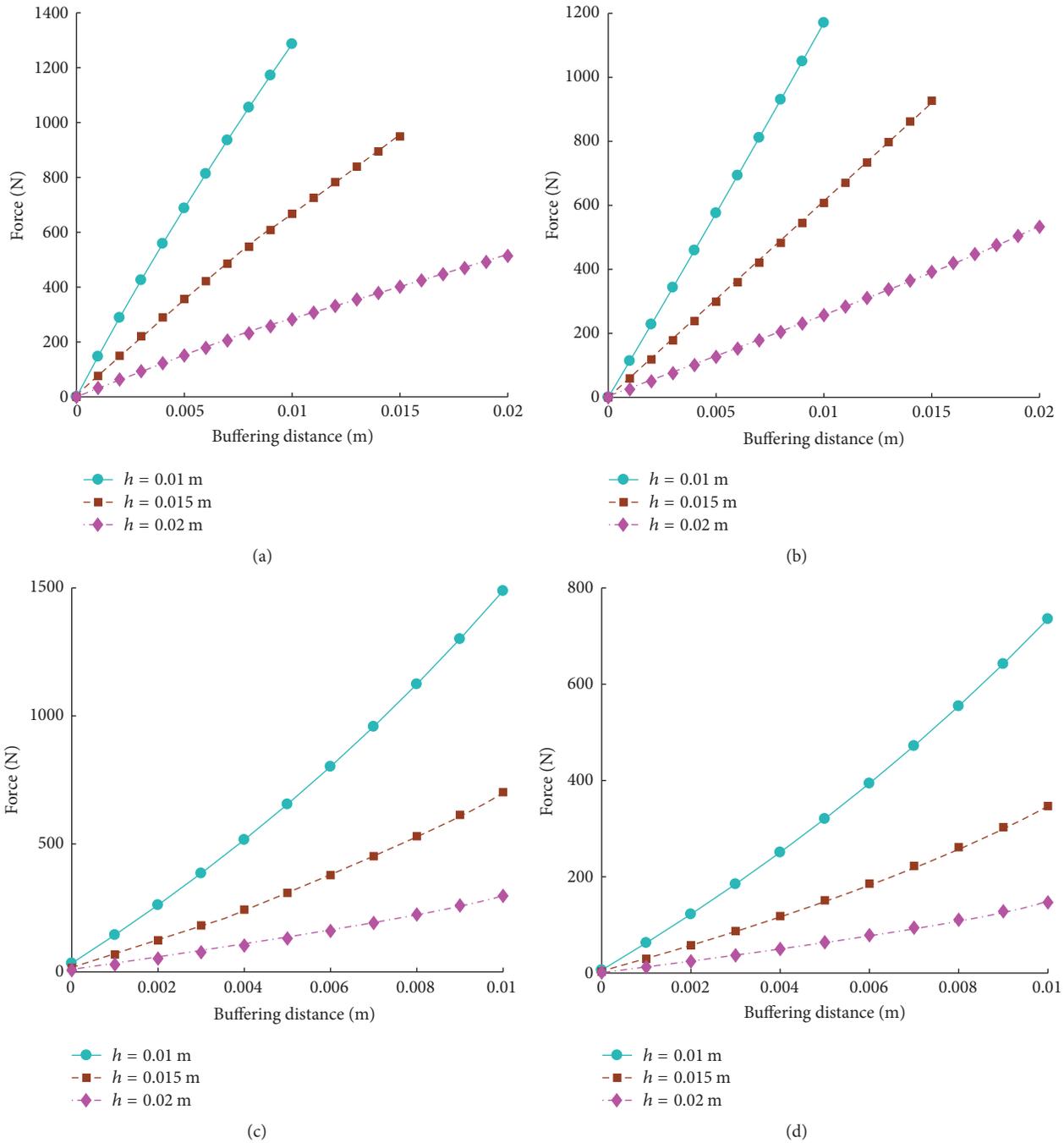


FIGURE 13: Support force of the ground to the leg. (a) Support force along x_0 -axis with the first walking way. (b) Support force along y_0 -axis with the first walking way. (c) Support force along x_0 -axis with the second walking way. (d) Support force along y_0 -axis with the second walking way.

of the force the trunk suffers more stable in leg state, but it will require the larger torque of the drive motor in the same case because the motor also needs to overcome the deformation. Besides, small stiffness will increase the deformation of the structure. Considering the effect of stiffness on the motion performance when the robot is in wheel state and leg state, the structural stiffness should be considered synthetically.

5. Experiments

The control flow chart is seen in Figure 14. When the leg state is needed for the robot, the current state should be determined. If the robot is already in the leg state, the terrain conditions should be determined, and the two different gates can be chosen. On the contrary, if the robot is in the wheel

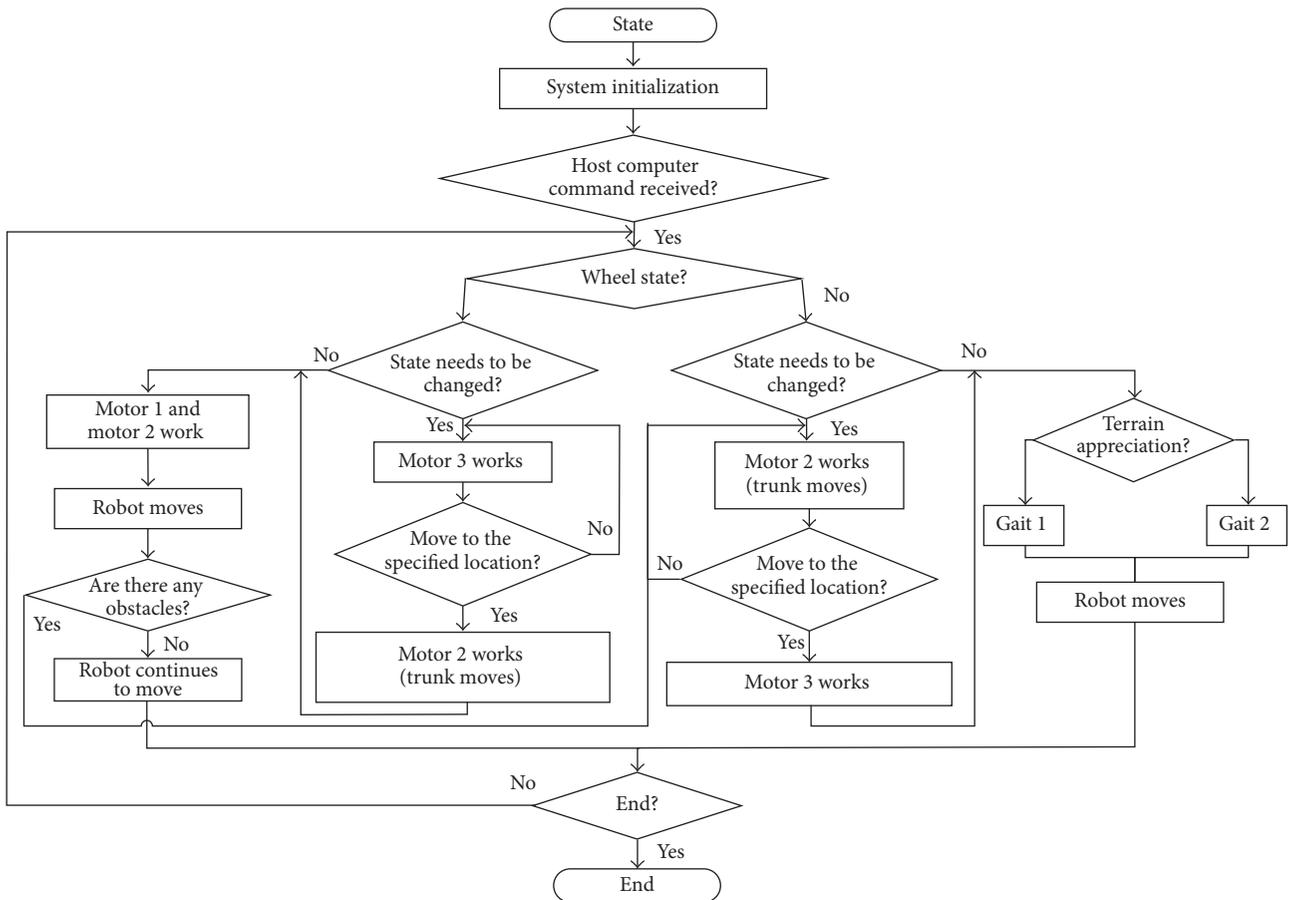


FIGURE 14: Control flow chart.

state, motor 2 works to make the trunk lifted. When the trunk is lifted to the specified position, motor 3 works to make the left half arc structure of the wheel rotate. At this time, the motion state of the robot changes. When the wheel state is needed for the robot and the robot is already in the wheel state, motor 1 and motor 2 work together, and the robot moves forward. When it encounters obstacles, the motion state can be changed from wheel state to leg state. When the wheel state is needed but the robot is in the leg state, the motor works to make the left half arc structure of the wheel rotate, and when the leg structure changes to the wheel structure, motor 2 works to make the center of rotation from the eccentric position to the wheel center. According to the above control method, the robot can realize the conversion of the leg state and wheel state.

The wheel-legged rescue robot is shown in Figure 15. The control hardware is located on the trunk of the robot. Attitude conversion process from wheel state to leg state is shown in Figure 16. First, the trunk of the robot is lifted with the driving of the motor. Then the right half arc structure rotates by the driving of motor 3 until it fits with the left half arc structure. At this time, the rescue robot is in the leg state.

The robot can achieve different leg movement states according to the different terrain environment. When the



FIGURE 15: Wheel-legged rescue robot.

robot walks on a rough road, the gait is shown in Figure 17. In the walking process, the two wheel-legged structures which are in a diagonal position move simultaneously. And there are two wheel-legged structures in the support phase, the other two wheel-legged structures in the swing phase. This kind of gait makes the robot move fast, but it cannot get over the high obstacles.

When the robot encounters a high barrier to climb, the gait is shown in Figure 18. The height of the obstacle is 15 cm. First, two wheel-legged structures which are in front of the robot are on the edge of the obstacle in turn. Then the robot climbs the obstacle with the driving of motor 1 and motor 2.

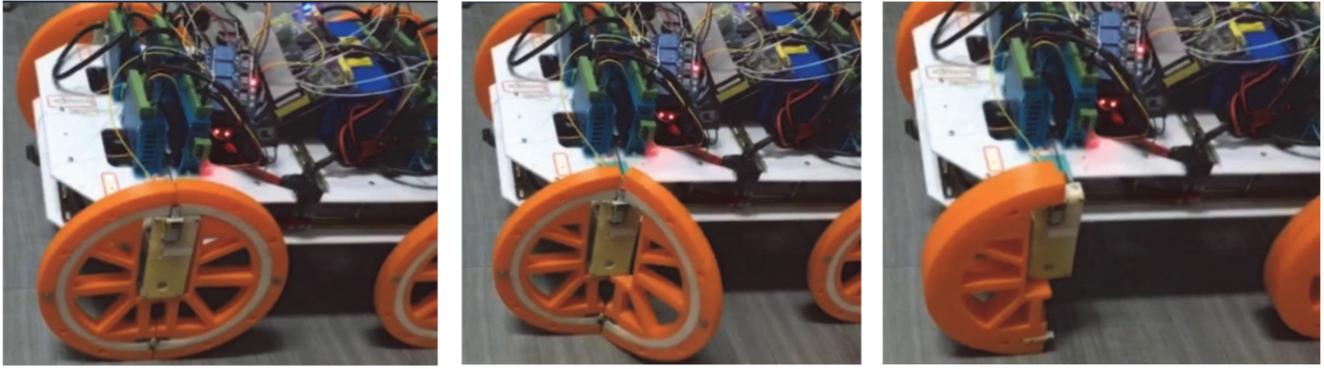
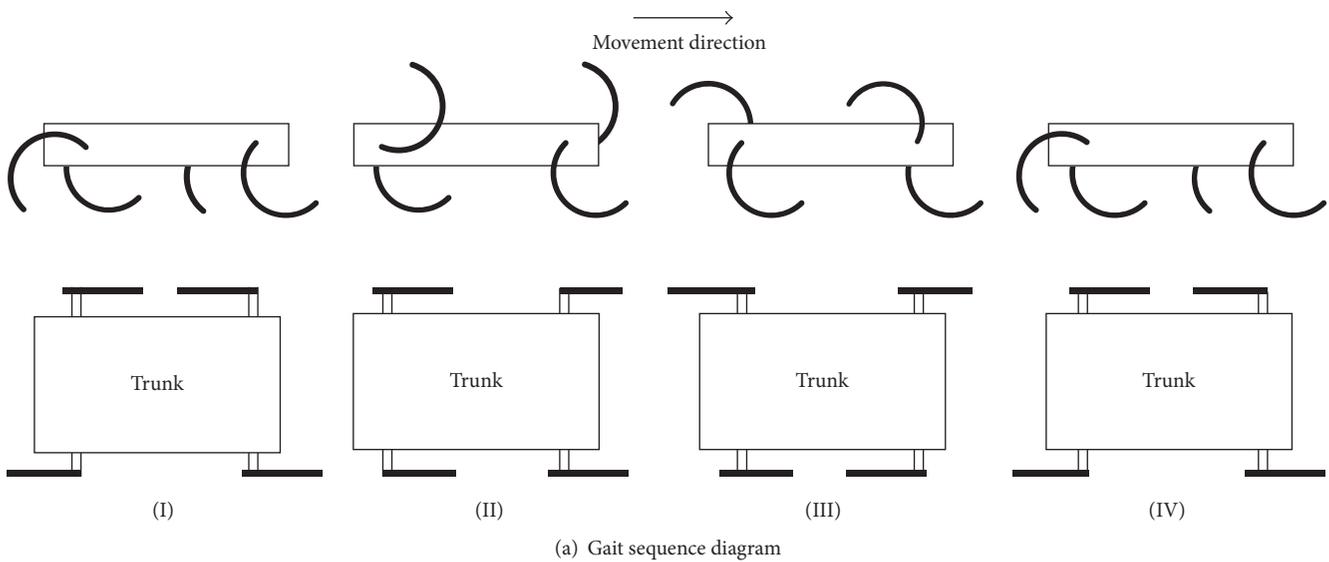
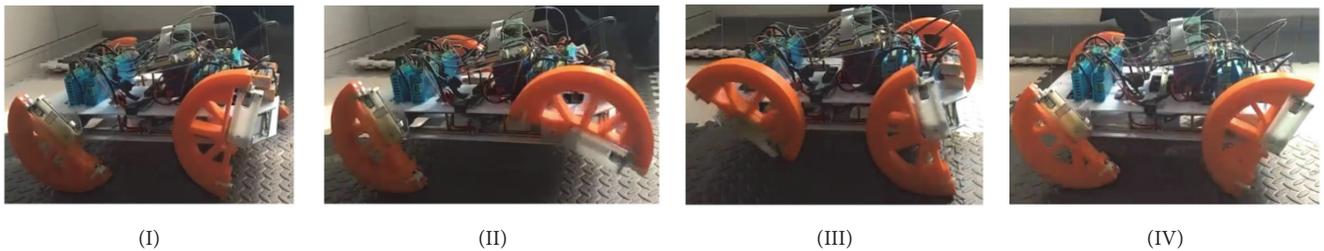


FIGURE 16: Conversion process from wheel state to leg state.



(a) Gait sequence diagram



(b) Motion diagram of the robot

FIGURE 17: Gait I of the robot (walk on rough road).

In particular, the height of the obstruction cannot be higher than the diameter of the wheel. The robot may move slowly with this kind of gait, but the obstacles it can climb can be higher than gait I.

In addition, because the wheels are elastic, the robot can reduce the vibration and impact in the rugged environment, and the robot can achieve a smooth movement. Therefore, the robot has good performance and can adapt to the complex and changeable postdisaster environment.

6. Discussion

A new wheel-legged rescue robot is designed in this paper. For the mechanical structure, the wheel of the robot can rotate 180° around the vertical axis and turn into a semicircular structure. Because it does not require any space other than the space occupied by the robot with wheel state during the state change process, it has the advantage of small occupation space. Moreover, compared with the series leg structure and

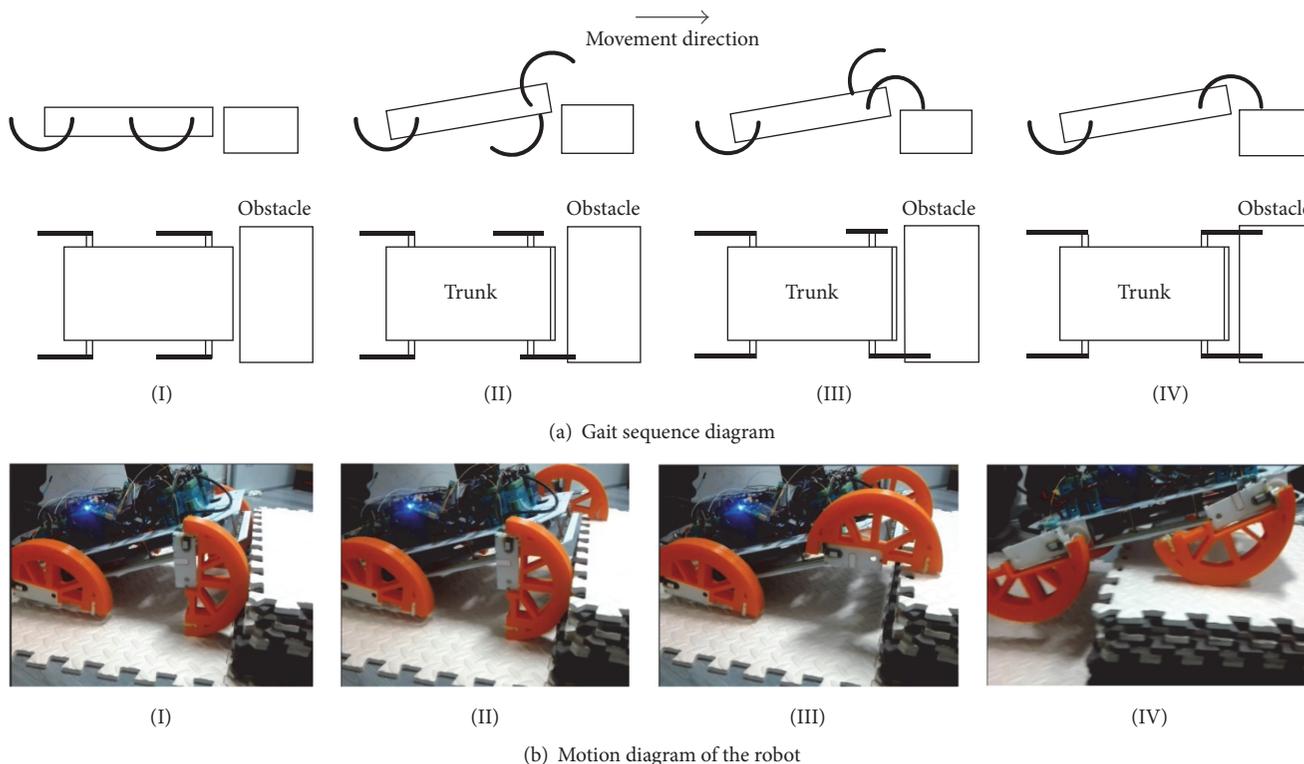


FIGURE 18: Gait 2 of the robot (overcoming a barrier).

similar parallel leg structure, the structure of semicircle arc leg is simpler and easier to control. This provides a new idea for the design of wheel-legged robot. For the movement state analysis, it provides a reference for motors selection and control. For the structural stiffness analysis, considering the wheel state and leg state share the same structure to contact with the ground and the effects of structural stiffness on the performance of robot with wheel state and leg state are different, the research results can provide a basis for wheel stiffness design. The above research results provide a useful reference for mechanism design, parameter selection, and control strategy.

However, there are still many problems for the wheel-legged rescue robot. For example, there is no waterproof structure, intelligent control algorithm is not studied, and the kinematic performance when the robot is in sea wave is not analyzed. All this needs further improvement.

7. Conclusion

This paper focuses on the design and analysis for a wheel-legged rescue robot. Considering the existing wheel-legged robot has the disadvantages of complex structure, difficult control, large space required for wheel-legged translation, and limited movement environment, a novel wheel-legged rescue robot is designed, and the wheel becomes semicircular leg by the rotation of half wheel along vertical axis. The robot can carry out a variety of motion states and can move on land

or in waves. Based on the analysis results of motion states and the effect of structural stiffness of wheel-legged structure on motion performance, the prototype of wheel-legged rescue robot is designed, and the experiments show that the robot has good performance. The present study provided a basis for designing the wheel-legged rescue robot. Further extensions to this work include the design for waterproof structure, research on intelligent control method, and the test in real postdisaster environment.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province, China (Grant no. BK20160185), project supported by the National Science Foundation for Young Scientists of China (Grant no. 51705201), the Fundamental Research Funds for the Central Universities (Grant no. JUSRP11718), the Foundation of Jiangsu Key Laboratory of Advanced Food Manufacturing Equipment and Technology (Grant no. FM-2016-08), the Jiangnan University Student's Innovation Training Project, no. 2016343Y, and the National College Student's Innovation Training Program, Project no. 201710295078Y.

References

- [1] T. Zhang, H. Shang, J. Xu, K. Liu, Y. Zhang, and J. Wang, "Application of robot technology in search and rescue in earthquake ruins," *Journal of Natural Disasters*, vol. 21, no. 5, pp. 108–112, 2012.
- [2] X. P. Dong and X. B. Wang, "Development of rescue robot technology and its application in disaster," *Journal of Disaster Prevention & Mitigation Engineering*, vol. 27, no. 1, pp. 112–117, 2007.
- [3] H. Lang, M. T. Khan, K.-K. Tan, and C. W. de Silva, "Application of visual servo control in autonomous mobile rescue robots," *International Journal of Computers, Communications & Control*, vol. 11, no. 5, pp. 685–696, 2016.
- [4] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 91–103, 2009.
- [5] A. Bodenhamer, B. Pettijohn, J. L. Pezzaniti et al., "3D vision upgrade kit for the TALON robot system," in *Proceedings of the Stereoscopic Displays and Applications XXI*, vol. 7524, pp. 195–211, SPIE - The International Society for Optical Engineering, San Jose, Calif, USA, January 2010.
- [6] K. Ito and H. Maruyama, "Semi-autonomous serially connected multi-crawler robot for search and rescue," *Advanced Robotics*, vol. 30, no. 7, pp. 489–503, 2016.
- [7] W. Wang, W. Dong, Y. Su, D. Wu, and Z. Du, "Development of search-and-rescue robots for underground coal mine applications," *Journal of Field Robotics*, vol. 31, no. 3, pp. 386–407, 2014.
- [8] N. Suzuki and Y. Yamazaki, "Basic research on the driving performance of an autonomous rescue robot with obstacles," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO '15*, pp. 982–987, Zhuhai, China, December 2015.
- [9] S. Liu and D. Sun, "Minimizing energy consumption of wheeled mobile robots via optimal motion planning," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 401–411, 2014.
- [10] Y. Pan, F. Gao, C. Qi, and X. Chai, "Human-tracking strategies for a six-legged rescue robot based on distance and view," *Chinese Journal of Mechanical Engineering*, vol. 29, no. 2, pp. 219–230, 2016.
- [11] P. Wang, J. Li, and Y. Zhang, "The nonfragile controller with covariance constraint for stable motion of quadruped search-rescue robot," *Advances in Mechanical Engineering*, vol. 2014, Article ID 917381, 10 pages, 2014.
- [12] M. Neumann, T. Predki, L. Heckes, and P. Labenda, "Snake-like, tracked, mobile robot with active flippers for urban search-and-rescue tasks," *Industrial Robot: An International Journal*, vol. 40, no. 3, pp. 246–250, 2013.
- [13] R. Haraguchi, K. Osuka, S. Makita, S. Tadokoro et al., "The development of the mobile inspection robot for rescue activity, MOIRA2," in *Proceedings of the 12th International Conference on Advanced Robotics, ICAR '05*, pp. 498–505, Seattle, WA, USA, July 2005.
- [14] Z. Zhang, D. Chen, and K. Chen, "Analysis and comparison of three leg models for bionic locust robot based on landing buffering performance," *Science China Technological Sciences*, vol. 59, no. 9, pp. 1413–1427, 2016.
- [15] B. H. Wilcox, "ATHLETE: A cargo and habitat transporter for the moon," in *Proceedings of the IEEE Aerospace Conference*, pp. 1–7, Montana, Mont, USA, March 2009.
- [16] K. Hauser, T. Bretl, J. C. Latombe, and B. Wilcox, "Motion Planning for A Six-Legged Lunar Robot," in *Workshop on the Algorithmic Foundations of Robotics*, vol. 47 of *Springer Tracts in Advanced Robotics*, pp. 301–316, Springer, Berlin, Germany, 2008.
- [17] F. B. Amar, V. Budanov, P. Bidaud et al., "A high mobility redundantly actuated mini-rover for self adaptation to terrain characteristics," in *Proceedings of the 3rd International Conference on Climbing and Walking Robots*, pp. 105–112, Madrid, Spain, 2000.
- [18] C. Grand, F. Benamar, and F. Plumet, "Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 477–495, 2010.
- [19] A. Gronowicz and J. Szrek, "Design of LegVan Wheel-Legged Robots Mechanical and Control System," in *SYROM 2009*, pp. 145–158, Springer Netherlands, Netherlands, 2010.
- [20] A. Suzumura and Y. Fujimoto, "Real-time motion generation and control systems for high wheel-legged robot mobility," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3648–3659, 2014.
- [21] X. Ding, K. Li, and K. Xu, "Dynamics and wheel's slip ratio of a wheel-legged robot in wheeled motion considering the change of height," *Chinese Journal of Mechanical Engineering*, vol. 25, no. 5, pp. 1060–1067, 2012.
- [22] X. Wang, X. Wang, H. Yu et al., "Dynamic analysis for the leg mechanism of a wheel-leg hybrid rescue robot," in *Proceedings of the 10th UKACC International Conference on Control, CONTROL '14*, pp. 504–508, Loughborough, UK, July 2014.
- [23] A. Alamdari and V. N. Krovi, "Design of articulated leg-wheel subsystem by kinetostatic optimization," *Mechanism and Machine Theory*, vol. 100, pp. 222–234, 2016.
- [24] Y. Luo, Q. Li, and Z. Liu, "Design and optimization of wheel-legged robot: rolling-wolf," *Chinese Journal of Mechanical Engineering*, vol. 27, no. 6, pp. 1133–1142, 2014.
- [25] R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguett, "Innovative design for wheeled locomotion in rough terrain," *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 151–162, 2002.
- [26] Y.-S. Kim, G.-P. Jung, H. Kim, K.-J. Cho, and C.-N. Chu, "Wheel transformer: a wheel-leg hybrid robot with passive transformable wheels," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1487–1498, 2014.
- [27] K. Tadakuma, R. Tadakuma, A. Maruyama et al., "Mechanical design of the wheel-leg hybrid mobile robot to realize a large wheel diameter," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '10*, pp. 3358–3365, Taipei, Taiwan, October 2010.
- [28] S.-C. Chen, K.-J. Huang, W.-H. Chen, S.-Y. Shen, C.-H. Li, and P.-C. Lin, "Quattroped: a leg-wheel transformable robot," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 730–742, 2014.
- [29] Z. Ma, "Design and simulation of mobile and hopping robot," in *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts, IEEE ARSO '16*, pp. 273–278, China, July 2016.
- [30] Z. Z. Guo, *Structure Optimization and Strength Analysis of The Underwater Robot*, Northeastern University, Shenyang, China, 2013.
- [31] L. S. Pontryagin, *Ordinary Differential Equations*, Addison-Wesley, New York, NY, USA, 1962.
- [32] B. W. McCormick, *Aerodynamics, Aeronautics and Flight Mechanics*, John Wiley and Sons, New York, NY, USA, 2nd edition, 1995.

- [33] G. Y. Pan, *Foundation and Application of Automotive Vibration*, Peking University Press, Beijing, China, 2013.
- [34] L. L. Howell, A. Midha, and T. W. Norton, "Evaluation of equivalent spring stiffness for use in a pseudo-rigid-body model of large-deflection compliant mechanisms," *Journal of Mechanical Design*, vol. 118, no. 1, pp. 126–131, 1996.
- [35] Z. Feng, Y. Yu, and W. Wang, "2R pseudo-rigid-body model of compliant mechanisms with compliant links to simulate tip characteristic," *Jixie Gongcheng Xuebao/Journal of Mechanical Engineering*, vol. 47, no. 1, pp. 36–42, 2011.
- [36] H. J. Su, "A load independent pseudo-rigid-body 3R model for determining large deflection of beams in compliant mechanisms," in *Proceedings of the ASME International Design Engineering Technical Conferences Computers and Information in Engineering Conference*, pp. 109–121, Brooklyn, NY, USA, 2008.

Research Article

Fuzzy Logic Controller Design for Intelligent Robots

Ching-Han Chen,¹ Chien-Chun Wang,¹ Yi Tun Wang,² and Po Tung Wang³

¹Department of Information Engineering, National Central University of Computer Science, Taoyuan, Taiwan

²Department of Information Management, National Central University of Computer Science, Taoyuan, Taiwan

³Department of Bio-Industrial Mechatronics Engineering, National Taiwan University, Taipei, Taiwan

Correspondence should be addressed to Chien-Chun Wang; patrick@g.ncu.edu.tw

Received 27 April 2017; Revised 22 July 2017; Accepted 10 August 2017; Published 27 September 2017

Academic Editor: Rafael Morales

Copyright © 2017 Ching-Han Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a fuzzy logic controller by which a robot can imitate biological behaviors such as avoiding obstacles or following walls. The proposed structure is implemented by integrating multiple ultrasonic sensors into a robot to collect data from a real-world environment. The decisions that govern the robot's behavior and autopilot navigation are driven by a field programmable gate array- (FPGA-) based fuzzy logic controller. The validity of the proposed controller was demonstrated by simulating three real-world scenarios to test the bionic behavior of a custom-built robot. The results revealed satisfactorily intelligent performance of the proposed fuzzy logic controller. The controller enabled the robot to demonstrate intelligent behaviors in complex environments. Furthermore, the robot's bionic functions satisfied its design objectives.

1. Introduction

Robot research is a popular scientific research [1–4] and is undoubtedly to enhance the quality of human life, but in the environment, the wheel structure is not suitable for use in the rugged terrain environment; in [5–7], it is very easy to study insects, whether they are moving on flat roads or walking on irregular roads.

With the increasingly widespread application of robots in today's fast-changing and diverse environment, it is essential for robots to possess autonomous movement capacity and intelligent decision-making processes, as well as behavioral control through sensory awareness of the surrounding environment to complete tasks in complex situations. In this regard, navigation and obstacle avoidance are the two crucial concerns that require attention.

Fuzzy logic [8–11] has been deemed appropriate for applications in automatic navigation of robots. This is mainly because of its capacity to process large quantities of incomplete and inaccurate input signals. Such signal processing can enable automatic navigation for robots in uncertain environments. Abundant research has been reported on the application of fuzzy theory in automatic navigation for

robots. A typical application requires the robot to carry various sensors for sensing environmental information. The outputs of the sensors serve as inputs to the fuzzy controller. Expert experience is adopted to prebuild a fuzzy rule database, which is required for the robot's subsequent behaviors. Fuzzification, fuzzy inference, and defuzzification generate decisions that control the robot's behaviors [12, 13], enabling the robot to navigate automatically.

A fuzzy logic controller can accept input from a diverse range of sensors. Ultrasonic sensors can detect the distance between a robot and obstacles [14–16]. Global positioning systems can detect the robot's current position [17–19]. With fuzzy inference, the final output enables a robot to differentiate between various environments and to perform the behaviors desired by the designer. For example, differences in wheel speeds can enable a wheeled robot to turn at an angle and roll in a new direction to avoid an obstacle. Regarding multilegged robots, the final input may be the rotational angle or forward velocity [17, 20, 21].

At present, the development of the living tools and the adaptability of the environment are far less than the evolution of the ability of organisms, regardless of how long these organisms are experiencing long-term evolution, and

whether there is a high degree of adaptability to the habitat of living space, whether it is in sensory organs, exercise patterns, learning mechanisms, organ structures, monomers, or all life structures, which is more efficient, so the use of physical characteristics will be an important future development indicator. This study focuses on the behavior of the recurrence of biological patterns for the software direction, the use of its evolution and behavior patterns, and the software processing efficiency.

2. Behavior of Intelligent Robots

2.1. Behavior of Bionic Robots. The term “bionic robots” refers to robots that mimic the body structures, functions, problem-solving behavior, and motions of living creatures, with simple mechanical structures or electronic devices. Multilegged bionic robots belong to this category; their motion patterns and tread movements resemble those of insects and spiders. Such a robot uses each leg, with its embedded multiple rotational joints, to mimic the behavioral patterns of insects. Among such robots, the six-legged type is the most common type of bionic robot.

2.2. Applying Fuzzy Controller in Obstacle Avoiding. The central nervous system contains the brain and spinal cord; the brain is responsible for the movement of the discriminant action and the spinal cord is responsible for the aggregation of sensory signals and finishing the brain cortex movement area issued by the action instructions and links the various parts of the neural network; it also contains the reflection of this part of the movement.

Robots in unknown environments need sensors to detect environmental conditions; after their sensors have measured environmental information, the robots’ controllers can generate decisions to control the robots’ behaviors. In this study, multiple ultrasonic sensors were adopted to measure distances between a robot and the environmental obstacles that surrounded it.

In this study, the neural pulse signal of the organism is used as the control structure of the imitation of the electronic circuit, and the information between the main control system and the subcontrol system is integrated. The main control system must transmit the analog signal. The auxiliary control system only needs to transmit the sensed analog signal to the main system and receives the motion instruction given by the main system, and the auxiliary functions of the main system and subsystems are complementary to each other.

Multiple sets of ultrasonic distance measurements were used as input to develop an intelligent navigation system. With this distance information, the fuzzy logic controller enabled the robot to safely complete tasks in an unknown environment. An individual ultrasonic sensor is shown in Figure 1(a). A photograph of the six-legged robot with an operative sensor array is shown in Figure 1(b). The positions of the distance sensing ultrasonic sensors are shown in Figure 1(c): right (S1), front (S2), left (S3), right front (S4), and left front (S5).

The control system of this study combines the main controller, subcontroller, organization, and computer software.

Six-foot robot body uses 18 servo motors, six MCU sub-controllers, UART message transmission, and the main controller of the news, and the design of wireless monitoring equipment, the use of RF module for two-way data transmission, and timely feedback related information and PC are used to analyze the action. You can adjust the six-foot robot in manual mode or automatic mode, which can be used flexibly.

The master/slave controller system can communicate with each other through the Bluetooth system and the monitoring system, and the monitoring system can manually operate the six-legged robot manually.

Figure 2 shows foot movement angle relative relationship [7]; M2 will be disturbed by the M2-M3 mechanism and the M2 motor mechanism; it must be specified within $\pm 40^\circ$, to face the adjacent body operation collision damage.

M3 mechanism set contact with the ground for the vertical ground 90° , so the M2 and M3 must be the opposite of the servo motor angle; related mechanical joints are as follows:

$$M2 - M3 = 57 \text{ mm.}$$

$$M2\text{-Machine bottom} = 42.7 \text{ mm.}$$

$$M3\text{-the ground} = 108 \text{ mm.}$$

$$L1 = 26.62 \text{ mm.}$$

$$H3 = 42.7 \text{ mm.}$$

$$H2 = \sin \theta * (M2 - M3).$$

$$H4 = H2 + L3 - H3 = \sin \theta * 57 \text{ mm} + 108 \text{ mm} - 42.7 \text{ mm.}$$

Using the above formula, the design of Figure 3 shows the movement angle and displacement of the map were from 0° , 10° , 20° , 30° , 40° , -10° , -20° , -30° , and -40° , for foot control design.

And then the corresponding H2 and H4 movement angle and displacement volume are as shown in Table 1.

Each foot institution is limited to positive and negative 60 degrees as a range of services, such as the six-foot body map shown in Figure 4, to avoid the impact of mutual action agencies and to prevent damage to the organization.

2.2.1. Design of Obstacle Avoidance. Figure 5 shows obstacle avoidance is a priority for a moving robot that must avoid obstacles and prevent collisions. We propose an efficient real-time avoidance system by which robots can avoid obstacles while moving as directed by automatic navigation. The distance information collected from sensors 1, 2, and 3 is used for the ultrasonic sensing system. Using the information about the robot’s surroundings collected from these sensors as input, the proposed fuzzy logic controller generates instructions by which the robot can avoid obstacles.

Table 3 shows the complete obstacle avoidance procedure, in which the robot receives the ultrasonically sensed distances, from front, right, and left sensors, defined as d_f , d_r , d_l , respectively. These three distance variables are fed as input into the obstacle avoidance fuzzy controller. The motor control board calculates the velocity of the robot (V_e) and directional angles to be modified (θ_e). On receiving

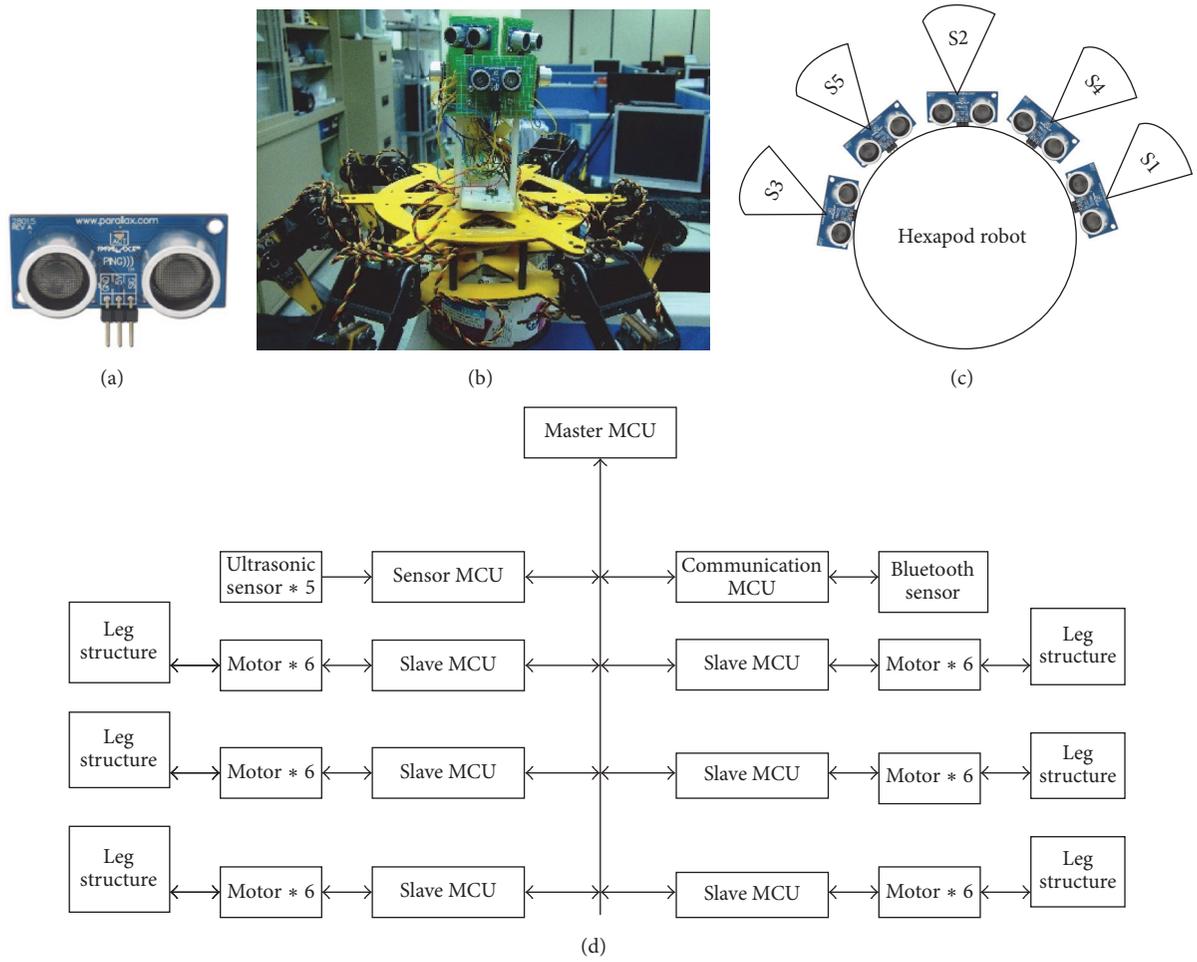


FIGURE 1: Ultrasonic sensing system. (a) Ultrasonic sensor (sourced from Parallax, Inc.). (b) Appearance of the MIAT six-legged robot. (c) Disposition of ultrasonic sensors. (d) Six-foot robot control architecture.

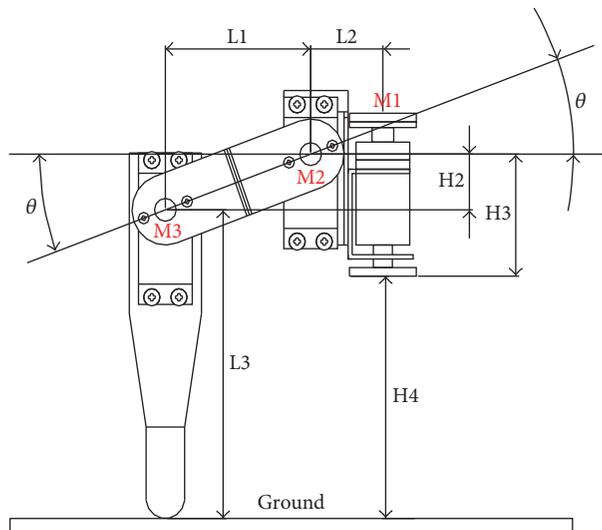


FIGURE 2: Foot movement angle relative relation diagram.

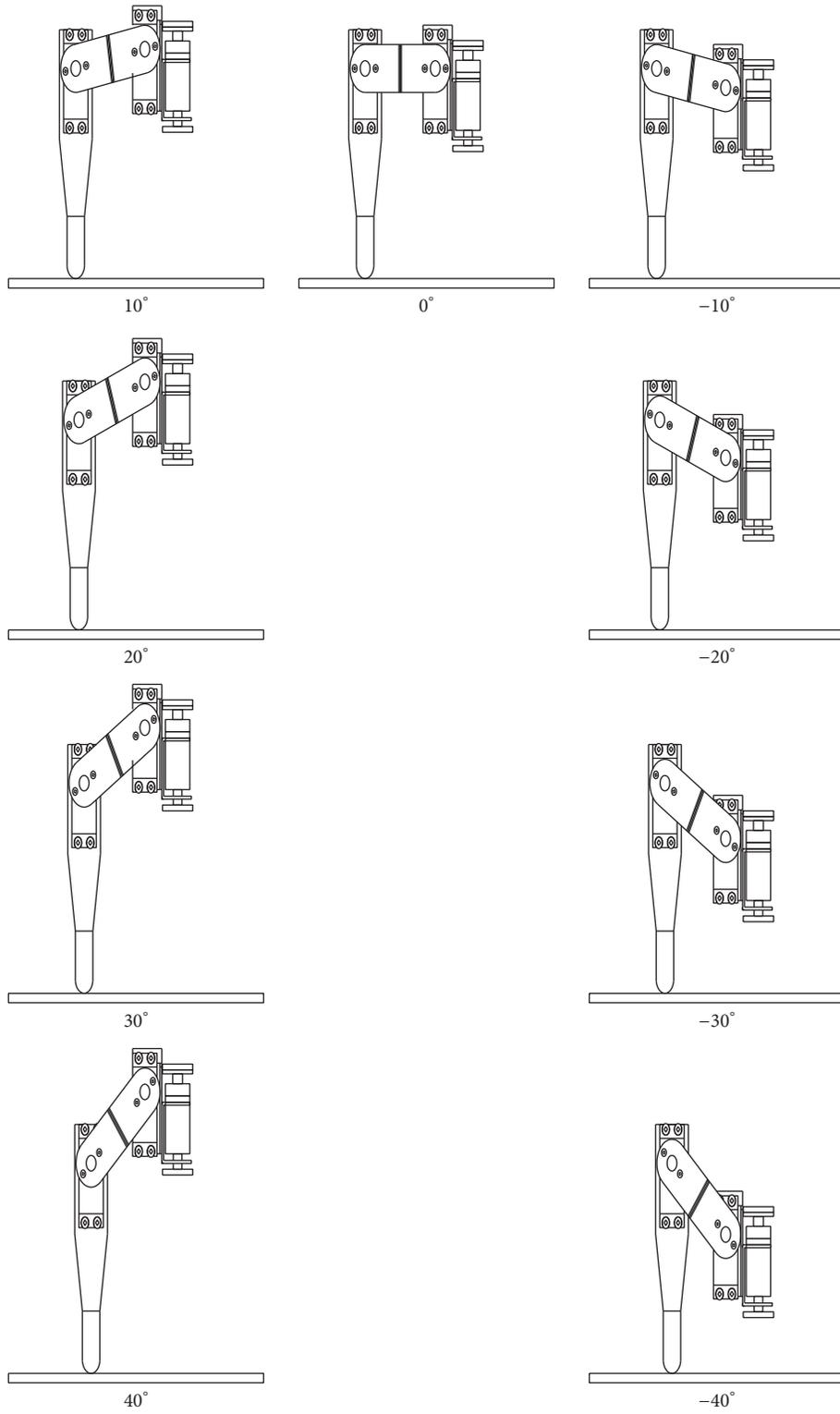


FIGURE 3: Foot movement angle (plus or minus 40 degrees).

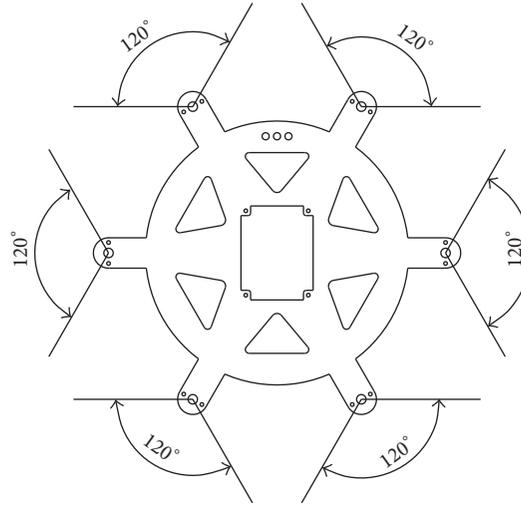


FIGURE 4: Six-foot main body.

TABLE 1: Foot movement data.

θ	H2 (mm)	H4 (mm)
-40°	-36.64	28.66
-35°	-32.69	32.61
-30°	-28.50	36.80
-25°	-24.09	41.21
-20°	-19.50	45.80
-15°	-14.75	50.55
-10°	-9.90	55.40
-5°	-4.97	60.33
0°	0	65.30
5°	4.97	70.27
10°	9.90	75.20
15°	14.75	80.05
20°	19.50	84.80
25°	24.09	89.39
30°	28.50	93.80
35°	32.69	97.99
40°	36.64	101.94

TABLE 2: Linguistic terms of the input.

Distance	Linguistic term
0 cm~40 cm	Close (CC)
20 cm~60 cm	Near (NC)
40 cm~80 cm	Far (FC)

TABLE 3: Linguistic terms of the output.

Speed	Steering
No movement (NM)	Forward (FW)
Slow movement (SM)	Turn right (RT)
Medium movement (MM)	Turn left (LT)
Fast movement (BM)	
Very fast movement (HM)	

the output from the fuzzy logic controller, the robot can change its current velocity and navigational directions to avoid any obstacle. Each input to the fuzzy controller is assigned the same membership function (MF), as illustrated in Figure 6(a). The output is shown in Figures 6(b) and 6(c). All the MFs are triangular. The linguistic terms of the input and output are shown in Tables 2 and 3.

For the real-world robot, the widely known max-min operation was used in the fuzzy inference, and the center-of-gravity method was adopted in fuzzification. Regarding the fuzzy rule database, the researchers' experience and knowledge of partially successful attempts were utilized to build relevant rules, as shown in Table 4.

2.2.2. Wall Following. The robot can follow walls, thus enabling it to walk along the boundaries of the testing environment. The operational definition is "motion in a direction parallel to the nearest wall at a consistent distance from that wall." The procedure for wall following is illustrated in Figure 7.

The right and left ultrasonic sensors (S1 and S2 in Figure 1) are used to detect the side distances of the robot, defined as d_r and d_l , respectively. The sensing system first judges which side is closer to the robot and chooses the distance to the nearest side as the baseline, enabling the robot to move along the nearest wall. The robot evaluates its current location against that in the previous cycle (farther from or nearer to the side wall). If the distance in the current cycle is longer than that in the previous cycle, it signifies that the robot has moved farther away from the wall. Therefore, the robot must turn toward the wall. Conversely, a shorter distance in this cycle than that in the previous cycle means that the robot has moved closer to the wall. In such a case, the robot must turn away from

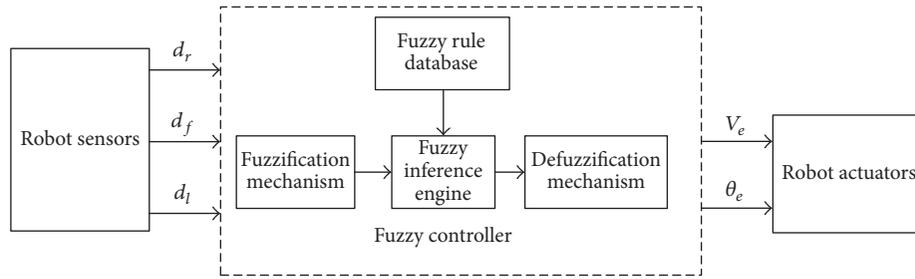
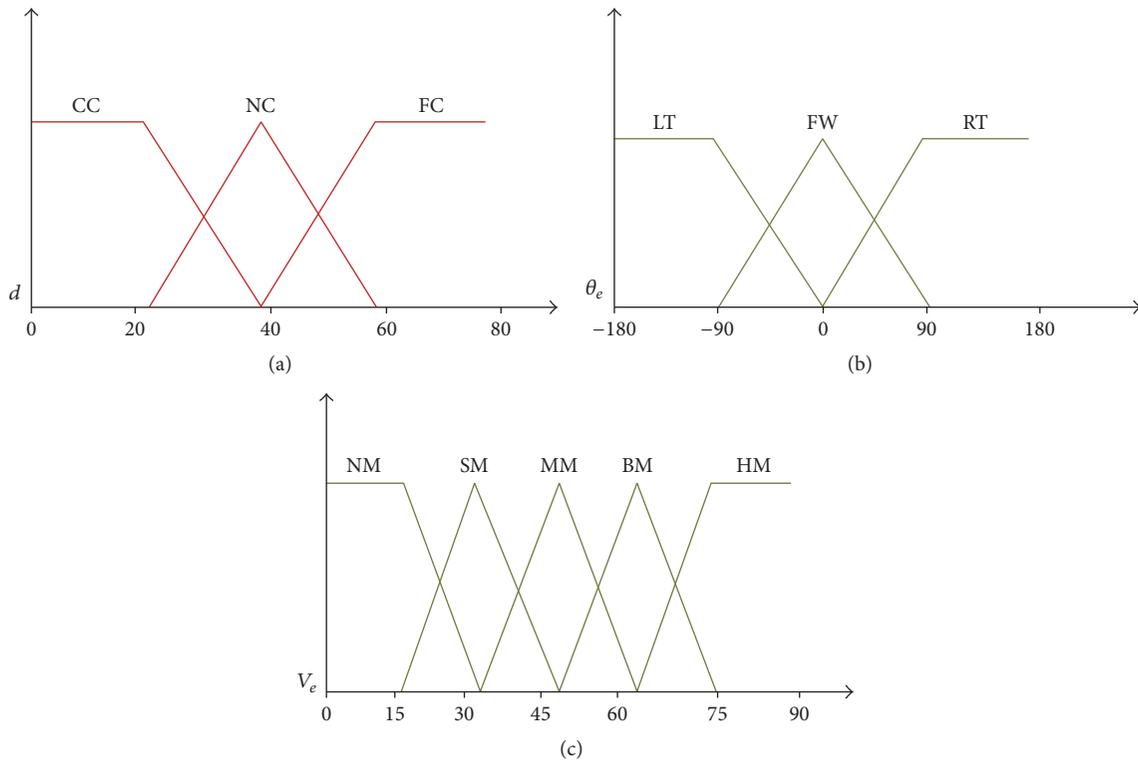


FIGURE 5: Obstacle avoidance process.

FIGURE 6: MF of obstacle avoidance fuzzy controller. (a) MFs of inputs d_f , d_r , and d_l . (b) MF of output θ_e ; (c) MF of output V_e .

the wall. The difference between the wall distances in the two cycles (d_e) is then taken as one of the inputs to the fuzzy logic controller. The other input is the velocity of the robot (V_c). A relatively fast velocity means that the robot has gained considerable speed by moving away from or closer to the wall, thus requiring more angular modifications, and vice versa.

Figure 8 illustrates the wall following behavior. In Case A, the robot first detects a shorter distance from the left wall and moves along it. Because the robot's motion is parallel to the wall, the robot-wall distance differences in the two cycles should be zero, thus requiring no angular modification. In Case B, the robot begins to deviate from the left wall; thus, the robot-wall distance difference between the two cycles should increase. Using the distance difference and the current velocity as inputs to the fuzzy logic controller, a turning angle θ_e toward the wall can be obtained, enabling the robot to

turn toward the wall and return to correct the wall following behavior.

The MFs of the wall following input are illustrated in Figures 9(a) and 9(b). The MF of the output is shown in Figure 9(c). All the MFs are obtained as triangles. The linguistic terms of the input and the output are shown in Table 5. The widely known max-min operation is used in fuzzy inference and the center-of-gravity method is adopted for defuzzification. The wall following fuzzy controller rules are shown in Table 6.

3. Experiments and Results

In this study, we designed a total of 18 servo motors of six feet as the induction unit of the neural current, combined with the fuzzy ultrasonic controller, six angles of the design, through

TABLE 4: Obstacle avoidance fuzzy controller rules.

Rule(s)	Input(s)			Output(s)	
	C: close; N: near; F: far			Speed	Steering
	S1	S2	S3		
(1)	C	C	C	NM	LT
(2)	C	C	N	NM	LT
(3)	C	C	F	NM	LT
(4)	C	N	C	NM	FW
(5)	C	N	N	NM	LT
(6)	C	N	F	NM	LT
(7)	C	F	C	NM	FW
(8)	C	F	N	NM	FW
(9)	C	F	F	BM	FW
(10)	N	C	C	NM	RT
(11)	N	C	N	SM	RT
(12)	N	C	F	MM	LT
(13)	N	N	C	MM	FW
(14)	N	N	N	BM	FW
(15)	N	N	F	HM	LT
(16)	N	F	C	HM	FW
(17)	N	F	N	BM	FW
(18)	N	F	F	BM	FW
(19)	F	C	C	NM	RT
(20)	F	C	N	NM	RT
(21)	F	C	F	HM	RT
(22)	F	N	C	SM	RT
(23)	F	N	N	MM	FW
(24)	F	N	F	MM	FW
(25)	F	F	C	SM	FW
(26)	F	F	N	BM	FW
(27)	F	F	F	HM	FW

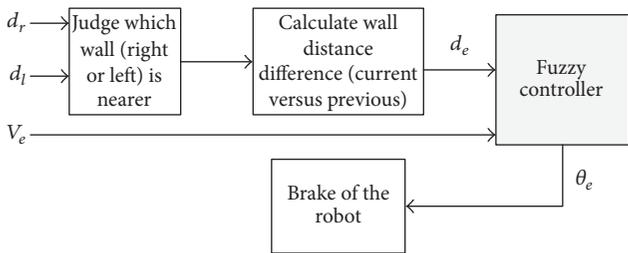


FIGURE 7: Wall following procedure.

the three-foot system for the neural current sensing control. The judgment of the stability and the identification of the obstacle in front of the ultrasonic wave are described below. The following describes the control system. The walking state of the six-legged robot is the ratio of the height to the time of the six-legged robot shown in Figure 10, and the center of the body is highly displaced.

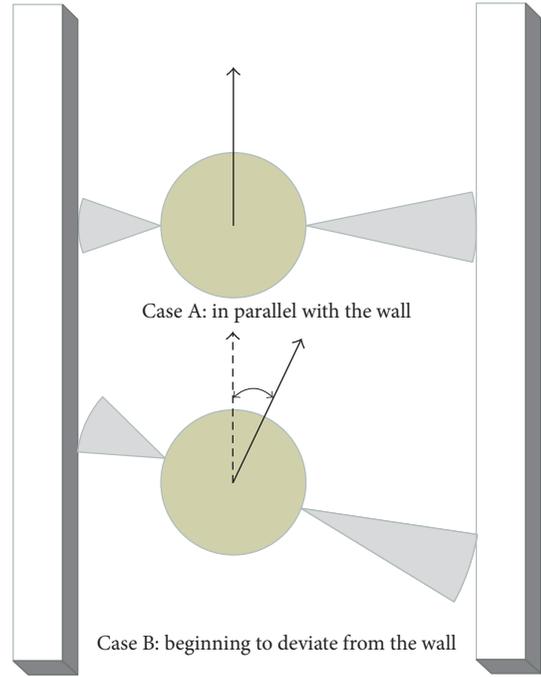


FIGURE 8: Illustration of wall following.

TABLE 5: Linguistic terms of the input and output.

Difference of distance	Velocity	Angle
Small	Slow	Small
Medium	Medium	Medium
Large	Fast	Large

TABLE 6: Wall following fuzzy controller rules.

Rule(s)	If	Then	
	d_e	V_c	θ_e
(1)	Small	Slow	Small
(2)	Small	Medium	Small
(3)	Small	Fast	Medium
(4)	Medium	Slow	Medium
(5)	Medium	Medium	Medium
(6)	Medium	Fast	Large
(7)	Large	Slow	Large
(8)	Large	Medium	Large
(9)	Large	Fast	Large

Figure 11 shows the six-legged robotic gait diagram showing the footsteps of the left and right sides of the six-legged robot, and the footsteps of the six-foot robots are located on the ground.

3.1. The Experiment. After the robot's bionic behavior controlling chip was designed and implemented, the chip was

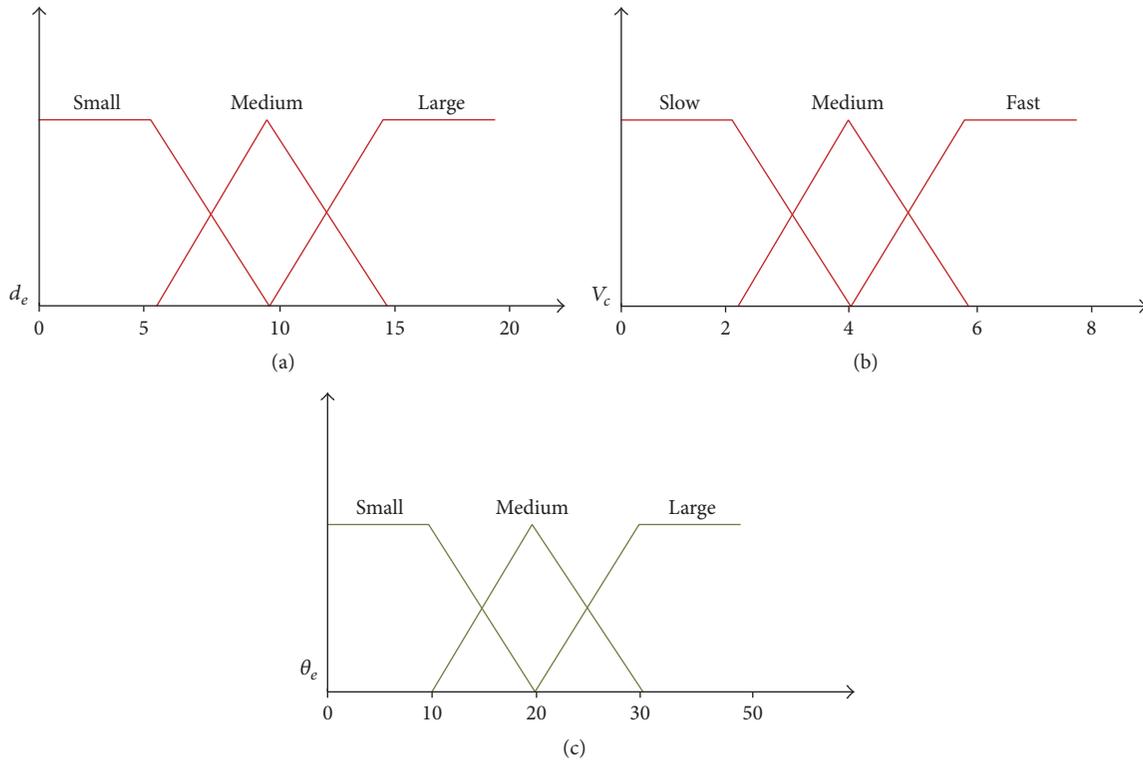


FIGURE 9: MF of wall following fuzzy controller. (a) MFs of input d_e . (b) MF of input V_c . (c) MF of output θ_e .

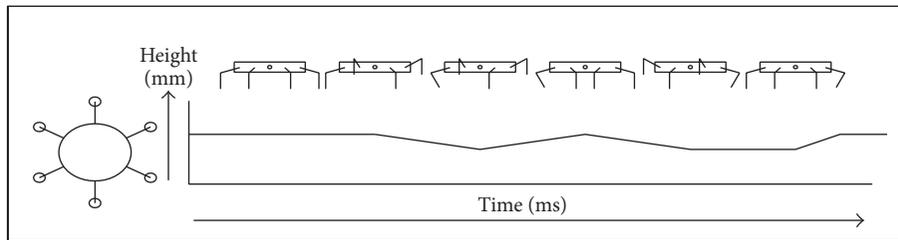


FIGURE 10: The height state of the six-legged robot when moving.

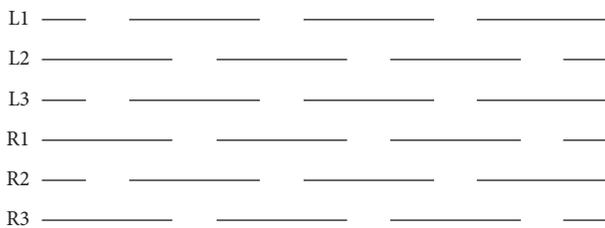


FIGURE 11: Six-foot robots gait.

installed in a real robot for function verification and performance analysis. The hardware comprised a field programmable gate array board, motor control board, and a six-legged robot, as shown in Figure 1(b).

3.2. Experiment of the Robot's Bionic Controlling Behavior

Scenario 1: Obstacle Avoidance. The purpose of this experiment was to examine whether the proposed navigation system could successfully lead the robot to avoid obstacles. The path was 1 m wide, and the width of the obstacle (a rectangular solid) was 20 cm. As shown in Figure 12, the robot first moved upward following instructions from the developed fuzzy database. While moving, the robot detected objects on both its sides at Point A but continued to move forward because the distance ahead was still considerable. The robot detected an object close ahead and a wall blocking its right side when it reached Point B. Immediately, the fuzzy logic controller responded with a left turn. At Point C, obstacles were present at the front and on the left side. The obstacle on the right side was farther away; therefore,

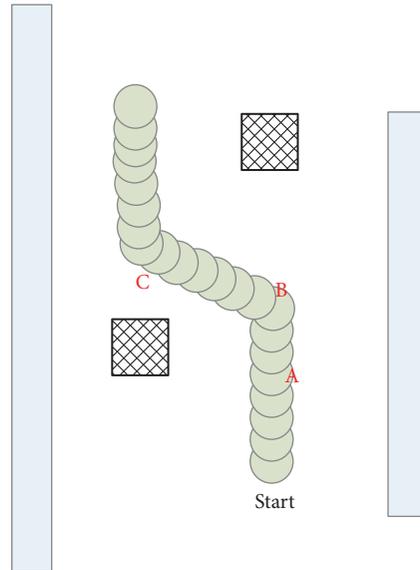


FIGURE 12: Moving path of Scenario 1.

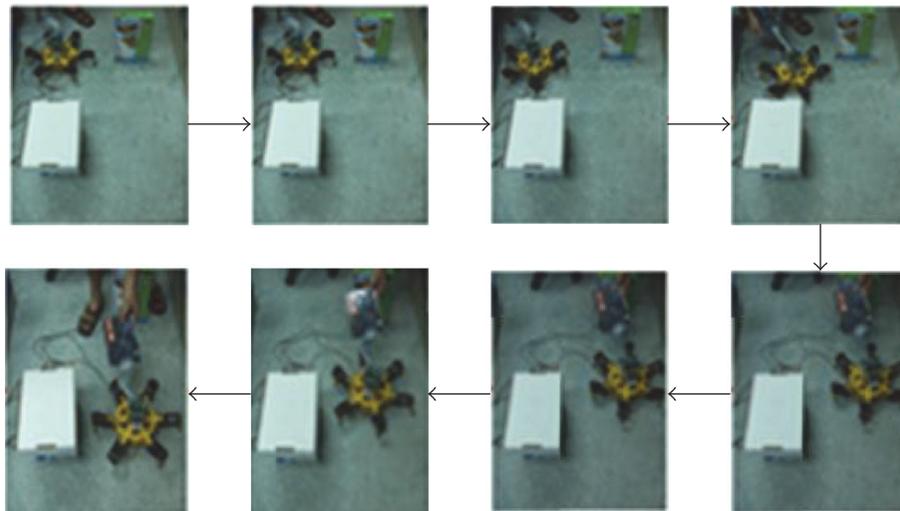


FIGURE 13: Real experiment of Scenario 1.

the fuzzy logic controller responded with a right turn. With no further obstacles lying ahead, the robot continued moving forward. The real experiment is illustrated in Figure 13.

Scenario 2: Dead End with Three Surrounding Walls. The purpose of this experiment was to examine whether the robot could make its way out of a dead end. The dead end was 50 cm wide, as shown in Figure 14. The robot moved forward, following instructions from the developed fuzzy database. As it moved, it detected objects on both sides at Point A. However, it continued moving forward because there was a considerable expanse of empty space before it. At Point B, the robot then detected an object that obstructed its forward path and a wall blocking its right side. Immediately, the fuzzy logic controller responded with a left turn. At Point C, the obstacles appeared at the front and on the left side. The obstacle to the

left side was more distant; thus the fuzzy controller ordered a left turn. At Point D, the robot turned left because no obstacles were found on the left. After the robot had turned, it found no further obstacles lying ahead and continued to move forward. The real experiment is shown in Figure 15.

Scenario 3: Wall Following and Obstacle Avoidance. This experiment tested whether the proposed navigation system correctly integrated wall following, obstacle avoidance, and target tracing. As shown in Figure 16, the wall was 60 cm long, and the obstacle (a rectangular solid) was 20 cm wide. At Point A, the robot calculated that its conditions called for a wall following behavior. At Point B, the robot deviated gradually from the wall, and the fuzzy controller modified the robot's direction away from the wall. At Point C, the robot detected obstacles at its front and on its left side. In the

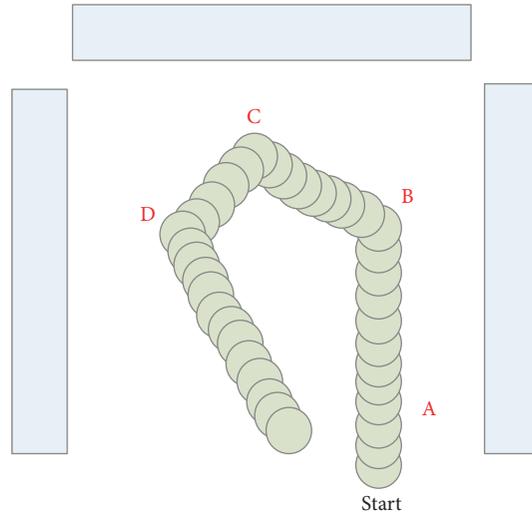


FIGURE 14: Moving path of Scenario 2.

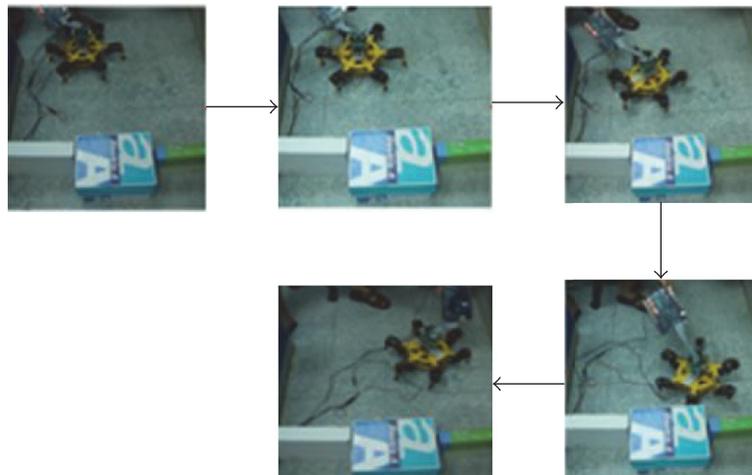


FIGURE 15: Real experiment of Scenario 2.

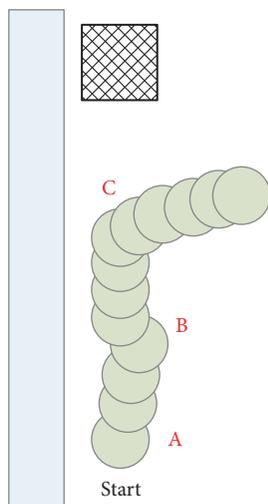


FIGURE 16: Moving path of Scenario 3.

proposed integrated system, obstacle avoidance has higher priority than wall following does; the fuzzy controller ordered a right turn, and the robot turned right and performed obstacle avoidance. The real experiment is shown in Figure 17.

3.3. Discussion. In the current study, a real robot was tested; the accuracy levels of its intelligent behavioral functions were verified. Thus, three scenarios were designed to examine the robot's bionic behaviors. The results reveal that the proposed fuzzy logic controller enabled the robot to perform in complex environments and to demonstrate intelligent behaviors. The robot was designed for bionic functions; its performance demonstrated those bionic functions.

4. Conclusions

The current study, with fuzzy control as its core, proposes control methods for intelligent behaviors such as obstacle

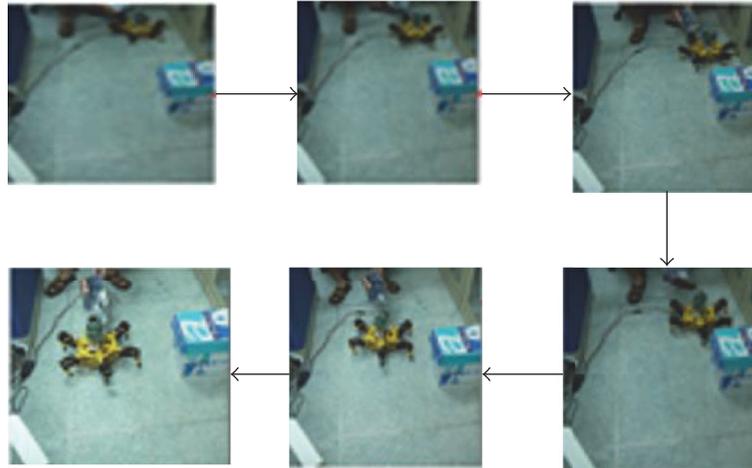


FIGURE 17: Real experiment of Scenario 3.

avoidance, wall following, and attack. The system integrates three control methods within an inclusive structure that can be used for further development of bionic behavior controllers for intelligent robots.

In this study, the controller platform is established to determine whether the surrounding environment is suitable for walking through the most basic bio-current signal and discrete event control through the most primitive and consistent simulation of the organism. In the robot's reflex movement which is directly used for environmental interaction, this method can let the six-legged robot know the front path of obstacles and quickly take action to avoid unnecessary collision; the future can use GPS, gyroscope, pressure sensors, temperature and humidity sensors, and infrared sensors, increasing fusion sensing in the system, so that, overall, it is more perfect.

In the system of signal transmission, we use the simplest 8-bit transmission to significantly reduce the chip computing, reduce system load, as far as possible in a short time to resolve and control the system, so that the signal from the main system to the subcontroller can be immediately transferred. This method allows the system time to deal with not only other major messages to judge, but also the least information for the most effective signal processing, and to be closer to the simulation of biological movement structure.

In the study of the structure of the robot, the behavior can be smoother, real-time to achieve effective steering behavior, dodge behavior; these have a high degree of evolution, functional verification, and system analysis; we have established three situations, dodge obstacles, to avoid the dead end and walking in a wall of space; the results show that the proposed fuzzy logic controller is able to successfully guide the robot behavior with the autonomous model.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. S. Ayas, I. H. Altas, and E. Sahin, "An optimized fuzzy logic controller for a parallel mechanism rehabilitation robot," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '15)*, pp. 1–6, IEEE, August 2015.
- [2] Y. C. Chou, K. J. Huang, W. S. Yu, and P. C. Lin, "Model-based development of leaping in a hexapod robot," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 40–54, 2015.
- [3] Y. Zhai, P. Gao, Y. Sun et al., "Gait planning for a multi-motion mode wheel-legged hexapod robot," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '16)*, pp. 449–454, December 2016.
- [4] L. Zhang, D. Li, F. Yang, and C. Liu, "Development and attitude control of a Hexapod bionic-Robot," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO 16)*, pp. 77–82, December 2016.
- [5] E. Passos and L. Almeida, "Simulations results of an insect robot," in *Proceedings of the International Conference on Neural Networks (IJCNN '93)*, vol. 2, pp. 1837–1840, Nagoya, Japan, 1993.
- [6] L.-C. Liao, K.-Y. Huang, and B.-C. Tseng, "Design and implementation of a quadruped robot insect," in *Proceedings of the 12th IEEE International Conference on Mechatronics and Automation (ICMA '15)*, pp. 269–273, August 2015.
- [7] W. A. Lewinger, H. M. Reekie, and B. Webb, "A hexapod robot modeled on the stick insect, *Carausius morosus*," in *Proceedings of the 15th International Conference on Advanced Robotics (ICAR '11)*, pp. 541–548, June 2011.
- [8] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, NJ, USA, 1995.
- [9] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [10] M. V. Bobyr, S. A. Kulabukhov, and N. A. Milostnaya, "Fuzzy control system of robot angular attitude," in *Proceedings of the 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM '16)*, pp. 1–6, Chelyabinsk, Russia, May 2016.
- [11] J. J. Buckley, "Theory of the fuzzy controller: an introduction," *Fuzzy Sets and Systems*, vol. 51, no. 3, pp. 249–258, 1992.

- [12] H. Kolbari, S. Sadeghnejad, A. T. Parizi, S. Rashidi, and J. H. Baltes, "Extended fuzzy logic controller for uncertain teleoperation system," in *Proceedings of the 4th International Conference on Robotics and Mechatronics (ICROM '16)*, pp. 78–83, Tehran, Iran, October 2016.
- [13] R. N. Bandara and S. Gaspe, "Fuzzy logic controller design for an Unmanned Aerial Vehicle," in *Proceedings of the IEEE International Conference on Information and Automation for Sustainability (ICIAfS '16)*, pp. 1–5, Galle, Sri Lanka, December 2016.
- [14] A. Ramirez-Serrano and M. Boumedine, "Ultrasonic sensing and fuzzy logic control for navigation in unknown static environments," in *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT '96)*, pp. 54–59, IEEE, Kaiserslautern, Germany, 1996.
- [15] S. Thongchai and K. Kawamura, "Application of fuzzy control to a sonar-based obstacle avoidance mobile robot," in *Proceedings of the IEEE International Conference on Control Applications*, pp. 425–430, IEEE, 2000.
- [16] C. Mohan, B. Surgenor, L. Monteiro, and V. Nazari, "Tracking with a fuzzy logic controller as applied to a pneumatic robot for polishing," in *Proceedings of the IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE '17)*, pp. 1–6, Windsor, Canada, April 2017.
- [17] S. Zein-Sabatto, A. Sekmen, and P. Koseyaporn, "Fuzzy behaviors for control of mobile robots," *Journal of Systemics, Cybernetics and Informatics*, vol. 1, pp. 68–74, 2003.
- [18] Y. Hada and K. Takase, "Multiple mobile robot navigation using the indoor global positioning system (iGPS)," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium*, vol. 2, pp. 1005–1010, 2001, Cat. No.01CH37180.
- [19] F. Ibrahim, A. A. Abouelsoud, and A. M. R. F. Elbab, "Time-varying fuzzy logic controller of skid-steering mobile robot based on Lyapunov stability," in *Proceedings of the 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP '16)*, pp. 1–6, November 2016.
- [20] K. Park and N. Zhang, "Behavior-based autonomous robot navigation on challenging terrain: a dual fuzzy logic approach," in *Proceedings of the IEEE Symposium on Foundations of Computational Intelligence (FOCI '07)*, pp. 239–244, IEEE, April 2007.
- [21] X. Zhao, Q. Luo, and B. Han, "Research on the real time obstacle avoidance control technology of biologically inspired hexapod robot," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 2306–2310, IEEE, June 2008.

Research Article

Inverse Kinematic Solutions of Dual Redundant Camera Robot Based on Genetic Algorithm

Libo Zhang,¹ Jingjie He,² and Su Wang¹

¹Department of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

²Advanced Innovation Center for Future Visual Entertainment, Beijing Film Academy, Beijing 100088, China

Correspondence should be addressed to Su Wang; wangsu@buaa.edu.cn

Received 14 June 2017; Accepted 14 August 2017; Published 26 September 2017

Academic Editor: Rafael Morales

Copyright © 2017 Libo 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.

Inverse kinematic solutions for a dual redundant camera robot in position are examined in order to alleviate operation difficulty and reduce time. The inverse kinematic algorithm is based on a basic genetic algorithm, and the genetic algorithm which is used to solve the problem of a redundant robot is mainly optimized in the joint space. On this basis, the genetic algorithm improvement strategies are studied. In this paper, a genetic algorithm with constrained 2 redundant degrees of freedom (DOF) is proposed through setting 2 parameter variables, with more flexible structure of optimization objective function and more efficient algorithm than basic genetic algorithm. Finally, the result of inverse kinematic algorithm is achieved in terms of the physical prototype.

1. Introduction

Coupled with mechanism geometric flexibility, a robot intelligent control system serves to complete a variety of complex operational tasks [1]. The geometric structure of a redundant robot is highly flexible, which provides valuable sources for both theoretical research and practicability [2]. The inverse kinematic solutions for a redundant robot are expressed as solutions of nonlinear equations, which generally can only be solved based on numerical iterative methods. Moreover, analytic solutions can be achieved only in some special geometric structures [3]. For special structures, the algebraic method employed to obtain inverse kinematic solutions can effectively reduce computational complexity, which is also conducive to online applications [4–13]. There are infinitely many solutions which can be used to solve inverse kinematics of a redundant robot. As a result, the identification of optimal solution is of relevant significance [14–16].

Generally, a genetic algorithm is employed to optimize inverse kinematic solutions of a redundant robot in the joint space [17]. Optimization objective function, sometimes called fitness function, consists of the minimum position error and rotation angle of each joint and solves to inverse kinematics in accordance with explicit expression from transformation matrix of a redundant robot [18]. Although optimization

accuracy can achieve a desired effect, there is a certain gap between optimal solutions and target position, because only the end position is optimized.

This paper analyzes an 8 DOF serial camera robot, which boasts flexible movement and accurate positioning ability. It requires a control of position and orientation of end-effector in position to solve inverse kinematics of camera robot primarily [19, 20]. This camera robot has 2 redundant DOF, complicated mechanical structure, and large size of sample space, which renders it challenging to achieve effective inverse kinematic solutions. Therefore, genetic algorithm for the inverse kinematics is proposed. End-effector minimum position and orientation error are commonly considered as optimization objective function of a basic genetic algorithm, which fails to identify the effective inverse kinematic solutions due to the large size of sample space, poor convergence of a genetic algorithm, and running over time. According to structural characteristics of the camera robot, a subinterval of optimal solutions is determined in advance to reduce dimension of the solution space. Subsequently, the relative position between body structure and target point is identified. Finally, a theoretical effective initial population is set by virtue of pattern search algorithm to reduce operation time and improve quality of the optimal solution. The directors can

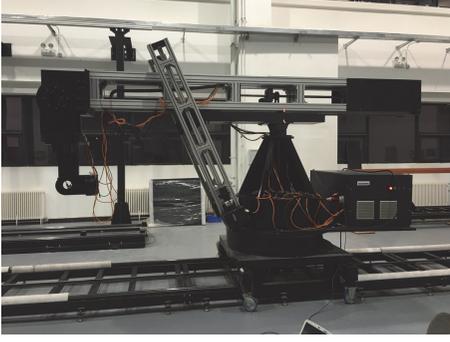


FIGURE 1: Camera robot physical prototype.

directly specify end-effector position without any support of professional operators and automatically select a best track to audition through the genetic algorithm. Meanwhile, total running time of this algorithm is in an allowed time.

2. Inverse Kinematic Solutions Based on Genetic Algorithm with Physical Constrained 2 Redundant DOF

2.1. Establishing Mathematical Mode. In this paper, the camera robot is an 8 DOF PRRPR-S robot. Compared with the traditional industrial robot, upper arms can be extended, and the robot body can be moved in a linear orbit, which can be found in Figure 1.

The kinematic model of this camera robot is established based on an assumption that robot is a rigid body. DH model is adopted for the first 7 joints of the robot [21–24], and the last one employs 6-parameter model. Finally, connecting rod coordinate system is created, as presented in Figure 2.

There are 8 movement axes. As shown in Figure 2, r_1 is the displacement of bottom linear motion, θ_2 is the rotation angle of bottom annular rotator, θ_3 is the rotation angle of top pitch rotator, r_4 is the displacement of top linear motion, θ_5 is the rotation angle of top distal pitch rotator, θ_6 is the rotation angle of end-effector posture rotator, θ_7 is the rotation angle of end-effector pitch rotator, θ_{ee} is the rotation angle of end-effector roll rotator, and the major structure of camera robot is the annular structure.

According to the position and orientation analysis of robot, the transformation matrix is obtained as

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & r_1 + r_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} \cos\left(\theta_2 - \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_2 - \frac{\pi}{2}\right) & 0 \\ \sin\left(\theta_2 - \frac{\pi}{2}\right) & 0 & \cos\left(\theta_2 - \frac{\pi}{2}\right) & 0 \\ 0 & -1 & 0 & r_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} \cos\left(\theta_3 - \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_3 - \frac{\pi}{2}\right) & l_3 \cdot \cos\left(\theta_3 - \frac{\pi}{2}\right) \\ \sin\left(\theta_3 - \frac{\pi}{2}\right) & 0 & \cos\left(\theta_3 - \frac{\pi}{2}\right) & l_3 \cdot \sin\left(\theta_3 - \frac{\pi}{2}\right) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & r_4 + r_{40} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_5 = \begin{bmatrix} \cos\left(\theta_5 + \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_5 + \frac{\pi}{2}\right) & 0 \\ \sin\left(\theta_5 + \frac{\pi}{2}\right) & 0 & \cos\left(\theta_5 + \frac{\pi}{2}\right) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_6 = \begin{bmatrix} \cos\theta_6 & 0 & \sin\theta_6 & 0 \\ \sin\theta_6 & 0 & -\cos\theta_6 & 0 \\ 0 & 1 & 0 & r_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_7 = \begin{bmatrix} \cos\left(\theta_7 - \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_7 - \frac{\pi}{2}\right) & 0 \\ \sin\left(\theta_7 - \frac{\pi}{2}\right) & 0 & \cos\left(\theta_7 - \frac{\pi}{2}\right) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_{ee} = \begin{bmatrix} \cos\theta_{ee} & \sin\theta_{ee} & 0 & 0 \\ -\sin\theta_{ee} & \cos\theta_{ee} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

(1)

Then the forward kinematic matrices of the robot are

$$\mathbf{T}_{ee} = \mathbf{A}_0 \cdot \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 \cdot \mathbf{A}_4 \cdot \mathbf{A}_5 \cdot \mathbf{A}_6 \cdot \mathbf{A}_7 \cdot \mathbf{A}_{ee}. \quad (2)$$

2.2. Inverse Kinematic Basic Genetic Algorithm in the Joint Space. Inverse kinematic basic genetic algorithm for the camera robot is currently an explicit expression of end-effector position according to the robot transformation matrix, and the key to solve inverse kinematic solutions is on the basis of the forward kinematic equation with genetic algorithm.

Make CRCP camera robot current position zero point; paT is homogeneous matrix of target position.

$$\text{CRCP} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$\text{paT} = \begin{bmatrix} 1 & 0 & 0 & 5000 \\ 0 & 1 & 0 & 1500 \\ 0 & 0 & 1 & 1500 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

In terms of this algorithm, the basic genetic algorithm parameters are set in the joint space, and an experiment is simulated by MATLAB. It turns out that computing time with this algorithm is about 2094.838654 s, and the minimum value of fitness function is 62858.09908986.

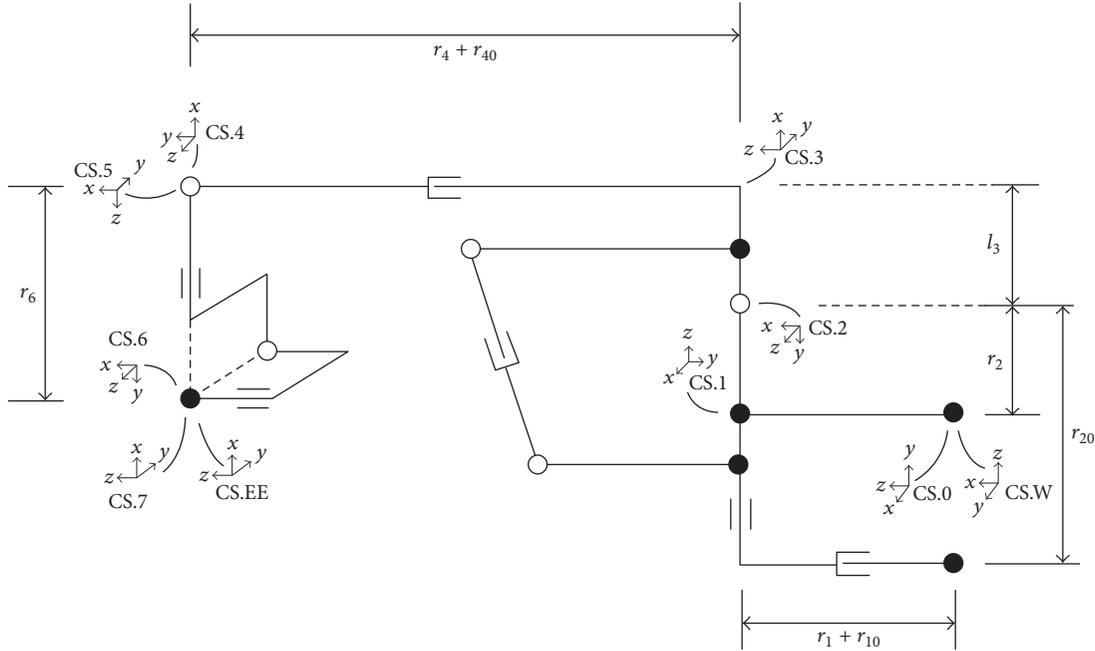


FIGURE 2: Connecting rod coordinate system.

The solution set of effective inverse solution for camera robot is

$$\text{CRTP} = [6851.62 \quad 2.7255 \quad 0.0195 \quad 132.3 \quad -2.6053 \quad -1.059 \quad 3.4355 \quad -0.5001]. \quad (4)$$

Robot reaches the target position, as shown in Figure 3.

According to the experiment results, camera robot does not reach the target position, and computing time of this algorithm takes about half an hour, which is beyond endurance in practice. That is principally due to a large number of camera robot joints, large movement of each axis especially linear motion at the top and the bottom, uncertain optimization objective function of algorithm, and complex camera robot mechanism, as well as many transformational matrices.

Therefore, the inverse kinematic solutions based on basic implicit genetic algorithm with constrained 2 redundant DOF are herein proposed to replace explicit expression of fitness function on individual genes.

2.3. Inverse Kinematic Solutions Based on Implicit Genetic Algorithm with Physical Constrained 2 Redundant DOF. The inverse kinematic solutions of camera robot are obtained by genetic algorithm, which takes an arbitrary set of 2-dimensional vector of 2 redundant DOF as an individual in physical constraint. Effective inverse solutions which are obtained with least relative weight of each joint angle variation are extracted in infinite inverse solutions through limiting optimization objective function. This indicates that, in the same movement time, there is small variation with large load inertia and large variation with small load inertia.

Make CRCP camera robot current position, CRTP camera robot target position, and CRMD movement displacement of each axis.

$$\begin{aligned} \text{CRCP} &= [\mathbf{r}_{1C} \quad \theta_{2C} \quad \theta_{3C} \quad \mathbf{r}_{4C} \quad \theta_{5C} \quad \theta_{6C} \quad \theta_{7C} \quad \theta_{eeC}]^T \\ \text{CRTP} &= [\mathbf{r}_{1T} \quad \theta_{2T} \quad \theta_{3T} \quad \mathbf{r}_{4T} \quad \theta_{5T} \quad \theta_{6T} \quad \theta_{7T} \quad \theta_{eeT}]^T \\ \text{CRMD} &= \text{CRTP} - \text{CRCP} = [\Delta \mathbf{r}_1 \quad \Delta \theta_2 \quad \Delta \theta_3 \quad \Delta \mathbf{r}_4 \quad \Delta \theta_5 \quad \Delta \theta_6 \quad \Delta \theta_7 \quad \Delta \theta_{ee}]^T \\ &= [\mathbf{r}_{1T} - \mathbf{r}_{1C} \quad \theta_{2T} - \theta_{2C} \quad \theta_{3T} - \theta_{3C} \quad \mathbf{r}_{4T} - \mathbf{r}_{4C} \quad \theta_{5T} - \theta_{5C} \quad \theta_{6T} - \theta_{6C} \quad \theta_{7T} - \theta_{7C} \quad \theta_{eeT} - \theta_{eeC}]^T. \end{aligned} \quad (5)$$

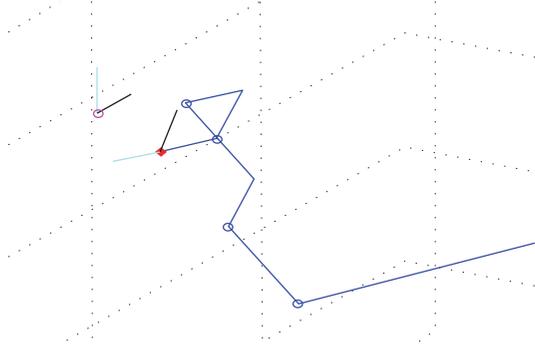


FIGURE 3: The target position on basic genetic algorithm.

Absolute movement displacement of each axis is considered to be a measure of movement, represented by CRMADA.

$$\text{CRMADA} = [|\Delta \mathbf{r}_1| \quad |\Delta \theta_2| \quad |\Delta \theta_3| \quad |\Delta \mathbf{r}_4| \quad |\Delta \theta_5| \quad |\Delta \theta_6| \quad |\Delta \theta_7| \quad |\Delta \theta_{ee}|]^T. \quad (6)$$

Considering load rotary inertia of each motor, make CRMW movement weight of each axis.

$$\text{CRMW} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3 \quad \mathbf{w}_4 \quad \mathbf{w}_5 \quad \mathbf{w}_6 \quad \mathbf{w}_7 \quad \mathbf{w}_{ee}] = [50 \quad 200 \quad 100 \quad 50 \quad 30 \quad 1 \quad 1 \quad 1]. \quad (7)$$

In this paper, in terms of weight assignment principle, the larger the load rotary inertia is, the larger the weight is, and vice versa. After achieving optimization objective function, the rotary inertia becomes larger, the displacement becomes smaller, and vice versa. Besides, the smaller the value of the optimization objective function, the better the optimization performance to camera robot operation.

Based on our knowledge, the bottom line motion almost does not lead to vibration of end-effector of camera robot. Therefore, the movement of bottom linear motion axis can be large. Conversely, the bottom annular rotating axis has a great impact on end-effector with movement as small as possible. Movement of the top linear motion should be smaller without requirement of a higher operational height. There can be more movement of end-effector, and load rotary inertia of each motor on end-effector is fairly small.

On this basis, camera robot movement amplitude weighting as the optimization objective function is represented by CRMMAW.

$$\begin{aligned} \text{CRMMAW} &= \text{CRMW} \cdot \text{CRMADA} \\ &= [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3 \quad \mathbf{w}_4 \quad \mathbf{w}_5 \quad \mathbf{w}_6 \quad \mathbf{w}_7 \quad \mathbf{w}_{ee}] \\ &\cdot [|\Delta \mathbf{r}_1| \quad |\Delta \theta_2| \quad |\Delta \theta_3| \quad |\Delta \mathbf{r}_4| \quad |\Delta \theta_5| \quad |\Delta \theta_6| \quad |\Delta \theta_7| \quad |\Delta \theta_{ee}|]^T \quad (8) \\ &= \mathbf{w}_1 \cdot |\Delta \mathbf{r}_1| + \mathbf{w}_2 \cdot |\Delta \theta_2| + \mathbf{w}_3 \cdot |\Delta \theta_3| + \mathbf{w}_4 \cdot |\Delta \mathbf{r}_4| + \mathbf{w}_5 \\ &\cdot |\Delta \theta_5| + \mathbf{w}_6 \cdot |\Delta \theta_6| + \mathbf{w}_7 \cdot |\Delta \theta_7| + \mathbf{w}_{ee} \cdot |\Delta \theta_{ee}|. \end{aligned}$$

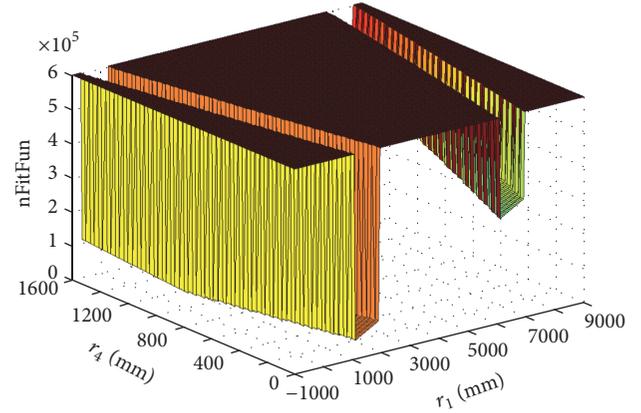


FIGURE 4: The graph of fitness function.

The larger the value of CRMMAW, the larger the movement amplitude weighting and the greater the maximum value of acceleration in the motion process, and vice versa. In addition, the solutions minimizing value of CRMMAW are extracted from the infinite number of inverse solutions in an ideal situation.

For invalid inverse solution $[r_1 \quad r_4]$, the movement amplitude weighting is set as a very large value, that is, $n\text{CRMMAW}_F = 600000$.

Fitness function to be optimized is called objective function in the classical optimization algorithm. In this paper, the minimum value of objective function is found through using the setting of GA toolbox in MATLAB, and the movement amplitude weighting is taken as optimization objective function, which is included in fitness function $HijGAFcn$,

$$\text{FitnessFunction} = HijGAFcn. \quad (9)$$

In function structure, $[nFitFcn] = HijGAFcn(\mathbf{r})$, function variable $\mathbf{r} = [r_1 \quad r_4]$. In order to develop a more profound understanding of the optimization space, the image of fitness function $HijGAFcn$ under the target position paT is drawn, as presented in Figure 4.

Figure 4 shows that the minimum value of fitness function exists in a space consisting of $r_1 \in [1000, 2000]$ and $r_4 \in [0, 500]$, which is related to the weight setting of optimization objective function.

A set of random $[r_1 \quad r_4]$ as a 2-dimensional individual is introduced to the genetic algorithm, Individual = FUTS = $[r_1 \quad r_4]$. The 50-group target value of bottom linear motion axis and top linear motion axis are randomly selected, with 2 genes for each individual. As a result, there are 10 rows and 2 columns in population matrix with identical individuals in the population at the same time.

The setting of zero position CRTP and target position paT are the same as basic genetic algorithm. Genetic algorithm parameters are set, and experiment is simulated in MATLAB. It turns out that computing time with this algorithm is about 112.775544 s, and the minimum value of fitness function is 390739. Average and minimum values of fitness function change with the genetic algebra, as shown in Figure 5.

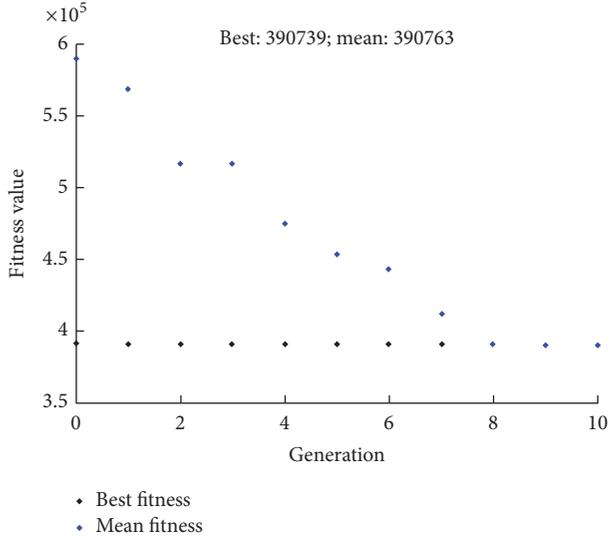


FIGURE 5: Average and minimum values of fitness function change with the genetic algebra on implicit genetic algorithm.

In this simulation experiment, although the minimum value of fitness function is consistent with the mean value of population after 8 generation, it fails to converge all the time.

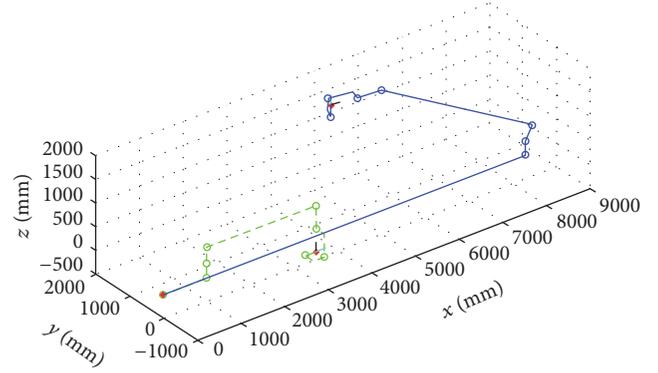


FIGURE 6: The target position on implicit genetic algorithm.

At this time, the individual corresponding value of fitness function can be obtained as

$$[r_1 \ r_4] = [7296.7 \ 505.9]. \quad (10)$$

The solution set of effective inverse solution for camera robot is

$$\text{CRTP} = [7296.7 \ 2.71459 \ -0.04783 \ 505.9 \ -2.37323 \ 0 \ 0.85027 \ -0.42700]. \quad (11)$$

Robot reaches the target position, as shown in Figure 6. The green dashed line is initial position, and the blue line is target point position.

Contrasting with the graph of fitness function, it shows that $r_1 \in [7000, 8000]$ is local optimization after optimizing, and the optimization effect is not ideal. It is due to the fact that the range of bottom and top linear motion is larger, and the possibility of obtaining effective solution from random values in 2 redundant DOF is quite low.

3. The Improved Genetic Algorithm

3.1. Improved Genetic Algorithm Based on Theoretical Constraints on r_1 and Physical Constraints on r_4 . The implicit genetic algorithm with physical constrained 2 redundant DOF is improved through analyzing range of motion for dual redundant camera robot. Known target position, the movement range of bottom linear motion r_1 , and top linear motion r_4 are prejudged with mechanism characteristics of camera robot, which refers to limiting the range of genetic variables of genetic algorithm to improve the computational efficiency of genetic algorithm. The improved genetic algorithm flow chart is presented in Figure 7.

In Figure 6, paT represents a target point; vRITS is the range of r_1 with theoretical constraints. vR4PS is the range of r_4 with physical constraints, and *RITSFcn* is obtained from the analysis of camera robot workspace and motion range.

vRITS can be deduced out from *RITSFcn* through processing paT, which only considers the physical constraint on r_4 . Given paT, r_1 , and r_4 , the inverse kinematic of camera robot is operated by *HijGAFcn*, the inverse kinematic solutions are substituted into CRMAW, and CRMAW corresponding optimum solution $[r_1 \ r_4]$ is treated as the current function fitness *ngFitFcn*.

Prior to solving inverse kinematics of the camera robot, if r_1 subinterval existing minimum fitness function value is determined at first, the computation time can be reduced by half. The intersection point of top pitch axis of camera robot ring body and plane perpendicular to the bottom line is defined as S , namely, shoulder point. The X coordinates S_x is specified as decision point of robot main body, and the pT coordinate pT $_x$ is identified as decision point of target point. In addition, according to analysis of robot motion range, the target point in robot workspace is arbitrarily given, and the motion range of bottom linear motion can be obtained as

$$r_1 \in [r_{1a}, r_{1b}] \cup [r_{1c}, r_{1d}]. \quad (12)$$

When $S_x \leq \text{pT}_x$, r_1 is limited to $[r_{1a}, r_{1b}]$, and when $S_x > \text{pT}_x$, r_1 is limited to $[r_{1c}, r_{1d}]$.

The setting of initial position, target position, and parameters of genetic algorithm are the same as before. Based on the simulation results, the computing time is about 109.602341 s with this improved genetic algorithm, and the minimum

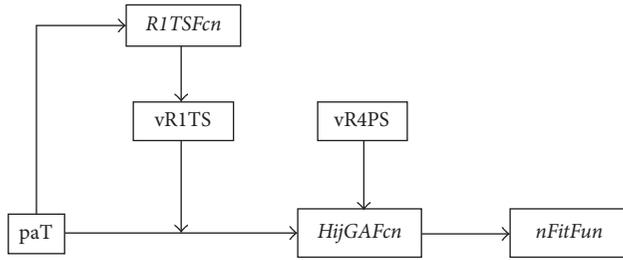


FIGURE 7: Improved genetic algorithm flow chart.

$$\text{CRTP} = [1403.06 \quad 0.5238 \quad -0.01892 \quad 0.42 \quad -2.57024 \quad 0 \quad 1.0184 \quad -2.6178]. \quad (14)$$

Pose of the camera robot when target position is reached is shown in Figure 9.

Although efficiency of implicit genetic algorithm has been improved, the genetic algorithm has a slow convergence rate at each time. Besides, a rather large gap exists between the minimum value and average value of fitness function, and the minimum value of fitness function is also different, even the failure of optimization algorithm. It is due to fact that the selection of population individual is still random, and the optimal individual still can not be obtained as expected, even though expanding population size and increasing the truncated generation.

3.2. Setting the Initial Population and Using Pattern Search Algorithm. In order to further improve algorithm, the r_1 is averagely selected in its theoretical intervals, the value range of r_4 is determined according to each r_1 value, and the intermediate value of interval is selected as the value of r_4 . Setting on the basis of this initial population, the individual is evenly distributed in theoretical range. In addition, it can solve the problem that the excellent gene r_4 is lost due to matching with inappropriate gene r_1 . However, the more accurate local optimization ability for complex fitness function is poor, and population size and truncated generation are limited by the

$$\text{CRTP} = [1401.9 \quad 0.52360 \quad -0.01931 \quad 0.000001 \quad -2.56796 \quad 0 \quad 1.01647 \quad -2.617994]. \quad (16)$$

The camera robot position when reaching target position is shown in Figure 11.

According to experimental results, it can be found that each time the results are basically consistent, which is close to the global minimum of fitness function. In accordance with the 10 key points, total running time is about 0.5 hours, and the running time is considered as acceptable.

3.3. Comparison. In the same setting of initial position, target position, and parameters of genetic algorithm, the final improved algorithm in comparison with initial algorithm

value of fitness function is 390739. Average and minimum values of fitness function change with the genetic algebra, as shown in Figure 8.

After improvement, the minimum values of fitness function is 70361.4, and the individual corresponding value of fitness function is

$$[r_1 \quad r_4] = [1403.06 \quad 0.42]. \quad (13)$$

Effective inverse solution for the camera robot is

computing time. According to Figure 3, value distribution of fitness function boasts certain regularity, and the minimum function value is few in number. As a result, after identifying the global optimal solutions, the pattern search method is used to address the precise optimal solutions [25].

The setting of initial position, target position, and parameters of genetic algorithm is the same as before. 20 values are selected in effective theoretical intervals of r_1 , the value range of r_4 is determined according to each r_1 value, and the intermediate value of r_4 interval and r_1 forms an individual. The initial population individual constitutes an initial population matrix gInitPop , the reference value of *HybridFcn* is set as *patternsearch*, and then simulation experiment is conducted for 10 times.

After the experiment, computing time is on average about 203.6 s at a time, and the minimum values of fitness function in 10 experiments are shown in Figure 10.

The minimum value of fitness function is about 70283 in 10 experiments, and the optimal individual difference is very small, which is nearly

$$[r_1 \quad r_4] = [1401.92379 \quad 0.000001]. \quad (15)$$

Corresponding inverse solution for the camera robot is

implicit genetic algorithm with physical constrained 2 dual redundant DOF, and the motion range of each axis is shown in Table 1.

The comparison between motion range and movement weight of each axis demonstrates that the movement of bottom linear motion, the rotation angle of bottom annular rotator, and the rotation angle of top pitch rotator are reduced significantly. It indicates that translation and rotation of main structure are very small, and the top distal pitch axis has more movement in comparison with top pitch axis, when camera robot reaches the target point. End-effector with

TABLE 1: Comparison of genetic algorithm (GA).

Parameter	Initial position	Basic GA	Implicit GA	Improved GA
r_1	0	6851.62	7296.7	1401.9
θ_2	0	2.7255	2.71459	0.52360
θ_3	0	0.0195	-0.04783	-0.01931
r_4	0	132.30	505.9	0.000001
θ_5	0	-2.6053	-2.37323	-2.56794
θ_6	0	-1.0590	0	0
θ_7	0	3.4355	0.850267	1.01647
θ_{ce}	0	-1.5001	-0.42700	-2.61799

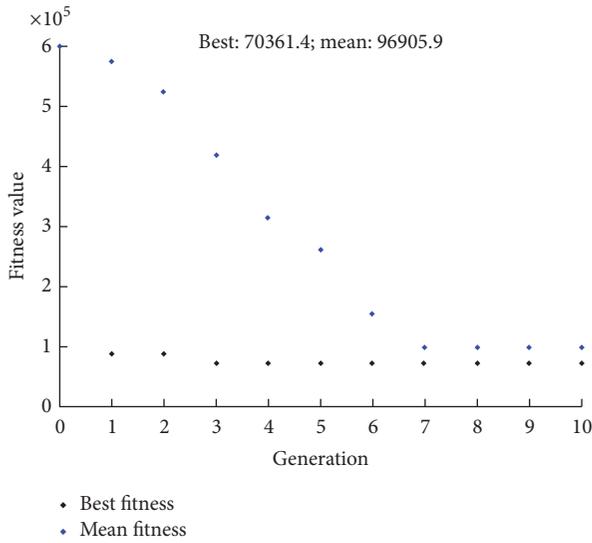


FIGURE 8: Average and minimum values of fitness function change with the genetic algebra by limiting genetic variables.

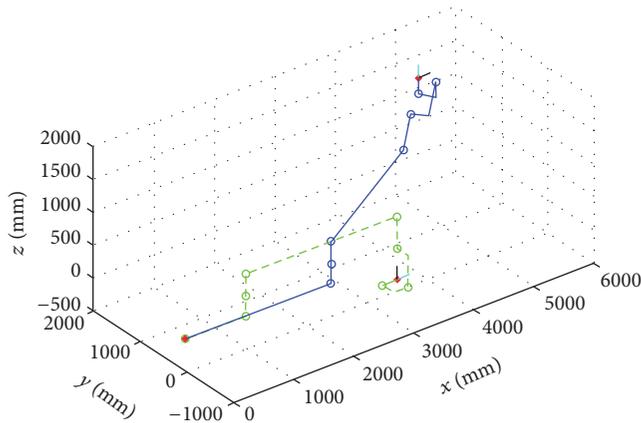


FIGURE 9: The target position by limiting genetic variables.

3 axes and small load inertia has more movement than the unimproved one. This comparison achieves the desired optimization results, and the movement of axis with smaller load inertia is larger. On the contrary, the movement of axis with larger load inertia is smaller. Besides, the top linear movement is about 0, which accords with the idea that the

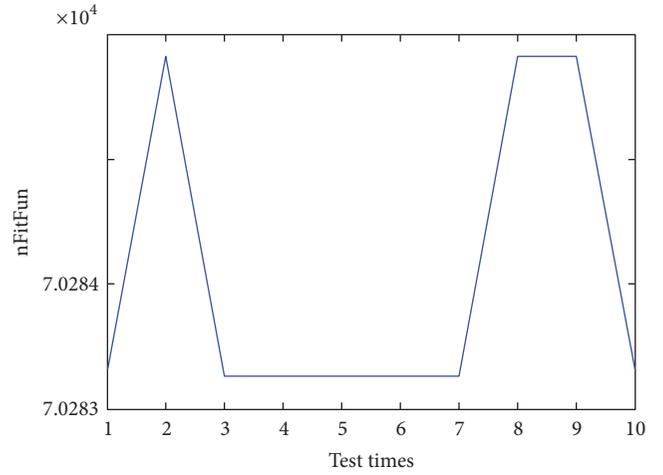


FIGURE 10: The minimum values of fitness function in 10 experiments.

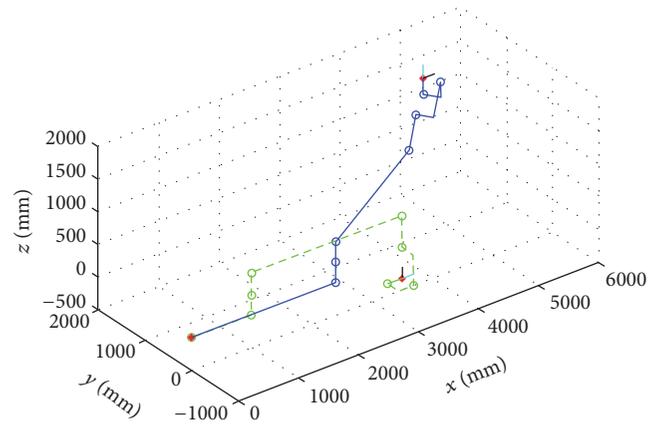


FIGURE 11: The target position on improved genetic algorithm.

movement of top linear motion is reduced as much as possible in the case of operational height.

4. Experimental Validation

Considered as Coordinate Measuring Machine, Metronor is used to measure the space position of target point. Besides,

the results of optimal solution obtained by final improved algorithm are verified on the physical prototype.

The initial position of camera robot is

$$\text{CRCP} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]. \quad (17)$$

Pose matrix of end-effector coordinate system relatively to inverse solutions world coordinate system is

$$\text{paZ} = \begin{bmatrix} 0 & 0 & 1 & 3500 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -330 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

$$\text{CRTP} = [1401.9 \ 0.52360 \ -0.01931 \ 0 \ -2.56796 \ 0 \ 1.01647 \ -2.617994]. \quad (21)$$

According to kinematic model and reduction ratio of the robot physical prototype, the position input command vector of host computer is obtained as

$$\text{StU} = [1800168 \ -268860 \ 796132 \ 0 \ -11313862 \ 0 \ -3105818 \ -395250]. \quad (22)$$

The end-effector coordinate system is set, as shown in Figure 12.

The end-effector coordinate system is represented by 3 space points, coordinate origin \mathbf{O} , point \mathbf{X} in the x -axis, and point \mathbf{Z} in the z -axis, respectively. In addition, Metronor system is employed to determine the coordinates of this 3 target space points, digital camera placement, and shooting angle, as shown in Figure 13.

Measurements of 3 points in the end coordinate system are shown in Table 2

Among them, coordinates of all point are relative to the world coordinate system of Metronor. The coordinate system of end-effector adopts right-hand rule. Y direction vector is obtained by the cross product of 2 unit vector in X and Z direction, as shown in Table 3.

The zero position of the end-effector coordinate system which is relative to the world coordinate system of Metronor is obtained based on Table 3. The homogeneous matrix of the zero position is

$$\text{paZ}^M = \begin{bmatrix} -0.0448 & 0.3106 & -0.9504 & 1281.147 \\ 0.9794 & -0.1734 & -0.0561 & -1217.745 \\ -0.1969 & -0.9333 & -0.3059 & -7351.774 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (23)$$

The matrix of target position is

$$\text{paT} = \begin{bmatrix} 1 & 0 & 0 & 5000 \\ 0 & 1 & 0 & 1500 \\ 0 & 0 & 1 & 1500 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (19)$$

Pose homogeneous matrix of target point coordinate system based on end-effector coordinate system is

$$\text{paT}^Z = (\text{paZ})^{-1} \cdot \text{paT} = \begin{bmatrix} 0 & 0 & 1 & 1830 \\ 0 & -1 & 0 & -1500 \\ 1 & 0 & 0 & 1500 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

Optimum solution obtained by improved algorithm is

And the homogeneous matrix of the target position is

$$\text{paT}^M = \begin{bmatrix} -0.92 & -0.3848 & -0.0618 & -686.575 \\ -0.1239 & 0.1049 & 0.9884 & 762.754 \\ -0.3719 & 0.917 & -0.139 & -6850.280 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (24)$$

Thus, when the paZ is a reference coordinate system, the homogeneous matrix of paT relative to paZ is

$$\text{paT}^Z = (\text{paZ}^M)^{-1} \cdot \text{paT}^M = \begin{bmatrix} -0.055 & -0.065 & 0.998 & 1856.540 \\ 0.083 & -0.996 & -0.061 & -1426.166 \\ 0.998 & 0.082 & -0.002 & 1516.816 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (25)$$

Compared with (20), it can be seen that the measurement is almost consistent with theoretical position. The target pose error between actual measurement and theoretical simulation is primarily attributed to the parameter error of robot links and the artificial error of three coordinate measuring systems.

TABLE 2: Measurements of 3 points in the end coordinate system.

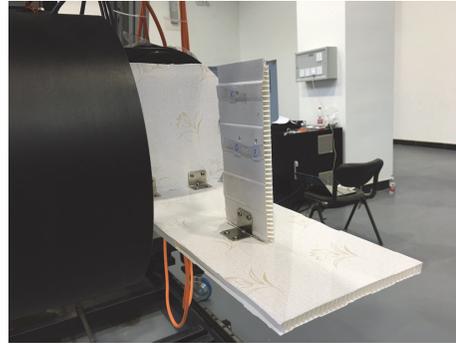
Parameter	Zero position	Target position
O_Z	$[1281.147 \quad -1217.745 \quad -7351.774]$	$[-686.575 \quad 762.754 \quad -6850.280]$
X_Z	$[1278.963 \quad -1170.012 \quad -7361.370]$	$[-731.258 \quad 756.734 \quad -6868.342]$
Z_Z	$[1248.746 \quad -1219.657 \quad -7362.201]$	$[-688.626 \quad 795.572 \quad -6854.896]$

TABLE 3: Direction vector in the end coordinate system.

Parameter	Vector	Unit vector
$O_Z X_Z$	$[-2.184 \quad 47.733 \quad -9.596]$	$[-0.0448 \quad 0.9794 \quad -0.1969]$
$O_Z Y_Z$	—	$[0.3106 \quad -0.1734 \quad -0.9333]$
$O_Z Z_Z$	$[-32.401 \quad -1.912 \quad -10.427]$	$[-0.9504 \quad -0.0561 \quad -0.3059]$
$O_T X_T$	$[-44.683 \quad -6.02 \quad -18.062]$	$[-0.92 \quad -0.1239 \quad -0.3719]$
$O_T Y_T$	—	$[-0.3848 \quad 0.1049 \quad 0.9170]$
$O_T Z_T$	$[-2.051 \quad 32.818 \quad -4.616]$	$[-0.0618 \quad 0.9884 \quad -0.139]$



(a)



(b)

FIGURE 12: (a) Camera robot with end-effector. (b) Measuring surface of coordinate frame.



(a)



(b)

FIGURE 13: (a) Digital camera placement. (b) digital camera shooting angle.

5. Conclusions

The present study aims to reduce the difficulty and time used for the inverse kinematic algorithm and the complex camera robot. It is hard to obtain effective inverse kinematic solutions due to the large size of sample space, poor convergence of GA, and running over time. Therefore, a new algorithm is proposed and improved. Although the optimal solution can

be obtained based on a new algorithm, stability of the algorithm is relatively poor even with expanding population size and increasing truncated generation. In addition, the inverse kinematic solutions are obtained through analyzing the range of motion for dual redundant camera robot. The improved algorithm achieved by setting the initial population and using pattern search algorithm can converge to a global optimum, which meets the requirements of practical application.

Finally, the zero position of physical prototype is calibrated through using coordinate measuring machine, and the camera robot is driven to specified target position based on optimal inverse solutions. The actual measurement position of end-effector is basically consistent with the theoretical results, and the goal of alleviating operation difficulty and reducing operation time of camera robot is consequently reached.

Conflicts of Interest

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

References

- [1] C. S. G. Lee and M. Ziegler, "Geometric approach in solving inverse kinematics of puma robots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 20, no. 6, pp. 695–706, 1984.
- [2] T. Asfour and R. Dillmann, "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematic problem," in *Proceedings of the IEEE/RSJ International Conference on IROS*, pp. 1407–1412, October 2003.
- [3] A. C. Nearchou, "Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm," *Mechanism and Machine Theory*, vol. 33, no. 3, pp. 273–292, 1998.
- [4] M. A. Ali, H. A. Park, and C. S. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, pp. 704–709, Taipei, Taiwan, October 2010.
- [5] M. Kallmann, "Analytical inverse kinematics with body posture control," *Computer Animation and Virtual Worlds*, vol. 19, no. 2, pp. 79–91, 2008.
- [6] J. Duffy, *Analysis of Mechanisms And Robot Manipulators*, Edward Arnold, London, UK, 1980.
- [7] A. A. Goldenberg, B. Benhabib, and R. G. Fenton, "A Complete Generalized Solution to the Inverse Kinematics of Robots," *IEEE Journal on Robotics and Automation*, vol. 1, no. 1, pp. 14–20, 1985.
- [8] J. Angeles, "On the numerical solution for the inverse kinematic problem," *The International Journal of Robotics Research*, vol. 4, pp. 21–37, 1985.
- [9] B. Damas and J. Santos-Victor, "An online algorithm for simultaneously learning forward and inverse kinematics," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012*, pp. 1499–1506, Vilamoura, Portugal, October 2012.
- [10] D. Manocha and J. F. Canny, "Efficient inverse kinematics for general 6R manipulators," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 648–657, 1994.
- [11] O. Kanoun, F. Lamiroux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.
- [12] C. Chen and D. Jackson, "Parameterization and evaluation of robotic orientation workspace: A geometric treatment," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 656–663, 2011.
- [13] J. Lenareie, "An Efficient Numerical Approach for Calculating the Inverse Kinematics for Robot Manipulators," *Robotica*, vol. 3, no. 1, pp. 21–26, 1985.
- [14] S. Lee and A. K. Bejczy, "Redundant arm kinematic control based on parameterization," in *Proceedings of the IEEE ICRA*, pp. 458–465, 1991.
- [15] D. Tolani and N. I. Badler, "Real-time inverse kinematics of the human arm," *Presence: Teleoperators and Virtual Environments*, vol. 5, no. 4, pp. 393–401, 1996.
- [16] K. Kreutz-Delgado, M. Long, and H. Seraji, "Kinematic Analysis of 7-DOF Manipulators," *The International Journal of Robotics Research*, vol. 11, no. 5, pp. 469–481, 1992.
- [17] S. Kucuk and Z. Bingul, "Inverse kinematics solutions for industrial robot manipulators with offset wrists," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 38, no. 7-8, pp. 1983–1999, 2014.
- [18] Y. Y. Jin and W. Qin, "Adaptive genetic algorithms based on simulated annealing for the inverse kinematic of the manipulator," *Jinlong Machinery & Electronics Co., LTD*, vol. 1, pp. 61–63, 2007.
- [19] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131–1142, 2008.
- [20] G. K. Singh and J. Claassens, "An analytical solution for the inverse kinematics of a redundant 7DoF manipulator with link offsets," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, pp. 2976–2982, Taipei, Taiwan, October 2010.
- [21] C. E. Wilson and J. P. Sadler, *Kinematic and Dynamics of Machinery*, Harper & Row Publishers, New York, NY, USA, 1983.
- [22] D. E. Whitney, *Mechanical Assemblies: Their Design Manufacture and Role in Product Development*, Oxford University Press, New York, NY, USA, 2004.
- [23] R. P. Judd and A. B. Knasinski, "Technique to calibrate industrial robots with experimental verification," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 20–30, 1990.
- [24] H. W. Stone and A. C. Sanderson, "Statistical performance evaluation of the S-model arm signature identification technique," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 939–946, 1988.
- [25] A. A. Ata and T. R. Myo, "Optimal point-to-point trajectory tracking of redundant manipulators using generalized pattern search," *International Journal of Advanced Robotic Systems*, vol. 2, no. 3, pp. 249–253, 2005.

Research Article

Robust Course Keeping Control of a Fully Submerged Hydrofoil Vessel without Velocity Measurement: An Iterative Learning Approach

Sheng Liu, Changkui Xu, and Lanyong Zhang

College of Automation, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Changkui Xu; xuchangkui@hrbeu.edu.cn

Received 15 March 2017; Accepted 13 June 2017; Published 20 July 2017

Academic Editor: Rafael Morales

Copyright © 2017 Sheng Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a novel robust output feedback control methodology for the course keeping control of a fully submerged hydrofoil vessel. Based on a sampled-data iterative learning strategy, an iterative learning observer is established for the estimation of system states and the generalized disturbances. With the state observer, a feedback linearized iterative sliding mode controller is designed for the stabilization of the lateral dynamics of the fully submerged hydrofoil vessel. The stability of the overall closed-loop system is analyzed based on Lyapunov stability theory. Comparative simulation results verify the effectiveness of the proposed control scheme and show the dominance of the disturbance rejection performance.

1. Introduction

As an advanced marine vehicle, the fully submerged hydrofoil vessel (FSHV) can cruise at a high speed against rough sea waves. The lift force of the hydrofoils generated by the high-speed fluid elevates the ship hull up from the water, which highly reduces the wave resistance and friction on the ship. However, the lift force also destabilizes the open-loop system of the FSHV. Therefore, it is necessary to equip an autopilot for this type of marine vehicles [1–3]. Currently, the commercial control systems equipped on board are based on optimal control theory [4], which has a weak disturbance rejection property. For the sake of the high cruising speed, the nonlinear hydrodynamic damping of the FSHV cannot be neglected [5, 6]. Therefore, the widely used linear model of marine surface vessels is no longer applicable given the strong coupling between yaw and roll dynamics. Moreover, the model uncertainties and disturbances caused by wind, waves, and currents prevent precise steering of this species of marine vehicle.

The path of marine surface vehicles is usually a straight line or straight lines formed by waypoints at the open sea. Hence, there is little coupling between the longitudinal dynamics and the lateral dynamics. Some literatures present

the riding control design of the FSHV [4, 7]. This paper mainly focuses on the steering control of the FSHV.

For the nonlinear steering control of marine vessels, a series of control methodologies have been proposed, such as advanced sliding mode control [8–10], robust control [11, 12], and adaptive control [13, 14]. But these passive disturbance attenuation control methods may suffer from actuator chattering, system conservation, derivative explosion, and rigorous proof for stability. Intelligent approximation tools such as neural network, fuzzy logic, and SVM are introduced as feedforward components for state estimate and disturbance compensation, but the convergence rate of weight function and the problem of extremum solving still remain to be settled [6, 12].

Disturbance rejection control is also widely used in applications of rigid body dynamics and servo systems [15–17]. Disturbance observer based control (DOBC) approaches provide an active way to handle disturbances and improve the robustness of the closed-loop systems. The effectiveness of DOBC has been shown in many fields such as manipulator tracking control [18], missile guidance and control [19–21], and motor control [22, 23]. However, traditional DOB methodology cannot be used in nonlinear dynamics due to the limitation of linear system theory, and the low pass filter

must be designed strictly such that the convergence of the disturbance estimate can be guaranteed [16].

Most of the methodologies for motion control of marine vehicles in the existing literatures employ full-state feedback. In many practical applications, only angle signals can be measured and utilized [24–32] since the sensor information of angle velocity usually contains irregular noises caused by the environment and properties of electric components, which are difficult to filter [5, 33]. Therefore, it is necessary to design a state observer to make state estimation for the system. State observer based control schemes have been developed for many species of nonlinear systems such as nonlinear time-delay systems [34, 35], Lipschitz nonlinear systems [24, 26, 36], and other structured systems [15, 27, 28]. According to the design theory, state observers can be classified, for example, as Luenberger observer [29, 36], high-gain observer [30], and sliding mode observer [31, 32]. For nonlinear systems with model uncertainties and external disturbances, there are the following disadvantages of these above-mentioned state observers.

- (1) In most control engineering applications, the system structures do not match these observers very well, so the observers require complicated coordinated transformations.
- (2) Many state observer methodologies are designed for SISO systems, but the practicality and stability of these methodologies for MIMO systems still need to be further discussed.
- (3) Most of these observers estimate only the system states, while the disturbance estimate could not be overall considered.

The course keeping control of the FSHV suffers severely from model uncertainties and wave disturbances. Therefore, it is of great necessity to design an observer which is able to estimate system states as well as the generalized disturbances integrately. Furthermore, in order to guarantee the disturbance rejection performance of the control scheme, the feedback controller of the inner loop should also achieve disturbance attenuation properties, such that the residual error of the disturbance estimate can be compensated through a composite strategy.

In this paper, a novel output feedback control methodology is proposed for the course keeping control of the FSHV based on a sampled-data iterative learning approach. An iterative learning observer is established for the state estimation as well as generalized disturbances. Then a sliding model controller with an iterative learning sliding manifold is presented for the stabilization of the lateral dynamics of the FSHV. The design of the iterative sliding surface introduces a memory effect for the controller, which includes both the current and past information of the system states and outperforms the standard sliding mode control in the improvement of the transient performance.

The rest of the paper is organized as follows. In Section 2, the mathematical model of the lateral dynamics of the FSHV is established. In Section 3, an observer based sliding mode control is proposed based on iterative learning

approach. Then we analyze the Lyapunov stability of the overall closed-loop system in Section 4. Simulation is carried out in Section 5 to validate the effectiveness of the proposed methodology, followed by Conclusions in Section 6.

2. Problem Formulation

A typical configuration of a fully submerged hydrofoil vessel is shown in Figure 1 [4]. The T-shaped bow foil is equipped with two controlled flaps, acting together. The aft foil has a pair of central flaps and two pairs of ailerons. Struts of the aft foil are equipped with rudders, which are used for roll and yaw dynamics together with the ailerons. The bow foil and the central part of the aft foil are for longitudinal motion control.

Course keeping control, or so-called ship heading control, is the primary task of the autopilots of ships. The control objective of course keeping control is to stabilize the yaw angle to a desired heading angle. As to the course keeping control for conventional marine surface vessels, the roll dynamics is rarely considered due to the weakness of the control actuators for roll dynamics. The roll restoring moments provide static stability, which the FSHV does not have. Therefore, it is necessary to take the roll dynamics into consideration for the course keeping control design for the FSHV.

The surge speed u is usually controlled by an individual propulsion system and is kept at a fixed speed u_0 when a ship is maneuvered in the course keeping mode. Accordingly, a 2-DOF steering model of the FSHV is shown as

$$\begin{aligned} \dot{\eta} &= J(\eta)v \\ M\dot{v} + C(u_0, v)v + D(u_0, v)v + G(\eta)\eta &= bu + \tau_d, \end{aligned} \quad (1)$$

where $\eta = [\phi, \psi]^T$, in which ϕ and ψ denote the roll angle and heading angle of the FSHV with coordinates in the earth-fixed frame, respectively; $v = [p, r]^T$, in which p and r represent the angular velocities with coordinates in the body-fixed frame, respectively; $J = \begin{bmatrix} 1 & 0 \\ 0 & \cos\phi \end{bmatrix}$ is the Jacobian transformation matrix related to the above frames; $M = \begin{bmatrix} I_x - K_p & mx_g - K_r \\ mx_g - N_p & I_z - N_r \end{bmatrix}$ is the inertia including added mass; and $C(u_0, v) = \begin{bmatrix} 0 & 0 \\ 0 & mx_g u_0 \end{bmatrix}$ denotes the Coriolis and centripetal matrix. Selecting the origin point of the body-fixed frame at the center of gravity yields $x_g = 0$. $D(u_0, v)$ is the coupling interaction caused by the nonlinear hydrodynamic damping, which is defined as $D = D_L + D_N$. $D_L = \begin{bmatrix} -K_p & -K_r \\ -N_p & -N_r \end{bmatrix}$ stands for the linear part and $D_N = \begin{bmatrix} 0 & -K_{r\phi\phi}\phi^2 \\ 0 & -N_{rr\phi}r\phi \end{bmatrix}$ represents the nonlinear damping, respectively. K_p , $K_{r\phi\phi}$, K_r , $K_{r\phi\phi}$, and $N_{r\phi\phi}$, N_p , N_r , $N_{rr\phi}$ are the hydrodynamic coefficients. In the low-speed mode, the high-order terms and coupling interactions among the forces from each DOF are not considered. Therefore, $D(u_0, v)$ is often regarded as a linear term and even cancelled in many literatures. However, given the high-speed character of the FSHV, the nonlinear damping hydrodynamics can no longer be neglected. $G(\eta)$ represents the gravity term which is described as $G(\eta) = [W\overline{GM}_T, 0]^T$, where $W = mg$ is the weight and \overline{GM}_T is the transverse metacenter height [37].

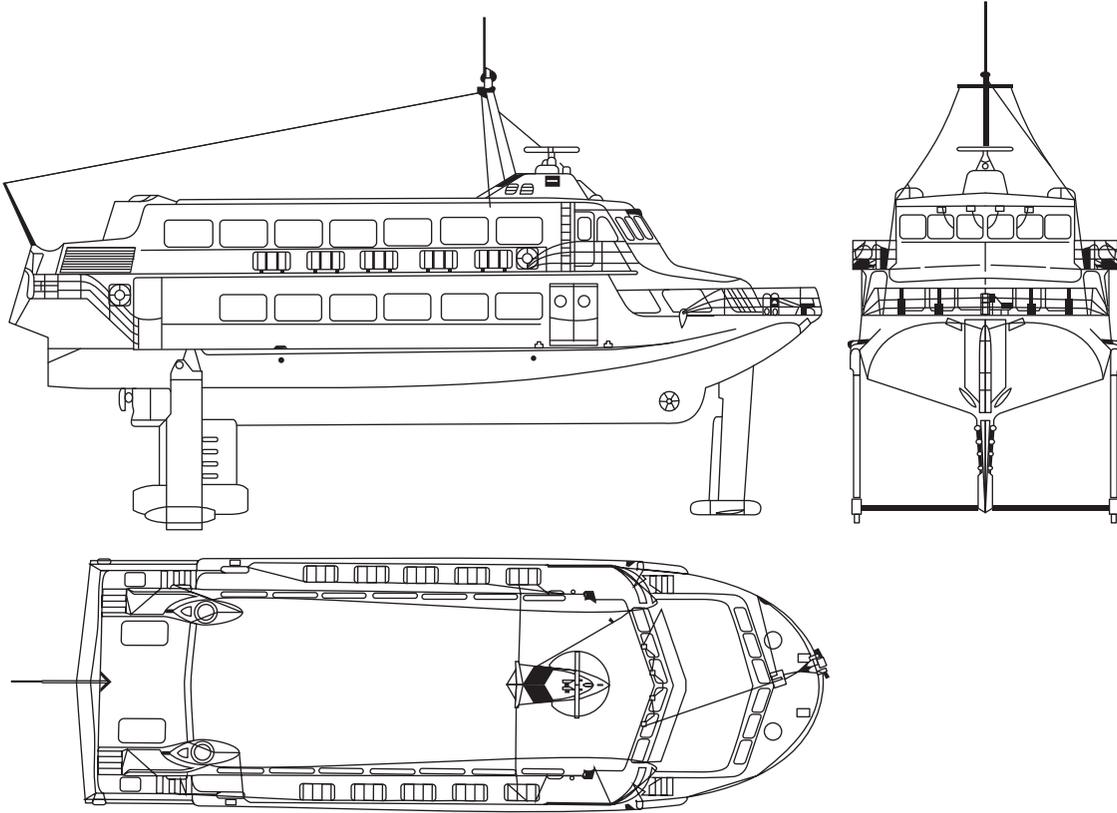


FIGURE 1: Fully submerged hydrofoil vessel.

$b = \begin{bmatrix} K_{\delta_R} & K_{\delta_A} \\ N_{\delta_R} & N_{\delta_A} \end{bmatrix}$ is the control moment coefficient matrix subjected to the control surface, and $u = [\delta_R, \delta_A]^T$ is the control input, where δ_R and δ_A represent the rudder angle and aileron angle of the hydrofoil system, respectively. As for the modeling of the FSHV, it is reasonable to select the origin point of the body-fixed frame at the center of gravity, thus yielding $x_g = y_g = z_g = 0$.

Considering the course keeping problem of the FSHV, the first equation in (1) can be simplified as $\dot{\eta} = v$. By defining $x_1 \triangleq \eta$, $x_2 \triangleq v$, the steering model of the FSHV can be regarded as a second-order system as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= F(x_1, x_2) + \xi(x_1, x_2) + \bar{b}u + \bar{\tau}_d, \end{aligned} \quad (2)$$

where

$$\begin{aligned} F(x_1, x_2) &= -M^{-1} [C(u_0, x_2)x_2 + D_L(u_0, x_2)x_2 + G(x_1)x_1], \\ \xi(x_1, x_2) &= -M^{-1}D_N(u_0, x_2)x_2, \\ \bar{b} &= M^{-1}b, \\ \bar{\tau}_d &= M^{-1}\tau_d. \end{aligned} \quad (3)$$

If the model uncertainties are considered in the modeling and control of the FSHV, the following notations are introduced:

$$\begin{aligned} F(x_1, x_2) &= F_0(x_1, x_2) + F_\Delta(x_1, x_2), \\ \xi(x_1, x_2) &= \xi_0(x_1, x_2) + \xi_\Delta(x_1, x_2), \end{aligned} \quad (4)$$

where subscript 0 denotes the nominal part of the corresponding matrix and the subscript Δ represents the perturbed part of the system dynamics. The yaw/roll dynamics of the FSHV with model uncertainties can be rewritten as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= F_0(x_1, x_2) + \xi_0(x_1, x_2) + \bar{b}u + f_d(t), \end{aligned} \quad (5)$$

where $f_d(t) = F_\Delta(x_1, x_2) + \xi_\Delta(x_1, x_2) + \bar{\tau}_d$ is the generalized disturbances of the system.

Remark 1. When ships sail at a fixed speed, $M(\cdot)$, $C(\cdot)$, $D_L(\cdot)$, and $G(\cdot)$ are linear matrices with constant element parameters, so $F_0(x_1, x_2)$ is also a linear function, while $\zeta_0(x_1, x_2)$ represents the nonlinear damping of the coupling hydrodynamics of the FSHV.

Hence, the control objective is to design an observer based output feedback controller for the course keeping of the FSHV using an iterative learning approach. The control structure of the system is summarized in Figure 2. For the

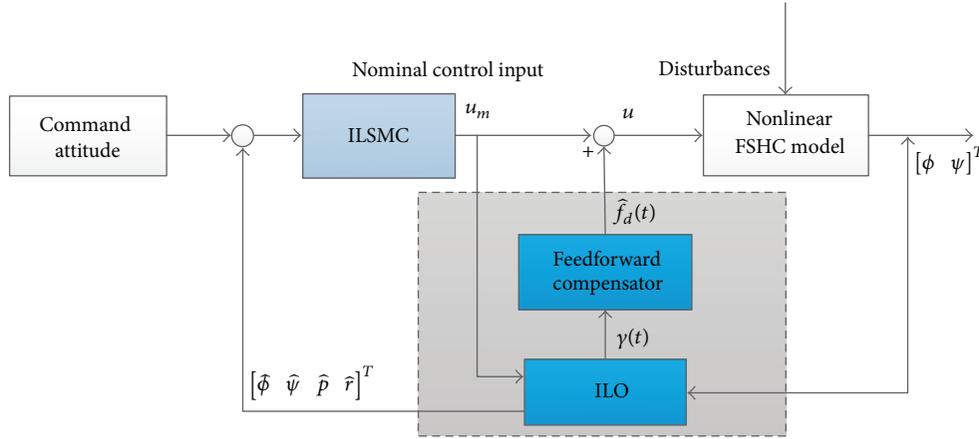


FIGURE 2: Control structure for the course keeping of FSHV.

subsequent analysis, the following assumptions are carried out for convenience.

Assumption 2 (see [25–28, 37]). The nonlinear function $\xi(\cdot)$ is continuously differentiable and satisfies Lipschitz condition with Lipschitz constant σ ; that is,

$$\|\xi(x) - \xi(\hat{x})\| \leq \sigma \|x - \hat{x}\|. \quad (6)$$

Remark 3. For marine vehicle systems, the high-order coupling hydrodynamics is sufficiently smooth. Therefore, it is reasonable that the nonlinear function $\xi(\cdot)$ satisfies Lipschitz condition.

Assumption 4. The disturbance $f_d(t)$ is bounded; namely, $\|f_d(t)\| \leq \bar{f}_d$.

3. Output Feedback Control Design of the FSHV

3.1. Iterative Learning Observer Design. In this section, an iterative learning observer (ILO) is established for estimation of system states as well as the disturbances. To facilitate the subsequent design, the system dynamics in (5) are rewritten into a generalized state space model as

$$\begin{aligned} \dot{x} &= Ax + \xi(x) + Bu + Ef_d(t) \\ y &= Cx, \end{aligned} \quad (7)$$

where $A = \begin{bmatrix} 0 & I \\ -M^{-1}G & -M^{-1}(C+D_I) \end{bmatrix}$, $\xi(x) = \begin{bmatrix} 0 \\ -M^{-1}D_N x_2 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ b \end{bmatrix}$, $C = [I \ 0]$, $E = \begin{bmatrix} 0 \\ I \end{bmatrix}$, and $x = [x_1 \ x_2]^T$, $y = x_1$.

Based on (7), an ILO is designed as

$$\begin{aligned} \hat{x} &= A\hat{x} + \xi(\hat{x}) + Bu(t) + L(y - \hat{y}) + E\gamma(t) \\ \gamma(t) &= \mu_1\gamma(t - \tau) + \mu_2[y(t) - \hat{y}(t)] \\ \hat{y}(t) &= C\hat{x}, \end{aligned} \quad (8)$$

where \hat{x} is the estimation value of x , $\hat{y}(t)$ is the estimation of system output at time t , τ is the sampling time interval, γ is

called ILO input, $\gamma(t)$ is the current ILO input and $\gamma(t - \tau)$ is the ILO input at the last sampling period. L and μ_i ($i = 1, 2$) are two gain matrices with appropriate dimensions to be determined.

Remark 5. It can be seen from (8) that if the gain matrix μ_1 is selected as a zero matrix, the ILO will degenerate into a conventional Luenberger observer. Therefore, zero matrix should be avoided for the gain tuning in order to maintain the capability of the estimation for the generalized disturbance.

Remark 6. It is necessary to clarify that the ILO input $\gamma(t)$ proposed in (8) is not a discrete-time equation since the sampling time τ can be chosen as any value theoretically. τ used in the control design in Part B also matches this instruction, which can be further indicated by the Lyapunov function in stability analysis.

Remark 7. Compared with the conventional adaptive state observers, the proposed ILO has several advantages. It is able to estimate both system states and time-varying disturbance at the same time. Additionally, persistence of excitation (PE) is not required, which is necessary for the adaptive observers to update the estimation law online.

Subtracting (8) from (7) results in the estimation error dynamics as

$$\dot{\tilde{x}} = (A - LC)\tilde{x} + [\xi(x) - \xi(\hat{x})] + E[f_d(t) - \gamma(t)], \quad (9)$$

where $\tilde{x} \triangleq x - \hat{x}$ is the estimate error.

The disturbance estimation error \tilde{d} and an auxiliary variable d_τ are defined as follows:

$$\begin{aligned} \tilde{d} &= f_d(t) - \gamma(t), \\ d_\tau &= f_d(t) - \mu_1 f_d(t - \tau), \end{aligned} \quad (10)$$

and then it yields

$$\begin{aligned} \tilde{d}(t) &= d(t) - \mu_1\gamma(t - \tau) - \mu_2C\tilde{x}(t) \\ &= \mu_1\tilde{d}(t - \tau) - \mu_2C\tilde{x}(t) + d_\tau. \end{aligned} \quad (11)$$

By substituting (11) into (9), it yields

$$\begin{aligned} \dot{\tilde{x}} &= (A - LC)\tilde{x} + E\mu_1\tilde{d}(t - \tau) - E\mu_2C\tilde{x}(t) + Ed_\tau \\ &+ [\xi(x) - \xi(\hat{x})]. \end{aligned} \quad (12)$$

Theorem 8. *Given the system dynamics in (7) with the proposed ILO designed in (8), the state estimate error is bounded if the following conditions hold:*

$$\begin{aligned} (A - LC)^T P + P(A - LC) &= -Q, \\ PE &= \rho(\mu_2C)^T, \quad \rho > 1, \\ 0 &< (\rho + m^2)\mu_1^T\mu_1 \leq I. \end{aligned} \quad (13)$$

Proof. Define the following Lyapunov function candidate:

$$V_1 = \tilde{x}^T(t)P\tilde{x}(t) + \int_{t-\tau}^t \tilde{d}^T(\alpha)\tilde{d}(\alpha)d\alpha, \quad (14)$$

where P is a positive definite symmetric matrix.

Based on (9) and (12), differentiating V_1 with respect to time yields

$$\begin{aligned} \dot{V}_1 &= \tilde{x}^T \left[(A - LC)^T P + P(A - LC) \right] \tilde{x} \\ &+ 2\tilde{x}^T(t)PE\mu_1\tilde{d}(t - \tau) - 2\tilde{x}(t)PE\mu_2C\tilde{x}(t) \\ &+ 2\tilde{x}^T(t)PEd_\tau(t) - \varepsilon\tilde{d}^T(t)\tilde{d}(t) + \rho\tilde{d}^T(t)\tilde{d}(t) \\ &- \tilde{d}^T(t - \tau)\tilde{d}(t - \tau) + 2\tilde{x}^T(t)P[\xi(x) - \xi(\hat{x})], \end{aligned} \quad (15)$$

where ρ and ε are positive constants and $\rho - \varepsilon = 1$.

Substituting (11) into (15), we can obtain

$$\begin{aligned} \dot{V}_1 &= \tilde{x}^T \left[(A - LC)^T P + P(A - LC) \right] \tilde{x} \\ &+ 2\tilde{x}^T(t)PE\mu_1\tilde{d}(t - \tau) \\ &+ 2\tilde{x}^T(t)P[\xi(x) - \xi(\hat{x})] - 2\tilde{x}(t)PE\mu_2C\tilde{x}(t) \\ &+ 2\tilde{x}^T(t)PEd_\tau(t) - \varepsilon\tilde{d}^T(t)\tilde{d}(t) \\ &+ \rho\tilde{d}(t - \tau)\mu_1^T\mu_1\tilde{d}(t - \tau) + \rho d_\tau^T d_\tau \\ &+ 2\rho\tilde{d}^T(t - \tau)\mu_1^T d_\tau(t) - 2\rho\tilde{x}^T(\mu_2C)^T d_\tau(t) \\ &- \tilde{d}^T(t - \tau)\tilde{d}(t - \tau). \end{aligned} \quad (16)$$

For any $Q = Q^T > 0$, there exists $P = P^T > 0$ satisfying the following Riccati equation:

$$\begin{aligned} (A - LC)^T P + P(A - LC) &= -Q, \\ PE &= \rho(\mu_2C)^T. \end{aligned} \quad (17)$$

Then it yields

$$\begin{aligned} \dot{V}_1 &\leq -\lambda_{\min}(Q)\|\tilde{x}\|^2 - \varepsilon\|\tilde{d}(t)\|^2 - \tilde{d}^T(t - \tau)\tilde{d}(t - \tau) \\ &+ \rho\tilde{d}^T(t - \tau)\mu_1^T\mu_1\tilde{d}(t - \tau) \\ &+ 2\rho\tilde{d}(t - \tau)\mu_1^T d_\tau(t) + \rho\tilde{d}_\tau^2 + 2\eta\|P\|\|\tilde{x}\|^2. \end{aligned} \quad (18)$$

By using the inequality in

$$\begin{aligned} 2\rho\tilde{d}^T(t - \tau)\mu_1^T d_\tau(t) &\leq m^2\tilde{d}^T(t - \tau)\mu_1^T\mu_1\tilde{d}(t - \tau) \\ &+ \frac{\rho^2}{m^2}d_\tau^T(t)d_\tau(t), \end{aligned} \quad (19)$$

It follows that

$$\begin{aligned} \dot{V}_1 &\leq -\lambda_{\min}(Q)\|\tilde{x}\|^2 - \varepsilon\|\tilde{d}(t)\|^2 \\ &+ \tilde{d}^T(t - \tau)\left[(\rho + m^2)\mu_1^T\mu_1 - I\right]\tilde{d}(t - \tau) \\ &+ \left(\rho + \frac{\rho^2}{m^2}\right)\tilde{d}_\tau^2 + 2\eta\lambda_{\max}(P)\|\tilde{x}\|^2; \end{aligned} \quad (20)$$

namely,

$$\begin{aligned} \dot{V}_1 &\leq -k\|\tilde{x}\|^2 - \varepsilon\|\tilde{d}(t)\|^2 + \left(\rho + \frac{\rho^2}{m^2}\right)\|\tilde{d}_\tau\|^2 \\ &+ \tilde{d}^T(t - \tau)\left((\rho + m^2)\mu_1^T\mu_1 - I\right)\tilde{d}(t - \tau), \end{aligned} \quad (21)$$

where $k = \lambda_{\min}(Q) - 2\eta\lambda_{\max}(P) > 0$.

If $0 < (\rho + m^2)\mu_1^T\mu_1 \leq I$ holds, the estimation error as well as the generalized disturbances is uniformly ultimately bounded (UUB) [38]. \square

3.2. Observer Based Iterative Sliding Mode Controller Design.

As to the controller design, an ILO based sliding mode controller is proposed for the output feedback course keeping control problem of the FSHV based on an iterative learning sliding surface. Before the controller design, we first make the following transformation.

By utilizing the feedback linearization method and replacing the system states with the estimated values, the system model in (2) can be rewritten as

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2, \\ \dot{\hat{x}}_2 &= F_0(\hat{x}_1, \hat{x}_2) + \chi + f_d(t), \end{aligned} \quad (22)$$

where χ is a virtual control variable defined as

$$\chi = \xi_0(\hat{x}_1, \hat{x}_2) + \bar{b}u. \quad (23)$$

Based on the proposed ILO, the estimate of $f_d(t)$ can be obtained as $\hat{f}_d = \gamma$.

By using \hat{f}_d as a feedforward compensator, the virtual control can be designed as $\chi = \chi_s - \hat{f}_d$. Then it yields

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 \\ \dot{\hat{x}}_2 &= F_0(\hat{x}_1, \hat{x}_2) + \chi_s + e_d, \end{aligned} \quad (24)$$

where e_d is the residual error of \hat{f}_d and $|e_d| \leq \bar{e}_d$.

Based on (24), the iterative learning sliding mode controller can be designed as follows.

Define a novel sliding surface with an iterative item in the following form:

$$s = \hat{x}_2 - K\hat{x}_1 - K_\tau \hat{x}_1(t - \tau), \quad (25)$$

where $K, K_\tau \in R^{2 \times 2}$ are sliding parameters to be designed.

Remark 9. According to (25), it is clear that if the time interval τ is chosen as $\tau = 0$, the sliding surface will become $s = \hat{x}_2 - (K + K_\tau)\hat{x}_1$, which is actually a conventional linear sliding surface. Hence, the existence of τ implies the improvement of the dynamic performance of the iterative learning sliding surface.

Let $s = 0$, then it yields

$$\dot{\hat{x}}_1 = K\hat{x}_1 + K_\tau \hat{x}_1(t - \tau). \quad (26)$$

Define the Lyapunov function V_2 as

$$V_2 = \hat{x}_1^T \Phi \hat{x}_1 + \int_{t-\tau}^t \hat{x}_1^T(\omega) Z \hat{x}_1(\omega) d\omega, \quad (27)$$

where Φ, Z are positive definite matrices.

Differentiating V_2 with respect to time yields

$$\begin{aligned} \dot{V}_2 &= \dot{\hat{x}}_1^T \Phi \hat{x}_1 + \hat{x}_1^T \Phi \dot{\hat{x}}_1 + \hat{x}_1^T(t) Z \hat{x}_1(t) \\ &\quad - \hat{x}_1^T(t - \tau) Z \hat{x}_1(t - \tau). \end{aligned} \quad (28)$$

Substituting (26) into (28), we can obtain

$$\begin{aligned} \dot{V}_2 &= \hat{x}_1^T (K^T \Phi + \Phi K) \hat{x}_1 + \hat{x}_1^T \Phi K_\tau \hat{x}_1(t - \tau) \\ &\quad + \hat{x}_1^T(t - \tau) K_\tau^T \Phi \hat{x}_1(t - \tau) + \hat{x}_1^T Z \hat{x}_1 \\ &\quad - \hat{x}_1^T(t - \tau) Z \hat{x}_1(t - \tau). \end{aligned} \quad (29)$$

Denote $\Lambda \triangleq [\hat{x}_1^T(t) \quad \hat{x}_1^T(t - \tau)]^T$; then (29) can be rewritten as

$$\dot{V}_2 = \Lambda^T \begin{bmatrix} K^T \Phi + \Phi K + Z & \Phi K_\tau \\ \Phi K_\tau & -Z \end{bmatrix} \Lambda. \quad (30)$$

By appropriately tuning K, K_τ, Φ , and Z , there exists a positive definite matrix Γ such that

$$\begin{bmatrix} K^T \Phi + \Phi K + Z & \Phi K_\tau \\ \Phi K_\tau & -Z \end{bmatrix} = -\Gamma. \quad (31)$$

Then it yields

$$\dot{V}_2 = -\Lambda^T \Gamma \Lambda \leq -\lambda_{\min}(\Gamma) \|\Lambda\|^2, \quad (32)$$

which indicates that the iterative learning sliding motion (25) is asymptotically stable. Therefore, once the system trajectory reaches the sliding surface, it can be kept at the sliding surface.

Based on the proposed sliding surface, the control law χ_s can be designed as

$$\begin{aligned} \chi_s &= -[F_0(\hat{x}_1, \hat{x}_2) - K\hat{x}_2 - K_\tau \hat{x}_2(t - \tau) + \alpha s] \\ &\quad - \bar{e}_d \operatorname{sgn}(s), \end{aligned} \quad (33)$$

where α is a positive scalar.

Then the final control input u is obtained as

$$u = \bar{b}^{-1} [\chi_s(\hat{x}_1, \hat{x}_2) - \hat{f}_d - \xi_0(x_1, x_2)]. \quad (34)$$

Therefore, the iterative learning sliding mode controller for the course keeping of the FSHV in an output feedback scheme is implemented.

4. Stability Analysis

The closed-loop stability of the observer-controller structure of system (5) is analyzed in this part.

Theorem 10. *Given the system model for the course keeping control of the FSHV in (5), with the ILO proposed in (8) and the feedback linearization based iterative learning sliding mode controller in (34), uniformly ultimate boundedness can be guaranteed for the closed-loop system at the equilibrium point.*

Proof. The Lyapunov function of the closed-loop system is selected as

$$V_3 = V_1 + \frac{1}{2} s^T s. \quad (35)$$

Differentiating V_3 with respect to time, we can obtain

$$\begin{aligned} \dot{V}_3 &= \tilde{x}^T [(A - LC)^T P + P(A - LC)] \tilde{x} \\ &\quad + 2\tilde{x}^T(t) PE\mu_1 \tilde{d}(t - \tau) - 2\tilde{x}(t) PE\mu_2 C\tilde{x}(t) \\ &\quad + 2\tilde{x}^T(t) PE d_\tau(t) - \tilde{e}^T(t) \tilde{d}(t) + \rho \tilde{d}^T(t) \tilde{d}(t) \\ &\quad - \tilde{d}^T(t - \tau) \tilde{d}(t - \tau) + 2\tilde{x}^T(t) P [\xi(x) - \xi(\tilde{x})] \\ &\quad + s^T \dot{s}. \end{aligned} \quad (36)$$

Based on the results of (21), (24), and (25), substituting the control law (34) into (36), we have

$$\begin{aligned} \dot{V}_3 &\leq -k \|\tilde{x}\|^2 - \varepsilon \|\tilde{d}(t)\|^2 + \left(\rho + \frac{\rho^2}{m^2}\right) \bar{d}_\tau^2 \\ &\quad + \tilde{d}^T(t - \tau) ((\rho + m^2) \mu_1^T \mu_1 - I) \tilde{d}(t - \tau) \\ &\quad + s [e_d - \bar{e}_d \operatorname{sgn}(s) - \alpha s] \end{aligned} \quad (37)$$

$$\begin{aligned} \dot{V}_3 &\leq -k \|\tilde{x}\|^2 - \varepsilon \|\tilde{d}(t)\|^2 + \left(\rho + \frac{\rho^2}{m^2}\right) \bar{d}_\tau^2 \\ &\quad + \tilde{d}^T(t - \tau) ((\rho + m^2) \mu_1^T \mu_1 - I) \tilde{d}(t - \tau) \\ &\quad + \|s\| \|e_d\| - \bar{e}_d s \operatorname{sgn}(s) - \alpha \|s\|^2 \end{aligned} \quad (38)$$

$$\begin{aligned} \dot{V}_3 &\leq -k \|\tilde{x}\|^2 - \varepsilon \|\tilde{d}(t)\|^2 - \alpha \|s\|^2 + \left(\rho + \frac{\rho^2}{m^2}\right) \bar{d}_\tau^2 \\ &\quad + \tilde{d}^T(t - \tau) ((\rho + m^2) \mu_1^T \mu_1 - I) \tilde{d}(t - \tau). \end{aligned} \quad (39)$$

TABLE 1: Model parameters of the FSHV.

Parameter	Value	SI-unit
u_0	23.15	m/s
m	$2.62 * 10^5$	kg
I_x	$2.59 * 10^5$	kgm ²
I_z	$1.47 * 10^7$	kgm ²
\overline{GM}_T	0.025	m
K_p	$-13.0354 * 10^5$	kgm ² /s
K_r	$-2.4864 * 10^5$	kgm ² /s
$K_{rr\phi}$	$-19.0883 * 10^5$	kgm ² /s
$K_{r\phi\phi}$	$12.5615 * 10^5$	kgm ² /s
N_p	$-0.2182 * 10^7$	kgm ² /s
N_r	$-1.3818 * 10^7$	kgm ² /s
$N_{rr\phi}$	$0.9261 * 10^7$	kgm ² /s
$N_{r\phi\phi}$	$-1.5141 * 10^7$	kgm ² /s
K_{δ_R}	-1.046	kgm ² /s
K_{δ_A}	5.84	kgm ² /s
N_{δ_R}	0.3925	kgm ² /s
N_{δ_A}	-0.2308	kgm ² /s

Together with the results of Theorem 8, if the conditions

$$\begin{aligned}
 (A - LC)^T P + P(A - LC) &= -Q, \\
 PE &= \rho(\mu_2 C)^T, \quad \rho > 1, \\
 0 < (\rho + m^2) \mu_1^T \mu_1 &\leq I, \quad (40)
 \end{aligned}$$

$$\begin{bmatrix} K^T \Phi + \Phi K + Z & \Phi K_r \\ \Phi K_r & -Z \end{bmatrix} = -\Gamma$$

hold, the equilibrium point of system (5) is uniformly ultimately bounded. \square

5. Simulation

In this section, a mathematical model of a FSHV [2, 3] is applied to validate the performance of the proposed approach. The physical parameters of the FSHV are given in Table 1.

The nominal system model with norm-bounded perturbation items is used to simulate the model uncertainties; namely, $p_{\Delta jj} = p_{jj} + 0.15p_{jj} \cdot \text{rand}(-1, 1)$, where $p_{\Delta jj}$ is the parameter with perturbation and p_{jj} is the nominal model parameter of (1). To simulate the wave disturbance, the Pierson-Moskowitz (PM) spectrum is selected as the standard wave energy spectrum [39], which can be expressed as

$$S(\omega) = A\omega^{-5} \exp(-B\omega^{-4}), \quad (41)$$

where $A = 8.1 \cdot 10^{-3} g^2$, $B = 3.11H_{1/3}^{-2}$, and $H_{1/3}$ is the significant wave height [37].

When a marine vessel is sailing in the open sea at a fixed speed, the encounter frequency is

$$\omega_e = \omega - \frac{\omega^2}{gU \cos \beta}, \quad (42)$$

where U is the sailing speed and β is the wave-to-course angle. So the wave energy spectrum related to the encounter frequency is

$$S(\omega_e) = \frac{S(\omega)}{1 - 2\omega U \cos \beta / g}. \quad (43)$$

According to the strip theory and equivalent energy division method, the disturbance moments of the yaw and roll dynamics can be calculated.

To verify the effectiveness of the proposed control scheme, the following control methodologies are demonstrated for comparison.

- (1) Iterative learning observer based sliding mode controller (ILSMC): this is the observer based sliding mode control approach using iterative learning strategy proposed in the previous sections. In order to reduce the chattering phenomenon, a saturated function $\text{sat}(\cdot)$ is utilized instead of $\text{sgn}(\cdot)$. The control parameters are listed as follows:

$$\begin{aligned}
 L &= \begin{bmatrix} 11.58 & -2.85 \\ -2.75 & 22.94 \\ 55.47 & 57.26 \\ 75.91 & 54.89 \end{bmatrix}, \\
 \mu_1 &= \text{diag}[0.97 \quad 0.78], \\
 \mu_2 &= \begin{bmatrix} 127.31 & 58.11 \\ 52.15 & 49.82 \end{bmatrix}, \\
 K &= \begin{bmatrix} 11.12 & 3.14 \\ 1.73 & 2.11 \end{bmatrix}, \\
 K_r &= \begin{bmatrix} 4.54 & 1.71 \\ 0.63 & 0.49 \end{bmatrix}, \\
 \alpha &= 0.35, \\
 \bar{e}_d &= 0.07.
 \end{aligned} \quad (44)$$

- (2) Iterative learning observer with standard sliding mode controller (ILOSSMC): in this method, a standard sliding surface is used instead of the iterative learning sliding manifold to verify the advantage of the proposed control approach. In the ILOSSMC, $K_r = 0$, and the other parameters are the same as the ILSMC.
- (3) Linear extended state observer with standard sliding mode controller (LESOSSMC): in this method, the system model is linearized by an auxiliary feedback

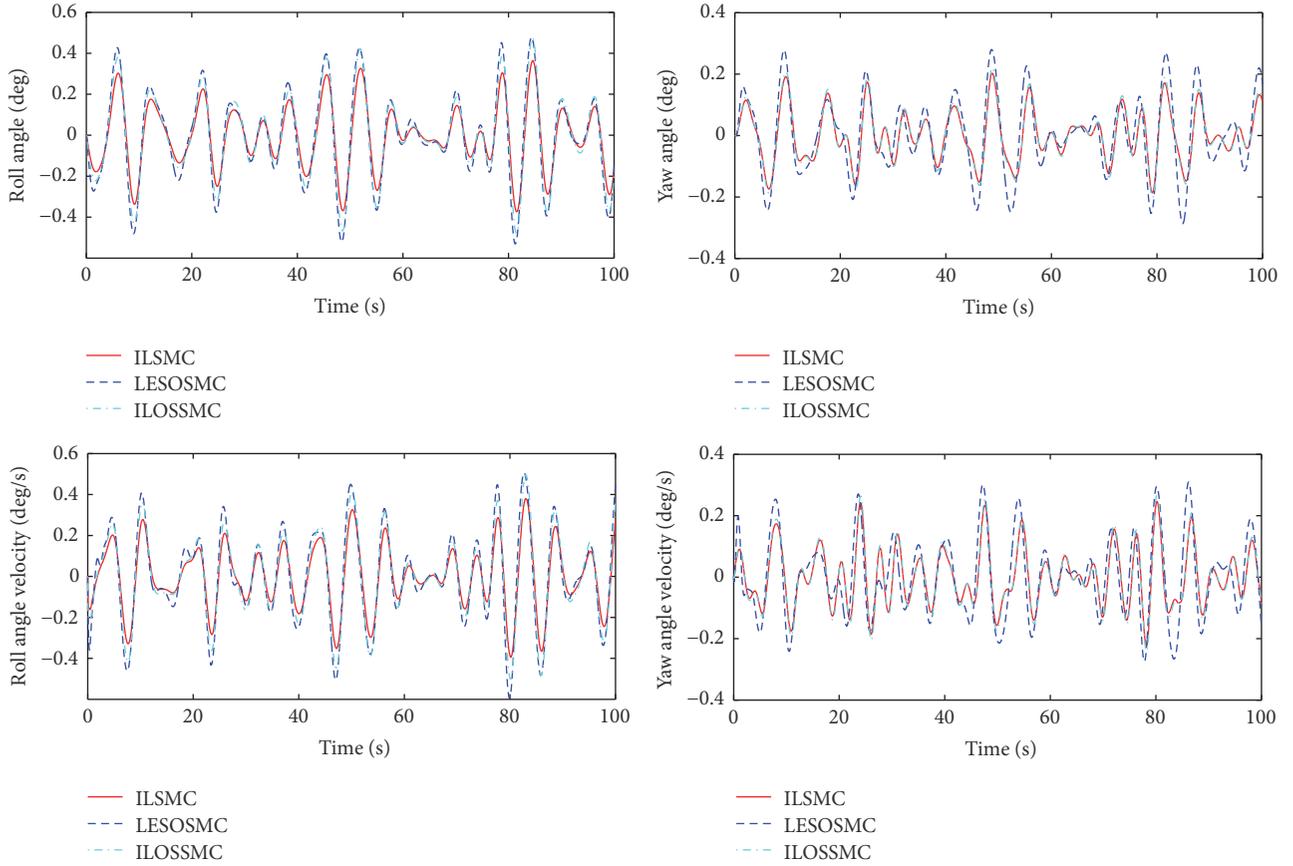


FIGURE 3: Stabilization effect of system states (Case 1).

linearization approach. Based on this approach, a linear extended state observer (LESO) is designed to estimate system states as well as the generalized disturbances. The sliding mode controller design is similar to the ILOSSMC. The LESO is designed as follows:

$$\begin{aligned}
 e_1 &= x_1 - \hat{z}_1, \\
 \dot{\hat{z}}_1 &= \hat{z}_2 + l_1 e_1, \\
 \dot{\hat{z}}_2 &= F_0(\hat{z}_1, \hat{z}_2) + \chi + z_3 + l_2 e_1, \\
 \dot{\hat{z}}_3 &= l_3 e_1,
 \end{aligned} \tag{45}$$

where x_1 is the system output, \hat{z}_1, \hat{z}_2 are the estimate values of x_1, x_2 , and z_3 is the extended state for the estimate of the generalized disturbances. The parameters of the LESO are selected as follows: $l_1 = 54$, $l_2 = 189$, and $l_3 = 3472$.

The above control schemes are tested individually for the course keeping of the FSHV in two typical cases, namely, Case 1 (sea state 3, $H_{1/3} = 1.5$ m) and Case 2 (sea state 5, $H_{1/3} = 3.8$ m). Figures 3 and 4 show the stabilization of roll and yaw dynamics based on three control methods on the condition $H_{1/3} = 1.5$ m and $H_{1/3} = 3.8$ m, respectively.

The rudder angle and aileron angle of the two cases are presented in Figures 5 and 6. From Figures 3–6, we can see that based on the online iterative optimization, the stabilization performance of the ILSMC is better than that of the LESOSMC against wave disturbances, and the system consumption of the ILSMC does not increase. Comparing the ILSMC with the ILOSSMC, we can see that the iterative learning item in the sliding manifold (25) plays an important role in the system stabilization. Utilizing the learning item, the system control can be automatically regulated according to the tendency of the system trajectory, which decreases the conservation in control design.

As to the state estimate, Figures 7 and 8 show the real system state and the state estimate of the ILO and the LESO in Case 1, and Figures 9 and 10 present the system estimate in Case 2. It can be shown from Figures 7–10 that the estimate accuracy of the ILO is higher than that of the LESO. Because of the high-gain character of the LESO, the peak phenomenon will occur if the observer gain selected is too large, while this problem will not happen in the ILO. Because of the design of $\gamma(t)$ in the ILO, the estimating values can be updated dynamically according to the sample time τ , which enhances the state estimate precision for the system. In addition, $\gamma(t)$ can detect the generalized disturbances and compensate them in the feedforward loop. The comparison of the disturbance estimate between the ILO and the LESO

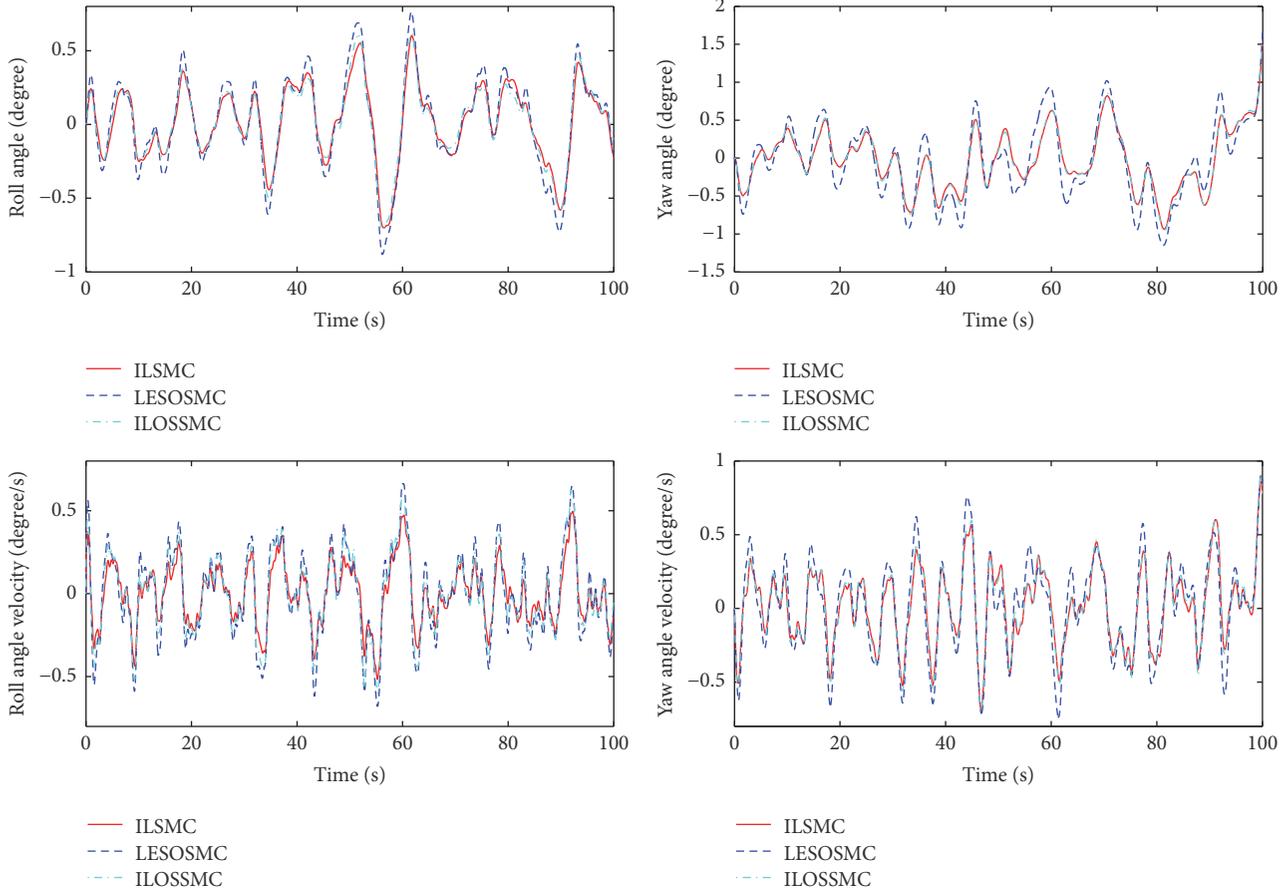


FIGURE 4: Stabilization effect of system states (Case 2).

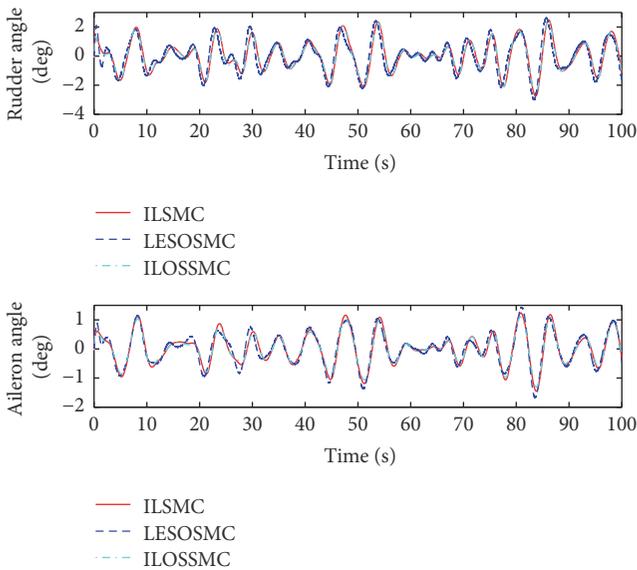


FIGURE 5: Control inputs of rudder and aileron (Case 1).

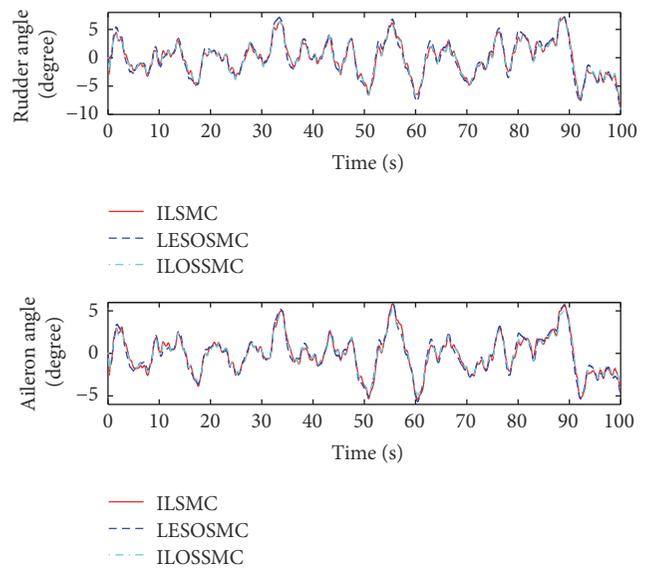


FIGURE 6: Control inputs of rudder and aileron (Case 2).

is illustrated in Figure 11 (Case 1) and Figure 12 (Case 2). It is illustrated that with the increase of the disturbances, the gaps in disturbances observing of the two methods become

more obvious, which signifies the advantages of the ILO in severely perturbed systems. Using fixed observer gain, the ILO represents a self-adaptation property in state estimate,

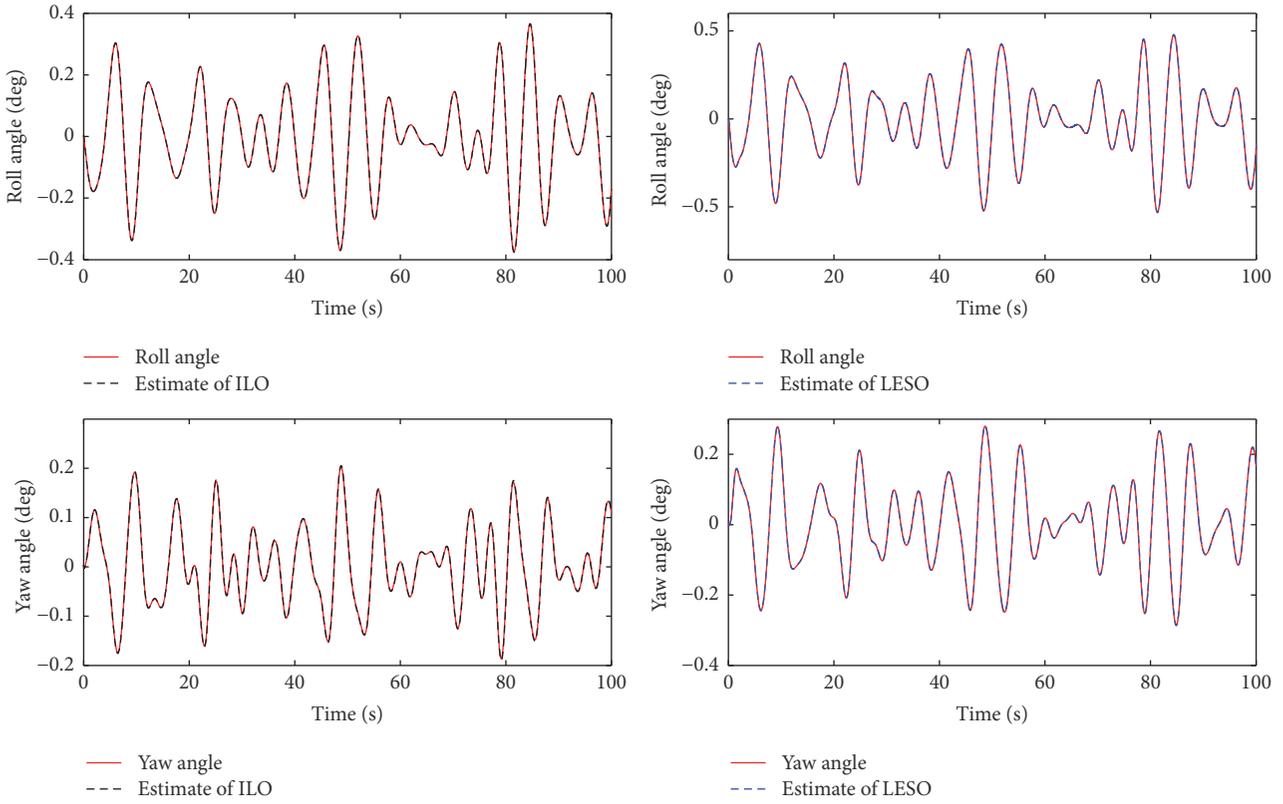


FIGURE 7: Roll/yaw angle estimate performance (Case 1).

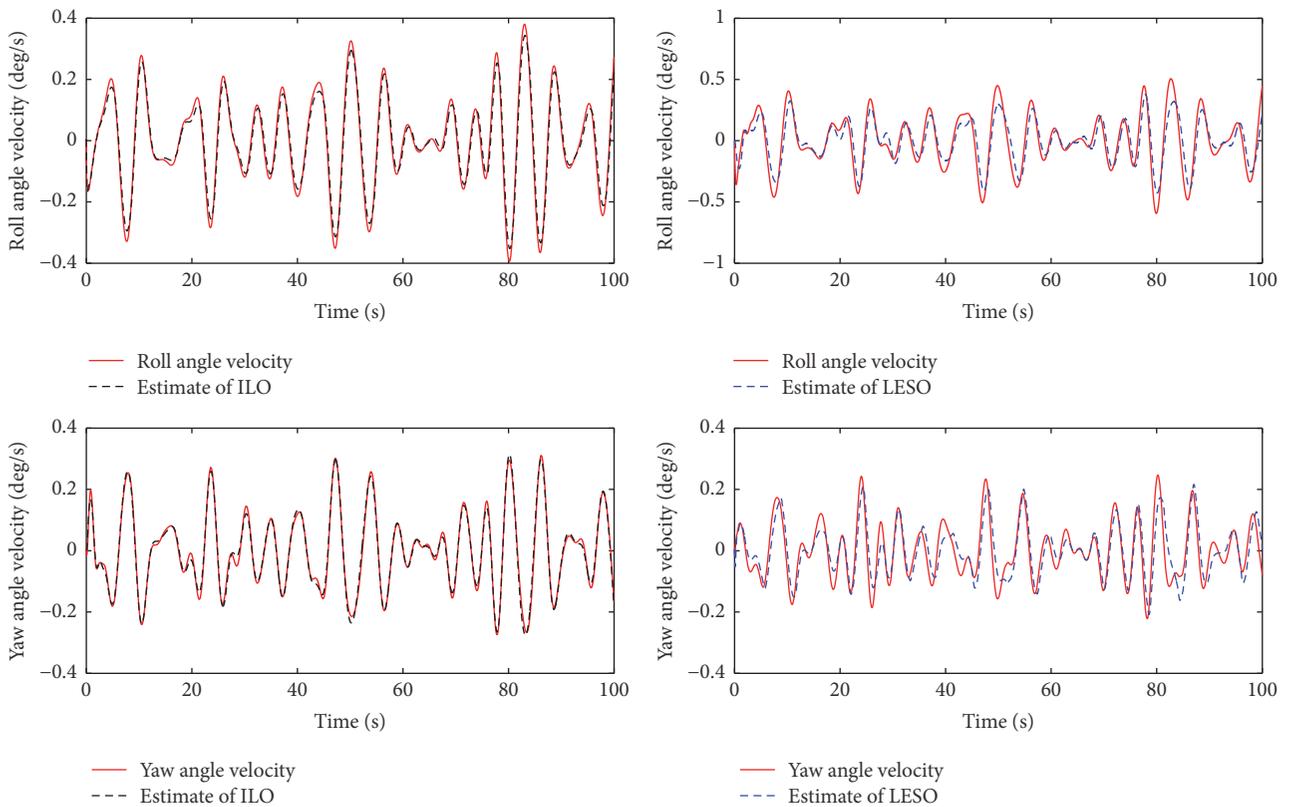


FIGURE 8: Angle velocity estimate performance (Case 1).

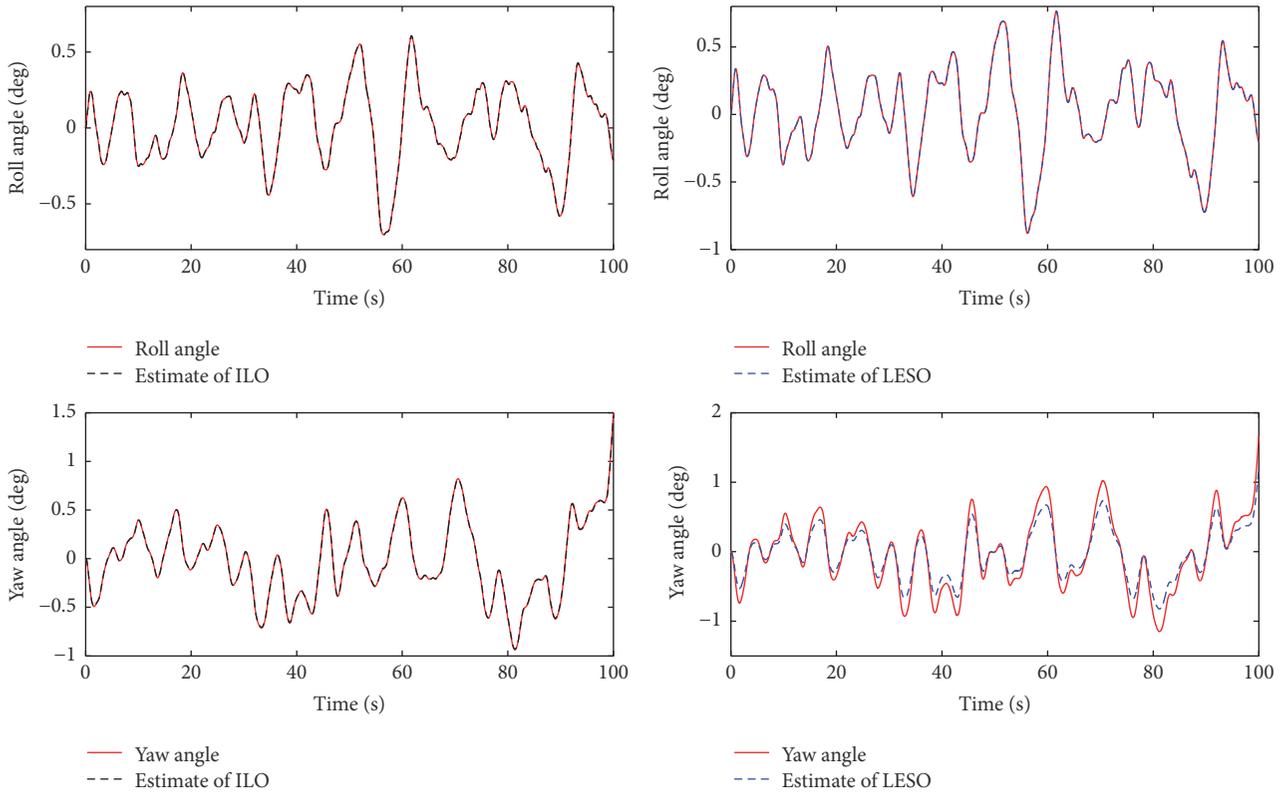


FIGURE 9: Roll/yaw angle estimate performance (Case 2).

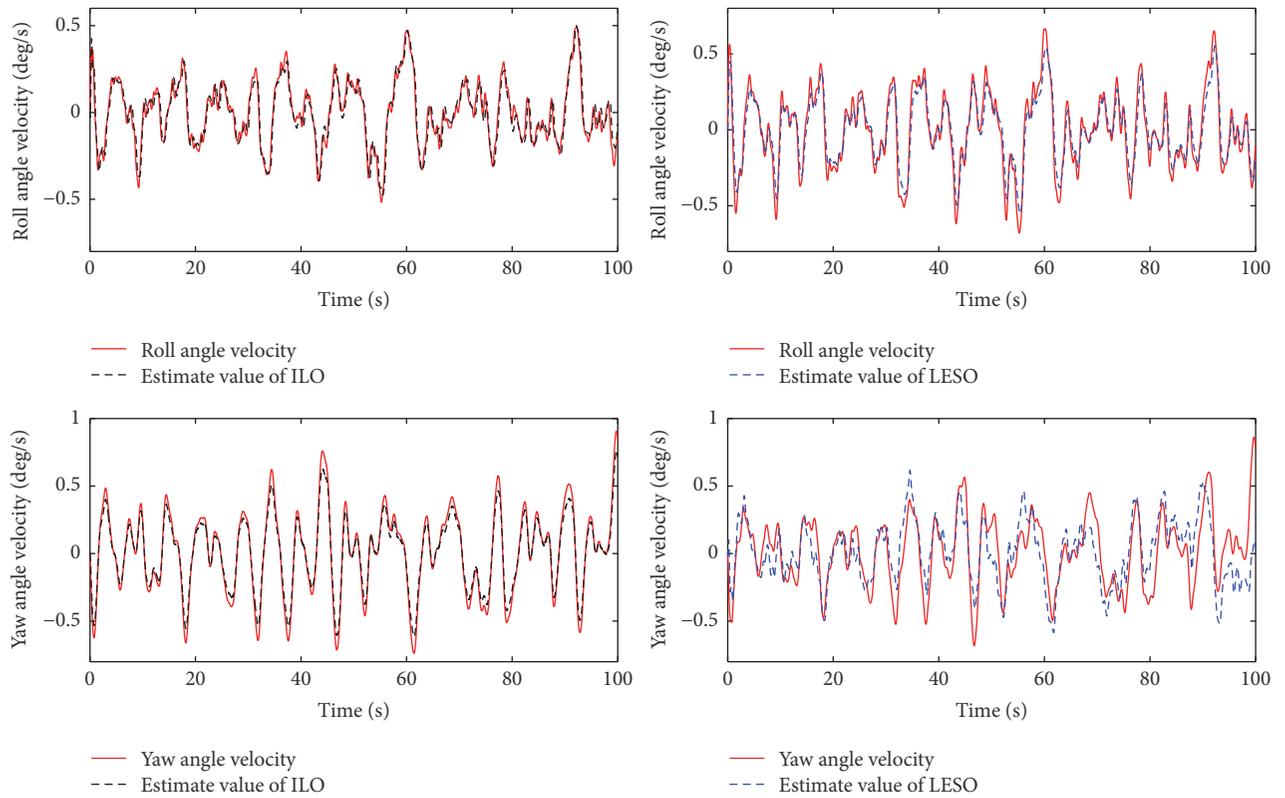


FIGURE 10: Angle velocity estimate performance (Case 2).

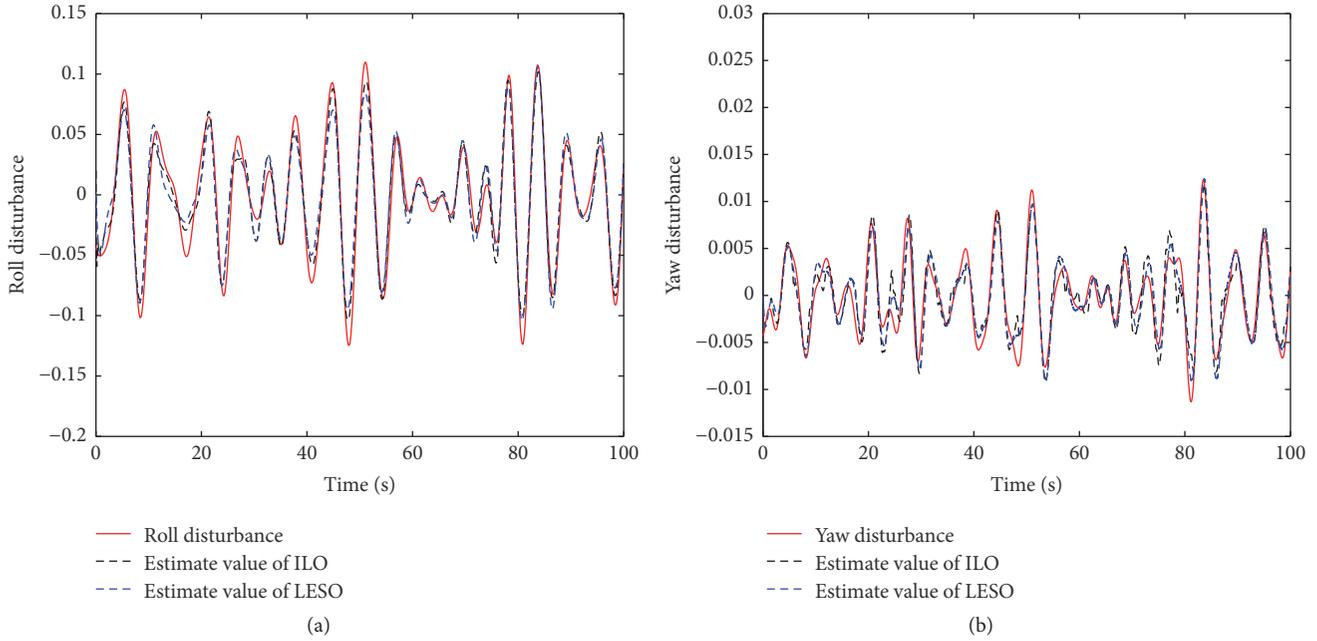


FIGURE 11: Disturbance estimate performance (Case 1): (a) the roll disturbance estimate; (b) the yaw disturbance estimate.

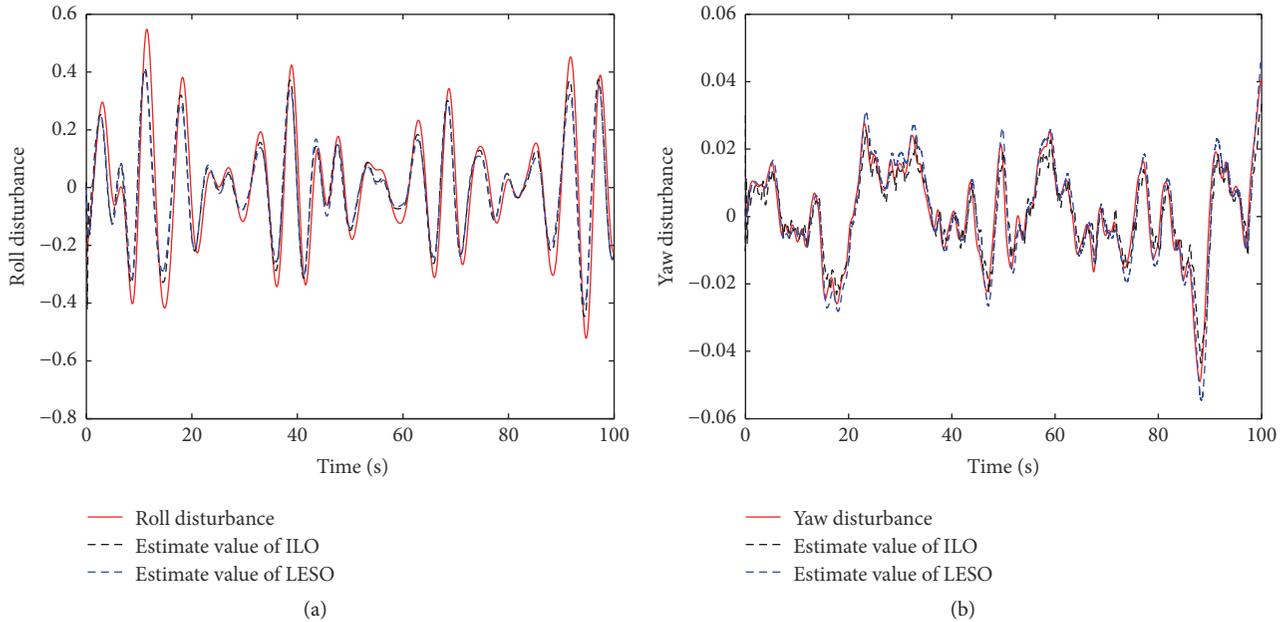


FIGURE 12: Disturbance estimate performance (Case 2): (a) the roll disturbance estimate; (b) the yaw disturbance estimate.

which achieves a better disturbance rejection performance than the LESO. The root mean square errors (RMSE) of the above state observers are listed in Table 2.

6. Conclusion

In this paper, an output feedback control scheme is investigated for the course keeping control of the fully submerged hydrofoil vessel based on an iterative learning strategy. An iterative learning observer is illustrated for the estimation of

system states as well as the generalized disturbances. Then an iterative sliding model controller is established for the attitude stabilization of the fully submerged hydrofoil vessel. Based on the ILO, the disturbances can be estimated through the inner loop, and the iterative sliding mode controller attenuates the residual error of the estimated disturbances in the outer loop. Based on Lyapunov stability theory, uniform ultimate boundedness of the overall closed-loop system is guaranteed. With the intervention of the iterative item, performance of the transient states can be improved for both state observer

TABLE 2: Estimate performance comparison of the observers.

RMSE		$\tilde{\phi}$	$\tilde{\psi}$	\tilde{p}	\tilde{r}	$\tilde{\tau}_{droll}$	$\tilde{\tau}_{dyaw}$
Case 1	ILO	0.164	0.088	0.161	0.092	0.048	0.004
	LESO	0.215	0.122	0.219	0.126	0.049	0.006
Case 2	ILO	0.258	0.184	0.187	0.161	0.061	0.012
	LESO	0.317	0.298	0.258	0.214	0.063	0.013

and controller. Comparative simulation results illustrate that the proposed approach achieves better estimate accuracy and disturbance rejection performance. The system conservation also decreases compared with conventional approaches. In future work, time-varying sampling time is to be considered based on the time scale and frequency characteristic of the system. And the control objective for the FSHV will extend to the kinematic level with goals such as path following and trajectory tracking.

Conflicts of Interest

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

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant 51579047.

References

- [1] Y. Saito and T. Ikebuchi, "Fully submerged hydrofoil vessel," in *Proceedings of the 7th Marine Dynamics Symposium on Prediction of Seakeeping Qualities of High-Speed Vessel*, pp. 107–141, 1990.
- [2] H. Imamura, Y. Saito, and M. Asao, "Automatic control system for jetfoil," *Kawasaki Technical Review*, no. 107, pp. 1–9, 1990.
- [3] S. Liu, C. Xu, and Y. Wang, "Disturbance rejection control for the course keeping of the fully-submerged hydrofoil craft," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 747–751, IEEE, Chengdu, China, July 2016.
- [4] S.-H. Kim and H. Yamato, "On the design of a longitudinal motion control system of a fully-submerged hydrofoil craft based on the optimal preview servo system," *Ocean Engineering*, vol. 31, no. 13, pp. 1637–1653, 2004.
- [5] K. D. Do and J. Pan, "State- and output-feedback robust path-following controllers for underactuated ships using Serret-Frenet frame," *Ocean Engineering*, vol. 31, no. 5-6, pp. 587–613, 2004.
- [6] L. J. Zhang, H. M. Jia, and X. Qi, "NNFFC-adaptive output feedback trajectory tracking control for a surface ship at high speed," *Ocean Engineering*, vol. 38, no. 13, pp. 1430–1438, 2011.
- [7] S.-H. Kim and H. Yamato, "An experimental study of the longitudinal motion control of a fully submerged hydrofoil model in following seas," *Ocean Engineering*, vol. 31, no. 5-6, pp. 523–537, 2004.
- [8] A. Rosales and I. Boiko, "Disturbance attenuation for systems with second-order sliding modes via linear compensators," *IET Control Theory & Applications*, vol. 9, no. 4, pp. 526–537, 2015.
- [9] Y. Feng, F. Han, and X. Yu, "Chattering free full-order sliding-mode control," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 50, no. 4, pp. 1310–1314, 2014.
- [10] J. D. Sánchez-Torres, A. Navarrete-Guzman, G. Rubio-Astorga, and A. G. Loukianov, "High order integral nested sliding mode control," in *Proceedings of the 13th International Workshop on Variable Structure Systems, VSS 2014*, IEEE, Nantes, France, July 2014.
- [11] S.-W. Ji, V. P. Bui, B. Balachandran, and Y.-B. Kim, "Robust control allocation design for marine vessel," *Ocean Engineering*, vol. 63, pp. 105–111, 2013.
- [12] J. Du, Y. Yang, D. Wang, and C. Guo, "A robust adaptive neural networks controller for maritime dynamic positioning system," *Neurocomputing*, vol. 110, pp. 128–136, 2013.
- [13] N. E. Kahveci and P. A. Ioannou, "Adaptive steering control for uncertain ship dynamics and stability analysis," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 49, no. 3, pp. 685–697, 2013.
- [14] P. M. Patre, W. MacKunis, M. Johnson, and W. E. Dixon, "Composite adaptive control for Euler-Lagrange systems with additive disturbances," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 46, no. 1, pp. 140–147, 2010.
- [15] H. Hammouri, G. Bornard, and K. Busawon, "High gain observer for structured multi-output nonlinear systems," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 55, no. 4, pp. 987–992, 2010.
- [16] L. Wang and J. Su, "Robust Disturbance Rejection Control for Attitude Tracking of an Aircraft," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2361–2368, 2015.
- [17] D. Ginoya, P. D. Shendge, and S. B. Phadke, "Sliding mode control for mismatched uncertain systems using an extended disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 1983–1992, 2014.
- [18] A. Mohammadi, M. Tavakoli, H. J. Marquez, and F. Hashemzadeh, "Nonlinear disturbance observer design for robotic manipulators," *Control Engineering Practice*, vol. 21, no. 3, pp. 253–267, 2013.
- [19] H. Sun, S. Li, J. Yang, and L. Guo, "Non-linear disturbance observer-based back-stepping control for airbreathing hypersonic vehicles with mismatched disturbances," *IET Control Theory and Applications*, vol. 8, no. 17, pp. 1852–1865, 2014.
- [20] S. Xingling and W. Honglun, "Back-stepping active disturbance rejection control design for integrated missile guidance and control system via reduced-order ESO," *ISA Transactions*, vol. 57, pp. 10–22, 2015.
- [21] W. H. Chen, "Nonlinear Disturbance Observer-Enhanced Dynamic Inversion Control of Missiles," *Journal of Guidance Control & Dynamics*, vol. 26, no. 1, pp. 161–166, 2012.
- [22] J. Wang, S. Li, J. Yang, B. Wu, and Q. Li, "Extended state observer-based sliding mode control for PWM-based DC-DC buck power converter systems with mismatched disturbances," *IET Control Theory and Applications*, vol. 9, no. 4, pp. 579–586, 2015.
- [23] C. Wang, X. Li, L. Guo, and Y. W. Li, "A nonlinear-disturbance-observer-based DC-Bus voltage control for a hybrid AC/DC microgrid," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 6162–6177, 2014.
- [24] M. Ekramian, F. Sheikholeslam, S. Hosseinnia, and M. J. Yazdanpanah, "Adaptive state observer for Lipschitz nonlinear

- systems,” *Systems and Control Letters*, vol. 62, no. 4, pp. 319–323, 2013.
- [25] Y. Li and S. Tong, “Adaptive Fuzzy Output-Feedback Stabilization Control for a Class of Switched Nonstrict-Feedback Nonlinear Systems,” *IEEE Transactions on Cybernetics*, 2016.
- [26] S. Tong, Y. Li, and S. Sui, “Adaptive fuzzy tracking control design for uncertain non-strict feedback nonlinear systems,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1441–1454, 2016.
- [27] Y. Li, S. Tong, L. Liu, and G. Feng, “Adaptive output-feedback control design with prescribed performance for switched nonlinear systems,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 80, pp. 225–231, 2017.
- [28] S. Tong, Y. Li, and S. Sui, “Adaptive Fuzzy Output Feedback Control for Switched Nonstrict-Feedback Nonlinear Systems With Input Nonlinearities,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1426–1440, 2016.
- [29] N. Sakamoto, B. Reháč, and K. Ueno, “Nonlinear Luenberger observer design via invariant manifold computation,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 37–42, 2014.
- [30] A. Alessandri and A. Rossi, “Increasing-gain observers for nonlinear systems: stability and design,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 57, pp. 180–188, 2015.
- [31] C. Mu and C. Sun, “A new finite time convergence condition for super-twisting observer based on Lyapunov analysis,” *Asian Journal of Control*, vol. 17, no. 3, pp. 1050–1060, 2015.
- [32] J. M. Daly and D. W. L. Wang, “Output feedback sliding mode control in the presence of unknown disturbances,” *Systems and Control Letters*, vol. 58, no. 3, pp. 188–193, 2009.
- [33] H. v. Grip, A. Saberi, and T. A. Johansen, “Observers for interconnected nonlinear and linear systems,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 48, no. 7, pp. 1339–1346, 2012.
- [34] M. Ghanes, J. De Leon, and J.-P. Barbot, “Observer design for nonlinear systems under unknown time-varying delays,” *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 58, no. 6, pp. 1529–1534, 2013.
- [35] M. Ekramian, F. Sheikholeslam, S. Hosseinnia, and M. J. Yazdanpanah, “Adaptive state observer for Lipschitz nonlinear systems,” *Systems and Control Letters*, vol. 62, no. 4, pp. 319–323, 2013.
- [36] M. H. Holakooie, M. Ojaghi, and A. Taheri, “Full-order Luenberger observer based on fuzzy-logic control for sensorless field-oriented control of a single-sided linear induction motor,” *ISA Transactions*, vol. 60, pp. 96–108, 2016.
- [37] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons, West Sussex, UK, 2011.
- [38] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Inc, Upper Saddle River, NJ, USA, 3rd edition, 2002.
- [39] R. P. Panneer Selvam and S. K. Bhattacharyya, “System identification of a coupled two DOF moored floating body in random ocean waves,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 128, no. 3, pp. 191–202, 2006.