



applied sciences

Special Issue Reprint

Motion Control for Robots and Automation

Edited by
Yi Fang, Yuxin Sun and Dongfang Liu

mdpi.com/journal/applsci



Motion Control for Robots and Automation

Motion Control for Robots and Automation

Guest Editors

Yi Fang

Yuxin Sun

Dongfang Liu



Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Yi Fang

School of Automation and
Intelligent Sensing
Shanghai Jiao Tong University
Shanghai
China

Yuxin Sun

School of Mechanical
Engineering
Shanghai Jiao Tong University
Shanghai
China

Dongfang Liu

Kate Gleason College of
Engineering
Rochester Institute of
Technology
Rochester
USA

Editorial Office

MDPI AG

Grosspeteranlage 5

4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Applied Sciences* (ISSN 2076-3417), freely accessible at: https://www.mdpi.com/journal/applsci/special_issues/C4NA54Z4F6.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. <i>Journal Name</i> Year , Volume Number, Page Range.
--

ISBN 978-3-7258-7655-6 (Hbk)

ISBN 978-3-7258-7656-3 (PDF)

<https://doi.org/10.3390/books978-3-7258-7656-3>

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

Contents

About the Editors	vii
-----------------------------	-----

Yi Fang, Yuxin Sun and Dongfang Liu

Motion Control for Robots and Automation

Reprinted from: <i>Appl. Sci.</i> 2026 , <i>16</i> , 3560, https://doi.org/10.3390/app16073560	1
---	---

Qianyou Zhao, Le Gao, Duidi Wu, Xinyao Meng, Jin Qi and Jie Hu

E-GTN: Advanced Terrain Sensing Framework for Enhancing Intelligent Decision Making of Excavators

Reprinted from: <i>Appl. Sci.</i> 2024 , <i>14</i> , 6974, https://doi.org/10.3390/app14166974	6
---	---

Nengkai Wu, Dongyao Jia, Ziqi Li and Zihao He

Trajectory Planning of Robotic Arm Based on Particle Swarm Optimization Algorithm

Reprinted from: <i>Appl. Sci.</i> 2024 , <i>14</i> , 8234, https://doi.org/10.3390/app14188234	20
---	----

Ruiao Zou and Nan Wang

Imbalanced Data Parameter Optimization of Convolutional Neural Networks Based on Analysis of Variance

Reprinted from: <i>Appl. Sci.</i> 2024 , <i>14</i> , 9071, https://doi.org/10.3390/app14199071	36
---	----

Yu Guo, Ziyi Cheng, Yuanlong Zhang, Gaoxuan Wang and Jundong Zhang

Study on Image Classification Algorithm Based on Multi-Scale Feature Fusion and Domain Adaptation

Reprinted from: <i>Appl. Sci.</i> 2024 , <i>14</i> , 10531, https://doi.org/10.3390/app142210531	62
--	----

Binqing Cai, Zhukai Ye, Shiwei Chen and Xun Liang

Reducing Safety Risks in Construction Tower Crane Operations: A Dynamic Path Planning Model

Reprinted from: <i>Appl. Sci.</i> 2024 , <i>14</i> , 10599, https://doi.org/10.3390/app142210599	82
--	----

Xining Cui, Yi Fang and Chaochen Gu

Residual Vibration Reduction in Flexible Systems Based on Trapezoidal Velocity Profiles

Reprinted from: <i>Appl. Sci.</i> 2025 , <i>11</i> , 1791, https://doi.org/10.3390/app15041791	102
---	-----

Kai Wang, Jinnan Lu and Haodong Zhou

Employing Fuzzy Adaptive and Event-Triggered Approaches to Achieve Formation Control of Nonholonomic Mobile Robots Under Complete State Constraints

Reprinted from: <i>Appl. Sci.</i> 2025 , <i>15</i> , 2827, https://doi.org/10.3390/app15052827	125
---	-----

Tymon F. Wawrzyniak, Ignacy D. Orłowski and Marek A. Galewski

Three-Dimensional Path-Following with Articulated 6DoF Robot and ToF Sensors

Reprinted from: <i>Appl. Sci.</i> 2025 , <i>15</i> , 2917, https://doi.org/10.3390/app15062917	143
---	-----

Weijie Liu, Miao Jia, Shuo Zhang, Siyu Zhu, Jin Qi and Jie Hu

A Lightweight TA-YOLOv8 Method for the Spot Weld Surface Anomaly Detection of Body in White

Reprinted from: <i>Appl. Sci.</i> 2025 , <i>15</i> , 2931, https://doi.org/10.3390/app15062931	168
---	-----

Xinrui Liu, Rutao Wang and Zongsheng Wang

Research on a Point Cloud Registration Method Based on Dynamic Neighborhood Features

Reprinted from: <i>Appl. Sci.</i> 2025 , <i>15</i> , 4036, https://doi.org/10.3390/app15074036	188
---	-----

Sen Yan, Yanping Zhu, Wenlong Chen, Jianqiang Zhang, Chenyang Zhu and Qi Chen Dynamic Obstacle Avoidance for Robotic Arms Using Deep Reinforcement Learning with Adaptive Reward Mechanisms Reprinted from: <i>Appl. Sci.</i> 2025 , <i>15</i> , 4496, https://doi.org/10.3390/app15084496	203
Yixiang Hu, Daxue Liu, Jinhong Li, Junxiang Li and Tao Wu Multi-Robot Task Allocation with Spatiotemporal Constraints via Edge-Enhanced Attention Networks Reprinted from: <i>Appl. Sci.</i> 2026 , <i>16</i> , 904, https://doi.org/10.3390/app16020904	220
Rong Zhang, Jianjun Qin, Luyang Wang and Guotong Li Deep Reinforcement Learning-Based Water Jet Control for Robotic Manipulators Using an Improved Experience Replay Mechanism Reprinted from: <i>Appl. Sci.</i> 2026 , <i>16</i> , 3099, https://doi.org/10.3390/app16063099	246
Junwoo Lee, Seunghwa Oh and Jungwon Seo Manipulation Models for Robotic High-Arc Object Transfer and Their Implementation Reprinted from: <i>Appl. Sci.</i> 2026 , <i>16</i> , 3205, https://doi.org/10.3390/app16073205	267
Hua Chen, Xiaogang Xiong, Koki Honda, Shouta Okunami and Motoji Yamamoto FES Control of a Finger MP Joint with a Proxy-Based Super-Twisting Algorithm Reprinted from: <i>Appl. Sci.</i> 2024 , <i>14</i> , 4905, https://doi.org/10.3390/app14114905	286

About the Editors

Yi Fang

Yi Fang received an Engineer's degree in mechanical systems engineering from the Université de Technologie de Compiègne, France, in 2014, and a Ph.D. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2020. His research interests include robot motion planning, vibration control, bioinspired design, and multi-objective optimization. From 2024 to 2026, he was a Postdoctoral Fellow with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, supported by the Hong Kong Scholars Program. He is currently a Postdoctoral fellow with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China.

Yuxin Sun

Yuxin Sun received a B.E. degree in mechanical engineering from Sichuan University, Chengdu, China, in 2012, and a Ph.D. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017. He is currently an Associate Professor with the Meta Robotics Institute, Shanghai Jiao Tong University. His current research interests include intelligent manufacturing, fault diagnosis, and robotics.

Dongfang Liu

Dongfang Liu is presently an Assistant Professor in the Department of Computer Engineering at the Rochester Institute of Technology (RIT). He obtained his Ph.D. from Purdue University in 2021. Dr. Liu's scholarly pursuits revolve around artificial general intelligence (AGI), guided by a broader mission to cultivate versatile AI systems that effectively tackle pressing societal dilemmas. His scholarly endeavors have garnered substantial acclaim, as evidenced by the endorsement of his research by the National Science Foundation (NSF). Throughout his academic trajectory, Dr. Liu has made significant contributions to the AI field, a fact underscored by his prolific publication record. His work has been published in distinguished conferences, including CVPR, ECCV, ICCV, ICLR, NeurIPS, ICML, AAAI, IJCAI, ACL, EMNLP, WWW, IROS, among several others.

Motion Control for Robots and Automation

Yi Fang ^{1,*}, Yuxin Sun ² and Dongfang Liu ³

¹ School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China

² School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

³ Department of Computer Engineering, Kate Gleason College of Engineering, Rochester Institute of Technology, Rochester, NY 14623, USA

* Correspondence: apocalypse@sjtu.edu.cn

1. Introduction

Motion control constitutes one of the most fundamental aspects of robotics and automation engineering. The ability to generate, execute, and regulate the motion of robotic mechanisms determines the performance of systems, ranging from precision industrial manipulators to autonomous mobile platforms operating in unstructured environments [1,2]. Manufacturing processes demand higher throughput with tighter tolerances, construction operations move toward automation for improved safety, and rehabilitation systems require precise actuation of human joints; the field of motion control continues to evolve in both breadth and technical depth to meet these needs [3,4].

The classical foundations of robotic motion control rest on well-established principles of kinematics, dynamics, and feedback control theory. Interpolation-based trajectory planning, proportional-integral-derivative (PID) regulation, and model-based optimal control have served the robotics community for decades and remain indispensable in many practical applications [5]. However, the growing complexity of modern robotic tasks has motivated the development of more advanced approaches. Sliding mode control and its higher-order variants offer robustness against model uncertainties and external disturbances [6]. Optimization-based methods, including metaheuristic algorithms such as particle swarm optimization (PSO) and genetic algorithms, provide powerful tools for addressing trajectory planning problems with complex constraints [7]. Event-triggered and adaptive control architectures reduce communication and computation overhead in networked multi-robot systems [8].

In parallel, the rapid progress of artificial intelligence has introduced new capabilities into the motion control pipeline. Deep reinforcement learning enables robots to acquire manipulation and navigation skills through trial-and-error interaction with their environments, bypassing the need for explicit analytical models [9]. Convolutional neural networks and their variants have become standard tools for visual perception tasks that inform motion planning decisions, including object detection, defect classification, and scene understanding [10]. Graph neural networks and attention mechanisms are finding application in combinatorial optimization problems such as multi-robot task allocation [11]. The convergence of these data-driven techniques with established control-theoretic methods represents a defining trend in contemporary robotics research [12].

This Special Issue, entitled “Motion Control for Robots and Automation,” was established to capture recent advances across this evolving landscape. The fifteen research contributions collected here address topics spanning trajectory planning and vibration suppression, advanced control strategies for diverse robotic platforms, path planning and

safety-oriented automation, environmental perception and quality inspection, and data-driven methods for visual intelligence. The contributing works involve a broad range of application domains, including industrial manufacturing, construction machinery, rehabilitation engineering, multi-robot coordination, and autonomous excavation. The following sections provide an overview of the published articles and conclude with a discussion of emerging themes and future research directions.

2. An Overview of Published Articles

Trajectory planning and vibration suppression in industrial robotic applications form the first thematic cluster. The challenge of optimizing S-shaped velocity profiles for robotic arms was taken up by Wu et al. [13], who presented a robot trajectory planning particle swarm optimization (RTPPSO) technique incorporating adaptive weight strategies and random perturbation terms to overcome premature convergence. Simulation results confirmed improved acceleration characteristics and reduced execution times relative to standard particle swarm optimization and genetic algorithm methods. Residual vibration during high-speed motion of lightweight industrial components motivated the analytical study of Cui et al. [14], in which two vibration suppression conditions for trapezoidal velocity profiles were derived from the relationship between acceleration switching times and system natural periods. Experiments on two robot manipulators demonstrated vibration reductions exceeding 90% without requiring controller modifications. A rather different motion paradigm was explored in [15], where Lee et al. developed non-prehensile dynamic manipulation techniques for rapid object transfer. Two complementary methods, a two-fingered scoop-and-flick and a one-fingered topple-and-flick, were validated on a custom platform, showing that impulsive contact interactions can achieve accurate high-arc transfers as efficient alternatives to conventional grasp-based operations.

Advanced control strategies for diverse robotic platforms constitute the second group of contributions. To address the challenge of servo control for finger joint rehabilitation through functional electrical stimulation, Chen et al. [16] developed the proxy-based super-twisting algorithm, which effectively prevents integrator windup during actuator saturation. The algorithm was implemented on Arduino hardware with an implicit Euler method to suppress numerical chattering, and experiments with human volunteers confirmed superior tracking accuracy over conventional methods. Formation control of multiple nonholonomic mobile robots under full-state constraints was the focus of [17], where Wang et al. combined fuzzy logic approximation of unknown dynamics with barrier Lyapunov functions and a backstepping framework. Their event-triggered communication mechanism achieved an 88.17% resource-saving rate for a four-robot group while maintaining bounded formation tracking errors. Reinforcement learning provides the methodological foundation for the next two contributions. Yan et al. [18] tackled obstacle avoidance for robotic arms by integrating inverse-kinematics-guided exploration with an adaptive reward function, achieving faster convergence and higher success rates than standard soft actor-critic (SAC) and deep deterministic policy gradient (DDPG) in dynamic environments. The problem of controlling a manipulator to direct water jets at target positions was addressed in [19] through Goal-Priority Hindsight Experience Replay (GPHER), a method that dynamically adjusts sampling priorities based on goal-space distance to improve performance in difficult boundary regions of the workspace.

Path planning and safety-oriented automation represent another important thread running through the collection. Recognizing the risk that dropped loads pose to ground workers during tower crane operations, Cai et al. [20] developed a dynamic path planning model that combines You Only Look Once version 8 (YOLOv8) instance segmentation with an improved dynamic window approach, validated on actual construction sites. The concept of

line-following was extended into three-dimensional space by Wawrzyniak et al. [21], who equipped an articulated 6-Degrees-of-Freedom (DoF) robot with six low-cost Time-of-Flight (ToF) sensors and achieved reliable centimeter-level accuracy with minimal computational overhead. In the domain of multi-robot coordination, Hu et al. [22] formulated the task allocation problem under spatiotemporal and spatial constraints as an asynchronous Markov Decision Process (MDP) over a directed heterogeneous graph and proposed the Edge-Enhanced Attention Network (E2AN), whose encoder incorporates edge attributes reflecting actual path costs under obstacles.

Environmental perception and quality inspection also received significant attention. A notable contribution to terrain sensing for autonomous excavation came from Zhao et al. [23], who introduced a framework that processes Light Detection and Ranging (LiDAR) point clouds through a custom GridNet to extract terrain features informing a reinforcement learning agent. Surface anomaly detection on spot welds in automotive production was the subject of [24], where Liu et al. proposed improved YOLOv8 incorporating task-aligned detection heads and cross-scale feature fusion alongside a balanced generative adversarial network (GAN) for data augmentation. Efficient point cloud registration under varying data density conditions was investigated by Liu et al. [25], who presented a method based on dynamic neighborhood features that adapts the neighborhood radius to local geometry, achieving competitive accuracy across standard benchmark datasets. Finally, two contributions addressed the optimization of visual intelligence methods from complementary angles: Zou and Wang [26] systematically optimized convolutional neural network (CNN) parameters for imbalanced image classification through analysis of variance with fractional factorial design, while Guo et al. [27] integrated multi-scale feature extraction with adversarial domain adaptation in their multi-scale, multi-channel Taylor adversarial domain adaptation network (MMTADAN) algorithm for cross-domain image classification.

3. Conclusions

The fifteen articles collected in this Special Issue reflect both the breadth and the depth of contemporary research in robotic motion control. Several themes cut across the individual contributions. Classical control and analytical optimization methods continue to provide reliability and formal guarantees, while reinforcement learning and neural networks contribute adaptability to complex, unmodeled dynamics. Computational efficiency is a recurring concern, evidenced by embedded platform implementations, lightweight network architectures, and event-triggered communication strategies. Safety has emerged as a prominent design objective, particularly in construction and human–robot coexistence scenarios where vision-guided dynamic path planning shows considerable promise. Multi-robot coordination is advancing toward richer environmental representations that go beyond Euclidean distance assumptions to capture the true spatial structure of operating environments.

Several open questions merit further investigation. Reliable sim-to-real transfer of learned policies, extension to scenarios involving deformable objects and unstructured outdoor environments, and the potential integration of foundation models and large language models into robotic planning pipelines all represent promising research frontiers. Equally important, as robotic systems increasingly share workspaces with humans, research on safety verification, ethical governance, and user trust will be essential for translating laboratory achievements into dependable real-world deployments. We hope this Special Issue offers useful perspectives for researchers and engineers advancing the science and practice of motion control in robotics and automation.

Author Contributions: Writing—original draft preparation, Y.F.; writing—review and editing, Y.S. and D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hägele, M.; Nilsson, K.; Pires, J.N.; Bischoff, R. Industrial Robotics. In *Springer Handbook of Robotics*, 2nd ed.; Siciliano, B., Khatib, O., Eds.; Springer: Cham, Switzerland, 2016; pp. 1385–1422.
2. Craig, J.J. *Introduction to Robotics: Mechanics and Control*, 3rd ed.; Pearson Education India: New Delhi, India, 2009.
3. Banyai, A.D.; Brişan, C. Robotics in Physical Rehabilitation: Systematic Review. *Healthcare* **2024**, *12*, 1720. [CrossRef] [PubMed]
4. Merckaert, K.; Convens, B.; Wu, C.-J.; Roncone, A.; Nicotra, M.M.; Vanderborght, B. Real-Time Motion Control of Robotic Manipulators for Safe Human–Robot Coexistence. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102223. [CrossRef]
5. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*; Springer: London, UK, 2009.
6. Chaudhary, K.S.; Kumar, N. Fractional Order Fast Terminal Sliding Mode Control Scheme for Tracking Control of Robot Manipulators. *ISA Trans.* **2023**, *142*, 57–69. [CrossRef]
7. Kucuk, S. Optimal Trajectory Generation Algorithm for Serial and Parallel Manipulators. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 219–232. [CrossRef]
8. Abbas, M.; Dwivedy, S.K. Adaptive Control for Networked Uncertain Cooperative Dual-Arm Manipulators: An Event-Triggered Approach. *Robotica* **2022**, *40*, 1951–1978. [CrossRef]
9. Tang, C.; Abbatematteo, B.; Hu, J.; Chandra, R.; Martín-Martín, R.; Stone, P. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. *Annu. Rev. Control Robot. Auton. Syst.* **2025**, *8*, 153–188. [CrossRef]
10. Al-Shanoon, A.; Lang, H. Robotic Manipulation Based on 3-D Visual Servoing and Deep Neural Networks. *Robot. Auton. Syst.* **2022**, *152*, 104041. [CrossRef]
11. Zhou, Y.; Xiao, J.; Zhou, Y.; Loianno, G. Multi-Robot Collaborative Perception with Graph Neural Networks. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2289–2296. [CrossRef]
12. Brunke, L.; Greeff, M.; Hall, A.W.; Yuan, Z.; Zhou, S.; Panerati, J.; Schoellig, A.P. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annu. Rev. Control Robot. Auton. Syst.* **2022**, *5*, 411–444. [CrossRef]
13. Wu, N.; Jia, D.; Li, Z.; He, Z. Trajectory Planning of Robotic Arm Based on Particle Swarm Optimization Algorithm. *Appl. Sci.* **2024**, *14*, 8234. [CrossRef]
14. Cui, X.; Fang, Y.; Gu, C. Residual Vibration Reduction in Flexible Systems Based on Trapezoidal Velocity Profiles. *Appl. Sci.* **2025**, *15*, 1791. [CrossRef]
15. Liu, D.; Lee, J.; Oh, S.; Seo, J. Manipulation Models for Robotic High-Arc Object Transfer and Their Implementation. *Appl. Sci.* **2026**, *16*, 3205. [CrossRef]
16. Chen, H.; Xiong, X.; Honda, K.; Okunami, S.; Yamamoto, M. FES Control of a Finger MP Joint with a Proxy-Based Super-Twisting Algorithm. *Appl. Sci.* **2024**, *14*, 4905. [CrossRef]
17. Wang, K.; Lu, J.; Zhou, H. Employing Fuzzy Adaptive and Event-Triggered Approaches to Achieve Formation Control of Nonholonomic Mobile Robots Under Complete State Constraints. *Appl. Sci.* **2025**, *15*, 2827. [CrossRef]
18. Yan, S.; Zhu, Y.; Chen, W.; Zhang, J.; Zhu, C.; Chen, Q. Dynamic Obstacle Avoidance for Robotic Arms Using Deep Reinforcement Learning with Adaptive Reward Mechanisms. *Appl. Sci.* **2025**, *15*, 4496. [CrossRef]
19. Zhang, R.; Qin, J.; Wang, L.; Li, G. Deep Reinforcement Learning-Based Water Jet Control for Robotic Manipulators Using an Improved Experience Replay Mechanism. *Appl. Sci.* **2026**, *16*, 3099. [CrossRef]
20. Cai, B.; Ye, Z.; Chen, S.; Liang, X. Reducing Safety Risks in Construction Tower Crane Operations: A Dynamic Path Planning Model. *Appl. Sci.* **2024**, *14*, 10599. [CrossRef]
21. Wawrzyniak, T.F.; Orłowski, I.D.; Galewski, M.A. Three-Dimensional Path-Following with Articulated 6DoF Robot and ToF Sensors. *Appl. Sci.* **2025**, *15*, 2917. [CrossRef]
22. Hu, Y.; Liu, D.; Li, J.; Li, J.; Wu, T. Multi-Robot Task Allocation with Spatiotemporal Constraints via Edge-Enhanced Attention Networks. *Appl. Sci.* **2026**, *16*, 904. [CrossRef]
23. Zhao, Q.; Gao, L.; Wu, D.; Meng, X.; Qi, J.; Hu, J. E-GTN: Advanced Terrain Sensing Framework for Enhancing Intelligent Decision Making of Excavators. *Appl. Sci.* **2024**, *14*, 6974. [CrossRef]
24. Liu, W.; Jia, M.; Zhang, S.; Zhu, S.; Qi, J.; Hu, J. A Lightweight TA-YOLOv8 Method for the Spot Weld Surface Anomaly Detection of Body in White. *Appl. Sci.* **2025**, *15*, 2931. [CrossRef]
25. Liu, X.; Wang, R.; Wang, Z. Research on a Point Cloud Registration Method Based on Dynamic Neighborhood Features. *Appl. Sci.* **2025**, *15*, 4036. [CrossRef]

26. Zou, R.; Wang, N. Imbalanced Data Parameter Optimization of Convolutional Neural Networks Based on Analysis of Variance. *Appl. Sci.* **2024**, *14*, 9071. [CrossRef]
27. Guo, Y.; Cheng, Z.; Zhang, Y.; Wang, G.; Zhang, J.; Fernández-Caballero, A. Study on Image Classification Algorithm Based on Multi-Scale Feature Fusion and Domain Adaptation. *Appl. Sci.* **2024**, *14*, 10531. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

E-GTN: Advanced Terrain Sensing Framework for Enhancing Intelligent Decision Making of Excavators

Qianyou Zhao ¹, Le Gao ², Duidi Wu ¹, Xinyao Meng ¹, Jin Qi ^{1,*} and Jie Hu ^{1,*}¹ School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 201100, China² Sany Heavy Machinery Co., Ltd., Kunshan, Suzhou 215300, China

* Correspondence: jinhuaqj@sjtu.edu.cn (J.Q.); hujie@sjtu.edu.cn (J.H.)

Abstract: The shift towards autonomous excavators in construction and mining is a significant leap towards enhancing operational efficiency and ensuring worker safety. However, it presents challenges, such as the need for sophisticated decision making and environmental perception due to complex terrains and diverse conditions. Our study introduces the E-GTN framework, a novel approach tailored for autonomous excavation that leverages advanced multisensor fusion and a custom-designed convolutional neural network to address these challenges. Results demonstrate that GridNet effectively processes grid data, enabling the reinforcement learning algorithm to make informed decisions, thereby ensuring efficient and intelligent autonomous excavator performance. The study concludes that the E-GTN framework offers a robust solution for the challenges in unmanned excavator operations, providing a valuable platform for future advancements in the field.

Keywords: excavator automation; point cloud; reinforcement learning; terrain feature extraction

1. Introduction

Excavators, the backbone of construction, mining, and infrastructure projects, are transitioning from being manually operated to becoming autonomous as part of digitization and intelligentization in the construction machinery industry [1]. This shift towards unmanned excavation is not just a leap in operational efficiency and production capacity, especially in the mining sector, but also a step forward in worker safety by reducing exposure to hazardous conditions. However, the transition to unmanned systems introduces considerable challenges, as the current reliance on expert experience and dynamic programming for trajectory planning is limited by their inability to match the efficiency of manual excavation and their poor adaptability to diverse environmental conditions and soil types [2].

The challenges of unmanned excavation are multifaceted, involving the intricacies of data complexity and the need for robust and reliable operations [3]. Excavators must navigate complex terrains, adapt to varying excavation areas, and integrate a wealth of mechanical and sensor data to accurately perceive and respond to their environment [4]. Discrepancies between virtual simulations and real-world scenarios necessitate careful data sampling and raise issues of integration among simulation platforms, control systems, and reinforcement learning algorithms. These complexities can lead to overfitting and a degradation in performance, underscoring the need for advanced solutions that can handle the sophisticated decision making and environmental perception required for effective unmanned excavator operations [5].

Addressing key gaps in autonomous excavation, we identify:

1. **Terrain Adaptability:** Autonomous systems lack flexibility, falling short in efficiency compared to manual excavation.
2. **Simulation–Reality Discrepancy:** Complex environmental methods struggle with overfitting due to differences between simulations and actual conditions.

3. Static Decision Models: Existing systems need dynamic decision making to adapt in real-time to environmental changes.

To address these challenges, we designed the E-GTN framework. It is a specialized pipeline designed for the autonomous mining domain, focusing on terrain information extraction related to reinforcement learning tasks in unmanned mining operations. The E-GTN framework first uses LiDAR to capture raw point cloud data to reconstruct terrain features in multiple ways. The terrain feature extraction module is the core of the E-GTN framework and utilizes a convolutional neural network for advanced feature extraction. Our customized GridNet proficiently processes grid data to extract key terrain features that contribute to reinforcement learning tasks in autonomous mining. Finally, the decision module utilizes the features extracted by the GridNet to inform the reinforcement learning algorithm, ensuring efficient and intelligent autonomous excavator operation. E-GTN ensures a smooth transition from data collection to extraction for terrain features, simplifying the entire process. The operational environment awareness technology provides the necessary technical support for the next path generation worker, which is the precursor of the excavation trajectory generation technology.

Our main contributions are as follows:

1. We introduce advanced multisensor fusion for terrain environment reconstruction within the E-GTN framework, significantly enhancing the terrain's geometric representation and laying the groundwork for high-fidelity environmental reconstruction.
2. We present a convolutional network-based perception for grid-based excavation environments through the Terrain Feature Extraction module, enabling the application of our custom-designed model, GridNet, to extract salient terrain features crucial for the reinforcement learning algorithm.
3. We model the decision-making process as a Markov decision process (MDP), and develop an advanced deep reinforcement learning (DRL) algorithm for excavation tasks, which provides a comprehensive platform benefiting scholars and practitioners in related fields.

2. Related Work

In the field of autonomous machinery, unmanned excavators, while not as complex as self-driving cars, face a unique set of challenges. The character of the typical excavator working environment is static and there is a reduction in the variety of objects compared to an autonomous driving scenario. However, the terrain features of excavator scenarios are highly variable and detailed in form.

Significant research, both domestic and international, has been conducted on environmental perception technologies for unmanned excavators. Pioneering systems developed by scholars like Stentz et al. [6] utilized LiDAR on excavators for environmental perception, laying the groundwork for subsequent systems. Despite their limitations in integrating driving and positioning, these early systems demonstrated the potential for stationary data collection. Yamamoto et al. [7], tailoring to specific excavation projects, developed an automated hydraulic excavator prototype based on 3D information, focusing on environmental perception through a combination of LiDAR and cameras, supplemented with gyroscopes and GPS for comprehensive data collection.

Further advancements by Chae et al. [8] introduced a mobile 3D environment recognition system for civil engineering, utilizing movable 3D laser scanners to scan and input site terrain for processing. Shariati et al. [9] proposed a multi-frame convolutional approach, providing a complete solution for unmanned excavators by exploiting temporal information between continuous frames and extracting richer features through CNNs. This mature technology has been tested globally, achieving recognition accuracy above 90% at 10 fps, serving as a foundational step in various implementations. Forkel's work [10] on large-scale grid mapping with LiDAR for autonomous driving was notable but lacked focus on under-vehicle information crucial for unmanned excavator perception. Collaborative research by Baidu and the University of Maryland [3] developed an Autonomous Excava-

tor System (AES) that combined cameras, positioning systems, and LiDAR for extended autonomous excavation.

Building upon the foundation set by previous research, our method innovatively processes LiDAR data through a deep learning-based terrain processing network, which efficiently converts complex 3D point clouds into 2D pseudo-images for feature extraction, as summarized in Table 1. This method not only simplifies data processing but also focuses more on the extraction of terrain features. By integrating these features with a DRL algorithm, our system achieves intelligent trajectory generation with improved computational efficiency, signifying a significant advancement over existing research.

Table 1. Comparative analysis of excavator environmental perception methods.

Input	Features	Our Contributions
[6,7] LiDAR + Cameras + GPS	Integration of early environmental perception with 3D information for excavation projects	Advanced multisensor fusion and terrain processing network based on deep learning
[8] Movable 3D laser scanners	Mobile scanning for civil engineering	Efficient data processing through pseudo-imaging
[9] Cameras	Rich feature extraction but low efficiency	Custom GridNet for salient terrain feature extraction
[3,10] LiDAR	Large-scale 3D point cloud reconstruction and traditional control algorithm for excavation	Combination of reduced dimensionality and DRL algorithm

3. Methods

We introduce the E-GTN framework, a specialized pipeline designed for the autonomous excavation domain, focusing primarily on the extraction of terrain information pertinent to reinforcement learning tasks in unmanned excavation operations. The overall structure of E-GTN is shown in Figure 1.

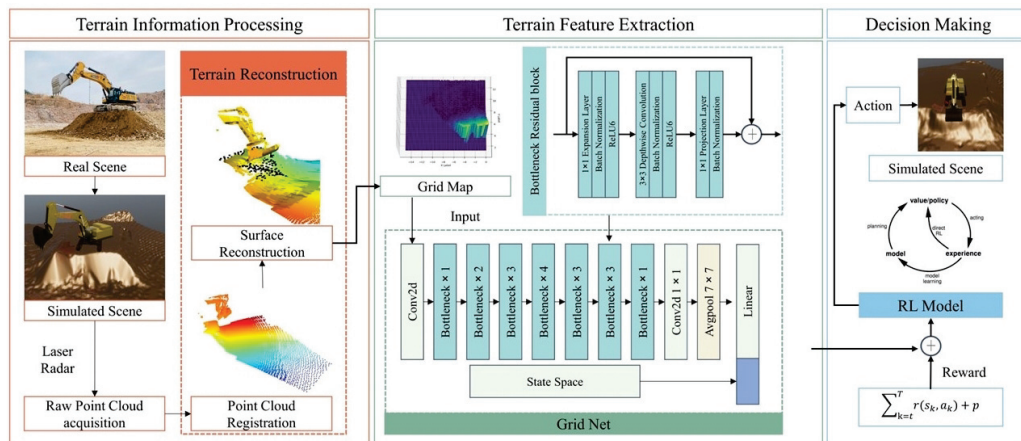


Figure 1. Overall architecture of E-GTN.

In the terrain information processing stage of our E-GTN framework, we commence with the capture of raw point cloud data using laser radar. These data are then processed using voxel grid downsampling and normal estimation to reduce complexity and enhance feature representation. The resulting downsampled point cloud is essential for creating a detailed and computationally manageable model of the terrain, which is critical for accurate environmental reconstruction and subsequent analysis.

The terrain feature extraction module is the centerpiece of the E-GTN framework, where the downsampled point cloud is meticulously transformed into a structured grid map. This grid map undergoes normalization and is treated as a pseudo-image, allowing us to leverage convolutional neural networks (CNNs) for advanced feature extraction. Our tailored model GridNet adeptly processes the grid data to distill critical terrain features that are instrumental for the reinforcement learning tasks in autonomous excavation.

Finally, the decision-making module utilizes the features extracted by GridNet to inform the reinforcement learning algorithm. While this section is less emphasized, it is still pivotal as it enables the autonomous system to make strategic decisions based on the terrain information provided. The integration of these modules within the E-GTN framework ensures a seamless flow from data acquisition to decision execution, facilitating efficient and intelligent autonomous excavator operations.

3.1. Terrain Information Processing

3.1.1. Raw Point Cloud Acquisition

Given the raw point cloud data P , we apply a voxel grid downsampling method. For each voxel V with edge length l , the subset of points $P_V \subseteq P$ is replaced with their centroid c_V to obtain the downsampled point cloud P' :

$$c_V = \frac{1}{|P_V|} \sum_{p \in P_V} p, \quad P' = \bigcup_V c_V \quad (1)$$

where $|P_V|$ denotes the number of points in voxel V . The next step involves estimating the normal at each point in P' . Using a neighborhood defined by a radius search in a KD-tree structure [11], the normal vector $\vec{n}_p$ at point p is computed as the eigenvector corresponding to the smallest eigenvalue of the covariance matrix constructed from the neighbors of p .

For feature description, we compute Fast Point Feature Histograms (FPFHs) [12] for each point p in P' based on the normal vectors and the relative positions within a neighborhood $\mathcal{N}(p)$:

$$FPFH(p) = SPFH(p) + \frac{1}{k} \sum_{p_k \in \mathcal{N}(p)} \frac{1}{\omega_{pk}} SPFH(p_k) \quad (2)$$

where $SPFH(p)$ is the Simplified Point Feature Histogram for point p , ω_{pk} is the inverse Euclidean distance between p and p_k , and k is the number of neighbors.

To align the point clouds into a common coordinate frame, we first apply an initial rough alignment using feature matching. A RANSAC-based approach with $FPFH$ features estimates the transformation (R_i, t_i) by maximizing the consensus set of matched features:

$$(R_i, t_i) = \arg \max_{R, t} |\{(p, q) \in P' \times Q' : \|(Rp + t) - q\| < \epsilon\}| \quad (3)$$

where Q' is the downsampled point cloud from the other laser scanner and ϵ is a threshold for matching.

Subsequently, the Iterative Closest Point (ICP) algorithm [13] refines the transformation (R, t) by minimizing the mean squared error between the corresponding points:

$$(R^*, t^*) = \arg \min_{R, t} \sum_{(p, q) \in \mathcal{C}} \|(Rp + t) - q\|^2 \quad (4)$$

where $\mathcal{C}$ is the set of corresponding point pairs between the registered point cloud P' and the target cloud Q' , and (R^*, t^*) is the optimized transformation. Through these steps, we ensure the integrity and computational efficiency of the data, leading to a high-fidelity reconstruction of the excavation site suitable for further simulation and analysis.

3.1.2. Environment Reconstruction

In our study, the region of interest (ROI) is delineated by an axis-aligned bounding box to focus on terrain changes in front of the excavator bucket. The ROI is defined by coordinates $\min_{ROI}$ and $\max_{ROI}$, significantly reducing data processing volume and increasing efficiency.

The Poisson Surface Reconstruction algorithm [14] seeks a smooth surface that approximates the normals of the point cloud. The mathematical formulation involves solving the Poisson equation:

$$\Delta f = \text{div}(\mathbf{N}) \quad (5)$$

where f is the scalar field whose gradient approximates the point cloud normals $\mathbf{N}$, Δ is the Laplacian, and div is the divergence operator. The normals $\mathbf{N}$ are estimated from the input point cloud, and the divergence of $\mathbf{N}$ forms the right-hand side of the equation, representing the flux of the normal vectors.

The boundary conditions for the domain Ω are typically set to Neumann conditions:

$$\left. \frac{\partial f}{\partial n} \right|_{\partial\Omega} = \mathbf{N} \cdot \mathbf{n} \quad (6)$$

where $\frac{\partial f}{\partial n}$ is the derivative of f in the direction of the outward normal $\mathbf{n}$ on the boundary $\partial\Omega$.

After solving the Poisson equation, the surface S is extracted as the isosurface where the scalar field f equals a chosen iso-value τ , typically zero:

$$S = \{\mathbf{x} \in \Omega \mid f(\mathbf{x}) = \tau\} \quad (7)$$

resulting in a triangular mesh that represents the continuous surface of the point cloud data. This reconstruction fills gaps and creates a model with uniform density, facilitating further processing and analysis.

3.2. Terrain Feature Extraction

3.2.1. Point Cloud to Grid Dimensionality Reduction Mapping

In the preprocessing phase, the original point cloud data, which include a set of points each with coordinates, intensity, and environmental label, are denoted as $\mathbf{P} = \{\mathbf{p}_i\}$, where $\mathbf{p}_i = [x_i, y_i, z_i, I_i, l_i]$ and $i = 1, \dots, N$. Here, N is the total number of points in the point cloud, I_i represents the intensity, and l_i is the label of the i -th point.

Filtering based on the label l retains only the points associated with the target environment, resulting in a reduced point cloud $\mathbf{P}' = \{\mathbf{p}'_i\}$, where $\mathbf{p}'_i = [x'_i, y'_i, z'_i]$ and $i' = 1, \dots, N'$. Here, N' is the count of points in the filtered point cloud.

The grid map parameters are computed as follows:

$$\text{grid_width} = \left\lfloor \frac{W}{\Delta x} \right\rfloor, \quad \text{grid_height} = \left\lfloor \frac{H}{\Delta y} \right\rfloor \quad (8)$$

where W and H are the width and height of the scene, respectively, and Δx and Δy are the grid resolutions in the respective dimensions. An empty grid map $\mathbf{G}$ is initialized with zero values:

$$\mathbf{G} = \mathbf{0}_{\text{grid_height} \times \text{grid_width}} \quad (9)$$

For each point $\mathbf{p}'_i$ within the specified range, the corresponding grid indices (i_x, i_y) are calculated by:

$$i_x = \left\lfloor \frac{x'_i - x_{\min}}{\Delta x} \right\rfloor, \quad i_y = \left\lfloor \frac{y'_i - y_{\min}}{\Delta y} \right\rfloor \quad (10)$$

where $x_{\min}$ and $y_{\min}$ are the minimum coordinates of the grid. The z-coordinate is mapped to the grid if it falls within the specified range $[z_{\min}, z_{\max}]$:

$$\mathbf{G}(i_y, i_x) = \begin{cases} z'_i & \text{if } z_{\min} \leq z'_i \leq z_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The grid $\mathbf{G}$ is then reshaped into a vector for efficient storage and processing:

$$\mathbf{g} = \text{reshape}(\mathbf{G}, \text{grid_height} \cdot \text{grid_width}) \quad (12)$$

For visualization and further analysis, the grid map is normalized and converted into a pseudo-image $\mathbf{I}_{\text{grid}}$:

$$\mathbf{I}_{\text{grid}} = \frac{\mathbf{G} - \min(\mathbf{G})}{\max(\mathbf{G}) - \min(\mathbf{G})} \quad (13)$$

This pseudo-image is then scaled to the range $[0, 1]$ to standardize the data for subsequent processing steps:

$$\mathbf{I}_{\text{grid}} = \frac{\mathbf{I}_{\text{grid}}}{255} \quad (14)$$

The normalized grid map $\mathbf{I}_{\text{grid}}$ is now ready for feature extraction and environmental perception tasks, providing a consistent input for machine learning algorithms.

3.2.2. Terrain Feature Extraction Network

To facilitate the processing of grid data by convolutional neural networks (CNNs), which are typically designed for RGB images, the channel dimension of the grid data must be expanded to simulate a three-channel image. This expansion is denoted by the operation `expand_channels`, which replicates the single-channel grid pseudo-image $\mathbf{G}$ across three channels, yielding a tensor $\mathbf{Grid}_{\text{RGB}}$ with dimensions $[3, H_g, W_g]$, where H_g and W_g represent the height and width of the grid, respectively:

$$\mathbf{Grid}_{\text{RGB}} = \text{expand_channels}(\mathbf{G}) \in \mathbb{R}^{3 \times H_g \times W_g} \quad (15)$$

For batch processing, the tensor is further extended to accommodate a batch size of bs , resulting in a four-dimensional tensor $\mathbf{Grid}_{\text{batch}}$:

$$\mathbf{Grid}_{\text{batch}} \in \mathbb{R}^{bs \times 3 \times H_g \times W_g} \quad (16)$$

The environmental state vector **State** is defined to encapsulate additional necessary inputs from the environment, excluding the target point cloud. It is represented as $\mathbf{State} \in \mathbb{R}^{bs \times S_d}$, where S_d is the dimensionality of the state information.

The feature extraction network, denoted as GridNet, is composed of a CNN model $\text{CNN}_{\text{GridNet}}$ for processing grid data and a fully connected network $\text{FCN}_{\text{GridNet}}$ for processing state data. The CNN model, incorporating depthwise separable convolutions and an inverted residual structure, transforms the input dimensions as follows:

$$\text{CNN}_{\text{GridNet}} : \mathbb{R}^{3 \times H_g \times W_g} \rightarrow \mathbb{R}^{C_g} \quad (17)$$

where C_g represents the number of feature channels produced by the CNN. The fully connected network processes the state vector:

$$\text{FCN}_{\text{GridNet}} : \mathbb{R}^{S_d} \rightarrow \mathbb{R}^{C_s} \quad (18)$$

where C_s is the size of the feature vector for the state data.

The environmental feature vector **Feat** is obtained by concatenating the output of the CNN and FCN models:

$$\mathbf{Feat} = \text{concat}(\text{CNN}_{\text{GridNet}}(\mathbf{Grid}_{\text{RGB}}), \text{FCN}_{\text{GridNet}}(\mathbf{State})) \in \mathbb{R}^{C_g + C_s} \quad (19)$$

MobileNetV2 [15] is selected for its efficiency and is specifically adapted for this application. This structure initially expands the channel count of the input feature map through a lightweight expansion layer. It elevates the dimensions before applying depthwise separable convolutions for feature extraction. Finally, a linear projection layer reduces the dimensions back to the original size.

3.3. Decision Making

Within the scope of excavator operations, we confront the dynamic optimization challenge by conceptualizing the joint optimization of excavation strategies as an MDP [16]. To effectively address this MDP, we engineered an adept DRL [17] algorithm. This algorithm is specifically tailored to navigate the intricacies of continuous excavation tasks, thereby refining the decision-making process in a real-time setting.

The MDP framework of our DRL model is succinctly encapsulated by the tuple $\langle S, A, P, r \rangle$, where S delineates the state space, A signifies the action space, P represents the set of state transition probabilities, and r constitutes the reward function.

3.3.1. State Space

At any decision epoch t , the state space $S(t)$ is formulated to provide a detailed depiction of the excavator's operational status and its environmental engagement. The state space is concisely defined as:

$$S(t) \triangleq \{J(t), S(t), V_{soil}(t)\} \quad (20)$$

where $J(t)$ encompasses the joint angles of the excavator, $S(t)$ entails the angular velocities of the joints, and $V_{soil}(t)$ quantifies the volume of soil excavated at time t .

Given the continuum of the state space, we model the state transition probability as a probability density function f , which quantifies the probability of migrating to the subsequent state $s(t+1)$ consequent to executing an action $a(t)$:

$$Pr(s(t+1)|s(t), a(t)) = \int_{S^{t+1}} f(s(t), a(t), s') ds' \quad (21)$$

3.3.2. Action Space

The action space $A(t)$, at time t , is comprised of a series of potential excavator movements, each represented by the discrete adjustments in the joint angles:

$$A(t) \triangleq \{\Delta J_{swing}(t), \Delta J_{boom}(t), \Delta J_{arm}(t), \Delta J_{bucket}(t)\} \quad (22)$$

Each component of $A(t)$ specifies an incremental modification to a corresponding joint angle, thereby facilitating precise control over the excavator's movements and excavation activities.

4. Experiments

4.1. Data Acquisition

Data for our study were sourced from actual excavation sites, as depicted in Figure 2a. The data collection involved an SY870 excavator [18] performing earthmoving tasks, with the operator ensuring consistent excavation criteria and ground leveling post-operation. The SY870 excavator is manufactured by Sany Group, headquartered in Changsha, China. The excavator's position remained fixed during the tasks, and the terrain was restored after each session to maintain consistent conditions for every new excavation. The SY870 is a hefty machine with a 4.5 cubic meter bucket capacity. In the unloading phase, excavated soil was loaded into mining trucks.

Throughout the excavation processes, the excavator's location was kept constant. The operator used a uniform standard to determine the end of the excavation process. The goal was to keep the post-excavation ground as level as possible and to ensure a constant

excavation speed. To avoid operational anomalies, the operator restored the terrain after each excavation to ensure the initial conditions for subsequent operations were as consistent as possible. Figure 2b shows the work area of the excavation operation, with Area A being the excavation zone and Area B being the unloading zone. The width of the spoil platform at the excavation site was 13 m, and the work area was 6.5 m by 1.4 m by 4 m.

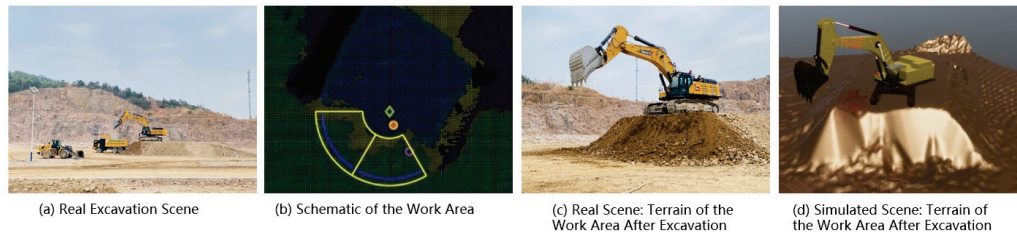


Figure 2. Excavation operation: Real to simulated.

To simulate actual operations, radar devices were installed on both sides of the excavator's cabin and boom to collect real-time data about the surrounding environment and terrain. The excavator's LiDAR is the Ouster OS1. The Ouster OS1 is manufactured by Ouster, Inc., headquartered in San Francisco, CA, USA. This is a high-performance LiDAR with a detection range of up to 100 m and an angular sampling accuracy of $\pm 0.01^\circ$ vertically and $\pm 0.01^\circ$ horizontally. The excavator's four joint angles are fitted with IMU SYD T500 sensors. The manufacturer of the IMU SYD T500 sensors is SYD Dynamics, based in Odense, Southern Denmark. This is an integrated inertial measurement unit that incorporates multiple sensors, capable of outputting measurements from a 3-axis gyroscope and a 3-axis accelerometer with a timestamp resolution of less than 1 μ s. During operations, these data were stored at a frequency of 50 Hz in Rosbag packages. The dataset includes the angular velocity of each joint of the excavator, joint force, piston rod speed, volume of soil excavated, ground support force, and LIDAR point clouds. The operator's excavation trajectory, followed during the operation, was transmitted to a simulation platform (Unity) via the UDP protocol in Matlab 2023a to acquire data on the excavation process in a simulated environment. Figure 2c illustrates the terrain of the work area after the completion of the excavation in the real scenario. Figure 2d displays the terrain of the work area after the excavation in the simulated scenario, which is essentially consistent with the terrain in the real scenario.

4.2. Environment Reconstruction

After capturing the original point cloud data, we processed the terrain information based on the point cloud. The process began with preliminary steps, such as downsampling, normal estimation, and FPFH feature extraction. This was followed by initial registration and then precise alignment using the ICP method for the point clouds collected by the left and right radars of the bucket. Next, we reconstructed the point cloud by first removing outliers and any unreasonable points. We then defined the ROI for the actual excavation process. After selecting the ROI, we used Poisson surface reconstruction techniques to rebuild the point cloud, filling in gaps and preparing for the conversion of the point cloud to a grid and feature extraction.

Our methodology employed the OpenCV library for normal estimation and point cloud registration. This approach facilitated rapid neighborhood searches and maintained a balance between accuracy and computational efficiency.

The preliminary registration aimed to align the left and right radar point clouds roughly, setting the stage for more precise alignment. We utilized OpenCV's feature-based method to quickly estimate an initial transformation matrix.

For fine-tuning the alignment, we applied the ICP algorithm, optimizing the point clouds' alignment accuracy. The processed point clouds, an initial transformation from the preliminary step, and a set maximum distance for corresponding points were fed

into OpenCV's ICP implementation. This step refined the point clouds' alignment by minimizing the direct Euclidean distance between corresponding points.

Figure 3a shows the unregistered point clouds from the left and right radars, where the misalignment is evident. Figure 3b illustrates the point clouds after preliminary registration, showing basic alignment. After ICP registration, as shown in Figure 3c, the left and right point clouds are well aligned, resulting in a fused point cloud.

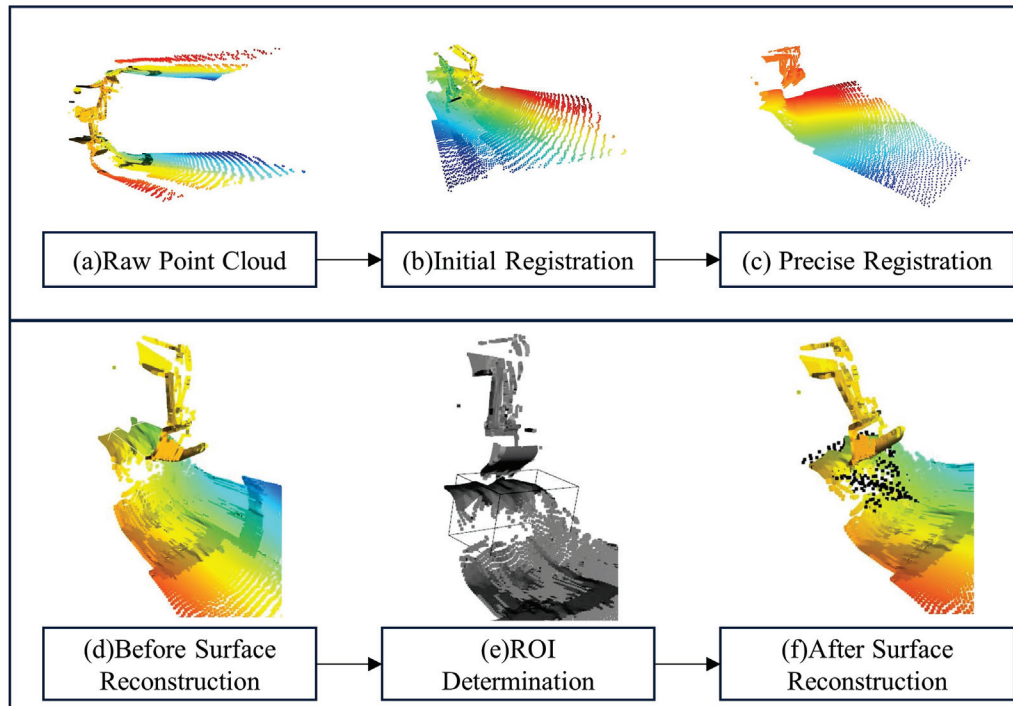


Figure 3. Terrain reconstruction based on point cloud.

In our research, we focused on the terrain changes in front of the excavator bucket before and after each scoop. In Open3D, we created an axis-aligned bounding box as the ROI to study the terrain changes in detail. As shown in Figure 3e, we defined the ROI boundaries as $(3, 2, -4)$ and $(6, 5, 1)$, with 'x+' pointing down and 'z-' to the right.

Surface reconstruction organized the points in the point cloud into a smooth surface, effectively representing the continuity and smoothness of the surface. The reconstructed model usually has uniform data density, making the surface features more consistent and continuous. The pre-reconstruction point cloud, as seen in Figure 3d, shows an uneven distribution of points within the ROI and some gaps. After reconstruction, as shown in Figure 3f, the holes and missing parts in the point cloud were filled in, adding points to the area (marked in black), transforming the sparse and incomplete point cloud into a continuous surface model, ready for further modeling operations.

4.3. Terrain Feature Extraction

Firstly, we convert the point cloud into a grid format. The grid map's width and height are calculated by dividing the real-world scene's width and height by the resolution. We initialize an empty grid map with the shape $(\text{grid_height}, \text{grid_width})$, which corresponds to the size of the grid map. In our study, we focused on a region of interest, selecting a local grid map of $(150, 120)$ with 18,000 grids, representing a $15 \text{ m} \times 12 \text{ m}$ real-world scene at a resolution of 0.1 m . We then create an array matching the grid map size, representing the empty grid map as an all-zero array, initializing each grid value to zero. The point cloud 'lidar' is then iterated over, mapping the points onto the grid map, uniformly stretching them to fit the grid map. The grid map values are the z-coordinates sampled within each grid's range, thus reducing the three-dimensional point cloud to a two-dimensional

grid map. This pseudo-image, equivalent to a single-channel grayscale image, is then dimensionally expanded by duplication to meet the model's input requirements, unlike RGB images which have three channels.

Next, we use the MobileNetV2 network as the backbone for GridNet to perform deep learning feature extraction. We select the network's last layer, extracting image features as a (512, 1) vector, which is then passed through a linear layer to generate a (256, 1) feature vector. This vector encapsulates the basic features of the terrain state, providing a compact yet information-rich representation. These vectors are particularly suitable as input states for the RL algorithm in the next step of the unmanned excavator's path generation. Compared to other networks mentioned, MobileNetV2 requires fewer computational resources, has a lower parameter count, and operates efficiently, making it suitable for use on mobile devices. In the final grid information extraction, MobileNetV2 processes a total of 300 scoops, including the first and middle scoops. The extracted grid feature information and state feature information are then input into the reinforcement learning model.

This section details training and testing the Decision Transformer (DT) [19], an offline RL algorithm, for generating excavator trajectories. The DT model handles sequence lengths up to 40 and segments up to 200, accommodating the unpredictable data lengths in real-world applications. Its linear embedding layer, operating in a 512-dimensional space, and a decoder with three layers and eight attention heads each, allow for detailed input representation. The model uses a tanh function for action outputs within a continuous range and employs a 0.1 dropout rate for robustness against overfitting. ReLU activations facilitate learning complex patterns. Training datasets with varied segment lengths are batch-sampled and preprocessed. Actions are scaled between -1 and 1 , with states and features standardized. The AdamW optimizer, with a learning rate and weight decay of 1×10^{-3} , progressively increases the rate over 100 iterations. A custom loop minimizes the error between predicted and actual actions across 500 iterations of 100 steps each.

The trend in the training loss statistics for the algorithm is indicative of its learning performance. As shown in Figure 4, the mean training loss (a) decreases significantly in the initial steps, suggesting rapid learning and model improvement. The loss continues to gradually decrease, stabilizing towards the end of the training process, which is a sign of the model converging to a solution. The standard deviation of the training loss (b) exhibits a few peaks, which indicates variability in the loss across different training batches. However, similar to the mean, the standard deviation generally decreases over time, implying that the model's predictions are becoming more consistent as training progresses.

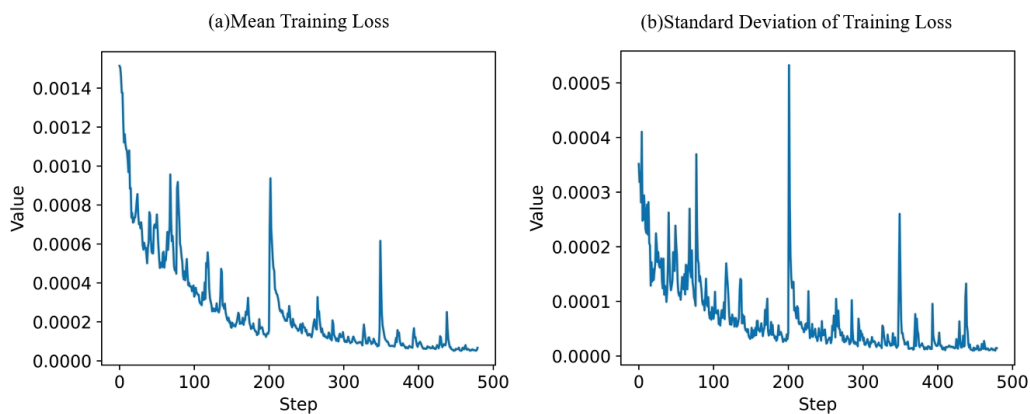


Figure 4. Model training loss.

5. Results

5.1. Decision-Making Results

The excavator's decision-making process is a real-time and multi-channel processing dynamic system, as shown in Figure 5. It involves three primary channels: the terrain

reconstruction channel (Channel 1), the excavator control channel (Channel 2), and the DRL model operation channel (Channel 3).

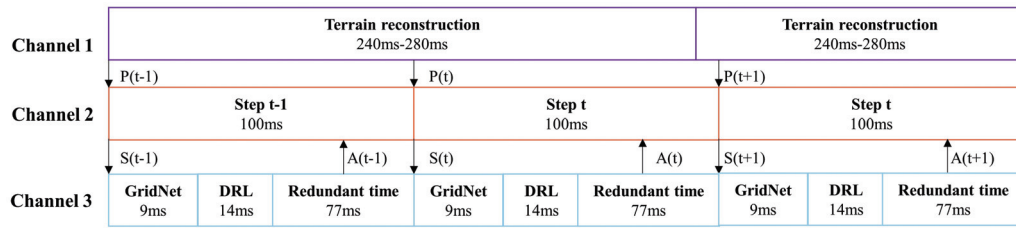


Figure 5. Real-time decision-making process.

Channel 1: This channel employs two LIDAR scanners to capture point cloud data $P(t)$, which are then used to create a 3D reconstruction of the terrain using the algorithm described in Section 4.2. The processing time for terrain reconstruction ranges from 240 ms to 280 ms.

Channel 2: Operating at a frequency of 10 Hz, this channel executes every 100 ms. At each time step t , it receives the point cloud information $P(t)$ from the terrain reconstruction channel and the current joint information of the excavator, combining them to form the state $S(t)$, which encompasses the immediate conditions of both the environment and the excavator.

Channel 3: The state $S(t)$ is relayed to the DRL model operation channel where it first undergoes processing by the GridNet, taking 9 ms, followed by decision making through the DRL model, which takes 14 ms. Ultimately, the DRL model outputs an action $A(t)$, a vector containing increments for four joint angles, to guide the excavator's subsequent movement.

The complete operational results of the E-GTN are depicted in Figure 6. Within the simulated environment, the excavator's bucket has a capacity of 4.5 cubic meters. Under the implementation of our method, coupled with the GridNet's terrain feature extraction, the system achieved an average excavation volume of 3.7 cubic meters per scoop with a standard deviation of 0.27 cubic meters. In contrast, without the feature extraction, the average excavation volume dropped significantly to 0.45 cubic meters per scoop, with a standard deviation of 0.46 cubic meters. These findings underscore the efficacy of integrating GridNet in enhancing the precision of the excavation process, optimizing the excavator's operational capacity, and highlighting the importance of terrain feature extraction in the decision-making process.

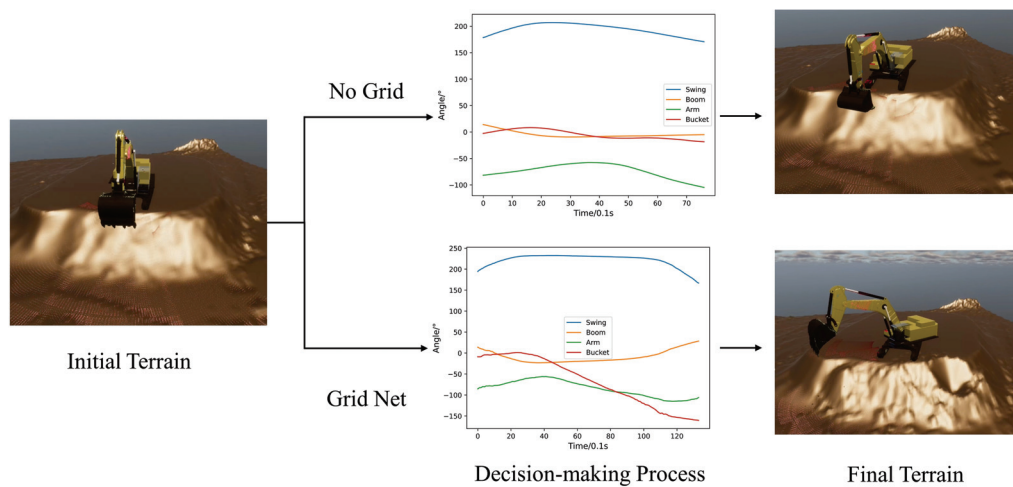


Figure 6. Excavation result based on EGTN.

5.2. Feature Extraction Results

Upon completing the training, we extracted the terrain features before and after excavation and input them into our GridNet. The visualized results are shown in Figure 7. A comparison between the feature maps before and after training reveals a noticeable

difference: post-training, there is an increased level of feature activation, indicating that the network's ability to extract features has improved.

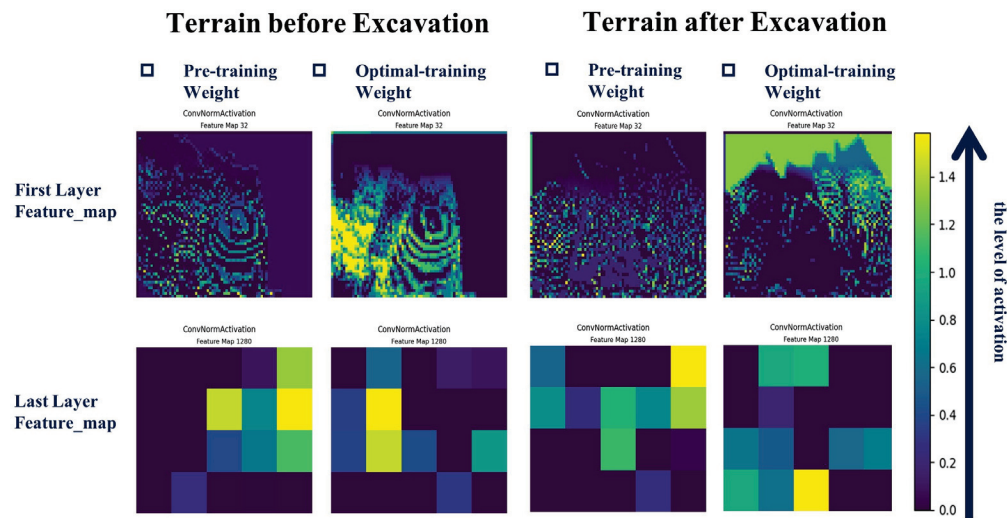


Figure 7. Comparison of feature map activations.

The images on the left represent the terrain before excavation, while the images on the right depict the terrain after excavation. In both cases, the feature maps are shown for pre-training and optimal-training weights. For the first layer feature maps, the pre-training images are more chaotic and less defined, while the optimal-training images show more distinct patterns and clearer feature representations. This suggests that the network has learned to emphasize relevant features for the task at hand.

Similarly, the last layer feature maps exhibit a stark contrast. The pre-training images are almost uniform with little to no distinctive features, whereas the optimal-training images display well-defined, high-activation regions. This signifies that the network, after training, can better identify and highlight the most critical features of the terrain.

The scale on the right indicates the level of activation, with higher values corresponding to stronger activations. The optimal-training feature maps generally show higher activation levels compared to the pre-training ones, which is a strong indicator of the network's enhanced feature extraction capability after training.

In our study, we employed principal component analysis (PCA) as a preliminary method to assess the effectiveness of feature extraction by GridNet. Figure 8b presents the PCA results of the feature vectors extracted by GridNet. The first two principal components account for an explained variance ratio of 0.39 and 0.16, respectively, cumulatively explaining over 50% of the variance. The scatter plot associated with these components shows a relative dispersion of data points, lacking a clear clustering trend. This dispersion suggests that the extracted features have a high level of complexity and may capture more nuanced variations within the data.

Conversely, Figure 8a illustrates the PCA results of the raw grid data. Here, the first two principal components have an explained variance ratio of 0.25 and 0.19, which together explain less than 50% of the total variance. The scatter plot reveals a concentrated distribution of data points, indicating a strong clustering tendency. This concentration implies that the raw data may inherently contain less complexity or fewer distinct patterns before the application of feature extraction techniques.

The comparative analysis of the PCA results before and after the application of GridNet allows us to infer that the feature extraction process not only accounts for a greater proportion of the variance but also increases the complexity of the feature representation. This is indicated by the spread of data points in the scatter plot post-extraction, which contrasts with the more clustered distribution seen in the raw data. The ability of GridNet

to elucidate a higher amount of variance and to disperse the data points suggests that it is capturing more significant variables within the dataset.

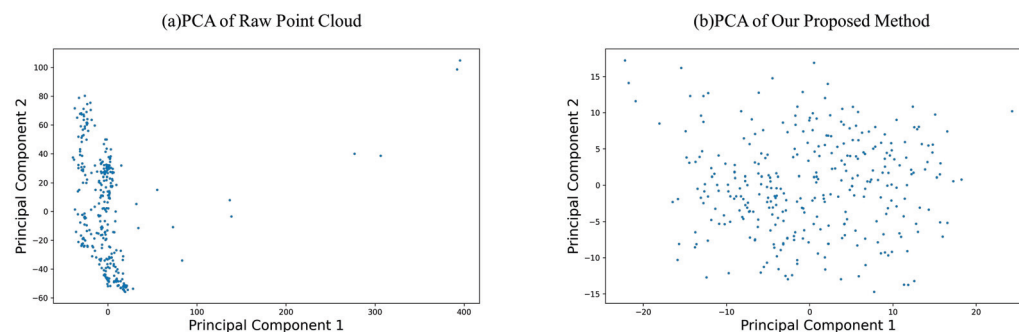


Figure 8. Comparison of feature map activations.

The feature extraction process through GridNet, as evidenced by the PCA, appears to be effective in distilling important variables from the data. This is an encouraging sign that the network can isolate and amplify aspects of the data that are most informative for subsequent analysis. The increased complexity and variance explained by the extracted features are indicative of GridNet’s robustness in representing terrain data in a way that is conducive to advanced terrain analysis and decision-making processes in automated systems.

6. Conclusions

In conclusion, our E-GTN framework has proven to be a robust solution to the challenges posed by unmanned excavator operations. By integrating advanced multisensor fusion, we have significantly enhanced the geometric representation of the terrain, which is foundational for high-fidelity environmental reconstruction. Our GridNet has successfully processed grid data to extract salient terrain features, demonstrating its critical role in the reinforcement learning tasks for autonomous excavation. The decision-making module, informed by these extracted features, has facilitated efficient and intelligent operation, showcasing the potential of our framework to revolutionize the construction machinery industry. The successful application of our E-GTN framework underscores its value as a comprehensive platform for scholars and practitioners in related fields, paving the way for future advancements in autonomous excavation technologies.

Expanding on our work, future iterations of the E-GTN framework could employ adaptive algorithms for better performance across varied soils. Real-time learning capabilities may enable self-optimizing excavators. Additionally, deploying this technology in extreme settings, like underwater or underground, could redefine automated digging. These forward-looking applications not only build on the current strengths of the E-GTN framework but also chart a course for its evolution, solidifying its role as a transformative force in the industry.

Author Contributions: Conceptualization: Q.Z.; methodology: Q.Z., D.W. and X.M.; formal analysis and investigation: Q.Z. and D.W.; writing—original draft preparation: Q.Z.; writing—review and editing: J.Q. and J.H.; funding acquisition: L.G. and J.H.; resources: L.G.; supervision: J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (52035007, U23B20102), Ministry of Education “Human Factors and Ergonomics” University Industry Collaborative Education Project (No. 202209LH16) and the Xie Yubai Design Scientific Research Foundation (XYB-DS-202401).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available on request to the authors.

Acknowledgments: We would like to extend our sincere thanks to Sany Company for their support and contribution to this research.

Conflicts of Interest: Author Le Gao was employed by the company Sany Heavy Machinery Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hemami, A.; Hassani, F. An overview of autonomous loading of bulk material. In Proceedings of the 26th International Symposium on Automation and Robotics in Construction, Austin, TX, USA, 24–27 June 2009; International Association for Automation and Robotics in Construction (IAARC): Austin, TX, USA, 2009; pp. 405–411.
2. Dadhich, S.; Bodin, U.; Andersson, U. Key challenges in automation of earth-moving machines. *Autom. Constr.* **2016**, *68*, 212–222. [CrossRef]
3. Zhang, L.; Zhao, J.; Long, P.; Wang, L.; Qian, L.; Lu, F.; Song, X.; Manocha, D. An autonomous excavator system for material loading tasks. *Sci. Robot.* **2021**, *6*, eabc3164. [CrossRef] [PubMed]
4. IndustryResearch. Global Excavator Market Report. 2020. Available online: <https://www.industryresearch.co/global-excavator-market-18836753> (accessed on 30 March 2024).
5. Afshar, R.R.; Zhang, Y.; Vanschoren, J.; Kaymak, U. Automated reinforcement learning: An overview. *arXiv* **2022**, arXiv:2201.05000.
6. Stentz, A.; Bares, J.; Singh, S.; Rowe, P. A robotic excavator for autonomous truck loading. *Auton. Robot.* **1999**, *7*, 175–186. [CrossRef]
7. Yamamoto, H.; Moteki, M.; Ootuki, T.; Yanagisawa, Y.; Nozue, A.; Yamaguchi, T. Development of the autonomous hydraulic excavator prototype using 3-D information for motion planning and control. *Trans. Soc. Instrum. Control. Eng.* **2012**, *48*, 488–497. [CrossRef]
8. Chae, M.J.; Lee, G.W.; Kim, J.Y.; Park, J.W.; Cho, M.Y. A 3D surface modeling system for intelligent excavation system. *Autom. Constr.* **2011**, *20*, 808–817. [CrossRef]
9. Shariati, H.; Yeraliyev, A.; Terai, B.; Tafazoli, S.; Ramezani, M. Towards autonomous mining via intelligent excavators. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 26–32.
10. Forkel, B.; Kallwies, J.; Wuensche, H.J. Probabilistic terrain estimation for autonomous off-road driving. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 13864–13870.
11. Foley, T.; Sugerman, J. KD-tree acceleration structures for a GPU raytracer. In Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware, Los Angeles, CA, USA, 30–31 July 2005; pp. 15–22.
12. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 3212–3217.
13. Chetverikov, D.; Svirko, D.; Stepanov, D.; Krsek, P. The trimmed iterative closest point algorithm. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec, QC, Canada, 11–15 August 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 3, pp. 545–548.
14. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson surface reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Italy, 26–28 June 2006; Volume 7.
15. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
16. Parkes, D.C.; Singh, S. An MDP-based approach to online mechanism design. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*; MIT Press: Cambridge, MA, USA, 2003.
17. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An introduction to deep reinforcement learning. *Found. Trends® Mach. Learn.* **2018**, *11*, 219–354. [CrossRef]
18. SY750H | Large Excavator. 2024. Available online: https://www.sanyglobal.com/product/excavator/large_excavator/115/847/ (accessed on 30 March 2024).
19. Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15084–15097.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Trajectory Planning of Robotic Arm Based on Particle Swarm Optimization Algorithm

Nengkai Wu, Dongyao Jia *, Ziqi Li and Zihao He

School of Automation and Intelligence, Beijing Jiaotong University, Beijing 100044, China;
19111054@bjtu.edu.cn (N.W.); 21111057@bjtu.edu.cn (Z.L.); 22115032@bjtu.edu.cn (Z.H.)

* Correspondence: dongyaojia1974@163.com

Abstract: Achieving vibration-free smooth motion of industrial robotic arms in a short period is an important research topic. Existing path planning algorithms often sacrifice smoothness in pursuit of efficient motion. A robotic trajectory planning particle swarm optimization algorithm (RTPPSO) is introduced for optimizing joint angles or paths of mechanical arm movements. The RTPPSO algorithm is enhanced through the introduction of adaptive weight strategies and random perturbation terms. Subsequently, the RTPPSO algorithm is utilized to plan selected parameters of an S-shaped velocity profile, iterating to obtain the optimal solution. Experimental results demonstrate that this velocity planning algorithm significantly improves the acceleration of the robotic arm, surpassing traditional trial-and-error velocity planning methods.

Keywords: mechanical arm; trajectory planning; particle swarm optimization; optimization algorithms; motion control

1. Introduction

The motion control of industrial manipulators typically involves a planning layer and a control layer. The former mainly focuses on understanding the relationship between the manipulator's position, speed, and time to realize the operation. Meanwhile, the control layer addresses how to ensure the actuator effectively follows the planned trajectory. Therefore, trajectory planning consists of basic research in the field of manipulator control. This has an important impact on the working efficiency, motion stability, and energy consumption of the manipulator [1]. Optimal trajectory planning aims to devise the optimal trajectory through task points, considering the predefined execution task points and motion constraints. The general trajectory planning algorithm transforms the original path into an optimized trajectory by planning parameters, such as the position, velocity, and acceleration for each task point on the manipulator's operational path. This enhances the accuracy and stability of the manipulator's trajectory [2]. Therefore, the purpose of this paper is to control the speed and acceleration of the manipulator during operation, aiming to enhance stability and reduce the impact and wear of the manipulator.

In recent years, trajectory planning of industrial manipulators has been considered an important research focus in the field of robot control. Moreover, the research for a manipulator's optimal path planning has also seen important progress. In more detail, interpolation algorithms consist of a common method in manipulator trajectory planning due to their advantages, including simple calculation and easy implementation [3–5]. For instance, Shi et al. [6] utilized an interpolation algorithm grounded in the forward-looking function to fine-tune the feed speed. This was achieved by predicting the future change in trajectory curvature. This approach not only ensured the precision of the interpolation algorithm but also enhanced processing efficiency. Moreover, Luo et al. [7] considered the use of a Lagrange interpolation function as the trajectory function for individual joints to optimize the energy consumption of an industrial manipulator and derive the optimal trajectory. In addition, Nadir et al. [8] used multiquadric radial basic functions to generate

the manipulator's motion trajectory through the minimization of the acceleration integral and the consideration of the boundary conditions and the kinematic limits. Compared to other methods, this algorithm has higher smoothness. However, its accuracy is affected by the number of interpolation functions and interpolation nodes; consequently, it may not be effective when dealing with complex trajectory curves.

Moreover, the swarm intelligent control algorithm is a heuristic algorithm developed to solve the motion path of a manipulator through the simulation of the swarm intelligent behavior in nature. This latter is characterized by its strong adaptability and ability to handle complex motion trajectory, including the genetic algorithm (GA) [9–11], the artificial immune algorithm [12], and the particle swarm optimization (PSO) algorithm [13,14].

The core idea of the PSO algorithm consists of finding the optimal solution of the problem by simulating the flock predation process and constantly adjusting the speed and position of the particles [15]. Compared to other similar algorithms, PSO has high efficiency, global optimization ability, adaptability, and expansibility. However, the traditional PSO algorithm is prone to premature convergence. Therefore, several researchers have improved the algorithm's performance by adjusting relevant parameters and combining them with other intelligent algorithms. For instance, Lv et al. [16] proposed an improved PSO algorithm combining the fuzzy reward and punishment theory to resolve the manipulator's trajectory planning problem when having multiple Degrees of Freedom (DoF). Therefore, the global search ability of the PSO algorithm is enhanced by adding a population exchange term, and the last elimination principle is applied to avoid falling into the local optimum. Moreover, Li et al. [17] adopted PSO as the optimizer algorithm to solve the vibration suppression problem of a flexible linkage manipulator and enhanced the efficiency and accuracy of trajectory planning. In addition, Du et al. [18] introduced chaotic sequence as a local search strategy in the PSO algorithm and employed a piecewise polynomial interpolation function to approximate the optimal trajectory. These improvements enhanced the search efficiency and optimization performance of the algorithm and were able to effectively shorten the running time of the manipulator. Moreover, they ensured the stability of the movement. Added to that, Yang et al. [19] introduced an innovative time optimization strategy, known as a Second-Order Continuous Polynomial Interpolation Function (SCPIF). This approach, coupled with the cosine subweighted PSO, addresses the challenge of time-optimal trajectory optimization.

Although the above research has made some achievements, there are still many shortcomings. Firstly, the existing particle swarm optimization algorithms [20,21] are often only applicable to specific types of problems; however, they lack universality in other categories. In addition, the performance of existing algorithms still depends on the choice of parameters, such as weight factors and inertia weights. Therefore, the optimal value needs to consider the actual situation of the industry. In the practical utilization process, in dealing with nonlinear optimization problems, the existing algorithms will fall into local optimal solutions, resulting in performance degradation.

As a result, considering the industrial requirements of the manipulator in practical application processes, as well as the limiting factors including the position, speed, and acceleration of each task point, a new control strategy with high speed and high stability is proposed to achieve the manipulator's functional requirements. Moreover, the algorithm will be applied for transporting liquid reagents. It is worth noting that the proposed velocity optimal trajectory optimization strategy has a high stability level. Compared to other improved PSO algorithms, it is easily implemented and applicable to a variety of scenarios.

To sum up, the contributions of this paper are as follows:

1. A new Robot Trajectory Planning Particle Swarm Optimization (RTPPSO) algorithm is proposed in this paper;
2. The PSO algorithm is applied to the trajectory planning and parameter optimization for each robot manipulator joint.

2. Trajectory Planning

2.1. Overview of Trajectory Planning

Trajectory planning typically focuses on two research areas: mobile robots and industrial robots. In the context of the mobile robot, trajectory planning mainly refers to the path trajectory planning of the robot. For example, whether the robot has a map or not, the kind of path trajectory to follow is determined at this level. For industrial robots, trajectory planning encompasses two directions. Firstly, it involves planning the curvilinear trajectory of the end of the manipulator, detailing the curvilinear profile of the manipulator during motion displacement, velocity, and acceleration. As for the trajectory planning, it is also divided into Cartesian space planning and joint space planning according to the coordinate system reference. In more detail, the joint space planning is mainly oriented toward a joint manipulator. In this configuration, the displacement, the angular displacement, the velocity, and the acceleration of each joint of the manipulator are used to compute the time function. Furthermore, the desired pose of the end actuator is achieved by computing the relevant parameters. However, the calculation process is simpler and does not cause singularity problems at the level of the manipulator. Yet, the obtained results after using the joint space planning are difficult to be transferred to the three-dimensional space, and the posture of the manipulator will change along with the velocity. The second representation is the Cartesian space planning. It is usually deployed in the end actuator to achieve straight lines, arcs, and other determined curves; moreover, it can intuitively observe the motion trajectory of the end of the manipulator.

In trajectory planning, the equations modelling the displacement, velocity, acceleration, and other variables are set according to the time parameters, and the function relationship between the displacement of each joint and time can be computed through the inverse kinematics equation of the manipulator.

Moreover, in this work, a four-DoF manipulator is analyzed in which three DoF comprise translational pairs. Each step of the manipulator is a motion from the starting point to the end point, that is, the trajectory planning from the start to the end points, and the parameters such as displacement, velocity, and acceleration must meet the smoothness and continuity characteristics. In this work, the Cartesian space planning algorithm is deployed to plan the trajectory of the four-DoF manipulator described earlier.

2.2. Pose Planning

Cartesian space trajectory planning includes both pose planning and velocity planning. This is realized through the design of the position of the robot end manipulator and the combination of the acceleration and deceleration control models.

The pose planning of the manipulator consists of solving the function of the pose of the end manipulator in the workspace with respect to time. The basic idea is described below. Firstly, the path sequence in Cartesian space is obtained, and the interpolation algorithm is applied to realize it. Consequently, the inverse kinematic operation is performed on the space coordinates for each joint constituting the manipulator. This operation maps the position and the attitude of a given point in Cartesian space to the position and attitude of each joint in joint space. Two commonly employed spatial interpolation methods are line interpolation and arc interpolation.

Trajectory planning in Cartesian space allows the determination of the specific path of the manipulator's end effector in 3D space. This path is defined by the specific task assigned to the manipulator.

During line interpolation, the starting point $P_1(x_1, y_1, z_1)$ and the ending point $P_2(x_2, y_2, z_2)$ of the terminal manipulator are expressed as follows:

$$s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

$$\begin{cases} x_k = x_1 + \lambda_k(x_2 - x_1) \\ y_k = y_1 + \lambda_k(y_2 - y_1) \\ z_k = z_1 + \lambda_k(z_2 - z_1) \end{cases} \quad (2)$$

where λ_k is the normalization factor ($\lambda_k \in [0, 1]$) and s denotes the straight-line distances between two points.

In the arc interpolation, for any not-on-a-line three points $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, and $P_3(x_3, y_3, z_3)$ expressed in the Cartesian coordinate system, the steps that apply the arc planning for these three points are defined as follows:

Step 1: According to the coordinates of the three points in space, the curve equation of the arc can be calculated with reference to the placement of these three. Then, the center coordinate $P_0(x_0, y_0, z_0)$ and radius r of the arc are generated. Let the equation of the plane where P_1 , P_2 , and P_3 are located be $Ax + By + Cz + D = 0$.

Step 2: Set up the relative coordinate system $o' - x'y'z'$ for the plane of the space arc within the absolute coordinate system $o - xyz$. The origin o' of this relative coordinate system is the center P_0 of the arc. Let the direction of the axis x' be P_0P_1 , as depicted in Figure 1. According to coordinates P_0 and P_1 , the direction cosine of the axis x' in the relative coordinate system can be expressed as follows:

$$\vec{n} = \left[\frac{x_1 - x_0}{r} \frac{y_1 - y_0}{r} \frac{z_1 - z_0}{r} \right]^T \quad (3)$$

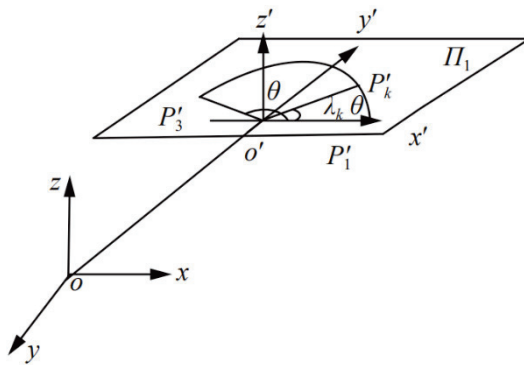


Figure 1. Circular interpolation based on relative coordinate system.

Consequently, the direction of the relative coordinate system z' axis is $\vec{P}_1P_2 \times \vec{P}_2P_3$, and its direction cosine is expressed as follows:

$$\vec{a} = \left[\frac{A}{k} \frac{B}{k} \frac{C}{k} \right]^T \quad (4)$$

where $k = \sqrt{A^2 + B^2 + C^2}$.

According to directions x' and z' , the direction of axis y' can be defined as $\vec{o} = \vec{a} \times \vec{n}$. Let $\vec{p} = [x_0 \ y_0 \ z_0]^T$, then the homogeneous transformation matrix T from the relative coordinate system $o' - x'y'z'$ to the absolute coordinate system $o - xyz$ can be obtained according to the homogeneous transformation principle represented in Equation (5):

$$T = \begin{bmatrix} \vec{n} & \vec{o} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

Let the homogeneous coordinate of any point P in the absolute coordinate system be $[x \ y \ z \ 1]^T$; then, the coordinate of point P in the relative coordinate system $o' - x'y'z'$ is $[x' \ y' \ z' \ 1]^T$. As a result, one can write

$$[x' \ y' \ z' \ 1]^T = T^{-1} [x \ y \ z \ 1]^T \quad (6)$$

Step 3: Find the angle θ between the starting point and the ending point of the arc. Referring to Equation (6), the corresponding coordinates of the three points P_1 , P_2 , and P_3 in space using the relative coordinate system are denoted as $P_1'(x_1', y_1', 0)$, $P_2'(x_2', y_2', 0)$, and $P_3'(x_3', y_3', 0)$.

Moreover, let the position angle at the end of the whole arc trajectory interpolation process be θ . The following is available:

$$\theta = \begin{cases} \arctan2(y_3', x_3') + 2\pi, y_3' < 0; \\ \arctan2(y_3', x_3'), y_3' \geq 0. \end{cases} \quad (7)$$

Step 4: Design the plane in the $x'y'$ plane of the relative coordinate system. Based on the homogeneous transformation principle, the following system is obtained:

$$\begin{cases} \theta_k = \lambda_k \cdot \theta \\ x_i' = r \cdot \cos \theta_k \\ y_i' = r \cdot \sin \theta_k \\ z_i' = 0 \end{cases} \quad (8)$$

Then, the interpolation calculation of the circular arc motion can be completed by the S-shaped velocity curve planning for λ_k . According to the constraint conditions, such as the arc length and the velocity, they can be divided into several cases, and the linear part can be referred to.

2.3. Velocity Planning

Point-to-point velocity planning can be conceptualized as a method for transitioning from a start point to an end point within a specified time. Currently, widely used velocity planning algorithms in motion planning encompass the ladder-type velocity planning algorithm and the S-type velocity planning algorithm. The trapezoidal curve planning algorithm is characterized by its simplicity, minimal time requirement, swift motor response, and high efficiency. However, it exhibits discontinuous acceleration, leading to a significant flexible impact and relatively lower motion accuracy.

Moreover, the S-shaped curve planning maintains continuous acceleration at any position, thereby avoiding the flexible impact observed in trapezoidal planning. This results in a smoother velocity profile and higher motion accuracy. Nonetheless, the S-shaped algorithm is more complex than trapezoidal programming. Both velocity curve planning approaches are introduced in the following paragraphs.

Trapezoidal curve trajectory planning, also known as linear acceleration and deceleration or T-shaped acceleration and deceleration, stands out as the most widely employed speed control method in manipulator control. Its simple structure and ease of calculation make it a prevalent choice in several motor systems. The acceleration and deceleration process of the trapezoidal velocity curve involves uniform acceleration and uniform deceleration, with the acceleration value maintained as a constant.

In trapezoidal curve planning, the velocity curve can be determined by specifying the starting and ending points of motion, the adding or subtracting of velocity, and the maximum velocity. This process is illustrated in Figure 2.

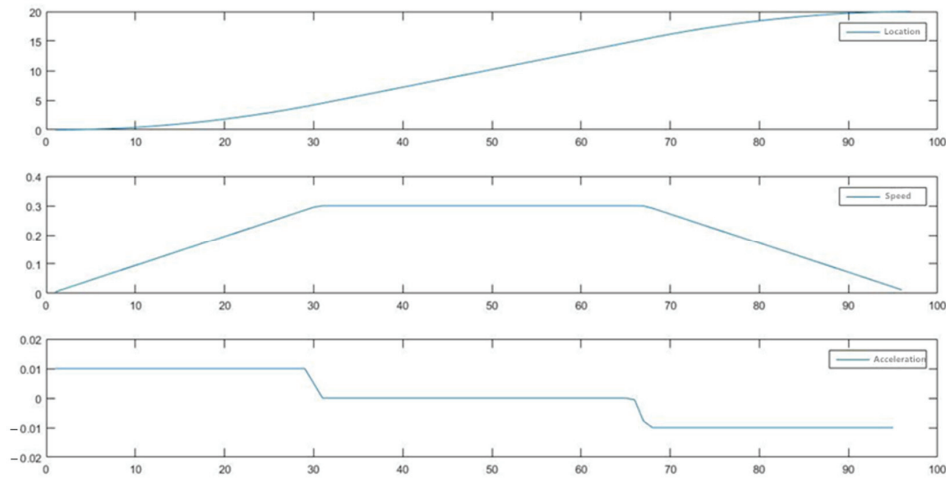


Figure 2. Trapezoidal speed planning.

The velocity curve of the trapezoidal velocity planning consists of three segments. The first one is a uniform acceleration, the second stage is a uniform velocity, and the third stage is a uniform deceleration. Notably, the rate of increase and decrease in speed during the rise and fall is the same, ensuring that the duration of both the uniform acceleration and deceleration stages is identical. This motion process can be expressed as a piecewise function, and its motion formula is given by

$$\begin{cases} s(t) = \frac{1}{2}at^2, 0 \leq t \leq \frac{v}{a} \\ s(t) = vt - \frac{v^2}{2a}, \frac{v}{a} \leq t \leq T - \frac{v}{a} \\ s(t) = \frac{2avT - 2v^2 - a^2(t-T)^2}{2a} \end{cases} \quad (9)$$

where T represents the total motion time; s represents position; a represents acceleration; and t represents time. Referring to Figure 2, it becomes evident that the acceleration curve in the trapezoidal velocity planning takes on a step-like form. It is notable that the acceleration is not continuous at the four time points: motion start time, acceleration end time, deceleration start time, and movement end time. This lack of continuity significantly affects the stationarity of the manipulator.

The S-shaped velocity curve is another widely employed velocity planning algorithm in the realm of mechanical control. Diverging from trapezoidal velocity planning, it addresses certain shortcomings associated with the trapezoidal velocity. Moreover, the change rate of acceleration can be controlled by S-type velocity planning. Notably, the acceleration curve in S-type velocity planning is continuous and devoid of sudden acceleration. Consequently, S-type velocity planning emerges as a control strategy that effectively limits vibration.

Furthermore, the S-shaped velocity planning is derived from the S-shaped velocity curve in the process of movement. The velocity curve of the most common S-shaped velocity planning is divided into seven segments as follows:

The first segment: increase the acceleration from zero to the preset a value with the constant jerk J .

The second segment: accelerate the operation with a constant acceleration a .

The third segment: reduce the acceleration from the preset value to reach zero with a constant negative jerk J .

The fourth segment: achieve with a constant speed v a uniform motion.

The fifth segment: reduce the acceleration from zero to the preset value $-a$ with a constant negative jerk J .

The sixth segment: decelerate with a constant acceleration $-a$.

The seventh segment: increase the acceleration from the preset $-a$ value to zero with a constant jerk J .

Compared to the three-stage planning of the trapezoidal velocity curve, the S-shaped curve is considerably more complex. By strategically planning acceleration in the acceleration stage, deceleration stage, and acceleration and deceleration stage, S-shaped curve planning achieves superior acceleration continuity compared to trapezoidal curve planning. This substantial improvement significantly impacts during the process of mechanical operation. Therefore, Figure 3 provides a schematic diagram of S-shaped velocity planning.

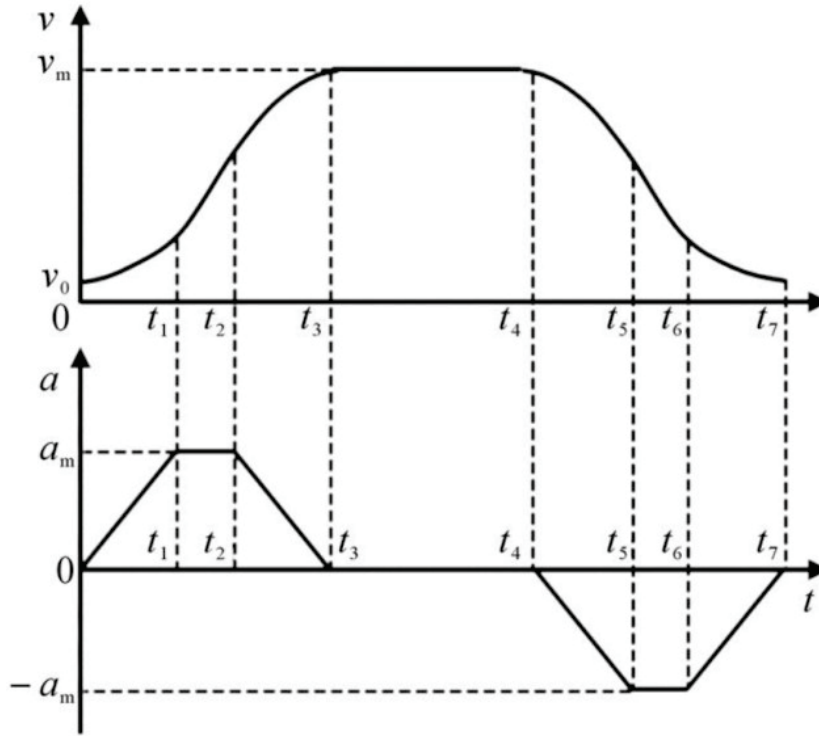


Figure 3. S-shaped speed planning.

For the S-shaped velocity curve, given the maximum acceleration $a_{\max}$ and the jerk J , the corresponding acceleration of each stage of the seven stages described previously is defined as follows:

$$a(t) = \begin{cases} Jt, 0 \leq t \leq t_1 \\ a_{\max}, t_1 \leq t \leq t_2 \\ a_{\max} - J(t - t_2), t_2 \leq t \leq t_3 \\ 0, t_3 \leq t \leq t_4 \\ -J(t - t_4), t_4 \leq t \leq t_5 \\ -a_{\max}, t_5 \leq t \leq t_6 \\ -a_{\max} + J(t - t_6), t_6 \leq t \leq t_7 \end{cases} \quad (10)$$

Thus, the velocity function corresponding to each stage can be obtained as follows:

$$v(t) = \begin{cases} \frac{1}{2}Jt^2, 0 \leq t \leq t_1 \\ v_1 + a_{\max}(t - t_1), t_1 \leq t \leq t_2 \\ v_2 + a_{\max}(t - t_2) - \frac{1}{2}J(t - t_2)^2, t_2 \leq t \leq t_3 \\ v_3, t_3 \leq t \leq t_4 \\ v_4 - \frac{1}{2}J(t - t_4)^2, t_4 \leq t \leq t_5 \\ v_5 - a_{\max}(t - t_5), t_5 \leq t \leq t_6 \\ v_6 - a_{\max}(t - t_6) - \frac{1}{2}J(t - t_6)^2, t_6 \leq t \leq t_7 \end{cases} \quad (11)$$

According to the time, speed, acceleration, and jerk of each stage, the final position can be expressed as follows:

$$s(t) = \begin{cases} \frac{1}{6}Jt^3, 0 \leq t \leq t_1 \\ s_1 + v_1(t - t_1) + \frac{1}{2}a_{\max}(t - t_1)^2, t_1 \leq t \leq t_2 \\ s_2 + v_2(t - t_2) + \frac{1}{2}a_{\max}(t - t_2)^2 - \frac{1}{6}J(t - t_2)^3, t_2 \leq t \leq t_3 \\ s_3 + v_3(t - t_3), t_3 \leq t \leq t_4 \\ s_4 + v_4(t - t_4) - \frac{1}{6}J(t - t_4)^3, t_4 \leq t \leq t_5 \\ s_5 + v_5(t - t_5) + \frac{1}{2}a_{\max}(t - t_5)^2, t_5 \leq t \leq t_6 \\ s_6 + v_6(t - t_6) - \frac{1}{2}a_{\max}(t - t_6)^2 + \frac{1}{6}J(t - t_6)^3, t_6 \leq t \leq t_7 \end{cases} \quad (12)$$

In the S-shaped curve, the speed determines the speed of the system, whereas the acceleration defines the speed of the system's acceleration and deceleration. The jerk determines the flexibility of the system; the greater the jerk is, the greater the impact caused by the operation of the manipulator will be. Assuming that the urgency is infinite, the S-shaped velocity curve is switched into a trapezoidal velocity curve.

Furthermore, the S-shaped velocity curve planning consists of planning the parameters relative to the jerk, the acceleration, the velocity, and so on, in each stage to realize a flexible operation in the manipulator. In practical application, the S-shaped velocity curve planning can also be removed from uniform acceleration and deceleration to reduce the number of stages to five. Furthermore, the uniform velocity can also be removed, yielding a total of four stages.

3. Improved Particle Swarm Optimization Algorithm

3.1. Particle Swarm Optimization

PSO, proposed by Eberhart and Kennedy in 1995, serves as a heuristic global search algorithm rooted in swarm intelligence, mimicking the foraging behavior of birds. The essence of the PSO algorithm lies in leveraging all individuals in the population to share information. This facilitates the movement of the entire population in the problem-solving space, transitioning from disorder to order in an evolutionary process, ultimately leading to the acquisition of the optimal solution for the processed problem. The algorithm is known for its simplicity, minimal parameter settings, ease of implementation, and robust global search ability. As a result, it has garnered attention both from researchers and in engineering fields, finding widespread applications.

Furthermore, PSO employs a particle to simulate the bird actions mentioned above. Each particle carries out automatic search in a multidimensional space, and the particle has only two attributes: velocity and position. Therefore, the optimal solution of a single particle is called the individual extreme value whose optimal represents the global optimal solution of this iteration. At the end of each iteration, the particle velocity and position are updated. Finally, the iteration is stopped whenever reaching the maximum number of iterations or the required fitness level; consequently, the optimal solution of the system is obtained. As for the operation process of the algorithm, it is divided into six steps, which are described below.

Step 1: Particle initialization. At this level, a random number is assigned to the position and velocity of each particle. Moreover, the maximum number of iterations, the number of independent variables of the objective function, and the maximum velocity of the particle are set.

Step 2: Whenever a particle updates its position and velocity, its fitness function determines its fitness based on its movement state.

Step 3: The individual optimal solution of each particle is updated. Each particle individual optimal solution is continuously recorded. When the fitness exceeds the individual optimal solution, this latter will be updated.

Step 4: During the learning process of each particle, the global best solution is updated. By comparing the individual best solutions of all particles, the current global best solution

is determined and compared with the historical global best solution. If it is better than the historical global best solution, the latter will be updated.

Step 5: All particles' velocities are updated with the current velocity, individual best, and global best as in Equation (13), and the next position of each particle is updated with the current position and velocity as in Equation (14).

$$V(t+1) = w \times V(t) + c_1 \times r_1 \times (pb - X(t)) + c_2 \times r_2 \times (gb - X(t)) \quad (13)$$

$$X(t+1) = X(t) + V(t+1) \quad (14)$$

In the equation, w represents the inertia weight; c_1 and c_2 represent the cognitive and social learning factors of the particle, respectively; r_1 and r_2 are random numbers between 0 and 1; pb and gb denote the individual best solution and global best solution of the particle, respectively; and V and X represent the velocity and position of the particle, respectively.

Step 6: Termination conditions are established. Different termination conditions are set for diverse problems. The set number of iterations is reached, and the difference between the computation meets the minimum bound.

3.2. Algorithm to Improve

The PSO algorithm has achieved good results in many different scenarios and practical applications. However, the problem of easily falling into local optima has not been well improved. To address this issue, we introduce a local disturbance term and attempt to use adaptive weights to improve the algorithm. The random disturbance term allows particles to jump out of the current region when they fall into the optimum in the early stage, avoiding the situation of falling into local optima. The adaptive weight coefficient adaptively adjusts the weights by calculating the current particle state. The RTPPSO algorithm modifies the PSO algorithm as follows.

To prevent particles from getting trapped in local optimal solutions, a random perturbation term is introduced. This technique aids particles in steering clear of local optima and effectively delving into potential global optima. Specifically, the updated particle position is subject to a random perturbation term, which is a vector crafted in alignment with a designated random distribution. This random perturbation term could follow a normal distribution, a uniform distribution, or another pertinent distribution. Consequently, each particle incorporates an element of randomness, thereby enhancing the diversity of the search. Thus, the particle velocity update formula is as follows:

$$V(t+1) = w \times V(t) + c_1 \times r_1 \times (pb - X(t)) + c_2 \times r_2 \times (gb - X(t)) + \alpha \times D \quad (15)$$

where α is the random disturbance coefficient, which controls the intensity of introduced randomness, and D is a random number between -1 and 1 .

Referring to Equation (16), it generates the updating principle of the inertia weight. In the equation, $w_{\min}$ and $w_{\max}$ represent the minimum and maximum values of the custom weights, respectively. $f(t)$ represents the fitness value of the current particle, while $f_{\min}$ and f_{avg} denote the minimum fitness value and average fitness value of the current particle swarm, respectively. The fitness value is calculated by the subsequent objective function value. These values are normalized; thus, they vary between 0 and 1.

$$w(t) = w_{\max} - (w_{\max} - w_{\min}) \times (f(t) - f_{\min}) / (f_{\text{avg}} - f_{\min}) \quad (16)$$

Moreover, Equation (15) can be updated and results in Equation (17):

$$V(t+1) = w(t) \times V(t) + c_1 \times r_1 \times (pb - X(t)) + c_2 \times r_2 \times (gb - X(t)) + \alpha \times D \quad (17)$$

Here, we also introduce adaptive weight changes for the parameters c_1 and c_2 in the particle velocity calculation formula, as shown in Equation (18). In this equation, $c_1(t)$ and $c_2(t)$ represent newly designed adaptive parameters; α and β are constants ranging from 0 to 1.

$$\begin{cases} c_1(t) = \alpha((f(t) - f_{\min}) / (f_{\text{avg}} - (f_{\min}))) + \beta \\ c_2(t) = \alpha(1 - (f(t) - f_{\min}) / (f_{\text{avg}} - (f_{\min}))) + \beta \end{cases} \quad (18)$$

The RTPPSO algorithm has undergone significant changes compared to the traditional PSO algorithm in the formula for updating particle velocities, and its flow chart is shown in Figure 4. All parameters have shifted from their original fixed weights to adaptive weights that can adjust with the changing adaptability of particles. After a series of transformations, the formula for particle velocities now appears as shown in Equation (19).

$$V(t+1) = w(t) \times V(t) + c_1(t) \times r_1 \times (pb - X(t)) + c_2(t) \times r_2 \times (gb - X(t)) + \alpha \times D \quad (19)$$

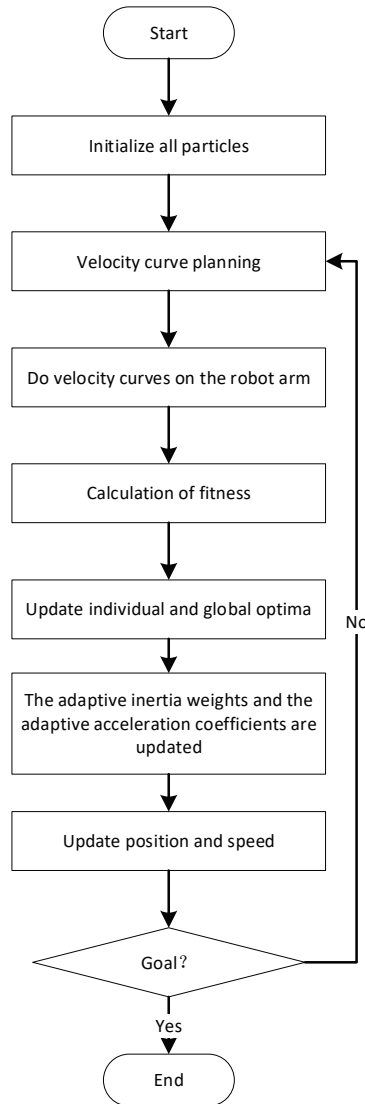


Figure 4. RTPPSO algorithm flow chart.

3.3. Model Implementation

During the actual operation process, the robotic arm transitions from a stationary state to movement and then to a final stop, requiring acceleration, constant speed, and

deceleration stages to ensure smooth operation and reduce vibrations. To achieve this, the velocity profile is composed of acceleration end points t_1 , constant speed end points t_2 , and maximum angular velocities $v_w^{\max}$, but where t_1 and t_2 are both known. Each axis motor has a different velocity profile based on the specific circumstances.

The angular velocity θ_1 in the first stage is represented in Equation (20), while θ_2 in the third stage is depicted in Equation (21).

$$\theta_1 = \frac{t}{t_1} 2\pi \quad (20)$$

$$\theta_2 = \frac{1-t}{1-t_2} 2\pi \quad (21)$$

The velocity curve can be categorized into three stages, and the angular velocity of each stage is expressed in Equation (22).

$$\begin{cases} v_w(t) = \frac{v_w^{\max}}{2\pi}(\theta_1 - \sin \theta_1), 0 < t \leq t_1 \\ v_w(t) = v_w^{\max}, t_1 < t \leq t_2 \\ v_w(t) = \frac{v_w^{\max}}{2\pi}(\theta_2 - \sin \theta_2), t_2 < t \leq 1 \end{cases} \quad (22)$$

If the velocity curve is known, the angular displacement $\Delta\theta$ can be calculated by integrating. The specific calculation process is shown in Equations (23)–(26). The maximum angular $v_w^{\max}$ velocity can also be calculated from these.

$$\begin{aligned} \Delta\theta &= \int_0^{t_1} \frac{v_w^{\max}}{2\pi} \left(\frac{t}{t_1} 2\pi - \sin\left(\frac{t}{t_1} 2\pi\right) \right) dt + \int_{t_1}^{t_2} v_w^{\max} dt \\ &+ \int_{t_2}^1 \frac{v_w^{\max}}{2\pi} \left(\frac{1-t}{1-t_2} 2\pi - \sin\left(\frac{1-t}{1-t_2} 2\pi\right) \right) dt \end{aligned} \quad (23)$$

$$\Delta\theta = \frac{v_w^{\max} \times t_1}{2} + v_w^{\max} \times (t_2 - t_1) + \frac{v_w^{\max} \times (1 - t_2)}{2} \quad (24)$$

$$\Delta\theta = \frac{v_w^{\max}(1 + t_2 - t_1)}{2} \quad (25)$$

$$v_w^{\max} = \frac{2 \times \Delta\theta}{1 + t_2 - t_1} \quad (26)$$

The objective function is a very important index in the trajectory rule, and the optimal trajectory can be obtained only by designing the objective function reasonably. This paper is mainly to reduce the jitter during the operation of the robot arm so that it can run smoothly and ensure the grabbing quality of the robot arm. Therefore, in the design of our objective function, we comprehensively consider the angular acceleration of the X axis, Y axis, Z axis, and rotation axis. The objective formula is as follows:

$$f = \min\{|A_{X,\max}|, |A_{Y,\max}|, |A_{Z,\max}|, |A_{Q,\max}|\} \quad (27)$$

where $A_{X,\max}$, $A_{Y,\max}$, $A_{Z,\max}$, and $A_{Q,\max}$ represent the maximum acceleration of the X axis, Y axis, Z axis, and rotation axis, respectively.

In the first step, the setup is initialized, and then the acceleration end time and uniform end time for each motor are updated.

In the second step, the step is to use the interpolation algorithm to fit the velocity curve. Then, Equations (20)–(26) are used to calculate the angular displacement, angular velocity, and maximum angular velocity of the motor.

In the third step, the step is to run the robot arm according to the designed speed curve.

In the fourth step, the fitness value is calculated using sensor information based on four reference indices, including the maximum acceleration of the X, Y, and Z axes, along with the rotation axis.

In the fifth step, the fitness values are calculated, the individual and global optimal values are updated, and the adaptive weight parameters and adaptive acceleration parameters of the particles are updated. Then, it judges whether the set number of iterations is reached, if not, it returns to step 1; If reached, the iteration ends. The output is the current optimal value.

4. Experimental Simulation

4.1. Experiment

In the practical utilization process, specific requirements exist for the speed and synchronization of the sample manipulator. In order to better realize the trajectory planning operation of the robot arm, the parameters used are designed as follows according to the following marks. The number of iterations is set to 9, the total number of particles is set to 7, and parameter $w_{\min}$ is set to 0, $w_{\max}$ is set to 1, α is set to 2.05, and β is set to 0.5 in the particle swarm optimization algorithm. To meet these requirements, in the simulation experiments, each action of the robotic arm must be completed within 3 s, as indicated by the red box annotations in Figure 5a,b.

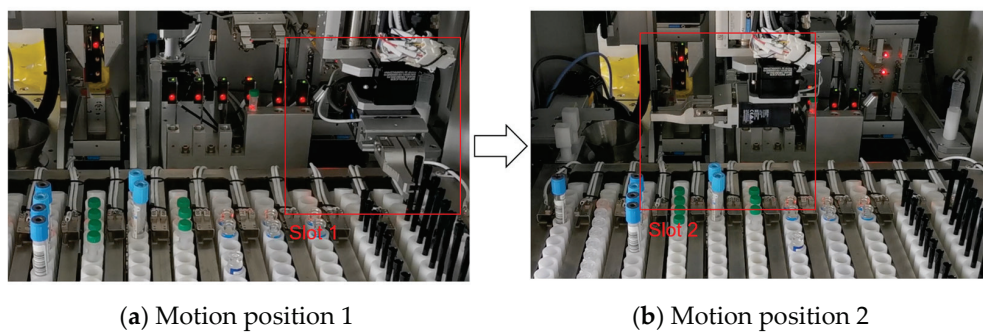


Figure 5. Machine arm motion illustration.

First of all, a simulation is conducted comparing the trapezoidal velocity curve to the S-shaped velocity curve. The S-shaped curve is designed in five sections. For a comprehensive analysis, the same maximum speed and movement time are used to ensure a clear comparison.

The velocity and acceleration curves of both are shown in Figure 6. It is evident from the figure that despite having the same maximum speed and displacement, the trapezoidal curve exhibits obvious acceleration discontinuity. This discontinuity is likely to cause a certain degree of jitter during the movement of the mechanical arm. Moreover, the S-shaped curve, owing to its continuous change in acceleration, avoids the impact of sudden acceleration during operation. However, the acceleration curve reveals that the maximum acceleration of the S-shaped curve surpasses that of the trapezoidal curve. When the maximum acceleration is excessively high and the acceleration time is short, the mechanical arm may experience jitter due to rapid acceleration changes. Consequently, adjustments are required for maximum acceleration in the S-shaped curve.

The manipulator's displacement is determined based on the starting point and the ending point. Velocity curve planning is then executed for this path, employing an improved particle swarm optimization algorithm for learning. Figure 7 illustrates the optimal particle velocity curve of the X axis after the first iteration. In the figure, the X axis represents time, and the Y axis represents motor speed. Observing the figure, it becomes apparent that the maximum velocity and acceleration of the first iteration are relatively large, with a relatively short running time. The acceleration time is small, and the acceleration is significant, indicating that the mechanical arm is greatly influenced by inertia during movement.

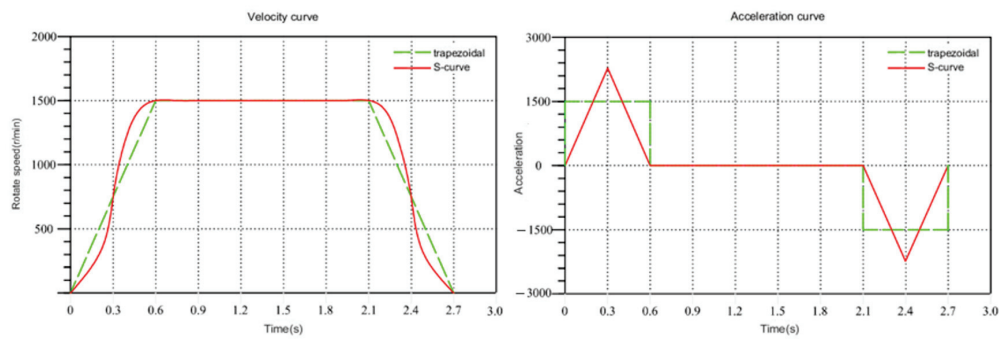


Figure 6. Comparison of trapezoidal curve and S-curve.

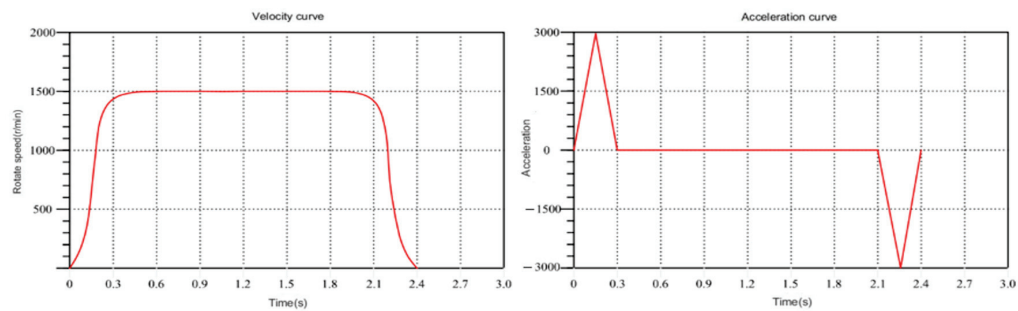


Figure 7. Velocity curve of the first iteration.

This influence is not conducive to the stable operation of the mechanical arm.

After multiple learning iterations, the global optimal solution is obtained. The iterative results enable the deviation of the velocity curve and acceleration curve, as shown in Figure 8. Compared to the results of the first iteration, the speed curve, optimized by the PSO algorithm, is smoother than that of the first iteration in the specified running time. Moreover, the maximum acceleration is greatly reduced, and the acceleration and deceleration time are also significantly improved. This is conducive to the stable operation of the manipulator.

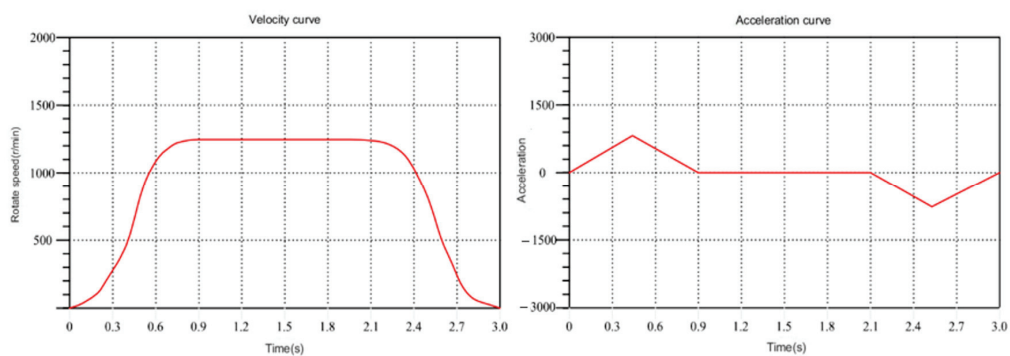


Figure 8. Velocity curve of global optimal solution.

According to the fitness function, the fitness value at the end of each iteration is computed, and the fitness learning curve is obtained as illustrated in Figure 9. According to the learning curve, using the improved PSO algorithm to plan the velocity curve, the optimal solution of the system is generated after a few iterations. This effectively improves the slow convergence of the traditional PSO algorithm and effectively avoids the influence of the local optimal solution on the result by altering the learning factor.

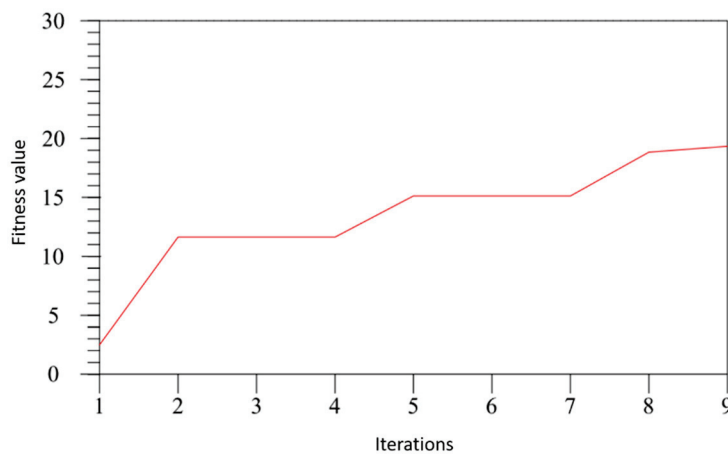


Figure 9. Fitness learning curve.

In this study, an improved PSO algorithm was proposed to plan the velocity curve of the motion of the industrial robot arm. After several learning periods, the stability of the operation of the robot arm was significantly improved, and the four-axis acceleration was in a relatively stable state. The results proved the feasibility and efficiency of the trajectory planning algorithm deployed in this paper.

4.2. Discussion

To validate the effectiveness of the proposed algorithm, we compared the acceleration and operational speed of the robotic arm using the proposed algorithm against other optimization algorithms. We recorded the differences before and after applying the algorithm. In the comparative experiment, we designed a set of rotations and grasping actions for the robotic arm to compare the time taken and acceleration variations.

According to Table 1, the RTPPSO algorithm demonstrates the best optimization effect on the robotic joint impacts. The average impacts on the drive arm joints before and after optimization are $2.23 \times 10^6 \text{ deg/s}^3$ and $4.58 \times 10^5 \text{ deg/s}^3$, respectively, resulting in a reduction in impacts by 79.45%. It can be seen from Table 2 that our algorithm also improves the grasping speed to some extent, which can effectively improve the grasping efficiency of the robotic arm. Compared to other algorithms, the RTPPSO algorithm has better realized trajectory planning work, enhancing the operational efficiency of the robotic arm while effectively reducing vibration issues during operation. Compared to other algorithms, the RTPPSO algorithm better addresses vibration issues during operation. However, this study did not incorporate optimization of computational efficiency, which relates to the operational time of the robotic arm, into the design of the optimization problem. Further improvements are needed in future iterations to address this aspect.

Table 1. Impact optimization comparison.

Algorithm	Average Impact before Optimization (deg/s ³)	Average Impact after Optimization (deg/s ³)
GA	2.23×10^6	4.89×10^5
PSO	2.23×10^6	4.83×10^5
RTPPSO	2.23×10^6	4.58×10^5

Table 2. Time optimization simulation comparison.

Algorithm	Time before Optimization (s)	Time after Optimization (s)
GA	2.5	2.362
PSO	2.5	2.358
RTPPSO	2.5	2.278

5. Conclusions

This paper investigates the motion stability of industrial robotic arms. To address the challenge of optimizing system speed adjustments to reach the optimal positions, an improved Robot Trajectory Planning Particle Swarm Optimization (RTPPSO) algorithm is proposed. By incorporating a random perturbation term and designing adaptive weights, we effectively mitigated the issue of the PSO algorithm getting stuck in local optima. To validate the algorithm's effectiveness, we conducted simulation experiments. The final experimental curves demonstrate the favorable effects of our algorithm on speed planning, showing smooth operation with gentle acceleration and deceleration processes, without abrupt changes. Subsequent comparative experiments further confirmed that our algorithm can effectively ensure the smooth operation of the robotic arm.

In the future, we plan to validate this algorithm in realistic scenarios that involve obstacles or angular constraints. This will ensure that the algorithm proposed in this paper is applicable to a broader range of industrial settings.

Author Contributions: N.W.: conceptualization, methodology, software, writing—original draft preparation, manuscript writing, visualization, investigation, writing—reviewing and editing. D.J.: supervision. Z.L.: data curation, investigation. Z.H.: software, validation. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge that this work was supported by the Fundamental Research Funds for the Central Universities (grant 2022YJS020).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, J.; Zhu, X.; Song, T. Serial Manipulator Time-Jerk Optimal Trajectory Planning Based on Hybrid IWOA-PSO Algorithm. *IEEE Access* **2022**, *10*, 6592–6604. [CrossRef]
2. Batista, J.; Souza, D.; Silva, J.; Ramos, K.; Costa, J.; dos Reis, L.; Braga, A. Trajectory Planning Using Artificial Potential Fields with Metaheuristics. *IEEE Lat. Am. Trans.* **2020**, *18*, 914–922. [CrossRef]
3. Rubio, F.; Valero, F.; Sunyer, J.L.; Garrido, A. The simultaneous algorithm and the best interpolation function for trajectory planning. *Ind. Robot. Int. J. Robot. Res. Appl.* **2010**, *37*, 441–451. [CrossRef]
4. Sun, J.; Han, X.; Zuo, Y.; Tian, S.; Song, J.; Li, S. Trajectory Planning in Joint Space for a Pointing Mechanism Based on a Novel Hybrid Interpolation Algorithm and NSGA-II Algorithm. *IEEE Access* **2020**, *8*, 228628–228638. [CrossRef]
5. Chen, W.-C.; Chen, C.-S.; Lee, F.-C.; Chen, L.Y. High speed blending motion trajectory planning using a predefined absolute accuracy. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 2179–2193. [CrossRef]
6. Shi, C.; Ye, P. The look-ahead function-based interpolation algorithm for continuous micro-line trajectories. *Int. J. Adv. Manuf. Technol.* **2011**, *54*, 649–668. [CrossRef]
7. Luo, L.-P.; Yuan, C.; Yan, R.-J.; Yuan, Q.; Wu, J.; Shin, K.-S.; Han, C.-S. Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method. *Int. J. Precis. Eng. Manuf.* **2015**, *16*, 911–917. [CrossRef]
8. Nadir, B.; Mohammed, O.; Minh-Tuan, N.; Abderrezak, S. Optimal trajectory generation method to find a smooth robot joint trajectory based on multiquadric radial basis functions. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 297–312. [CrossRef]
9. Abu-Dakka, F.J.; Assad, I.F.; Alkhdour, R.M.; Abderahim, M. Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 389–406. [CrossRef]

10. Liu, Y.; Guo, C.; Weng, Y. Online Time-Optimal Trajectory Planning for Robotic Manipulators Using Adaptive Elite Genetic Algorithm with Singularity Avoidance. *IEEE Access* **2019**, *7*, 146301–146308. [CrossRef]
11. Wang, M.; Luo, J.; Zheng, L.; Yuan, J.; Walter, U. Generate optimal grasping trajectories to the end-effector using an improved genetic algorithm. *Adv. Space Res.* **2020**, *66*, 1803–1817. [CrossRef]
12. Chen, D.; Li, S.; Wang, J.; Feng, Y.; Liu, Y. A multi-objective trajectory planning method based on the improved immune clonal selection algorithm. *Robot. Comput. Manuf.* **2019**, *59*, 431–442. [CrossRef]
13. Jin, M.; Wu, D. Collision-Free and Energy-Saving Trajectory Planning for Large-Scale Redundant Manipulator Using Improved PSO. *Math. Probl. Eng.* **2013**, *2013*, 208628. [CrossRef]
14. Cao, B.; Sun, K.; Li, T.; Gu, Y.; Jin, M.; Liu, H. Trajectory Modified in Joint Space for Vibration Suppression of Manipulator. *IEEE Access* **2018**, *6*, 57969–57980. [CrossRef]
15. Liu, C.; Cao, G.-H.; Qu, Y.-Y.; Cheng, Y.-M. An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 1091–1099. [CrossRef]
16. Lv, X.; Yu, Z.; Liu, M.; Zhang, G.; Zhang, L. Direct Trajectory Planning Method Based on IEPPO and Fuzzy Rewards and Punishment Theory for Multi-Degree-of-Freedom Manipulators. *IEEE Access* **2019**, *7*, 20452–20461. [CrossRef]
17. Li, Y.; Ge, S.S.; Wei, Q.; Gan, T.; Tao, X. An Online Trajectory Planning Method of a Flexible-Link Manipulator Aiming at Vibration Suppression. *IEEE Access* **2020**, *8*, 130616–130632. [CrossRef]
18. DU, Y.; Chen, Y. Time Optimal Trajectory Planning Algorithm for Robotic Manipulator Based on Locally Chaotic Particle Swarm Optimization. *Chin. J. Electron.* **2022**, *31*, 906–914. [CrossRef]
19. Yang, Y.; Xu, H.-Z.; Li, S.-H.; Zhang, L.-L.; Yao, X.-M. Time-optimal trajectory optimization of serial robotic manipulator with kinematic and dynamic limits based on improved particle swarm optimization. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 1253–1264. [CrossRef]
20. Wang, L.; Wu, Q.; Lin, F.; Li, S.; Chen, D. A New Trajectory-Planning Beetle Swarm Optimization Algorithm for Trajectory Planning of Robot Manipulators. *IEEE Access* **2019**, *7*, 154331–154345. [CrossRef]
21. Zhang, W.; Fu, S. Time-optimal Trajectory Planning of Dulcimer Music Robot Based on PSO Algorithm. In Proceedings of the 32nd Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4769–4774.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Imbalanced Data Parameter Optimization of Convolutional Neural Networks Based on Analysis of Variance

Ruiao Zou and Nan Wang *

School of Mathematical Science, Heilongjiang University, Harbin 150080, China; 2221335@s.hlj.u.edu.cn

* Correspondence: 2003137@hlj.u.edu.cn

Abstract: Classifying imbalanced data is important due to the significant practical value of accurately categorizing minority class samples, garnering considerable interest in many scientific domains. This study primarily uses analysis of variance (ANOVA) to investigate the main and interaction effects of different parameters on imbalanced data, aiming to optimize convolutional neural network (CNN) parameters to improve minority class sample recognition. The CIFAR-10 and Fashion-MNIST datasets are used to extract samples with imbalance ratios of 25:1, 15:1, and 1:1. To thoroughly assess model performance on imbalanced data, we employ various evaluation metrics, such as accuracy, recall, F1 score, P-mean, and G-mean. In highly imbalanced datasets, optimizing the learning rate significantly affects all performance metrics. The interaction between the learning rate and kernel size significantly impacts minority class samples in moderately imbalanced datasets. Through parameter optimization, the accuracy of the CNN model on the 25:1 highly imbalanced CIFAR-10 and Fashion-MNIST datasets improves by 14.20% and 5.19% compared to the default model and by 8.21% and 3.87% compared to the undersampling model, respectively, while also enhancing other evaluation metrics for minority classes.

Keywords: imbalanced data; analysis of variance; parameter optimization; fractional factorial design

1. Introduction

In the current era, which is dominated by data, deep learning techniques, especially CNNs (convolutional neural networks), have achieved significant success in various fields, including but not limited to facial and image recognition [1] and medical imaging [2]. Despite the widespread application and success of CNNs in various domains, the research on CNN-based models for imbalanced data classification is not well developed. However, the issue of data imbalance is very common in real-life scenarios, such as in medical diagnosis [3], fraud detection [4], and fault detection [5].

For imbalanced data, many traditional methods are available [6], including under-sampling [7], oversampling [8], cost-sensitive learning [9], and ensemble learning [10]. These approaches enhance the classification performance of minority class samples by rebalancing the data, using ensemble models, or modifying the classifier's cost function. First, the Synthetic Minority Oversampling Technique (SMOTE) employs hybrid sampling to balance the sample distribution in the dataset by generating new minority class samples, achieving favorable results [11]. This method enhances the model's ability to recognize minority classes by synthesizing new minority class samples, allowing the model to focus more on these classes during training. Secondly, cost-sensitive methods improve the classification performance on imbalanced samples by introducing a weighted cost function in the CNN [12]. This approach increases the loss cost for small samples, thereby enhancing the model's learning of minority class features and reducing the bias towards majority class samples. Finally, ensemble learning methods, which combine multiple classifiers, have shown significant effectiveness in addressing imbalanced sample classification problems [10]. Techniques such as bagging [13] and boosting [14] aggregate multiple weak

classifiers into a strong classifier, effectively improving the model's recognition ability for minority class samples.

The performance and accuracy of CNNs are primarily influenced by network architecture and parameter selection. The LeNet-5 network architecture, developed by LeCun et al., marked the beginning of deep learning research [15]. Subsequently, the well-known AlexNet architecture, introduced by Krizhevsky et al., further advanced the field of deep learning [1]. The VGG network, created by Simonyan and Zisserman, and the ResNet network developed by He et al. are among the most widely used and best-performing networks to date [16,17]. In terms of parameter selection, researchers have made significant efforts. However, parameter selection for imbalanced data has not been extensively studied. This is primarily because there are many parameters that affect the classification performance of imbalanced data, and many of these parameters interact with each other. For example, learning rate, batch size, and regularization parameters can all have a significant impact on model performance [18]. Moreover, there is a diversity in the evaluation metrics for classification performance on imbalanced data. Unlike balanced data, metrics such as accuracy, ROC, and AUC do not comprehensively assess the performance of imbalanced classification problems [8,19]. In practical scenarios, metrics like recall for the minority class, G-mean, and F1 score might be of greater concern. For example, in disease diagnosis, where the number of patients with the disease can be small, whether 100% of the diseased individuals can be identified is often of utmost importance to doctors. In the face of the aforementioned complex situations, simple parameter-tuning methods often cannot meet the needs of parameter selection for imbalanced samples. Therefore, this paper proposes an experimental design and parameter selection method based on factor analysis, which systematically analyzes the impact of parameters on model performance.

Fractional factorial design [20] is a widely used statistical method in experimental design that is primarily employed to evaluate the impact of multiple factors and their interactions on experimental outcomes. By systematically varying the levels of multiple factors to study their effects on the response variable, fractional factorial design can effectively identify the main effects and interactions of each factor. The references provide a comprehensive and systematic approach for model parameter optimization [21,22].

This study proposes a systematic experimental method based on factorial design to improve the performance of the CNN model on imbalanced data. Our research uses analysis of variance (ANOVA) to explore the main effects and interaction effects of parameters such as the learning rate, dropout rate, and kernel size in handling imbalanced data, with the aim of identifying the optimal configuration for imbalanced datasets. The experimental results demonstrate that the optimized CNN model significantly improves the classification performance of minority class samples when handling 25:1 imbalanced datasets extracted from CIFAR-10 and Fashion-MNIST.

In this context, we summarize our contributions as follows:

- We combine domain expertise in a CNN with parameter optimization, first using fractional factorial design and ANOVA to evaluate the significant relationships between model parameters and evaluation metrics for minority class samples, such as accuracy, recall, P-mean, and G-mean. We attempt to use ANOVA to analyze whether these parameters exhibit main effects and interaction effects.
- Based on the ANOVA results, we identify and optimize the parameters that significantly impact the imbalanced data. The optimized model is compared with the undersampling method for various evaluation metrics for minority class classification in imbalanced data, thereby confirming the importance of parameter selection in handling imbalanced data.

2. Related Work

Methods for imbalanced data classification are typically categorized into two main types of approach: data-processing techniques and algorithm/model-oriented methods.

These studies have laid a solid theoretical foundation for extending deep learning methods to imbalanced data classification.

In terms of data-processing techniques, the main approaches are resampling and undersampling techniques [23–25]. These methods are designed to equalize the number of samples in different classes or to minimize noise, thereby improving the model's performance when dealing with imbalanced data. Undersampling eliminates data, thereby decreasing the total amount of information available for the model to learn. Conversely, oversampling boosts the number of minority class samples, which enlarges the training dataset and can lead to overfitting [26]. To address this issue, Zhang and Mani [27] proposed using K-nearest neighbors (K-NN) to calculate the distances between the sample to be classified and the samples in the training set, selecting most samples for removal. Chawla et al. [8] introduced the SMOTE, which creates synthetic samples for the minority class by interpolating between existing minority class samples. Unlike traditional methods that replicate minority class samples, the SMOTE reduces the risk of overfitting and significantly enhances classifier performance on imbalanced datasets.

In contrast to data-sampling methods, algorithmic approaches for addressing class imbalance do not modify the distribution of the training data. Instead, they adjust the learning or decision-making process, including cost-sensitive learning and ensemble learning. Cost-sensitive learning primarily involves adjusting learning algorithms based on the classification costs of different classes to effectively handle minority class samples [28]. In 2013, He et al. proposed a cost-sensitive learning method based on Gradient-Boosting Decision Trees, effectively improving the accuracy and robustness of models when dealing with imbalanced data [29]. Additionally, research has explored ways to enhance cost-sensitive learning capabilities by integrating deep learning techniques, such as designing loss functions suitable for deep neural networks or hierarchical cost-sensitive mechanisms [30]. Sun et al. [31] presented three cost-sensitive boosting techniques: AdaC1, AdaC2, and AdaC3. These methods introduce cost terms into the weight updates of AdaBoost, iteratively increasing the influence of the minority classes. Sun demonstrated that, in most cases, cost-sensitive boosting ensemble methods outperformed ordinary boosting methods. Wu et al. [32] combined cost-sensitive learning with Support Vector Machines (SVMs), employing a cost-sensitive SVM algorithm to adjust the margin and induce boundary drift, thereby improving the recall rate of minority classes.

Ensemble learning is a method that improves predictive performance by combining multiple base classifiers. Early ensemble learning methods, such as Random Forest, which combines bagging and decision trees, emphasize the importance of feature randomness [33]. Gradient-Boosting Machines (GBMs) and their improved versions, such as XGBoost and LightGBM, significantly enhance prediction accuracy through optimized loss functions and enhanced collaboration among learners [34]. Zhong et al. [35] integrated data-level methods with ensemble learning, combining predictions from multiple CNN sub-models to enhance classification performance.

Additionally, there are other methods at the algorithmic research level. For example, Zhou et al. [36] proposed a cumulative learning approach that decomposes class distributions into balanced subsets to address long-tail recognition issues. This method has demonstrated promising outcomes for managing highly imbalanced data. Hensman and Masko [37] studied the impact of training a CNN on multiclass imbalanced data and found that imbalanced data distribution significantly affects CNN performance, particularly with substantial impacts from distributions involving single minority classes and majority classes. These methods have limitations in improving the accuracy of imbalanced data, particularly the accuracy of minority class samples.

Research on parameter optimization specifically for imbalanced data is scarce. Existing studies on parameters have predominantly focused on balanced datasets, such as investigating the number of convolutional kernels and optimizing CNN hyperparameters using genetic algorithms, and methods for exploring the number of convolutional kernels [38]. The primary aim was to find the optimal number to enhance model performance. Using

genetic algorithms [39] for CNN hyperparameter optimization is an automated approach for tuning hyperparameters. Genetic algorithms simulate natural selection processes, evolving better hyperparameter combinations through operations like crossover, mutation, and selection. However, these methods primarily automate the generation of hyperparameters without fully leveraging domain expertise in the CNN field. Incorporating domain expertise in a CNN can also be an effective method for improving classification of minority class samples.

Therefore, despite some attempts, challenges, and unresolved issues, selecting CNN parameters for imbalanced data remains essential. Future research can aim to comprehensively integrate domain expertise in the CNN field with existing optimization techniques to develop more effective parameter selection methods for imbalanced data. This approach seeks to enhance the performance and generalization ability of CNN models for minority class samples. Thus, this paper discusses CNN parameter selection specifically for minority class samples in imbalanced data, with the aim of optimizing existing network structures to achieve ideal classification results for these samples.

3. Background Models

3.1. Convolutional Neural Network

In convolutional neural networks, convolutional layers, pooling layers, and fully connected layers create an efficient framework for feature extraction and classification, as shown in Figure 1. Convolutional layers extract local features from images, pooling layers reduce the size of feature maps and the number of parameters, lowering model complexity, and fully connected layers convert feature maps into one-dimensional vectors for classification. Although stacking multiple convolutional layers and using numerous convolutional kernels can build more complex network structures, it also increases the computational resource requirements and risk of overfitting. Therefore, when designing a CNN, a balance must be struck between network complexity and performance. A deep understanding of CNN structure and parameter selection is crucial for enhancing the performance of image-processing technologies.

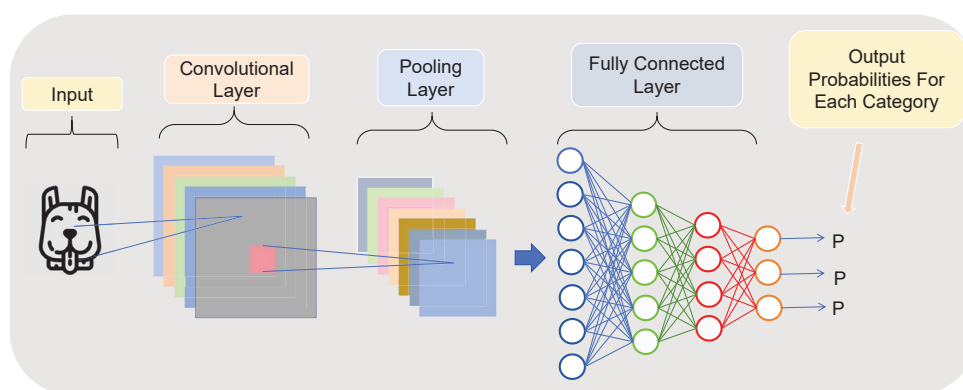


Figure 1. The working principle of a CNN.

3.2. Parameters in CNN Model

Learning rate. The learning rate determines the step size of the parameter updates during training. A too-small learning rate can result in slow convergence and potential trapping in local minima, whereas a large learning rate can result in oscillations and divergence during training. An appropriate learning rate strikes a balance between training speed and model performance.

Dropout rate. The dropout rate parameter in the dropout layer randomly drops some neurons and their connections during the training, as shown in Figure 2. Not applying dropout can lead to overfitting on the training set, whereas applying dropout can reduce overfitting and improve the model's generalization on unseen data. Therefore, the decision

to use dropout should be based on specific circumstances and dataset characteristics to achieve an optimal model performance. In our setup, dropout was applied differently after convolutional layers and fully connected layers [39], denoted as dropout rate 1 and dropout rate 2, respectively.

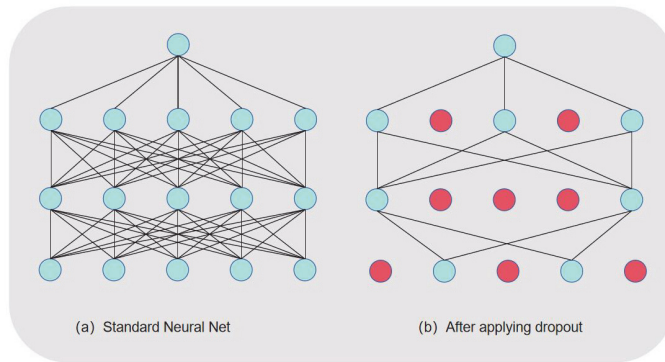


Figure 2. Dropout layer working principle.

Convolutional kernel size. Convolutional kernel size refers to the dimensions of the small matrix used for feature extraction in the CNN. Adjusting the kernel size can influence the scale and complexity of the features learned by the network, thereby impacting its performance across different tasks.

3.3. Evaluation Metrics

The confusion matrix is essential for evaluating the model performance in classification problems. It is a two-dimensional matrix with rows representing the true classes and columns representing the predicted classes, as shown in Table 1. Each element of the confusion matrix represents the prediction of the model for samples belonging to a certain true class.

Table 1. Confusion matrix of actual vs. predicted classifications.

		Predicted	
		Positive	Negative
Actual	Positive	<i>TP</i>	<i>FN</i>
	Negative	<i>FP</i>	<i>TN</i>

True positive (*TP*) refers to the number of actual positive instances that the model correctly predicts as positive; false positive (*FP*) refers to the number of negative instances that the model incorrectly predicts as positive; false negative (*FN*) refers to the number of positive instances the model incorrectly predicts as negative; and true negative (*TN*) refers to the number of actual negative instances that the model correctly predicts as unfavorable. Based on these metrics, we can calculate many evaluation metrics for assessing the performance of classification models, including accuracy, recall, F1 score, and specificity.

Accuracy: Accuracy refers to the proportion of samples correctly predicted by the model out of the total number of samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision: Precision refers to the proportion of samples that the model correctly predicts as positive out of all samples predicted as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall: Recall refers to the proportion of samples that the model correctly predicts as positive among all positive samples.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

Specificity: This measures a model's ability to correctly predict negative instances among all true negatives in a binary classification problem. Specifically, it refers to the proportion of true negatives that the model correctly identified as unfavorable.

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (4)$$

F1 score: F1 score is the harmonic mean of precision and recall and is used for the comprehensive evaluation of model performance.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

G-mean: The G-mean is a composite evaluation metric commonly used to assess the performance of binary classification models on imbalanced datasets.

$$\text{G-mean} = \sqrt{\text{Recall} \times \text{Precision}} \quad (6)$$

P-mean: P-mean evaluates the performance of a classifier by computing the geometric mean of precision and recall. Unlike the F1 score, the P-mean does not apply a weighted average to precision and recall, making it particularly valuable in scenarios where the costs of false positives and false negatives differ.

$$\text{P-mean} = \sqrt{\text{Recall} \times \text{Specificity}} \quad (7)$$

These evaluation metrics are all based on a binary classification performance evaluation. In this study, to address class imbalance in multiclass problems, we adopted the one-vs.-rest (OvR) [40] approach. This method treats the focus class (minority class samples) as positive. It combines the remaining nine classes into a negative class, transforming the multiclass problem into a series of binary classification problems. By utilizing these traditional binary classification metrics, we can better assess the performance of the model in handling imbalanced multiclass datasets, particularly its ability to recognize minority class samples, and overall classification performance. These metrics provide a comprehensive classifier evaluation across different classes, aiding in optimizing the model parameters to improve the classification accuracy and recall.

4. Statistical Model

4.1. Fractional Factorial Design

Fractional factorial design is a comprehensive experimental method that involves the cross-grouping of multiple factors at multiple levels. This allows the investigation of the main effects and interaction effects of two or more factors. When studying how to enhance the performance of convolutional neural networks on imbalanced datasets, fractional factorial design can be employed for comparative experiments [41]. This approach can explore the main effects and interaction effects of factors such as learning rate (LR), dropout rate 1 (DR 1), dropout rate 2 (DR 2), and kernel size (KS) on the accuracy, recall, G-mean, and P-mean of the CNN model. Dropout rate 1 and dropout rate 2 refer to the parameters in the dropout layers following the pooling layers and after the fully connected layers, respectively.

4.2. Fractional Factorial Design

ANOVA is a method used to compare the means of multiple groups by analyzing the total variance and separating the factor and error effects to test whether there are significant relationships between multiple groups. To validate the results of the fractional factorial design experiment, we used an ANOVA model to determine the impact of the explanatory variables LR, DR 1, DR 2, and KS on the dependent variables, which, in our experiment, were the evaluation metrics.

4.2.1. Fixed-Effects Model

In the fixed-effects model, the levels of the factors are specific levels of interest to the researcher. We assume that these levels are fixed and are only interested in the effects at these particular levels. This model is applicable when assessing the impact of specific treatments or experimental conditions on the response variable.

In this article, we establish the following model:

Let Y_{ijkl} be the observation value at the i level of the LR, the j level of DR 1, the k level of DR 2, and the l level of convolution KS. All observation values Y_{ijkl} are mutually independent and follow a normal distribution with a constant variance σ^2 .

We assume that the number of levels for each factor is $r = 5$, $c = 5$, $t = 5$, and $u = 2$, and each combination has the same number of replicate observations $s > 1$. Therefore, the total number of observations is $n = r \times c \times t \times u \times s$.

The formula for the fixed effects model is as follows:

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + (\alpha\beta)_{ij} + \dots + (\gamma\delta)_{kl} + \epsilon_{ijkl} \quad (8)$$

where μ is the overall mean, α_i is the main effect of the learning rate, β_j is the main effect of DR 1, γ_k is the main effect of DR 2, δ_l is the main effect of convolution kernel size, and ϵ_{ijkl} is the random error, which follows $N(0, \sigma^2)$. The rest are two-level interaction effects.

Through permutation tests, we test each (i, v) :

$$H_{vi} : e_v^T Z_i = 0 \quad \text{vs.} \quad K_{vi} : e_v^T Z_i \neq 0 \quad (9)$$

where Z_i is the aligned data vector at level i . We use e_v to represent the orthogonal vector associated with degree v . Thus, $e_v^T Z_i$ is the projection of the aligned data vector at level i to degree v . The null hypothesis H_{vi} indicates no significant interaction effect i and degree v .

When conducting hypothesis tests, we typically use the F-test to evaluate whether the model's main and interaction effects are significant. The sum of squares for main effects (SS_A , SS_B) and the sum of squares for interaction effects (SS_{AB}) measure the influence of different factors on the response variable.

For example, for LR (A) and DR 1 (B), the formulas for calculating the sum of squares for the main effects and interaction effects are as follows:

$$SS_A = s \sum_{i=1}^r (\bar{Y}_{i..} - \bar{Y})^2 \quad (10)$$

$$SS_B = s \sum_{j=1}^c (\bar{Y}_{.j.} - \bar{Y})^2 \quad (11)$$

$$SS_{AB} = s \sum_{i=1}^r \sum_{j=1}^c (\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y})^2 \quad (12)$$

4.2.2. Estimation of Marginal Means and 95% Confidence Intervals

To provide a more comprehensive interpretation of the effects, we employed the method of estimating marginal means and 95% confidence intervals. This method involves

calculating the marginal means for each factor level and providing confidence intervals for these estimates to represent the precision and variability of the effects visually.

The estimates of the marginal means are given by the following:

$$\hat{Y}_{A_i} = \hat{\mu} + \hat{\alpha}_i \quad (13)$$

$$\hat{Y}_{B_j} = \hat{\mu} + \hat{\beta}_j \quad (14)$$

$$\hat{Y}_{ij} = \hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j + (\alpha\beta)_{ij} \quad (15)$$

The 95% confidence interval (CI) is given by

$$CI = \hat{Y} \pm t_{\alpha/2, df} \cdot SE(\hat{Y}) \quad (16)$$

where $t_{\alpha/2, df}$ is the critical value from the t-distribution, and $SE(\hat{Y})$ is the standard error of the estimate, calculated using the residuals from the ANOVA model as follows:

$$SE(\hat{Y}_{ij}) = \sqrt{\frac{MSW}{n_{ij}}} \quad (17)$$

Through these steps, we can comprehensively analyze the impact of various factors on the performance of the model and further interpret these effects.

5. Experimental Design

This paper aims to provide a detailed explanation of the impact of convolutional neural network parameters on imbalanced data and their optimization, as shown in Figure 3. Next, we describe the functionalities of the proposed framework.

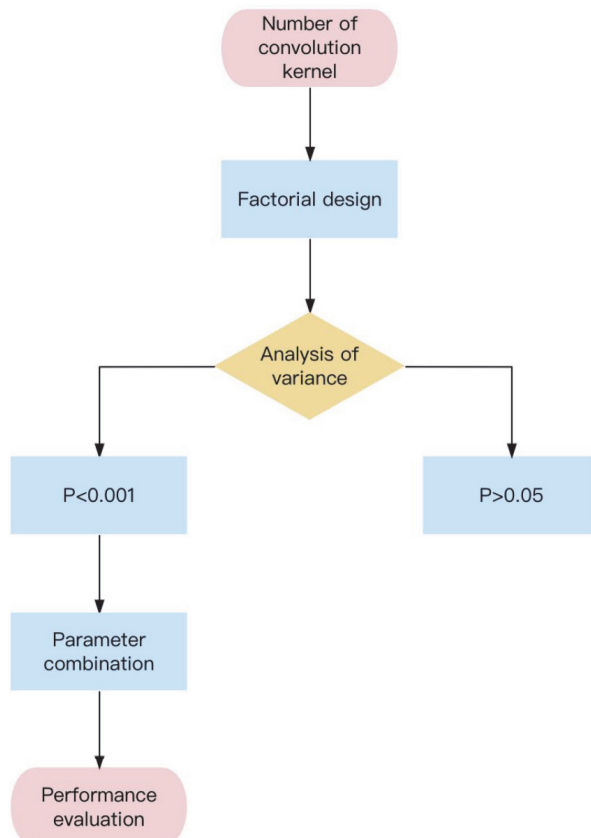


Figure 3. The framework proposed in this article.

5.1. Dataset Description

5.1.1. CIFAR-10 Dataset

CIFAR-10 is a commonly used dataset for image classification, primarily used for research and algorithm evaluation in the realm of computer vision. The dataset consists of color images categorized into 10 classes, with each class containing 6000 images of size 32×32 pixels, totaling 60,000 images. Among these, 50,000 images were used as training samples and 10,000 were used as test samples. These images cover various common object categories in daily life, including airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks, as shown in Figure 4.

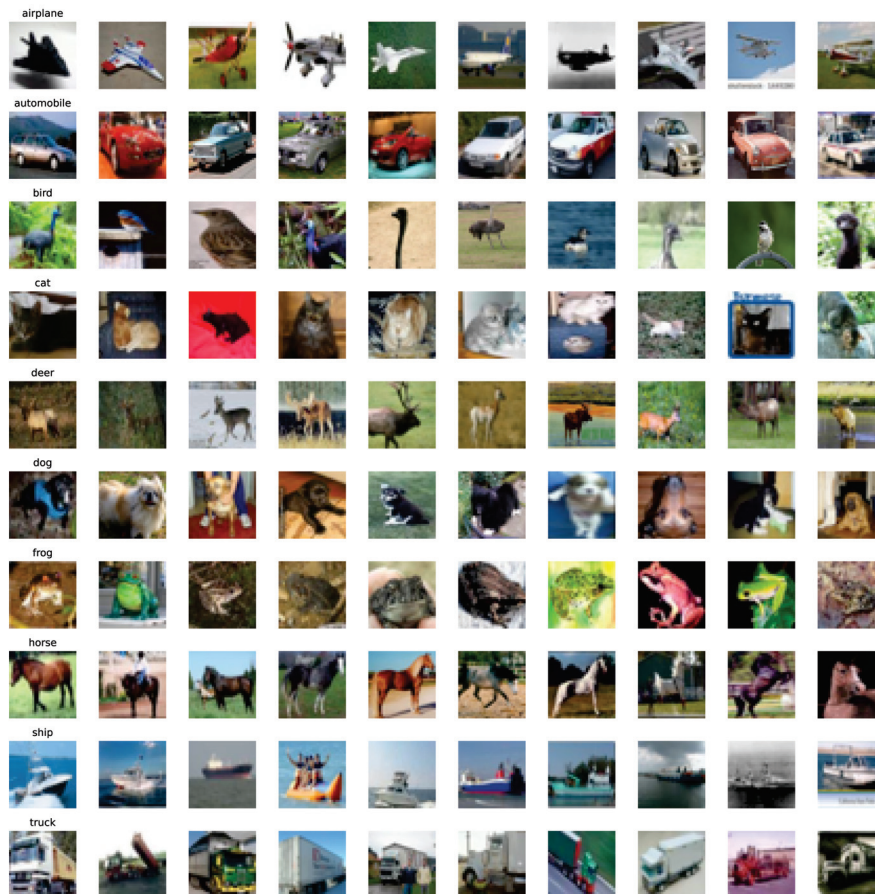


Figure 4. Images from the CIFAR-10 dataset. The dataset consists of images divided into 10 unique classes. Each class contains 5000 images from the training and 1000 from the test datasets. These are small-resolution RGB color images with a size of 32×32 pixels.

5.1.2. Fashion-MNIST Dataset

The Fashion-MNIST dataset, released by Zalando Research, is an image classification benchmark dataset containing 70,000 grayscale images. These images belong to ten different categories of clothing and footwear, including T-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot, as shown in Figure 5. Compared to the traditional MNIST handwritten digit dataset, the Fashion-MNIST dataset is more diverse and challenging, featuring rich textures and shapes that closely resemble actual clothing products. The Fashion-MNIST dataset is widely used in image classification, computer vision, machine learning, and deep learning, and provides a more practical benchmark for evaluating and developing advanced image-processing algorithms.



Figure 5. Images from the Fashion-MNIST dataset. The dataset consists of images divided into 10 unique classes. These images are grayscale with a resolution of 28×28 .

Because this study explores the impact of neural network parameters on imbalanced data, we extracted varying numbers of samples from ten classes of the CIFAR-10 and Fashion-MNIST datasets to create three datasets with different levels of balance: highly imbalanced, moderately imbalanced, and balanced, with imbalance ratios of 25:1, 15:1, and 1:1, respectively. The number of samples extracted for the training set from both datasets is listed in Table 2, where each class in the test set of both datasets contains 1000 images.

Table 2. The number of samples extracted from each class in both the CIFAR-10 dataset and the Fashion-MNIST dataset is identical.

Class	0	1	2	3	4	5	6	7	8	9
25:1	5000	200	500	1500	350	800	200	400	1000	3000
15:1	5000	330	500	1500	450	850	350	400	1000	3000
1:1	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000

5.2. Fractional Factorial Design

Each dataset may contain hundreds to thousands of eligible convolutional neural network models. The previous analysis identified four parameters that significantly affect imbalanced data: LR, DR 1, DR 2, and KS. LR, DR 1, and DR 2 each have five levels, whereas

KS has two levels, labeled as *A1-A4*, *B1-B4*, *C1-C4*, and *D1-D2*, respectively. Table 3 lists each factor and its levels. We conducted a systematic experiment using ANOVA on these factors and their levels to explore each factor's effects on and interactions with the evaluation metrics.

Table 3. Different levels of each factor.

Parameter	Factor Level					Note: Factor Levels
	Low	Lower	Medium	Higher	High	
LR (<i>A</i>)	0.001	0.003	0.01	0.032	0.1	5 Levels
DR 1 (<i>B</i>)	0.01	0.02	0.05	0.11	0.25	5 Levels
DR 2 (<i>C</i>)	0.01	0.02	0.05	0.11	0.25	5 Levels
KS (<i>D</i>)	3 × 3				5 × 5	2 Levels

In this paper, we adopt a method based on Lagrange multipliers to calculate the number of convolution kernels in five convolutional layers [38]. First, we define the number of kernels in each layer as n_i , where i denotes the index of the convolution layer. According to the structure of the convolutional neural network, we can compute the total number of learned parameters using the following formula:

$$P_{\text{all}} = k \times k \times n_0 \times n_1 + \sum_{i=2}^N (k \times k \times n_{i-1} \times n_i) = k^2 \times \sum_{i=1}^N (n_{i-1} \times n_i) \quad (18)$$

where P_{all} represents the total number of learned parameters, k is the kernel size, N is the total number of convolution layers, and n_0 is the number of input channels.

Next, we construct the optimization problem using the Lagrange function:

$$L(n, \lambda_0, \dots, \lambda_i, \dots, \lambda_N) = \sum_{i=1}^N \log(n_i) + \lambda_0 \left(\sum_{i=1}^N n_{i-1} \times n_i - C \right) + \sum_{i=1}^N \lambda_i (n_{i-1} - n_i) \quad (19)$$

where C represents the total number of parameters in the network. By calculating the partial derivatives of the Lagrange function, we can determine the optimal configuration of convolution kernels. In this process, we focus on the relationship between the model's discriminative performance and parameter distribution, ensuring that the selected parameter configuration effectively enhances the ability to recognize minority class samples. This way, we can not only obtain the number of kernels for each layer but also optimize model performance to better address the challenges posed by imbalanced datasets.

We calculated the number of convolutional kernels for the five layers to be 103, 105, 107, 109, and 179. Subsequently, we fixed the number of kernels and employ a fractional factorial design to explore the relationship between the parameters and the evaluation metrics for the minority class samples. The framework structure of the CNN model used in this study is shown in Table 4, and we adjusted the internal parameters as needed based on different requirements.

We conducted fractional factorial design experiments and training on three imbalanced datasets extracted from CIFAR-10 and Fashion-MNIST using the same convolutional neural network layers and batch sizes. Each dataset underwent four repeated experiments to minimize randomness, with a total of 250 data groups per experiment. Based on the ANOVA results, we evaluated the effects of different factors and their interactions with various performance metrics, such as accuracy, recall, F1 score, G-mean, and P-mean, focusing on minority class samples.

Table 4. CNN model structure.

Layers	Configuration
Conv_1	32 filters, kernel, and ReLU
MaxPool_1	2×2 kernel
Dropout_1	20%
Conv_2	64 filters, kernel, and ReLU
MaxPool_2	2×2 kernel
Dropout_2	20%
Conv_3	32 filters, kernel, and ReLU
Conv_4	32 filters, kernel, and ReLU
Conv_5	32 filters, kernel, and ReLU
MaxPool_3	2×2 kernel
Dropout_3	20%
Fully_1	256 neurons, ReLU
Fully_2	10 neurons, SoftMax

5.3. Analysis of Variance Results

5.3.1. ANOVA Results under Different Imbalance Ratios

We conducted comprehensive ANOVA experiments to gain a deeper understanding and quantify the impact of different parameters on the CNN model performance across various levels of imbalance. ANOVA was conducted on datasets with imbalance ratios of 25:1, 15:1, and 1:1, showing the main and interactions effects of LR, DR 1, DR 2, and KS on CNN model accuracy.

Table 5 presents partial results from the fractional factorial design experiments conducted on the CIFAR-10 dataset, encompassing a total of 250 data points. For clarity and conciseness, Table 5 displays only rows 1–4, 124–126, and 247–250, representing the overall dataset. Each parameter configuration was tested four times, and the accuracy, F1 score, G-mean, P-mean, and recall values were obtained for the minority class in each experiment. The same experiments were conducted using the Fashion-MNIST dataset. These data were used for variance analysis to assess the main and interaction effects of different factors on model performance. Through repeated experiments, we can better understand the impact of each parameter on the model performance and identify the optimal parameter combination to enhance the model's classification performance on imbalanced datasets. The data in the table demonstrate the consistency and variability of the results under the same parameter configurations, providing reliable support data for the subsequent variance analysis.

Comparing the results from Tables 6–8, it is evident that LR, KS, and their interaction significantly impact the model accuracy across all imbalance ratios. The results indicate that these two parameters are crucial tuning factors across different levels of imbalanced datasets.

In the CIFAR-10 dataset with an imbalance ratio of 25:1, both LR and KS significantly affected model accuracy, with p -values of 6.508×10^{-8} and 0.036, respectively. In the same dataset, the interaction effect between LR and KS ($LR \times KS$) also showed a significant impact, with an F-value of 27.600 and a p -value of 7.358×10^{-19} . The combination of these two parameters significantly influenced the model performance.

LR and KS exhibited significant effects on the CIFAR-10 dataset with an imbalance ratio of 15:1, with p -values of 1.622×10^{-11} and 4.925×10^{-7} , respectively. The interaction effect between LR and KS ($LR \times KS$) was particularly noteworthy, with an F-value of 38.768 and a p -value of 5.079×10^{-25} , indicating a highly significant impact.

In the balanced CIFAR-10 dataset, LR and KS still significantly affected accuracy, with p -values of 1.125×10^{-7} and 4.982×10^{-10} , respectively. Furthermore, the interaction effect between LR and KS ($LR \times KS$) also demonstrated a significant impact, with an F-value of 209.146 and a p -value of 5.673×10^{-17} . However, DR 1 and DR 2 did not exhibit significant main or interaction effects on any dataset, with p -values greater than 0.1.

Table 5. Partial performance metrics and parameter values for each run of the CIFAR-10 dataset with an imbalance ratio of 25:1.

Count	LR (5 Levels)	DR 1 (5 Levels)	DR 2 (5 Levels)	KS (2 Levels)	Accuracy (Y_A)	F1 Score (Y_F)	G-Mean (Y_G)	P-Mean (Y_P)	Recall (Y_R)
1	1	1	1	1	0.559	0.562	0.538	0.543	0.708
					0.549	0.554	0.530	0.533	0.701
					0.562	0.561	0.539	0.530	0.703
					0.554	0.533	0.533	0.533	0.706
2	1	1	2	1	0.592	0.555	0.543	0.538	0.711
					0.550	0.552	0.530	0.527	0.702
					0.555	0.554	0.543	0.543	0.710
					0.552	0.552	0.543	0.538	0.707
3	1	1	3	1	0.555	0.559	0.533	0.543	0.704
					0.543	0.556	0.518	0.539	0.693
					0.555	0.554	0.533	0.527	0.707
					0.552	0.556	0.533	0.543	0.704
4	1	1	4	1	0.552	0.537	0.535	0.507	0.705
					0.542	0.536	0.518	0.520	0.694
					0.545	0.537	0.535	0.520	0.698
					0.552	0.536	0.520	0.507	0.699
...	...	...	...	...	...	...	...	...	...
124	5	5	4	1	0.657	0.647	0.788	0.673	0.768
					0.667	0.648	0.787	0.674	0.767
					0.659	0.659	0.779	0.676	0.769
					0.654	0.661	0.789	0.672	0.776
125	5	5	5	1	0.647	0.642	0.780	0.674	0.768
					0.650	0.648	0.787	0.674	0.767
					0.645	0.643	0.781	0.675	0.768
					0.646	0.640	0.783	0.672	0.770
126	1	1	1	2	0.633	0.626	0.770	0.661	0.768
					0.632	0.625	0.769	0.659	0.767
					0.634	0.627	0.774	0.663	0.759
					0.631	0.624	0.772	0.662	0.764
...	...	...	...	...	...	...	...	...	...
247	5	5	2	2	0.578	0.571	0.561	0.559	0.726
					0.586	0.576	0.573	0.573	0.736
					0.578	0.571	0.561	0.559	0.726
					0.586	0.573	0.561	0.559	0.732
248	5	5	3	2	0.572	0.553	0.545	0.533	0.712
					0.590	0.550	0.553	0.556	0.736
					0.572	0.553	0.545	0.533	0.712
					0.590	0.553	0.545	0.533	0.736
249	5	5	4	2	0.563	0.568	0.525	0.525	0.718
					0.569	0.569	0.560	0.561	0.725
					0.563	0.568	0.525	0.525	0.718
					0.569	0.569	0.560	0.561	0.725
250	5	5	5	2	0.564	0.575	0.537	0.546	0.729
					0.569	0.564	0.555	0.548	0.715
					0.575	0.566	0.727	0.607	0.761
					0.564	0.548	0.715	0.611	0.755

Just as we conducted experiments on the CIFAR-10 dataset, we carried out the same experiments on the Fashion-MNIST dataset and achieved similar results. Across different imbalance ratios, we found significant main effects and interaction effects of LR and KS on the CNN model accuracy. In the Fashion-MNIST dataset with an imbalance ratio of 25:1, LR and KS exhibited p -values of 4.341×10^{-10} and 9.776×10^{-4} , respectively. The interaction effect (LR $\times$ KS) showed an F-value of 27.322 and a p -value of 1.069×10^{-18} . Under the 15:1 imbalance ratio, LR and KS had p -values and F-values of 1.338×10^{-8} and 8.591×10^{-4} , respectively, with an interaction effect (LR $\times$ KS) p -value of 3.810×10^{-8} and F-value of 13.800. In the balanced dataset, LR and KS significantly influenced the CNN model accuracy with p -values of 3.601×10^{-9} and 1.184×10^{-3} , respectively. The interaction effect (LR $\times$ KS) had an F-value of 12.222, although its impact was slightly diminished compared with that in the highly imbalanced dataset. These results underscore that the observed effects are not incidental but robust validations of their impact on model performance.

Table 6. ANOVA results for accuracy on the CIFAR-10 dataset with an imbalance ratio of 25:1.

	Degrees of Freedom	Sums of Squares	Mean Squares	F-Values	p -Values
LR	4	4172.556	1043.139	10.565	6.508×10^{-8}
DR 1	4	112.928	28.232	0.245	0.912
DR 2	4	171.201	42.800	0.372	0.828
KS	1	498.492	498.492	7.373	0.036
LR*DR 1	16	1390.666	86.917	0.862	6.137×10^{-1}
LR*DR 2	16	871.953	54.497	0.530	9.298×10^{-1}
LR*KS	4	7464.556	1866.139	27.600	7.358×10^{-19}
DR 1*DR 2	16	1145.995	71.625	0.598	0.883
DR 1*KS	4	49.896	12.474	0.108	0.979
DR 2*KS	4	104.523	26.131	0.227	0.922

Table 7. ANOVA results for accuracy on the CIFAR-10 dataset with an imbalance ratio of 15:1.

	Degrees of Freedom	Sums of Squares	Mean Squares	F-Values	p -Values
LR	4	4264.931	1066.232	15.958	1.622×10^{-11}
DR 1	4	217.170	54.292	0.812	5.182×10^{-1}
DR 2	4	58.650	14.662	0.213	9.309×10^{-1}
KS	1	155.630	155.630	7.929	4.925×10^{-2}
LR*DR 1	16	646.592	40.412	0.604	8.784×10^{-1}
LR*DR 2	16	361.287	22.580	0.328	9.937×10^{-1}
LR*KS	4	6178.579	1544.644	38.768	5.079×10^{-25}
DR 1*DR 2	16	646.767	40.422	0.598	0.958
DR 1*KS	4	205.396	51.349	0.629	0.642
DR 2*KS	4	115.262	28.815	0.348	0.844

Table 8. ANOVA results for accuracy on the CIFAR-10 dataset with an imbalance ratio of 1:1.

	Degrees of Freedom	Sums of Squares	Mean Squares	F-Values	p -Values
LR	4	377.506	94.376	27.176	1.125×10^{-7}
DR 1	4	8.587	2.146	0.431	0.786
DR 2	4	4.653	1.163	0.232	0.919
KS	1	177.729	177.729	41.954	4.982×10^{-10}
LR*DR 1	16	19.541	1.221	0.334	9.930×10^{-1}
LR*DR 2	16	12.175	0.760	0.205	9.996×10^{-1}
LR*KS	4	523.034	130.758	209.146	5.673×10^{-17}
DR 1*DR 2	16	4.330	0.270	0.050	0.929
DR 1*KS	4	3.162	0.790	0.182	9.472×10^{-1}
DR 2*KS	4	2.519	0.629	0.144	9.651×10^{-1}

The previous section demonstrates the significant impact of various parameters on accuracy. To further explore the effects of these parameters on recall under different imbalance ratios, we conducted an analysis of variance to obtain the p -values for the main effects and interaction effects of learning rate, dropout rate 1, dropout rate 2, and kernel size on the accuracy of minority class samples. Due to the substantial differences in the p -values, we standardized each p -value using the following formula:

$$\text{Normalized Value} = \frac{x - \text{Minimum}}{\text{Maximum} - \text{Minimum}} \quad (20)$$

This standardization process ensures that the effects of different parameters at various imbalance ratios can be compared on the same scale, thereby guaranteeing fairness and accuracy in the data analysis. By employing this method, we were able to illustrate and compare the substantial impact of various parameters on recall.

As can be seen from Figure 6, the impact of different parameters on the model performance varied across different imbalance ratios. In highly imbalanced datasets, LR, KS, and their interaction effects significantly influenced the model. In moderately imbalanced datasets, the effects of LR and KS were more pronounced. In balanced datasets, the interaction effects of LR and KS ($\text{LR} \times \text{KS}$) and the interaction effects of DR 1 and KS ($\text{DR 1} \times \text{KS}$) significantly impacted model performance. Based on these findings, it is possible to more precisely select and adjust the parameters of CNN models to improve their performance on imbalanced data.

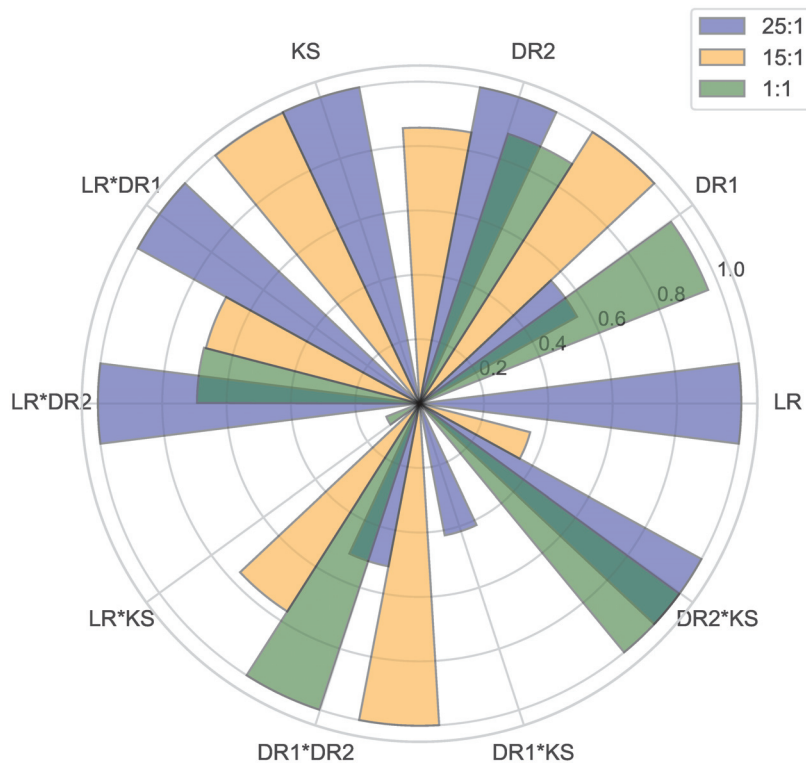


Figure 6. Effect of KS on recall in datasets with different imbalance ratios from the CIFAR-10 dataset.

In summary, LR and KS significantly impacted the overall model performance across all imbalanced datasets, especially in highly imbalanced datasets. The interaction between these two parameters also significantly influences the model's performance. In balanced datasets, LR and KS still significantly affect the model performance. However, their effects are less pronounced than those for highly imbalanced datasets, whereas minor parameters such as DR 1 and DR 2 have a minor impact on model performance. In subsequent optimizations, attention should be focused on LR and KS, and their optimal combinations should be explored to enhance the performance of CNN models on imbalanced datasets.

5.3.2. Detailed Analysis of 25:1 Imbalanced Data on the CIFAR-10 Dataset

To gain a deeper understanding of the impact of the parameters on extremely imbalanced datasets, we conducted a detailed analysis of data with an imbalance ratio of 25:1. At this ratio, the model performance is susceptible to parameter settings, making the identification of critical parameters crucial for optimizing the model performance.

Because the independent variables that significantly impact the evaluation metrics are LR and KS, we do not discuss the main effect results of the dropout rate in detail. Figure 7 shows the main effects of LR on various metrics, highlighting the significant impact of LR on each metric. The best performance for each metric occurred when LR ranged from 0.001 to 0.031, peaking at approximately 0.031. Therefore, selecting an appropriate LR is crucial for imbalanced data classification and can significantly enhance various metrics.

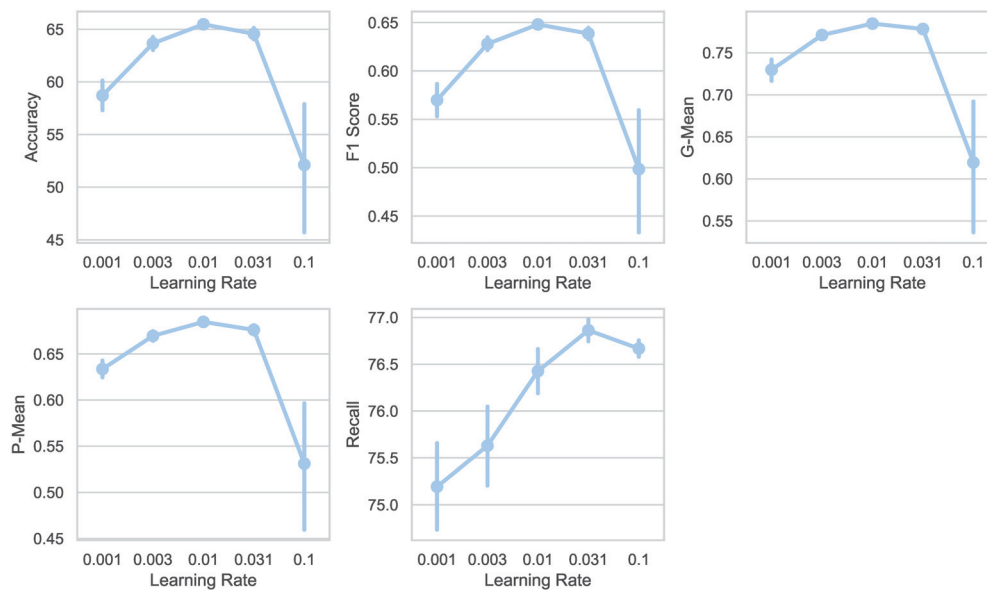


Figure 7. Plot of main effect of learning rate on evaluation metrics for the CIFAR-10 dataset.

Figure 8 shows the overall impact of KS on various evaluation metrics for imbalanced and balanced data. The results indicate that smaller kernel sizes perform better in terms of accuracy, F1 score, G-mean, and P-mean, reflecting better overall balanced performance. On the other hand, larger kernel sizes significantly improve recall, but cause a decline in other performance metrics; thus, there is a significant trade-off when choosing the kernel size for imbalanced data classification. The selection of the most suitable kernel configuration should consider the specific performance requirements of the application and the importance of the different evaluation metrics.

Figure 9 illustrates the interaction effects of different parameter combinations on the accuracy of the imbalanced data classification. In the first subplot, the x-axis represents the LR, the y-axis represents the mean accuracy, and the colors denote different levels of DR 1. It can be observed that accuracy increases as the learning rate increases from 0.001 to 0.031 but sharply declines at a learning rate of 0.1. The impact of different DR 1 levels on the accuracy is relatively minor. Similar to in the previous subplot, the effect of different DR 2 levels on the accuracy is also relatively small. Additionally, the combined effects of DR 1 and DR 2 on the accuracy are complex.

In summary, LR and KS are the primary factors that significantly affect various performance metrics, while DR 1, DR 2, and their interactions have a minor impact on these metrics. To optimize the performance, the focus should be on the learning rate, kernel size, and interaction. LR = 0.01, a moderate learning rate, provides an adequate gradient update step size during model training. This allows the model to effectively learn the features of the training data while avoiding oscillations and instability caused by overly large gradient

steps. DR 1 = 0.02, DR 2 = 0.02, and relatively low dropout rates help prevent overfitting while ensuring the model's training efficiency and generalization ability. These two parameters enhance the robustness of the model while maintaining its simplicity. $KS = 3 \times 3$, a smaller kernel size, allows the model to capture detailed image features better, thereby improving its ability to recognize minority class samples. The smaller kernel also reduces the number of parameters, thereby lowering the computational complexity. In the dataset with an imbalance ratio of 25:1, using fractional factorial design and ANOVA, we identified that the optimal parameter combination is A3B2C2D1.

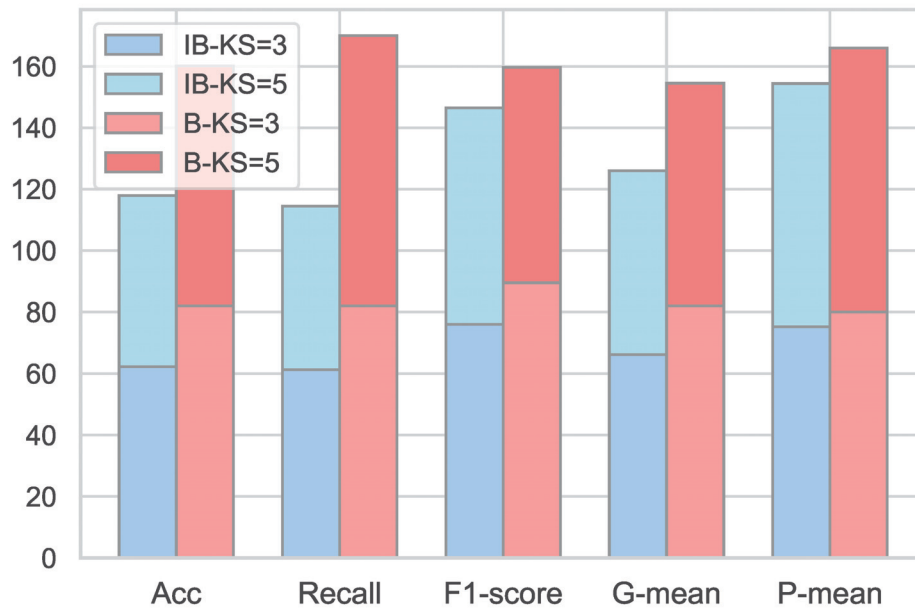


Figure 8. Plot of main effect of KS on evaluation metrics in highly imbalanced and balanced datasets from the CIFAR-10. (IB stands for imbalanced data, and B stands for balanced data).

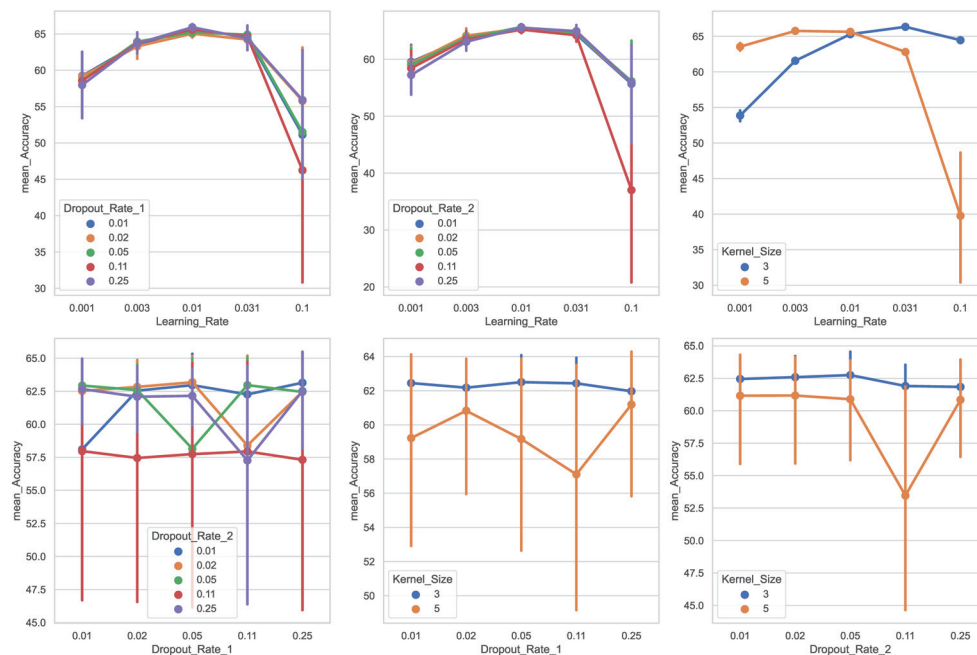


Figure 9. Plot of interaction effect of four variables on evaluation metrics for the CIFAR-10 dataset.

5.3.3. Detailed Analysis of 25:1 Imbalanced Data on the Fashion-MINST Dataset

To validate the model's reliability, enhance the credibility of the research, and demonstrate its performance across various scenarios, we conducted experiments using 25:1 imbalanced data extracted from the Fashion-MNIST dataset. Based on the setup in Table 3, we considered four factors: LR, DR 1, DR 2, and KS. The first three factors had five levels, labeled from A1 to C5, whereas KS had two levels. We performed an ANOVA to evaluate the main effects of these parameters and their interaction effects on the minority class samples. The experimental design included 250 experiments across the four factors, each repeated four times to ensure the robustness and accuracy of the results.

Based on the ANOVA results for the Fashion-MNIST dataset, as shown in Figure 10, the optimal parameter combination A2B3C3D1 was determined through an analysis of the interaction effects of LR, DR 1, DR 2, and KS. Specifically, with a lower learning rate of 0.031, the model exhibited higher and more stable accuracy when both dropout rate 1 and dropout rate 2 were 0.05, and the model's average accuracy under different combinations was relatively good and stable. When the kernel size was 3, the model performed most stably across various parameter combinations. In summary, the optimal scheme is a learning rate of 0.031, dropout rate 1 of 0.05, dropout rate 2 of 0.05, and kernel size of 3. This combination ensures the model performance while reducing the risk of overfitting and instability.

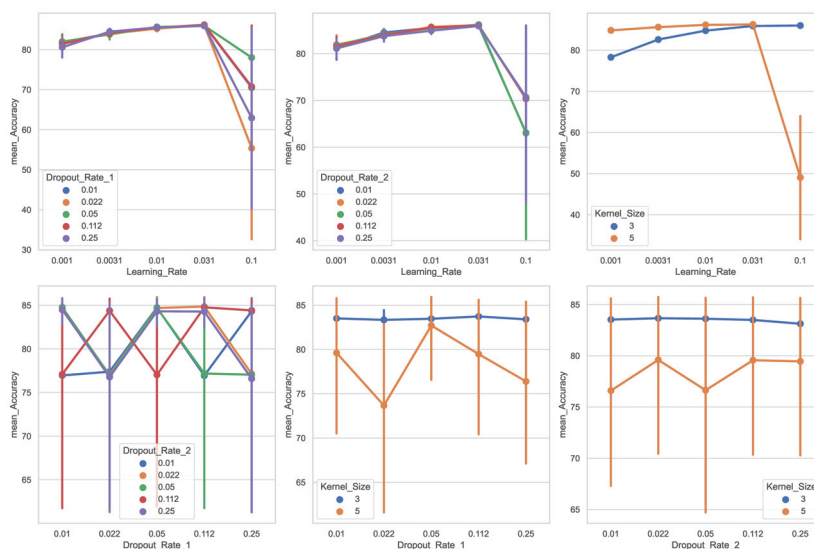


Figure 10. Plot of interaction effect of four variables on evaluation metrics for the Fashion-MNIST dataset.

5.4. Parameter Optimization

The basic framework of the models we used included five convolutional layers, each with a 3×3 kernel size, three max-pooling layers, and three dropout layers, as shown in Table 4. In our experiments, we designed five different CNN models to explore the effect of parameter tuning on model performance on an imbalanced dataset. We compared these with a traditional undersampling model for handling imbalanced data.

Model 1: This is the baseline model using the default number of convolution kernels (32, 64, 96, 128, 256), a default learning rate of 0.001, kernel size of 3×3 , and dropout rate of 0.2.

Model 2: Based on Model 1, the number of convolution kernels is adjusted to 103, 105, 107, 109, and 179 while the other settings are unchanged.

Model 3: Similar to Model 2 but with a kernel size of 5×5 , keeping the other parameters the same.

Model 4: Based on the baseline model, this model adjusts both the number of convolution kernels and learning rate while the other parameters remain at their default values.

Model 5: The adjustments made in Models 3 and 4 are combined by changing the convolution kernel number, size, and learning rate.

The models were trained using the cross-entropy loss function, and evaluated based on five metrics: accuracy, recall, F1 score, G-mean, and P-mean. These metrics were employed to thoroughly assess the performance across each data category.

During the training process, the number of epochs was not fixed. Instead, we set a stopping condition; if the difference in training accuracy between the current epoch and the previous epoch was less than 1×10^{-2} (i.e., 0.01), the training stopped. This approach helped avoid overtraining the model once a certain accuracy level was reached.

To optimize the CNN's performance on the minority class samples, we employed the two significant parameters mentioned above: LR and KS. By iteratively adjusting these parameters, we identified the optimal network configuration that performs best in terms of primary and interaction effects, enhancing the recognition capability of minority class samples.

5.4.1. Results of CIFAR-10 Dataset

Table 9 presents the final experimental results, with accuracy representing the overall accuracy and the remaining metrics evaluating the performance of minority class samples. Based on the analysis of the experimental results, we observed that modifying the kernel size and learning rate had a significant impact on the CNN model's performance, especially with imbalanced datasets. Model 1, serving as the baseline model, showed lower metrics, particularly recall and F1 score, at 0.22 and 0.36, respectively, indicating poor performance on minority class samples. Although Model 2's overall accuracy decreased to 0.49, its recall significantly improved to 0.16, demonstrating an enhanced ability to recognize minority class samples. Model 3, with the kernel size adjusted to 5×5 , significantly improved the overall accuracy, recall, F1 score, and G-mean, reaching 0.65, 0.33, 0.72, and 0.78, respectively, indicating a substantial improvement in recognition and accuracy for the minority class samples. Model 4 adjusted only the learning rate, achieving a 0.67 overall accuracy and improved performance on minority class samples, particularly in F1 score and G-mean, with results of 0.71 and 0.76, respectively. Model 5, which combined adjustments to the number of kernels, kernel size, and learning rate, achieved performance metrics close to those of Model 4 and slightly better than those of Model 3, with a recall of 30.96% and an F1 score of 0.71. The results demonstrate that simultaneously adjusting multiple parameters can yield better performance, as shown in Figure 11.

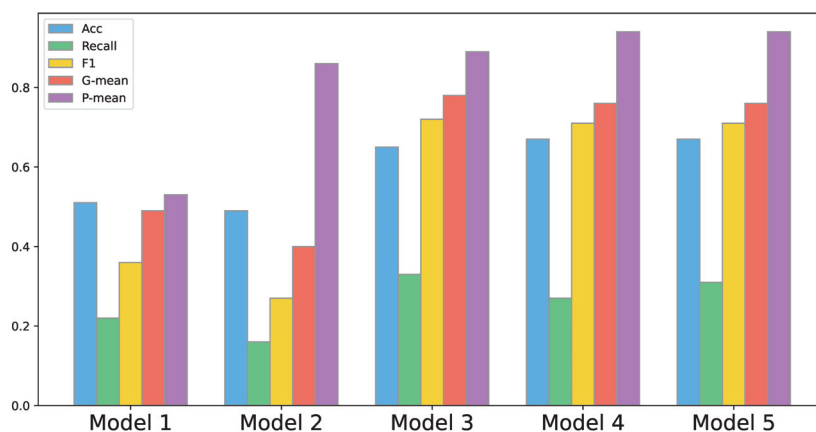


Figure 11. Performance metrics of minority class samples across the five models for the CIFAR-10 dataset.

Table 9. Performance of minority class samples in the CIFAR-10 dataset across five models.

Model	Accuracy	Recall	F1	G-Mean	P-Mean
1	0.51	0.22	0.36	0.49	0.53
2	0.49	0.16	0.27	0.40	0.86
3	0.65	0.33	0.72	0.78	0.89
4	0.67	0.27	0.71	0.76	0.94
5	0.67	0.34	0.71	0.76	0.94

Based on the analysis of variance and tuning process described above, the optimal configuration for a dataset with an imbalance ratio of 25:1 was confirmed to be (0.013, 0.2, 0.2, 3×3).

Table 10 presents the performance metrics of the parameter-optimized CNN model, Model 1 with undersampling, and Model 1 with SMOTE for the CIFAR-10 dataset for each class. Compared to the standard data-processing methods, the parameter-optimized model demonstrated superior performance in most categories in terms of recall, F1 score, G-mean, and P-mean, especially for the minority class. In this study, the minority class corresponds to the automobile category. For the minority class, the recall, F1 score, G-mean, and P-mean of Model 5 reached 0.31, 0.71, 0.76, and 0.94, respectively, whereas Model 1 with undersampling only achieved 0.23, 0.37, 0.48, and 0.52, respectively. The results indicate that, under a 25:1 imbalance ratio, optimizing the parameter selection improves the classification performance of the minority class, highlighting the importance of optimizing parameter selection in handling imbalanced data scenarios.

Compared to undersampling, Model 1 with SMOTE showed mixed results for the minority class. The SMOTE resulted in a lower recall than undersampling, only reaching 0.12, though it slightly improved the F1 score to 0.22. Despite the slight improvement in F1 score, the G-mean and P-mean values for the SMOTE were still significantly lower than those of the parameter-optimized Model 5. This indicates that, while the SMOTE's synthetic samples provide some improvements in certain metrics, its effectiveness in enhancing recall for the minority class is limited. This further underscores that, in cases of extreme imbalance, optimizing model parameters is more effective than solely relying on SMOTE oversampling.

Table 10. Performance metrics for each class in the CIFAR-10 dataset (bold text indicates Model 5).

	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Recall	0.90	0.31	0.38	0.59	0.46	0.43	0.85	0.53	0.73	0.91
	0.87	0.23	0.35	0.58	0.22	0.32	0.81	0.38	0.79	0.84
	0.90	0.12	0.32	0.58	0.40	0.42	0.19	0.28	0.74	0.79
F1 score	0.68	0.71	0.49	0.57	0.54	0.60	0.77	0.69	0.82	0.77
	0.60	0.37	0.41	0.44	0.33	0.41	0.70	0.52	0.73	0.65
	0.54	0.22	0.38	0.41	0.43	0.43	0.32	0.42	0.70	0.60
G-mean	0.91	0.76	0.61	0.78	0.65	0.73	0.91	0.79	0.88	0.93
	0.88	0.48	0.58	0.72	0.47	0.56	0.88	0.61	0.87	0.88
	0.87	0.35	0.55	0.71	0.62	0.63	0.44	0.52	0.84	0.85
P-mean	0.55	0.94	0.71	0.50	0.72	0.64	0.70	0.75	0.86	0.66
	0.45	0.52	0.49	0.47	0.52	0.50	0.45	0.49	0.46	0.45
	0.37	0.46	0.41	0.39	0.40	0.40	0.44	0.42	0.38	0.37

In terms of the G-mean and P-mean metrics, Model 5 demonstrated a more balanced and consistent performance. For other class data, Model 5 achieved a recall rate of 0.90 for the airplane category, compared to 0.87 achieved by Model 1 with undersampling. In the frog category, Model 5 had an F1 score of 0.77, while Model 1 had an undersampling score of 0.70.

Figure 12 illustrates this phenomenon. Compared with using undersampling data processing alone (red region), the model with parameter optimization (blue region) significantly improved the metrics for minority classes (such as the car category). Recall rate, F1 score, G-mean, and P-mean were significantly enhanced. The results indicate that parameter optimization notably improved the model's ability to identify minority class samples. Furthermore, this method not only improved the performance of minority class samples, but also maintained high performance across other classes and, in some cases, showed improvement.

This further underscores the importance of parameter selection, particularly when dealing with minority class samples. Optimizing the parameter settings can significantly enhance the recognition capabilities and classification performance, resulting in better performance on imbalanced datasets.

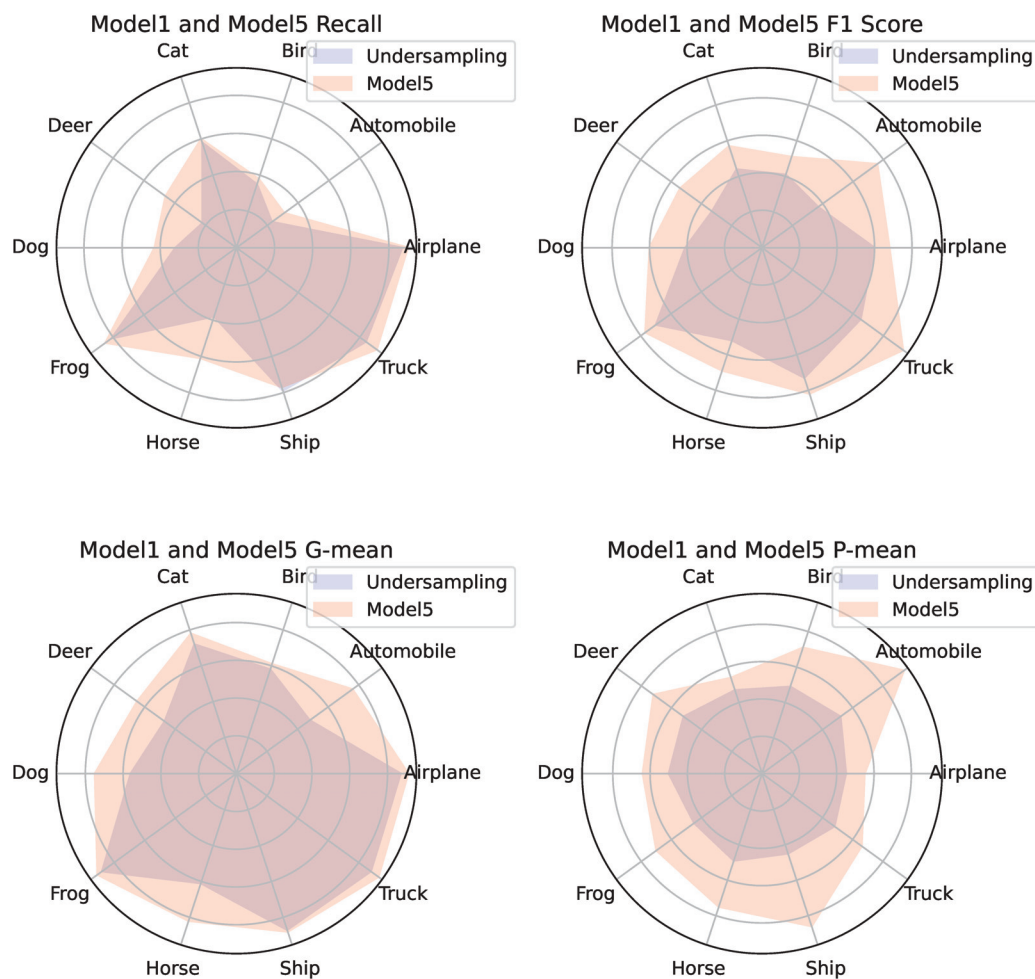


Figure 12. Evaluation metrics of Model 5 and undersampling compared to Model 5 on various classes in the CIFAR-10 dataset.

5.4.2. Results of Fashion-MNIST Dataset

The same methodology was applied to the Fashion-MNIST dataset. Based on the ANOVA results, the optimal configuration for the 25:1 imbalanced dataset was determined to be (0.031, 0.05, 0.05, 3×3). The results obtained from parameter optimization, specifically, the results of Model 5, were compared with the performance of the five models defined above. Figure 13 and Table 11 present the final results of the five models. The experimental findings demonstrate that adjusting the number of convolution kernels, their size, and the learning rate affects the performance of CNN models, especially in handling imbalanced datasets. Model 1 showed poor performance on minority class samples with a recall of 0.28

and an F1 score of 0.42. Model 2 adjusted the number of convolution kernels and improved the recall to 0.31, with a slight increase in overall accuracy. Model 3 further changed the kernel size to 5×5 , maintaining an accuracy of 0.84 but experiencing slight decreases in the recall and F1 score. Model 4, adjusting the learning rate, showed improved performance with a recall of 0.28 and an F1 score of 0.41. Model 5, which combined adjustments to kernel number, size, and learning rate, achieved the best performance with an overall accuracy of 0.87, recall of 0.38, and F1 score of 0.50. These results are consistent with those of the experiments on the CIFAR-10 dataset, highlighting the significant impact of parameter selection on the recognition performance of minority class samples.

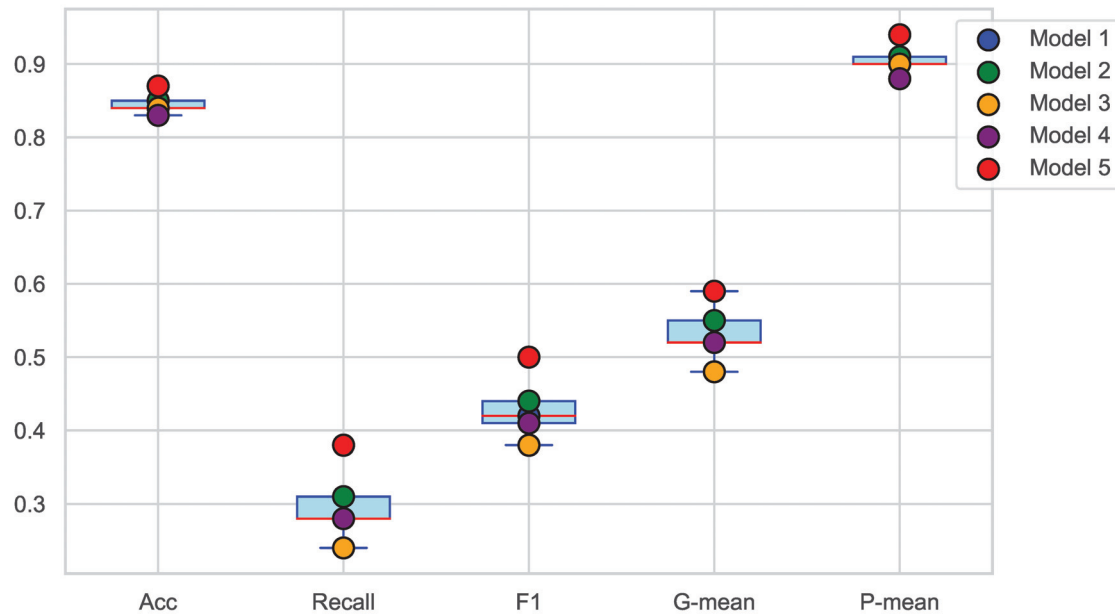


Figure 13. Performance metrics box plot of the Fashion-MNIST dataset across five models with model highlights.(The red line represents the median).

Table 11. Performance of minority class samples on the Fashion-MNIST dataset across five models.

Model	Accuracy	Recall	F1	G-Mean	P-Mean
1	0.84	0.28	0.42	0.52	0.90
2	0.85	0.31	0.44	0.55	0.91
3	0.84	0.24	0.38	0.48	0.90
4	0.83	0.28	0.41	0.52	0.88
5	0.87	0.38	0.50	0.59	0.94

Table 12 presents the performance metrics for each class for the Fashion-MNIST dataset for Model 5 and the undersampled model, with bold text representing the results of Model 5. Both T-shirt and shirt are minority samples. Model 5 performed exceptionally well for class 1, with a recall of 0.97, an F1 score of 0.78, a G-mean of 0.95, and a P-mean of 0.77. In contrast, for class 6, the undersampled model achieved a recall and F1 score of 0.46 and 0.53, respectively, while these metrics significantly improved to 0.88 and 0.61 through parameter adjustments in Model 5.

Specifically, the data in the table show that, although class 6 performed poorly after undersampling, its recall and F1 score significantly improved through parameter adjustments in Model 5. The results indicate that appropriate adjustments to the number and size of convolution kernels and the learning rate can effectively enhance the recognition performance of minority samples (especially class 6), thereby improving the model's overall performance when handling imbalanced datasets.

Table 12. Performance metrics for each class in the Fashion-MNIST dataset (bold text indicates Model 5).

	T-Shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle Boot
Recall	0.97	0.96	0.90	0.90	0.75	0.98	0.46	0.84	0.97	0.99
	0.97	0.96	0.87	0.90	0.62	0.98	0.38	0.88	0.97	0.98
F1 score	0.78	0.97	0.75	0.89	0.73	0.98	0.53	0.95	0.97	0.98
	0.77	0.78	0.75	0.87	0.70	0.97	0.44	0.92	0.97	0.95
G-mean	0.95	0.97	0.91	0.94	0.84	0.98	0.61	0.94	0.98	0.98
	0.95	0.97	0.91	0.94	0.78	0.98	0.51	0.88	0.98	0.98
P-mean	0.77	0.77	0.78	0.77	0.81	0.77	0.95	0.78	0.78	0.77
	0.77	0.77	0.78	0.77	0.79	0.77	0.91	0.78	0.77	0.77

In summary, Table 12 and Figure 14 indicate that Model 5, through adjustments to the number and size of convolution kernels and learning rate, significantly improved the performance of minority class samples in imbalanced datasets. This improvement was particularly notable for class 6, the shirt category. Model 5 demonstrated superior overall performance in terms of recall, F1 score, G-mean, and P-mean compared with the undersampling method, highlighting the effectiveness of parameter adjustments in enhancing the recognition performance of minority class samples.

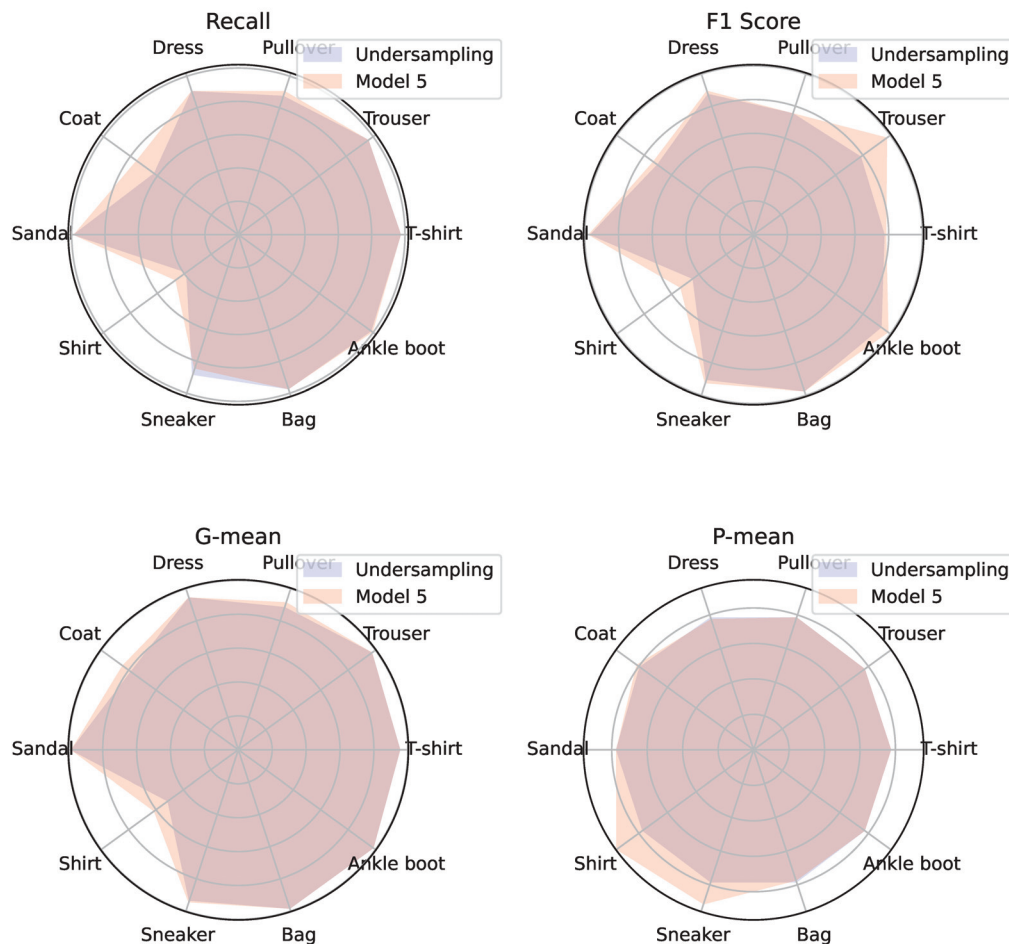


Figure 14. Evaluation metrics of Model 5 and undersampling compared to Model 5 on various classes in the Fashion-MNIST dataset.

6. Conclusions

This study aimed to investigate the impact of CNN parameter distributions on imbalanced data and improve the classification performance of minority class samples by optimizing the CNN model parameters. Quantitative evaluation metrics included accuracy, recall, F1 score, G-mean, and P-mean. We found that parameter selection significantly influence imbalanced data and specifically optimized additional hyperparameters of the CNN model for the CIFAR-10 dataset with a maximum imbalance ratio of 25:1 and the Fashion-MNIST dataset. The results demonstrate that the learning rate and kernel size notably affect the performance of the CNN model for imbalanced data. In CIFAR-10, adjusting the learning rate to 0.013 and setting the kernel size to 3×3 , and, in Fashion-MNIST, adjusting the learning rate to 0.0031 and setting the kernel size to 3×3 , led to improvements in accuracy by 14.20% and 3.87%, respectively, compared to traditional undersampling methods. For CIFAR-10, the recall, F1 score, G-mean, and P-mean of the minority class samples improved by 8%, 34%, 28%, and 42%, respectively, compared to the undersampling methods. For Fashion-MNIST, these metrics improved by 12%, 9%, 10%, and 4%, respectively. These findings underscore the importance of parameter selection in handling imbalanced data, particularly kernel size and learning rate adjustment, which play a crucial role in recognizing minority class samples.

Despite achieving significant results in this study, further exploration of parameter optimization methods is necessary. This study focused solely on understanding and studying a single CNN model and did not explore other models. Considering the relatively scarce research on deep learning methods in the context of imbalanced and large datasets, further exploration will significantly advance deep learning methods in big data analytics. Future research on deep learning will likely continue in this challenging direction and attract continued attention from those in academia and industry. For more details, please refer to our GitHub project: <https://github.com/2221335/Ao> (accessed on 27 September 2024).

Author Contributions: Conceptualization, N.W.; methodology, N.W.; software, R.Z.; validation, R.Z. and N.W.; formal analysis, R.Z.; investigation, R.Z.; writing—original draft preparation, R.Z.; writing—review and editing, N.W.; visualization, R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project of “Double First-Class” Disciplines in Heilongjiang Province, grant number LJGXCG2022-057.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used for this study are publicly available.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. AlexNet: ImageNet classification with deep convolutional neural networks. *Proc. NIPS* **2018**, 1097–1105.
2. Seera, M.; Lim, C.P. A hybrid intelligent system for medical data classification. *Expert Syst. Appl.* **2014**, *41*, 2239–2249. [CrossRef]
3. Alam, T.M.; Shaikat, K.; Khan, W.A.; Hameed, I.A.; Almuqren, L.A.; Raza, M.A.; Aslam, M.; Luo, S. An efficient deep learning-based skin cancer classifier for an imbalanced dataset. *Diagnostics* **2022**, *12*, 2115. [CrossRef]
4. Awoyemi, J.O.; Adetunmbi, A.O.; Oluwadare, S.A. Credit card fraud detection using machine learning techniques: A comparative analysis. In Proceedings of the 2017 International Conference on Computing Networking and Informatics (ICCNi), Lagos, Nigeria, 29–31 October 2017; IEEE: New York, NY, USA, 2017; pp. 1–9.
5. Sun, Y.; Zhao, T.; Zou, Z.; Chen, Y.; Zhang, H. Imbalanced data fault diagnosis of hydrogen sensors using deep convolutional generative adversarial network with convolutional neural network. *Rev. Sci. Instrum.* **2021**, *92*, 095007. [CrossRef]
6. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
7. Drummond, C.; Holte, R.C. C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In Proceedings of the Workshop on Learning from Imbalanced Datasets II, Washington, DC, USA, 21 August 2003; Volume 11.
8. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]

9. Elkan, C. The foundations of cost-sensitive learning. In Proceedings of the International Joint Conference on Artificial Intelligence, Seattle, WA, USA, 4–10 August 2001; Lawrence Erlbaum Associates Ltd.: Mahwah, NJ, USA, 2001; Volume 17, pp. 973–978.
10. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.)* **2011**, *42*, 463–484. [CrossRef]
11. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
13. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
14. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
15. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
16. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Wang, S.; Yu, Q.; Zhang, X.; Liu, B.; Zhang, C. Learning from imbalanced data: A review and experimental study. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–35.
19. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
20. Das, M.N.; Giri, N.C. *Design and Analysis of Experiments*; New Age International: Delhi, India, 1979.
21. Lujan-Moreno, G.A.; Howard, P.R.; Rojas, O.G.; Montgomery, D.C. Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Syst. Appl.* **2018**, *109*, 195–205. [CrossRef]
22. Garofalo, S.; Giovagnoli, S.; Orsoni, M.; Starita, F.; Benassi, M. Interaction effect: Are you doing the right thing? *PLoS ONE* **2022**, *17*, e0271668. [CrossRef]
23. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Special issue on learning from imbalanced data sets. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 1–6. [CrossRef]
24. Shin, K.; Han, J.; Kang, S. MI-MOTE: Multiple imputation-based minority oversampling technique for imbalanced and incomplete data classification. *Inf. Sci.* **2021**, *575*, 80–89. [CrossRef]
25. Wang, X.; Xu, J.; Zeng, T.; Jing, L. Local distribution-based adaptive minority oversampling for imbalanced data classification. *Neurocomputing* **2021**, *422*, 200–213. [CrossRef]
26. Seliya, N.; Khoshgoftaar, T.M.; Van Hulse, J. A study on the relationships of classifier performance metrics. In Proceedings of the 2009 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, NJ, USA, 2–4 November 2009; IEEE: New York, NY, USA, 2009; pp. 59–66.
27. Mani, I.; Zhang, I. kNN approach to unbalanced data distributions: A case study involving information extraction. In Proceedings of the Workshop on Learning from Imbalanced Datasets, Washington, DC, USA, 21 August 2003; ICML: Vienna, Austria, 2003; Volume 126, pp. 1–7.
28. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
29. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; IEEE: New York, NY, USA, 2008; pp. 1322–1328.
30. Huang, C.; Li, Y.; Loy, C.C.; Tang, X. Learning deep representation for imbalanced classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5375–5384.
31. Mease, D.; Wyner, A.J.; Buja, A. Boosted classification trees and class probability/quantile estimation. *J. Mach. Learn. Res.* **2007**, *8*, 409–439.
32. Wu, C.; Wang, N.; Wang, Y. Increasing Minority Recall Support Vector Machine Model for Imbalanced Data Classification. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 6647557. [CrossRef]
33. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
34. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
35. Zhong, X.; Wang, N. Ensemble learning method based on CNN for class imbalanced data. *J. Supercomput.* **2024**, *80*, 10090–10121. [CrossRef]
36. Zhou, B.; Cui, Q.; Wei, X.S.; Chen, Z.M. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9719–9728.
37. Ilham, A.; Silva, J.; Mercado-Caruso, N.; Tapias-Ruiz, D.; Lezama, O.B.P. Impact of class imbalance on convolutional neural network training in multi-class problems. In Proceedings of the Image Processing and Capsule Networks: ICIPCN 2020, Bangkok, Thailand, 6–7 May 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 309–318.

38. Liao, L.; Zhao, Y.; Wei, S.; Wei, Y.; Wang, J. Parameter distribution balanced CNNs. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4600–4609. [CrossRef] [PubMed]
39. Yoo, J.H.; Yoon, H.i.; Kim, H.G.; Yoon, H.S.; Han, S.S. Optimization of hyper-parameter for CNN model using genetic algorithm. In Proceedings of the 2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), Kuala Lumpur, Malaysia, 25 November 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
40. Temraz, M.; Keane, M.T. Solving the class imbalance problem using a counterfactual method for data augmentation. *Mach. Learn. Appl.* **2022**, *9*, 100375. [CrossRef]
41. Nasiri, H.; Alavi, S.A. A Novel Framework Based on Deep Learning and ANOVA Feature Selection Method for Diagnosis of COVID-19 Cases from Chest X-Ray Images. *Comput. Intell. Neurosci.* **2022**, *2022*, 4694567. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Study on Image Classification Algorithm Based on Multi-Scale Feature Fusion and Domain Adaptation

Yu Guo ^{1,†}, Ziyi Cheng ^{2,†}, Yuanlong Zhang ³, Gaoxuan Wang ^{4,*} and Jundong Zhang ¹¹ Marine Engineering College, Dalian Maritime University, Dalian 116000, China² Department of Electrical and Electronic Engineering, The University of Manchester, Manchester M13 9PL, UK³ School of Mathematical Sciences, University of Nottingham, Nottingham NG9 2SE, UK⁴ National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430078, China

* Correspondence: gxwang@cug.edu.cn

† These authors contributed equally to this work.

Abstract: This paper introduces the MMTADAN, an innovative algorithm designed to enhance cross-domain image classification. By integrating multi-scale feature extraction with Taylor series-based detail enhancement and adversarial domain adaptation, the MMTADAN effectively aligns features between the source and target domains. The proposed approach addresses the critical challenge of generalizing classification models across diverse datasets, demonstrating significant improvements in performance. The findings suggest that retaining essential image details through multi-scale extraction and Taylor series enhancement can lead to better classification outcomes, making the MMTADAN a valuable contribution to the field of image classification.

Keywords: statistical learning; deep learning; domain adaptation; generative adversarial network

1. Introduction

Machine learning, as a discipline that equips machines with “intelligence”, has become deeply integrated into various sectors of our lives and industries [1–4]. Cheng et al. [5] present a novel self-supervised adversarial training method for Monocular Depth Estimation (MDE) models, enhancing robustness against physical attacks by leveraging view synthesis and incorporating ℓ_0 -norm-bounded perturbations, without the need for ground-truth depth. Fang et al. [6] introduce a smooth and time-optimal S-curve trajectory planning method for robotic manipulators, using a piecewise sigmoid function to create infinitely differentiable trajectories, balancing efficiency and smoothness under given constraints, with validation through simulations and experiments. Liu et al. [7] propose an Approximate Maximum Likelihood Estimator (AMLE) to efficiently estimate the instantaneous frequency (IF) for nonstationary signals with intersecting or closely spaced IFs, improving upon the traditional MLE by reducing the computational complexity and handling time-varying amplitudes. Fang et al. [8] present a methodology for generating online, smooth joint trajectories for robots using an improved sinusoidal jerk model, offering a computationally efficient solution for time-optimal trajectory planning with kinematic constraints, leading to a better performance in terms of efficiency and jerk suppression compared to the existing techniques. Qin et al. [9] present a method based on the synchroextracting chirplet transform (SECT) for early chatter recognition in robotic drilling, demonstrating a superior performance in detecting chatter onset more quickly and effectively than the existing methods, making it practical for real-time vibration suppression. Sun et al. [10] propose a variable-scale wavelet packet entropy (VSWPE) method for detecting machining chatter, including scenarios with the beat effect, using beat frequency estimation and optimal demodulation techniques to improve the detection accuracy and prevent false alarms, demonstrating effectiveness in both simulations and machining tests.

In real-world applications, machine learning has seen considerable success in areas such as computer vision, natural language processing, and recommendation systems [11–16]. Computer vision, in particular, is a significant research area that simulates human visual systems to understand the real world through computational tools. From early efforts focused on manually designing image features to current research using deep learning algorithms for automated feature extraction, image analysis has remained a key component of artificial intelligence research. The hope is to use AI technologies to address challenges in the field of computer vision. In contrast to supervised learning with labeled data, today's information-rich world produces vast amounts of unlabeled image data across industries. These unlabeled datasets, though easy to obtain and rich in information, pose challenges in training deep learning models. This has created an urgent need to resolve the issue of how to effectively use small amounts of labeled data while incorporating large volumes of unlabeled data into deep learning models. Moreover, transferring a deep learning model trained on one dataset to another related but different dataset is a pressing challenge in the current deep learning development. One promising research direction addressing these challenges is domain adaptation. Traditional machine learning and deep learning tasks typically assume that the training data and deployment data share the same distribution, but this assumption often does not hold in practice. Thus, domain adaptation has become a critical research area.

Sohail et al. [17] provide a comprehensive review of deep transfer learning (DTL) and domain adaptation (DA) techniques for 3D point cloud (3DPC) processing, highlighting recent advancements, datasets, evaluation metrics, and applications like object detection, segmentation, and denoising while addressing current challenges and suggesting future research directions. Özince et al. [18] introduce a suite of sparsity-aware complex-valued least-mean kurtosis (CLMK) algorithms, including l0-CLMK, l0-ACLМК, ZA-CLMK, ZA-ACLМК, RZA-CLMK, and RZA-ACLМК, aimed at improving sparse system identification. Simulation results demonstrate superior performances in the convergence rate, tracking, and steady-state error compared to the existing sparsity-aware algorithms in both synthetic and real-world scenarios. Lu et al. [19] propose the generalized Jensen–Rényi divergence (GJRD) as a method for efficiently handling multiple data distributions in machine learning scenarios, addressing the limitations of traditional pairwise divergence measures. A non-parametric empirical estimator based on kernel density estimation is derived for the GJRD, which is then integrated into a deep clustering framework (GJRD-DC). Experimental results show that the GJRD-DC achieves state-of-the-art performances on challenging datasets, and the code is available online. Peng et al. [20] propose novel regularization-based frequency-domain diffusion algorithms to address the limitations of the existing methods for distributed estimation with missing inputs, offering an enhanced performance through bias elimination, power normalization, and fast convergence under colored inputs, as well as providing stability analysis and effective power estimation techniques. Xian et al. [21] propose a novel approach for unsupervised person re-identification by leveraging multiple-source datasets, using expert-specific clustering and dual-similarity distillation to enhance the domain adaptation while maintaining the feature diversity through representation decorrelation, with experiments validating its effectiveness on key benchmarks. Fang et al. [22] present a source-free collaborative domain adaptation (SCDA) framework for resting-state fMRI, addressing cross-site data heterogeneity by utilizing a pretrained source model and unlabeled target data, with a multi-perspective feature enrichment method and unsupervised pretraining, demonstrating effectiveness in cross-scanner and cross-study prediction.

In the field of image processing, domain adaptation has seen significant advancements. Peng et al. [23] propose a Disentanglement-Inspired Single-Source Domain Generalization Network (DSDGnet) for cross-scene hyperspectral image (HSI) classification, addressing spectral heterogeneity by extracting domain-invariant representations through a style transfer and progressive disentanglement approach, with experiments demonstrating a superior performance over the existing methods across multiple HSI datasets.

Yang et al. [24] introduce a novel unsupervised domain adaptation method, the contrastive domain adaptation network (CDANet), for building extractions from high-spatial-resolution imagery, leveraging adversarial and contrastive learning with a multitask generator and dual discriminators to enhance the edge detection and alignment of cross-domain pixel features, achieving a superior performance compared to the existing methods across multiple datasets. Tang et al. [25] propose a domain-adaptive noise reduction framework (DANRF) for low-dose CT denoising, combining supervised and unsupervised methods through iterative knowledge transfer, knowledge distillation, and style generalization learning to address domain gaps and improve performances on real-world data, with experiments demonstrating its effectiveness on multi-source datasets. Jecklin et al. [26] present a novel approach to overcome the domain gap between synthetic and real fluoroscopic images for the intraoperative 3D reconstruction of the spine in orthopedic surgeries, utilizing a unique paired dataset and transfer learning. By integrating style transfer and refining the X23D model, the method achieves high accuracy and a good real-time performance, offering a promising solution for enhancing surgical navigation and planning. Guo et al. [27] propose novel regularization-based frequency-domain diffusion algorithms for distributed estimation with missing input data, improving upon the existing methods by addressing the input color and complexity issues through bias elimination, power normalization, and periodic updates, with simulations demonstrating the algorithms' superior performance and theoretical validity. Moraes et al. [28] review the critical role of training data in the land cover (LC) classification of satellite imagery, identifying key research topics across 114 peer-reviewed studies. The findings are categorized into four main topics: construction of the training dataset, sample quality, sampling design, and advanced learning techniques. Subtopics include methods for the sample collection, cleaning, size, class balance, and distribution. The review provides a comprehensive synthesis of these aspects, offering insights to guide future LC mapping projects. Chen et al. [29] propose the Multi-Scale Global and Category-Attention Feature Alignment Network (MGCAN) to improve the extraction of un-collapsed buildings from post-disaster high-resolution remote sensing images by effectively aligning pre- and post-disaster features, significantly enhancing the accuracy and outperforming the existing methods. Okafor et al. [30] enhanced wheat head detection across varying domains by applying Fourier domain adaptation (FDA), adaptive alpha beta gamma correction (AABG), and random guided filter (RGF) preprocessing, achieving an improved detection accuracy (a mAP of 0.6534) compared to the baseline and addressing challenges related to domain variations in wheat images.

In the realm of image retrieval, traditional image feature descriptors often result in high-dimensional vectors. The Scale-Invariant Feature Transform (SIFT) [31], for instance, generates local feature descriptors with dimensions typically ranging from 12,800 to 128,000 per image. Even deep learning-based retrieval models often produce feature vectors exceeding 1024 dimensions. In the era of big data, this high dimensionality significantly increases storage requirements and computational costs, making it difficult to meet real-time image retrieval demands. To address these challenges, we propose a novel algorithm, the multi-scale, multi-channel Taylor adversarial domain adaptation network (MMTADAN), which integrates statistical methods and deep learning to solve the problem of domain adaptation in image processing. Our approach employs a multi-scale, multi-channel feature extraction network, detail enhancement using Taylor series-based analysis, and adversarial domain adaptation to align features across domains.

2. Related Background Knowledge

2.1. Taylor Series

If a function $(f(x_0))$ has an n -th derivative at point x_0 , then there exists a neighborhood of x_0 , and for any x within this neighborhood, the function can be expressed as follows:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + o((x - x_0)^n) \quad (1)$$

Extending the Taylor formula into an infinite series yields the Taylor series:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!} (x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n + \dots \quad (2)$$

This expansion provides a way to approximate smooth functions by polynomials around a specific point.

2.2. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) consist of two primary components: a generator and a discriminator, as shown in Figure 1. The generator takes random noise sampled from a specific distribution and attempts to generate data that resemble the true data distribution, whereas the discriminator receives both real data samples and the generator's output, attempting to distinguish between the real and generated data. The generator and discriminator iteratively improve by competing against each other in a process known as adversarial learning. This iterative process continues until the generator produces data that are indistinguishable from the real data and the discriminator cannot reliably differentiate between the two, achieving a Nash equilibrium.

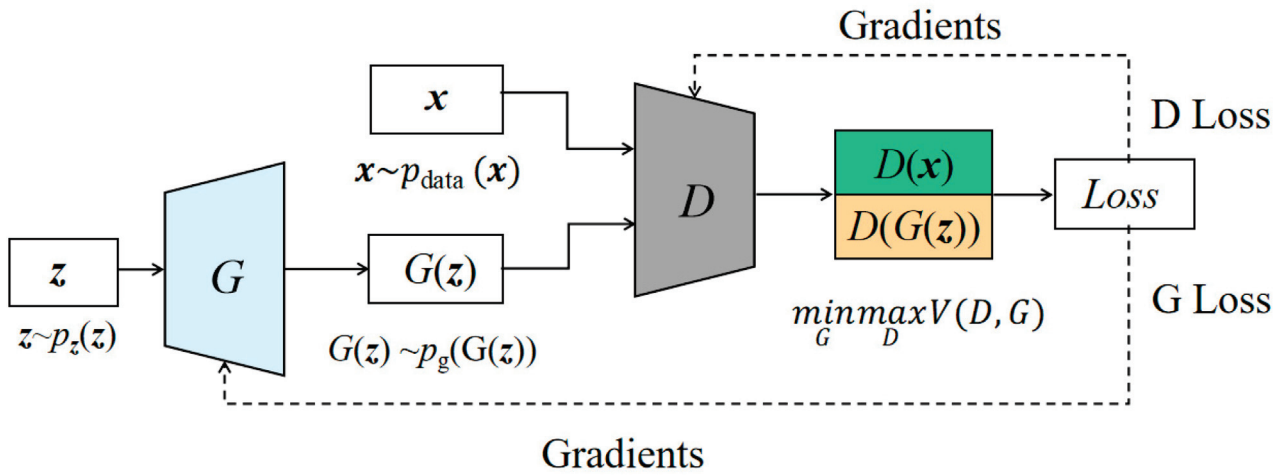


Figure 1. Structure of a Generative Adversarial Network.

Let P_{data} represent the true data distribution and P_z represent the noise distribution, which can follow any arbitrary distribution. The adversarial optimization process for GANs can be expressed as follows:

$$\min_G \max_D E_{x \sim P_{data}} \log[D(x)] + E_{z \sim P_z} \log[1 - D(G(z))] \quad (3)$$

Figure 1 illustrates the structure of a GAN, where the generator and discriminator are engaged in a minimax game, refining their performances through adversarial training.

2.3. Attention Mechanism

Convolutional Neural Networks (CNNs) possess strong data fitting capabilities; however, due to limitations imposed by optimization algorithms and computational resources, enhancing the CNN performance on large-scale datasets can be challenging. Computational power often becomes a bottleneck that restricts model improvements. Inspired by the human nervous system, the brain cannot handle overloaded visual tasks, but through the visual attention mechanism, humans focus on key objects, significantly improving their processing efficiency [32]. Attention mechanisms have a long history of research, and in 2014, Google and the DeepMind team successfully integrated attention mechanisms with Recurrent Neural Networks (RNNs), giving rise to Attention-based RNNs (AttRNNs).

These networks are widely used in sequence-to-sequence tasks such as machine translation and speech recognition. Common types of attention mechanisms include channel attention, spatial attention, and hybrid attention.

Channel attention (CA), also referred to as item-wise attention, focuses on assigning different weights to each feature map obtained from various convolutional kernels. The idea is that each kernel extracts a different feature map, capturing varying amounts of important information. By applying a weight coefficient to each feature map, the network can emphasize the feature maps that contain more relevant information.

Spatial attention (SA), also known as location-wise attention, takes the data features as input and learns a weight mask through a neural network. This mask predicts the importance of different parts of the feature map, enhancing or suppressing specific features depending on their relevance.

Hybrid attention (HA) combines both spatial and channel attention mechanisms to enhance the model's ability to focus on the most important parts of the input. A classic example of a hybrid attention module is the Convolutional Block Attention Module (CBAM) [33], which incorporates both the Channel Attention Module (CAM) and Spatial Attention Module (SAM). The CAM computes the attention weights for each channel, allowing the network to prioritize specific channel features, while the SAM calculates attention weights for each spatial location, focusing the network on important regions in the image. The outputs of the CAM and SAM are combined through element-wise multiplication to generate the final attention weights.

The CBAM enhances the network's sensitivity to both channel and spatial features, improving the classification accuracy. Furthermore, the CBAM can be seamlessly integrated into existing CNN architectures as a submodule, improving the model's expressive power and robustness. This module has been widely applied in tasks such as object detection, image classification, and semantic segmentation, where it has consistently demonstrated an excellent performance.

3. Method

3.1. Feature Fusion Network

In computer vision tasks such as object detection, image classification, and semantic segmentation, fusing multi-scale features significantly improves the performance of CNNs. Shallow feature maps, obtained with fewer convolutional operations, capture texture and geometric features of the input image but contain more redundant and low-level semantic information. In contrast, deeper feature maps, produced with more convolutions, offer better semantic representations but can lose critical details. When the performance of a network reaches a certain bottleneck, integrating multi-scale features can provide richer information, enhancing the model's robustness.

Christian et al. [34] proposed the Inception network, which combines multi-scale convolution and feature concatenation. The key idea is to use convolutional kernels of different sizes within the same layer to capture multi-scale features, followed by concatenating these features to achieve more comprehensive feature representations. Shang et al. [35] introduced the Res2Net architecture, a variant of ResNet, which splits the input features into smaller sub-feature maps processed through separate paths and then combines the outputs. This approach enables the extraction of more diverse and detailed features.

Image detail enhancement algorithms can be applied to a wide range of electronic products. In this subsection, we derive and analyze the initial model formulation for detail enhancement, proposing a rapid detail enhancement algorithm based on Taylor series analysis. This subsection introduces the Residual Extraction Network based on Taylor series analysis, referred to as the T-NET. Additionally, the T-NET employs an information entropy-based algorithm to integrate ten layers of residual features. Furthermore, we integrate the T-NET with the super-resolution algorithm CARN-M [36] to achieve both the super-resolution and detail enhancement of images. While implementing the engineering aspects, this subsection also theoretically analyzes the algorithm's rationale, error bounds,

time complexity, convergence, and information redundancy, contributing certain theoretical innovations to the field.

Given a signal ($f(x)$), according to detail enhancement theory, the signal can be decomposed into a base layer ($b(x)$) and a detail layer ($d(x)$), where $f(x) = b(x) + d(x)$. The existing algorithms aim to optimize the approximation of the $b(x)$ based on various priors and models. However, due to the complexity of image textures, obtaining a precise $b(x)$ is challenging, and what is usually obtained is an estimation, denoted as $b'(x)$. Therefore, the enhanced signal can be modeled as follows:

$$e(x) = b'(x) + \alpha \times (f(x) - b'(x)) \quad (4)$$

However, Equation (4) can be rewritten as shown in Equation (4) as $t = \alpha - 1$. For a given input image ($f(x)$), a network can be used to generate the detail layer signal ($f(x)$), where $d'(x) = \text{Net}(f(x))$ is an estimation of the true image detail layer. Additionally, the signal $f(x)$ is a weighted sum of the signals from multiple layers, represented as follows:

$$f(x) = \sum_{k=1}^n (w_k \times f_k(x)),$$

where w_k and $f_k(x)$ are the weights of each layer, and k is the output of the k -th layer of the network. Thus, the task of detail enhancement is reduced to designing the w_k and $f_k(x)$. Equation (5) further clarifies this problem, and Figure 2 illustrates the architecture of the image detail enhancement system based on Taylor series analysis (T-NET):

$$\begin{aligned} e(x) &= f(x) + (\alpha - 1) \times (f(x) - b'(x)) \\ &= f(x) + t \times d'(x) = f(x) + t \times \text{Net}(f(x)) \\ &= f(x) + t \times \sum_{k=1}^n w_k \times f_k(x) \end{aligned} \quad (5)$$

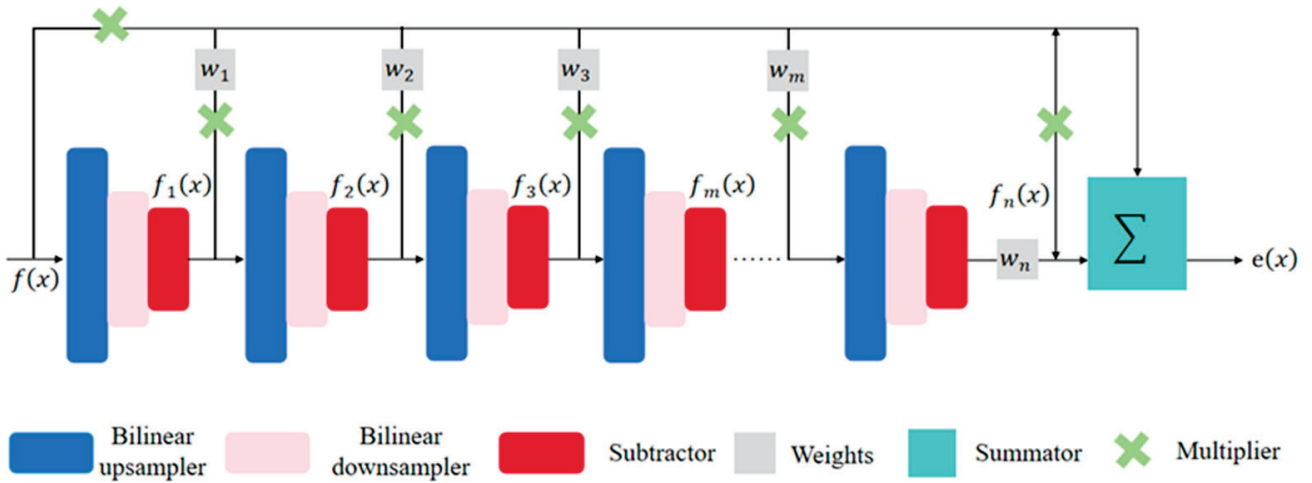


Figure 2. Image detail enhancement system based on Taylor series analysis (T-NET).

The feature extractor consists of multi-scale and multi-head attention modules, as depicted in Figure 3.

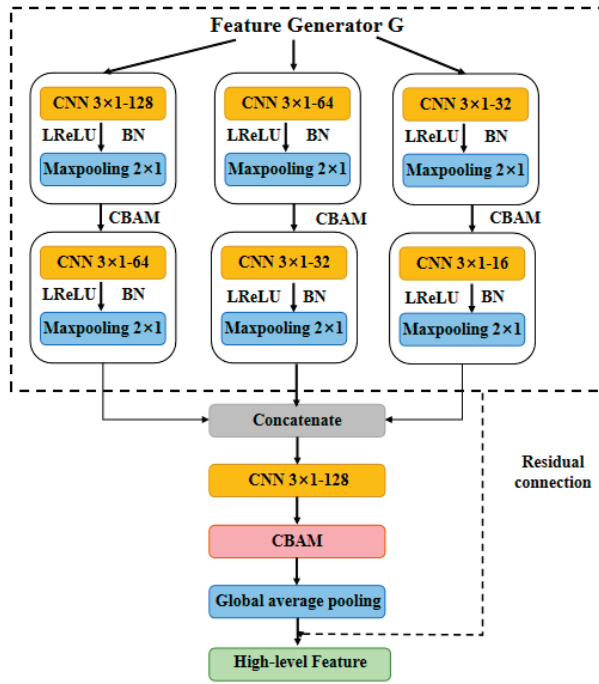


Figure 3. Schematic of the multi-head, multi-scale feature extractor.

The overall process of the MMTADAN algorithm is illustrated in Figure 4.

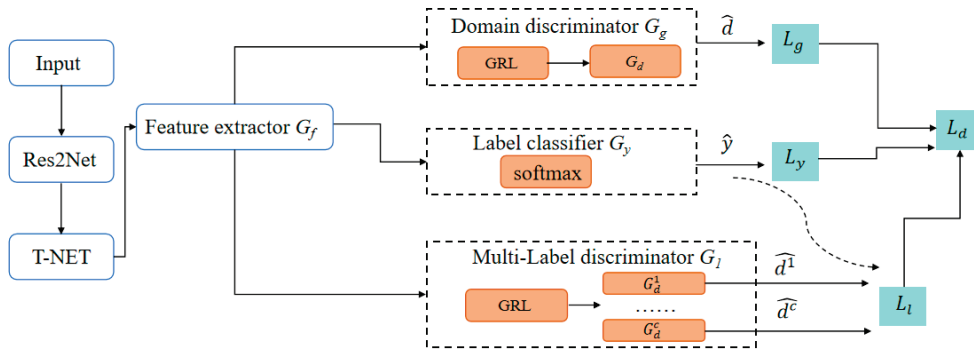


Figure 4. MMTADAN algorithm.

Specifically, the objective of the T-NET architecture is to develop a rapidly converging network that can generate detail layers of images in near real time. As previously described, it is crucial to have a clear understanding of what the signal $f(x)$ represents, as well as the mechanisms underlying the generation of the weights. Additionally, in accordance with the requirements of the detail enhancement algorithm, it is necessary to protect the signal $f(x)$ and avoid excessive variations in the $\nabla f(x)$. Thus, a network with edge-preserving capabilities must be designed.

3.1.1. Taylor Unit

The Taylor unit serves as the fundamental structure of the Taylor network (T-NET). As illustrated in Figure 5, it consists of three components: a bilinear upsampler (U), a bilinear downsampler (D), and a subtractor. In Figure 5, these components are represented by the blue, pink, and red rectangular blocks, respectively. Based on the transmission characteristics of the signal, we can derive that $f_n(x) = f_{n-1}(x) - D(U(f_{n-1}(x)))$. However, bilinear interpolation is a relatively coarse interpolation method, which allows the signal $f_n(x)$ to be treated as the residual of signal $f_{n-1}(x)$.

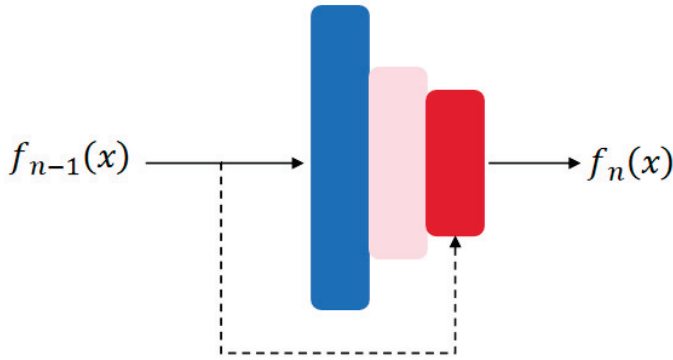


Figure 5. Schematic of the multi-head, multi-scale feature extractor.

It is evident that the residual image and the gradient image appear very similar, leading to the modeling of the signal $f_n(x)$ as $p \times \nabla f_{n-1}(x) + \gamma$. This assumption is reasonable since bilinear interpolation is a linear process, indicating that the $f_n(x)$ and signal $f_{n-1}(x)$ are linearly related. Consequently, $f_n(x) = a \times f_{n-1}(x) + b$, where a and b are known constants.

In this subsection, $x \approx x_0 \Rightarrow x - x_0$ is very small; thus, if $f_{n-1}(x_0) + \nabla f_{n-1}(x_0)(x - x_0)$ is used to estimate Equation (6), the signal $f_n(x)$ can be represented as $\rho \times f_{n-1}(x) + \gamma$:

$$\begin{aligned} f_{n-1}(x) &\Rightarrow f_n(x) \approx a(f_{n-1}(x_0) + \nabla f_{n-1}(x_0)(x - x_0)) + b \\ &\approx a\lambda \nabla f_{n-1}(x) + af_{n-1}(x_0) + b \\ &= p \times \nabla f_{n-1}(x) + \gamma \end{aligned} \quad (6)$$

In the above formula, $0 < \rho < 1$, and γ varies primarily between 50 and 150. By sampling m training pairs from the database, the following formula can be used to estimate the parameters p_k and γ_k for the k -th layer, resulting in a simple least-squares model. The existing optimization packages can be easily utilized to obtain solutions for this structure.

3.1.2. Weighted Residual Regression

Inspired by block matrix processing algorithms, weighted residuals are learned for each layer of the residual network $f_k(x)$. Initially, each weighted layer is decomposed into non-overlapping image patches of a 4×4 size. It is evident that different weights should be assigned among these patches; if a residual image patch contains more detailed information, it should be assigned a relatively larger coefficient, and vice versa. Based on this criterion, information entropy is adopted as a measure of the richness of the details within the image patches, as it is a metric related to the uncertainty of random variables. Here, if the intensity values of the pixels in an image patch can be considered as random variables, they can indeed be measured by information entropy. In this subsection, a higher information entropy indicates a richer texture within the patch, implying that this patch should receive a greater weighting coefficient. The calculation of the information entropy is as follows: $E_i = \sum_{\mu} \rho(x_{\mu}) \log_2 \frac{1}{\rho(x_{\mu})} = -\sum_{\mu} \rho(x_{\mu}) \log_2 \rho(x_{\mu})$ for a 4×4 residual image patch, where $\rho(x_{\mu})$ represents the probability value of pixel x having intensity (μ) in patch i .

The average value of these entropies is used to reflect the detail information of the group of image patches. Ultimately, these entropy values are normalized to serve as the final coefficient: $w_{ki} = \frac{1}{k!} \times w_i = \frac{1}{k!} \times \frac{E_i}{\sum_{i=1}^m E_i}$.

It can be readily proven that $\sum_{i=1}^m w'_{ki} = \frac{1}{k!}$ and $\sum_{k=1}^{10} \sum_{i=1}^m w'_{ki} \approx e - 1 \approx 2$ hold true for the m groups of patches in the 10-layer Taylor network (T-NET). To summarize the descriptions in the preceding sections, the following formula is presented, where τ denotes the coefficient for detail enhancement, existing to accommodate various types of images.

For clarity, this includes a manually adjustable parameter. Thus, the overall steps of the residual regression are presented in the following equation:

$$e(x) = f(x) + \sum_{k=1}^{10} w_k \times f_k(x) = f(x) + \tau \times \sum_{k=1}^{10} \sum_{i=1}^m \frac{1}{k!} \times \frac{E_i}{\sum_{i=1}^m E_i} \times f_k(x_i) \quad (7)$$

3.2. Algorithm Design

3.2.1. Label Classifier

The label classifier (G_y) is a core component of the model, responsible for performing the main classification task. Once the model is trained, only the feature extractor (G_f) and label classifier (G_y) are used for predicting the image classification. The label classifier consists of a fully connected neural network that takes the feature (f) corresponding to the input data (x) and outputs the prediction of x 's category, represented as a C-dimensional vector in Figure 4. Since the label classifier is a supervised classification model, it is trained using labeled data from the source domain. The loss function used for the label classifier is the cross-entropy loss (L_y), which is represented as follows:

$$L_y = -\frac{1}{n_s} \sum_{x_i \in D_s} \sum_{c=1}^C P_{x_i \in c} \log(\hat{y}) \quad (8)$$

$\hat{y} = G_y(G_f(x_i))$, where $P_{x_i \in c}$ is the probability that x_i belongs to class c . The label classifier serves two functions during the training process: (1) it performs supervised learning using labeled source domain data and target domain data with reliable pseudo-labels (represented by the solid line in Figure 4 for the calculation of L_y); (2) it generates pseudo-labels for the target domain data to guide the multi-class domain discriminator in aligning conditional distributions (represented by the dashed line in Figure 4 for the calculation of L_l).

3.2.2. Domain Discriminator

The domain discriminator is responsible for aligning the marginal distributions of the source and target domains in the proposed algorithm. The idea is inspired by the Domain Adversarial Neural Network (DANN) [37] and consists of two main components: a gradient reversal layer (GRL), shown in Figure 4, and a fully connected classifier that distinguishes whether the feature (f) comes from the source or target domain. After the data (x) are processed by the feature extractor, the resulting feature (f) first passes through the GRL, after which the fully connected classifier determines whether the data are from the source or target domain. The result is represented by $\hat{d}$ in Figure 4. The domain discriminator is a supervised learning classifier, since it only needs to classify data into two categories (the source or target domain). Both source domain data and target domain data are used in the training of the domain discriminator. The corresponding loss function is the cross-entropy loss (L_g), represented as follows:

$$L_g = \frac{1}{n_s + n_t} \sum_{x_i \in D_s \cup D_t} L_d(\hat{d}, d_i) \quad (9)$$

$\hat{d} = G_d(R_\lambda(G_f(x_i)))$, where n_s and n_t are the numbers of source and target domain samples, respectively. The GRL acts as an identity function during forward propagation but reverses the gradient during backpropagation, as expressed in Equations (10) and (11), where $\mathbf{I}$ is the identity matrix. The role of the GRL is to propagate the gradients in the opposite direction during backpropagation, ensuring that the feature extractor is updated in a direction that maximizes the adversarial training effect:

$$R_\lambda(x) = x \quad (10)$$

$$\frac{dR_\lambda}{dx} = -\lambda I \quad (11)$$

3.2.3. Multi-Class Domain Discriminator

The multi-class domain discriminator (G_l) is responsible for aligning the conditional distributions between the source and target domains. It consists of C domain discriminators (where C is the number of data categories), each with a structure similar to that of the domain discriminator, including a GRL and a fully connected domain classifier. Each domain classifier is responsible for aligning the features of one specific category, ensuring the alignment of the conditional distributions between the source and target domains. The loss function for the multi-class domain discriminator (G_l) is as follows:

$$L_l = \frac{1}{n_s + n_t} \sum_{c=1}^C \sum_{x_m \in D_s \cup D_t} \hat{y}_m^c L_d^c(\hat{d}_c, d_c) \quad (12)$$

$\hat{d}_k = G_d^c(R(G_f(x_m)))$ and n_t are the numbers of datasets in the source domain and the target domain, respectively. If x_m is from the target domain data, it is the $\hat{y}_m^c$ item of the C -dimensional vector output by the label classifier. If x_m comes from the source domain data, it is the $\hat{y}_m^c$ item of the label corresponding to the data x_m .

In this design, we drew inspiration from multi-adversarial domain adaptation (MADA) [38], wherein different domain discriminators are used for different categories to align the conditional distributions. However, our implementation differs from MADA in several ways. First, instead of using separate GRL and fully connected classifiers for each of the C domain discriminators, which can lead to a significant increase in the network parameters when the number of classes is large, we use a single fully connected network with C output nodes. Each node represents the domain classification for a specific category. Additionally, MADA requires the availability of target domain labels, but in our case, target domain labels are not available and are instead part of the model's learning objective. In MADA, pseudo-labels generated during training are used to guide the domain discriminator, but these pseudo-labels can be inaccurate, leading to error amplification. In our approach, we impose constraints on the pseudo-label acquisition. Before performing conditional distribution alignment, we first align the marginal distributions, ensuring the accuracy of the pseudo-labels. Furthermore, we introduce a confidence threshold (δ), where only target domain data with predicted class confidence values higher than the δ are included in the conditional distribution alignment.

3.3. Cost Function and Algorithm Optimization Process

In summary, the cost function of this model is as follows:

$$L(\theta_f, \theta_y, \theta_g, \theta_l) = L_y(\theta_f, \theta_y) + L_g(\theta_f, \theta_g) + L_l(\theta_f, \theta_l) \quad (13)$$

The parameters to be optimized include θ_f for the feature extractor (G_f), θ_y for the label classifier (G_y), θ_g for the domain discriminator (G_g), and θ_l for the multi-class domain discriminator (G_l). The goal of the model's training is to optimize these parameters. The training process is divided into two stages. In the first stage, the objective is to align the marginal distributions between the source and target domains, requiring the optimization of θ_y , θ_y , and θ_g , using the loss function $L_1(\theta_f, \theta_y, \theta_g) = L_y(\theta_f, \theta_y) + L_g(\theta_f, \theta_g)$. This stage's optimization is similar to the training process in the DANN. Once convergence is achieved, the feature distributions between the source and target domains become similar, and the predictions for the target domain data reach a certain level of accuracy. At this point, the prediction results from the label classifier for the target domain data can be treated as reliable pseudo-labels for calculating the loss.

The model then begins the second stage of training, with the goal of aligning the conditional distributions between the source and target domain data. In this stage, the parameters θ_f , θ_y , θ_g , and θ_l need to be optimized, using the loss function $L_2(\theta_f, \theta_y, \theta_g, \theta_l) = L_y(\theta_f, \theta_y) + L_g(\theta_f, \theta_g) + L_l(\theta_f, \theta_l)$. This training phase involves two main components: first, the alignment of the conditional distributions is achieved through the local discriminator (G_1); second, a dynamic dataset is maintained, which consists of target domain samples with high confidence levels. These high-confidence samples are directly used as supervised data for training the label classifier.

4. Experiments

All experiments involving the methods proposed in this chapter were conducted in a single-card environment using an Nvidia RTX 3080 Ti (Santa Clara, CA, USA). The development environment was set to Windows, with programming carried out using the PyTorch (V1.5.0) framework. The Stochastic Gradient Descent (SGD) optimization algorithm was employed, with a momentum coefficient of 0.9. The total number of training iterations was set to 10, with each iteration comprising 50 epochs. After each iteration, the optimal model was saved.

4.1. Dataset Description

4.1.1. Medical Image Datasets

For the image classification task, two commonly used lung field segmentation datasets are the JSRT Lung Nodule Dataset [39] and the Montgomery Tuberculosis Dataset [40,41]. The JSRT Lung Nodule Dataset contains 247 chest X-ray images with a resolution of 2048×2048 and a 12-bit grayscale, collected by the Japanese Society of Radiological Technology. Among them, 154 images contain lung nodules, and 93 do not. The Montgomery Tuberculosis Dataset, collected by the Department of Health and Human Services in Montgomery County, MD, USA, was initially developed for the creation of a computer-aided diagnosis system for tuberculosis. This dataset contains 138 chest X-ray images with a resolution of either 4020×4892 or 4892×4020 and a 12-bit grayscale, with 80 normal chest images and 58 tuberculosis images. Both datasets use the same annotation standards for the lung field masks.

4.1.2. Office–Home Image Dataset

Office–Home is a standard dataset for domain adaptation. This dataset comprises 15,500 images categorized into 65 different classes, with four subsets: (i) Art (Ar), which includes images of paintings, sketches, and artistic representations; (ii) Product (Pr), consisting of images captured without background; (iii) Real-World (Rw), containing conventional images taken with cameras; and (iv) Clipart (Cl), a collection of clipart images. The performances of the methods were evaluated on four randomly selected migration tasks out of the possible twelve: $Ar \rightarrow Pr$; $Pr \rightarrow Rw$; $Rw \rightarrow Cl$; and $Cl \rightarrow Ar$, where the symbols before and after the arrow indicate the source and target domains, respectively.

4.2. Comparison Methods and Evaluation Metrics

Several comparison methods were employed in the experiments:

- (1) Transformer [42] (Baseline): the Transformer model, trained on source domain data, was directly applied to target domain data without any domain adaptation;
- (2) DANN [37]: The DANN is a domain adaptation method that trains a neural network to learn features that are both discriminative for the source domain task and invariant to domain differences. This is achieved by using a gradient reversal layer to promote domain invariance, allowing the model to generalize to the target domain without labeled data;
- (3) LPJT [43]: LPJT is a domain adaptation method that jointly optimizes feature and sample adaptation while preserving the local sample consistency. It utilizes label prop-

agation to predict new instances and supports both homogeneous and heterogeneous domain transfers;

- (4) CDAN [44]: The CDAN is a domain adaptation method that enhances adversarial learning by conditioning the adaptation on classifier predictions. It employs multilinear conditioning to capture the cross-covariance between features and predictions, and entropy conditioning to manage uncertainty, achieving state-of-the-art results on multiple datasets.

The evaluation metrics used in the experiments included the accuracy, precision, recall, and F1 score. The accuracy measures the proportion of correct predictions out of the total number of cases. The precision (positive predictive value) measures the proportion of true-positive predictions among all positive predictions. The recall (sensitivity) measures the proportion of true-positive predictions among all actual positives. The precision and recall are often conflicting metrics, and the F1 score balances both, giving them equal weight. The accuracy and F1 score are the most important metrics. The formulas for the evaluation metrics are as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$precision = \frac{TP}{TP + FP} \quad (15)$$

$$recall = \frac{TP}{TP + FN} \quad (16)$$

$$F1score = \frac{2 \times precision \times recall}{precision + recall} \quad (17)$$

where TP, TN, FP and FN represent the numbers of true positives, true negatives, false positives, and false negatives, respectively. In this experiment, the normal class was treated as the positive class.

4.3. Experimental Results Analysis

4.3.1. Experimental Results

Two experimental tasks were established based on the medicine datasets: JSRT $\rightarrow$ Montgomery and Montgomery $\rightarrow$ JSRT. The results are summarized in Table 1. Additionally, four experimental tasks were set up using the Office–Home dataset, with the results detailed in Table 2.

Table 1. Experimental results of various methods for the medicine datasets.

Task	Method	Accuracy	Precision	Recall	F1
JSRT $\rightarrow$ Montgomery	Transformer	38.12%	45.36%	10.45%	0.16
	DANN	83.24%	85.67%	35.78%	0.49
	LPJT	79.34%	60.21%	30.65%	0.40
	CDAN	65.78%	80.45%	60.12%	0.69
	MMTADAN	92.57%	89.32%	85.76%	0.83
Montgomery $\rightarrow$ JSRT	Transformer	30.47%	55.36%	12.15%	0.19
	DANN	75.84%	82.37%	40.56%	0.55
	LPJT	78.46%	65.29%	35.12%	0.46
	CDAN	70.23%	78.41%	55.48%	0.65
	MMTADAN	91.37%	85.78%	88.12%	0.87

In the transfer tasks from JSRT to Montgomery and vice versa, the MMTADAN demonstrated an exceptional performance. Specifically, for the JSRT $\rightarrow$ Montgomery task, the MMTADAN achieved an accuracy of 92.57%, significantly surpassing the DANN (83.24%) and LPJT (79.34%). The MMTADAN also attained precision and recall rates of

89.32% and 85.76%, respectively, highlighting its superior capability in correctly identifying pulmonary nodules. The F1 score reached 0.83, reflecting a well-balanced performance in the classification task.

Table 2. Experimental results of various methods for the Office–Home dataset.

Task	Method	Accuracy	Precision	Recall	F1
Ar → Pr	Transformer	60.34%	65.48%	50.12%	0.57
	DANN	70.43%	75.32%	55.78%	0.64
	LPJT	68.29%	70.41%	52.16%	0.60
	CDAN	75.84%	78.23%	65.74%	0.71
	MMTADAN	85.67%	88.14%	80.47%	0.84
Pr → Rw	Transformer	58.23%	60.32%	48.47%	0.54
	DANN	67.56%	70.48%	55.12%	0.62
	LPJT	66.45%	68.78%	54.11%	0.61
	CDAN	72.89%	75.32%	63.56%	0.69
	MMTADAN	83.42%	85.78%	78.64%	0.82
Rw → Cl	Transformer	55.47%	57.64%	45.32%	0.51
	DANN	65.36%	68.12%	52.73%	0.60
	LPJT	63.42%	65.87%	50.63%	0.58
	CDAN	70.45%	73.34%	60.76%	0.66
	MMTADAN	82.57%	85.12%	75.47%	0.80
Cl → Ar	Transformer	60.23%	63.41%	50.76%	0.57
	DANN	68.14%	70.23%	55.12%	0.62
	LPJT	67.35%	69.42%	52.67%	0.60
	CDAN	74.42%	76.34%	64.23%	0.69
	MMTADAN	86.24%	88.42%	80.32%	0.85

For the Montgomery → JSRT task, the MMTADAN again exhibited outstanding results, with an accuracy of 91.37%, well above the other methods. Notably, in the precision (85.78%) and recall (88.12%), the MMTADAN showcased its strong adaptability in the target domain, achieving an F1 score of 0.87, which confirms its advantage in handling imbalanced datasets.

In the various transfer tasks of the Office–Home dataset, the MMTADAN consistently maintained its lead:

Ar → Pr: the MMTADAN achieved an accuracy of 85.67%, outperforming all other methods. Its precision and recall were 88.14% and 80.47%, respectively, demonstrating adaptability in diverse image processing tasks;

Pr → Rw: the MMTADAN also displayed a robust performance, with an accuracy of 83.42%, significantly higher than those of the DANN and CDAN;

Rw → Cl: the MMTADAN reached an accuracy of 82.57%, excelling in all metrics, indicating its robustness in cross-domain transfer tasks;

Cl → Ar: with an accuracy of 86.24% and a precision of 88.42%, the MMTADAN reaffirmed its exceptional capability in complex image classification tasks.

Overall, the MMTADAN consistently exhibited superior accuracy, precision, recall, and F1 scores across all the experiments. These results not only validate its effectiveness in medical image classification and domain adaptation tasks but also highlight its adaptability and flexibility when confronted with diverse datasets and tasks. In summary, the MMTADAN's outstanding performance across multiple transfer tasks, particularly in the accuracy and F1 scores, underscores its potential as an effective domain adaptation method. A comparative analysis with other methods distinctly illustrates the MMTADAN's exceptional performance in handling complex datasets, further supporting its practical application value.

4.3.2. Ablation Experiments

To further elucidate the impacts of the pseudo-label classifier and feature fusion modules within the MMTADAN on the image classification performance, we compared

the algorithm without the pseudo-label classifier (NLC) and the algorithm without the feature fusion module (NFF).

Using the Office–Home dataset, four transfer tasks were set up, with the results are displayed in Table 3.

Table 3. Ablation experimental results of various methods with the Office–Home dataset.

Task	Method	Accuracy	Precision	Recall	F1
Ar → Pr	NLC	75.43%	77.34%	66.23%	0.71
	NFF	67.54%	70.23%	55.32%	0.62
	MMTADAN	85.67%	88.14%	80.47%	0.84
Pr → Rw	NLC	69.23%	72.34%	58.43%	0.65
	NFF	62.45%	64.78%	52.34%	0.58
	MMTADAN	83.42%	85.78%	78.64%	0.82
Rw → Cl	NLC	72.47%	75.34%	65.21%	0.70
	NFF	65.14%	67.34%	53.24%	0.60
	MMTADAN	82.57%	85.12%	75.47%	0.80
Cl → Ar	NLC	58.34%	60.12%	47.32%	0.54
	NFF	76.87%	78.65%	68.34%	0.73
	MMTADAN	86.24%	88.42%	80.32%	0.85

The results clearly indicate that both the pseudo-label classifier and the feature fusion module play significant roles in enhancing the model performance.

Impact of the Pseudo-Label Classifier (NLC vs. MMTADAN)

The model’s performance was markedly inferior in all tasks without the pseudo-label classifier (NLC) compared to the full MMTADAN:

Ar → Pr: The NLC achieved an accuracy of 75.43%, which is 10 percentage points lower than the MMTADAN’s 85.67%. The F1 score also decreased by 0.13 (NLC: 0.71; MMTADAN: 0.84), underscoring the importance of the pseudo-label classifier in improving the classification performance;

Pr → Rw: The accuracy for the NLC was 69.23%, representing a decline of 14 percentage points compared to the MMTADAN’s 83.42%. The recall difference was particularly pronounced (NLC: 58.43%; MMTADAN: 78.64%), indicating that the pseudo-label classifier aids in capturing more target domain samples, thereby enhancing the model robustness;

Rw → Cl: the NLC recorded an F1 score of 0.70, significantly lower than the MMTADAN’s 0.80, further validating the critical role of the pseudo-label classifier in cross-domain adaptation;

Cl → Ar: The NLC performed the worst in this task, with an accuracy of only 58.34% and an F1 score of 0.54, both substantially below the MMTADAN’s 86.24% and 0.85. This indicates that the pseudo-label classifier substantially enhances the model performance when handling images from different domains.

Impact of the Feature Fusion Module (NFF vs. MMTADAN)

The model’s performance also significantly declined without the feature fusion module (NFF):

Ar → Pr: The NFF achieved an accuracy of just 67.54%, an 18-percentage-point drop from the MMTADAN’s 85.67%. The F1 score similarly fell from 0.84 to 0.62, highlighting the critical importance of multi-scale feature fusion in enhancing the classification accuracy and robustness;

Pr → Rw: The NFF’s accuracy was 62.45%, a 21-percentage-point reduction compared to the MMTADAN’s 83.42%. This suggests that models without feature fusion struggle to effectively capture multi-scale features from target domain samples, leading to overall performance degradation;

Rw → Cl: the NFF’s recall rate was 53.24%, markedly lower than the MMTADAN’s 75.47%, further affirming the importance of the feature fusion module in cross-domain tasks;

CI $\rightarrow$ Ar: Although the NFF performed slightly better in this task, with an accuracy of 76.87%, it still fell short of the MMTADAN's 86.24%. This indicates that the feature fusion module generally provides advantages in handling diverse image features.

The contribution of the pseudo-label classifier is evident: the NLC's performance was significantly inferior to that of the MMTADAN, demonstrating that the pseudo-label classifier effectively enhances the model classification performance and generalization capability in scenarios with limited target domain data labels. The feature fusion module also showed a notable performance drop in the NFF model compared to the complete MMTADAN, indicating that multi-scale feature fusion captures richer and more nuanced image features, thereby improving the classification performance.

In addition, to demonstrate the performance of the T-NET, three objective metrics were used to evaluate each algorithm: the Structural Similarity Index (SSIM), the Signal-Preserving Ability (SPA), and the Edge-Preserving Ability (EPA). The formula for the SSIM is defined as follows: $\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$, where μ_x and μ_y are the means of x and y , σ_x and σ_y are the variances of x and y , and σ_{xy} is the covariance between x and y . c_1 and c_2 are very small constants, both set to 0.001. Specifically, the SPA is defined as $\sum_{i=1}^k |f(x_i) - f'(x_i)|$, and the EPA is defined as $\sum_{i=1}^{k'} |\nabla(x_i) - \nabla f'(x_i)|$. In these two formulas, k and k' represent the lengths of the one-dimensional signals $f(x)$ and $\nabla f(x)$, respectively, while $f'(x)$ denotes the result of filtering the signal $f(x)$, which is the output of various detail enhancement algorithms applied to the signal $f(x)$. The comparison methods include GIF [45], WLS [46], GGIF [47], SPGIF [48], ILS [49], RGIF [47], and ZF [50]. From Table 4, it can be seen that the T-NET performed excellently in the SSIM, SPA, and EPA tests, demonstrating its superior performance.

Table 4. Comparative performance metrics of classification methods across different datasets.

Method	JSRT SSIM/SPA/EPA	Montgomery SSIM/SPA/EPA	Office-Home SSIM/SPA/EPA
GIF	0.876/3191/51	0.832/2171/107	0.814/3560/231
WLS	0.830/3659/91	0.837/3659/499	0.892/1239/763
GGIF	0.894/2673/32	0.889/3894/175	0.990/1284/46
SPGIF	0.876/6531/139	0.735/7922/646	0.961/2867/109
ILS	0.892/4150/41	0.754/5154/423	0.841/4294/23
RGIF	0.840/2579/118	0.881/5271/718	0.847/4571/225
ZF	0.792/3550/67	0.716/4891/675	0.841/3441/19
T-NET	0.985/673/15	0.993/865/30	0.983/764/16

4.3.3. Visualization

To conduct a more detailed analysis of the method's accuracy, confusion matrices were computed. Figure 6 illustrates the experimental results of the MMTADAN and the comparative methods for the JSRT dataset.

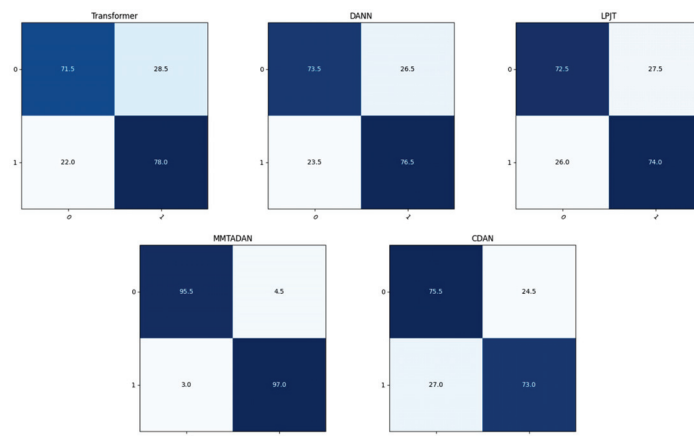


Figure 6. Experimental results of confusion matrices of various methods for the medicine dataset.

The confusion matrices in Figure 6 indicate that the Transformer method performs the worst. While the DANN, LPJT, and CDAN have improved the accuracy of health status recognition, their overall performances remain average. In contrast, the MMTADAN demonstrates effective classification.

The Receiver Operating Characteristic (ROC) curve evaluates the classifier performance by plotting the True-Positive Rate (TPR) against the False-Positive Rate (FPR). The precision–recall (PR) curve assesses the model performance by graphing the precision against the recall. The TPR and FPR are calculated using true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs) for the ROC curve, while the precision and recall are computed for the PR curve. Figure 7 presents the ROC curves for different models for the Montgomery dataset. In the ROC curves, the closer a curve is to the upper left corner, the better its performance. The ROC curve corresponding to the MMTADAN is the nearest to the upper left corner, confirming its superior effectiveness.

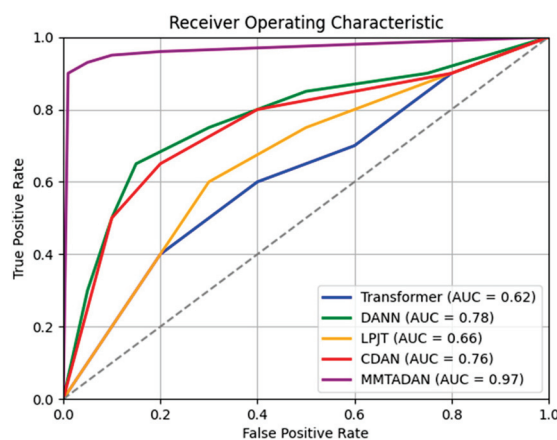


Figure 7. ROC curves of different models for the Montgomery dataset.

To validate the classification performances of the methods under consideration, a series of Monte Carlo experiments were conducted. For each experiment, the samples were randomly divided into three groups: training, validation, and testing. Care was taken to ensure that the numbers of samples for each type were approximately equal across all three groups. The training segments were used to train individual classifiers, while the validation segments were employed to train an alpha ensemble using the scores of the trained classifiers. Finally, the test segments were utilized to determine the performance. Two performance metrics were computed: the Balanced Accuracy (BAC) and F1 score (F1), which were estimated as the mean accuracy for each class and the harmonic mean of the precision and recall, respectively. The final results represent the averages of 100 Monte Carlo experiments. The experimental results are presented in Table 5.

Table 5. Average classification performances for different datasets.

Method	JSRT BAC (%) / F1 Score (%)	Montgomery BAC (%) / F1 Score (%)	Office–Home BAC (%) / F1 Score (%)
Transformer	71.05 ± 2.94 / 69.23 ± 2.78	71.23 ± 2.88 / 69.50 ± 3.12	70.89 ± 2.56 / 68.89 ± 2.65
DANN	80.01 ± 2.75 / 78.50 ± 3.01	80.50 ± 3.05 / 79.03 ± 2.96	79.84 ± 2.69 / 78.21 ± 2.72
LPJT	79.96 ± 3.10 / 77.08 ± 2.89	80.14 ± 3.20 / 77.64 ± 2.75	80.48 ± 2.93 / 77.46 ± 3.07
CDAN	80.23 ± 2.57 / 79.04 ± 2.64	80.32 ± 2.48 / 79.27 ± 2.59	80.05 ± 2.71 / 78.45 ± 2.99
MMTADAN	96.00 ± 2.55 / 95.01 ± 2.82	96.50 ± 2.62 / 95.25 ± 2.98	96.25 ± 2.53 / 94.93 ± 2.84

As shown in Table 5, the Transformer performed poorly on the Office–Home dataset (65% BAC), indicating that this method struggles with more complex or diverse datasets. While the DANN showed a good performance on the JSRT and Montgomery, its performance dropped to 75% on the Office–Home dataset. This may suggest that the DANN is

sensitive to certain types of data or that there were significant differences in the feature distribution between the training and Office–Home datasets. Similarly, the LPJT’s result for the Office–Home dataset (76.5% BAC) is lower than those for the JSRT and Montgomery datasets, highlighting its limitations in specific contexts. The CDAN also exhibited a decrease in its performance on the Office–Home dataset (77% BAC), possibly due to the complexity of the data, affecting the model’s generalization capability. Despite the MMTADAN’s excellent performance on the JSRT and Montgomery datasets (96% and 96.5% BACs, respectively), its BAC for the Office–Home dataset is 90%. This result indicates that the MMTADAN maintained a relatively stable performance across the different datasets, although it was somewhat affected on the Office–Home method, and it remains the best-performing method among all of them.

Furthermore, Table 6 displays the *p*-value calculations and 95% confidence intervals for the different methods for the Office–Home dataset. Statistical analysis reveals that the MMTADAN showed a significant advantage in the classification performance, with its BAC and F1 scores markedly higher than those of the other methods. Through rigorous statistical analysis, we confirmed the reliability of the experimental results and provide valuable references for subsequent research.

Table 6. Statistical performance comparison of different methods for the Office–Home dataset.

Method	<i>p</i> -Value (BAC)	<i>p</i> -Value (F1)	95% CI for BAC (%)	95% CI for F1 (%)
Transformer	<0.001	<0.001	(68.00, 73.78)	(66.00, 71.78)
DANN	<0.001	<0.001	(78.00, 81.68)	(77.00, 79.98)
LPJT	<0.001	<0.001	(79.00, 81.96)	(75.00, 79.92)
CDAN	<0.001	<0.001	(79.00, 81.10)	(77.00, 80.90)
MMTADAN	-	-	(96.25, 96.95)	(94.93, 95.73)

Using Big O notation, we estimated the time complexity of each method. Subsequently, after training each method, we recorded the time required for the model inference in detail. These data aided us in comparing the performances of the different models to evaluate their feasibility and advantages in practical applications. The time complexities were as follows: Transformer: $O(n \cdot d^2)$ (where *n* is the sequence length and *d* is the feature dimension); DANN: $O(n \cdot d \cdot m)$ (where *n* is the number of samples and *m* is the number of features); LPJT, CDAN, and MMTADAN: the complexity was analyzed based on specific implementations. The results are shown in Table 7.

Table 7. Time complexity and performance metrics of different methods.

Method	Time Complexity	Training Time (Seconds)	Inference Time (Seconds)
Transformer	$O(n \cdot d^2)$	150	20
DANN	$O(n \cdot d \cdot m)$	120	18
LPJT	$O(n \cdot d \cdot m)$	110	15
CDAN	$O(n \cdot d \cdot m)$	115	17
MMTADAN	$O(n \cdot d \cdot m)$	90	10

As shown in Table 7, the MMTADAN had the shortest training time at only 90 s, indicating high efficiency during the model training. In contrast, the Transformer took the longest time to train, at 150 s, likely due to its complex architecture and computational overhead from the self-attention mechanism. The MMTADAN also excelled in the inference time, requiring only 10 s, making it more advantageous for real-time applications. Other methods have inference times ranging from 15 to 20 s, indicating lower efficiencies during inference. The MMTADAN outperformed other methods in both the training and inference times, showcasing its exceptional computational efficiency and performance. Therefore, the MMTADAN is regarded as the best-performing method in this experiment.

5. Conclusions

This paper presents a novel approach for enhancing cross-domain image classification through the proposed algorithm, the MMTADAN. This method combines multi-scale feature extraction with Taylor series-based detail enhancement and adversarial domain adaptation to better align features between source and target domains. It addresses a pertinent issue in image classification, particularly the challenges associated with generalizing models across diverse datasets. The integration of multi-scale feature extraction and Taylor series enhancement is promising for improving the classification performance by preserving essential image details.

Author Contributions: Conceptualization, G.W.; Methodology, Y.G.; Software, Y.Z.; Validation, Y.G., Y.Z. and J.Z.; Formal analysis, Z.C.; Investigation, J.Z.; Resources, Z.C.; Data curation, Z.C. and J.Z.; Writing—original draft, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Major Scientific Research Instrument Development Project (62127806) and High-Technology Ship Research Program (CBG3N21-3-3).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Ding, L.; Sun, Y.; Xiong, Z. Dual-Mode Type Algorithm for Chatter Detection in Turning Considering Beat Vibration. In Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, China, 8–12 July 2019; pp. 654–659.
- Fang, Y.; Hu, J.; Shao, Q.; Qi, J. Fifth Order Trajectory Planning for Reducing Residual Vibration. In Proceedings of the 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), Toyonaka, Japan, 3–5 July 2019; pp. 999–1004.
- Gao, T.; Yang, J.; Tang, Q. A Multi-Source Domain Information Fusion Network for Rotating Machinery Fault Diagnosis under Variable Operating Conditions. *Inf. Fusion* **2024**, *106*, 102278. [CrossRef]
- Gao, T.; Yang, J.; Wang, W.; Fan, X. A Domain Feature Decoupling Network for Rotating Machinery Fault Diagnosis under Unseen Operating Conditions. *Reliab. Eng. Syst. Saf.* **2024**, *252*, 110449. [CrossRef]
- Cheng, Z.; Han, C.; Liang, J.; Wang, Q.; Zhang, X.; Liu, D. Self-Supervised Adversarial Training of Monocular Depth Estimation against Physical-World Attacks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 1–17. [CrossRef] [PubMed]
- Fang, Y.; Hu, J.; Liu, W.; Shao, Q.; Qi, J.; Peng, Y. Smooth and Time-Optimal S-Curve Trajectory Planning for Automated Robots and Machines. *Mech. Mach. Theory* **2019**, *137*, 127–153. [CrossRef]
- Liu, Y.; Sun, Y.; Xiong, Z. An Approximate Maximum Likelihood Estimator for Instantaneous Frequency Estimation of Multicomponent Nonstationary Signals. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 6501509. [CrossRef]
- Fang, Y.; Qi, J.; Hu, J.; Wang, W.; Peng, Y. An Approach for Jerk-Continuous Trajectory Generation of Robotic Manipulators with Kinematical Constraints. *Mech. Mach. Theory* **2020**, *153*, 103957. [CrossRef]
- Qin, C.; Sun, Y.; Tao, J.; Zeng, H.; Li, Y.; Liu, C. A Chatter Recognition Approach for Robotic Drilling System Based on Synchroextracting Chirplet Transform. *IEEE Sens. J.* **2023**, *23*, 27670–27683. [CrossRef]
- Sun, Y.; Liu, C.; Sun, L.; Xiong, Z.; Zhu, X. Chatter Detection With Beat Effect Based on Beat Frequency Estimation. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 2166–2174. [CrossRef]
- Salazar, A.; Vergara, L.; Safont, G. Generative Adversarial Networks and Markov Random Fields for Oversampling Very Small Training Sets. *Expert Syst. Appl.* **2021**, *163*, 113819. [CrossRef]
- Pereira, L.M.; Salazar, A.; Vergara, L. A Comparative Analysis of Early and Late Fusion for the Multimodal Two-Class Problem. *IEEE Access* **2023**, *11*, 84283–84300. [CrossRef]
- Salazar, A.; Safont, G.; Vergara, L. A New Application of Ultrasound Signal Processing for Archaeological Ceramic Classification. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3082–3086.
- Salazar, A.; Safont, G.; Vergara, L.; Vidal, E. Graph Regularization Methods in Soft Detector Fusion. *IEEE Access* **2023**, *11*, 144747–144759. [CrossRef]
- Salazar, A.; Pereira, L.M.; Vergara, L. Experimental Study on Decision Fusion Parameters Using Alpha Integration. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; pp. 53–57.

16. Pereira, L.M.; Salazar, A.; Vergara, L. Simultaneous Analysis of FMRI and EEG Biosignals: A Multimodal Fusion Approach. In Proceedings of the 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2021; pp. 1673–1677.
17. Sohail, S.S.; Himeur, Y.; Kheddar, H.; Amira, A.; Fadli, F.; Atalla, S.; Copiaco, A.; Mansoor, W. Advancing 3D Point Cloud Understanding through Deep Transfer Learning: A Comprehensive Survey. *Inf. Fusion* **2025**, *113*, 102601. [CrossRef]
18. Özince, N.; Mengüç, E.C.; Emlek, A. Sparsity-Aware Complex-Valued Least Mean Kurtosis Algorithms. *Signal Process.* **2025**, *226*, 109637. [CrossRef]
19. Lu, M.; Xing, L.; Chen, B. Measuring Generalized Divergence for Multiple Distributions with Application to Deep Clustering. *Pattern Recognit.* **2025**, *157*, 110864. [CrossRef]
20. Peng, Y.; Zhang, S.; Zhou, Z. Frequency-Domain Diffusion Adaptation over Networks with Missing Input Data. *Signal Process.* **2025**, *226*, 109661. [CrossRef]
21. Xian, Y.; Peng, Y.-X.; Sun, X.; Zheng, W.-S. Distilling Consistent Relations for Multi-Source Domain Adaptive Person Re-Identification. *Pattern Recognit.* **2025**, *157*, 110821. [CrossRef]
22. Fang, Y.; Wu, J.; Wang, Q.; Qiu, S.; Bozoki, A.; Liu, M. Source-Free Collaborative Domain Adaptation via Multi-Perspective Feature Enrichment for Functional MRI Analysis. *Pattern Recognit.* **2025**, *157*, 110912. [CrossRef] [PubMed]
23. Peng, D.; Wu, J.; Han, T.; Li, Y.; Wen, Y.; Yang, G.; Qu, L. Disentanglement-Inspired Single-Source Domain-Generalization Network for Cross-Scene Hyperspectral Image Classification. *Knowl.-Based Syst.* **2024**, *303*, 112413. [CrossRef]
24. Yang, M.; Yang, R.; Tao, S.; Zhang, X.; Wang, M. Unsupervised Domain Adaptive Building Semantic Segmentation Network by Edge-Enhanced Contrastive Learning. *Neural Netw.* **2024**, *179*, 106581. [CrossRef]
25. Tang, Y.; Lyu, T.; Jin, H.; Du, Q.; Wang, J.; Li, Y.; Li, M.; Chen, Y.; Zheng, J. Domain Adaptive Noise Reduction with Iterative Knowledge Transfer and Style Generalization Learning. *Med. Image Anal.* **2024**, *98*, 1033027. [CrossRef]
26. Jecklin, S.; Shen, Y.; Gout, A.; Suter, D.; Calvet, L.; Zingg, L.; Straub, J.; Cavalcanti, N.A.; Farshad, M.; Fürnstahl, P.; et al. Domain Adaptation Strategies for 3D Reconstruction of the Lumbar Spine Using Real Fluoroscopy Data. *Med. Image Anal.* **2024**, *98*, 103322. [CrossRef]
27. Guo, X.; Yin, J.; Yang, J. Fine Classification of Crops Based on an Inductive Transfer Learning Method with Compact Polarimetric SAR Images. *GIScience Remote Sens.* **2024**, *61*, 2319939. [CrossRef]
28. Moraes, D.; Campagnolo, M.L.; Caetano, M. Training Data in Satellite Image Classification for Land Cover Mapping: A Review. *Eur. J. Remote Sens.* **2024**, *57*, 2341414. [CrossRef]
29. Chen, M.; Wu, J.; Mao, T.; Du, R.; Zhao, B.; Lin, J.; Zhang, J. An Improved Method for Rapid Un-Collapsed Building Extraction from Post-Disaster High-Resolution Remote Sensing Imagery Based on Multi-Scale Feature Alignment. *Int. J. Digit. Earth* **2024**, *17*, 2344599. [CrossRef]
30. Okafor, S.C.; Wei, L.; Boamah, S.; Zhang, L.; Diallo, M.B. Enhanced Wheat Head Detection in Images Using Fourier Domain Adaptation and Random Guided Filter: Détection Améliorée Des Têtes de Blé Dans Les Images à l’aide de l’adaptation Du Domaine Fourier et Du Filtre Guidé Aléatoire. *Can. J. Remote Sens.* **2024**, *50*, 2367479. [CrossRef]
31. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Corfu, Greece, 20–25 September 1999; Volume 2, pp. 1150–1157.
32. Chikkerur, S.; Serre, T.; Tan, C.; Poggio, T. What and Where: A Bayesian Inference Theory of Attention. *Vis. Res.* **2010**, *50*, 2233–2247. [CrossRef]
33. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19.
34. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
35. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef]
36. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 294–310.
37. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *arXiv* **2016**, arXiv:1505.07818.
38. Pei, Z.; Cao, Z.; Long, M.; Wang, J. Multi-Adversarial Domain Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
39. van Ginneken, B.; Stegmann, M.B.; Loog, M. Segmentation of Anatomical Structures in Chest Radiographs Using Supervised Methods: A Comparative Study on a Public Database. *Med. Image Anal.* **2006**, *10*, 19–40. [CrossRef]
40. Candemir, S.; Jaeger, S.; Palaniappan, K.; Musco, J.P.; Singh, R.K.; Xue, Z.; Karargyris, A.; Antani, S.; Thoma, G.; McDonald, C.J. Lung Segmentation in Chest Radiographs Using Anatomical Atlases With Nonrigid Registration. *IEEE Trans. Med. Imaging* **2014**, *33*, 577–590. [CrossRef]

41. Jaeger, S.; Candemir, S.; Antani, S.; Wáng, Y.-X.J.; Lu, P.-X.; Thoma, G. Two Public Chest X-Ray Datasets for Computer-Aided Screening of Pulmonary Diseases. *Quant. Imaging Med. Surg.* **2014**, *4*, 475–477. [CrossRef] [PubMed]
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–8 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
43. Li, J.; Jing, M.; Lu, K.; Zhu, L.; Shen, H.T. Locality Preserving Joint Transfer for Domain Adaptation. *IEEE Trans. Image Process.* **2019**, *28*, 6103–6115. [CrossRef]
44. Long, M.; Cao, Z.; Wang, J.; Jordan, M.I. Conditional Adversarial Domain Adaptation. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 1647–1657.
45. He, K.; Sun, J.; Tang, X. Guided Image Filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1397–1409. [CrossRef] [PubMed]
46. Farberman, Z.; Fattal, R.; Lischinski, D.; Szeliski, R. Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation. *ACM Trans. Graph.* **2008**, *27*, 1–10. [CrossRef]
47. Kou, F.; Chen, W.; Wen, C.; Li, Z. Gradient Domain Guided Image Filtering. *IEEE Trans. Image Process.* **2015**, *24*, 4528–4539. [CrossRef] [PubMed]
48. Cheng, J.; Li, Z.; Gu, Z.; Fu, H.; Wong, D.W.K.; Liu, J. Structure-Preserving Guided Retinal Image Filtering and Its Application for Optic Disk Analysis. *IEEE Trans. Med. Imaging* **2018**, *37*, 2536–2546. [CrossRef]
49. Liu, W.; Zhang, P.; Huang, X.; Yang, J.; Shen, C.; Reid, I. Real-Time Image Smoothing via Iterative Least Squares. *ACM Trans. Graph.* **2020**, *39*, 1–24. [CrossRef]
50. Tao, X.; Zhou, C.; Shen, X.; Wang, J.; Jia, J. Zero-Order Reverse Filtering. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 222–230.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Reducing Safety Risks in Construction Tower Crane Operations: A Dynamic Path Planning Model

Binqing Cai ¹, Zhukai Ye ¹, Shiwei Chen ^{1,*} and Xun Liang ²

¹ School of Management, Fujian University of Technology, Fuzhou 350118, China; binq1107@163.com (B.C.); yezhukai@163.com (Z.Y.)

² School of Information Technology, Renmin University of China, Beijing 100872, China; xliang@ruc.edu.cn

* Correspondence: 19932265@fjut.edu.cn

Abstract: Tower cranes are the most used equipment in construction projects, and the path planning of tower crane operations directly affects the safety performance of construction projects. Traditional tower crane operations rely on only the driving experience and manual path planning of crane operators. Poor judgement and bad path planning may increase safety risks and even cause severe construction safety accidents. To reduce safety risks in construction tower crane operations, this research proposes a dynamic path planning model for tower crane operations based on computer vision technology and dynamic path planning algorithms. The proposed model consists of three modules: first, a path information collection module preprocessing the video data to capture relevant operational path information; second, a path safety risk evaluation module employing You Only Look Once version 8 (YOLOv8) instance segmentation to identify potential risk factors along the operational path, e.g., potential drop zones and the positions of nearby workers; and finally, a path planning module utilizing an improved Dynamic Window Approach for tower cranes (TC-DWA) to avoid risky areas and optimize the operational path for enhanced safety. A prototype based on the theoretical model was constructed and tested on actual construction projects. Through experimental scenarios, it was found that each tower crane operation poses safety risks to 3–4 workers on average, and the proposed prototype can significantly reduce the safety risks of dropped loads from tower crane operations affecting ground workers and important equipment. A comparison between the proposed model and other regular algorithms was also conducted, and the results show that compared with traditional RRT and APF algorithms, the proposed model reduces the average maximum collision times by 50. This research provides a theoretical model and a preliminary prototype to provide dynamic path planning and reduce safety risks in tower crane operations. Future research will be conducted from the aspects of multiple device monitoring and system optimization to increase the analysis speed and accuracy, as well as on human–computer interactions between tower crane operators and the path planning guidance model.

Keywords: construction; tower crane; safety risks; path planning; TC-DWA

1. Introduction

Tower cranes are among of the most used equipment in construction projects, playing a critical role in transporting materials and building components on construction sites [1]. The operations of tower cranes largely depend on the experience and manual path planning of crane operators. Poor judgment and inadequate path planning by operators as a result of human inaccurate distance calculation and restricted view on the construction sites can significantly increase safety risks, potentially leading to severe construction accidents, such as lifted objects striking already constructed parts of a building or falling on workers. Several severe crane accidents have happened globally during the past several years, e.g., in August 2023, a devastating crane collapse occurred in Thane, a city in Western India, resulting in 20 fatalities and three injuries [2]. In December 2021, a tower crane collapse in

Turin, Italy, led to three deaths and three injuries [3]. Between 2011 and 2020, the United States recorded 129 construction crane accidents [4]. In Australia, crane-related accidents in the construction sector resulted in 47 fatalities between 2003 and 2015, accounting for 8.5% of all construction industry fatalities [4,5]. In China, tower crane accidents result in an average of 1.17 fatalities per year, and the consequences of such accidents are often more severe than typical construction incidents [6]. Many fatalities involving cranes and ground workers are caused by contact with objects and equipment, particularly crane loads and parts [7]. Due to the nature of high-altitude operations, it is difficult for ground workers to consistently monitor the lifted loads above them, making the risk of falling objects a critical safety issue that needs to be addressed.

Currently, ensuring the safety of tower crane operations relies heavily on manual monitoring, which is both time-consuming and labor-intensive [8]. To address these challenges, researchers have explored various solutions. In the field of sensor-based monitoring, Zhou et al. [9] developed a digital twin-based monitoring system for port cranes that integrates multi-source heterogeneous data from various sensors, enabling real-time operation monitoring and control program simulation. Similarly, Lai et al. [10] proposed a shape-performance integrated digital twin framework that combines multiple sensors with analytical, numerical, and AI models to enable real-time structural analysis and predictive maintenance of cranes. Meanwhile, with the increasing availability and affordability of surveillance cameras on construction sites, there has been increasing interest in leveraging computer vision technology to monitor and assess operational safety [11–13]. Yang et al. [14] were the first to utilize video data from cameras mounted above cranes to identify hazardous areas using Mask Region-based Convolutional Neural Network (Mask R-CNN) and pixel coordinate distance conversion methods. Cao et al. [15] employed a convolutional neural network (CNN) to detect whether workers on construction sites were wearing safety helmets in surveillance videos, thus reducing the risk of injury. Wang et al. [16] developed a system to automatically detect and three-dimensionally (3D) locate the position of tower crane hooks using cameras. Although these studies have contributed to the identification of potential safety risks during tower crane operations, they have seldom provided solutions for mitigating those risks. For example, while operators may be informed of the presence of workers near a load, there is no guidance on how to adjust crane movements to avoid endangering those workers.

Path planning technology has been widely adopted across various crane applications, demonstrating its maturity and versatility in the industry. In offshore applications, time-optimal trajectory planning methods have been developed specifically for offshore cranes operating on non-inertial platforms. These methods incorporate multiple physical constraints to achieve safe and efficient material handling in challenging offshore environments [17–19]. In construction sites, autonomous tower cranes have implemented offline trajectory planning algorithms to enable precise payload movement along straight lines while efficiently navigating through obstacles [20,21]. To further enhance safety and efficiency in tower crane operations at construction sites, researchers have developed various intelligent algorithms for path planning optimization [22–24]. Zhu et al. [25] compared and summarized three intelligent algorithms, including the node generation-based A-star (A*) search algorithm, the sampling-based Rapidly Exploring Random Tree (RRT), and the heuristic Genetic Algorithm (GA). Du et al. [26] studied the optimization of tower crane path points based on construction crane operation standards and proposed an improved RRT Connect algorithm. By introducing heuristic information into the path search, the efficiency and quality of the algorithm were enhanced. Lin et al. [27] proposed an automated lift path planning method for tower cranes based on environmental point clouds, utilizing an optimized A* algorithm to identify executable paths. However, in real-world crane operations, the dynamic environment is complicated by the unpredictable movement of ground workers. Traditional path planning methods may not provide effective decision support to mitigate the risk of falling loads.

To address these limitations in tower crane operations, this study aims to develop an effective method for mitigating the risk of falling loads to ground workers during tower crane operations, thereby providing operational support to tower crane operators and ensuring safety. We propose a dynamic path planning model for tower crane operations that integrates computer vision and path planning algorithms. This model includes three key modules: a path information collection module, a path safety risk evaluation module, and a path planning module. It identifies on-site information necessary for the dynamic operation of tower cranes, perceives safety risks, and proposes path planning solutions. The instance segmentation capability of the You Only Look Once version 8 (YOLOv8) framework is demonstrated through a case study, and the effectiveness of the improved Dynamic Window Approach for tower cranes (TC-DWA) algorithm for path planning is validated via simulation experiments. The study also involves the construction of a safety monitoring platform for tower crane operations, which revealed a significant number of unsafe conditions on construction sites. This platform provides driving reminders and suggestions for adjusting the operation paths of tower crane operators.

2. A Dynamic Path Planning Model for Tower Crane Operations

During the lifting and lowering phases of tower crane operations, operators often engage in tasks that reduce hazards for surrounding workers. The risk of dropped loads is greatest during horizontal operations in the air, where the area of potential impact is the widest. Therefore, this study focuses on the horizontal operational process of the tower crane to develop path planning strategies. In tower crane operations, dangers arise when lifted objects pass over the heads of ground workers due to disengagement. To prevent such accidents, lifted loads can be projected onto the ground to determine whether there is a risk of collision in the horizontal direction with the workers.

The dynamic path planning model framework for tower crane operations proposed in this study is illustrated in Figure 1. First, a camera mounted above the hook monitors the tower crane operations, collecting real-time path information from close range. The path information collection module preprocesses the video data to extract relevant operational path information and generate a dataset of construction path images. Secondly, a safety risk evaluation module for tower crane operations is developed based on the YOLOv8 instance segmentation framework, which defines potential risk factors along the operational path, such as drop zones and the positions of nearby workers. The YOLOv8-seg model is employed to segment potential drop zones and identify protected objects in the video. The pixel coordinate information obtained is then converted into real-world coordinates to determine the actual state of each object, including its size and position as scene information, thereby enabling dynamic risk evaluation during crane operations. Finally, the path planning module applies an improved TC-DWA algorithm to avoid risky areas and optimize the operational path for enhanced safety.

2.1. Path Information Collection Module

The tower crane operation process involves transporting lifted objects from the starting point to the unloading point, often spanning multiple buildings during construction. Using long-range cameras to monitor tower crane operations can lead to line-of-sight obstructions [28]. To obtain more accurate data, a close-range camera is installed above the hook to capture video of the tower crane operations. In this study, a Hikvision camera is used, employing the Real-Time Streaming Protocol (RSTP) video streaming protocol to acquire real-time monitoring video and collect path information.

The collected tower crane operation videos are converted into an image dataset. Due to the complex construction site environment, some images may have suboptimal lighting conditions, necessitating data preprocessing. Blurry, duplicate, and other invalid images are filtered out, and the image sizes are standardized for input into the network. To enrich the dataset, data augmentation techniques such as translation, rotation, flipping, cropping, and brightness adjustment are applied to the construction images.

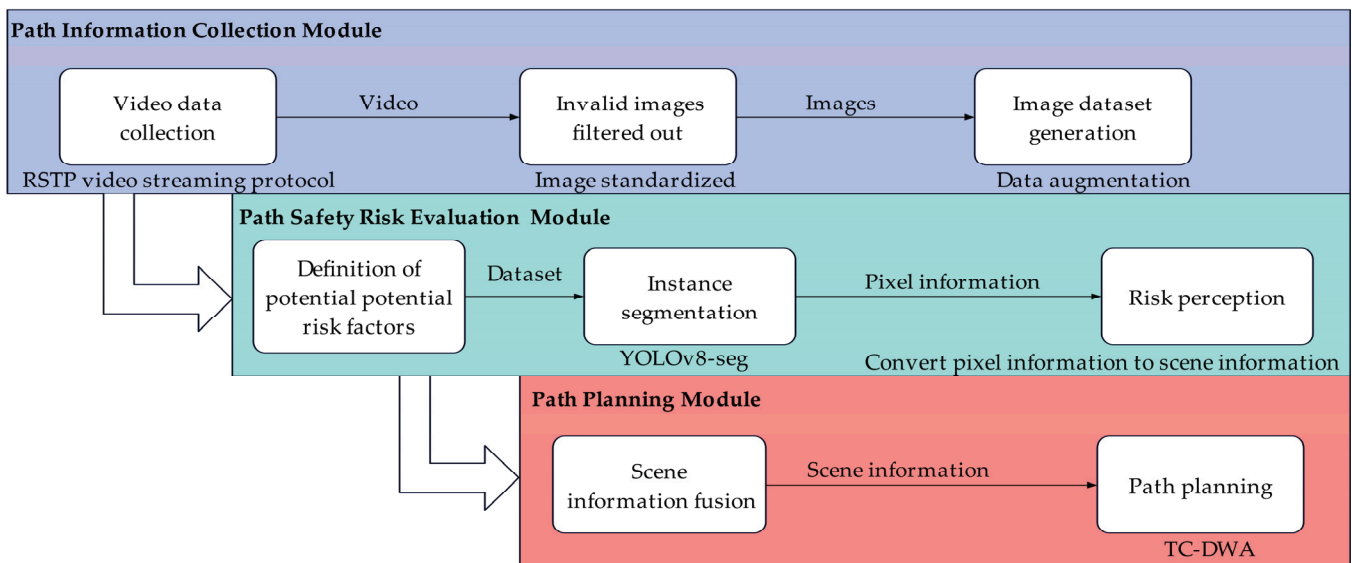


Figure 1. Dynamic path planning model framework for tower crane operations.

2.2. Path Safety Risk Evaluation Module

To estimate the risk of falling objects during tower crane operations, it is necessary to define the potential drop zones of the operations [29]. The Occupational Safety and Health Administration (OSHA) in the United States has established health and safety requirements for construction cranes in its Federal Regulations (CFR) [30]. CFR 1926.1425 defines the drop zone as “an area where materials that are reasonably foreseeable to be partially or fully suspended during an accident may fall (including, but not limited to, the area directly beneath the load being lifted)”. Based on the CFR’s definition, in this research, it is assumed that the dropped load follows a parabolic trajectory, as shown in Figure 2, and the operational hazard radius can be set around the tower crane hook to define the potential drop zone. In construction projects, workers, critical machinery, and important work areas are significant protected objects. This study focuses on mobile workers rather than static objects.

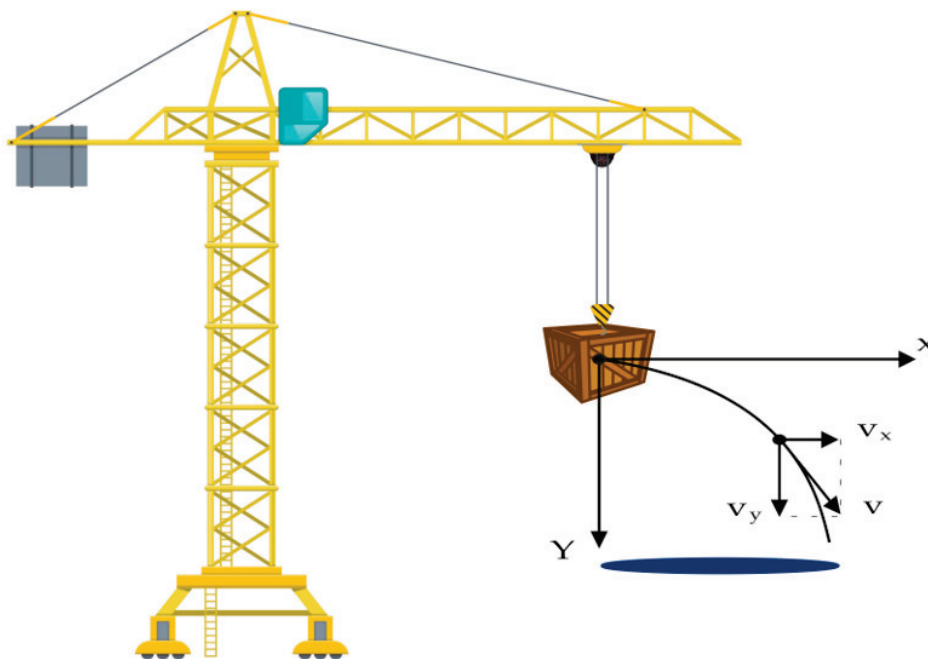


Figure 2. Potential drop zone of the load.

CNN, R-CNN, Mask R-CNN, and other models used in deep learning have shown significant effects in object detection, image segmentation, trajectory tracking, and other fields. However, two-stage region-based neural networks such as R-CNN have high computational costs in image recognition and classification. To achieve real-time operation, a cutting-edge object detector called YOLO emerged in 2016, employing a single-stage detection method that directly predicts bounding boxes and classes within a grid on the image, significantly speeding up the detection process compared to existing frameworks [31]. Therefore, YOLO has been widely applied and continually updated [32,33]. He et al. [34] used YOLOv5 for automatic identification of reflective vests, achieving an average accuracy of over 80%, meeting practical needs. Demetriou [35] examined real-time localization and classification models for construction site debris using various deep learning models, finding that the YOLOv7 model demonstrated stronger inference speed and accuracy. To further calculate the distance between cranes and workers, mere object detection is insufficient; instance segmentation is needed to provide accurate pixel-level segmentation for each object instance in the image. YOLOv8 leads in accuracy and precision over YOLOv7, with YOLOv8-seg featuring instance segmentation capabilities [36–38]. This study employs YOLOv8-seg as an advanced instance segmentation algorithm, which offers higher accuracy and faster speed compared to previous YOLO versions (YOLOv5, YOLOv6, YOLOv7). The YOLO model, once trained, can perform instance segmentation in tower crane operation videos, outputting classes, shapes, coordinates, and confidence scores. The YOLOv8-seg model mainly consists of three modules: the input module, the backbone network module, and the task head module, as illustrated in Figure 3.

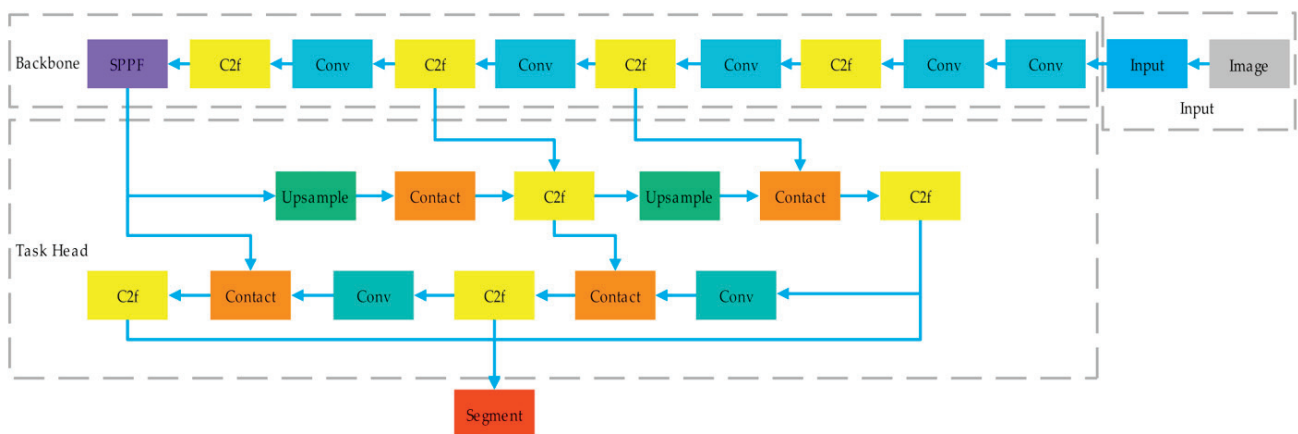


Figure 3. YOLOv8-seg framework.

1. **Input Module:** This module is used for preprocessing images to ensure they can be correctly handled by the model. Generally, the main tasks of the input module include resizing, normalizing, and padding the images. All YOLO algorithms require input images to be converted to a fixed size before being fed into the detection model for training. The standardized image size designed for this study is $640 \times 640 \times 3$.
2. **Backbone Module:** This module is built based on CNN. The convolutional layers (Conv) are used to extract and process features from the input feature map, capturing richer semantic information. CSP Bottleneck with 2 Convolutions (C2f) consists of 1×1 convolution layers and multiple Bottleneck layers, focusing on processing and merging features to extract more detailed feature information. Spatial Pyramid Pooling—Fast (SPPF) is a variant of spatial pyramid pooling used to perform pooling operations at different scales and integrate the results to extract multi-scale feature information, thereby enhancing the understanding and processing of targets of varying sizes.
3. **Task Head Module:** This module is constructed based on the Path Aggregation Network (PAN) framework, which integrates concepts from the Feature Pyramid

Network (FPN). Up-sampling and contact modules for up-sampling and connection are utilized, which enhances the effective use of multi-scale feature information in instance segmentation tasks, thereby improving detection accuracy and robustness.

Images are preprocessed to a standardized resolution of $640 \times 640 \times 3$ pixels within the input module. The preprocessed images are then fed into the backbone module, where a series of CNN operations extract features. This process involves convolutional operations, with 1×1 convolutional layers in the C2f component and Bottleneck layers working in concert to process and merge features, thereby extracting semantic information from the input images. The features extracted by the backbone are passed to the task head module for further refinement, specifically for instance segmentation. Up-sampling and contact modules are used to integrate multi-scale feature information, which is critical for accurately detecting and segmenting objects of varying sizes. After feature enhancement in the task head module, the model generates the final output. This output includes the classification of instances into predefined classes, the determination of their shapes, the specification of their bounding box coordinates, and the assignment of confidence scores for each detection.

Through instance segmentation, the potential drop zone (the operational hazard radius around the hook) and the protected objects (workers) can be represented in pixel coordinates within the plane projection. This study employs a method to determine the actual coordinates of the objects based on a simple proportional relationship between pixel coordinates and real-world coordinates [14,39,40]. The conversion between pixel coordinates and actual coordinates is described by the following Equation (1) [41]:

$$(x_{actual}, y_{actual}) = \frac{L_a}{L_p} (x_{pixel}, y_{pixel}) \quad (1)$$

where (x_{actual}, y_{actual}) represents the real coordinates, (x_{pixel}, y_{pixel}) represents the pixel coordinates, L_a is the real length of the object, and L_p is the measured length of the object in the image. In this study, the objects are projected onto a horizontal plane, and thus the Z-axis coordinate is not considered. Previous experiments [14] have shown that this method yields a positioning error of approximately 3%. As illustrated in Figure 4, based on the coordinate conversion, the distance between the two closest green grid points that surround the tower crane hook's center blue point is 0.5 m. By real-time monitoring of the potential drop zone and the environment of protected objects, the study identifies the actual distance between the drop zone and the workers during tower crane operations, thereby dynamically perceiving the existing safety risks.



Figure 4. Construction site grid points.

2.3. Path Planning Module

The Dynamic Window Approach (DWA) is a commonly used dynamic path planning algorithm in robotics [42]. The core idea of DWA is to use the current state of the robot (position, speed, etc.) and surrounding obstacles to generate a dynamic feasible region within a velocity space, known as the velocity window. By dynamically adjusting the velocity window and selecting optimal velocity commands, mobile robots can safely avoid

obstacles while moving quickly toward their targets. Due to its excellent performance, Hou et al. [43] have continuously improved DWA for dynamic path planning. Yao et al. [44] proposed a fuzzy logic improved dynamic windows approach for automatic navigation of mobile robots. Sun et al. [45] dynamically modified the weight coefficient combination of the dynamic window method to cope with the constantly changing dynamic environment and suboptimal global planning path problems that arise after local obstacle avoidance. The dynamic window method is continuously improving to adapt to more complex path planning problems [46–48]. Therefore, this study attempts to apply an improved version of DWA to tower crane operation path planning for the first time.

In solving the path planning problem for horizontal tower crane operations, the real-world environment can be simplified to a two-dimensional space. A working hazard radius is defined around the hook's center, creating a circular area, while workers in the environment are modeled as rectangular obstacles, with their top-left and bottom-right corners marked, as shown in Figure 5.

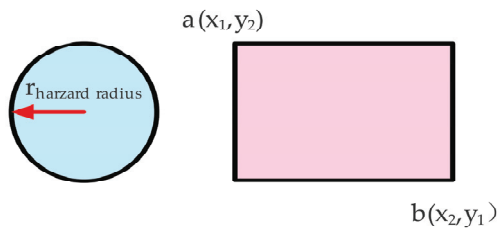


Figure 5. Modeling of potential drop zone and workers.

Assuming the tower crane's base coordinate is at the origin $O(0,0)$, the starting point of the construction task is denoted as S , and the endpoint as E . The solution space for path planning is defined as shown in Equation (2):

$$P = [0, X] \times [0, Y] \cup [-X, 0] \times [0, Y] \quad (2)$$

where $X, Y \in \mathbb{R}^+$, $P \subset \mathbb{R}^2$.

X, Y denotes the upper boundary of the construction environment. Each obstacle in the environment is represented as ob_i , with the set of obstacles denoted as $ob = \{ob_1, ob_2, \dots, ob_n\}$, where n is the total number of obstacles. $P_{obs} = ob_1 \cup ob_2 \cup \dots \cup ob_n$ represent the infeasible regions in the environment; thus, each obstacle area satisfies the following condition: $P_{obs} = \{(x, y) \mid x_{i1} \leq x \leq x_{i2}, y_{i1} \leq y \leq y_{i2}, i = 1, 2, \dots, N\}$. The feasible solution region can be expressed as follows: $P_{free} = \{(x, y) \mid 0 \leq x \leq X, 0 \leq y \leq Y, (x, y) \notin ob_i, i = 1, 2, \dots, n\}$. If there exists a feasible path point represented as $P_j(x_j, y_j)$, where $P_j \in P_{free}$, then a safe working path for the tower crane will consist of $P_T = \{P_0, P_1, P_2, \dots, P_n, P_{n+1}\}$. The tower crane operation can be divided into two motion modes: the rotation of the boom and the trolley's amplitude adjustment. Therefore, the following physical constraints must also be satisfied during the motion:

1. Rotation Radius: $r, r_{min} \leq r \leq r_{max}$, where $r_j = \sqrt{x_j^2 + y_j^2}, j = 0, 1, 2, \dots, n + 1$.
2. Velocity Constraints: $V_s, V_s = \{(v, \omega) \mid v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\}$, where v, ω are linear and angular velocities, respectively.
3. Acceleration Constraints: $V_a, V_a = \left\{ \begin{array}{l} (v, \omega) \mid v \in [v_c - \dot{v}_m \times \Delta t, v_c + \dot{v}_m \times \Delta t] \\ \omega \in [\omega_c - \dot{\omega}_m \times \Delta t, \omega_c + \dot{\omega}_m \times \Delta t] \end{array} \right\}$, where v_c, ω_c are the current linear and angular accelerations and are their respective maximum values.
4. Braking Distance Constraint: $V_{ob} = \left\{ \begin{array}{l} (v, \omega) \mid v \leq \sqrt{2 \times dist(v, \omega) \times \dot{v}} \\ \omega \leq \sqrt{2 \times dist(v, \omega) \times \dot{\omega}} \end{array} \right\}$, where $dist(v, \omega)$ is the distance from the simulated trajectory to the nearest obstacle.

The trajectory evaluation function of the TC-DWA algorithm consists of three sub-functions:

$$G(v, \omega) = \sigma[\alpha \times \text{heading}(v, \omega) + \beta \times \text{dist}(v, \omega) + \gamma \times \text{vel}(v, \omega)] \quad (3)$$

where α, β, γ are the weights corresponding to each sub-function; $\text{heading}(v, \omega)$ indicates the deviation between the endpoint of the simulated trajectory and the target point, with smaller values signifying closer trajectories; $\text{dist}(v, \omega)$ represents the distance between the simulated trajectory and protected objects, with larger values indicating safer trajectories; $\text{vel}(v, \omega)$ signifies the current motion speed, where higher values allow for quicker approaches to the endpoint. A higher value of the total trajectory evaluation function $G(v, \omega)$ signifies better overall trajectory performance. To balance the impact of various indicators on the evaluation results, normalization is applied to the three sub-evaluation functions, as shown in Equation (3) [49]. The motion modes of the tower crane are categorized into boom rotation and trolley amplitude adjustment. Each mode requires a corresponding kinematic model.

The kinematic equation for the boom rotation model is shown in Equation (4).

$$\begin{cases} x = r \times \cos(\text{yaw} + d\theta) \\ y = r \times \sin(\text{yaw} + d\theta) \\ d\theta = w \times dt \end{cases} \quad (4)$$

The kinematic equation for the trolley amplitude adjustment is shown in Equation (5).

$$\begin{cases} x = x + v \times \cos(\text{yaw}) \times dt \\ y = y + v \times \sin(\text{yaw}) \times dt \end{cases} \quad (5)$$

where yaw is the current yaw angle, and $d\theta$ is equal to the angular velocity w multiplied by the time increment dt .

The framework of the path optimization algorithm is illustrated in Figure 6. It outlines the steps involved in the TC-DWA algorithm for optimizing the tower crane's operational path. The process begins with the collection of real-time data from the crane's environment, followed by the identification of obstacles and the potential drop zones. Next, the algorithm evaluates the current trajectory based on various constraints, including speed, acceleration, and safety distances from protected objects. The trajectory evaluation function incorporates multiple criteria to ensure a smooth and safe path, adjusting as necessary based on dynamic environmental feedback. Finally, the algorithm generates an optimized path that balances efficiency and safety, allowing the tower crane to perform its tasks effectively while minimizing risks to workers and equipment.

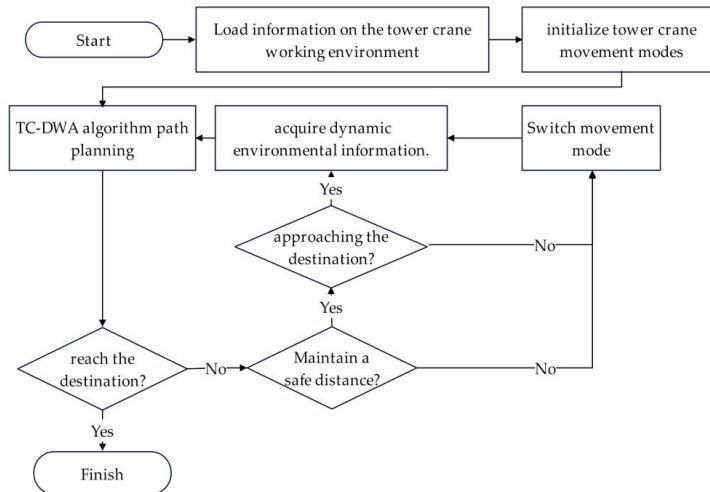


Figure 6. Framework of path optimization algorithm for tower crane operations.

This model aims to optimize the dynamic path of tower cranes during operation, ensuring the safe transportation of lifted loads while minimizing the risk of drops affecting ground workers, and striving to select the shortest path. In certain situations, when the current motion state is unable to execute the planned route, a switch in motion mode will be implemented.

In the long-distance operation scenario, the tower crane is responsible for transporting loads from the material storage area to the worker's operation area on the main construction site. During this phase, the transportation range is extensive, and the rotation angle is significant, which may necessitate traversing around critical equipment. A common scenario is illustrated in Figure 7.

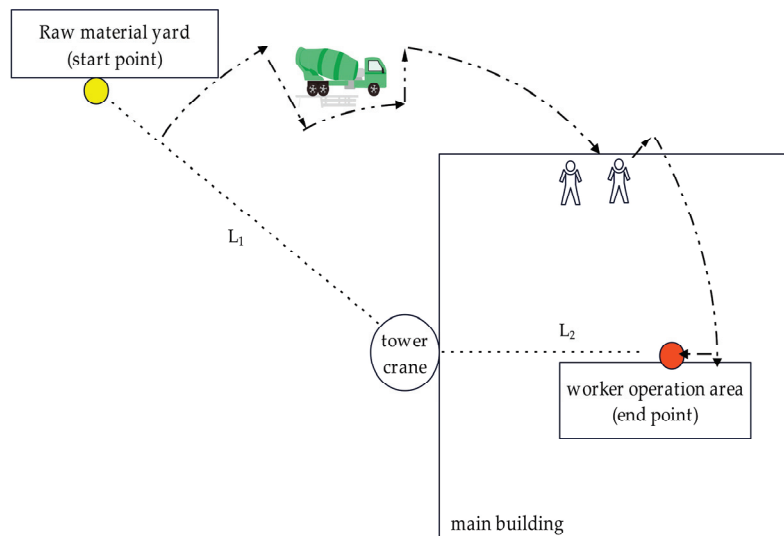


Figure 7. Long-distance operation scenario for a tower crane.

The path optimization strategy is as follows:

- Initially assess the distance L_1 from the tower crane to the material storage area and the distance L_2 to the worker's operation area.
- If $L_1 > L_2$, initialize the motion mode to linear movement of the trolley to reduce the rotation radius, adjusting the crane's rotation radius L_1 to L_2 .
- Throughout the operation, if the forward path encounters critical equipment, switch the motion mode to minimize the radius for obstacle avoidance. Upon successful avoidance, revert to the rotation mode.
- If the path ahead encounters ground workers, increase the radius for obstacle avoidance until the rotation radius aligns with the endpoint.
- Finally, perform trolley extension to safely transport the load to the destination.

In the short-distance operation scenario, the tower crane transports loads to different worker operation areas within the main construction site, characterized by a limited movement range and smaller rotation angles. A common scenario is depicted in Figure 8.

- The path optimization strategy is as follows: Begin by assessing the distance L_1 from the tower crane to the starting point of the worker's operation area and the distance L_2 to the endpoint.
- If $L_1 < L_2$, directly engage the rotation movement.
- During operation, if the forward path encounters ground workers, switch the motion mode for obstacle avoidance. Upon successful avoidance, revert to the rotation mode.
- Continue rotating until the rotation radius aligns with the endpoint, and then execute trolley extension to safely transport the load to the destination.

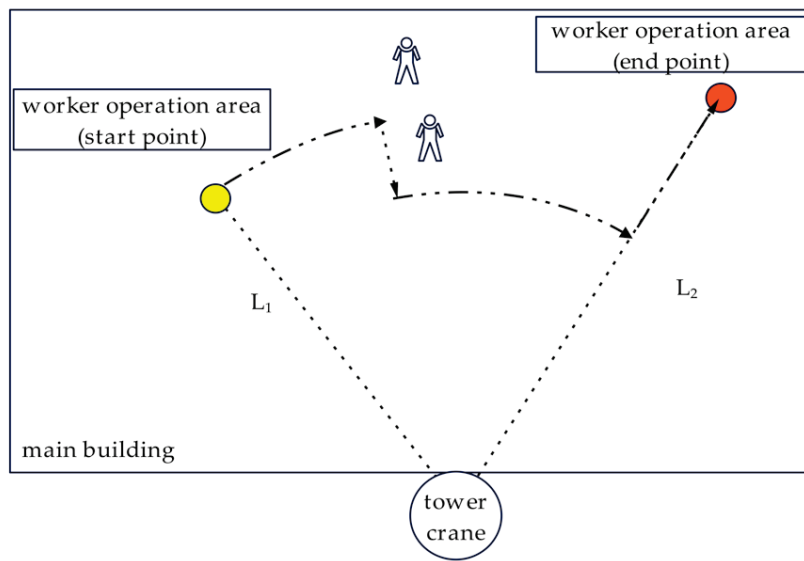


Figure 8. Short-distance operation scenario for tower crane.

In this study, an integrated tower crane construction monitoring program interface was developed based on the experimental platform parameters shown in Table 1, as depicted in Figure 9, which includes capabilities for risk evaluation and path planning. The interface displays a live video feed of the crane operation on the left side, where instance segmentation is conducted using YOLOv8. This sophisticated algorithm facilitates the identification and tracking of distinct objects within the video frame, thereby enhancing the system's capacity to monitor the crane's operation in real time. On the upper right side of the interface, a corresponding diagram is presented, reflecting the monitoring information from the video feed. This diagram offers a visual representation of the distance between the hook center and any workers present, enabling the system to detect risk situations during the operation. The system is designed to indicate a safe status during normal operations and to alert operators when workers enter a dangerous radius, thereby signaling a hazardous state. By setting the base position of the tower crane and the destination point, the system suggests the optimal direction of movement for the crane's center. The example diagrams included in the study illustrate both safe and hazardous conditions, with the crane's base positioned at the lower left and moving towards the upper left endpoint.

Table 1. Configuration parameters of the experimental platform.

Device	Configuration
Computer	Lenovo G5000 IRH8 (Lenovo Corporation, Beijing, China)
Operation system	Windows 11 (64-bit)
CPU	13th Gen Intel® Core™ i7-13700H 2.40 GHz
RAM	16 G
GPU	NVIDIA RTX4060 Laptop GPU, 8 G
GPU accelerator	Cuda 11.7
Framework	Pytorch 1.13.0
Scripting language	Python 3.8

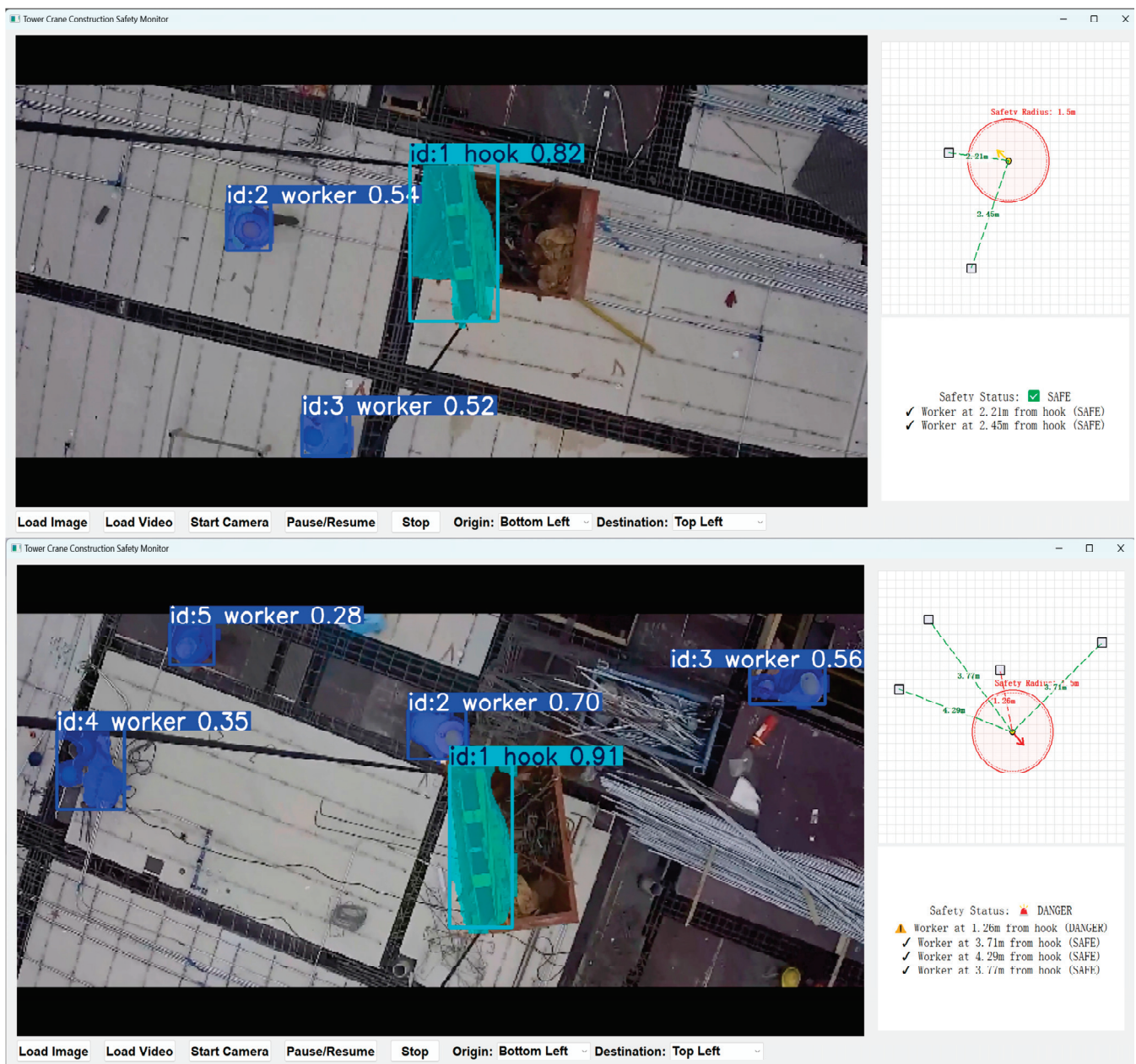


Figure 9. Tower crane construction monitoring program interface.

3. Experimental Simulation

3.1. Instance Segmentation Experiment

In this study, the image data used for model training and testing were sourced from a vertically mounted downward-facing camera on a tower crane at a construction site in Fujian, China. Multiple scenarios featuring lifting operations were selected, and 896 images were annotated using LabelMe version 5.3.1 image annotation software. The You Only Look Once version 8 Segmentation (YOLOv8-seg) network model was then trained and tested. The configuration parameters for the software and hardware platform used in this study are presented in Table 1.

Using the YOLOv8-seg network model for training, the dataset was randomly divided into a training set and a validation set in an 8:2 ratio. The batch size was set to 16, the number of epochs to 200, weight decay to 0.0005, and the learning rate to 0.01, with the initial weight model file being yolov8n-seg.pt. The model's performance was tested using the following methods. Precision, recall, and mAP@0.5 scores were considered as

evaluation metrics [50]. TP, FP, and FN refer to True Positive, False Positive, and False Negative, respectively. TP represents the number of instances that the model correctly predicts as positive samples, FP represents the number of instances where the model incorrectly predicts positive samples, and FN represents the number of instances where the model incorrectly predicts negative samples. Precision refers to the proportion of instances correctly predicted as positive by the model to all instances predicted as positive by the model. Recall rate refers to the proportion of instances correctly predicted by the model as positive samples to all instances that are actually positive samples. Average Precision (AP) is the area under the precision–recall curve, which evaluates the model’s performance across different confidence thresholds. Mean Average Precision (mAP) is the average of AP values across all classes, used to evaluate the model’s overall performance in multi-class detection tasks. mAP@0.5 represents the average precision at an Intersection over Union (IoU) threshold of 0.5. The calculation formulas are shown in Equations (6)–(9) [51], respectively.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$AP = \int_0^1 p(r) dr \quad (8)$$

$$mAP = \frac{1}{x} \sum_{i=1}^x AP_i \quad (9)$$

The experimental results are presented in Table 2, showing that the YOLOv8-seg model achieved instance segmentation precision, recall, and mAP@0.5 scores close to 1 for all objects. This indicates that the trained YOLOv8-seg model meets the instance segmentation requirements for the hooks and workers in the experimental videos, as illustrated in Figure 10.

Table 2. The instance segmentation training results.

Class	Precision	Recall	mAP@0.5
Worker	0.985	0.992	0.99
Hook	0.975	0.994	0.99

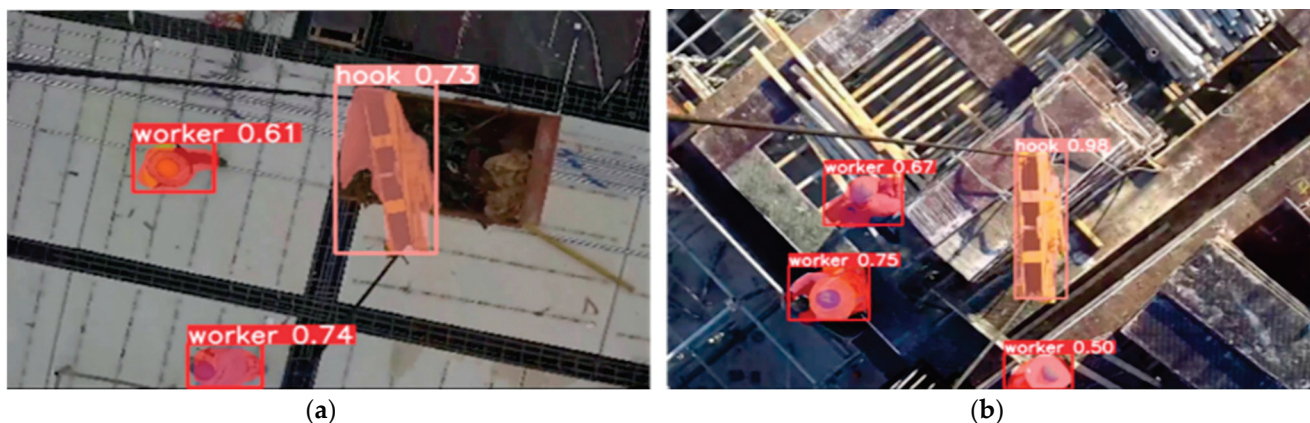


Figure 10. (a) Instance segmentation of the tower crane transportation process; (b) Instance segmentation of the tower crane unloading process.

3.2. Path Planning Simulation

The parameters for the improved Dynamic Window Approach for tower cranes (TC-DWA) experimental environment are set as detailed in Table 3, establishing common

scenarios E1 for long-distance crane operations and E2 for short-distance crane operations. The number of workers is set to be between 6 and 8, with their positions randomly assigned. Workers move slowly during the construction phase, and their speed is set to range from 0 to 1 m/s. In the long-distance scenario, important machinery remains stationary and can be simulated by using stationary workers. The TC-DWA parameters are outlined in Table 4, based on the specifications of the tower crane model at the construction site. The rotation range of the crane is set between 0 and π , with the rotation radius configured between 10 m and 55 m. Figure 11 illustrates the simulated tower crane operation scenario, where “origin” denotes the base of the tower, “obstacle” represents the moving workers, and the arrows indicate the direction of worker movement. The tower crane operation is modeled as moving from the start point to the end point with the tower body as the origin. The evaluation metrics of TC-DWA include the planned path length and the number of avoidance failures (when the planned route does not maintain a safe distance from obstacles).

Table 3. TC-DWA experimental environment parameters.

Scenario	Number of Workers	Starting Point S	End Point E
E1	6~8	(−36, 10)	(10, 30)
E2	6~8	(18, 2)	(2,30)

Table 4. TC-DWA parameter settings.

Parameter	Value
Maximum linear velocity (m/s)	1
Minimum linear velocity (m/s)	−1
Maximum angular velocity (degree/s)	7.98
Minimum angular velocity (degree/s)	−7.98
Linear acceleration (m/s ²)	0.5
Angular acceleration (degree/s ²)	15
Linear velocity resolution (m/s)	0.1
Angular acceleration resolution (rad/s)	2
Safety distance (m)	1.5
Simulation prediction time (s)	1
Unit time Δt (s)	0.5
Detection window range (m)	2
Maximum rotation radius (m)	55
Minimum rotation radius (m)	10

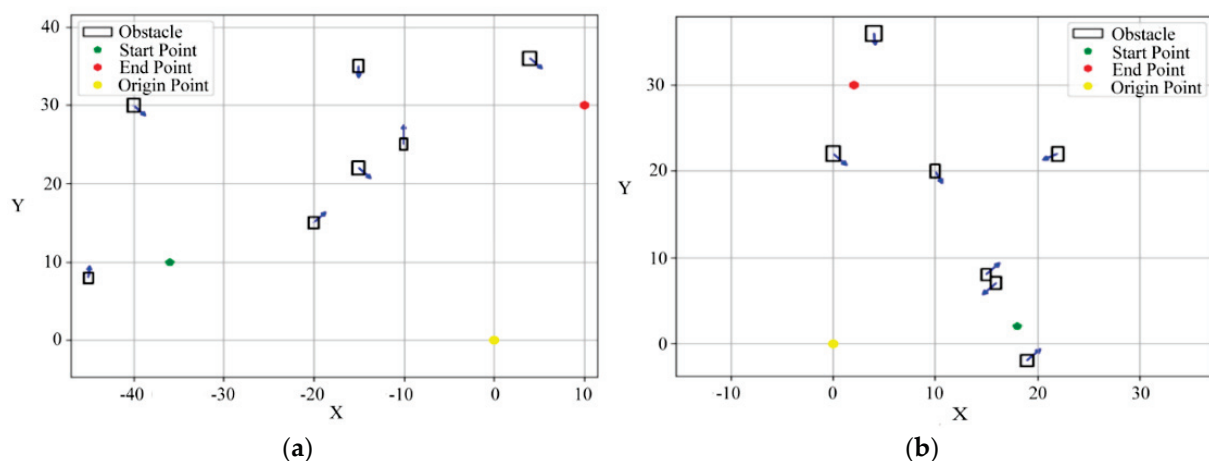


Figure 11. (a) Simulated tower crane operation scenario E1; (b) Simulated tower crane operation scenario E2.

Figure 12 shows that the proposed model can successfully provide multiple tower crane paths in various complex scenarios, validating the effectiveness of the method. In Scenario E1, based on the start and end positions, the system initiates the trolley extension mode to shorten the radius, followed by rotation. When the presence of workers along the rotation path is predicted, the model switches to trolley extension mode to increase the radius and avoid moving workers before resuming rotation. This process is repeated until the rotation path closely aligns with the endpoint, allowing the crane to shorten the radius and deliver the load to the destination. After simulating 20 random movements of workers on the job site, the crane successfully avoids workers in 20 instances of long-distance operation, with an average path length of 71.22 m. In Scenario E2, based on the start and end positions, the crane initiates the boom rotation mode. When workers are detected along the rotation path, it switches to trolley extension mode to increase the radius, avoids the moving workers, and then resumes rotation, repeating this until the load reaches the endpoint. The simulation of 20 random worker movements again results in safe avoidance and successful delivery of the load, with an average path length of 42.49 m.

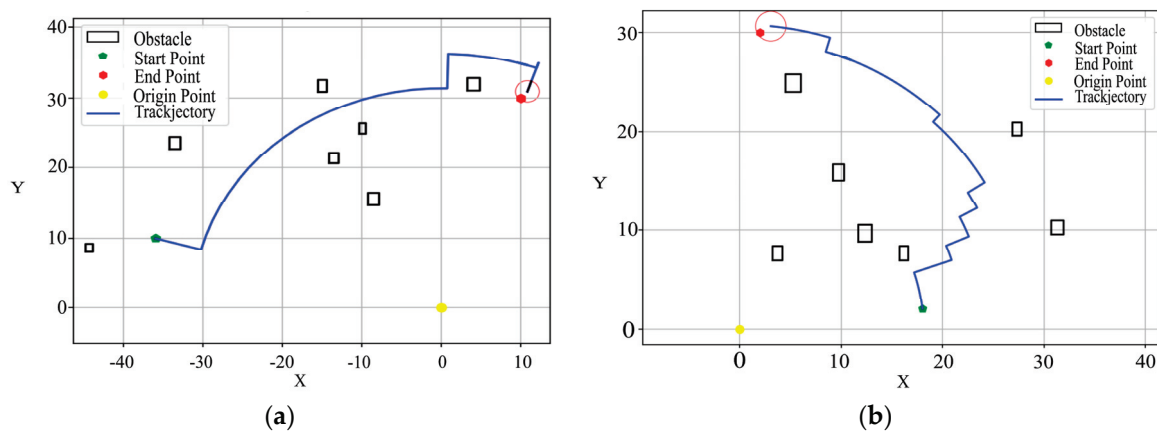


Figure 12. (a) Simulation of TC–DWA tower crane operation in scenario E1; (b) Simulation of TC–DWA tower crane operation in scenario E2.

4. Discussion

It has been demonstrated that the tower crane effectively plans paths using the TC-DWA algorithm across various scenarios, highlighting its adaptability and precision. This section aims to delve deeper into the comparative analysis of the TC-DWA algorithm against traditional path planning methods, specifically the Rapidly Exploring Random Tree (RRT) and the Artificial Potential Field (APF) methods, to establish the ultimate effectiveness of our approach in dynamic environments.

The RRT algorithm is a well-known path planning technique that operates by randomly sampling points in the configuration space from the starting point. It constructs a tree by connecting the nearest point that is free of obstacles to each sampled point until it explores the vicinity of the endpoint. While RRT is capable of exploring large spaces and can find paths in complicated environments, it often generates paths that are not optimal or smooth. This can result in operational inefficiencies, particularly in scenarios requiring precise movements, such as tower crane operations. The inherent randomness in path generation can lead to routes that are unnecessarily convoluted, as evidenced by the simulation results presented in Figure 13. The steep and complex paths generated by RRT highlight its limitations in adhering to the operational constraints of tower cranes, such as avoiding workers and maintaining safe distances from obstacles.

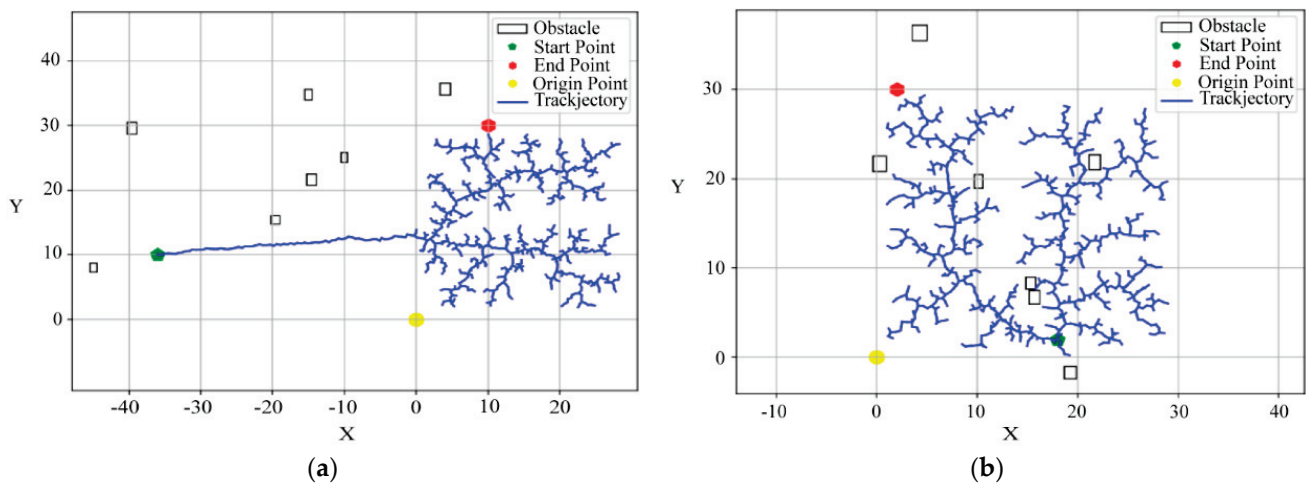


Figure 13. (a) Simulation of RRT tower crane operation in scenario E1; (b) Simulation of RRT tower crane operation in scenario E2.

On the other hand, the APF method represents a dynamic path planning approach that utilizes a potential field to guide movement. Figure 14 displays the APF algorithm's performance in similar scenarios. In this model, the goal point exerts an attractive force, while obstacles produce repulsive forces, directing the movement of the robot along the gradient of the potential field. While this method offers smoother paths, it can fail to respect the physical constraints of tower crane operations. Our simulations indicated that while APF managed to achieve shorter path lengths, it frequently encountered difficulties in avoiding moving obstacles, especially as construction conditions became more complex. The failures observed in the APF method underscore the need for a path planning solution that can dynamically adapt to the changing environment while ensuring safety and efficiency.

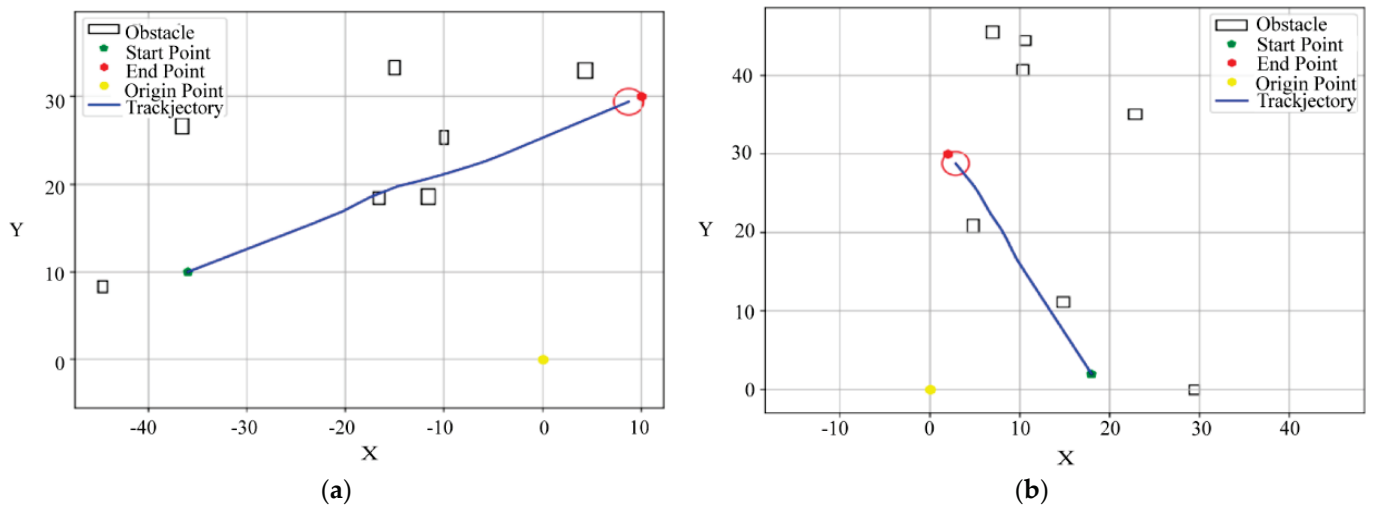


Figure 14. (a) Simulation of APF tower crane operation in scenario E1; (b) Simulation of APF tower crane operation in scenario E2.

The comparative analysis of the three methods includes planned path lengths and the number of avoidance failures (instances where the planned route did not maintain a safe distance from obstacles). All results were run 20 times, and the mean (Mean) and standard deviation (SD) are provided. The best results for each experimental scenario are highlighted in bold. From Table 5, it is evident that the traditional static path planning algorithm, RRT, struggles to handle complex and dynamic environments, failing to generate high-quality planned paths and effectively managing moving obstacles. The APF algorithm, as a

dynamic path planning method, achieved the shortest path lengths in all three scenarios; however, it encountered situations where it could not avoid moving obstacles, and its success rate in obstacle avoidance decreased with increasing complexity of construction conditions. The TC-DWA algorithm proposed in this study, while resulting in longer operational path lengths, successfully avoided obstacles in all scenarios, ensuring worker safety. Due to the variations in construction environments, the increased path lengths fluctuated between 1.36 and 1.44 times compared to the path lengths planned by the APF algorithm.

Table 5. Experimental results obtained from TC-DWA and other comparative algorithms.

Method	Scenario	Metric	Path Length (m)	Number of Obstacle Avoidance Failures
TC-DWA	E1	Mean	71.22	0
	E1	SD	16.51	0
	E2	Mean	42.49	0
	E2	SD	11.06	0
APF	E1	Mean	49.32	1.38
	E1	SD	0.23	1.52
	E2	Mean	31.19	1.62
	E2	SD	0.44	1.47
RRT	E1	Mean	67.05	33.76
	E1	SD	3.58	26.94
	E2	Mean	39.39	77.27
	E2	SD	1.70	20.26

The increase in path lengths observed with the TC-DWA method can be attributed to its focus on safety, as the algorithm actively adjusts its path in response to the detection of nearby workers. This aspect of TC-DWA enhances operational safety by prioritizing the avoidance of potential hazards, even if it results in a longer travel distance. In practical applications, the safety of workers on-site is paramount, and the trade-off between path length and safety becomes a critical consideration. The ability of TC-DWA to maintain a safety buffer around moving workers while still achieving successful load delivery positions it as a superior choice for tower crane operations in dynamic environments. Furthermore, the experiments conducted illustrate the dynamic capabilities of the TC-DWA algorithm in real-time scenarios. The algorithm's ability to adapt to unexpected changes in the environment—such as the sudden movement of workers—demonstrates its practical applicability and robustness. The simulations performed in this study underscore the importance of integrating real-time data from computer vision systems into the path planning process, as this can significantly enhance the algorithm's situational awareness and responsiveness.

Through the safety risk monitoring of 20 segments of tower crane operation videos, which range from 1 to 3 min in length and have a frame rate of 24 frames per second (FPS), we observed an average of 286.6 safety hazard warnings per complete crane operation process. Considering the video frame rate, this equates to an average of 11.9 hazardous situations per video, posing safety risks to 3–4 workers. These findings underscore the prevalence of safety hazards in crane operations and the necessity for real-time monitoring systems.

In situations where hazardous conditions are detected, the TC-DWA algorithm provides corresponding directional movement suggestions, effectively reducing the risks associated with tower crane operations. The algorithm's ability to dynamically adjust the crane's path in response to detected risks is a significant advancement in ensuring the safety of workers on construction sites. The integration of YOLOv8 for risk evaluation and the TC-DWA algorithm for path planning has demonstrated a synergistic effect in enhancing the safety of tower crane operations. The system not only detects potential

hazards but also actively participates in mitigating these risks by providing actionable guidance to operators. The study's results highlight the importance of integrating advanced computer vision and path planning algorithms into tower crane monitoring systems. The combination of real-time video analysis and dynamic path planning offers a new solution to improve safety standards in construction environments.

5. Conclusions

This study introduces a dynamic path planning model for tower crane operations that leverages computer vision technology and advanced path planning algorithms. By utilizing a camera strategically positioned above the crane's hook, real-time path information during operational activities is collected. This innovative approach enhances the accuracy of the crane's movements and integrates a safety risk evaluation module using the YOLOv8 instance segmentation framework. This model effectively identifies potential drop zones for lifted objects and the relative positions of nearby workers, enabling a dynamic awareness of risks during crane operations.

The proposed path planning module, grounded in an improved TC-DWA, autonomously generates paths that prioritize obstacle avoidance. Through comprehensive simulation experiments, effective path planning are successfully demonstrated in both long-distance and short-distance operation scenarios. The results of our evaluations highlighted the algorithm's robust performance, confirming its ability to adapt to varying operational conditions.

Comparative analyses with traditional path planning algorithms further underscored the superiority of the TC-DWA method. It facilitates efficient route planning and significantly enhances safety measures by effectively avoiding obstacles, thereby reducing the risk of accidents involving ground workers. This dual focus on efficiency and safety positions our method as a valuable contribution to modern construction practices.

As the construction industry increasingly emphasizes the need for improved safety and operational efficiency, dynamic path planning for tower crane operations emerges as an essential area for research and development. Our findings advocate for the integration of real-time data and advanced algorithms into crane operations, which can lead to better decision-making and reduced operator workload.

There are several limitations in the current study that warrant further investigation. Although the TC-DWA algorithm provides effective path planning suggestions, the successful implementation of these suggestions heavily relies on the crane operator's response time and ability to execute the recommended actions promptly. During crane operations, operators may face challenges in processing and responding to the perceived information in real-time, especially when multiple hazardous situations occur simultaneously. Additionally, the accuracy of vision-based monitoring systems is inherently limited by the capabilities of surveillance cameras alone. Factors such as varying lighting conditions, occlusions, and complex construction site environments may affect the system's detection accuracy. Future research should focus on developing more automated control mechanisms and integrating multiple types of sensing devices to build a more robust and reliable monitoring system. This multi-modal approach could provide more comprehensive and accurate safety monitoring while reducing the reliance on single-source visual data.

Future work can be conducted to develop a more comprehensive monitoring system that integrates multiple sensing devices to enhance detection accuracy. By combining the proposed model and various data sources such as cameras, LiDAR, and IoT sensors, a more reliable and robust safety monitoring and path planning system can be built. Furthermore, exploring how the proposed model could be integrated with site-wide safety systems, such as collision avoidance or real-time hazard detection, could also enhance its applicability in diverse construction environments. Future studies could also focus on the human–computer interaction aspect, investigating how operators of cranes and other heavy machinery interact with these decision-support systems. This research could help inform the design of more intuitive interfaces, improving usability and increasing adoption across the construction industry. Understanding how operators respond to system-generated

suggestions or decisions, and how these interactions affect decision-making efficiency and safety performance, will be crucial for refining the system's effectiveness in practice. Finally, long-term studies are needed to assess the actual impact of deploying such models in real-world construction sites. Evaluating key metrics such as accident reduction rates, worker injury statistics, and overall operational safety before and after the implementation of the model will provide a roadmap for validating its long-term effectiveness and refining its design.

Author Contributions: Methodology, Z.Y. and B.C.; Software, Z.Y.; Validation, S.C. and B.C.; Resources, X.L.; Writing—original draft, Z.Y.; Writing—review and editing, B.C., S.C. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Department of Finance of Fujian Province, grant number GY-S22001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Data were obtained from Fujian Provincial Erjian Construction Group Co., Ltd., and are available from the corresponding author with the permission of Fujian Provincial Erjian Construction Group Co., Ltd.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hussein, M.; Zayed, T. Crane operations and planning in modular integrated construction: Mixed review of literature. *Autom. Constr.* **2021**, *122*, 103466. [CrossRef]
2. China News. Crane Collapse in Western India Kills 20 and Injures 3. [WWW Document]. 2023. Available online: <https://www.chinanews.com.cn/gj/2023/08-02/10054165.shtml> (accessed on 2 November 2024).
3. China News. A Crane Collapse Accident Occurred in Turin, Italy, Resulting in 3 Deaths and 3 Injuries. [WWW Document]. 2021. Available online: <https://www.chinanews.com.cn/gj/2021/12-20/9633902.shtml> (accessed on 2 November 2024).
4. Chen, J.; Chi, H.-L.; Du, Q.; Wu, P. Investigation of Operational Concerns of Construction Crane Operators: An Approach Integrating Factor Clustering and Prioritization. *J. Manag. Eng.* **2022**, *38*, 04022020. [CrossRef]
5. Hu, S.; Fang, Y.; Moehler, R. Estimating and visualizing the exposure to tower crane operation hazards on construction sites. *Saf. Sci.* **2023**, *160*, 106044. [CrossRef]
6. Zhang, X.; Zhang, W.; Jiang, L.; Zhao, T. Identification of Critical Causes of Tower-Crane Accidents through System Thinking and Case Analysis. *J. Constr. Eng. Manag.* **2020**, *146*, 04020071. [CrossRef]
7. Cheng, T.; Teizer, J. Modeling Tower Crane Operator Visibility to Minimize the Risk of Limited Situational Awareness. *J. Comput. Civ. Eng.* **2014**, *28*, 04014004. [CrossRef]
8. Yu, Y.; Guo, H.; Ding, Q.; Li, H.; Skitmore, M. An experimental study of real-time identification of construction workers' unsafe behaviors. *Autom. Constr.* **2017**, *82*, 193–206. [CrossRef]
9. Zhou, Y.; Fu, Z.; Zhang, J.; Li, W.; Gao, C. A Digital Twin-Based Operation Status Monitoring System for Port Cranes. *Sensors* **2022**, *22*, 3216. [CrossRef]
10. Lai, X.; Wang, S.; Guo, Z.; Zhang, C.; Sun, W.; Song, X. Designing a Shape-Performance Integrated Digital Twin Based on Multiple Models and Dynamic Data: A Boom Crane Example. *J. Mech. Des.* **2021**, *143*, 071703. [CrossRef]
11. Jang, Y.; Jeong, I.; Younesi Heravi, M.; Sarkar, S.; Shin, H.; Ahn, Y. Multi-Camera-Based Human Activity Recognition for Human-Robot Collaboration in Construction. *Sensors* **2023**, *23*, 6997. [CrossRef]
12. Kang, K.-S.; Cho, Y.-W.; Jin, K.-H.; Kim, Y.-B.; Ryu, H.-G. Application of one-stage instance segmentation with weather conditions in surveillance cameras at construction sites. *Autom. Constr.* **2022**, *133*, 104034. [CrossRef]
13. Kamooni, A.M.; Gostar, A.K.; Tennakoon, R.; Bab-Hadiashar, A.; Accadia, D.; Thorpe, J.; Hoseinnezhad, R. Random Finite Set-Based Anomaly Detection for Safety Monitoring in Construction Sites. *IEEE Access* **2019**, *7*, 105710–105720. [CrossRef]
14. Yang, Z.; Yuan, Y.B.; Zhang, M.Y.; Zhao, X.F.; Zhang, Y.; Tian, B.Q. Safety Distance Identification for Crane Drivers Based on Mask R-CNN. *Sensors* **2019**, *19*, 2789. [CrossRef] [PubMed]
15. Cao, W.G.; Zhang, J.N.; Cai, C.X.; Chen, Q.; Zhao, Y.; Lou, Y.M.; Jiang, W.; Gui, G. CNN-Based Intelligent Safety Surveillance in Green IoT Applications. *China Commun.* **2021**, *18*, 108–119. [CrossRef]
16. Wang, J.; Zhang, Q.; Yang, B.; Zhang, B. Vision-Based Automated Recognition and 3D Localization Framework for Tower Cranes Using Far-Field Cameras. *Sensors* **2023**, *23*, 4851. [CrossRef]
17. Liu, Z.; Chu, Y.; Li, G.; Hildre, H.P.; Zhang, H. A co-simulation approach to onboard support of marine operation: A Palfinger crane path planning case. *Simulation* **2024**, *100*, 919–930. [CrossRef]

18. Chen, H.; Zhang, R.; Liu, W.; Chen, H. A time optimal trajectory planning method for offshore cranes with ship roll motions. *J. Frankl. Inst.* **2022**, *359*, 6099–6122. [CrossRef]
19. Lu, B.; Lin, J.; Fang, Y.; Hao, Y.; Cao, H. Online trajectory planning for three-dimensional offshore boom cranes. *Autom. Constr.* **2022**, *140*, 104372. [CrossRef]
20. Burkhardt, M.; Gienger, A.; Sawodny, O. Optimization-Based Multipoint Trajectory Planning Along Straight Lines for Tower Cranes. *IEEE Trans. Control Syst. Technol.* **2024**, *32*, 290–297. [CrossRef]
21. Li, X.; Chi, H.-I.; Wu, P.; Shen, G.Q. Smart work packaging-enabled constraint-free path re-planning for tower crane in prefabricated products assembly process. *Adv. Eng. Inform.* **2020**, *43*, 101008. [CrossRef]
22. Yu, L.; Huang, M.M.; Jiang, S.; Wang, C.; Wu, M. Unmanned aircraft path planning for construction safety inspections. *Autom. Constr.* **2023**, *154*, 105005. [CrossRef]
23. Yao, M.; Feng, X.; Li, P.; Li, Y.; Peng, Z.; Lu, Z. Object-level complete coverage path planning for excavators in earthwork construction. *Sci. Rep.* **2023**, *13*, 12818. [CrossRef] [PubMed]
24. Zhou, Z.; Abdi, E.; Chea, C.P.; Bai, Y. Global path planning for autonomous construction vehicles in building construction: A comparative study with a focus on vehicle kinematic characteristics. *J. Build. Eng.* **2024**, *93*, 109837. [CrossRef]
25. Zhu, A.; Zhang, Z.; Pan, W. Technologies, levels and directions of crane-lift automation in construction. *Autom. Constr.* **2023**, *153*, 104960. [CrossRef]
26. Du, X.; Yu, X.-Z.; Yun, X.-Y. Scenario modeling and operation path planning of autonomous driving tower cranes. *J. Phys. Conf. Ser.* **2022**, *2365*, 012020. [CrossRef]
27. Lin, X.; Han, Y.; Guo, H.; Luo, Z.; Guo, Z. Lift path planning for tower cranes based on environmental point clouds. *Autom. Constr.* **2023**, *155*, 105046. [CrossRef]
28. Assadzadeh, A.; Arashpour, M.; Bab-Hadiashar, A.; Ngo, T.; Li, H. Automatic far-field camera calibration for construction scene analysis. *Comput. Aided Civ. Infrastruct. Eng.* **2021**, *36*, 1073–1090. [CrossRef]
29. Chian, E.Y.T.; Goh, Y.M.; Tian, J.; Guo, B.H.W. Dynamic identification of crane load fall zone: A computer vision approach. *Saf. Sci.* **2022**, *156*, 105904. [CrossRef]
30. Occupational Safety and Health Administration. [WWW Document]. 2010. Available online: <https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926.1401> (accessed on 27 September 2024).
31. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
32. Wu, X.; He, Z.; Liu, S.; Yu, Z. Visual Modeling Method for Ship Piping Network Programs in Engine Simulators. *Appl. Sci.* **2024**, *14*, 9194. [CrossRef]
33. Zhao, S.; Kang, F.; He, L.; Li, J.; Si, Y.; Xu, Y. Intelligent Structural Health Monitoring and Noncontact Measurement Method of Small Reservoir Dams Using UAV Photogrammetry and Anomaly Detection. *Appl. Sci.* **2024**, *14*, 9156. [CrossRef]
34. He, X.; Ma, P.; Chen, Y.; Liu, Y. An Automatic Reflective Clothing Detection Algorithm Based on YOLOv5 for Work Type Recognition. In Proceedings of the 2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS), Chengdu, China, 3–5 August 2022; pp. 396–401. [CrossRef]
35. Demetriou, D.; Mavromatidis, P.; Robert, P.M.; Papadopoulos, H.; Petrou, M.F.; Nicolaides, D. Real-time construction demolition waste detection using state-of-the-art deep learning methods; single-stage vs two-stage detectors. *Waste Manag.* **2023**, *167*, 194–203. [CrossRef]
36. Wu, L.; Chen, L.; Li, J.; Shi, J.; Wan, J. SNW YOLOv8: Improving the YOLOv8 network for real-time monitoring of lump coal. *Meas. Sci. Technol.* **2024**, *35*, 105406. [CrossRef]
37. Ni, Y.; Huo, J.; Hou, Y.; Wang, J.; Guo, P. Detection of Underground Dangerous Area Based on Improving YOLOv8. *Electronics* **2024**, *13*, 623. [CrossRef]
38. Li, S.; Huang, H.; Meng, X.; Wang, M.; Li, Y.; Xie, L. A Glove-Wearing Detection Algorithm Based on Improved YOLOv8. *Sensors* **2023**, *23*, 9906. [CrossRef] [PubMed]
39. Kim, D.; Lee, S.; Kamat, V.R. Proximity Prediction of Mobile Objects to Prevent Contact-Driven Accidents in Co-Robotic Construction. *J. Comput. Civ. Eng.* **2020**, *34*, 04020022. [CrossRef]
40. Son, H.; Seong, H.; Choi, H.; Kim, C. Real-Time Vision-Based Warning System for Prevention of Collisions between Workers and Heavy Equipment. *J. Comput. Civ. Eng.* **2019**, *33*, 04019029. [CrossRef]
41. Zhang, M.; Ge, S. Vision and Trajectory-Based Dynamic Collision Prewarning Mechanism for Tower Cranes. *J. Constr. Eng. Manag.* **2022**, *148*, 04022057. [CrossRef]
42. Kobayashi, M.; Motoi, N. Local Path Planning: Dynamic Window Approach With Virtual Manipulators Considering Dynamic Obstacles. *IEEE Access* **2022**, *10*, 17018–17029. [CrossRef]
43. Hou, J.T.; Jiang, W.S.; Luo, Z.; Yang, L.; Hu, X.F.; Guo, B. Dynamic Path Planning for Mobile Robots by Integrating Improved Sparrow Search Algorithm and Dynamic Window Approach. *Actuators* **2024**, *13*, 24. [CrossRef]
44. Yao, M.; Deng, H.; Feng, X.; Li, P.; Li, Y.; Liu, H. Improved dynamic windows approach based on energy consumption management and fuzzy logic control for local path planning of mobile robots. *Comput. Ind. Eng.* **2024**, *187*, 109767. [CrossRef]
45. Sun, Y.; Wang, W.; Xu, M.; Huang, L.; Shi, K.; Zou, C.; Chen, B. Local Path Planning for Mobile Robots Based on Fuzzy Dynamic Window Algorithm. *Sensors* **2023**, *23*, 8260. [CrossRef]

46. Chang, L.; Shan, L.; Jiang, C.; Dai, Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robot.* **2021**, *45*, 51–76. [CrossRef]
47. Yang, Z.; Li, N.; Zhang, Y.; Li, J. Mobile Robot Path Planning Based on Improved Particle Swarm Optimization and Improved Dynamic Window Approach. *J. Robot.* **2023**, *2023*, 6619841. [CrossRef]
48. Liu, A.; Liu, C.; Li, L.; Wang, R.; Lu, Z. An improved fuzzy-controlled local path planning algorithm based on dynamic window approach. *J. Field Robot.* **2024**, 1–25. [CrossRef]
49. Yang, D.; Su, C.; Wu, H.; Xu, X.; Zhao, X. Construction of Novel Self-Adaptive Dynamic Window Approach Combined With Fuzzy Neural Network in Complex Dynamic Environments. *IEEE Access* **2022**, *10*, 104375–104383. [CrossRef]
50. You, C.; Kong, H. Improved Steel Surface Defect Detection Algorithm Based on YOLOv8. *IEEE Access* **2024**, *12*, 99570–99577. [CrossRef]
51. Liu, Q.; Huang, W.; Duan, X.; Wei, J.; Hu, T.; Yu, J.; Huang, J. DSW-YOLOv8n: A New Underwater Target Detection Algorithm Based on Improved YOLOv8n. *Electronics* **2023**, *12*, 3892. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Residual Vibration Reduction in Flexible Systems Based on Trapezoidal Velocity Profiles

Xining Cui ¹, Yi Fang ^{2,*} and Chaochen Gu ²¹ School of Computer Science, Liaocheng University, Liaocheng 252059, China; cuixining@lcu.edu.cn² School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

* Correspondence: apocalypse@sjtu.edu.cn

Abstract: Industrial parts are increasingly being designed to be more lightweight in modern manufacturing for energy saving and material cost reduction. However, the high-speed motion of flexible systems tends to excite severe residual vibrations that result in positioning accuracy degradation and loss of productivity. This study proposes a closed-form trajectory optimization method for vibration suppression based on trapezoidal velocity profiles, which are most widely used in industrial robots and machines. First, the formulation and minimum time solution under actuator limits of the motion profile are defined. Then, the relationship between the trajectory parameters and the vibration response is investigated. It is shown that residual vibration can be eliminated by properly tuning the acceleration/deceleration switching times according to the natural frequency. Based on the derived vibration suppression conditions, a tuning procedure for time parameters compliant with actuator limits is established to generate fast and precise movement. A main advantage of the proposed method is easy implementation for general machines without requiring extra computational resources or modification to the control system. The effectiveness and practicality of the proposed approach are verified through experiments conducted on a robot. The experimental results show that the optimized trajectory achieves superior residual vibration reduction performance.

Keywords: trajectory optimization; residual vibration; motion profile; motion control; trapezoidal velocity profile

1. Introduction

The advent of modern manufacturing innovations has spurred increased research interest in robotic automation [1,2]. Industrial applications now employ robot manipulators for various tasks [3]. Achieving precise object manipulation requires minimizing residual vibrations caused by resonance between the robot and the object upon reaching the target [4]. These vibrations can compromise task quality, increase production cost, and extend completion times, as settling periods are necessary post-movement.

Recent studies have highlighted the significance of vibration analysis in robotic systems. Borys et al. [5] demonstrated structural vibration phenomena at the robot flange and mounted tools during spot welding applications using triaxial accelerometers and high-speed camera measurements. The dynamic performance of robotic manipulators has been extensively investigated through modal analysis, as evidenced by Mishra and Singh [6] in their investigation of flexibility effects on two-link planar manipulators. Furthermore, Do

et al. [7] developed an analytical framework for the modal analysis of flexible-joint robotic manipulators with rigid links, based on linearized dynamic modeling.

Although vibration reduction can be achieved through hardware design optimization or introducing damping components [8], changing the system structure for each task in flexible production environments is impractical. Motion control offers an alternative approach to vibration reduction without altering mechanical design [9,10], which typically employs feedback or feedforward methods [11]. While feedback control measures and compensates for positioning errors, it necessitates additional sensors and modifications to the actuator control scheme. In contrast, feedforward control influences the vibration behavior by manipulating the robot's driving motion profile. This feedforward approach does not require any revision of the actuator control scheme, as profiles are imposed during trajectory generation. Established methods include creating smooth motion profiles [12,13] and implementing preshaping techniques [14,15].

Given the same maximum velocity, maximum acceleration, and displacement, the trapezoidal velocity profile (TVP) provides minimum motion duration compared to other profiles [16]. This profile offers a good balance between simplicity and performance for many applications. It is widely used in robots and machine tools due to its easy implementation. A main drawback is its discontinuous acceleration profile, which may cause undesirable vibrations for flexible systems. Previous research has explored various smooth velocity profiles to reduce robot vibrations, including polynomial and S-curve profiles [17,18]. Boryga et al. [19] planned smooth trajectories using the asymmetric higher-degree polynomial profiles under kinematic constraints, guaranteeing continuity of displacement, velocity, acceleration, and jerk. Ekrem and Aksoy [20] carried out collision-free trajectory planning using the fifth-order polynomial to ensure continuous position, speed, and acceleration between the starting and target points. Abe [21] developed an optimal trajectory planning technique to achieve the minimum vibration in two-link rigid-flexible manipulators undergoing large deformations based on a cubic spline function and the particle swarm optimization algorithm, taking into account the effects of geometric nonlinearities. Cui et al. [22] utilized quintic polynomials to design joint trajectory functions and optimized the trajectory with particle swarm optimization for suppressing residual vibration while satisfying the acceleration limits of flexible manipulators. However, these approaches require additional computation due to increased polynomial coefficients. They also demand larger acceleration and torque compared to trapezoidal profiles for equivalent displacements.

S-curve motion profiles offer another approach to ensure motion smoothness [23]. By minimizing or bounding jerk, researchers have demonstrated vibration reduction. However, determining appropriate trajectory parameters for robot actuators remains challenging. Wu et al. proposed a trajectory planning particle swarm optimization algorithm for optimizing the joint angles or paths of robot movements [24]. Fang et al. [25] investigated the systematic design of high-order S-curve trajectories adapted to vibratory behavior under derivative bounds for high-speed operations. Kudara et al. [26] presented an angular velocity trajectory generation scheme based on a two-stage S-curve profile to control the load sway for a rotary crane boom in a minimum motion time while considering the crane constraints.

In addition, some innovative motion laws based on radial basis, trigonometric, and elliptic functions were also investigated in recent studies [27–29]. Bruzzone and Stretti [30] applied an elliptic jerk profile to the Cartesian space position control of serial robots. Esau et al. [31] defined an acceleration reference profile based on a Gaussian function to ensure a continuous jerk contour for industrial machines, which reduced detrimental vibration effects and improved the machine stability. A general framework for robust vibration

mitigation under model parametric uncertainties with three motion profile primitives was introduced in [32] for underactuated mechanical systems. However, commercial industrial robots and machines often feature closed architecture controllers, preventing users from integrating custom profile algorithms or preshaping methods [33,34]. For safety reasons, motion parameter adjustments are restricted to skilled personnel using the provided trajectory planning method. Most commercial robots employ trapezoidal velocity profiles for trajectory planning [35]. For instance, major robot manufacturers allow operators to input maximum velocity and acceleration values to configure trapezoidal velocity profiles in their industrial robots. Recently, some studies attempted to optimize the vibration level by trial and error [36–38], suggesting that trapezoidal profiles can achieve good performance by properly tuning the time parameters [39,40]. Nevertheless, a systematic approach for designing trapezoidal trajectories with actuator limits and vibration suppression constraints is still lacking.

This study proposes an analytical approach for designing trapezoidal velocity profiles under actuator limits to reduce residual vibration in flexible motion systems. The method focuses on determining the optimal acceleration/deceleration time by analyzing the correlation between the vibration response characteristics and the natural frequency at which resonance may occur during high-speed movements, such that the trajectory duration and vibration can be minimized. Unlike previous trajectory planning methods that demand extensive computational resources, the proposed approach leverages a simple trapezoidal profile, thus avoiding numerical complexity, and does not require additional modification to the control architecture. It offers a practical solution for engineers and designers to enable vibration suppression in robotic systems, potentially improving precision and efficiency in various industrial applications.

The paper is structured as follows: Section 2 gives the formulation of the trapezoidal velocity profile and the time-optimal solution. Section 3 presents a theoretical analysis for vibration behavior and a tuning strategy for determining the optimal time parameters to minimize residual vibration with actuator limits based on the modal frequency. Then, parameter sensitivity evaluation is reported in Section 4 to show the influence of variations in natural frequency and damping ratio on the trajectories with different vibration suppression conditions. Section 5 details the experiments conducted on two industrial robots to validate the proposed method. Finally, Section 6 gives the conclusions.

2. Formulation of Trapezoidal Velocity Profile

The trapezoidal velocity profile, which is also referred to as the linear segment with parabolic blend (LSPB) trajectory, is characterized by three distinct phases: acceleration, cruise, and deceleration. The velocity curve over time resembles a trapezoid, as shown in Figure 1, where t_0 , t_1 , t_2 , and t_3 indicate the switching times of each phase. In the acceleration phase $[t_0, t_1]$, the system starts from rest and accelerates at a constant rate, such that the velocity increases linearly over time until it reaches the maximum velocity. In the cruise phase $[t_1, t_2]$, the system moves at a constant maximum velocity. In the deceleration phase $[t_2, t_3]$, the system decelerates at the same rate as the acceleration phase until the system comes to a stop. The acceleration profile features a bang-off-bang type, which is described by the following piecewise function:

$$\ddot{q}(t) = \begin{cases} a, & t_0 \leq t \leq t_1 \\ 0, & \text{else} \\ -a, & t_2 \leq t \leq t_3 \end{cases} \quad (1)$$

where a is the peak acceleration. Let T_a and T_v denote the time for the acceleration/deceleration phase and the time for the cruise phase, respectively. The total motion duration is, thus, $T_{tot} = 2T_a + T_v$. Let q_0 and q_1 denote the initial and the final position, respectively. The position profile $q(t)$ and velocity profile $\dot{q}(t)$ can be calculated through direct integration of Equation (1).

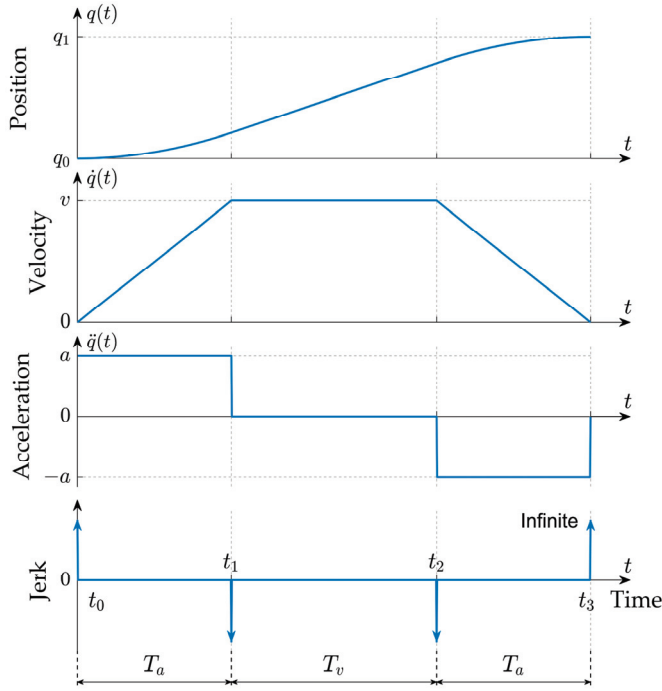


Figure 1. Standard trapezoidal velocity profile.

It can be derived that the peak velocity v and the terminal position are:

$$v = aT_a \quad (2)$$

$$q(t_3) = q_0 + aT_a(T_a + T_v) \quad (3)$$

It is necessary to determine a , T_a and T_v such that the trajectory can achieve the target displacement while respecting the actuator limits of maximum velocity $V_{\max}$ and maximum acceleration $A_{\max}$. The constraint conditions imposed by the target displacement and actuator limits are expressed as follows:

$$h = aT_a(T_a + T_v), \quad a \leq A_{\max}, \quad v \leq V_{\max} \quad (4)$$

where $h = q_1 - q_0$ denotes the total displacement. The time optimal solution is obtained by pushing the kinematic values to their physical limits. The peak acceleration can always attain its limit value, i.e., $a = A_{\max}$. In contrast, if the target displacement is too short, the maximum velocity may not be attained. In this case, the whole profile consists of only the acceleration phase and the deceleration phase with $h = A_{\max}T_a^2$; otherwise, the cruise phase is required for achieving the desired displacement, $V_{\max} = A_{\max}T_a$.

Without loss of generality, we assume that the target displacement and peak acceleration are positive values. The acceleration/deceleration time that satisfies all the constraints is calculated as follows:

$$T_a = \min \left\{ \frac{V_{\max}}{A_{\max}}, \sqrt{\frac{h}{A_{\max}}} \right\} \quad (5)$$

Two cases are distinguished here: if $T_a = V_{\max}/A_{\max}$, the trajectory type belongs to the standard one, so the time of the cruise phase is calculated as follows:

$$T_v = \frac{h - A_{\max} T_a^2}{V_{\max}} \quad (6)$$

Otherwise, if $T_a = \sqrt{h/A_{\max}}$, $T_v = 0$. The cruise phase disappears, and the trapezoidal velocity profile becomes a triangular one, as shown in Figure 2.

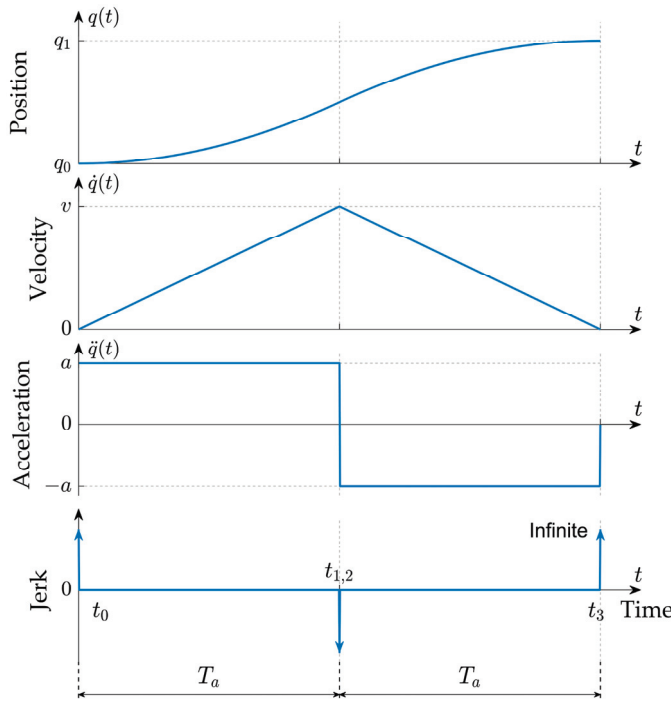


Figure 2. Triangular velocity profile.

The above solution minimizes the motion duration without violating the actuator limits. However, undesirable residual vibrations can occur at the end of the point-to-point motion due to the presence of infinite jerks. Motion-induced residual vibration negatively affects motion precision, particularly in flexible systems. It also generates unwanted noise and reduces the overall system lifespan. Therefore, it is necessary to design optimal trajectory parameters for residual vibration suppression.

3. Optimization of Parameters for Vibration Suppression

3.1. Analysis of Vibration Suppression Condition

To analyze the amplitude of the motion-induced vibration, a standard second-order system, which is defined by the transfer function shown in Equation (7) [41], is considered in this study. The system's vibration response $y(t)$ to a stepwise change in acceleration can be expressed as the sum of individual step responses [42], as shown in Equation (8).

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (7)$$

$$y(t) = \sum_{i=1}^4 y_i(t) = \frac{-a}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_0)} \sin(\omega_d(t-t_0) + \phi) u(t-t_0) + \frac{a}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_1)} \sin(\omega_d(t-t_1) + \phi) u(t-t_1) \\ + \frac{a}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_2)} \sin(\omega_d(t-t_2) + \phi) u(t-t_2) - \frac{a}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n(t-t_3)} \sin(\omega_d(t-t_3) + \phi) u(t-t_3) \quad (8)$$

where $\phi = \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}$, ω_n and ζ are the undamped natural frequency and the damping ratio of the system, respectively. $u(t)$ is the unit step function and $\omega_d = \omega_n \sqrt{1-\zeta^2}$ represents the damped natural frequency.

Using trigonometric angle difference formulas and auxiliary angle formulas, the residual vibration response can be written as follows from Equation (8):

$$y(t) = \frac{-e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sqrt{\left(\sum_{i=0}^3 A_i e^{\zeta\omega_n t_i} \cos(\omega_d t_i) \right)^2 + \left(\sum_{i=0}^3 A_i e^{\zeta\omega_n t_i} \sin(\omega_d t_i) \right)^2} \sin(\omega_d t - \varphi + \phi) \quad (9)$$

where $A_0 = a$, $A_1 = -a$, $A_2 = -a$, $A_3 = a$, and $\varphi = \tan^{-1} \frac{\sum_{i=0}^3 A_i e^{\zeta\omega_n t_i} \sin(\omega_d t_i)}{\sum_{i=0}^3 A_i e^{\zeta\omega_n t_i} \cos(\omega_d t_i)}$.

Thus, the amplitude of residual vibration at motion completion time can be obtained as follows:

$$Amp = \frac{e^{-\zeta\omega_n t_3}}{\sqrt{1-\zeta^2}} \sqrt{\left(\sum_{i=0}^3 A_i e^{\zeta\omega_n t_i} \cos(\omega_d t_i) \right)^2 + \left(\sum_{i=0}^3 A_i e^{\zeta\omega_n t_i} \sin(\omega_d t_i) \right)^2} \quad (10)$$

For lightly damped systems, ζ is nearly equal to zero, and the above equation can be simplified as:

$$Amp = \left| \sum_{i=0}^3 A_i e^{j\omega_n t_i} \right| = a \left| \prod_{i=1}^2 (1 - e^{j\omega_n t_{2i-1}}) \right| = 4a \sin\left(\frac{\omega_n t_1}{2}\right) \sin\left(\frac{\omega_n t_2}{2}\right) \quad (11)$$

It is shown that the amplitude of the motion-induced residual vibration can be represented as a function of the peak acceleration a , time period t_1 , time period t_2 , and natural frequency ω_n . This implies that the time parameters can be suitably designed based on the natural frequency to optimize the vibration amplitude. Note that the vibration suppression conditions can be derived by setting the vibration amplitude to zero:

$$Amp = 0 \Rightarrow \begin{cases} \sin\left(\frac{\omega_n t_1}{2}\right) = 0 \\ \sin\left(\frac{\omega_n t_2}{2}\right) = 0 \end{cases} \quad (12)$$

Therefore, the residual vibration becomes zero whenever one of the following conditions is satisfied:

$$C1: T_a = t_1 = k_1 T_n, \quad k_1 \in \mathbb{N}^+, \quad (13)$$

$$C2: T_a + T_v = t_2 = k_2 T_n, \quad k_2 \in \mathbb{N}^+, \quad (14)$$

where $T_n = 2\pi/\omega_n$ is the natural period of vibration. This conclusion extends the vibration suppression condition derived in [39]. The two conditions are distinguished in the vibration cancellation mechanism. If condition C1 is satisfied, the vibrations caused by the jerks at the start and end times of the acceleration/deceleration phase cancel each other out, i.e., the

first impulse response is compensated by the second one, and the third impulse response is compensated by the fourth one, leading to zero vibration, both in the cruise phase and after motion completion (see Figure 3). If condition C2 is satisfied, the vibrations caused by the jerks during the acceleration phase are canceled out by those during the deceleration phase, i.e., the first impulse response is compensated by the third one, and the second impulse response is compensated by the fourth one, leading to zero vibration only after motion completion (see Figure 4).

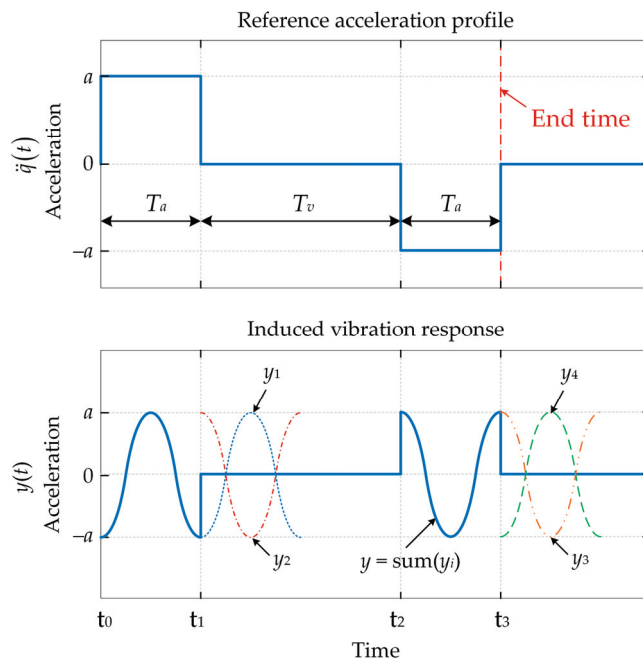


Figure 3. (Top) Reference acceleration profile and (bottom) induced vibration response when vibration suppression condition C1 is satisfied.

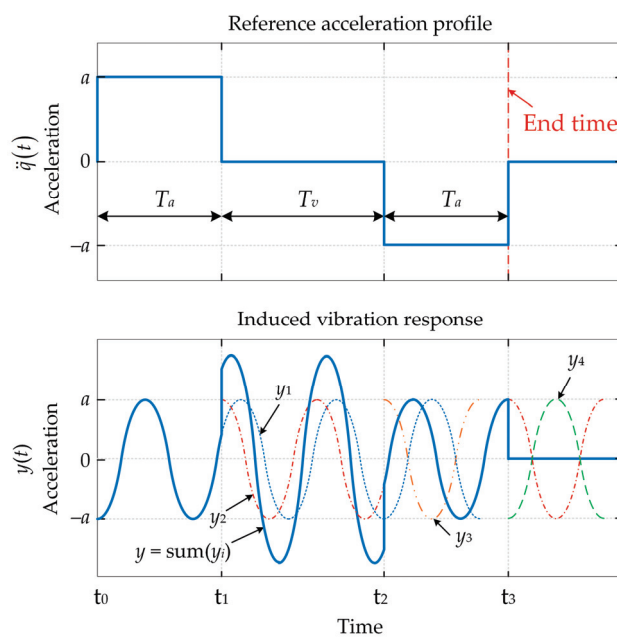


Figure 4. (Top) Reference acceleration profile and (bottom) induced vibration response when vibration suppression condition C2 is satisfied.

3.2. Tuning Procedure

To obtain the optimized vibration-free solution, we can first calculate the time-optimal parameters T_a , T_v , and corresponding t_1 , t_2 under actuator limits according to Equations (5) and (6) and then modulate them according to the vibration suppression conditions. Note that there exist infinite solutions that can satisfy either one of the two conditions. To improve motion efficiency, it is expected that the trajectory duration can be minimized while satisfying these conditions. In practice, between conditions C1 and C2, the one that yields a shorter motion duration can be chosen. Alternatively, both conditions can be chosen simultaneously to improve robustness. The optimization problem can be described as follows:

$$\begin{cases} \min_{t_1, t_2, a} T_{tot} = 2T_a + T_v = t_1 + t_2 \\ \text{s.t. } vt_2 = h \\ t_1 = k_1 T_n \text{ or } t_2 = k_2 T_n \ (k_1, k_2 \in \mathbb{N}^+), t_1 > 0, t_2 \geq t_1 \\ 0 < a \leq A_{\max}, v = at_1 \leq V_{\max} \end{cases} \quad (15)$$

3.2.1. Optimization with Condition C1

We use the hat signal to distinguish the parameters after tuning. If condition C1 is chosen, the value of t_1 is modulated as follows:

$$\hat{t}_1 = \text{ceil}\left(\frac{t_1}{T_n}\right) T_n \quad (16)$$

where $\text{ceil}()$ returns the nearest integer that is greater than or equal to the input value. Note that since $\hat{T}_v \geq 0$, the constraint $\hat{t}_2 \geq \hat{t}_1$ must hold. To ensure that $\hat{t}_2$ meets this constraint, the value of t_2 is modulated as follows:

$$\hat{t}_2 = \max\{\hat{t}_1, t_2\} \quad (17)$$

Then, $\hat{T}_a$ and $\hat{T}_v$ can be obtained, and the peak acceleration is recomputed by:

$$\hat{a} = \frac{h}{\hat{t}_1 \hat{t}_2} \quad (18)$$

The peak velocity is $\hat{v} = h/\hat{t}_2$. Since $\hat{t}_1$ and $\hat{t}_2$ are not smaller than their original values, the peak acceleration and velocity after tuning definitely fulfill the actuator limits, guaranteeing the feasibility of the trajectory.

3.2.2. Optimization with Condition C2

If condition C2 is chosen, the value of t_1 does not need adjustment, namely $\hat{t}_1 = t_1$. The value of t_2 is modulated as follows:

$$\hat{t}_2 = \text{ceil}\left(\frac{t_2}{T_n}\right) T_n \quad (19)$$

Since the adjusted value is no smaller than its original value, the constraint $\hat{t}_2 \geq \hat{t}_1$ must hold. Then, the peak acceleration is recomputed by Equation (18). Similarly, the actuator limits are still satisfied in this case.

3.2.3. Optimization with Conditions C1 and C2

If both conditions C1 and C2 are chosen, the value of t_2 can be modulated by Equation (19). Note that, in this case, the value of $\hat{t}_1$ is not necessarily larger than or equal to the

original value, as the peak velocity may be changed. Since the total motion duration T_{tot} is $\hat{t}_1 + \hat{t}_2$, $\hat{t}_1$ should be as small as possible to minimize $\hat{T}_{tot}$. The selection of this value should satisfy the acceleration limit condition:

$$\hat{a} = \frac{h}{\hat{t}_1 \hat{t}_2} \leq A_{\max} \Rightarrow \hat{t}_1 \geq \frac{h}{A_{\max} \hat{t}_2} \quad (20)$$

Therefore, the value of t_1 is modulated as follows:

$$\hat{t}_1 = \text{ceil}\left(\frac{h}{A_{\max} \hat{t}_2 T_n}\right) T_n \quad (21)$$

Then, the peak acceleration is recomputed by Equation (18).

4. Sensitivity Analysis

As stated above, the vibration suppression conditions C1 and C2 provide two options for parameter tuning, offering flexibility in implementation. If either of these conditions is satisfied, complete residual vibration elimination can be achieved for an undamped system. The two conditions depend on the system parameter ω_n . Since modeling errors are inevitable in practice, these conditions may not achieve zero vibration. In this case, satisfying both C1 and C2 would improve robustness against parameter uncertainties. This characteristic can be better understood through sensitivity curves [43], which plot the dimensionless residual vibration percentage against variations in natural frequency and damping ratio (see Figure 5). For a clearer visualization, Figure 6 gives an enlarged view of Figure 5.

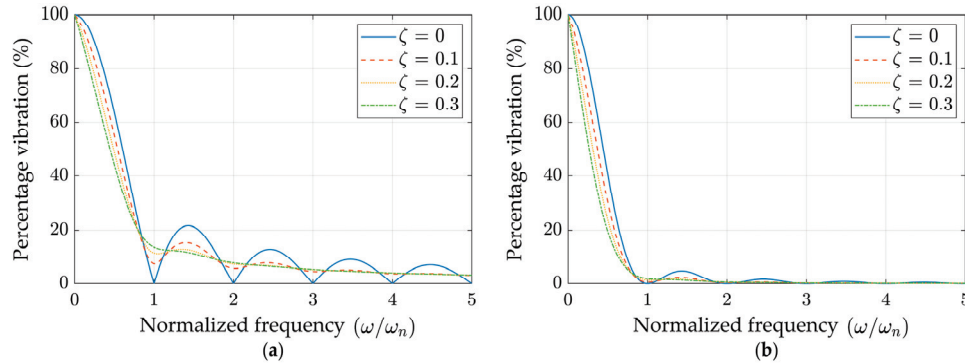


Figure 5. Sensitivity curves of the profiles with respect to variations in the actual natural frequency ω and damping value ζ : (a) optimized trajectory with only one condition C1 or C2; (b) optimized trajectory with both conditions C1 and C2.

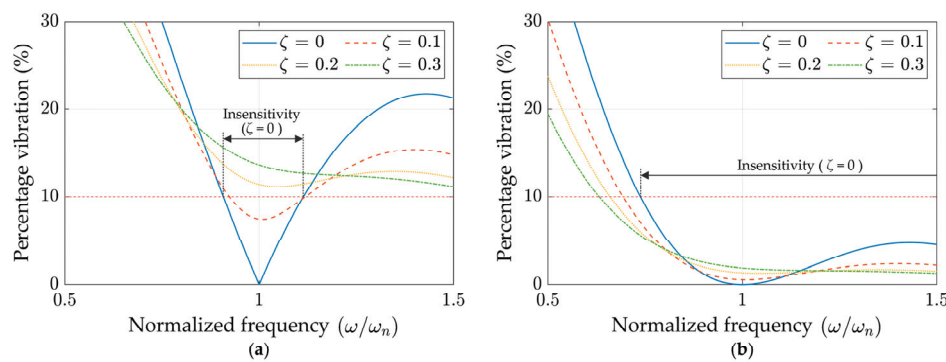


Figure 6. Enlarged view of sensitivity curves in Figure 5: (a) optimized trajectory with only one condition C1 or C2; (b) optimized trajectory with both conditions C1 and C2.

The reference maximum vibration for trapezoidal profiles represents the full-scale value. The reduction trend of the vibration level with increasing actual natural frequency is induced by the relative decrease in the peak acceleration value given by Equation (18). While residual vibration can be eliminated for lightly damped modes, this capability deteriorates as the mismatch between estimated and actual natural frequencies grows. However, the optimized profile with double conditions maintains significantly lower residual vibration than the optimized profile with a single condition within a certain range of modeling error. For quantitative comparison, we define the insensitivity as the range of normalized natural frequency where the dimensionless residual vibration percentage remains below 10%, as indicated by the horizontal dashed red line in Figure 6. Observe that the double conditions produce an optimized profile that exhibits superior robustness compared to the single condition. Particularly, for lightly damped systems, the optimized profile with a single condition provides an insensitivity value of 0.206, while the optimized profile with double conditions provides an infinite insensitivity value as the tolerance is satisfied over the entire frequency band of $\omega/\omega_n > 0.74$. Given that practical modeling errors typically fall within this range, the latter is preferable under parameter uncertainties. Additionally, damping values below 0.1 have a minimal impact on residual vibration levels. This observation supports the common assumption of lightly damped modes for dominant deformations in industrial machines.

5. Experimental Validation

5.1. Overview of Experimental Design

The experimental validation was conducted using two different industrial robots, UR5e and AUBO-i5, to demonstrate the broad applicability of the proposed method. As previously mentioned, the method acts on the motion planning level and is built upon the trapezoidal trajectory inherently integrated within commercial robotic control systems. Therefore, it can be readily implemented on general industrial robots. The experiments were designed to evaluate the effectiveness under varying payload conditions and provide performance comparisons across different motion profiles. In the first experiment, a relatively light payload was used, and the optimized trajectories with different vibration suppression conditions were compared to the unoptimized trapezoidal trajectory provided by the robot. In the second experiment, a more complex and heavier payload was used, and the optimized trajectories were further compared with benchmark methods. In this study, since the robots feature rigid links and possess payload capacities significantly lower than their own weight, changes in payload have negligible effects on the robots' dynamic performance. The vibratory response is predominantly determined by the fundamental flexible mode of the payload. The proposed method optimizes trajectories based on the measured modal frequency to eliminate harmonic components near the resonant mode.

5.2. Experiments on UR5e Robot

5.2.1. Experiment 1: Setup

In the first experiment, tests were conducted on a UR5e robot from Universal Robots. A flexible part was grasped by the end-effector of the robot, which was required to perform a linear transfer with a moving distance of 0.55 m, as shown in Figure 7. The part is made of polypropylene, and the technical drawing in Figure 8 shows its principal dimensions. These physical properties affect the dominant mode of the part. The part with the sensor weighs approximately 0.4 kg and is mounted perpendicular to the direction of motion. The starting and ending positions of the end-effector were (80, 600, 0) mm and (80, 600, 550) mm with the orientation fixed to $(\pi, -\pi/2, 0)$ rad using RPY representation. The velocity and

acceleration constraints are defined as: $V_{\max} = 0.65 \text{ m/s}$, $A_{\max} = 2.4 \text{ m/s}^2$. The high-speed motion of the robot can excite significant vibrations in the flexible part along the z-axis direction. To measure the vibration response of the part, an ICM-20602 accelerometer from TDK InvenSense was attached to the tip of the part. The acceleration signal was recorded by the accelerometer with a sampling rate of 500 Hz. The settling time is defined as the duration from when the movement starts until the part's residual vibration remains consistently within $\pm 2 \text{ m/s}^2$. Using the impact hammer test, the flexible part is approximately characterized by a natural frequency of 5.58 Hz and a damping ratio of 0.01.

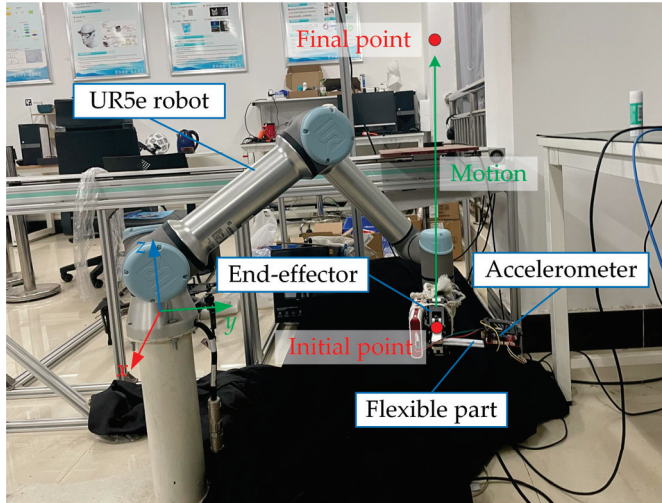


Figure 7. Experimental setup.

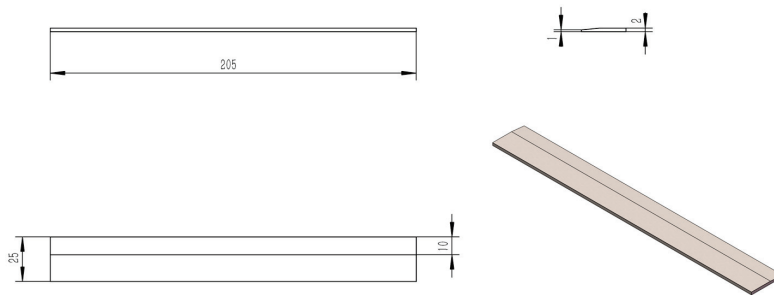


Figure 8. Technical drawing of the first part (unit: mm).

5.2.2. Optimal Parameter Computation

The parameters for the optimized trapezoidal trajectories with different vibration suppression conditions are computed based on the vibration period (0.179 s). To compute these parameters, the minimum time parameters for the original trapezoidal profile are first calculated according to Equations (5) and (6) as follows:

$$T_a = \min \left\{ \frac{0.65}{2.4}, \sqrt{\frac{0.55}{2.4}} \right\} = 0.271 \text{ s}, T_v = \frac{0.55 - 2.4 \times 0.271^2}{0.65} = 0.575 \text{ s} \quad (22)$$

Thus, the corresponding switching times are $t_1 = T_a = 0.271 \text{ s}$, $t_2 = T_a + T_v = 0.846 \text{ s}$. Since the acceleration time is determined by the velocity constraint, both the peak velocity and acceleration attain their maximum allowable values, i.e., $v = 0.65 \text{ m/s}$, $a = 2.4 \text{ m/s}^2$. Then, the vibration period is taken into account to tune the above original parameters. In particular, if condition C1 is selected, the optimized parameters are calculated using Equations (16)–(18):

$$\hat{t}_1 = \text{ceil}\left(\frac{0.271}{0.179}\right) \times 0.179 = 0.358 \text{ s}, \hat{t}_2 = \max\{0.358, 0.846\} = 0.846 \text{ s} \quad (23)$$

$$\hat{a} = \frac{0.55}{0.358 \times 0.846} = 1.814 \text{ m/s}^2 \quad (24)$$

The peak velocity does not change as t_2 remains the same. We can then obtain $\hat{T}_a = \hat{t}_1 = 0.358 \text{ s}$, $\hat{T}_v = \hat{t}_2 - \hat{t}_1 = 0.488 \text{ s}$.

If condition C2 is selected, the optimized parameters are calculated from the formulae in Section 3.2.2 as follows:

$$\hat{t}_1 = t_1 = 0.271 \text{ s}, \hat{t}_2 = \text{ceil}\left(\frac{0.846}{0.179}\right) \times 0.179 = 0.896 \text{ s} \quad (25)$$

$$\hat{v} = \frac{0.55}{0.896} = 0.614 \text{ m/s}, \hat{a} = \frac{0.55}{0.271 \times 0.896} = 2.267 \text{ m/s}^2 \quad (26)$$

We can then obtain $\hat{T}_a = \hat{t}_1 = 0.271 \text{ s}$, $\hat{T}_v = \hat{t}_2 - \hat{t}_1 = 0.625 \text{ s}$.

Finally, if both C1 and C2 are selected, the optimized parameters are calculated from Equations (18)–(21):

$$\hat{t}_2 = \text{ceil}\left(\frac{0.846}{0.179}\right) \times 0.179 = 0.896 \text{ s}, \hat{t}_1 = \text{ceil}\left(\frac{0.55}{2.4 \times 0.896 \times 0.179}\right) \times 0.179 = 0.358 \text{ s} \quad (27)$$

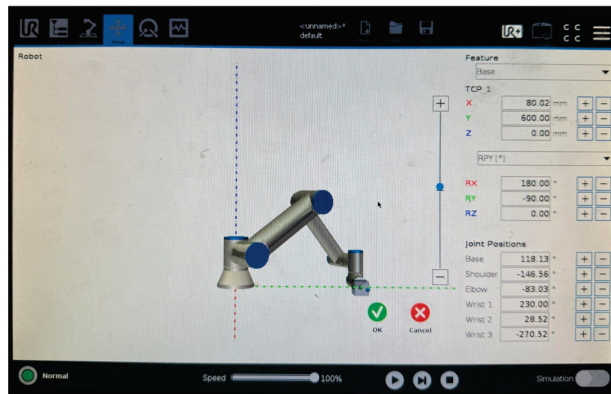
$$\hat{v} = \frac{0.55}{0.896} = 0.614 \text{ m/s}, \hat{a} = \frac{0.55}{0.358 \times 0.896} = 1.713 \text{ m/s}^2 \quad (28)$$

We can then obtain $\hat{T}_a = \hat{t}_1 = 0.358 \text{ s}$, $\hat{T}_v = \hat{t}_2 - \hat{t}_1 = 0.538 \text{ s}$. These parameters can be directly set in the robot controller to generate the desired trajectories. Therefore, the implementation can be achieved through standard robot programming interfaces without specialized expertise. The corresponding input panels are depicted in Figure 9. The MoveL command is used to achieve a precise straight-line movement of the robot tool between two path points. The necessary parameters for determining the movement include the reference frame, the poses of the desired path points, and the tool speed and acceleration.

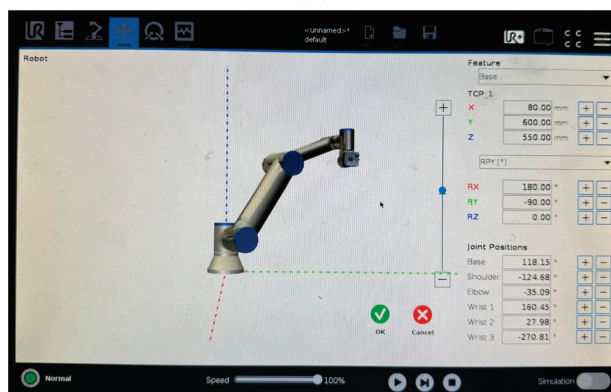
The unoptimized trapezoidal profile, which is the robot's default setting, is taken as a baseline for comparison. Table 1 lists all the computed time and kinematic parameters. Note that the original unoptimized trapezoidal profile has a shorter duration than the optimized trajectories. Therefore, to provide a more precise comparison, the unoptimized trapezoidal profile is uniformly scaled to match the duration of each optimized trajectory; namely, the acceleration time and cruise time are scaled with the same ratio. These scaled trajectories are denoted as scaling1, scaling2, and scaling3, whose durations correspond to those of the optimized trajectories with C1, C2, and both C1 and C2, respectively. For instance, the duration of the unoptimized trajectory scaling1 is identical to that of the optimized trajectory with C1, so its time parameters are calculated as follows: $t_1 = T_a = 0.271 \times 1.205/1.117 = 0.292 \text{ s}$, $T_v = 0.575 \times 1.205/1.117 = 0.621 \text{ s}$.

Table 1. Computed trajectory parameters for experiment 1.

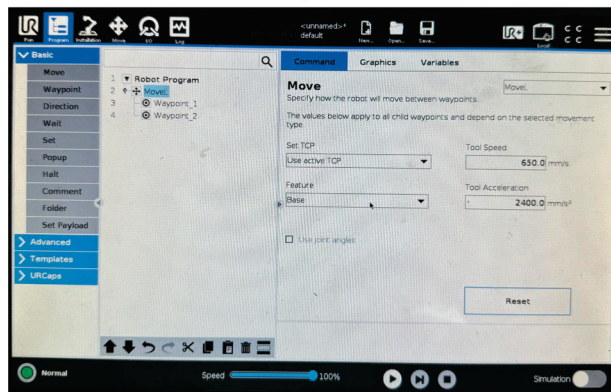
Trajectory		T_a (s)	T_v (s)	T_{tot} (s)	v (m/s)	a (m/s ²)
Unoptimized trapezoidal velocity	Original	0.271	0.575	1.117	0.65	2.4
	Scaling1	0.292	0.621	1.205	0.603	2.064
	Scaling2	0.283	0.601	1.167	0.622	2.199
	Scaling3	0.304	0.646	1.255	0.579	1.903
Optimized trapezoidal velocity	C1	0.358	0.488	1.205	0.65	1.814
	C2	0.271	0.625	1.167	0.614	2.267
	C1, C2	0.358	0.538	1.255	0.614	1.713



(a)



(b)



(c)

Figure 9. User input panels of the robot controller: (a) initial waypoint setting; (b) final waypoint setting; (c) movement parameter setting.

5.2.3. Experiment 1: Results

The generated position, velocity, and acceleration profiles in the Cartesian coordinate system are depicted in Figures 10 and 11. It is shown that all the motion trajectories are bounded within the kinematic constraints. The measured vibration responses obtained from different trajectories are presented in Figures 12–15. The vertical dashed red lines indicate the completion time of the motion, and the light-blue area between the horizontal gray dashed lines represents the allowable error band. The comparison results in terms of the peak-to-peak residual vibration amplitude and settling time are summarized in Table 2.

It can be seen that the unoptimized trapezoidal profile produces significant residual vibration close to 12 m/s^2 , requiring a considerably longer period for the vibration to dissipate to acceptable levels. This would delay subsequent operations after the mo-

tion is completed. In contrast, while the optimized trapezoidal profile proposed in this work slightly increases the motion completion time relative to the unoptimized one, this trade-off leads to a substantial reduction in the vibration level. As a result, the system's response reaches the required tolerance more quickly with the optimized profile, leading to a shortened overall settling time. A slight vibration still remains, even when the trajectory parameters are properly tuned, due to the tracking error and estimation error in the modal properties. Different selections of vibration suppression conditions produce different trajectory durations and vibration amplitudes. When condition C2 is chosen, the trajectory duration is the shortest, only 0.05 s longer than that of the unoptimized one. The optimized trajectories reduce the residual vibration by 71.6% under condition C1 and by 60.2% under condition C2, compared to the unoptimized trajectory. Notably, the selection of both conditions C1 and C2 produces the best performance in terms of vibration mitigation, as the vibration level is reduced by 76.2%. The settling time is identical to the trajectory duration, as the vibration amplitude settles within the desired bounds almost immediately after motion completion. Therefore, the proposed approach provides several benefits, including faster settling, which enables higher production rates in pick-and-place operations, and improved positioning accuracy, which leads to enhanced product quality.

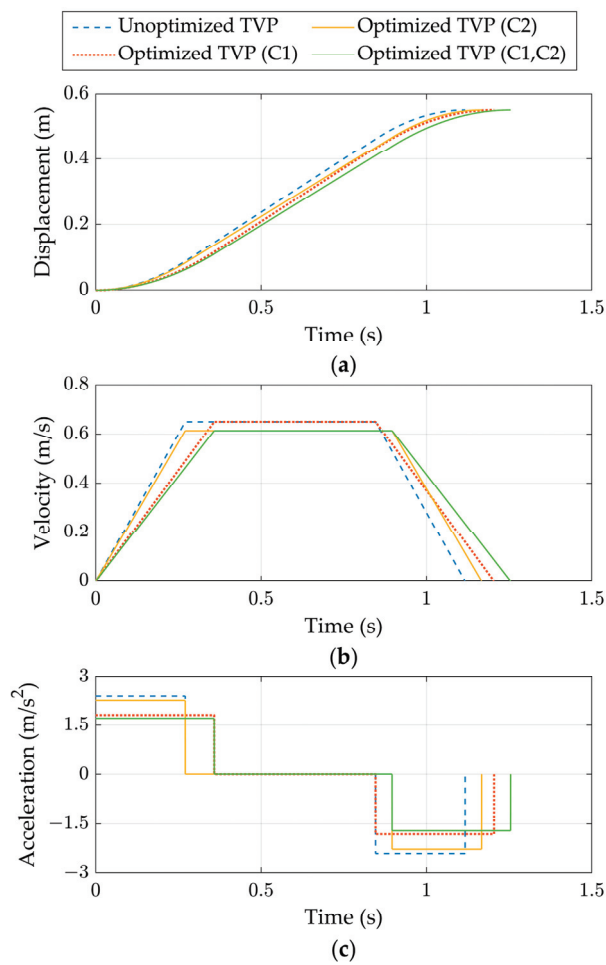


Figure 10. (a) Position; (b) velocity and (c) acceleration profiles generated by the unoptimized and optimized trapezoidal trajectory planning techniques (TVP: trapezoidal velocity profile).

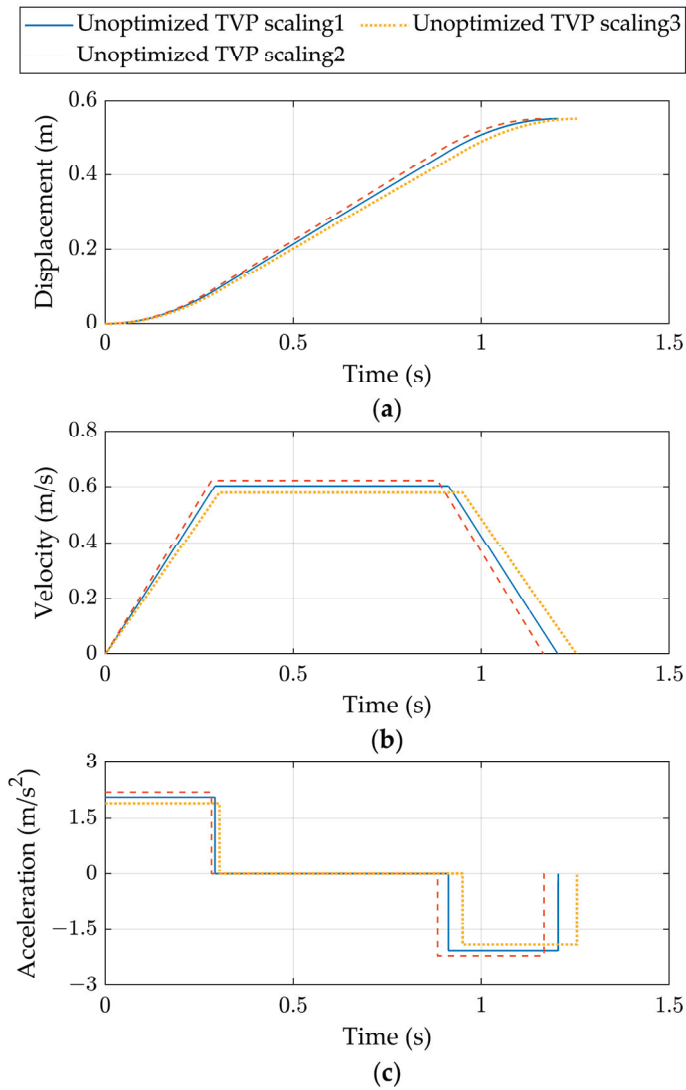


Figure 11. (a) Position; (b) velocity and (c) acceleration profiles generated by the unoptimized trapezoidal trajectory planning technique after scaling (TVP: trapezoidal velocity profile).

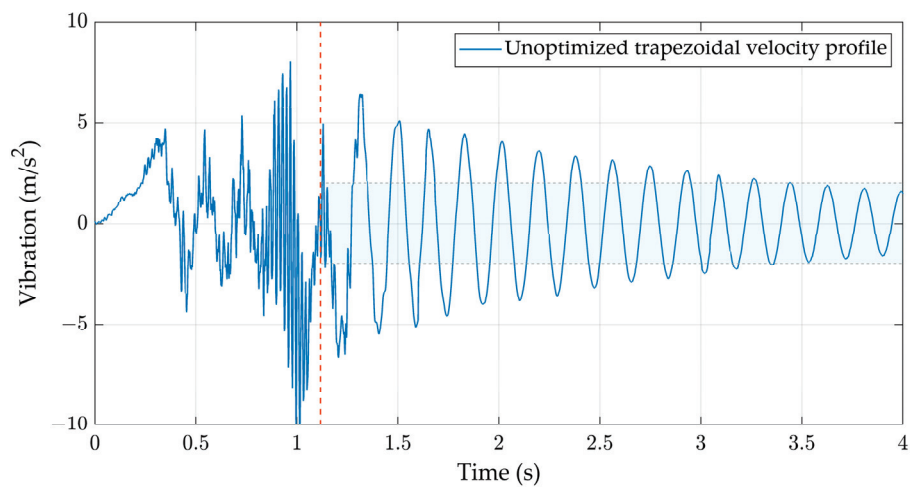


Figure 12. Vibration response with the unoptimized trapezoidal velocity trajectory.

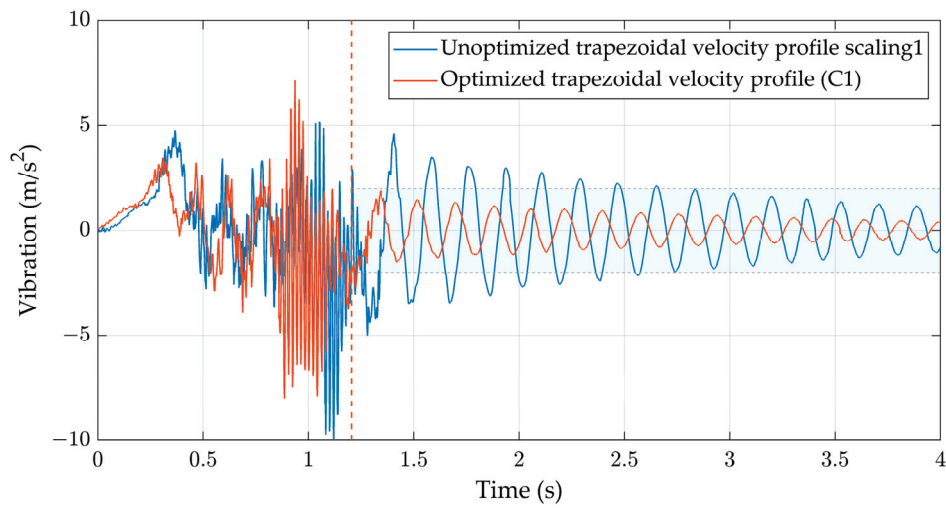


Figure 13. Vibration responses with the unoptimized trapezoidal velocity trajectory after scaling and optimized trapezoidal velocity trajectory (condition C1 is chosen).

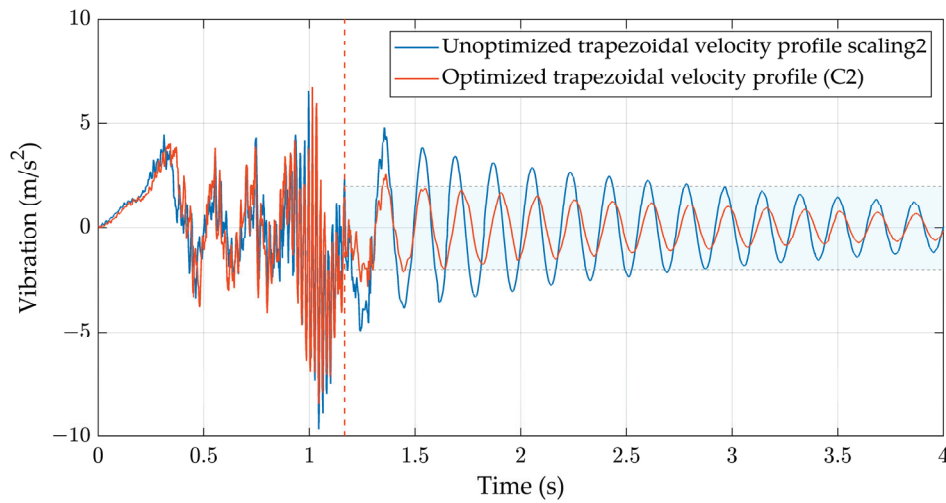


Figure 14. Vibration responses with the unoptimized trapezoidal velocity trajectory after scaling and optimized trapezoidal velocity trajectory (condition C2 is chosen).

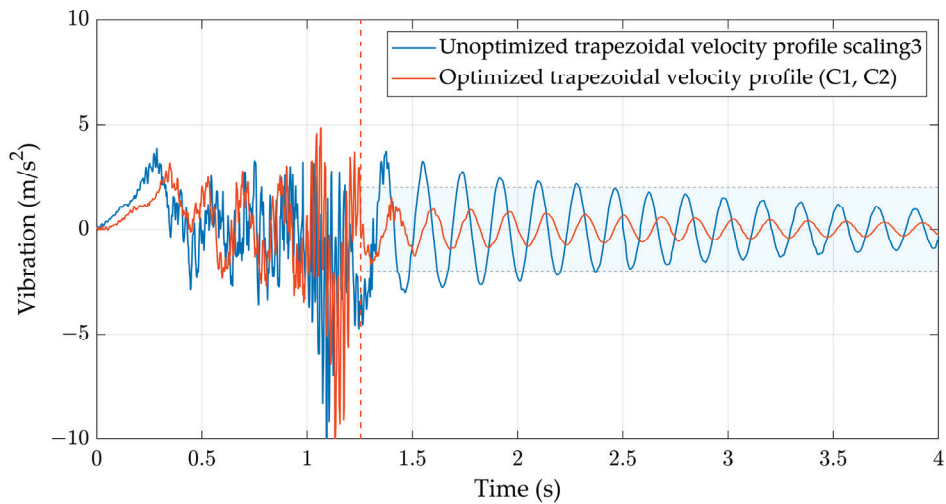


Figure 15. Vibration responses with the unoptimized trapezoidal velocity trajectory after scaling and optimized trapezoidal velocity trajectory (conditions C1 and C2 are chosen).

Table 2. Comparative results of residual vibrations and settling times induced by different motion profiles.

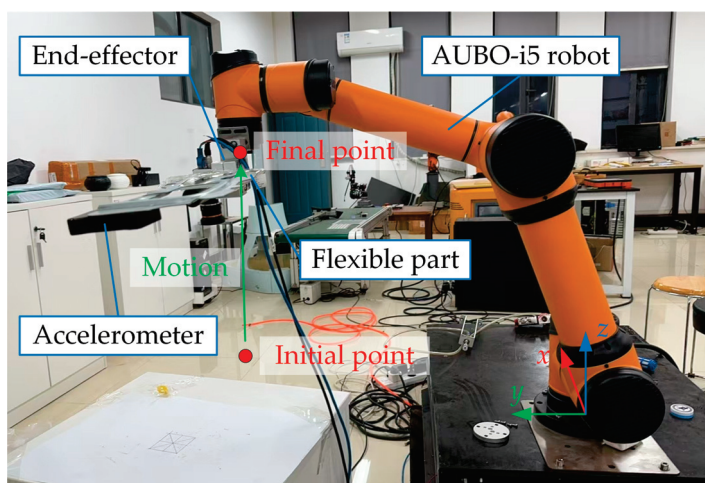
Trajectory		Residual Vibration (m/s ²)	Settling Time (s)
Unoptimized trapezoidal velocity	Original	11.92	3.444
	Scaling1	8.86 (ref1)	2.664 (ref1)
	Scaling2	8.70 (ref2)	2.794 (ref2)
	Scaling3	6.74 (ref3)	2.376 (ref3)
Optimized trapezoidal velocity	C1	3.42 (−51.8%)	1.232 (−53.8%)
	C2	4.71 (−44.4%)	1.454 (−48.0%)
	C1, C2	2.83 (−60.8%)	1.255 (−47.2%)

This method is particularly cost-effective because it leverages existing robot controllers and does not require any additional hardware or software modification. The only cost incurred is the slightly longer motion duration. However, this delay is compensated for by the significant reduction in settling time. For instance, compared to the unoptimized trajectory, the trajectory optimized with C1 yields a duration increase of 0.088 s but reduces the total settling time by 2.212 s. Additionally, the scaled trapezoidal trajectories mitigate the vibration to some extent relative to the original unoptimized one due to the peak acceleration reduction caused by longer durations. However, as shown in Table 2, compared to the proposed approach, the residual vibration reduction achieved by simple time-scaling is relatively limited, even with identical overall motion durations.

5.3. Experiments on AUBO-i5 Robot

5.3.1. Experiment 2: Setup

To validate the applicability of the proposed method across different robotic platforms and payload conditions, the second experiment was conducted on an AUBO-i5 robot with a heavier transferred part attached to the end-effector, as shown in Figure 16. The part is made of aluminum alloy, and the technical drawing in Figure 17 depicts its principal dimensions. The flexible part is approximately characterized by a natural frequency of 8.6 Hz and a damping ratio of 0.011. The weight of the part is about 2.2 kg. Similarly, the robot performed a straight-line motion with a moving distance of 0.4 m from (−50, −620, 100) mm to (−50, −620, 500) mm with the orientation fixed to (0, 0, $-\pi/2$) rad using RPY representation. The velocity and acceleration constraints are defined as: $V_{\max} = 0.55$ m/s, $A_{\max} = 2.5$ m/s². A wireless accelerometer was mounted on the part to measure the vibration response along the z-axis direction with a sampling rate of 200 Hz. The settling time is defined as the duration from when the movement starts until the part's residual vibration remains consistently within ± 0.5 m/s².

**Figure 16.** Experimental setup of AUBO-i5 robot.

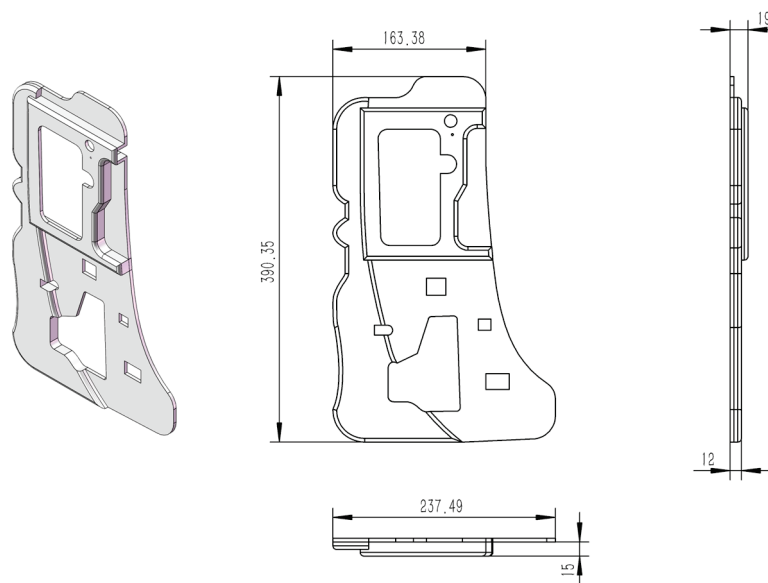


Figure 17. Technical drawing of the second part (unit: mm).

The unoptimized trapezoidal profile is taken as a baseline. The parameters for the optimized trapezoidal trajectories with single and double conditions are computed based on the vibration period (0.116 s). Table 3 lists all the computed time and kinematic parameters. Additionally, the optimized S-curve [10] and seventh-order polynomial [17] profiles introduced in previous studies were also tested for comparison. As mentioned earlier, an important advantage of the proposed method is that its implementation does not require any additional resources. Therefore, the optimized trapezoidal trajectories can be generated directly using built-in commands provided by the robot control panel, while the S-curve and polynomial profiles were implemented on a host computer with the Ubuntu system that served as the primary controller. A Python program was developed to incorporate the algorithms for computing the position data. The generated data are then converted into specific commands using MoveIt within the ROS environment and sent to the robot control cabinet, thereby driving the robot to track the trajectory.

Table 3. Computed trajectory parameters for experiment 2.

Trajectory	T_a (s)	T_v (s)	T_{tot} (s)	v (m/s)	a (m/s ²)
Unoptimized trapezoidal velocity	0.220	0.507	0.947	0.55	2.5
Optimized C1	0.233	0.495	0.960	0.55	2.365
Optimized trapezoidal velocity C1, C2	0.233	0.581	1.047	0.491	2.113

5.3.2. Experiment 2: Results

The generated position, velocity, and acceleration profiles in the Cartesian coordinate system are depicted in Figure 18. Note that the seventh-order polynomial trajectory proposed in [17] results in a peak velocity higher than its limit value, as this method only considers the acceleration limit for parameter design. The measured vibration responses are presented in Figure 19, and the results are summarized in Table 4. The colored vertical dashed lines indicate the motion completion time for each trajectory.

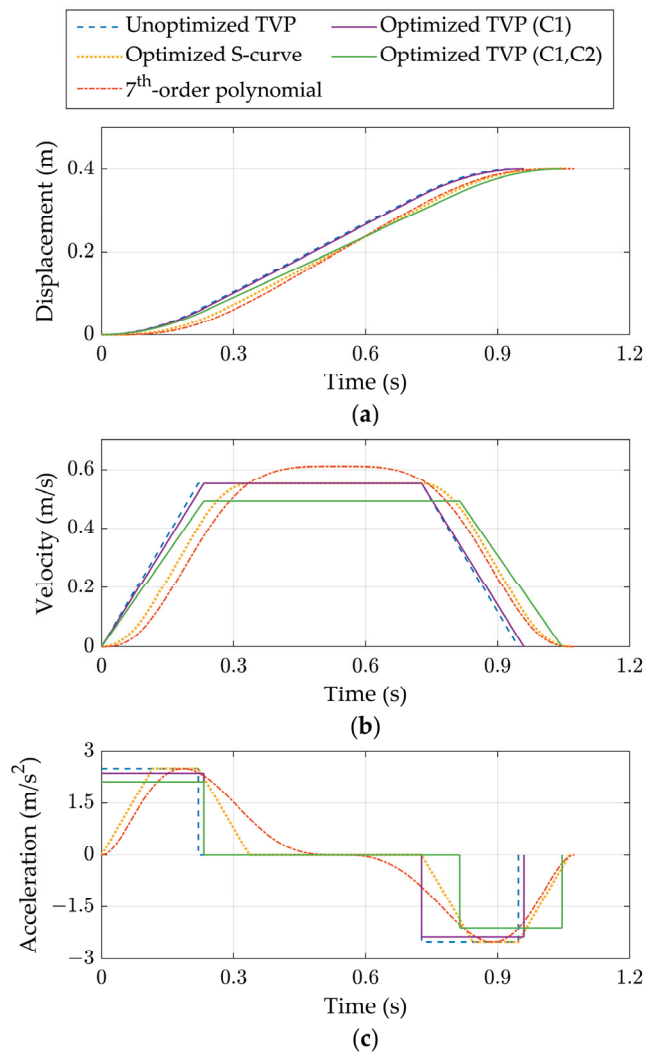


Figure 18. (a) Position; (b) velocity and (c) acceleration profiles generated by the unoptimized trapezoidal trajectory, optimized S-curve, seventh-order polynomial, and optimized trapezoidal trajectory planning techniques (TVP: trapezoidal velocity profile).

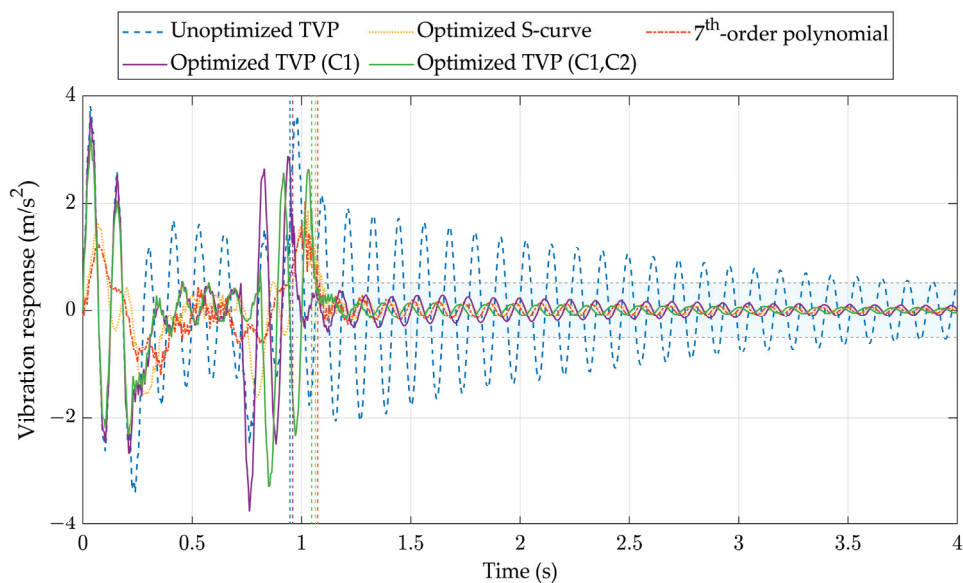


Figure 19. Measured vibration responses with different techniques.

Table 4. Comparison of results produced by different motion profiles.

Trajectory	Residual Vibration (m/s ²)	Execution Time (s)	Settling Time (s)
Unoptimized trapezoidal velocity	4.26 (ref)	0.947 (ref)	4.010 (ref)
Optimized S-curve [10]	0.33 (−92.3%)	1.064 (+12.3%)	1.095 (−72.7%)
Seventh-order polynomial [17]	0.46 (−89.2%)	1.074 (+13.4%)	1.085 (−72.9%)
Optimized C1	0.79 (−81.5%)	0.960 (+1.3%)	0.980 (−75.6%)
Optimized trapezoidal velocity C1, C2	0.35 (−91.8%)	1.047 (+10.5%)	1.065 (−73.4%)

As expected, the optimized trapezoidal trajectory achieves significantly better performance than the unoptimized one. The optimized trapezoidal trajectory with C1 results in an 81.5% attenuation of residual vibration, while the optimized trapezoidal trajectory with both C1 and C2 results in a 91.8% attenuation. In this experiment, the remaining residual vibration is smaller than in the previous experiment, as the part is characterized by a relatively higher mode, thus being less influenced by parameter estimation errors. Notably, the residual vibration amplitude caused by the optimized trapezoidal trajectory with double conditions is comparable to that from the optimized S-curve profile and slightly smaller than that from the seventh-order polynomial profile while leading to shorter time delay. The optimized trapezoidal trajectory with C1 yields the shortest settling time, as the resulting residual vibration is marginally larger but has a remarkably shorter motion duration than the other optimized trajectories. In practice, choosing a suitable profile could help minimize the settling time. A limitation of the proposed method is that the vibration during the acceleration and deceleration phases of the movement is higher than that of the S-curve and polynomial profiles. If the vibration during the entire motion is also required to be attenuated, smoother profiles may be preferred. Since the proposed method operates at the motion planning level using standard programming, it demonstrates good adaptability to various load conditions and robot types.

In this study, the two robots possess similar link dimensions. If the robot has flexible links with a significantly greater length or a considerably larger payload mass, its dynamic performance may be substantially affected by the payload. Longer links lead to lower natural frequencies and increased dynamic coupling, resulting in higher deflection amplitudes. The flexible element type impacts mass distribution during different robot poses and inertial forces and may result in multiple vibration modes and complex mode shapes. In this case, the influence of the payload on dynamic performance needs to be considered. However, within the scope of this study, the robots maintain good dynamic performance as long as the payload is within the allowable range. Therefore, given the same robot type and payload conditions, the proposed method maintains consistent vibration reduction performance across the payload range relative to benchmark techniques.

Overall, the results clearly demonstrate the effectiveness of the proposed trajectory optimization method for high-speed linear point-to-point motion, which is a common movement pattern in practice. Furthermore, this method can be extended to more complex movement types and systems by incorporating specific considerations, such as multi-axis synchronization or multimode elimination strategies, which will be addressed in our future work.

6. Conclusions

This study proposes a systematic optimization method for the trapezoidal velocity profile to reduce motion-induced vibrations in flexible systems. One main advantage is its easy implementation in common industrial robots and machines without requiring computational complexity and modification of the controller. The mathematical expressions of the profile and the minimum time solution under actuator limits are first developed. By

examining the dynamic response characteristics, it is found that the vibration amplitude is associated with the acceleration/deceleration switching times. Two vibration suppression conditions are derived. Based on these conditions, an analytical parameter tuning procedure is then given to generate motion with minimized vibration and execution time according to the natural period. Experiments conducted on two robot manipulators with different attached flexible parts demonstrate the effectiveness and applicability of the proposed method in reducing residual vibration and settling time. This approach offers a practical solution for improving precision and productivity in industrial applications.

This study opens several avenues for future research. One potential direction is exploring trajectory optimization techniques for more complex scenarios, such as multi-mode systems or multi-axis synchronized motions with varying dynamic properties. Additionally, extending the method to other flexible systems, such as those with multiple flexible links or significant payload mass, could further broaden its applicability. Experimental validation will be performed using robots with larger link lengths and more flexible payloads that exhibit complex vibration patterns, requiring consideration of their influence on the robot's dynamic performance. Investigating the integration of machine learning algorithms to predict and reduce vibrations in real-time represents another promising direction. These efforts could lead to more robust and versatile vibration reduction solutions across diverse applications.

Author Contributions: Conceptualization, X.C. and Y.F.; methodology, X.C. and Y.F.; software, X.C.; validation, Y.F. and C.G.; investigation, X.C.; resources, C.G.; writing—original draft, X.C. and Y.F.; writing—review and editing, Y.F.; supervision, C.G.; project administration, C.G.; funding acquisition, Y.F. and C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant numbers 52105031, 62273235) and the China Postdoctoral Science Foundation (grant number 2022T150407).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ribeiro, J.; Lima, R.; Eckhardt, T.; Paiva, S. Robotic Process Automation and Artificial Intelligence in Industry 4.0—A Literature Review. *Procedia Comput. Sci.* **2021**, *181*, 51–58. [CrossRef]
2. Arents, J.; Greitans, M. Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Appl. Sci.* **2022**, *12*, 937. [CrossRef]
3. Fang, Y.; Hu, J.; Qi, J.; Liu, W.; Wang, W.; Peng, Y. Planning Trigonometric Frequency Central Pattern Generator Trajectory for Cyclic Tasks of Robot Manipulators. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2019**, *233*, 4014–4031. [CrossRef]
4. Tian, Y.; Yue, X.; Wang, L.; Feng, Y. Vibration Suppression of Collaborative Robot Based on Modified Trajectory Planning. *Ind. Robot* **2023**, *50*, 45–55. [CrossRef]
5. Borys, S.; Kaczmarek, W.; Laskowski, D.; Polak, R. Experimental Study of the Vibration of the Spot Welding Gun at a Robotic Station. *Appl. Sci.* **2022**, *12*, 12209. [CrossRef]
6. Mishra, N.; Singh, S.P. Determination of Modes of Vibration for Accurate Modelling of the Flexibility Effects on Dynamics of a Two Link Flexible Manipulator. *Int. J. Non-Linear Mech.* **2022**, *141*, 103943. [CrossRef]
7. Do, T.T.; Vu, V.H.; Liu, Z. Linearization of Dynamic Equations for Vibration and Modal Analysis of Flexible Joint Manipulators. *Mech. Mach. Theory* **2022**, *167*, 104516. [CrossRef]
8. Biglari, H.; Golmohammadi, M.; Hayati, S.; Hemmati, S. Vibration Reduction of a Flexible Robot Link Using a Frictional Damper. *JVC/J. Vib. Control* **2021**, *27*, 985–997. [CrossRef]

9. Zheng, K. Research on Intelligent Vibration Suppression Control of High-Speed Lightweight Delta Robot. *JVC/J. Vib. Control* **2022**, *28*, 3042–3057. [CrossRef]
10. Kim, J.; Croft, E.A. Preshaping Input Trajectories of Industrial Robots for Vibration Suppression. *Robot. Comput. Integr. Manuf.* **2018**, *54*, 35–44. [CrossRef]
11. Yang, W.; Li, S.; Luo, X. Data Driven Vibration Control: A Review. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1898–1917. [CrossRef]
12. Chettibi, T. Smooth Point-to-Point Trajectory Planning for Robot Manipulators by Using Radial Basis Functions. *Robotica* **2019**, *37*, 539–559. [CrossRef]
13. Fang, Y.; Zhu, G.N.; Zhao, Y.; Gu, C. Design Procedure for Motion Profiles with Sinusoidal Jerk for Vibration Reduction. *Appl. Sci.* **2023**, *13*, 13320. [CrossRef]
14. Mohamed, Z.; Tokhi, M.O. Command Shaping Techniques for Vibration Control of a Flexible Robot Manipulator. *Mechatronics* **2004**, *14*, 69–90. [CrossRef]
15. Liu, S.; Wu, C.; Liang, L.; Zhao, B.; Sun, R. Research on Vibration Suppression Methods for Industrial Robot Time-Lag Filtering. *Machines* **2024**, *12*, 250. [CrossRef]
16. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*; Springer Science & Business Media: Berlin, Germany, 2010; ISBN 1846286417.
17. Boryga, M.; Graboś, A. Planning of Manipulator Motion Trajectory with Higher-Degree Polynomials Use. *Mech. Mach. Theory* **2009**, *44*, 1400–1419. [CrossRef]
18. Wu, M.; Mei, J.; Zhao, Y.; Niu, W. Vibration Reduction of Delta Robot Based on Trajectory Planning. *Mech. Mach. Theory* **2020**, *153*, 104004. [CrossRef]
19. Boryga, M.; Kołodziej, P.; Gołacki, K. The Use of Asymmetric Polynomial Profiles for Planning a Smooth Trajectory. *Appl. Sci.* **2022**, *12*, 12284. [CrossRef]
20. Ekrem, Ö.; Aksoy, B. Trajectory Planning for a 6-Axis Robotic Arm with Particle Swarm Optimization Algorithm. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106099. [CrossRef]
21. Abe, A. Trajectory Planning for Residual Vibration Suppression of a Two-Link Rigid-Flexible Manipulator Considering Large Deformation. *Mech. Mach. Theory* **2009**, *44*, 1627–1639. [CrossRef]
22. Cui, L.; Wang, H.; Chen, W. Trajectory Planning of a Spatial Flexible Manipulator for Vibration Suppression. *Robot. Auton. Syst.* **2020**, *123*, 103316. [CrossRef]
23. Nguyen, K.D.; Ng, T.C.; Chen, I.M. On Algorithms for Planning S-Curve Motion Profiles. *Int. J. Adv. Robot. Syst.* **2008**, *5*, 99–106. [CrossRef]
24. Wu, N.; Jia, D.; Li, Z.; He, Z. Trajectory Planning of Robotic Arm Based on Particle Swarm Optimization Algorithm. *Appl. Sci.* **2024**, *14*, 8234. [CrossRef]
25. Fang, Y.; Gu, C.; Zhao, Y.; Wang, W.; Guan, X. Smooth Trajectory Generation for Industrial Machines and Robots Based on High-Order S-Curve Profiles. *Mech. Mach. Theory* **2024**, *201*, 105747. [CrossRef]
26. Kudara, K.; Takahashi, H.; Sasai, S.; Sakurai, H.; Okubo, M.; Nakayama, H.; Maedo, T.; Uchiyama, N. Time-Optimal Motion Generation and Load-Sway Suppression for Rotary Cranes with a Two-Stage S-Curve Trajectory Based on Skilled Operation Analysis. In Proceedings of the 2024 European Control Conference (ECC), Stockholm, Sweden, 25–28 June 2024; pp. 2754–2759.
27. Nadir, B.; Mohammed, O.; Minh-Tuan, N.; Abderrezak, S. Optimal Trajectory Generation Method to Find a Smooth Robot Joint Trajectory Based on Multiquadric Radial Basis Functions. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 297–312. [CrossRef]
28. Olier, C.A.G.; Ramírez, H.E.M.; Higuera, L.M.M. A Novel Methodology for the Selection of the Optimal Velocity Profile for Planned Point-to-Point Trajectories in 1-DoF Manipulators. *Int. J. Adv. Manuf. Technol.* **2024**, *134*, 2051–2065. [CrossRef]
29. Fang, Y.; Hu, J.; Liu, W.; Chen, B.; Qi, J.; Ye, X. A CPG-Based Online Trajectory Planning Method for Industrial Manipulators. In Proceedings of the 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tokyo, Japan, 20–22 July 2016; pp. 41–46.
30. Bruzzone, L.; Stretti, D. Application of Elliptic Jerk Motion Profile to Cartesian Space Position Control of a Serial Robot. *Appl. Sci.* **2023**, *13*, 5601. [CrossRef]
31. Esau, G.V.; Carlos, R.D.; Ivonne, M.C.R.; Eduardo, C.Y.; Juan Manuel, L.H.; Omar, P.S. Novel Gaussian Acceleration Profile for Smooth Jerk-Bounded Trajectories. *IEEE Access* **2022**, *10*, 120714–120723. [CrossRef]
32. Boscariol, P.; Richiedei, D.; Trevisani, A. Direct Solutions for Robust Vibration Suppression through Motion Design. *JVC/J. Vib. Control* **2024**. [CrossRef]
33. Hägele, M.; Nilsson, K.; Pires, J.N.; Bischoff, R. Industrial Robotics. In *Springer Handbooks*; Springer: Cham, Switzerland, 2016; pp. 1385–1422.
34. Bilancia, P.; Schmidt, J.; Raffaelli, R.; Peruzzini, M.; Pellicciari, M. An Overview of Industrial Robots Control and Programming Approaches. *Appl. Sci.* **2023**, *13*, 2582. [CrossRef]

35. Haviland, J.; Corke, P. Robotics Software: Past, Present, and Future. *Annu. Rev. Control Robot. Auton. Syst.* **2024**, *7*, 253–283. [CrossRef]
36. Uyar, M. Investigation of Vibration Control Performance with Modified Motion Planning Based on Basic Functions for Composite Robot Manipulators. *Iran. J. Sci. Technol. Trans. Mech. Eng.* **2024**, *48*, 273–291. [CrossRef]
37. Malgaca, L.; Yavuz, Ş.; Akdağ, M.; Karagülle, H. Residual Vibration Control of a Single-Link Flexible Curved Manipulator. *Simul. Model. Pract. Theory* **2016**, *67*, 155–170. [CrossRef]
38. Karagülle, H.; Malgaca, L.; Dirilmiş, M.; Akdag, M.; Yavuz, Ş. Vibration Control of a Two-Link Flexible Manipulator. *JVC/J. Vib. Control* **2017**, *23*, 2023–2034. [CrossRef]
39. Yoon, H.J.; Chung, S.Y.; Kang, H.S.; Hwang, M.J. Trapezoidal Motion Profile to Suppress Residual Vibration of Flexible Object Moved by Robot. *Electronics* **2019**, *8*, 30. [CrossRef]
40. Uchiyama, N.; Honda, Y.; Sano, S. Residual Vibration Suppression and Energy Saving in Industrial Machines Using a Trapezoidal Velocity Profile. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 323–328.
41. Bolz, R.E. *CRC Handbook of Tables for Applied Engineering Science*; CRC Press: Boca Raton, FL, USA, 2019.
42. Oppenheim, A.V.; Willsky, A.S.; Nawab, S.H. *Signals & Systems*; Pearson Educación: London, UK, 1997; ISBN 0138147574.
43. Singer, N.C.; Seering, W.P. Preshaping Command Inputs to Reduce System Vibration. *J. Dyn. Syst. Meas. Control Trans. ASME* **1990**, *112*, 76–82. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Employing Fuzzy Adaptive and Event-Triggered Approaches to Achieve Formation Control of Nonholonomic Mobile Robots Under Complete State Constraints

Kai Wang ^{1,*}, Jinnan Lu ² and Haodong Zhou ³¹ The Electrical Engineering College, Liaoning University of Technology, Jinzhou 121001, China² Faculty of Mechanics and Engineering, Liaoning University of Engineering and Technology, Fuxin 123000, China; ljn-22@163.com³ The Navigation College, Dalian Maritime University, Dalian 116026, China; zhouhaodong1997@163.com

* Correspondence: ln6848132@163.com

Abstract: This article delves into the problem of fuzzy adaptive event-triggered (ET) formation control for nonholonomic mobile robots (NMRs) subject to full-state constraints. Fuzzy logic systems (FLSs) are employed to identify the unknown nonlinear functions within the system. To guarantee that all system states remain within their constraint boundaries, barrier Lyapunov functions (BLFs) are meticulously constructed. Subsequently, within the framework of the backstepping control design algorithm, we propose a novel fuzzy adaptive ET formation controller. Our ET mechanism can achieve an overall resource-saving rate of 88.17% for the four robots. Rigorous theoretical analysis demonstrates that the designed strategy not only ensures the stability of the controlled NMRs but also enables the formation tracking errors to converge to a small neighborhood around zero. Notably, the BLFs-based control approach presented herein endows the system with the capacity to avoid collisions to a certain degree, enhancing the overall safety and reliability of the robot formation. Finally, a simulation example is provided. The results vividly illustrate the effectiveness and practicality of the proposed theory, validating its potential for real-world applications in the field of nonholonomic mobile robot formation control.

Keywords: barrier Lyapunov function; backstepping control design; formation control; full-state constraints; nonholonomic mobile robot systems

1. Introduction

In recent years, the advantages of multi-agent systems (MASs) over individual agents [1] have become increasingly prominent. MASs are characterized by significantly higher working efficiency and enhanced compatibility, which have attracted extensive attention from researchers in various fields. In the realm of practical engineering, with the exponential growth of network information and the increasing complexity of missions, the coordination control of multiple mobile robots [2] multiple unmanned aerial vehicles [3] and multiple marine surface vehicles [4] has emerged as a crucial research area. These multi-agent systems play a vital role in modern engineering applications, such as industrial automation, intelligent transportation, and marine exploration. For multi-agent formation tracking control, the core goal is to enable multiple agents to collaboratively track a target while moving in a coordinated manner to maintain a predefined formation geometry. This not only requires precise motion control of each agent but also effective communication and

cooperation among them. In this process, maintaining connectivity and avoiding collisions are not only challenging aspects but also fundamental requirements for the stable operation of the multi-agent system. Ensuring seamless communication between robots to maintain connectivity is essential for coordinated movement, while collision-avoidance mechanisms are crucial to prevent potential accidents and ensure the safety and efficiency of the entire system. In the field of multi-agent formation control, a diverse array of strategies has been developed, each with its own unique characteristics and applications. Prominent among these are the leader–follower strategy [5], behavior-based strategy [6], and virtual structure strategy [7]. The leader–follower strategy stands out as one of the most widely adopted and fundamental approaches. Its popularity can be attributed to its simplicity and efficiency. In a leader–follower system, each follower robot only needs to focus on perceiving the information transmitted by its adjacent leader. This significantly reduces the computational burden on individual robots, as they are not required to process a vast amount of data related to the entire external environment.

NMRs' formation control has become a hot research topic due to its important applications in all aspects, such as target searching, environmental monitoring, resource exploring, and area data acquiring. Therefore, a large number of consensus control issues have been reported [8–16]. In [9], the trajectory tracking problem in the formation control of robots was studied. The authors in [10] proposed the angle-constrained-based adaptive formation shape and maneuvering control for NMRs, respectively. The result in [11] suggested a solution for the formation problem using full consensus in torque-controlled NMRs, accounting for the absence of velocity measurements and time-varying communication delays. In earlier studies, most scholars did not consider the internal dynamics of robots. To solve this problem, the use of fuzzy logic systems (FLSs) is a very effective method [17–20]. Hence, for unicycle-type mobile robots, a leader–follower (L-F) formation control protocol was proposed in [16]. In [17], the formation control problem of NMRs with unknown dead zones was studied. The results in [18] show the feasibility of the finite-time neural network adaptive control scheme based on dynamic surface technique. For high-order nonlinear systems, [19] proposed a robust fuzzy adaptive finite-time control scheme.

In practice, the limitations of the speed and position of the NMRs are often overlooked. The running speed and distance between NMRs should be kept within a certain range. The constraints of the robot velocity variables are just as important as the constraints of the output variables, and it makes sense to consider the full-state constraints when designing the controller. In order to ensure all state variables are within constraints, studying the full-state constraint problem in NMR control is important. For example, the Ln-type barrier function is used to constrain the boundaries of the specified performance function in [20]. In [21], the authors used a tan-type barrier function to realize the collision avoidance and connectivity maintenance of the robot. In [22], the formation and collision avoidance problem of NMRs based on the artificial potential field method is proposed. To suppress the violation of tracking error constraints, a barrier Lyapunov function-based control method was proposed to realize the tracking control of incomplete mobile robots under specified performance conditions in [21].

Note that the above control schemes do not effectively control NMRs with unknown dynamics. FLSs are often used to identify unknown functions due to their strong fitting and approximation ability. In [9], a compensator and data-driven control scheme based on FLSs are designed to solve the NMRs' formation control. In [23], the formation control issue of the networked non-complete mobile robot based on the L-F framework was studied. In [24], not only the NMR's adaptive output feedback formation tracking control problem was solved but the connectivity maintenance and collision avoidance were also realized.

In [25], an integrated L-F consensus formation framework using an optimized NN-based distributed model predictive control strategy was presented. Note that the formation control methods in [9,16–25] cannot possibly guarantee full-state constraints and save communication resources at the same time. To the best of our knowledge, there are no results on fuzzy adaptive event-triggered formation controllers for NMRs with full-state constraints. Therefore, designing an adaptive event-triggered formation controller for NMRs under full-state constraints is a challenging and important topic.

Inspired by the above analysis, this paper investigates the fuzzy adaptive formation control of NMRs with nonlinear uncertainty and full-state constraints. Our main contributions are summarized below:

- (1) By utilizing FLSs to estimate the unknown nonlinear functions, and constructing the barrier Lyapunov functions, this paper develops a fuzzy adaptive formation control approach within the adaptive backstepping control design framework. Although the work in [9,21,22] investigated the fuzzy adaptive formation control problems of NMRs, their methods do not consider full-state constraints. Such formation control methods do not constrain the state of the robot within a reasonable range.
- (2) In [24–26], authors investigated the fuzzy adaptive formation control problems of NMRs. These methods cannot decrease the unnecessary waste of resources resulting from the update of the controller signal when considering complete state constraints. Therefore, this paper designs an event-triggering mechanism to avoid unnecessary waste of resources. In Table 1, a comparison of different control algorithms is presented.

Table 1. The comparison results of control algorithm.

Controlled Nonlinear NMRs	Event-Triggered Mechanism	State Constraints	First-Order Filter	Unknown Nonlinear Function
The proposed control method	✓	✓	✓	✓
The control method in [17]	✓	×	✓	✓
The control method in [21]	×	×	×	✓
The control method in [22]	×	×	×	✓

where ✓ indicates that the situation presented was considered in the corresponding article in Table 1.

The rest of this paper is structured as follows. The problem description for this work is the main content in Section 2. Section 3 describes the process of design controller and stability analysis. The proposed event-triggering protocol with numerical examples and summarizes the conclusions is verified in Section 4.

Notations: $|\bullet|$ indicates the absolute value. $\|\bullet\|$ denotes the Euclidean norm for vectors or induced 2-norm for matrices.

2. Problem Formulation

2.1. System Descriptions

Consider a group of NMRs with one passive wheel and two actuated wheels in Figure 1, the mathematical model of the i th ($i = 1, \dots, n$) robot is given by [21]

$$\begin{cases} \dot{\eta}_i = J_i(\eta_i)\bar{\omega}_i \\ M_i\dot{\bar{\omega}}_i = -C_i(\dot{\eta}_i)\bar{\omega}_i - D_i\bar{\omega}_i + \tau_i \end{cases}$$

where $\eta_i = [x_i, y_i, \psi_i]^T$ is the i th robot's posture consisting of the position (x_i, y_i) and heading angle (orientation) (ψ_i) of the robot coordinated. $\bar{\omega}_i = [\bar{\omega}_{1i}, \bar{\omega}_{2i}]^T$ are the angular

velocities of the two wheels, $\tau_i = [\tau_{i1}, \tau_{i2}]^T$ is the control input. M_i , $C_i(\dot{\eta}_i)$, and D_i are the inertia matrix, the centripetal and Coriolis matrix, and the damping matrix, respectively. The matrices $J_i(\eta_i)$, M_i , $C_i(\dot{\eta}_i)$, and D_i in system (1) are given by

$$J_i(\eta_i) = \frac{r_i}{2} \begin{bmatrix} \cos \psi_i & \cos \psi_i \\ \sin \psi_i & \sin \psi_i \\ b_i^{-1} & -b_i^{-1} \end{bmatrix} \quad M_i = \begin{bmatrix} m_{11i} & m_{12i} \\ m_{12i} & m_{11i} \end{bmatrix}$$

$$C_i(\dot{\eta}_i) = \begin{bmatrix} 0 & c_i \dot{\psi}_i \\ -c_i \dot{\psi}_i & 0 \end{bmatrix} \quad D_i = \begin{bmatrix} d_{11i} & 0 \\ 0 & d_{22i} \end{bmatrix}$$

where r_i and b_i shown in Figure 1 are the wheel radius and half-width of the i th mobile robot, respectively. We consider that the NMR model (1), the robot parameters m_{11i} , m_{12i} , b_i , and r_i , the centripetal and Coriolis coefficient c_i , and the damping coefficients d_{11i} and d_{22i} are uncertain constants.

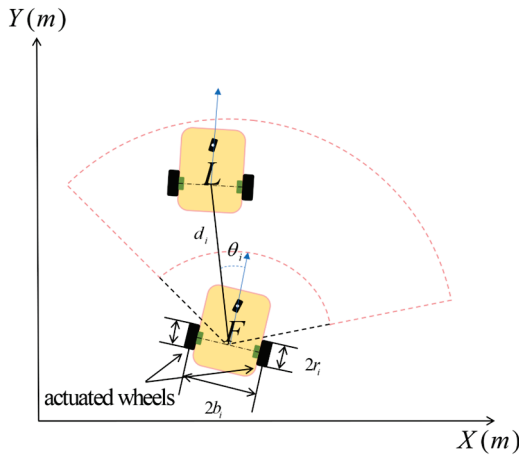


Figure 1. The structure of the L-F model.

To facilitate control development, we can convert the wheel angular velocities $(\bar{\omega}_{1i}, \bar{\omega}_{2i})$ to the robot's linear velocity v_i and angular velocity ω_i by the following equation:

$$\bar{v}_i = B_i^{-1} \bar{\omega}_i$$

with $\bar{v}_i = [v_i, \omega_i]^T$ and $B_i = \frac{1}{r_i} \begin{bmatrix} 1 & b_i \\ 1 & -b_i \end{bmatrix}$, where the matrix B_i is invertible, since the determination of matrix B_i is $\det(B_i) = -2b_i/r_i$. After the above description, we can obtain the following dynamic equations:

$$\begin{cases} \dot{x}_i = v_i \cos(\psi_i) \\ \dot{y}_i = v_i \sin(\psi_i) \\ \dot{\psi}_i = \omega_i \\ M_i \ddot{\omega}_i = -C_i(\dot{\eta}_i) \bar{\omega}_i - D_i \bar{\omega}_i + \tau_i \end{cases} \quad (1)$$

The inertia matrix parameters are given by $m_{11i} = 0.25b_i^{-2}r_i^2(m_i b_i^2 + I_i) + I_{\omega i}$, $m_{12i} = 0.25b_i^{-2}r_i^2(m_i b_i^2 - I_i)$, $m_i = m_{ci} + 2m_{\omega i}$, $I_i = m_{ci}a_i^2 + 2m_{\omega i}b_i^2 + I_{ci} + 2I_{mi}$ and $c_i = 0.5b_i^{-1}r_i^2m_{ci}a_i$, where m_{ci} and $m_{\omega i}$ are the body and wheel (with motor) masses; a_i is the distance between the wheel axis center and robot mass center; and I_{ci} , $I_{\omega i}$, and I_{mi} are the moment of inertia about the vertical axis, the wheel about the wheel axis, and the wheel about the wheel diameter, respectively.

From system (1), the dynamic equations can be defined by

$$\begin{cases} \dot{v}_i = f_{vi}(v_i) + \bar{m}_{1i}\tau_{vi} \\ \dot{\omega}_i = f_{\omega i}(\omega_i) + \bar{m}_{2i}\tau_{\omega i} \end{cases} \quad (2)$$

where f_{vi} and $f_{\omega i}$ are unknown nonlinear functions associated with v_i and ω_i , $\bar{m}_{1i} = \frac{r_i}{2(m_{11i}+m_{12i})}$, and $\bar{m}_{2i} = \frac{r_i}{2b_i(m_{11i}-m_{12i})}$. $\bar{m}_{1i}$ and $\bar{m}_{2i}$ are positive parameters.

The dynamic of the leader NMR is given by [27]

$$\begin{cases} \dot{x}_0 = v_0 \cos(\psi_0) \\ \dot{y}_0 = v_0 \sin(\psi_0) \\ \dot{\psi}_0 = \omega_0 \end{cases}$$

where (x_0, y_0) denotes the position of leader NMR, and ψ_0 is the heading angle. v_0 and ω_0 represent linear velocity and angular velocity, respectively.

Assumption 1 ([21]). The acceleration $\dot{v}_0$ is bounded such that $|\dot{v}_0| \leq v_0^*$, with v_0^* being constant.

From (1), we have

$$\dot{x}_i \sin \psi_i - \dot{y}_i \cos \psi_i = 0$$

which means the angle between the driving wheel's axis and the robot's speed direction is 90 degrees. In other words, the robot's axle-direction speed is always zero, thus it is also called nonholonomic constraint. The nonholonomic constraint ensures the robot's wheels roll without slipping.

2.2. Graph Theory

This subsection will introduce the definitions about the communication network topology.

The expression $\hat{G} \triangleq (\hat{S}, \hat{K})$ represents the transmission relationship between robot information, in which $\hat{S} \triangleq \{1, \dots, N\}$. $\hat{K} \subseteq \hat{S} \times \hat{S}$ indicates the edge set, and $(i, j) \in \hat{K}$ indicates that the i th robot obtains information from the j th robot in one direction. d_{ij} is the distance between i th and j th robot, and d_{Mi} is i th robot's maximum information transmission range. If $d_{ij} < d_{Mi}$, an L-F model can be formed. $L_i(t) = \{j | d_{ij} < d_{Mi}\}$ means the i th robot's leader set. In the L-F structure, the relationship between them can form a tree, thus forming the required L-F structure.

2.3. FLSs

The FLSs used to approximate uncertain model dynamics f_{vi} are $f_{\omega i}$ in (1), and the properties of FLSs can be seen in Lemma 1.

Lemma 1 ([26]). Let $f(x)$ be a continuous function on a compact set Λ . For any constant ε , there always exists a FLS $\hat{f}(x|\hat{\Theta}) = \hat{\Theta}^T \zeta(x)$ satisfying

$$\sup_{x \in \Lambda} |f(x) - \hat{\Theta}^T \zeta(x)| \leq \varepsilon \quad (3)$$

where $\omega(x) = [\omega_1(x), \omega_2(x), \dots, \omega_N(x)]^T / \sum_{i=1}^N \omega_i(x)$ is the fuzzy basic function vector and satisfies $\omega^T(\cdot)\omega(\cdot) \leq 1$. $\Theta = [\Theta_1, \Theta_2, \dots, \Theta_N]^T$ is the ideal weight vector and N is the fuzzy rule number.

Remark 1. In this paper, we use a FLS to approximate the unknown continuous functions in (2). Note that neural networks and type-2 FLSs have also the approximation property of Lemma 1. Therefore, the FLSs used in this paper can be replaced by other approximators.

Control objective: For NMRs (1) with full-state constraints, this paper develops a fuzzy adaptive ET formation control method such that the following hold:

- (1) All the closed-loop signals are semi-globally uniformly ultimately bounded (SGUUB).
- (2) All robots will follow the leader by keeping a desired formation pattern, i.e., the angle and relative distance errors converge asymptotically to zero without violating the angle and relative distance constraints.

3. Fuzzy Adaptive Formation Control Design

In this section, a fuzzy adaptive ET formation controller is constructed under the backstepping control design. With the help of log-type barrier Lyapunov functions, it is proven that the system (1) is stable under full-state constraints and does not collide.

3.1. ET Formation Controller Design

The backstepping control design is a recursive method. For NMRs (1), it contains a two-step design procedure. In Step 1, we first select a Lyapunov function for each subsystem, and based on the Lyapunov function, we design the virtual controllers to stabilize each subsystem, respectively, and design the weight updating laws. In the same way, in Step 2, we select a Lyapunov function for the subsystem, and based on the Lyapunov function, we design the formation controller to stabilize the whole system and design the weight updating law.

To achieve the formation control design, we define the angle and relative distance between i th and j th robot as

$$d_{ij}(t) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4)$$

$$\varphi_{ij}(t) = \text{atan2}(\tilde{y}_i, \tilde{x}_i) \quad (5)$$

$$\begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} = \begin{bmatrix} \cos \psi_i & \sin \psi_i \\ -\sin \psi_i & \cos \psi_i \end{bmatrix} \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \quad (6)$$

where $\text{atan2}(\tilde{y}_i, \tilde{x}_i) \in (-\pi, \pi]$ is the arctangent function of the two arguments (x, y) and returns the appropriate quadrant of the angle of point (x, y) .

In actual formation control, collision avoidance and field of view limitations among robots are crucial issues to consider. Therefore, to fulfill above range constraints, the constraints of distance and angle are described as

$$0 < d_{mi} < d_{ij}(t) < d_{Mi}, \forall t \geq 0 \quad (7)$$

$$-\frac{\pi}{2} < -\varphi_{Mi} < \varphi_{ij}(t) < \varphi_{Mi} < \frac{\pi}{2}, \forall t \geq 0 \quad (8)$$

where d_{Mi} and d_{mi} represent the maximum distance at which connectivity is maintained and the minimum distance at which a collision can be avoided, respectively. φ_{Mi} indicates the maximum angle that ensures continuous information transmission.

Define the coordinate transformation as

$$\begin{cases} z_{id} = d_{ij} - d_{ij,d} \\ z_{i\psi} = \psi_{ij} - \psi_{ij,d} \\ z_{iv} = v_i - \dot{h}_{i,1} \\ z_{i\omega} = \omega_i - \dot{h}_{i,2} \\ \chi_{i,1} = \dot{h}_{i,1} - \alpha_{i,1} \\ \chi_{i,2} = \dot{h}_{i,2} - \alpha_{i,2} \end{cases} \quad (9)$$

where $d_{ij,d}$ is the desired distance, which is a constant. $\psi_{ij,d}$ is the desired angle. $\alpha_{i,s}$, $\chi_{i,s}$ and $\dot{h}_{i,s}$, respectively, are virtual design control functions, first-order filter output errors, and variables ($s = 1, 2$). $\dot{h}_{i,s}$ are obtained through a first-order filter on intermediate virtual controls $\alpha_{i,1}$ and $\alpha_{i,2}$. We introduce the following first-order filter:

$$\delta_{i,1} \dot{\dot{h}}_{i,1} + \dot{h}_{i,1} = \alpha_{i,1}, \quad \dot{h}_{i,1}(0) = \alpha_{i,1} \quad (10)$$

$$\delta_{i,2} \dot{\dot{h}}_{i,2} + \dot{h}_{i,2} = \alpha_{i,2}, \quad \dot{h}_{i,2}(0) = \alpha_{i,2} \quad (11)$$

where $\delta_{i,1} > 0$ and $\delta_{i,2} > 0$ are design parameters.

Step 1: From (9), we have the following equations:

$$\dot{z}_{id} = -v_i \cos \varphi_i + v_{i-1} \cos \gamma_i \quad (12)$$

$$\dot{z}_{i\psi} = -\omega_i + \frac{v_i}{d_i} \sin \varphi_i - \frac{v_{i-1}}{d_i} \sin \gamma_i \quad (13)$$

where v_{i-1} and ψ_{i-1} are the linear velocity and orientation of robot R_{i-1} , respectively. $\gamma_i = \varphi_i + \psi_i - \psi_{i-1}$.

We select log-type barrier Lyapunov functions:

$$V_{11i} = \frac{1}{2} \log \frac{k_{d1}^2}{k_{d1}^2 - z_{id}^2} \quad (14)$$

$$V_{12i} = \frac{1}{2} \log \frac{k_{\varphi 1}^2}{k_{\varphi 1}^2 - z_{i\psi}^2} \quad (15)$$

where k_{d1} and $k_{\varphi 1}$ are design parameters.

We construct the virtual control signals α_{vi} and $\alpha_{\omega i}$ as

$$\alpha_{vi} = \frac{1}{\cos \varphi_i} (v_{i-1} \cos \gamma_i + k_{d1} z_{id}) \quad (16)$$

$$\alpha_{\omega i} = \frac{v_i}{d_i} \sin \varphi_i - \frac{v_{i-1}}{d_i} \sin \gamma_i + k_{\varphi 1} z_{i\psi} \quad (17)$$

with $k_{d1} > 0$ and $k_{\varphi 1} > 0$ being design constants. (Equation (16) defines the virtual control signals designed to stabilize the formation tracking errors. k_{d1} and $k_{\varphi 1}$ are positive control gains, while the logarithmic barrier function ensures the relative azimuth remains within the predefined constraints, thereby avoiding singularity issues.)

Remark 2. If $\varphi_i = \frac{\pi}{2}$, then there will be singularity problem. We predefine constraints on the relative azimuth. The formation tracking controller is designed by using the barrier Lyapunov synthesis method, ensuring that the relative azimuth remains within a predetermined range.

Step 2: From (9), we have

$$\dot{z}_{iv} = f_{vi}(v_i) + m_{1i}\tau_{vi} - \dot{h}_{i,1} \quad (18)$$

$$\dot{z}_{i\omega} = f_{\omega i}(\omega_i) + m_{2i}\tau_{\omega i} - \dot{h}_{i,2} \quad (19)$$

From (9), one has

$$\dot{\chi}_{i,s} = \dot{h}_{i,s} - \dot{\alpha}_{i,s} = N_{i,s}(\cdot) - \frac{\chi_{i,s}}{\delta_{i,s}} \quad (20)$$

where $N_{i,s}(\cdot)$ is a continuous function.

Since $f_{\omega i}(\omega_i)$ and $f_{vi}(v_i)$ are uncertain, the FLSs are utilized for approximating them. Assume that

$$f_{vi}(v_i) = G_{i,1}^{*T} \phi_{i,1}(v_i) + \varepsilon_{i,1} \quad (21)$$

$$f_{\omega i}(\omega_i) = G_{i,2}^{*T} \phi_{i,2}(\omega_i) + \varepsilon_{i,2} \quad (22)$$

where $G_{i,1} \in R^n$ and $G_{i,2} \in R^n$ are the ideal weights, $\phi_{i,1} \in R^n$ and $\phi_{i,2} \in R^n$ are the fuzzy basic functions. $\varepsilon_{i,1}$ and $\varepsilon_{i,2}$ are the approximated errors, and they satisfy $|\varepsilon_{i,1}| \leq \varepsilon_{i,1}^*$ and $|\varepsilon_{i,2}| \leq \varepsilon_{i,2}^*$, with $\varepsilon_{i,1}^*$ and $\varepsilon_{i,2}^*$ being unknown constants.

Thus, (18) and (19) can be rewritten as

$$\dot{z}_{iv} = G_{i,1}^{*T} \phi_{i,1}(v_i) + \varepsilon_{i,1} + m_{1i}\tau_{vi} - \dot{h}_{i,1} \quad (23)$$

$$\dot{z}_{i\omega} = G_{i,2}^{*T} \phi_{i,2}(\omega_i) + \varepsilon_{i,2} + m_{2i}\tau_{\omega i} - \dot{h}_{i,2} \quad (24)$$

Consider the log-type barrier Lyapunov function

$$V_{2i} = V_{11i} + V_{12i} + \frac{1}{2}z_{iv}^2 + \frac{1}{2}z_{i\omega}^2 + \frac{1}{2}\tilde{G}_{i,1}^T \tilde{G}_{i,1} + \frac{1}{2}\tilde{G}_{i,2}^T \tilde{G}_{i,2} + \frac{1}{2}\chi_{i,1}^2 + \frac{1}{2}\chi_{i,2}^2 \quad (25)$$

In order to decrease the unnecessary waste of resources resulting from the update of the controller signal, we design the following event-triggered mechanism:

$$\tau_{vi}(t) = \vartheta_{vi}(t_{vik}), t \in [t_{vik}, t_{vi(k+1)}) \quad (26)$$

$$\tau_{\omega i}(t) = \vartheta_{\omega i}(t_{\omega ik}), t \in [t_{\omega ik}, t_{\omega i(k+1)}) \quad (27)$$

$$t_{vi(k+1)} = \inf\{t \in R \mid |z_{vi}(t)| \geq \lambda_{1v}|\tau_{vi}| + \lambda_{2v}\} \quad (28)$$

$$t_{\omega i(k+1)} = \inf\{t \in R \mid |z_{\omega i}(t)| \geq \lambda_{1\omega}|\tau_{\omega i}| + \lambda_{2\omega}\} \quad (29)$$

with $z_{\kappa i}(t) = \tau_{\kappa i}(t) - \vartheta_{\kappa i}(t_{\kappa ik})$ representing the event-triggered error during time interval $[t_{\kappa ik}, t_{\kappa i(k+1)})$, where $\kappa = v, \omega$. $0 \leq \lambda_{1\kappa} \leq 1$ and $\lambda_{2\kappa} > 0$ are constants. $\vartheta_{\kappa i}(t_{\kappa ik})$ are trigger control signals.

The virtual controllers, trigger controllers, adaptive formation controllers and adaptive laws are given as

$$\rho_{vi} = \bar{m}_{iv}^{-1}(-c_{iv}z_{iv} - \hat{G}_{i,1}^T \phi_{i,1}(v_i) + \dot{h}_{i,1} + \frac{z_{id} \cos \varphi_i}{k_{d1}^2 - z_{id}^2} - \frac{1}{2}z_{iv} - (\frac{\bar{m}_{1i}\bar{\lambda}_{iv}\delta_{iv2}}{1 + \lambda_{iv}\delta_{iv1}})^2 z_{iv}) \quad (30)$$

$$\rho_{\omega i} = \bar{m}_{i\omega}^{-1}(-c_{i\omega}z_{i\omega} - \hat{G}_{i,2}^T \phi_{i,2}(\omega_i) + \dot{h}_{i,2} + \frac{z_{i\psi}}{k_{\varphi 1}^2 - z_{i\psi}^2} - \frac{1}{2}z_{i\omega} - (\frac{\bar{m}_{i\omega}\bar{\lambda}_{i\omega}\delta_{i\omega 2}}{1 + \lambda_{i\omega}\delta_{i\omega 1}})^2 z_{i\omega}) \quad (31)$$

$$\vartheta_{vi} = -(1 + \lambda_{iv})(\rho_{vi} \tanh \frac{z_{iv}\rho_{vi}}{\varepsilon} + \bar{\lambda}_{iv} \tanh \frac{z_{iv}\bar{\lambda}_{iv}}{\varepsilon}) \quad (32)$$

$$\vartheta_{\omega i} = -(1 + \lambda_{i\omega})(\rho_{\omega i} \tanh \frac{z_{i\omega}\rho_{\omega i}}{\varepsilon} + \bar{\lambda}_{i\omega} \tanh \frac{z_{i\omega}\bar{\lambda}_{i\omega}}{\varepsilon}) \quad (33)$$

$$\tau_{vi} = \frac{\vartheta_{vi}}{1 + \lambda_{iv}\delta_{iv1}} - \frac{\bar{\lambda}_{iv}\delta_{iv2}}{1 + \lambda_{iv}\delta_{iv1}} \quad (34)$$

$$\tau_{\omega i} = \frac{\vartheta_{\omega i}}{1 + \lambda_{i\omega}\delta_{i\omega 1}} - \frac{\bar{\lambda}_{i\omega}\delta_{i\omega 2}}{1 + \lambda_{i\omega}\delta_{i\omega 1}} \quad (35)$$

$$\dot{\hat{G}}_{i,1} = z_{i,v}^T \phi_{i,1}(v_i) + \sigma_{i,1} \hat{G}_{i,1} \quad (36)$$

$$\dot{\hat{G}}_{i,2} = z_{i,\omega}^T \phi_{i,2}(\omega_i) + \sigma_{i,2} \hat{G}_{i,2} \quad (37)$$

where $c_{i,v} > 0$, $c_{i,\omega} > 0$, $\sigma_{i,1} > 0$, and $\sigma_{i,2} > 0$ are design parameters. $\bar{\lambda}_{iv} \geq \frac{\lambda_{2v}}{1-\lambda_{1v}}$ and $\bar{\lambda}_{i\omega} \geq \frac{\lambda_{2\omega}}{1-\lambda_{1\omega}}$ are design parameters. $\varepsilon > 0$ are design constants.

3.2. Stability Analysis

The properties of the above control scheme are provided in the following theorem.

Theorem 1. For the NMRs (1), under above assumption 1, if the virtual controllers are adopted by (16) and (17), the trigger controllers are adopted by (32) and (33), the actual controller are adopted by (34) and (35), and the parameter updating laws are adopted by (36) and (37), and then the whole fuzzy adaptive ET formation control scheme can ensure the following:

- (1) The multiple underactuated NMRs are stable.
- (2) Each follower NMR can track the leader NMR.
- (3) The Zeno behavior can be excluded.

Proof. Differentiating (14) and (15) yields

$$\dot{V}_{11i} = \frac{z_{id}\dot{z}_{id}}{k_{d1}^2 - z_{id}^2} = \frac{z_{id}}{k_{d1}^2 - z_{id}^2} [-(z_{iv} + \alpha_{vi}) \cos \varphi_i + v_{i-1} \cos \gamma_i] \quad (38)$$

$$\dot{V}_{12i} = \frac{z_{i\psi}\dot{z}_{i\psi}}{k_{\varphi 1}^2 - z_{i\psi}^2} = \frac{z_{i\psi}}{k_{\varphi 1}^2 - z_{i\psi}^2} [-(z_{i\omega} + \alpha_{\omega i}) + \frac{v_i}{d_i} \sin \varphi_i - \frac{v_{i-1}}{d_i} \sin \gamma_i] \quad (39)$$

In the intervals $|z_{i\partial}| \leq k_{\partial 1}, \partial = d, \varphi$, we have

$$\log \frac{k_{\partial 1}^2}{k_{\partial 1}^2 - z_{i\partial}^2} \leq \frac{z_{i\partial}^2}{k_{\partial 1}^2 - z_{i\partial}^2} \quad (40)$$

Substituting the virtual control laws (16), (17), and (40) into (38) and (39), we obtain

$$\dot{V}_{11i} = -\frac{z_{id}z_{iv}\cos\varphi_i}{k_{d1}^2 - z_{id}^2} - \frac{k_{d1}z_{id}^2}{k_{d1}^2 - z_{id}^2} \leq -k_{d1}V_{11i} - \frac{z_{id}z_{iv}\cos\varphi_i}{k_{d1}^2 - e_{di}^2} \quad (41)$$

$$\dot{V}_{12i} = -\frac{z_{i\psi}z_{i\omega}}{k_{\varphi 1}^2 - z_{i\psi}^2} - \frac{k_{\varphi 1}z_{i\psi}^2}{k_{\varphi 1}^2 - z_{i\psi}^2} \leq -k_{\varphi 1}V_{12i} - \frac{z_{i\psi}z_{i\omega}}{k_{\varphi 1}^2 - z_{i\psi}^2} \quad (42)$$

Differentiating (25) yields

$$\begin{aligned} \dot{V}_{2i} &= \dot{V}_{11i} + \dot{V}_{12i} + z_{iv}\dot{z}_{iv} + z_{i\omega}\dot{z}_{i\omega} - \tilde{G}_{i,1}^T \dot{\hat{G}}_{i,1} - \tilde{G}_{i,2}^T \dot{\hat{G}}_{i,2} + \chi_{i,1}\dot{\chi}_{i,1} + \chi_{i,2}\dot{\chi}_{i,2} \\ &= \dot{V}_{11i} + \dot{V}_{12i} + z_{iv}(\tilde{G}_{i,1}^T \phi_{i,1}(v_i) + \varepsilon_{i,1} + m_{1i}\tau_{vi} - \dot{h}_{i,1}) \\ &\quad + z_{i\omega}(\tilde{G}_{i,2}^T \phi_{i,2}(\omega_i) + \varepsilon_{i,2} + m_{2i}\tau_{\omega i} - \dot{h}_{i,2}) \\ &\quad + \chi_{i,1}(N_{i,1}(\cdot) - \frac{\chi_{i,1}}{\delta_{i,1}}) + \chi_{i,2}(N_{i,2}(\cdot) - \frac{\chi_{i,2}}{\delta_{i,2}}) \\ &\quad + \tilde{G}_{i,1}^T(z_{iv}\phi_{i,1}(v_i) - \dot{\hat{G}}_{i,1}) + \tilde{G}_{i,2}^T(z_{i\omega}\phi_{i,2}(\omega_i) - \dot{\hat{G}}_{i,2}) \end{aligned} \quad (43)$$

According to [27], define the set $\Omega_{i,s} = \{ \sum_{i=1}^N (z_{ik}^2 + \tilde{G}_{i,s}^T \tilde{G}_{i,s} + \chi_{i,s}^2) \leq 2\xi \}$, where $\kappa = v, \omega, s = 1, 2$, $\xi > 0$ is a constant and $V(0) < \xi$. Since Ω_i is compact in $\mathbb{R}^{\dim(\Omega_i)}$, there exists a constant $\bar{N}_{i,s}$ with $|N_{i,s}(\cdot)| \leq \bar{N}_{i,s}$ on $\Omega_{i,s}$. ($\dim(\Omega_i)$ denotes the dimension of Ω_i .)

Invoke the fact that $0 \leq |\theta| - \theta \tanh(\theta/\ell) \leq 0.2785\ell$. According to (43), and event-triggered mechanism (26)–(29), we have

$$\begin{aligned} \bar{m}_{1i}z_{iv}\tau_{vi} - \bar{m}_{1i}z_{iv}\rho_{vi} &= \bar{m}_{1i}z_{iv}(\frac{\theta_{vi}}{1+\lambda_{iv}\delta_{iv1}} - \frac{\bar{\lambda}_{iv}\delta_{iv2}}{1+\lambda_{iv}\delta_{iv1}} - \rho_{vi}) \\ &\leq \frac{\bar{m}_{1i}(1+\lambda_{v1})}{1+\lambda_{iv}\delta_{iv1}}(|z_{iv}\rho_{vi}| - z_{iv}\rho_{vi}\tanh(\frac{z_{iv}\rho_{vi}}{\varepsilon})) + |z_{iv}\bar{\lambda}_{vi}| \\ &\quad - z_{iv}\bar{\lambda}_{vi}\tanh(\frac{z_{iv}\bar{\lambda}_{vi}}{\varepsilon}) - \frac{\bar{m}_{1i}z_{iv}\bar{\lambda}_{iv}\delta_{iv2}}{1+\lambda_{iv}\delta_{iv1}} \\ &\leq 0.557\ell \frac{\bar{m}_{1i}(1+\lambda_{v1})}{1+\lambda_{iv}\delta_{iv1}} + (\frac{\bar{m}_{1i}z_{iv}\bar{\lambda}_{iv}\delta_{iv2}}{1+\lambda_{iv}\delta_{iv1}})^2 + \frac{1}{4} \end{aligned} \quad (44)$$

$$\begin{aligned} \bar{m}_{1i}z_{i\omega}\tau_{\omega i} - \bar{m}_{1i}z_{i\omega}\rho_{\omega i} &= \bar{m}_{1i}z_{i\omega}(\frac{\theta_{\omega i}}{1+\lambda_{i\omega}\delta_{i\omega 1}} - \frac{\bar{\lambda}_{i\omega}\delta_{i\omega 2}}{1+\lambda_{i\omega}\delta_{i\omega 1}} - \rho_{\omega i}) \\ &\leq \frac{\bar{m}_{1i}(1+\lambda_{\omega 1})}{1+\lambda_{i\omega}\delta_{i\omega 1}}(|z_{i\omega}\rho_{\omega i}| - z_{i\omega}\rho_{\omega i}\tanh(\frac{z_{i\omega}\rho_{\omega i}}{\varepsilon})) + |z_{i\omega}\bar{\lambda}_{\omega i}| \\ &\quad - z_{i\omega}\bar{\lambda}_{\omega i}\tanh(\frac{z_{i\omega}\bar{\lambda}_{\omega i}}{\varepsilon}) - \frac{\bar{m}_{1i}z_{i\omega}\bar{\lambda}_{i\omega}\delta_{i\omega 2}}{1+\lambda_{i\omega}\delta_{i\omega 1}} \\ &\leq 0.557\ell \frac{\bar{m}_{1i}(1+\lambda_{\omega 1})}{1+\lambda_{i\omega}\delta_{i\omega 1}} + \bar{m}_{1i}(\frac{z_{i\omega}\bar{\lambda}_{i\omega}\delta_{i\omega 2}}{1+\lambda_{i\omega}\delta_{i\omega 1}})^2 + \frac{1}{4} \end{aligned} \quad (45)$$

According to Young's inequality $a^2 + b^2 \leq 2ab$, it is available

$$\begin{aligned} &z_{iv}\varepsilon_{i,1} + z_{i\omega}\varepsilon_{i,2} + \chi_{i,1}(N_{i,1}(\cdot) - \frac{\chi_{i,1}}{\delta_{i,1}}) + \chi_{i,2}(N_{i,2}(\cdot) - \frac{\chi_{i,2}}{\delta_{i,2}}) \\ &\leq \frac{1}{2}z_{iv}^2 + \frac{1}{2}z_{i\omega}^2 + \frac{1}{2}\varepsilon_{i,1}^2 + \frac{1}{2}\varepsilon_{i,2}^2 + (\frac{1}{2} - \frac{1}{\delta_{i,1}})\chi_{i,1}^2 + \frac{1}{2}N_{i,1}^2 \\ &\quad + (\frac{1}{2} - \frac{1}{\delta_{i,2}})\chi_{i,2}^2 + \frac{1}{2}N_{i,2}^2 \end{aligned} \quad (46)$$

$$\tilde{G}_{i,s}^T \hat{G}_{i,s} \leq \frac{1}{2}(\|G_{i,s}^*\|^2 - \|\tilde{G}_{i,s}\|^2) \quad (47)$$

where p_1 and p_2 are positive design parameters.

From (30)–(37) and (43)–(47), we have

$$\begin{aligned} \dot{V}_2 &\leq \left(\frac{1}{2} + c_{i,d}\right)z_{id}^2 + \left(\frac{1}{2} - c_{i,\psi}\right)z_{i\psi}^2 + (1 - c_{i,v})z_{iv}^2 + (1 - c_{i,\omega})z_{i\omega}^2 \\ &\quad - \frac{\sigma_{i,1}}{2}\|\tilde{G}_{i,1}\|^2 - \frac{\sigma_{i,2}}{2}\|\tilde{G}_{i,2}\|^2 + \left(\frac{1}{2} - \frac{1}{\delta_{i,1}}\right)\chi_{i,1}^2 + \left(\frac{1}{2} - \frac{1}{\delta_{i,2}}\right)\chi_{i,2}^2 \\ &\quad + \frac{1}{2}N_{i,1}^2 + \frac{1}{2}N_{i,2}^2 + \frac{1}{2}\varepsilon_{i,1}^2 + \frac{1}{2}\varepsilon_{i,2}^2 + \frac{\sigma_{i,1}}{2}\|G_{i,1}^*\|^2 + \frac{\sigma_{i,2}}{2}\|G_{i,2}^*\|^2 \\ &\quad + 0.557\ell\frac{\bar{m}_{1i}(1+\lambda_{v1})}{1+\lambda_{iv}\delta_{iv1}} + 0.557\ell\frac{\bar{m}_{1i}(1+\lambda_{\omega1})}{1+\lambda_{i\omega}\delta_{i\omega1}} + \frac{1}{2} \end{aligned} \quad (48)$$

Let

$$\begin{aligned} \bar{\Pi} &= \frac{1}{2}N_{i,1}^2 + \frac{1}{2}N_{i,2}^2 + \frac{1}{2}\varepsilon_{i,1}^2 + \frac{1}{2}\varepsilon_{i,2}^2 + \frac{\sigma_{i,1}}{2}\|G_{i,1}^*\|^2 + \frac{\sigma_{i,2}}{2}\|G_{i,2}^*\|^2 \\ &\quad + 0.557\ell\frac{\bar{m}_{1i}(1+\lambda_{v1})}{1+\lambda_{iv}\delta_{iv1}} + 0.557\ell\frac{\bar{m}_{1i}(1+\lambda_{\omega1})}{1+\lambda_{i\omega}\delta_{i\omega1}} + \frac{1}{2} \end{aligned} \quad (49)$$

Denote $V = \sum_{i=1}^N V_{i,2}$; according to (48) and (49), it yields that

$$\dot{V} \leq -\bar{\psi}V + \bar{\Pi} \quad (50)$$

where $\bar{\psi} = \min\left\{\frac{1}{2} + c_{i,d}, \frac{1}{2} - c_{i,\psi}, 1 - c_{i,v}, 1 - c_{i,\omega}, \frac{\sigma_{i,s}}{2}, \frac{1}{2} - \frac{1}{\delta_{i,s}}\right\}$.

Integrating (50) over $[0, t]$, it shows

$$0 \leq V(0)e^{-\bar{\psi}t} + \frac{\bar{\Pi}}{\bar{\psi}} \quad (51)$$

From (51), it is shown that all the system's closed-loop signals are bounded. Then, the fuzzy adaptive ET formation control errors based on $(2V(0)e^{-\bar{\psi}t} + \bar{\Pi}/\bar{\psi})^{1/2}$ for the union upper bound are given, and it is concluded that the event-triggered formation control error converges to the neighborhood of the origin. $\square$

3.3. The Exclusion of Zeno Phenomenon

In this part, the main content is to explain that there is no Zeno phenomenon in the fuzzy adaptive ET formation control method.

According to [17], the event-triggered (ET) mechanism (26)–(29) dynamically determines when control signals are updated based on predefined thresholds. To ensure practicality, we must prove that the time between consecutive triggering events $\tau = t_{i(k+1)} - t_{ik}$ is strictly positive and bounded below by a minimum interval $\tau > 0$, thereby excluding Zeno behavior.

According to the error $z_{i\tau m}(t) = \tau_{im}(t) - \alpha_{im\tau}(t_{ik})$, one has

$$\frac{d}{dt}|z_{i\tau m}| = \dot{z}_{i\tau m} \text{sign}(z_{i\tau m}) < |\dot{\tau}_{im}(t)|, \quad m = v, \omega$$

All the signals are bounded in the NMRs and the actuators signals τ_{iv} and $\tau_{i\omega}$ are differentiable. Then, existing ∂ can ensure $|\dot{\tau}_{im}(t)| \leq \partial$ is valid, where ∂ is a constant.

By integrating, one has

$$\int_{t_{ik}}^{t_{i(k+1)}} |\dot{z}_{i\tau m}| dt \leq \int_{t_{ik}}^{t_{i(k+1)}} \partial dt, \quad m = v, \omega$$

It means that $t_{i(k+1)} - t_{ik} \geq (\lambda_{\tau 1}|\tau_{im}(t)| + \lambda_{\tau 2})/\sigma = \gamma > 0$ for $\forall t \in [t_{ik}, t_{i(k+1)})$. So there is no Zeno phenomenon in the fuzzy adaptive ET formation control method.

4. Simulation Studies

This section verifies the feasibility of the proposed fuzzy adaptive ET formation control method. We choose the robot parameters and give the values in Table 2 from the work in [28].

Table 2. Robot parameters.

Parameter	Value	Unit	Parameter	Value	Unit
a_i	0.75	m	I_{ib}	15.625	$\text{kg} \cdot \text{m}^2$
c_i	0.3	m	$I_{i\omega}$	0.005	$\text{kg} \cdot \text{m}^2$
r_i	0.15	m	I_{im}	0.0025	$\text{kg} \cdot \text{m}^2$
m_{ib}	30	kg	d_{i11}	5	$\text{kg} \cdot \text{m}^2/\text{s}$
m_{io}	1	kg	d_{i12}	5	$\text{kg} \cdot \text{m}^2/\text{s}$

4.1. Simulation Verification

The leader's signal is $x_0 = y_0 = \sin t$. The robots' initial positions are given as $p_0 = [0, 0, 0]^T$, $p_1 = [0.1, 0, \pi/9]^T$, $p_2 = [-0.1, 0, \pi/9]^T$, $p_3 = [0.5, 1, \pi/4]^T$, $p_4 = [0.1, 0, \pi/4]^T$.

The initial function is as follows:

$$f_{vi} = \frac{2b_i c_i \omega_i^2 - v_i(d_{11i} + d_{22i}) - b_i \omega_i(d_{11i} - d_{22i})}{2(m_{11i} + m_{12i})}$$

$$f_{\omega i} = -\frac{2c_i v_i \omega_i + v_i(d_{11i} - d_{22i}) + b_i \omega_i(d_{11i} + d_{22i})}{2c_i(m_{11i} - m_{12i})}$$

To estimate the unknown nonlinear functions f_{vi} and $f_{\omega i}$, assume that the universes of discourse for variables κ_i are $[-4, 4]$. Define If-Then fuzzy rules as

L^j : if κ_i is $f_{\kappa i}$, then y_i is $P(\kappa = v, \omega; i = 1, 2, 3, 4; j = 1, 2, 3, 4, 5)$.

The fuzzy membership functions are given by $\mu_{f_{\kappa 1}} = 1/(1 + e^{[5(3+\kappa_1)/2]}), \mu_{f_{\kappa 2}} = e^{[-(2+\kappa_2)^2/2]}, \mu_{f_{\kappa 3}} = e^{[-(\kappa_3)^2/2]}, \mu_{f_{\kappa 4}} = e^{[-(-2+\kappa_4)^2/2]}, \mu_{f_{\kappa 5}} = 1/(1 + e^{[-5(-3+\kappa_5)/2]}), \kappa = v, \omega$. Consequently, the fuzzy membership functions are presented in Figure 2. The FLS $\hat{f}_{vi}(v_i) = G_{i,1}^T \Psi_{i,1}(v_i)$ and $\hat{f}_{\omega i}(\omega_i) = G_{i,2}^T \Psi_{i,2}(\omega_i)$ are employed to estimate $f_{vi}(v_i)$ and $f_{\omega i}(\omega_i)$.

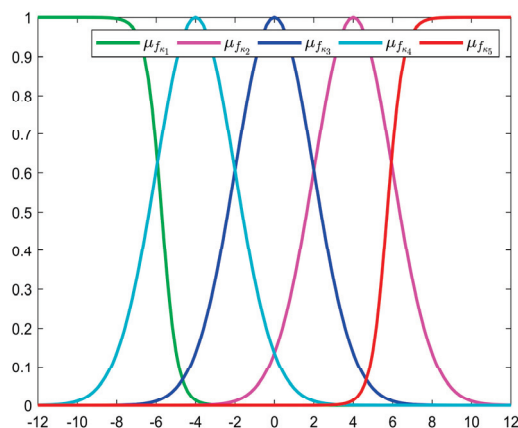


Figure 2. The fuzzy membership functions.

Remark 3. It is also worth pointing out that according to Lemma 1, an FLS can approximate an unknown nonlinear function when the number of If-Then rules is larger, and for every input variable in the FLS, its fuzzy membership functions are chosen as Gaussian types $\mu_{f_i}(\kappa_i) = \exp[-\frac{(\kappa_i - \mu_i)^2}{\gamma_i^2}]$, $i = 1, \dots, n$, where the center points μ_i belong to the universe of

discourse of the variable, and the width $\gamma_i > 0$. Moreover, the defined fuzzy membership functions should satisfy the following conditions.

- (1) Each pair of $(\mu_{f_k}(\kappa_i), \mu_{f_j}(\kappa_i))$ is required to be overlapped.
- (2) The whole sequence of fuzzy membership functions $\mu_{f_1}(\kappa_i), \dots, \mu_{f_n}(\kappa_i)$ is required to cover the universe of discourse of the variable x (the universe of discourse usually chosen as a symmetrical interval about zero).

The distance and angle of ensuring connectivity, collision avoidance, and the idealize are chosen as $d_{con,i} = 5$ m, $d_{col,i} = 3$ m, $\varphi_{con,i} = 1$ rad, $d_{des,i} = 4$ m, and $\varphi_{des,i} = 0$ rad, respectively. The parameters of the virtual controllers in expressions (16) and (17), along with the adaptive laws stated in (49) and (50), are given as $k_{d1} = k_{\varphi1} = 9$ and $\sigma_{i,1} = \sigma_{i,2} = 1$ for robots 1–4, $i = 1, 2, 3, 4$. The parameters of trigger controllers (32) and (33) are given as $\lambda_{iv} = 0.01$, $\bar{\lambda}_{iv} = 0.05$, and $\varepsilon = 0.9$.

The simulation results are exhibited in Figures 3–8. Figure 3 depicts the curves of NMRs. From Figure 3, it can be seen that Robot 1, Robot 2, Robot 3, and Robot 4 are stable and able to track Robot 0 to maintain the desired formation pattern. In the whole formation control process, there will be no collision between the robots, and the formation can be maintained within the scope of constraints. Figures 4 and 5 show the tracking error of the four robots in terms of distance and angle, respectively. It can be observed from Figures 4 and 5 that by adopting the formulated formation control approach, the tracking error of each NMR converges to a small neighborhood around zero even under full-state constraints. Figure 6 exhibits the trajectories of controllers. It can be observed from Figures 4 and 5 that by adopting the formulated formation control approach, the tracking error of each NMR converges to a small neighborhood around zero even under full-state constraints. Figures 7 and 8 display the triggering sequences of the four NMRs. Table 3 shows the specific number of triggers and the proportion of resource savings. It can be seen from them that the ET mechanism proposed in this study is capable of decreasing the unnecessary waste of resources resulting from the update of the controller signal. The above shows that the fuzzy formation control scheme presented in this article can achieve the control objective.

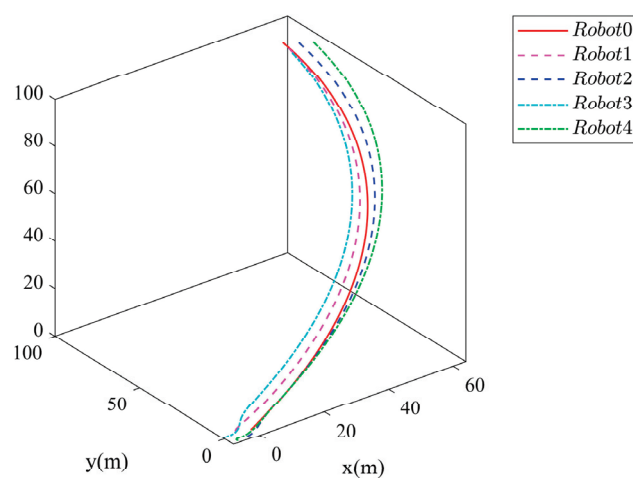


Figure 3. The trajectories of four NMRs.

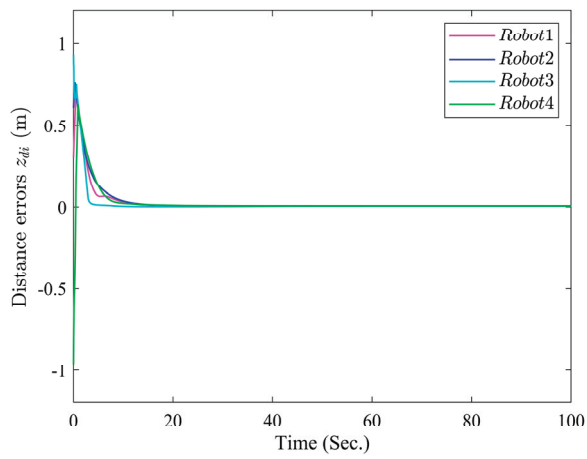


Figure 4. Curves of tracking errors z_{di} (m).

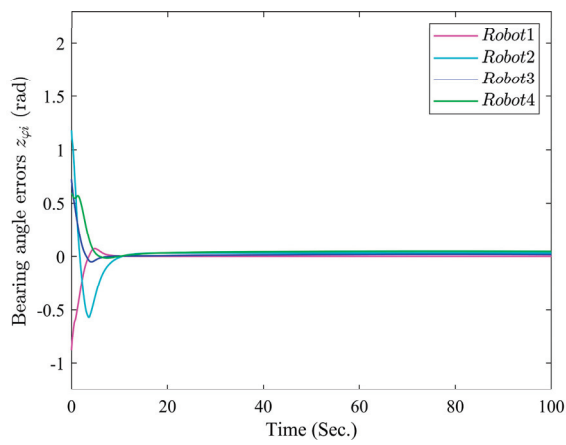


Figure 5. Curves of formation tracking errors $z_{\varphi i}$ (rad).

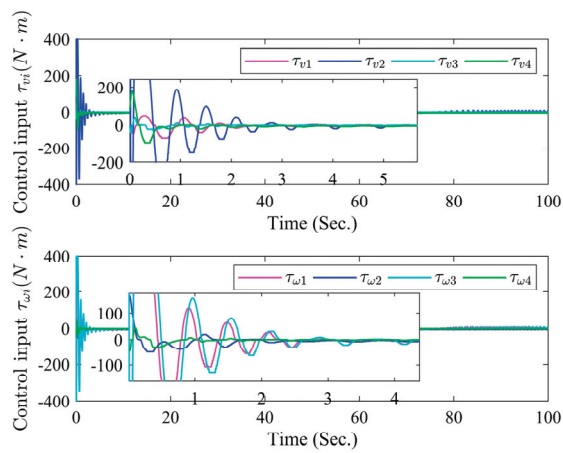


Figure 6. Curves of control inputs τ_{vi} and $\tau_{\omega i}$ (N · m).

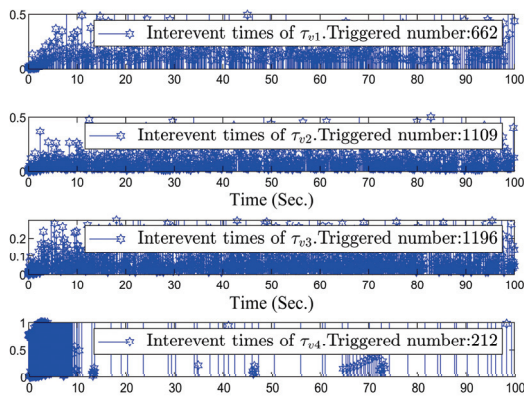


Figure 7. Triggering sequences of τ_{v1} , τ_{v2} , τ_{v3} , and τ_{v4} .

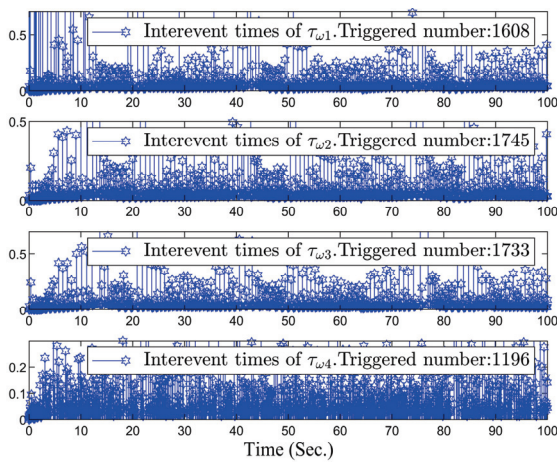


Figure 8. Triggering sequences of $\tau_{\omega1}$, $\tau_{\omega2}$, $\tau_{\omega3}$, and $\tau_{\omega4}$.

Table 3. Triggered times.

	Triggered Times of τ_{vi} in the Paper	Triggered Times of $\tau_{\omega i}$ in the Paper	Event Triggering Mechanisms Are Not Considered	Resource Saving Ratio
Robot 1	662	1608	20,000	88.65%
Robot 1	1109	1745	20,000	85.75%
Robot 1	1196	1733	20,000	85.36%
Robot 1	212	1196	20,000	92.96%
Total times	3179	6282	80,000	88.17%

4.2. Discussions

Case 1. The Effect of Design Parameters.

To explain the effect of the design parameters in virtual controllers (16) and (17) and controllers (34) and (35), we select different design parameters for simulation. For ease of comparison, define the sum of the absolute values of the formation tracking errors as $\bar{z}_{di} = \sum_{i=1}^4 |z_{di}|$. Then, the responses to the tracking error $\bar{z}_{di}$ under different design parameters are shown in Figure 9.

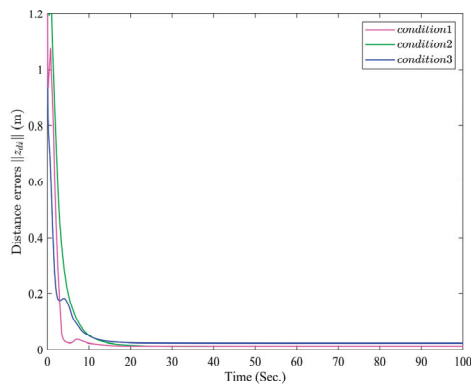


Figure 9. The responses of the tracking error $\bar{z}_{di}$ (m).

From Figure 9, we can see that selecting different parameters will affect the control performance of the system. Therefore, the formation tracking errors can be reduced by increasing the main design parameter. However, increasing the design parameters results in larger control energy. In practice, a tradeoff between improved control performance and the energy cost of control actions must be carefully considered.

Case 2. The Effect of external interference.

In the simulation, we considered the influence of time-varying external interference on each robot. Figure 9 shows that the distance formation error of the robot will become larger under external interference. Similarly, Figures 10 and 11 show that the robot's formation errors will also be affected by external interference. The simulation results show that the time-varying external interference affects the control performance of the system.

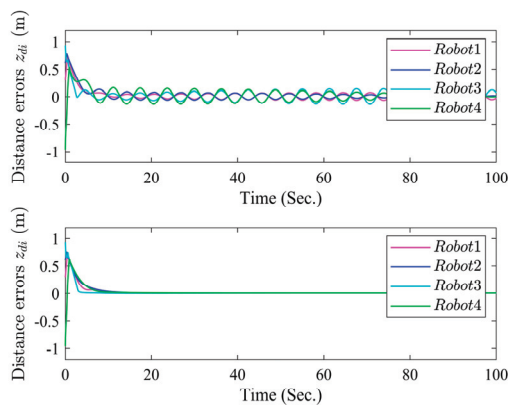


Figure 10. Curves of tracking errors z_{di} (m).

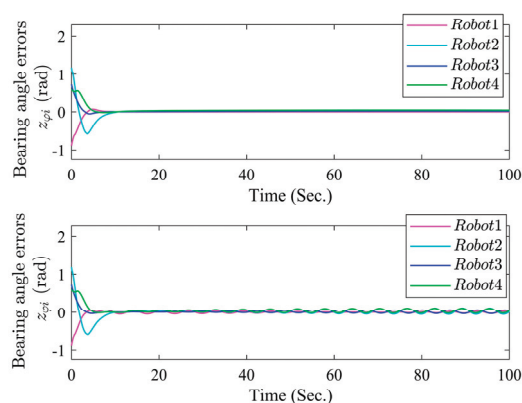


Figure 11. Curves of formation tracking errors z_{qi} (rad).

The adaptive fuzzy formation control method proposed in this paper is robust and can deal with some small external disturbances, but the formation error will be larger when there are external disturbances or when there are external disturbances.

5. Conclusions

We proposed a fuzzy adaptive ET formation control design method for NMRs with full-state constants. The event-triggered mechanism can effectively reduce the number of unnecessary updates of control signals. Then, a fuzzy adaptive ET formation control strategy for NMRs was presented via backstepping control technology. Our results demonstrate that all signals of the controlled NMRs are SGUUB, and the tracking errors even converge to a small neighborhood around zero. It should be noted that the proposed fuzzy ET formation control algorithm cannot handle the consensus control problem when there are communication delays and the system is under sensor or other network attacks. It is also uncertain whether such a formation control method is applicable to heterogeneous multi-agent robots. Therefore, extending the results of this study to the case of communication time delays is a meaningful future research direction.

Author Contributions: Conceptualization, K.W. and J.L.; methodology, K.W.; software, K.W.; validation, K.W. and H.Z.; writing—original draft preparation, K.W.; writing—review and editing, K.W., J.L. and H.Z.; visualization, J.L. and H.Z.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: (a) Anhui Intelligent Mining Technology and Equipment Engineering Research Center 2023 Open Funds (AIMTEERC202304). (b) Xinjiang Key Laboratory of Intelligent Exploit and Control of Open-pit Mine (XJQY2007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no competing interests.

References

1. An, L.; Yang, G.H. Collisions-free distributed optimal coordination for multiple Euler-Lagrangian systems. *IEEE Trans. Autom. Control* **2022**, *67*, 460–467.
2. Zhai, J.; Xu, G. A novel non-singular terminal sliding mode trajectory tracking control for robotic manipulators. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 391–395.
3. Hu, Z.; Jin, X. Formation control for an UAV team with environment-aware dynamic constraints. *IEEE Trans. Intell. Veh.* **2023**, *9*, 1465–1480. [CrossRef]
4. Peng, Z.; Wang, J.; Wang, D.; Han, Q. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Trans. Ind. Inform.* **2020**, *17*, 732–745.
5. Wang, Y.; Song, Y. Leader-following control of high-order multiagent systems under directed graphs: Pre-specified finite time approach. *Automatica* **2018**, *87*, 113–120.
6. Lee, G.; Chwa, D. Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. *Intell. Serv. Robot.* **2016**, *11*, 127–138.
7. Benzerrouk, A.; Adouane, L.; Martinet, P. Stable navigation in formation for a multi-robot system based on a constrained virtual structure. *Robot. Auton. Syst.* **2014**, *62*, 1806–1815.
8. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans. Ind. Inform.* **2013**, *9*, 427–438.
9. Wang, J.; Dong, H.; Chen, F.; Vu, M.T.; Shakibjoo, A.D.; Mohammadzadeh, A. Formation Control of Non-Holonomic Mobile Robots: Predictive Data-Driven Fuzzy Compensator. *Mathematics* **2023**, *11*, 1804. [CrossRef]

10. Li, K.; Fang, X.; Zhao, K.; Song, Y. Angle-constrained adaptive formation control with prescribed performance for multiple nonholonomic mobile robots. *IEEE Trans. Ind. Electron.* **2024**, 1–10. [CrossRef]
11. Romero, J.; Nuño, E.; Restrepo, E.; Sarras, I. Global consensus-based formation control of nonholonomic mobile robots with time-varying delays and without velocity measurements. *IEEE Trans. Autom. Control* **2024**, 69, 355–362. [CrossRef]
12. Shao, X.; Li, S.; Zhang, W.; Wu, E.Q. Distance-based elliptical circumnavigation control for non-holonomic robots with event-triggered unknown system dynamics estimators. *IEEE Trans. Intell. Transp. Syst.* **2023**, 24, 3986–3998. [CrossRef]
13. Wang, W.; Huang, J.; Wen, C. Distributed adaptive control for consensus tracking with application to formation control of nonholonomic mobile robots. *Automatica* **2014**, 50, 1254–1263. [CrossRef]
14. Min, X.; Baldi, S.; Yu, W. Funnell-based asymptotic control of leader-follower nonholonomic robots subject to formation constraints. *IEEE Trans. Control Netw. Syst.* **2023**, 10, 1313–1325. [CrossRef]
15. Yoo, S.J.; Park, B.S. Connectivity preservation and collision avoidance in networked nonholonomic multi-robot formation systems: Unified error transformation strategy. *Automatica* **2019**, 103, 274–281. [CrossRef]
16. Dai, S.L.; Lu, K.; Jin, X. Fixed-time formation control of unicycle-type mobile robots with visibility and performance constraints. *IEEE Trans. Ind. Electron.* **2021**, 68, 12615–12625. [CrossRef]
17. Li, Y.; Dong, S.; Li, K. Fixed-time command filter fuzzy adaptive formation control for nonholonomic multirobot systems with unknown dead-zones. *IEEE Trans. Intell. Transp. Syst.* **2024**, 25, 17305–17316. [CrossRef]
18. Li, K.; Li, Y. Adaptive neural network finite-time dynamic surface control for nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, 32, 5688–5697. [CrossRef]
19. Tong, S.; Li, K.; Li, Y. Robust fuzzy adaptive finite-time control for high-order nonlinear systems with unmodeled dynamics. *IEEE Trans. Fuzzy Syst.* **2021**, 29, 1576–1589.
20. Park, B.S.; Yoo, S.J. Connectivity-maintaining and collision-avoiding performance function approach for robust leader-follower formation control of multiple uncertain underactuated surface vessels. *Automatica* **2021**, 127, 109501.
21. Dai, S.L.; He, S.; Chen, X.; Jin, X. Adaptive leader-follower formation control of nonholonomic mobile robots with prescribed transient and steady-state performance. *IEEE Trans. Ind. Inform.* **2020**, 16, 3662–3671. [CrossRef]
22. Kuang, C.; Li, L.; Geng, Q. Formation Control and Collision Avoidance of Nonholonomic Mobile Robots. In Proceedings of the 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), Tianjin, China, 29–31 October 2021; pp. 1–6.
23. Wang, D.; Pan, S.; Zhou, J.; Pan, Q.; Miao, Z.; Yang, J. Event-triggered integral formation controller for networked nonholonomic mobile robots: Theory and experiment. *IEEE Trans. Intell. Transp. Syst.* **2023**, 24, 14620–14632. [CrossRef]
24. Xu, Y.; Wang, C.; Cai, X.; Li, Y.; Xu, L. Output-feedback formation tracking control of networked nonholonomic multi-robots with connectivity preservation and collision avoidance. *Neurocomputing* **2020**, 414, 267–277. [CrossRef]
25. Xiao, H.; Philip-Chen, C.; Lai, G.; Yu, D.; Zhang, Y. Integrated nonholonomic multi-robot consensus tracking formation using neural-network-optimized distributed model predictive control strategy. *Neurocomputing* **2023**, 518, 282–293. [CrossRef]
26. Wang, L. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1994.
27. Li, K.; Li, Y. Adaptive NN optimal consensus fault-tolerant control for stochastic nonlinear multiagent systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 34, 947–957. [CrossRef]
28. Yang, J.; Xiao, F.; TW, T.C. Event-triggered formation tracking control of nonholonomic mobile robots without velocity measurements. *Automatica* **2020**, 112, 108671. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Three-Dimensional Path-Following with Articulated 6DoF Robot and ToF Sensors

Tymon F. Wawrzyniak, Ignacy D. Orłowski and Marek A. Galewski *

Faculty of Mechanical Engineering and Ship Technology, Gdansk University of Technology, G. Narutowicza 11/12, 80-233 Gdansk, Poland; s190833@student.pg.edu.pl (T.F.W.); s184483@student.pg.edu.pl (I.D.O.)

* Correspondence: marek.galewski@pg.edu.pl

Abstract: This paper presents an algorithm for 3D path-following using an articulated 6-Degree-of-Freedom (DoF) robot as well as experimental verification of the proposed approach. This research extends the classic line-following concept, typically applied in 2D spaces, into a 3D space. This is achieved by equipping a standard industrial robot with a path detection tool featuring six low-cost Time-of-Flight (ToF) sensors. The primary objective is to enable the robot to follow a physically existing path defined in 3D space. The developed algorithm allows for step-by-step detection of the path's orientation and calculation of consecutive positions and orientations of the detection tool that are necessary for the robot arm to follow the path. Experimental tests conducted using a Nachi MZ04D robot demonstrated the reliability and effectiveness of the proposed solution.

Keywords: 3D path-following; articulated robot; robot control

1. Introduction

One of the basic tasks in robotics is to lead the robot arm (for example, the arm of an industrial robot) or the robot itself (for example, in mobile robotics) from the start point to the target point without collisions. This involves making strategic decisions and defining a sequence of valid robot configurations (including position, orientation, and speed) that constitute the obstacle-free path leading to the desired destination and taking environmental constraints into account [1]. Planning of the route can be global or local. In global path planning, all spaces where obstacles exist are predetermined so the obstacle-free route can be calculated in advance [2,3]. When the environment is unknown a priori, or it is not static, local path planning is needed to dynamically respond to the environment changes. Both approaches may also be combined, for example, by providing the reference path, calculated by the path planner first, and then, as the robot moves, by correcting it using local planning as the response to changes in the environment detected by the robot's sensors [4]. Recent studies in path planning, advanced inverse kinematics (IK), and sensor fusion present significant progress, addressing challenges in efficiency, adaptability, and precision [3]. Efficient and precise navigation often needs advanced sensors like IMUs (Inertial Measurement Units), LiDARs, radars, mono and stereo vision systems, etc. Data from various sensors are usually combined (which is called sensor fusion) to provide complex information about the state of the robot and the environment [3]. Then, advanced algorithms for calculating inverse kinematics task, path planning, obstacle detection, and avoidance are applied to allow the robot to reach the desired target position. When the robot moves, it usually follows the path calculated by some path planning algorithm;

however, this path usually does not exist physically in the environment. It is “virtual”, and various control techniques are applied to minimise the tracking error and, of course, to perform the task collision-free [5]. In particular, Artificial Intelligence (AI) methods are currently widely explored and applied with numerous successes for data processing and path planning [3,6]. However, despite undoubted advantages and often excellent results, there are also drawbacks and limitations. For example, they usually require preparing and tuning the AI model, which is not an easy task, and gathering a large amount of training data. Thus, the training process, particularly of deep learning algorithms, is often time-consuming, requires a high level of AI expert knowledge, demands high computational power, and consumes a lot of energy [3,7]. The additional drawback of some advanced methods are also the costs of some types of sensors, for example, 3D cameras and long-range LiDARs. This leaves a space for exploring simpler or cheaper solutions [8], especially when movement accuracy is not of the first importance.

One of the simplest variants for performing the task of moving a robot from the start point to the target position is to move it along a line that is physically defined in the environment. In such a case, and if there is only one possible path to the target (i.e., the path does not split), only local planning is needed, and it is sufficient to just move the robot along the path. This approach is commonly known as “line-following” and is usually applied in mobile robotics. Typically, the line is drawn on the surface where the robot moves. Thanks to sensors, the robot controller detects if the robot pulls off the line to the left or to the right and corrects the robot movement by turning the robot to the direction of the sensor that detected the line, to keep the robot back over the line (Figure 1). Such an approach focuses on correcting the local robot position error relative to the path. In this scenario, although the robot moves in a 2-dimensional (2D) space, the problem is 1-dimensional.

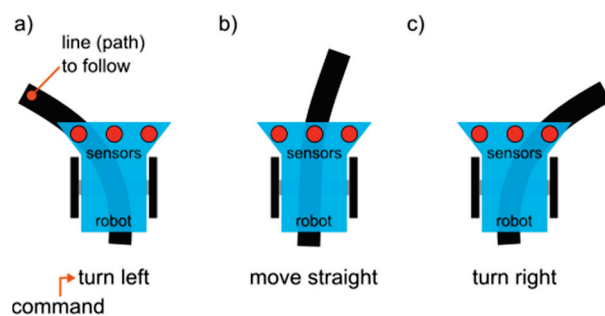


Figure 1. Control principles for typical line-following, 2-wheeled robot showing situations resulting in (a) turn left, (b) move straight, (c) turn right commands.

Due to the very simple control rules, this task is often presented as one of the first during robotic classes for students or as the introduction to mobile robotics for children. There are various types of line-following robots, starting from toys, through various enthusiast-level robots, advanced “racing” robots for line-following competitions [9–11], up to professional, industry-grade constructions. For example, some of the AGVs (Automated Guided Vehicles) in factories and logistic centres also utilise line-following principles in their control systems [12]. The line to follow might be a visual line painted or embedded (e.g., electrical or magnetic wire) in the floor or ceiling. Line detection and following also play an important role in autonomous driving systems applied in road cars [13,14]. In order to detect the line, various types of sensors might be used, for example, light reflection sensors, infrared sensors, magnetic sensors, video cameras, and others. Even though there are already plenty of line-following robots on the market, there are still new

constructions [15,16] and new control algorithms developed: for example, ones based on fuzzy sets [17], the dynamic PID algorithm [18], AI swarm algorithms [19], vision systems combined with infrared sensors [20], and others.

Although most of the line-following robots are ground vehicles that operate in a 2D space, there are also some implementations of this principle for industrial robots and for aerial robots. For example, in [21] the artificial Double Deep Q-Network was applied for performing the path-following task using an articulated robot. The network was trained to follow a path drawn on a flat surface and observed locally by a camera mounted on the robot arm. The achieved path tracking error was below 1 mm. In the paper [22], a system for extracting the features of and tracking a weld joint was presented. It utilised a laser scanner to detect the path defined by the contact line of two metal elements to weld. However, the system was tested only on a straight, flat (i.e., defined in the 2D space) path. An example of the application of line-following principles in aerial robotics is robotic competitions [23], where a robot equipped with a camera must reach the end of a line that is drawn on the floor. Another example is the application of UAVs (Unmanned Aerial Vehicles) for chemical spraying in agriculture [24], where a robot equipped with a camera detects lines of crops and flies along them. Another example is [25], where a UAV was used for pipeline inspections and the data from the vision system processed using a Convolutional Neural Network were used for navigating the drone along the pipe. However, the system was tested only for pipes lying flat on the ground and the UAV was flying at a fixed altitude. It must be noted that, even though these robots fly in a 3-dimensional (3D) space, they still follow a line detected on the ground, i.e., defined in a 2D space, at a constant flight level. There are very few attempts to implement the line-follower principle for following a path physically defined in a 3D space. One such example is [26], where a UAV is designed for an aerial power line inspection. It tracks the powerline using a LiDAR sensor. Another example, described in [27], is also a UAV for power line inspection. In this paper, the arrangement of the line to follow is extracted from the camera image of the aerial powerline. To keep the UAV centred and oriented with the lines, the PID controller tuned with the fuzzy logic algorithm is used. Unfortunately, the performance of the algorithm was tested in simulation environments only. It is also noted that it will need to merge the information acquired by the camera with laser sensors in order to determine the distance of the UAV from the followed power line. In [28], the control of an AUV (Autonomous Underwater Vehicle) for underwater cable tracking is presented. The tracking of a cable buried in the seabed is achieved by magnetic field sensing. Thanks to this, not only can the direction of the tracking be obtained but also the distance from the cable can be maintained. However, the control algorithm does not take into account the presence of environmental disturbances (e.g., water currents) or obstacles (e.g., underwater sand-waves). The authors admit that a more advanced algorithm enabling more than just fixed AUV-to-cable distance maintaining in the 3D space must be developed. The need for developing the AUV for monitoring underwater structures, including pipes and cables, as well as some preliminary design of such a platform is also presented in [29].

From the literature review, it appears that vision processing techniques have a high capability for detecting objects and shapes, including lines and paths, but they lack sufficient information on the scene's depth. Thus, they must be accompanied by some distance measurement techniques, for example, LiDARs, radars, 3D cameras, or stereo vision, which increases costs and computational power demand. It also introduces problems with the synchronisation and fusion of data from various systems [30]. An additional drawback, in the context of the task described further in this paper, is that some types of sensors have limited capabilities in terms of detecting objects located close to sensors (i.e., closer than a few centimetres).

It must be underlined here that there are a lot of works on the subject of 3D path-following or tracking, with a great variety of methods being applied [31,32], but the 3D “path” is usually defined as a set of the robot’s configuration states, including coordinates to be reached or passed by the robot on its way. These coordinates are just points defined in some space. There is no physical line to be followed. This is a very different case from the one that will be described in this paper.

The aim of this work is to enable continuous, step-by-step control of robot incremental motion following a path in 3D space. This is obtained based on the classic line-follower concept, and the control rules are developed in order to allow an articulated, 6DoF (Degree of Freedom) industrial robot to follow the physically existing path defined in 3D space. The literature on this defined problem is, to the authors’ best knowledge, very limited.

Nevertheless, a somewhat similar problem was investigated in [33], where the shape of a physically existing line was detected using a camera and the articulated robot learned how to move along it using reinforcement learning. However, even though the line was defined by a hose placed in the robot’s space, the path was flat, i.e., this was in fact a 2D problem. Moreover, the solution was not calculated step by step and, as the path was analysed globally, it belongs to the global planning methods.

In [34], a very wide review of vision methods for automatic robotic welding is provided. Welding involves accurate control of the robot effector movement along a path that is defined by two adjacent, contacting elements. This is a problem that may appear similar to tracking a 3D path. However, the path is not detected directly but is rather described by the intersection of adjacent surfaces or by a groove on the surface. The presence of adjacent surfaces makes the locating of the path easier, and in the case of groove detection, closer to a 2D line-following problem. Most of the methods described in [34] offer a weld seam tracking accuracy between 1 and 0.1 mm. In the most accurate solutions, vision systems are accompanied by laser scanners.

Another publication that presents a similar task is [35], where an industrial robot was controlled by an Artificial Neural Network trained to follow an arbitrary path drawn by a human demonstrator on a 3D surface. Line detection and tool orientation were extracted from the image obtained from the camera placed on the robot’s effector. A laser beam was used to mark the tool orientation relative to the path on the surface to help extract it from the image obtained from the camera. Three different reinforcement learning algorithms were compared in terms of path tracking accuracy, with the Proximal Policy Optimisation reinforcement learning algorithm achieving a maximum tracking error of 4 mm and mean error of 0.2 μm .

Contrary to the majority of the publications, this paper presents a method for following a path that is physically defined in a 3D space. The path position is detected using a simple tool held by an industrial 6 DoF robot. The tool is equipped with six low-cost Time-of-Flight (ToF) sensors. The task to be performed is to move the robot arm with the tool along the path from the start to the finish point, without a collision with the path. Only the start position is known a priori, before the start of the whole process, so the path position and orientation must be detected, and proper robot arm movements must be calculated on-line, as the following of the path progresses.

2. Materials and Methods

2.1. General Description of the Proposed Solution

The general concept of the algorithm is to extend the classic idea of the line-follower into a 3D space. The path shape is defined using a hose mounted inside the robot’s workspace. The path’s position is detected by the sensors placed on two perpendicular

arms of the path detection tool (Figure 2). Thanks to this, the path's existence may be detected in points located at the sensors' line-of-sight intersections. However, this allows for path detection only inside the 2D space of the detection tool. In order to detect the path's orientation in the 3D space, the robot moves the tool by a small amount and performs a second detection. Thanks to this, it is possible to calculate how the sensing tool should be moved along the path to the next position to perform the next detection. Then, the process is repeated, taking into account the tool's new (now current) position, and its previous position. When moving to subsequent positions, it is also important to always keep the sensing tool oriented perpendicular to the path. It must also be underlined that it is assumed that the path is not placed flat but bends in various directions in 3D. The details of the proposed solution are presented in the following subsections of this chapter.

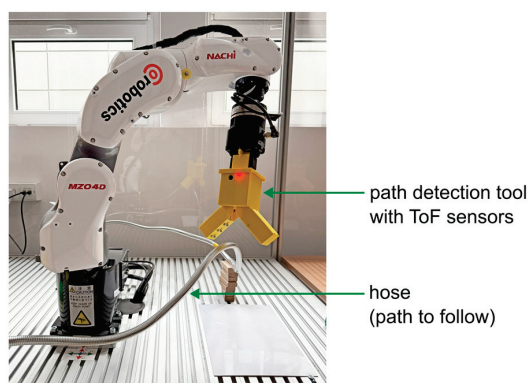


Figure 2. Robot holding path detection tool during the path-following.

2.2. Detected Path Position Calculation

Figure 3 presents an exemplary path detection tool with 6 sensor placements and the detection points of the path detection tool marked. The current tool position is always known, i.e., it is calculated based on the gripper global position returned by the robot controller. The centre of the tool's coordinate system (O , tool position) is located in the middle of the detection field. It lies at the intersection of the lines of sight associated with the two central sensors (No. 2 and 5 in Figure 3). The Y and Z axes lie in the tool plane, while the Z axis is perpendicular to this plane. The sensor spacing is a .

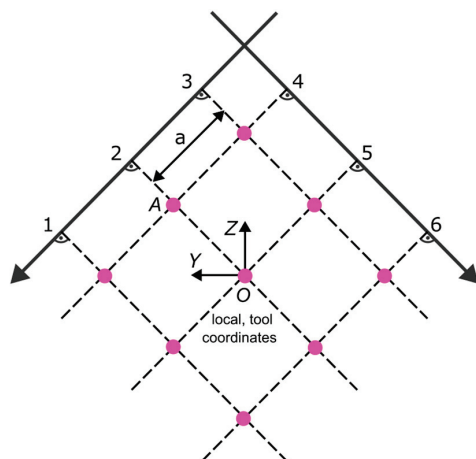


Figure 3. Path detection tool schema showing ToF sensors (1–6) placement, sensor separation a , and locations of detection points (purple circles), e.g., point A is the detection point at the intersection of the lines of sight of sensors 2 and 4.

Each of the intersection points of the sensors' lines of sight in the global system can be determined by knowing the position of the tool and its orientation in the form of Euler angles. Euler angles determine the orientation of the local tool coordinate system relative to the global system (located in the robot base). The first stage of the process of calculating these points is to determine the rotation matrix of the local tool system:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \quad (1)$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad (2)$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$R = R_z(\gamma) \cdot R_y(\beta) \cdot R_x(\alpha), \quad (4)$$

where α —rotation angle around the X axis (roll); β —rotation angle around the Y axis (pitch); γ —the angle of rotation around the Z axis (yaw); R_x, R_y, R_z —rotation matrices around individual axes, respectively, X, Y, Z; and R —tool rotation matrix.

The position of a point in the global coordinate system, knowing its position in the local coordinate system, can be determined using the following formula:

$$P = P_0 + R \cdot \vec{u}, \quad (5)$$

where P —the desired point position in the global system; P_0 —the centre of the local coordinate system; and $\vec{u}$ —the vector of the position in the local coordinate system (the vector between the centre of the local coordinate system and the position of the desired point in this system).

For example, the position of the point A in the global coordinate system (Figure 3) is

$$A = O + R \cdot \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}^T \cdot a. \quad (6)$$

The general formula for calculating detection point positions (intersections of sensors' lines of sight) is

$$P_i = O + R \cdot \hat{v} \cdot l. \quad (7)$$

where P_i —the intersection of sensors' lines of sight in the coordinate system of the robot base; $\hat{v}$ —unit vector; and l —the distance between the centre of the local coordinate system and the desired point i .

The next step is the calculation of the middle point based on the set of points where the path was detected, because it is possible that the path is detected in more than one point simultaneously. This is calculated according to the following formula:

$$S = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i, \frac{1}{n} \sum_{i=1}^n z_i \right), \quad (8)$$

where S —the midpoint, i.e., the point where the followed path is assumed to be; x_i, y_i, z_i —coordinates of points where the path was detected; and n —the number of points where the detection occurred.

In some cases, it is also possible that the path is not detected at all, for example, when the path is located between the sensors' lines of sight. In such case, $S = O$, which means that it is assumed that the path is in the middle of the detection tool and the robot should continue in the previously calculated direction. This is performed in order to deal with detection errors caused by the limited special resolution of the detection tool. It is also assumed that by moving forward, the path will eventually be again.

2.3. Calculation of the Next Tool Position

To determine the next position where the detection tool should be moved by the robot, it is necessary to calculate the appropriate vector (Figure 4). In the previous section, how to designate the S point that represents the detected path position was presented. Knowing the position of this point, the next step is to determine the vector between this point and the position of the detection tool (point O).

$$\vec{m} = \vec{OS} = \begin{bmatrix} x_S - x_O & y_S - y_O & z_S - z_O \end{bmatrix}, \quad (9)$$

where x_S, y_S, z_S —point S coordinates; x_O, y_O, z_O —point O coordinates.

Additionally, the tool displacement ($\vec{t}$ vector) from the position where the path was detected for the last time (point O') must be calculated:

$$\vec{t} = \vec{O'O}. \quad (10)$$

It must be noted that the O' position is not always the last (previous) tool position. This happens when the path is not detected by the tool. In this case, it is assumed that the path is located between the sensor's lines of sight, and it should be detected again in one of the next moves, so the tool is moved ahead in the previously calculated direction but the O' point used in (10) is still the one obtained when the path was detected the last time. Apart from handling temporary path loss, this also helps to react appropriately to path bends, depending on how sharp they are.

After summing up both vectors $\vec{t}$ and $\vec{m}$, the resultant vector $\vec{w}$ is obtained (11), which is the vector needed to determine the next tool position. Then, the vector $\vec{w}$ must be normalised (12). The resulting unit vector allows the determination of the new position shifted by a given distance d (Figure 4).

$$\vec{w} = \vec{O'S} = \vec{t} + \vec{m}, \quad (11)$$

$$\hat{w} = \frac{\vec{w}}{\|\vec{w}\|} = \begin{bmatrix} \frac{w_x}{\sqrt{w_x^2 + w_y^2 + w_z^2}} & \frac{w_y}{\sqrt{w_x^2 + w_y^2 + w_z^2}} & \frac{w_z}{\sqrt{w_x^2 + w_y^2 + w_z^2}} \end{bmatrix}, \quad (12)$$

$$P_n = O + \hat{w} \cdot d, \quad (13)$$

where $\vec{w}$ —vector between O' and S points; $\hat{w}$ —unit vector; d —tool displacement distance; and P_n —next tool position.

Figure 5 presents the general scheme of the path-following, showing 3 consecutive positions of the path detection tool.

An exception to the rules described above is the situation when the midpoint S is placed in the tool centre (point O). In this case, the next position is calculated based on the vector normal to the tool plane (tangent to the X axis of the tool).

$$P_n = O + R \cdot \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot d. \quad (14)$$

This is caused by the fact that tool rotation is calculated separately (which is described in the next section), and even though the path is in the tool centre, the $\vec{w}$ vector and normal vector may differ.

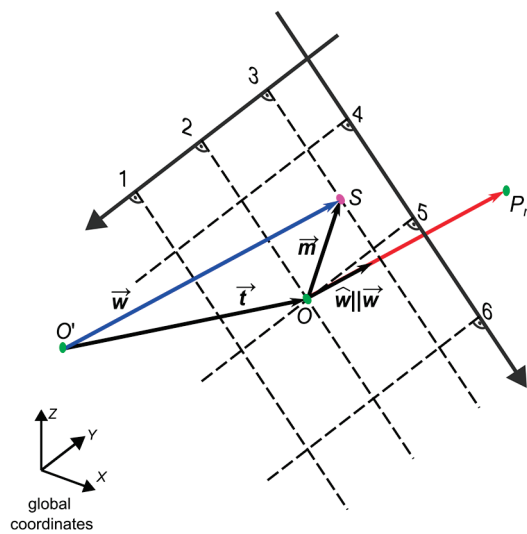


Figure 4. The $\vec{w}$ vector and its component vectors and location of P_n (next tool position) point in the global coordinate system; $\hat{w}$ is the unit vector parallel to $\vec{w}$. Other symbols: $\vec{t}$ —tool displacement vector; O —detection tool centre in the current position; O' —detection tool centre in the previous position; and S —detected path position.

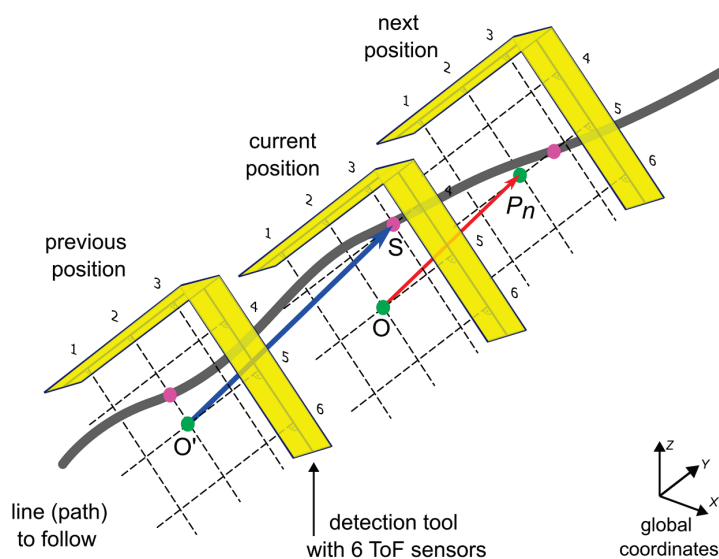


Figure 5. Three consecutive positions of the detection tool during path-following. Symbols: O —detection tool centre in the current position; O' —detection tool centre in the previous position; S —detected path position for the current detection tool position; and P_n —calculated next tool centre position, 1...6—ToF sensors.

2.4. Calculation of the Tool Rotation

The detection tool detection field, which lies in the Y and Z axes (in the local tool coordinate system), should be oriented perpendicularly to the path. In this orientation, it is easiest to detect the path and to avoid tool collision with the path. Therefore, the orientation of the tool is as important as the position in the Cartesian coordinates of the tool and must be determined on-line. The adopted method of determining Euler angles, analogous to the process of determining the sequential tool positions, is based on the calculation of the vector that defines the new orientation of the X axis of the tool's coordinate system. Thank to this, two rotation angles can be defined, and the third rotation angle (rotation around the X axis) is imposed. This tool orientation vector differs from the vector used for position calculation because it is possible to modify its parameters. By introducing this approach, it is possible to tune the algorithm and achieve smoother path-following.

To obtain the tool rotation vector, the vector normal to the plane lying in the Y and Z axes of the local tool coordinate system must be determined:

$$\hat{x} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \quad (15)$$

$$\vec{n} = R \cdot \hat{x}, \quad (16)$$

where $\hat{x}$ —unit vector oriented along X axis of the tool coordinate system; $\vec{n}$ —vector normal to the tool plane.

Using the previously determined $\vec{m}$ vector, the tool orientation vector may be calculated as follows:

$$\vec{w}_a = \vec{n} \cdot a + \vec{m} \cdot b, \quad (17)$$

where a, b —vector's modification parameters, experimentally adjusted during tests; $\vec{w}_a = \begin{bmatrix} w_{a,x} & w_{a,y} & w_{a,z} \end{bmatrix}$ —orientation vector (Figure 6).

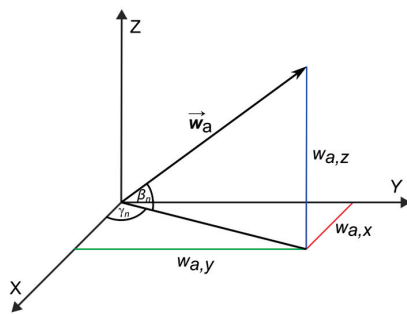


Figure 6. The $\vec{w}_a$ vector with its components, $w_{a,x}, w_{a,y}, w_{a,z}$, and two rotation angles that can be determined on the basis of this vector: β_n and γ_n .

If, in Equation (17), parameter a is set to the scalar value (i.e., length) of $\vec{t}$, the rotation is indirectly proportional to the path arc radius. When the radius is high, the path bend is small and there can be a few subsequent steps when the path is not detected by the tool. When the radius is low, the path bend is high, and the path is detected frequently. This means that normal vector length is dependent on the distance between the subsequent path detection positions. Parameter b , which is manually set constant with a value below 1, acts as tool rotation reduction coefficient. This prevents too high tool rotations when the path is detected in two subsequent positions in different detection tool points. In such a situation, the $\vec{m}$ vector may be much longer than vector $\vec{t}$. As a result, the orientation vector may be

oriented almost parallelly to the detection tool plane, i.e., it indicates very high rotation. This may cause the collision of the detection tool with the path and also results in high oscillations of the detection tool around the followed path. The reduction of the $\vec{m}$ vector prevents both situations. Thanks to this, smoother path-following is achieved, which, in turn, results in better path detection effectiveness. The only drawback of this solution is that in the case where the path has a low radius bend, the tool cannot be turned to the proper orientation in one step. In this case, the bend is negotiated in a few subsequent steps.

The trigonometric relationships and the Pythagorean theorem result in the following:

$$\tan \gamma_n = \frac{w_{a,y}}{w_{a,x}}, \quad (18)$$

$$\tan \beta_n = \frac{-w_{a,z}}{\sqrt{w_{a,x}^2 + w_{a,y}^2}}, \quad (19)$$

where γ_n —new rotation angle around the Z axis (yaw); β_n —new rotation angle around the Y axis (pitch).

The minus sign in Equation (19) results from the use of a right-handed coordinate system. In such a system, the positive angle β_n (pitch) corresponds to the counterclockwise rotation around the Y axis. In case the $w_{a,z}$ component of the vector is positive, it corresponds to a clockwise rotation. For this reason, the expression must include a minus sign to conform to the convention. As the result, using the arctangent function, new tool rotation angles can be determined:

$$\gamma_n = \text{atan} \frac{w_{a,y}}{w_{a,x}}, \quad (20)$$

$$\beta_n = \text{atan} \frac{-w_{a,z}}{\sqrt{w_{a,x}^2 + w_{a,y}^2}}, \quad (21)$$

and a constant value assigned to the angle of rotation around the X axis (roll):

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

Figure 7 presents the summary of combining calculations of both tool rotation and the next tool position.

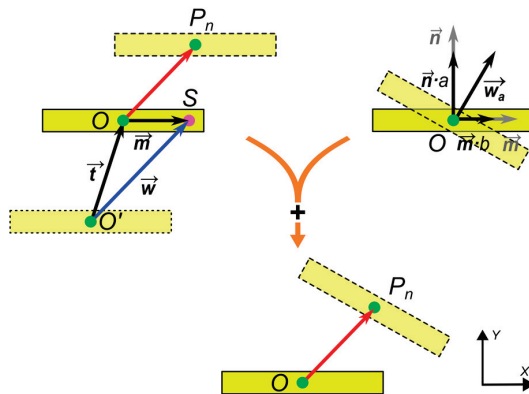


Figure 7. Example of combining the next position and tool rotation; detection tool seen from the top. Symbols: O—detection tool centre in the current position; O'—detection tool centre in the previous position; S—detected path position for the current detection tool position; P_n —calculated next tool centre position; vectors $\vec{w}_a$, $\vec{w}$, $\vec{t}$, $\vec{n}$, and $\vec{m}$ and a and b constants are described in the text.

2.5. Determining the Position of the Detection Tool Based on the Position of the Gripper

In previous sections, it was assumed that the position and orientation of the effector (detection tool) are known. However, during experimental tests, when the robot controller was queried for actual robot position, the gripper position was returned, because this is how the effector was defined in the robot controller. Of course, it is possible to define a new effector in the robot software, but it was decided to determine the position of the tool on the control computer side, i.e., in the Mathworks Matlab version 2023b (Mathworks, Natick, MA, USA) environment. This solution was selected to facilitate the incorporation of the tool geometry modifications, anticipated during project development, into the algorithm and to discard the need for adjusting the tool definition in the robot controller if any change in the tool is made. Additionally, thanks to this approach, the whole algorithm is more robot-independent, i.e., there are no configuration changes to be made in the robot controller settings.

The tool coordinate system is shifted (in relation to the gripper system) along the Z axis of the gripper by the distance l_e . Its orientation in relation to the gripper system can be described by the following angles: α_e —rotation angle around the X axis (roll); β_e —rotation angle around the Y axis (pitch); and γ_e —rotation angle around the Z axis (yaw). During the experiments, these angles were as follows: roll $180^\circ (\pi \text{ rad})$, pitch $0^\circ (0 \text{ rad})$, and yaw $-45^\circ (\frac{\pi}{4} \text{ rad})$.

In order to find the position of the tool, the first step is to determine the rotation matrix of the gripper system with respect to the global coordinate system. Using the relationships describing the rotation matrices around individual axes Equations (1)–(3), the rotation matrix of the gripper system has the following form:

$$R_c = R_z(\gamma_c) \cdot R_y(\beta_c) \cdot R_x(\alpha_c), \quad (23)$$

where $\alpha_c, \beta_c, \gamma_c$ —gripper rotation angles (roll, pitch, and yaw); R_c —gripper rotation matrix.

The tool coordinate system is shifted along the gripper Z axis in the positive direction. Knowing the position of the centre of the gripper coordinate system, it is possible to determine the position of the centre of the tool system using Equation (5). The formula for the position of the centre of the tool system takes the following form:

$$O = O_c + R_c \cdot \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \cdot l_e, \quad (24)$$

where O_c —gripper coordinate system centre.

Next, the orientation of the tool coordinate system relative to the global coordinate system must be determined. First, the rotation matrix of the tool system in relation to the gripper system is determined, again using Equations (1)–(3):

$$R_{ec} = R_z(\gamma_e) \cdot R_y(\beta_e) \cdot R_x(\alpha_e), \quad (25)$$

where R_{ec} —rotation matrix of the tool's coordinate system relative to the gripper's coordinate system.

Using matrices R_c and R_{ec} , it is possible to determine the rotation of the tool coordinate system in relation to the global coordinate system (i.e., to the robot base):

$$R_e = R_c \cdot R_{ec}, \quad (26)$$

where R_e —rotation matrix of the tool coordinate system relative to the coordinate system of the robot base.

On the basis of the determined matrix, R_e , it is possible to determine the orientation of the tool in the form of Euler angles. According to [36], however, there is always more than one possible solution and the selection of angles is arbitrary, because they result in the same orientation of the system.

3. Results

3.1. Experimental Setup

In order to prove the reliability of the proposed algorithm, experimental tests were carried out using the industrial, articulated, 6DoF robot Nachi MZ04D (Nachi Fujikoshi Corp., Tokyo, Japan) with a Nachi CFD controller [37]. The maximum load capacity of this robot is 4 kg, and the maximum reach of the flange is 541 mm. The robot is equipped with a Gimatic MPLF 2550 (Gimatic, Roncadelle, Italy) electric gripper and Schunk FT-AXIA 80 (Schunk, Lauffen/Neckar, Germany) force and torque sensor, which was installed on the robot but was not used during this research. The path to follow was set up using a shower hose (20 mm diameter) with a 3 mm copper wire inside, so that the hose retains its position when setting the path. The hose was attached at both ends to the worktable with metal brackets. Additionally, it was supported with some blocks and boxes to obtain the desired path, but these supports were placed outside the path-following zone.

The robot was enclosed in a safety cell with the dimensions $1250 \times 1250 \times 1250$ mm. In the robot controller software, the virtual safety walls were set to $-550 \dots +550$ mm in the X axis, $-550 \dots +550$ mm in the Y axis, and $+5 \dots +\infty$ mm in the Z axis. Similar boundaries were defined in the IK solver settings utilised in the path-following algorithm. The robot was placed in one of the didactic laboratories of the Gdansk University of Technology, which ensures stable environmental conditions (room temperature around 21°C , humidity around 30%, appropriate lighting).

The path detection was provided thanks to the specially designed detection tool (Figure 8). It consists of a box with communication electronics and six ST Microelectronics VL6180X (ST Microelectronics, Plan-les-Ouates, Switzerland) infrared Time-of-Flight sensors [38] placed on two perpendicular arms, three sensors per arm. These sensors can detect objects at a distance from 0 to 100 mm away from the sensor. The measurement accuracy is not explicitly provided in the datasheet; however, it contains the following information: measurement noise up to 2 mm, temperature-dependent drift 9 mm, and voltage-dependent drift 3 mm. All of the sensors were calibrated before use according to the procedure described by the manufacturer. Nevertheless, it must be noted that range (distance) measurement data are not directly used for path detection. Only the presence of the path in the detection zone of a sensor is used, not the exact distance value. Experimental verification proved that this is sufficient if only a small part of the path (hose) lies in the sensor's detection sight, as its detection is stable. It must also be noted that the detection field of each sensor is not a narrow, straight line but a cone with a vertex located in the sensor position and 25° vertex angle. Because of this, as sensors are separated by 22 mm (which makes 31 mm when considering the grid diagonal), and the hose diameter is 20 mm, it is common that the path is detected by more than one sensor. In such situations, path position is calculated using Equation (8). This means that the overall accuracy of the path centre detection is not worse than ± 16 mm.

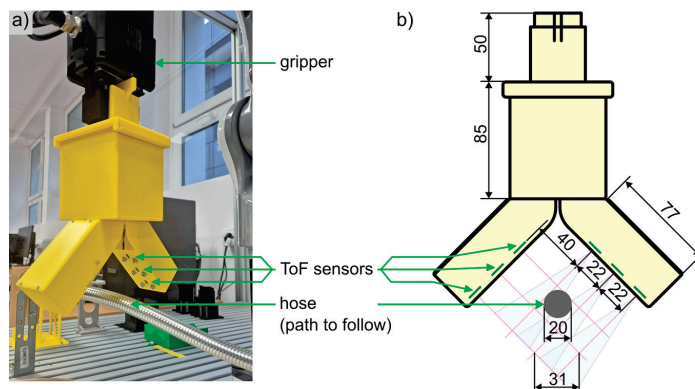


Figure 8. Path detection tool used during experimental verification: (a) view of the real tool in the robot gripper placed over the path to follow; (b) main tool diameters.

Data from sensors were read by an ST Microelectronics STM32G431 microcontroller (MCU) on an STM32 Nucleo board via I2C bus and then sent to a personal computer (PC). The communication between the tool and a control algorithm running in Matlab on the PC was provided via a wireless Bluetooth 4.0 BLE connection. The tool is independent from the robot, i.e., it is grasped by the robot's gripper at the beginning of the program and then moved to subsequent positions, but there is no connection (i.e., no wires) and no direct communication between the tool and the robot or the robot controller. For the designed detection tool, the l_e distance (Equation (24)) was 159.7 mm.

The software architecture of the whole system is presented in Figure 9. Data from the detection tool are read by the control program running in Matlab. This program, based on the current and previous detection data, calculates the next detection tool position, i.e., where the robot arm should be moved. Calculated robot joint positions are sent (via Ethernet) to the robot controller. On the controller, there are two programs running, the main one and the so-called User Task [39]. User Tasks are additional programs that can run in parallel with the main program. In the presented case, the User Task developed for the purpose of this research was responsible for two-way communication with the PC. It must be noted that although the User Task can perform various tasks associated with the robot control, it cannot contain commands that execute the physical moves of the robot arm. Moves can only be performed by commands included in the main program. Due to this, the User Task writes joint angles received from the PC to internal registers of the robot controller and the main program reads them and moves the robot to the desired position. Additionally, the developed User Task can also return current robot position as joint angles or Cartesian coordinates, operate the gripper, and perform some additional robot settings. Communication between the Nachi CFD robot controller and PC is provided via Ethernet and TCP protocol.

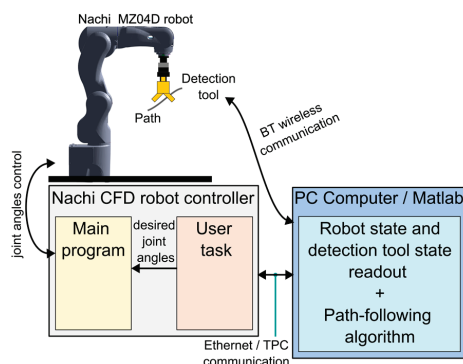


Figure 9. General software architecture of the presented system.

3.2. Algorithm Implementation, Constraints, and Supporting Solutions

The path-following procedure schema is presented in Figure 10. The process starts with the initialisation of the communication between the program running on the PC in Matlab and the robot controller as well as the detection tool. Once all hardware elements are connected, some initial moves of the robot are performed. These are detection tool pick-up and moving to the start position located over the path. Then, in the loop, for each stage the following steps are repeated for a given number of iterations:

- Read sensor data and interpreting these data:
The measurement field of the detection tool is represented as a 3×3 array (sensor array). Path detection is possible at nine points that are located at the intersections of the sensors' line of sight. Each of the points is represented by one element of the sensor array. The matrix rows represent the sensors located in the left arm of the path detection tool (sensors 1–3) and the columns represent the sensors in the right arm (sensors 4–6). The path is marked as detected at a given point when its presence is signalled by two sensors whose fields intersect in this point. Additionally, if the path is detected too close or too far from the sensor it is marked as a detection error to prevent unambiguous data input to the path-following algorithm. A detection error is not treated as a wrong readout. In such a case, the algorithm continues as if the path was not detected in this step and calculates the next step based on the last successful readout (the O' point of the last step when the path was detected is used).
- Read the current position of the gripper from the robot.
- Calculate the effector's (detection tool) current position and orientation based on the position of the gripper using Equations (23)–(25).
- Perform path-following algorithm step:
Calculate the effector's new position based on its current and previous positions and detected path position. First, the detected path position is calculated using Equations (1)–(8). Then, the next tool position and rotation is calculated using Equations (9)–(14) and (15)–(22). It must be noted here that according to the detection condition, a “long” or “short” step is performed (according to the value of the variable d in Equations (13) and (14)). A “short” step is performed when the path is detected only in the central point of the detection field or when it is not detected by any sensor. In all other cases, a “long” movement is performed. This is to prevent an insufficient correction of the detection tool position when the path trajectory bends or losing the path when moving straight ahead. Moreover, in some cases, due to the algorithm's response to detecting a path in a position other than the centre of the detection field, the calculated tool rotation may have a high angle value (see Section 2.4). This has a negative impact on the smoothness of the motion along the path and may also cause the loss of path detection. Rotation reduction is accomplished by reducing the length of the $\vec{m}$ vector. It is multiplied by a constant parameter (named b in Equation (17)), which should have a value greater than 0 and less than 1. This reduction may, however, cause the situation where the tool rotates by a smaller angle than is needed. Fortunately, in such a case the path-following algorithm will recognise in the next iteration that the path is still positioned on the edge of the detection field and, in order to follow the path, it must be rotated again. Eventually, the algorithm corrects the tool orientation in a few subsequent movement steps.
- Calculate inverse kinematics (IK):
The IK function calculates the articulated variables of the robot based on the detection tool position and orientation determined by the path-following algorithm. To calculate IK solution, the Matlab generalised IK object [40] is used, which internally uses the

Broyden–Fletcher–Goldfarb–Shanno (BFGS [41]) gradient projection algorithm solver. As a starting point for the solver to find the solution for the next position, the previous robot position is used. There are also some additional solver constraints applied that limit the robot body elements to stay inside safety cell, force the solver to prefer the downward tool orientation, and limit maximum tool rotation around its axes to less than 90° to prevent tool reversal, which will cause a collision with the path.

- Send new joint positions to the robot controller.
- Execute robot move:
Robot moves are executed with an arbitrary chosen constant speed. Read, in a loop, current robot position and wait until it reaches the assigned position.

Once the algorithm performs the desired number of iterations, the finalising moves are executed, i.e., the detection tool is stored back in its holder, the robot moves to the safe position, and program execution logs are saved. During the development of the software for communication between the PC and the Nachi robot controller, a few minor software elements and information on Nachi MZ04 robot performance provided in [42] were employed.

The full software developed for the purpose of this paper is available as the Supplementary Data provided as [S1]. This includes the main program for the robot controller, the User Task for the robot controller, the STM32 project for STM32G431 microcontroller, and the Matlab implementation of the path-following algorithm.

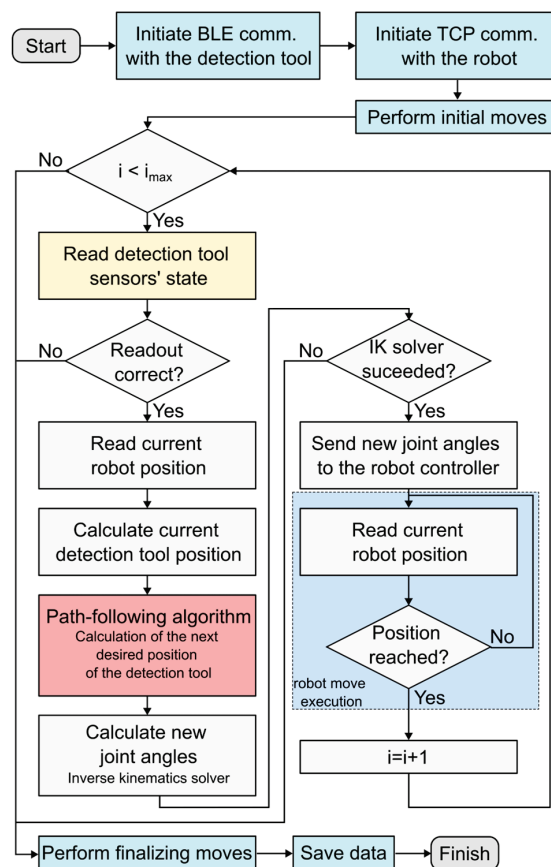


Figure 10. Path-following algorithm schema.

3.3. Experimental Verification

To demonstrate the effectiveness of 3D path-following using an industrial robot controlled by the developed algorithm and the proposed path detection tool, tests were

performed for four different path settings. Path 1 was tested twice and then tests for paths 2, 3, and 4 were performed. In each presented case, the task was successfully accomplished. Below, views of the recognised paths are visualised. In each figure, the robot is presented in the initial position, and the path is presented as a green line. Note that the figures show path positions as they were detected by the path detection tool and as they were represented in the algorithm. These are not the actual, real locations of the path but are close to them. Additionally, photos of some selected positions obtained by the robot during the program run are also presented. A full view of each run is presented in the video available as the Supplementary Data provided as [S2].

3.3.1. Path 1, Run 1

The first path was composed with some up and down bends with moderate arc radiuses. This path was successfully followed by the robot. Figure 11 presents four views of the detected path and Figure 12—three selected examples of robot positions obtained during path-following.

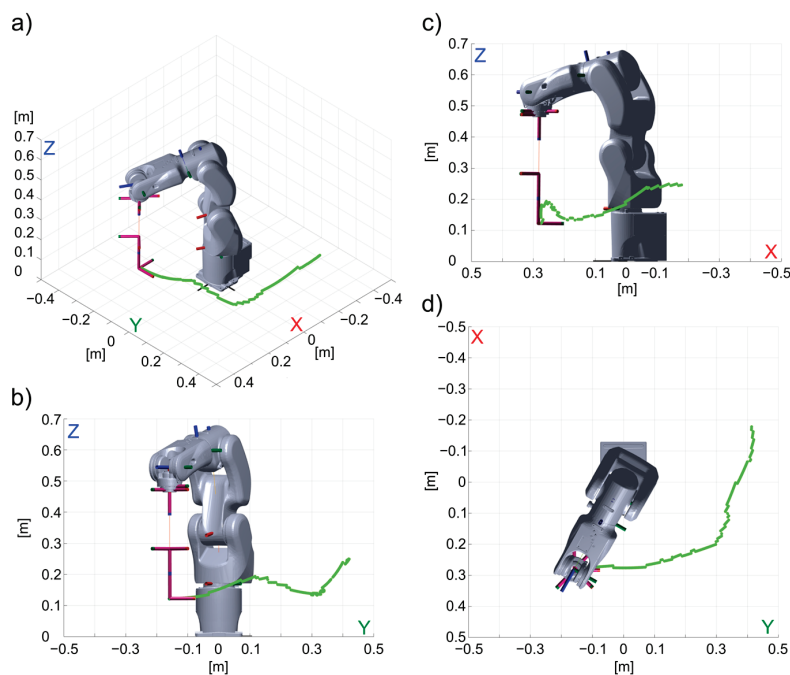


Figure 11. Path 1, run 1—path (green line) detected by the detection tool (a) isometric view, (b) front view (YZ plane), (c) side view (XZ plane), and (d) top view (YX plane).



Figure 12. Path 1, run 1—(a–c) three selected examples of robot positions obtained during path-following; full video available as Supplementary Material [S2].

3.3.2. Path 1, Run 2

The same path as in the previous run was used and it was also followed with success. Figure 13 presents four views of the detected path and Figure 14—three selected examples of robot positions obtained during path-following. Some differences in the detected path positions compared to run 1 may be noticed, but they are small. Figure 15 presents both runs overlayed, so the differences are easier to notice. The largest difference between detected positions of both paths is 15.2 mm (which is a value similar to the path detection accuracy described in Section 3.1), and the mean difference is 5.2 mm. This test confirms the repeatability of the results obtained using the proposed solution.

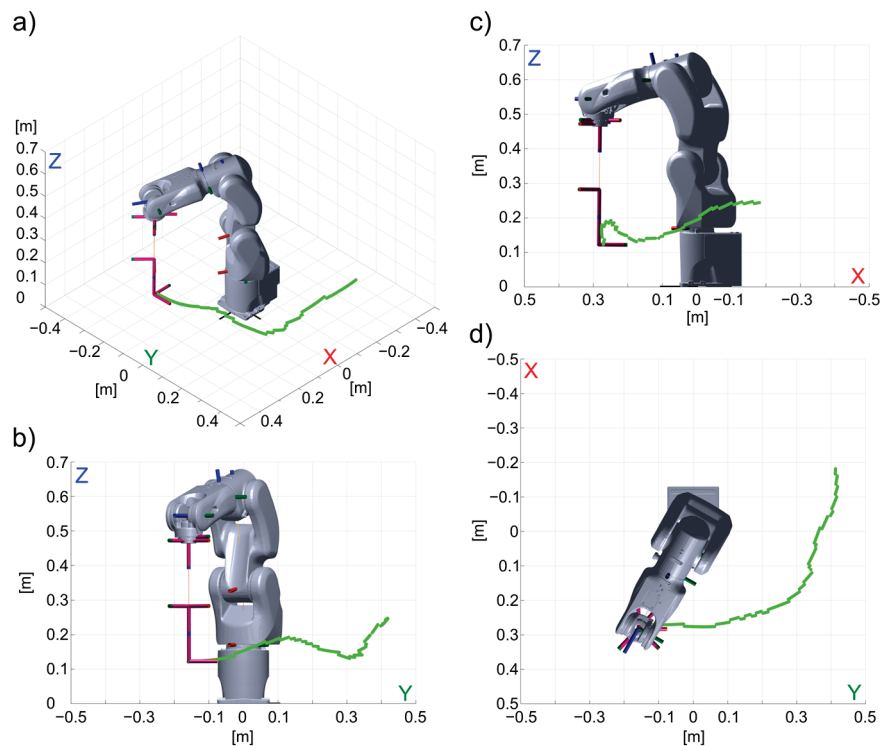


Figure 13. Path 1, run 2—path (green line) detected by the detection tool (a) isometric view, (b) front view (YZ plane), (c) side view (XZ plane), and (d) top view (YX plane).



Figure 14. Path 1, run 2—(a–c) three selected examples of robot positions obtained during path-following; full video available as Supplementary Material [S2].

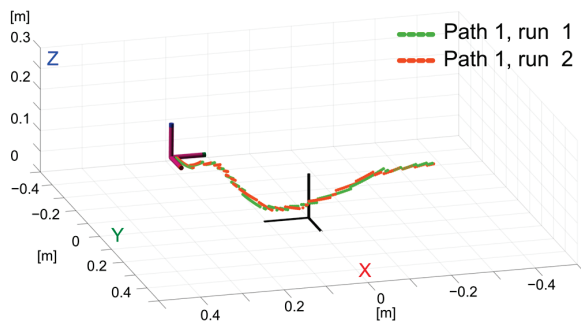


Figure 15. Path 1, run 1 and 2 overlaid; the robot body is removed for a clearer view.

3.3.3. Path 2, Run 1

In this test, the path had more up and down bends and path arcs had smaller radiiuses. This path was also successfully followed. Figure 16 presents four views of the detected path and Figure 17—three selected examples of robot positions obtained during path-following.

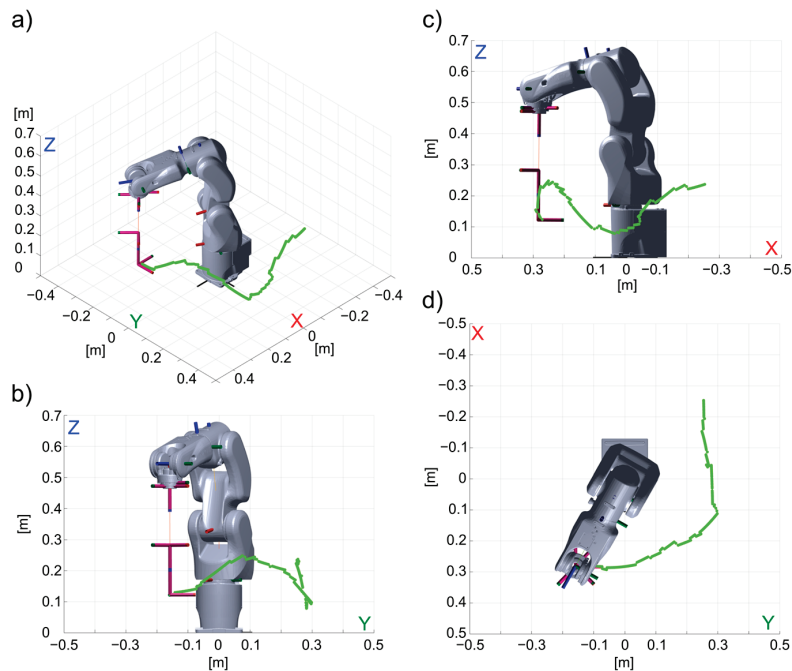


Figure 16. Path 2, run 1—path (green line) detected by the detection tool (a) isometric view, (b) front view (YZ plane), (c) side view (XZ plane), and (d) top view (YX plane).



Figure 17. Path 2, run 1—(a–c) three selected examples of robot positions obtained during path-following; full video available as Supplementary Material [S2].

3.3.4. Path 3, Run 1

This path had a longer, almost flat section at the beginning, followed by a high bend section. Additionally, it was placed closer to the robot base in its middle section. At the end of the run, the detection tool slightly collided with the path, but the algorithm was able to correct the tool trajectory and continued properly. This means that the path was, once again, followed with success. Figure 18 presents four views of the detected path and Figure 19—three selected examples of robot positions obtained during path-following.

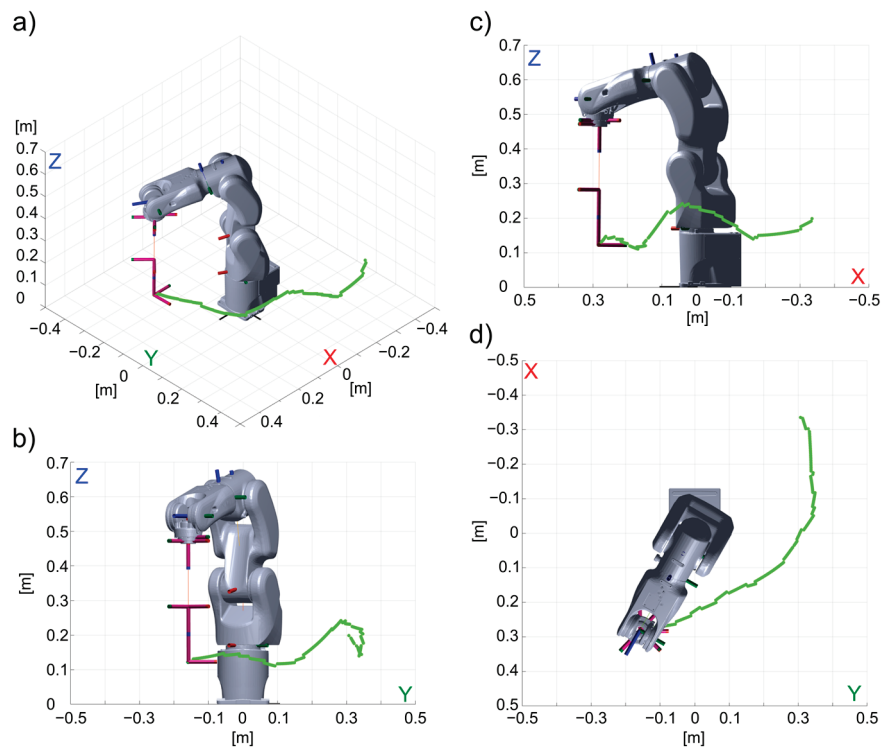


Figure 18. Path 3, run 1—path (green line) detected by the detection tool (a) isometric view, (b) front view (YZ plane), (c) side view (XZ plane), and (d) top view (YX plane).



Figure 19. Path 3, run 1—(a–c) three selected examples of robot positions obtained during path-following; full video available as Supplementary Material [S2].

3.3.5. Path 4, Run 1

This path was shorter, but it differed from the others by being bent in the “outbound” direction (i.e., away from the robot) and then going “up”. In this case, the boundary in the X+ axis was changed to 850 mm, as it was necessary to slightly cross the standard limits of the robot safety cell. The cell doors were open, so there was no collision with the cell

during the run. Figure 20 presents four views of the detected path and Figure 21—three selected examples of robot positions obtained during path-following.

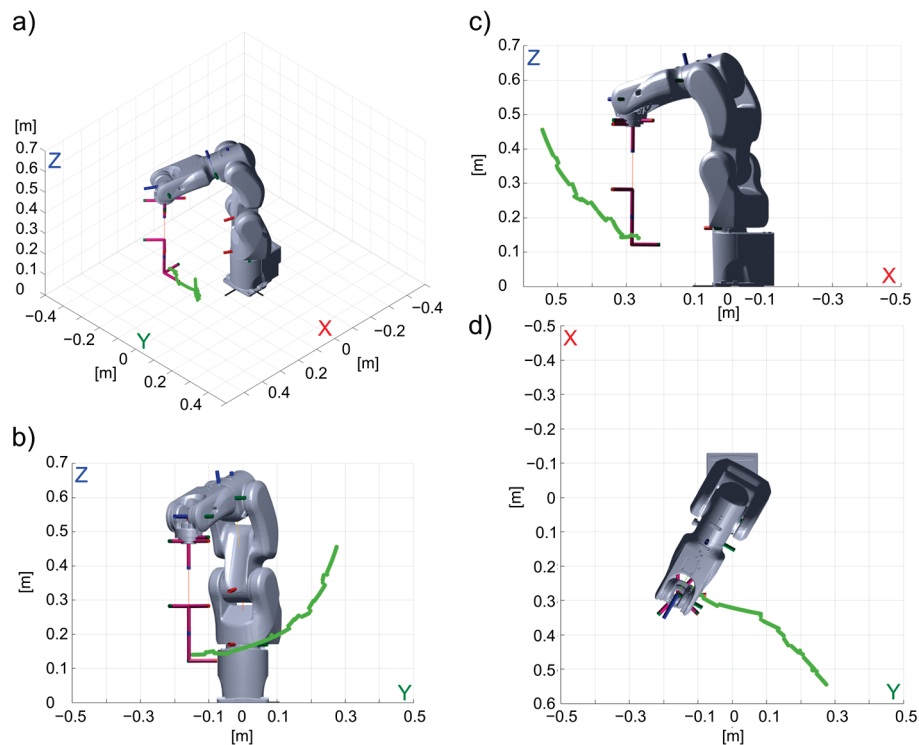


Figure 20. Path 4, run 1—path (green line) detected by the detection tool (a) isometric view, (b) front view (YZ plane), (c) side view (XZ plane), and (d) top view (YX plane).



Figure 21. Path 4, run 1—(a–c) three selected examples of robot positions obtained during path-following; full video available as Supplementary Material [S2].

4. Discussion

The proposed algorithm, despite the simplicity of the measurement tools used and despite being based on the simple, classically inspired line-follower concept, turned out to be effective in following the path defined in a 3D space. Its capabilities and practical effectiveness, however, are limited by the capabilities of the robot itself and the inverse kinematics solver. In particular, these are the issues associated with avoiding the robot's singular positions when tracing the path, the physical limitations of the robot's kinematics resulting from its construction, and the size of the working area, limitations of robot movements resulting from, for example, the wiring of the equipment installed on the robot and the physical dimensions of the detection tool and the path itself. During the tests, if the path-following run had to be interrupted (i.e., halted), in most cases it was caused by the

inability of the IK solver to find the proper solution due to the robot's limited workspace or because the expected position was simply impossible to reach for the robot due to its kinematic constraints. This means that the limits of the proposed solution do not lie in the path-following algorithm itself but rather in the limits of the employed robot.

During the experiments, the robot moves were performed with a limited, safe speed of 50 mm/s. Higher speeds may be applied, but when choosing them the step calculation time must be taken into consideration. A limitation might lie in the time needed for the calculation of the next position by the IK solver. During the presented tests, this was usually calculated in a short time (0.25 s on average), but in some difficult to reach positions it took a couple of seconds for the solver to provide a solution. The algorithm was run on a laptop computer with an Intel i5-1340p processor and 16 GB RAM. Another limitation may be caused by the time needed for reading the detection sensors' state and by the time needed for communication with the robot. During tests, one complete step (starting from the call to read sensor data up to executing and completing the move to the new position) was usually performed in approximately 1.7 s.

The algorithm should work properly when applied with almost no modifications on any articulated robot. Only the robot kinematics description and constraints for the IK solver must be updated to match a different robot. The application of the proposed solution should also be possible for mobile robots, but it would not be as straightforward as in the case of industrial robots and some elements would have to be adapted first. This is one possible goal for further algorithm development and research. Additionally, in the current form the algorithm does not include procedures for processing path splits or intersections, which may be a drawback. However, as each step is calculated based on local detection results only, the path as a whole (i.e., globally) does not have to be predetermined as it is detected step by step. This means that the path may change its shape during path-following. The only limitation is that the path cannot change too much in a close vicinity of the detection tool because in this case the path may leave the detection field. On the other hand, even if the path moves inside the detection field, this would be treated by the algorithm as a path bend, and it will adapt to the new path position.

The ability of the algorithm to follow a sharp path bend depends on many factors. First of all, the physical dimensions of the path and the detection tool limit the tool rotation in subsequent steps. Secondly, during calculations of tool rotation, rotation reduction is applied so the tool cannot be rotated too much. Without reduction, the tool may be extensively rotated in every move when the path is detected in different detection points in two subsequent steps. Reduction decreases this effect. Vector reduction values (a and b in (17)) are adjustable, and their values should therefore be the result of a compromise. Stronger reduction will limit the ability to follow path bends with small arcs but, on the other hand, will reduce the "waving" movement of the tool around the path. This will result in an overall smoother path-following. Additionally, path width (i.e., diameter), detection tool dimensions, sensor spacing and characteristics (ex. cone-shaped detection field), etc., also influence the ability to follow sharp path bends. During the tests, the smallest arc of bend was around 10 cm.

During the experiments, it appeared that sometimes, especially when the tool was moving away from the path, the path was lost and the algorithm was not able to locate it. When the tool was moving towards the path, such a problem almost never occurred. To prevent losing the path, more sensors should be used, and they should be placed more densely. Another solution may be to return to the last successful path detection position if the path is not detected for a given number of steps, and then to restart the moves in a slightly modified direction.

The ToF sensors that were used in the detection tool, as well as Bluetooth communication, proved to be very reliable. In practice, the only malfunctions were caused by the low energy level of the battery powering the tool. In such situations, communication was lost and the path-following procedure was aborted as the battery had to be replaced.

5. Conclusions

Thanks to the result of the presented research, it can be concluded that despite the use of only six low-cost ToF sensors (the price for one sensor breakout module with the sensor and integrated supporting electronics is approximately EUR 11, while one sensor itself costs approximately EUR 2), the low-cost microcontroller and Bluetooth (approximately EUR 15 for the MCU development board and EUR 12 for the Bluetooth module), and standard laptop computer, it was possible to meet the expected goals. This was achieved using simple technical tools and a simple algorithm, without the need for applying a complex method like, for example, AI algorithms, which need long-lasting and high computing power and time-consuming training.

The presented solution is easily scalable. An improvement may be sought by increasing the number of sensors, e.g., to 10 instead of 6. This would improve the number of detection points, increase path detection accuracy, and reduce the occurrence of situations in which the algorithm loses the path or sharp bends in the path occur. The other proposal is to use LiDAR sensors (instead of ToF sensors), which should reduce the size of the detection tool and allow the reading of the position of the path to be much more accurate, not only at the intersections points of the detection zones of ToF sensors.

The presented algorithm together with its implementation (including hardware elements) has a centimetre-level path-following accuracy. It is limited mostly by the detection tool spatial resolution. At the current stage of elaboration, the main goal of this research was to develop the methodology and test it in practice. Further development, as mentioned above, will be directed towards enhancing the path detection accuracy.

When comparing the results to other solutions, it is worth noting that path detection and path-following methods for 2D paths, especially those based on vision systems supported by lasers and AI data processing, achieve even submillimetre-level accuracy. The methods used for 3D navigation and positioning or object and obstacle recognition, used, for example, for UAV or autonomous car control, have various accuracies, starting from metres down to millimetres depending on what is needed for the particular application. The most accurate methods used for object localization in a 3D space are precise structured light scanners. Commercially available scanners may achieve very high, micrometre-level accuracies, but are very expensive and need high computational power for data processing. In this context, the method proposed in this paper may appear inferior, but it is always a trade-off between the desired accuracy and system costs, including hardware and software. Thus, the described solution, even though it cannot match the extremely high accuracies of some of the most advanced solutions, may find applications, especially after detection resolution and accuracy are increased, which is possible within limited costs.

Potential applications of the proposed path-following method are tasks performed in the robot's workspace that require moving along a previously unknown but physically defined path, for example, contactless scanning or monitoring of various long, narrow objects. After extending the method to mobile robots, specifically aerial or underwater, the proposed idea could be implemented in the tasks of monitoring power lines, pipelines, ropes, or chains, hanging in the air or placed underwater, especially if exact positioning or minimising the path-following error are not of the greatest importance, or the costs need to be kept to a minimum. After increasing path detection accuracy, this approach could also

be applied, for example, in some less accuracy-demanding welding, glueing, or 3D printing jobs. The method could also be used as a supplementary add-on to other methods, for example, those relying on vision systems, especially in adverse environments or conditions such as underwater operations, dark areas, or in severe weather conditions.

Supplementary Materials: The supporting information for this paper are available as Open Data. Link S1: Control software and experimental data. 2024. <https://doi.org/10.34808/qtp2-7s79>; Link S2: Video data set. 2024. <https://doi.org/10.34808/92gd-2y47>.

Author Contributions: Conceptualization, M.A.G.; methodology, T.F.W., I.D.O. and M.A.G.; validation, M.A.G.; software, T.F.W., I.D.O. and M.A.G.; investigation, T.F.W. and I.D.O.; data curation, T.F.W. and I.D.O.; writing—original draft preparation, T.F.W., I.D.O. and M.A.G.; writing—review and editing, T.F.W., I.D.O. and M.A.G.; visualisation, T.F.W., I.D.O. and M.A.G.; supervision, M.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the Supplementary Materials—see [S1,S2].

Acknowledgments: The Nachi MZ04 robot used for experimental investigation was acquired thanks to the internal Gdansk-Tech grant: IDUB CoreEduFacilities 14/2021/EDU—MecHaRo-Lab project.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Marashian, A.; Razminia, A. Mobile robot's path-planning and path-tracking in static and dynamic environments: Dynamic programming approach. *Robot. Auton. Syst.* **2024**, *172*, 104592. [CrossRef]
2. Namgung, H. Local Route Planning for Collision Avoidance of Maritime Autonomous Surface Ships in Compliance with COLREGs Rules. *Sustainability* **2022**, *14*, 198. [CrossRef]
3. Debnath, D.; Vanegas, F.; Sandino, J.; Hawary, A.F.; Gonzalez, F. A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications. *Remote Sens.* **2024**, *16*, 4019. [CrossRef]
4. Ou, X.; You, Z.; He, X. Local Path Planner for Mobile Robot Considering Future Positions of Obstacles. *Processes* **2024**, *12*, 984. [CrossRef]
5. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. [CrossRef]
6. Tan, C.S.; Mohd-Mokhtar, R.; Arshad, M.R. A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms. *IEEE Access* **2021**, *9*, 119310–119342. [CrossRef]
7. Thompson, N.; Greenewald, K.; Lee, K.; Manso, G.F. The Computational Limits of Deep Learning. In Proceedings of the Ninth Computing within Limits 2023, Virtual, 14–15 June 2023. [CrossRef]
8. Youn, W.; Ko, H.; Choi, H.; Choi, I.; Baek, J.-H.; Myung, H. Collision-free Autonomous Navigation of A Small UAV Using Low-cost Sensors in GPS-denied Environments. *Int. J. Control Autom. Syst.* **2021**, *19*, 953–968. [CrossRef]
9. Mekathlon—International Line Follower Robot Competition. Available online: <https://www.mekathlon.com/fastest-line-follower> (accessed on 18 December 2024).
10. Robotex International. Available online: <https://robotex.international/line-following/> (accessed on 18 December 2024).
11. Minaya, C.; Rosero, R.; Zambrano, M.; Catota, P. Application of Multilayer Neural Networks for Controlling a Line-Following Robot in Robotic Competitions. *J. Autom. Mob. Robot. Intell. Syst.* **2024**, *18*, 35–42. [CrossRef]
12. Magnum Automation Inc. AGV/AGC Assembly Line. Available online: <https://www.magnum-inc.com/systems/agvs-agcs/agc-assembly-line/> (accessed on 18 December 2024).
13. Mohammed, M.S.; Abduljabar, A.M.; Faisal, M.M.; Mahmmoud, B.M.; Abdulhussain, S.H.; Khan, W.; Liatsis, P.; Hussain, A. Low-cost autonomous car level 2: Design and implementation for conventional vehicles. *Results Eng.* **2023**, *17*, 100969. [CrossRef]

14. Zakaria, N.J.; Shapiai, M.I.; Ghani, R.A.; Yassin MN, M.; Ibrahim, M.Z.; Wahid, N. Lane Detection in Autonomous Vehicles: A Systematic Review. *IEEE Access* **2023**, *11*, 3729–3765. [CrossRef]
15. Anand, M.; Kalaisevi, P.; Arun Kumar, S.; Nithyavathy, N. Design and Development of Automated Guided Vehicle with Line Follower Concept using IR. In Proceedings of the 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 22–24 February 2023; pp. 1–11. [CrossRef]
16. Jang, J.-Y.; Yoon, S.-J.; Lin, C.-H. Automated Guided Vehicle (AGV) Driving System Using Vision Sensor and Color Code. *Electronics* **2023**, *12*, 1415. [CrossRef]
17. Bach, S.; Yi, S. An Efficient Approach for Line-Following Automated Guided Vehicles Based on Fuzzy Inference Mechanism. *J. Robot. Control* **2022**, *3*, 395–401. [CrossRef]
18. Engin, M.; Engin, D. Path Planning of Line Follower Robot. In Proceedings of the 5th European DSP Education and Research Conference (EDERC), Amsterdam, The Netherlands, 13–14 September 2012; pp. 1–5. [CrossRef]
19. Mahaleh, M.; Mirroshandel, S. Real-time application of swarm and evolutionary algorithms for line follower automated guided vehicles: A comprehensive study. *Evol. Intell.* **2022**, *15*, 119–140. [CrossRef]
20. Moshayedi, A.J.; Zanjani, S.M.; Xu, D.; Chen, X.; Wang, G.; Yang, S. Fusion based AGV Robot Navigation Solution Comparative Analysis and Vrep Simulation. In Proceedings of the 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Behshahr, Iran, 28–29 December 2022; pp. 1–11. [CrossRef]
21. Liu, G.; Sun, W.; Xie, W.; Xu, Y. Learning visual path-following skills for industrial robot using deep reinforcement learning. *Int. J. Adv. Manuf. Technol.* **2022**, *122*, 1099–1111. [CrossRef]
22. Manorathna, R.P.; Phairatt, P.; Ogun, P.; Widjanarko, T.; Chamberlain, M.; Justham, L.; Marimuthu, S.; Jackson, M.R. Feature extraction and tracking of a weld joint for adaptive robotic welding. In Proceedings of the 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10–12 December 2014; pp. 1368–1372. [CrossRef]
23. Mathoworks Minidrone Competition. Available online: <https://www.mathworks.com/academia/students/competitions/minidrones.html> (accessed on 18 December 2024).
24. Basso, M.; Pignaton de Freitas, E. A UAV Guidance System Using Crop Row Detection and Line Follower Algorithms. *J. Intell. Robot. Syst.* **2020**, *97*, 605–621. [CrossRef]
25. da Silva, Y.M.R.; Andrade, F.A.A.; Sousa, L.; de Castro, G.G.R.; Dias, J.T.; Berger, G.; Lima, J.; Pinto, M.F. Computer Vision Based Path Following for Autonomous Unmanned Aerial Systems in Unburied Pipeline Onshore Inspection. *Drones* **2022**, *6*, 410. [CrossRef]
26. Schofield, O.B.; Iversen, N.; Ebeid, E. Autonomous power line detection and tracking system using UAVs. *Microprocess. Microsyst.* **2022**, *94*, 104609. [CrossRef]
27. Pussente, G.A.N.; de Aguiar, E.P.; Marcato, A.L.M.; Pinto, M.F. UAV Power Line Tracking Control Based on a Type-2 Fuzzy-PID Approach. *Robotics* **2023**, *12*, 60. [CrossRef]
28. Xiang, X.; Yu, C.; Niu, Z.; Zhang, Q. Subsea Cable Tracking by Autonomous Underwater Vehicle with Magnetic Sensing Guidance. *Sensors* **2016**, *16*, 1335. [CrossRef]
29. Gerigk, M.K.; Gerigk, M. Application of unmanned USV surface and AUV underwater maritime platforms for the monitoring of offshore structures at sea. *Sci. J. Marit. Univ. Szczec.* **2023**, *76*, 89–100.
30. Ziebinski, A.; Mrozek, D.; Cupek, R.; Grzechca, D.; Fojcik, M.; Drewniak, M.; Kyrkjebø, E.; Lin, J.C.-W.; Øvsthus, K.; Biernacki, P. Challenges Associated with Sensors and Data Fusion for AGV-Driven Smart Manufacturing. In *Computational Science—ICCS 2021*; Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloat, P.M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12745. [CrossRef]
31. Ghambari, S.; Golabi, M.; Jourdan, L.; Lepagnot, J.; Idoumghar, L. UAV path planning techniques: A survey. *RAIRO-Oper. Res.* **2024**, *58*, 2951–2989. [CrossRef]
32. ul Husnain, A.; Mokhtar, N.; Mohamed Shah, N.; Dahari, M.; Iwahashi, M. A Systematic Literature Review (SLR) on Autonomous Path Planning of Unmanned Aerial Vehicles. *Drones* **2023**, *7*, 118. [CrossRef]
33. Meyes, R.; Tercan, H.; Roggendorf, S.; Thiele, T.; Büscher, C.; Obdenbusch, M.; Brecher, C.; Jeschke, S.; Meisen, T. Motion Planning for Industrial Robots using Reinforcement Learning. *Procedia CIRP* **2017**, *63*, 107–112. [CrossRef]
34. Guo, Q.; Yang, Z.; Xu, J.; Jiang, Y.; Wang, W.; Liu, Z.; Zhao, W.; Sun, Y. Progress, challenges and trends on vision sensing technologies in automatic/intelligent robotic welding: State-of-the-art review. *Robot. Comput.-Integr. Manuf.* **2024**, *89*, 102767. [CrossRef]
35. Maldonado-Ramirez, A.; Rios-Cabrera, R.; Lopez-Juarez, I. A visual path-following learning approach for industrial robots using DRL. *Robot. Comput.-Integr. Manuf.* **2021**, *71*, 102130. [CrossRef]
36. Slabaugh, G.G. *Computing Euler Angles from a Rotation Matrix*; Technical Report; University of London: London, UK, 1999; pp. 39–63.

37. Nachi Fujikoshi Corp. *Manipulator Instruction Manual*, MMZEN-288-013; Nachi Fujikoshi Corp: Toyama City, Japan, 2021.
38. ST Microelectronics. VL6180X Proximity and Ambient Light Sensing (ALS) Module Datasheet, 2016. Available online: <https://www.st.com/resource/en/datasheet/vl6180x.pdf> (accessed on 13 February 2025).
39. Nachi Fujikoshi Corp. *CFDS Controller Instruction Manual User Task*, CFDs-EN-123-001A; Nachi Fujikoshi Corp: Toyama City, Japan, 2021.
40. Mathworks, GeneralizedInverseKinematics Object Documentation. Available online: <https://www.mathworks.com/help/robotics/ref/generalizedinversekinematics-system-object.html> (accessed on 7 January 2025).
41. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. *Numerical Recipes in C, The Art of Scientific Computing*, 2nd ed.; Cambridge University Press: Cambridge, UK, 1992; ISBN 0-521-43108-5.
42. Liwiński, K.; Budzisz, D. Testing the Capabilities of the Nachi MZ04 Robot in Terms of Performing Tasks with External Control. B.Sc. (Eng.) Thesis, Gdańsk University of Technology, Gdańsk, Poland, 2024. (In Polish).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

A Lightweight TA-YOLOv8 Method for the Spot Weld Surface Anomaly Detection of Body in White

Wei jie Liu, Miao Jia, Shuo Zhang, Siyu Zhu, Jin Qi * and Jie Hu *

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 201100, China; weijie.liu@sjtu.edu.cn (W.L.); jiamiao@sjtu.edu.cn (M.J.); zs98815@sjtu.edu.cn (S.Z.); 601382@sjtu.edu.cn (S.Z.)

* Correspondence: jinhuaqj@sjtu.edu.cn (J.Q.); hujie@sjtu.edu.cn (J.H.)

Abstract: The deep learning architecture YOLO (You Only Look Once) has demonstrated its superior visual detection performance in various computer vision tasks and has been widely applied in the field of automatic surface defect detection. In this paper, we propose a lightweight YOLOv8-based method for the quality inspection of car body welding spots. We developed a TA-YOLOv8 network structure which has an improved Task-Aligned (TA) head detection, designed to handle a small sample size, imbalanced positive and negative samples, and high-noise characteristics of Body-in-White welding spot data. By learning with fewer parameters, the model achieves more efficient and accurate classification. Additionally, our algorithm framework can perform anomaly segmentation and classification on our open-world raw datasets obtained from actual production environments. The experimental results show that the lightweight module improves the processing speed by an average of 2.8%, with increases in detection the mAP@50-95 and recall rate of 1.35% and 0.1226, respectively.

Keywords: deep learning model; open-world data processing; welding spot defect detection; YOLO

1. Introduction

In automotive body welding production lines, various visual defects can occur at spot welds, such as edge welds, overlap, cold solder joint and burr. The formation of welding defects is closely related to sheet surface conditions, assembly precision, and welding parameter settings. For example, contaminants or coatings on the sheet surface can alter electrical conductivity and melting behavior, leading to burn-through or insufficient fusion. Poor assembly precision (e.g., excessive gaps or misalignment) affects nugget formation and joint quality. Improper welding parameters (e.g., an excessive current or insufficient welding time) may result in defects such as over-melting or incomplete fusion.

The images of spot welds are visually rich and can partially reflect the internal quality of the molten core. These defects are critical because they directly impact the mechanical strength and structural integrity of the welding spots. For instance, incomplete fusion and overlap can lead to weak weld nuggets, reducing load-bearing capacity and fatigue life under cyclic stresses. Burrs and edge welding may introduce stress concentrations, increasing the likelihood of crack initiation. Studies [1–3] have shown that defective welds often fail mechanical strength tests such as tensile shear and peel tests. Moreover, industry standards such as NQST and AHSS/UHSS welding guidelines set strict criteria to ensure that welds meet the required strength and durability for automotive safety.

Welding spot images provide rich and intuitive information, allowing for a quality assessment by analyzing the weld spot's shape, brightness distribution, and surface texture. To ensure a comprehensive evaluation, we referred to the automotive industry standard NQST (Nugget Quality Strength Level) and the welding guidelines for AHSS/UHSS steels established by automobile manufacturers.

Currently, the detection of defects on small samples in production lines still relies on manual visual inspection, resulting in low efficiency and delayed feedback. Machine vision-based methods for spot weld quality detection simulate manual visual inspection and offer advantages such as high efficiency, low cost and a high degree of automation. However, due to the complex working conditions in production lines and strong lighting disturbances, the requirements for the accuracy and robustness of visual detection algorithms are quite high. Visual recognition methods can be divided into traditional image processing methods and deep learning methods.

Conventional image processing methods often rely on manually created features to describe spot weld quality, including shape, color and texture. Specialized image processing techniques are used to extract these features for detection. Since spot welds are circular, the Hough circle transform [4] is commonly used to extract the spot weld contours. For example, Duda [5] used the Hough circle transformation and multi-contour clustering analysis to locate spot welds with incomplete circular contours. To quickly locate spot weld regions, Liang et al. [6] proposed an angle-assisted circle detection algorithm based on a randomized Hough transform to improve computational efficiency and robustness. Considering the elliptical geometry of spot welds, some researchers [7–9] calculated the area of the molten core by adjusting the major and minor axes of the spot weld footprints to estimate the spot weld quality. There are some alternative non-destructive testing methods for welding spot detection, such as ultrasonic testing, X-ray testing and thermography. Active thermography is a classical non-destructive testing (NDT) technique widely applied in spot weld inspection due to its ability to detect both surface and subsurface defects. These methods [10,11] involve applying an external heat source, such as optical, inductive, or ultrasonic excitation, to the weld region, followed by capturing the resulting thermal response using an infrared camera. Defects such as cracks, pores, and incomplete fusion cause localized temperature variations, which can be analyzed to infer defect location, size, and depth. However, its accuracy can be influenced by material thermal conductivity, surface emissivity, and excitation parameters.

To overcome these limitations, many researchers have introduced classifiers to recognize spot weld quality in addition to manually developed features. Valentin et al. [12] described spot weld quality based on surface texture features using a gray-level co-occurrence matrix and developed a support vector machine (SVM)-based classifier to achieve the intelligent detection of spot weld quality. Javaheri et al. [13] used unsupervised clustering to extract RGB color features from five positions of the spot weld surface representing the spot weld quality and built a 15-dimensional color feature and mechanical property model (Back Propagation Neural Network, BPNN) to predict spot weld quality. Alghannam et al. [14] developed an experimental platform to detect weld defects and extracted surface morphology features using image segmentation and mathematical morphology methods and then applied fuzzy logic rules based on a knowledge base to predict spot weld quality. Younes et al. [15] further improved the image processing algorithm to detect non-circular spot welds. They used least squares ellipse fitting to extract new geometric features of the spot welds and constructed a fuzzy SVM model for detecting weld defects and evaluating weld quality in automobile bodies.

Benefitting from the excellent capabilities of Convolutional Neural Networks (CNNs) for representation and learning, deep learning (DL)-based visual recognition has shown outstanding performance in terms of precision and speed and has gradually become a standard technique. Supervised methods are commonly used in the visual inspection of weld quality, primarily utilizing classical network structures such as AlexNet [16], ResNet [17], MobileNet [18], Faster R-CNN [19] and YOLO [20] series models. For example, Kim et al. [21] developed a model with two folding layers and one fully connected layer that is able to accurately predict the tensile shear strength and nugget diameter of cold-rolled and galvanized steel welds. Liu et al. [22] proposed an improved Faster RCNN model to detect welding spots online. Zhong Wang et al. [23] proposed a network model with a bottleneck structure of feature compression and expansion layers, which achieves high precision with fewer parameters.

Since the images captured on site often contain multiple welds, the models must support both weld quality detection and localization. Detection networks that locate and simultaneously categorize defective welds typically distinguish between one-stage and two-stage methods, with speed and accuracy being the primary considerations. Single-stage methods such as YOLO are favored in industrial applications because of their fast inference speed and low operational costs. Liu et al. [18] developed an efficient, lightweight weld detection network based on YOLO that is capable of localizing and evaluating welds under complex conditions. Similarly, Adarsh used Tiny-YOLOv3 [24] to identify regions of interest, refining the detection accuracy by retaining only the weld contours and using a modified Hough transform for circle detection.

In summary, CNN-based object recognition algorithms use annotated images to predict object positions and classes, mainly distinguishing between one-stage and two-stage methods. The two-stage approach, exemplified by Faster R-CNN, includes a region proposal network and a classifier, while one-stage methods such as YOLO omit the proposal network and provide faster, more compact models suitable for weld detection. The main contribution of this study lies in optimizing the structure of the conventional object detection algorithm and proposing a specific model for the detection of welds with small samples and objects. By incorporating Task-Aligned detection to improve the cross-class feature alignment, the effectiveness of the approach is verified using an on-site image dataset.

2. Architecture Basis

Over the past decade, the YOLO (You Only Look Once) family of algorithms has remained at the forefront of object detection research, with the release of YOLOv8 in 2022 representing a new milestone. Safaldin et al. [25] proposed an improved YOLOv8 to detect moving objects. Some improvements [26–28] have been made to the YOLOv8s model to make it perform better in their own studies. Real-time object detection is crucial for various applications, including autonomous vehicles, robotics and video surveillance. Among the numerous object detection methods, the YOLO framework stands out for its impressive speed and accuracy. Since its introduction, the YOLO series has been updated several times to address limitations and improve overall performance.

As illustrated in Figure 1, YOLOv8 integrates the design concepts of the YOLO series into the backbone and neck area and converts the C3 structure of YOLOv5 [29] into the C2f structure. This change improves the gradient flow and adjusts the number of channels for different model scales. The classic backbone, Darknet53, uses residual modules with skip connections to mitigate the problem of gradient disappearance in deep networks. Each residue module consists of DeepConv2D, Batch Normalization, and Leaky ReLU (DBL), skip connections and element-wise additions. In addition, a key Res-n component

in Darknet53 includes zero-padding to maintain consistent input-output sizes, followed by a DBL layer and multiple residual modules that generate three feature maps at different scales to support multi-scale detection.

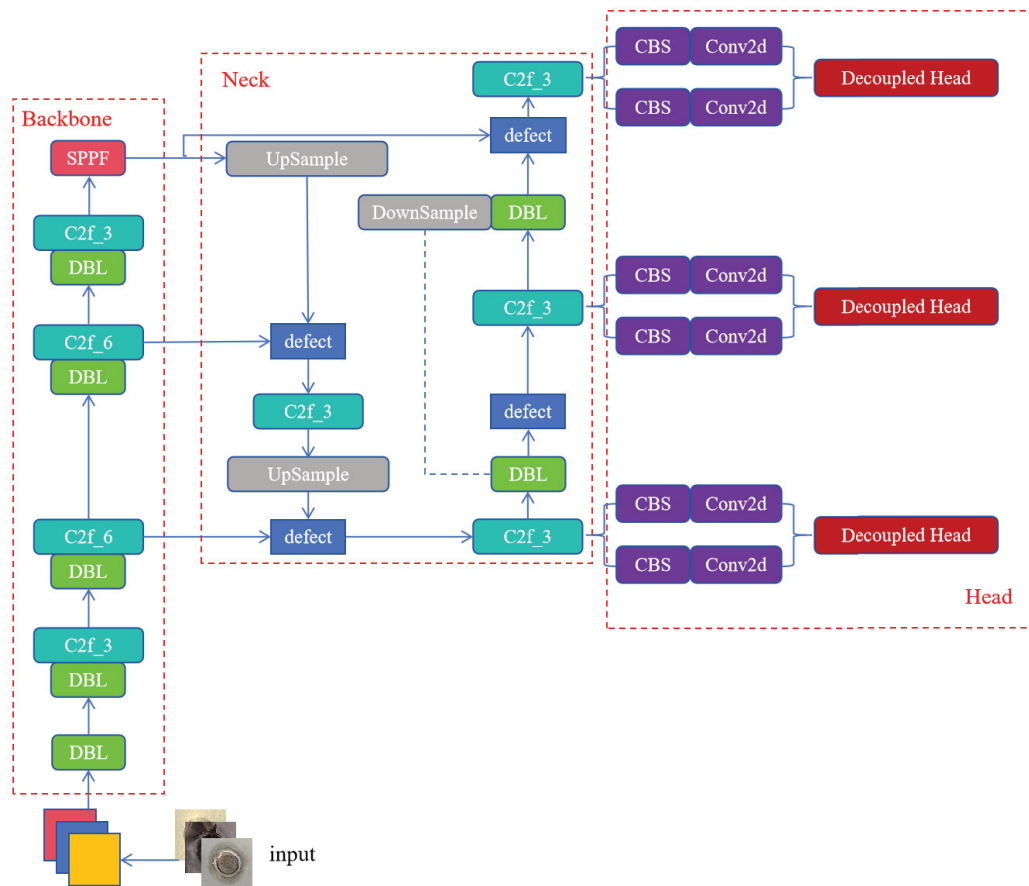


Figure 1. Architecture of YOLOv8 model. The different color parts of the input batches represent different image data. The different color parts of the architecture represent different function modules.

The neck section of YOLOv8 incorporates concepts from the Feature Pyramid Network (FPN) and enriches them with Path Aggregation Network (PAN) extensions to improve multi-scale feature fusion and increase detection and localization accuracy. The head section adopts a decoupled-head header structure that splits the classification and detection tasks for optimized processing.

3. Proposed Model and Algorithms

In the backbone module, the original Darknet-53 model in YOLOv8 contains an intermediate convolutional layer in which the number of filters increases from 512 to 1024 before dropping back to 512 in the next layer. Removing this intermediate layer eliminates 1024 convolutional filters, reducing the model’s convolution operations by 1024 and the number of 3×3 convolutional parameters by 9×1024 . This change significantly reduces the model’s parameters, resulting in a lighter model with improved performance. The backbone module is shown in Figure 2.

For the neck module, the structure of YOLOv8 adopts the PANet (Path Aggregation Network) by organizing the neck section with an FPN (Feature Pyramid Network) structure. Based on this concept, we propose a Multiple Cross-Layer FPN (MC-FPN) net-

work that combines both down-sampling and up-sampling paths with multiple cross-layer connections, enabling efficient feature fusion, as shown in Figure 3.

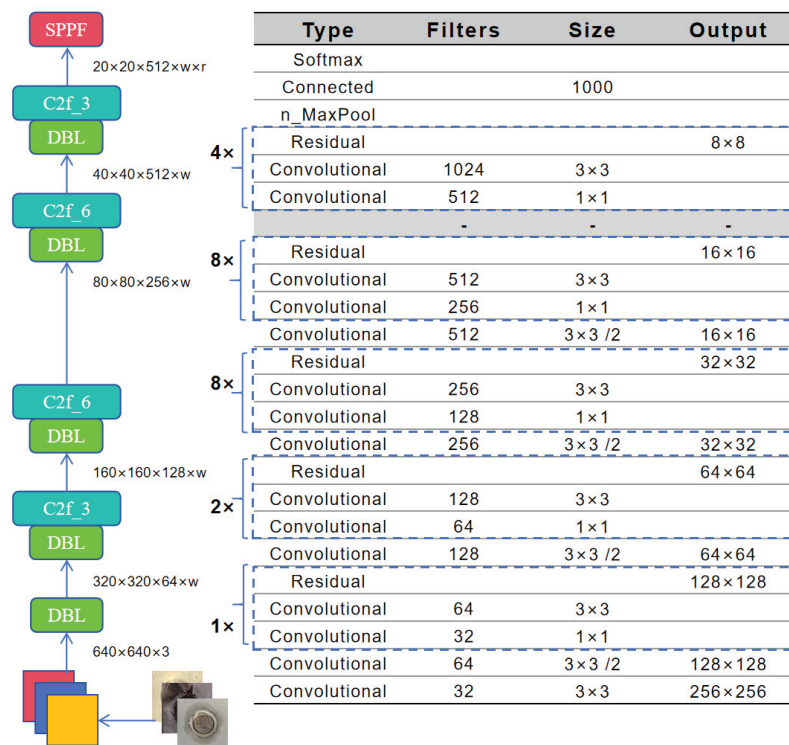


Figure 2. Improved backbone network of our architecture. On the left is a schematic diagram of the backbone network process for detecting spot weld images, while the right side shows the corresponding structure layer parameters and related information. The different color parts are the same as Figure 1.

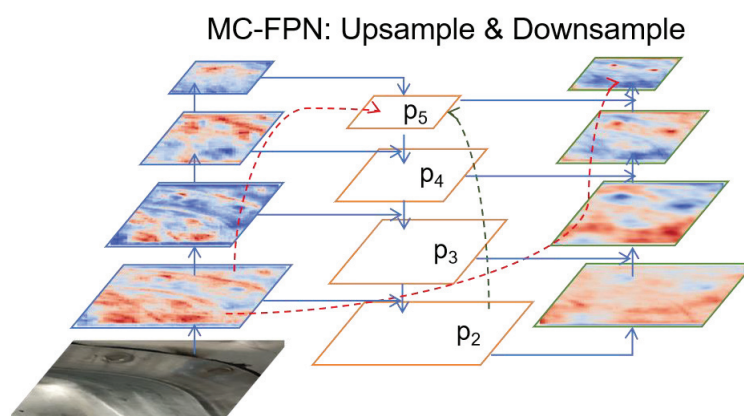


Figure 3. Proposed Multiple Cross-Layer FPN (MC-FPN) network. The different color dotted lines represent multiple cross layers, with P₂-P₅ being simplified representations of the intermediate connection layers.

The head module in YOLOv8 first branches into two CBS convolution modules and then passes through a Conv2D layer. The loss of the classification and the bounding box are calculated separately. In this section, a decoupled head structure is used to separate classification and detection tasks. By applying Distributional Focal Loss (DFL), the number of channels of the regression head is set to $4 \times \text{reg_max}$, where reg_max is normally set to 16. The bounding box head uses a Bbox loss for object identification, while the classification

head applies a Binary Cross-Entropy (BCE) loss function instead of VEL. This substitution is better suited for unbalanced data where the positive samples heavily outweigh the negative samples.

In this loss function, the indicator function A_{ij}^t represents whether the j -th anchor of the i -th grid is responsible for predicting a specific target; if it is, $A_{ij}^t = 1$; otherwise, $A_{ij}^t = 0$. The confidence score reflects the likelihood of an object being present within the network unit. Specifically, c_i represents the confidence of the ground truth box, and $\hat{c}_i$ is the confidence of the predicted box. $p_i(c)$ and $\hat{p}_i(c)$ denote the class probabilities of the ground truth and predicted boxes, respectively.

$$\begin{aligned}
 BCE_{Loss(c_i, p_i)} &= \mathcal{L}_{CIoU} \\
 &- \sum_{i=0}^{S \times S} \sum_{j=0}^B A_{ij}^t [c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i)] \\
 &- \sum_{i=0}^{S \times S} \sum_{j=0}^B A_{ij}^f [c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i)] \\
 &- \sum_{i=0}^{S \times S} \sum_{j=0}^{B=3} A_{ij}^t \sum_{c \in \text{class}} [p_i(c) \log(\hat{p}_i(c)) \\
 &+ (1 - p_i(c)) \log(1 - \hat{p}_i(c))]
 \end{aligned} \tag{1}$$

In contrast to the conventional single-level recognition head, which uses two parallel branches for classification and localization, a Task-Aligned head (T-head) is used in this study to improve the interaction between the two tasks. This collaborative structure allows for more accurate matching predictions. The T-head computes the features of the task interaction and performs the predictions via the Task-Aligned Predictor [30] (TAP). The efficient detection TA-head structure is shown in Figure 4.

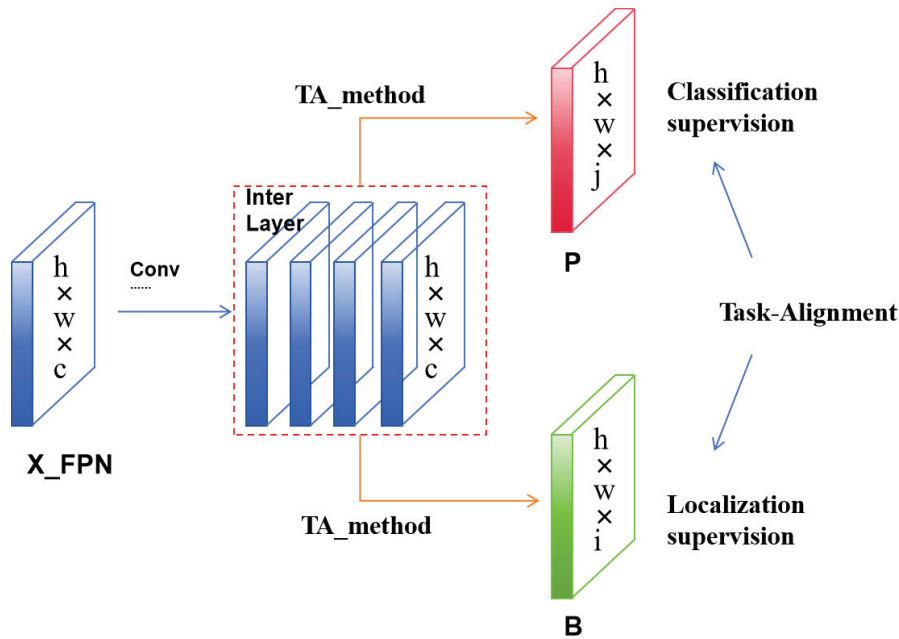


Figure 4. Task-Aligned head structure: to learn extensive task-interactive features from multiple convolutional layers.

To enhance the interaction between classification and localization, a feature extractor is proposed to learn extensive task-interactive features from multiple convolutional layers. This design facilitates task interaction and provides multi-level features with an effective receptive field across multiple scales for both tasks. Here, X_{FPN} represents FPN

features, and H , W , and c denote the height, width, and channel count, respectively, while i and j indicate the channel numbers for localization and classification tasks. The feature extractor computes task-interactive features using N consecutive transformation layers with activation functions:

$$X(k, Inter) = \begin{cases} \delta(conv(X_{FPN})), k = 1 \\ \delta(conv(X(k-1, inter))), k > 1 \end{cases}, \forall k \in \{1, 2, \dots, N\} \quad (2)$$

We conduct object classification and localization on the computed task-interactive features, allowing both tasks to effectively perceive each other's state. However, due to the single-branch design, task-interactive features inevitably introduce a certain degree of feature conflict between the two distinct tasks. Since object classification and localization have different objectives, they focus on different types of features (e.g., varying levels or receptive fields). Therefore, a layer attention mechanism dynamically computes these task-specific features to encourage task decoupling.

We further adopt Task-Aligned Learning (TAL) to guide our TA-head in making Task-Aligned predictions. From the perspective of task alignment, it dynamically selects high-quality anchors based on designated metrics while simultaneously addressing anchor assignment and weighting. For Task-Aligned sample assignment stage, to handle Non-Maximum Suppression [31] (NMS), anchor assignment for training instances must meet the following criteria:

1. Well-aligned anchors should jointly predict high classification scores with precise localization.
2. Misaligned anchors should receive lower classification scores and be suppressed.

Considering that classification scores and the Complete Intersection over Union [32] (CIoU) between the predicted and ground truth boxes indicate prediction quality for both tasks, this study uses a higher-order combination of classification scores and the CIoU to measure task alignment. The following metric is designed to calculate anchor alignment per instance:

$$t = v^\alpha \times w^\beta \quad (3)$$

where v and w represent the classification score and CIoU value, respectively, while α and β control the influence of the two tasks. Also, t plays a vital role in the joint optimization of both tasks towards task alignment objectives, encouraging the network to dynamically focus on high-quality anchors.

Training sample assignment is crucial to training an object detector effectively. To enhance task alignment, it is essential to focus on Task-Aligned anchors, using a simple assignment rule to select training samples: for each instance, the top m anchors with the highest t values are selected as positive samples, with the remaining anchors designated as negative samples. Training is completed by calculating new loss functions specifically crafted for classification and localization tasks.

4. Experiment

4.1. Environmental Settings

In order to verify the efficacy of the proposed approach, an experimental platform was set up employing Ubuntu 22.04 as the operating system and PyTorch 2.0 as the deep learning framework. YOLOv8 was employed as the baseline network model. The specific configuration of the experimental environment is elaborated in Table 1.

Table 1. Configuration and learning environment.

Basic Parameter and Hyperparameters	Value
Operating system	Ubuntu 22.04
Deep learning framework	PyTorch
Programming language	Python3.11
CPU	Intel(R) Xeon(R)
GPU	RTX 3090ti
RAM	100 GB
Data size	1024×1024
Batch size	64
Epoch	500
Momentum	0.845
Learning rate	0.002

4.2. Dataset

Our dataset consists of 1035 images taken in a real welding production environment, each containing a different number and size of spot welds. Figure 5a shows some samples we collected.

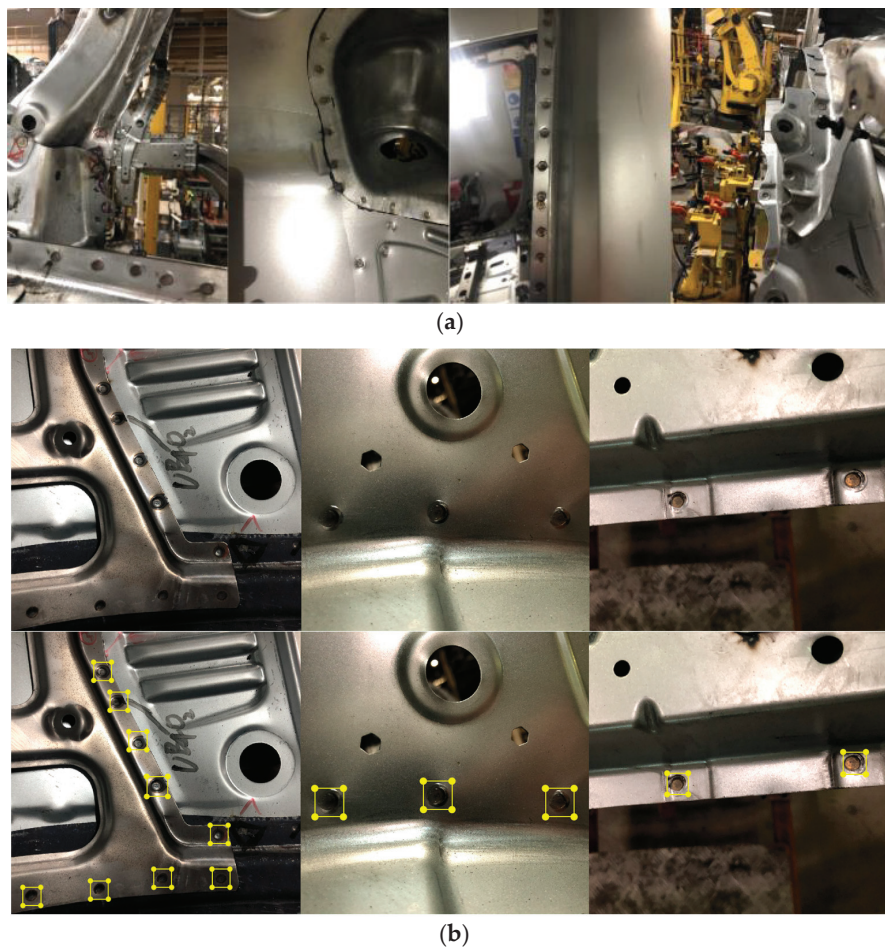


Figure 5. Welding spots sample images and annotated data in Body-in-White production lines. (a) shows the samples we collected in the production lines, while (b) shows the pretraining dataset and the labels (yellow squares).

The model is trained using a supervised learning paradigm that requires labeled data. Welding experts annotated the positions and quality of the spot welds in each image based on resistance spot weld signals and chisel test records. The labeled dataset was split into training, validation and test sets in a 10:1:1 ratio, with the input images resized to 1024×1024 , as shown in Figure 5b and Table 2. Prior to training, the dataset was processed using K-means clustering [33] to determine 9 anchor boxes corresponding to the dimensions of the weld spots. Training was performed for 500 epochs using the Adam optimizer with an initial learning rate of $1e-3$, with both momentum terms set to 0.9. A cosine learning rate scheduler with warm-up was used to optimize the network parameters. The stack size was set to 8 and Mosaic data expansion was applied. In our study, the choice of hyperparameters (e.g., learning rate, number of epochs, batch size, momentum) was based on established principles in deep learning and object detection tasks, as well as empirical evaluations during preliminary experiments. The dataset was expanded by random cropping, rotating and arranging, in particular increasing the number of small spot weld samples. The model parameters that performed best in the validation set were saved during training.

Table 2. Dataset of welding spots.

Class	Normal	Edge Weld	Overlap	Cold Weld	Burr	Distortion
training dataset	2250	1170	2400	510	780	1650
validation dataset	225	117	240	51	78	165
testing dataset	234	153	256	45	59	152

4.3. Results and Comparison

Our proposed TA-YOLOv8 model (hereinafter referred to as Weld Spot Defect Detection Model, WSDDM) was trained, validated and tested on a spot weld image dataset alongside several well-known object detection algorithms such as SSD [34], Faster R-CNN [35] and RetinaNet [36]. Figure 6 shows the performance of each trained model on the test set. When the IoU threshold is set to 0.5, the WSDDM shows a significant improvement in accuracy and speed over YOLOv8.

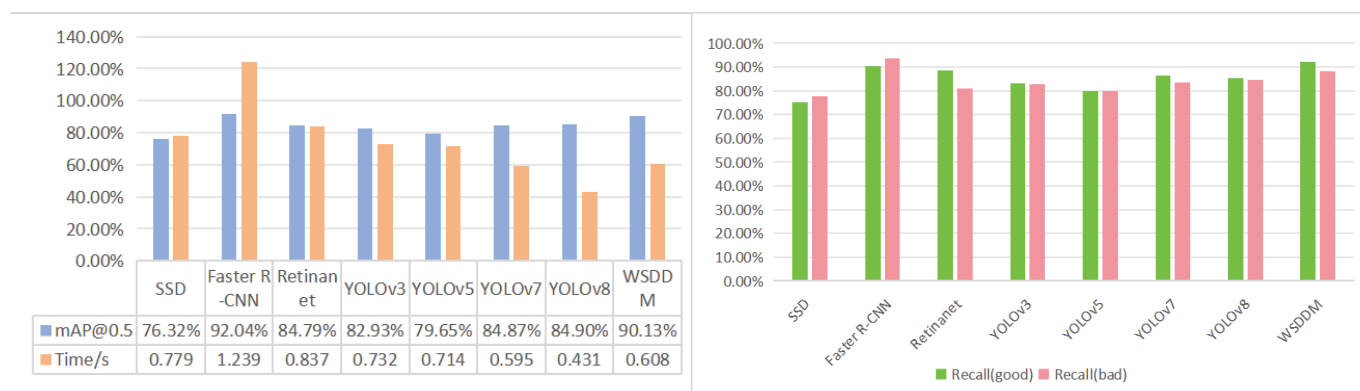


Figure 6. Performance comparison with typical object detection algorithms on test set.

This improvement is due to the improved feature fusion structure, backbone network adjustments, and refined loss functions of the model. While Faster R-CNN achieves the highest detection accuracy, it incurs significant time costs. The lightweight DarkNet53 network used in the WSDDM significantly increases the detection speed without sacrificing

accuracy. As can be seen in Figure 6, the WSDDM consistently identifies high- and low-quality spot welds and achieves an mAP of over 90%.

The texting results of the WSDDM are shown in Figure 7. A comparison of YOLOv8 and the WSDDM with images from the test set is also shown in Figure 7. The three columns show the results of YOLOv8, the WSDDM and manual annotations, with green boxes indicating acceptable spot welds and red boxes indicating defective spot welds. The numbers in each box indicate the level of confidence in the presence of spot welds. Despite different lighting conditions and irregular spot weld shapes, both DL models effectively detect spot welds, albeit with slight differences in the confidence level, and they can accurately distinguish similar round spot welds. For tasks involving the detection of dense, small spot welds, the WSDDM outperforms YOLOv8 in terms of detection performance and robustness across different spot weld sizes, with fewer missed or false detections. The model's dual, cross-scale feature fusion strategy increases sensitivity to detailed features, which is helpful in detecting small spot welds.



Figure 7. Some results of WSDDM and comparison between small welding spot detection models. We use green to represent the detected weld spots are normal, and red to represent the detected weld spots have defects or abnormalities.

Using the WSDDM method described in the previous chapter, each weld spot in the image was located and then cropped and saved, as shown in Figure 8. It provides an organized dataset of spot welds captured under real production conditions, categorized by various defect types as well as normal spots. Meanwhile, each defect type and normal class are represented by multiple images, which helps in creating a balanced dataset. For training object detection models, especially in tasks like small object detection, a balanced dataset with representative defect variations is crucial to achieve high accuracy and reduce model bias. Each image is annotated with labels indicating both class type and an ID number, which may refer to specific samples. This detailed annotation can aid in supervised learning, as the model can be trained to detect both the location and type of defect.

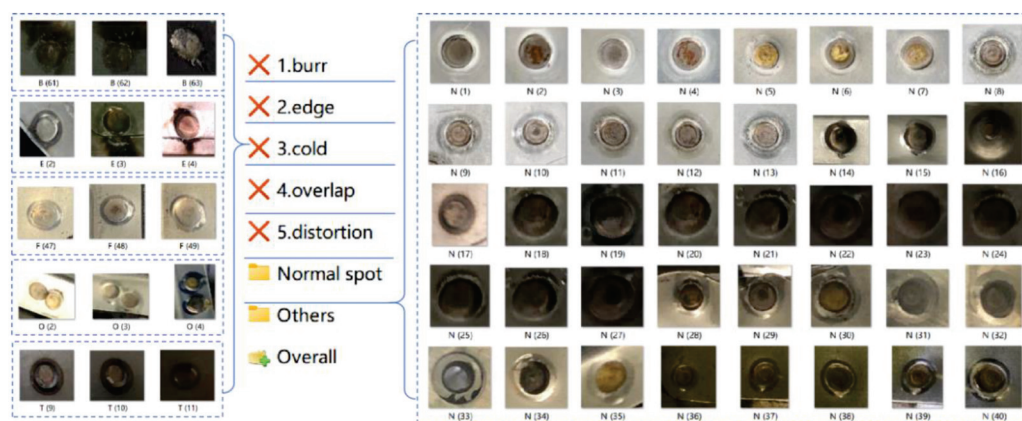


Figure 8. The weld spot dataset obtained from image segmentation using the WSDDM.

- (1) **Multi-Class Detection:** The dataset structure supports multi-class defect detection, which is ideal for models like YOLO and Faster R-CNN that are capable of handling multiple classes.
- (2) **Class Imbalance Consideration:** If any defect type is underrepresented, it may lead to class imbalance, affecting the model's ability to generalize across all defect types. Techniques like data augmentation, synthetic image generation, or class-weighted loss functions could mitigate this.
- (3) **Small Object Detection:** Given the small size of the welds, this dataset will test the model's ability to detect small objects, an area where Feature Pyramid Networks (FPNs) and attention mechanisms often prove beneficial.

4.4. Data Augmentation and Labeling

Before training the defect weld spot generation model, the cropped weld images were resized to 64×64 pixels and normalized to the range $[-1,1]$. The dataset includes normal welds and six types of weld defects, with a highly imbalanced class distribution. After simple affine transformations were applied to augment the data, the class distribution for the training, validation and testing sets is shown in Table 2. The defect weld spot generation model was trained using the Adam optimizer, with a learning rate of 0.002, and momentum parameters β_1 and β_2 set to 0.5 and 0.9, respectively. The batch size was 128, and the latent vector dimension was set to 128. For comparison, the ACGAN [37], cWGAN-GP [38], and BAGAN [39] models were trained on the same dataset, with hyperparameters largely kept consistent. Currently, there is a lack of standardized and accurate methods for evaluating generative model performance [40]. Therefore, this study uses both qualitative and quantitative methods to evaluate the performance of GAN models.

The qualitative evaluation assesses the performance of various GAN models by comparing the quality of generated weld spot image samples. This approach mainly relies on a visual assessment, focusing on the quality of selected images while potentially overlooking issues like overfitting and diversity. Figure 8 presents random samples generated by ACGAN, cWGAN-GP, BAGAN, and the proposed defect weld spot generation model, encompassing defect types such as “edge welding”, “overlap”, “cold solder joint”, “burr”, “distortion”, and “normal”.

Due to the high representation of normal weld samples in the dataset, each GAN model was able to generate high-quality images of normal welds. However, ACGAN and cWGAN-GP struggled with minority class defects, often producing images belonging to unintended classes, like “overlap” or “burn-through”. This limitation stems from the discriminator output structure and loss functions of these models, which are less effective for learning minority classes, making it challenging for the generator to synthesize accurate samples for these classes.

Both BAGAN and the defect weld spot generation model, however, effectively generated minority defect welds. Yet, the latent vectors produced by the AE model used to initialize BAGAN did not adequately differentiate between weld classes, complicating its training process. When weld images from different classes appeared visually similar, BAGAN struggled to generate images matching the intended class, such as “distortion” and “normal”. For classes with very few samples, like “burn-through,” BAGAN also failed to produce correct images. Additionally, BAGAN required an extensive fine-tuning of its network structure and hyperparameters. By contrast, the proposed defect weld spot generation model accurately generated images of specific defect types, demonstrating that it can effectively achieve data augmentation for defect classes, even when defect samples are limited.

Figure 9 illustrates the labeling process used in a deep learning dataset, where each image sample of weld spots is annotated with a classification label to differentiate between normal and defective spots. The left section displays various weld spot images grouped by visual characteristics, such as shape and condition, while the right section lists file names with corresponding binary labels, where '0' denotes a normal weld spot, and '1' represents a defective one.

					0024	0	0024.PNG	0	0
					0025	0	0025.PNG	0	0
					0026	0	0026.PNG	0	0
					0027	0	0027.PNG	0	0
					0028	1	0028.PNG	0	0028_label.PNG
					0029	0	0029.PNG	0	0
					0030	0	0030.PNG	0	0
					0031	0	0031.PNG	0	0
					0032	0	0032.PNG	0	0
					0033	0	0033.PNG	0	0
					0034	0	0034.PNG	0	0
					0035	0	0035.PNG	0	0
					0036	0	0036.PNG	0	0
					0037	0	0037.PNG	0	0
					0038	0	0038.PNG	0	0
					0039	0	0039.PNG	0	0
					0040	0	0040.PNG	0	0
					0041	0	0041.PNG	0	0
					0042	0	0042.PNG	0	0
					0043	0	0043.PNG	0	0
					0044	0	0044.PNG	0	0
					0045	0	0045.PNG	0	0
					0046	0	0046.PNG	0	0
					0047	0	0047.PNG	0	0
					0048	0	0048.PNG	0	0
					0049	1	0049.PNG	0	0049_label.PNG
					0050	1	0050.PNG	0	0050_label.PNG
					0051	0	0051.PNG	0	0
					0052	1	0052.PNG	0	0052_label.PNG
					0053	0	0053.PNG	0	0
					0054	0	0054.PNG	0	0
					0055	1	0055.PNG	0	0055_label.PNG
					0056	0	0056.PNG	0	0

Figure 9. Data augmentation and labeling.

The generated synthetic defect samples were used in the training of YOLOv8 as part of data augmentation, which can be considered a data preprocessing step. To evaluate the effectiveness of this approach, we conducted additional experiments. Without incorporating augmented data, the model's training performance was suboptimal, lacking a comparative and reference value. Thus, in addressing the challenges of small-sample object detection and severe class imbalance between positive and negative samples, this approach is essential.

4.5. Evaluation and Analysis

To objectively assess the performance of models used for detecting welding spot defects, several evaluation metrics are employed. These include GFLOPS (Giga Floating-point Operations Per Second), which quantifies the model's execution efficiency by measuring the number of floating-point operations performed per second. Model parameters are also considered to assess the model's size and complexity. Additionally, FPS (Frames Per Second) is used to gauge detection speed by recording the number of frames processed per second. Model accuracy is assessed using mean Average Precision (mAP), calculated as shown in Equation (4). The F_1 -score, representing a weighted average of precision and recall, is used to evaluate the model's overall performance and stability, with its calculation given in Equation (5).

$$mAP = \frac{\sum P_A}{N} \quad (4)$$

$$F_1 - S = 2 \times \frac{P \times R}{P + R} \quad (5)$$

In Equation (4), N denotes the total number of categories. P_A represents the area beneath the precision–recall curve, where recall values are plotted on the x-axis and precision values on the y-axis. The metric $mAP@0.5$ denotes the mean Average Precision at a CIoU threshold of 0.5. In Equation (5), precision measures the model's ability to distinguish between positive and negative samples, with higher precision indicating an improved distinction of negative samples. S represents score, P means precision, and R represents recall rate. Recall assesses the model's effectiveness in identifying positive samples, where higher recall signifies a more accurate detection of positive instances. The F_1 -S provides a balance between precision and recall, where a higher F_1 -S reflects a model with stronger overall performance and robustness.

The impact of integrating different attention modules on the model's detection accuracy is presented in Table 3. These experiments were conducted on the YOLOv8 model, in which the neck structure was reconfigured as shown in Figure 3. From the results in Table 3, it is evident that adding different attention modules slightly increases the computational load and parameter count. The model's $mAP@0.5$ value demonstrates some variability, where the addition of the channel attention module alone produces a moderate improvement in $mAP@0.5$; however, in specific instances, a reduction in accuracy is also observed.

When combining both channel and spatial attention within the CBAM module in YOLOv8, the network achieves further accuracy gains over the configuration with channel attention alone, with an increase of up to 0.9%. This suggests that spatial attention further enhances detection accuracy. Notably, the spatial attention mechanism in MC-FPN yields an impressive accuracy of 90.13%, representing a substantial improvement rate of 1.55%, while also providing faster detection speeds than the configuration with the CBAM module.

Based on the preceding data, we reconstruct the neck structure of YOLOv8 into MC-FPN, resulting in a 32% reduction in the model's parameter count and a 12.5% decrease

in computational complexity. Additionally, model accuracy, measured by mAP@0.5, improved by 1.9%. Replacing SPPF with SimSPPF had minimal effects on model accuracy, parameter count, and computational complexity, although it increased detection speed by 2 frames per second (FPS). The introduction of the Task-Aligned head module, while slightly increasing the parameter count and computational complexity, contributed an additional 1.3 percentage points in mAP@0.5 accuracy. Combining MC-FPN with the SPPF module improved model accuracy while reducing both computational complexity and parameter count. Overall, the enhanced YOLOv8 version, integrating MC-FPN, SimSPPF, and TA-Head modules, outperformed the baseline YOLOv8 model in terms of detection accuracy, computational complexity, and parameter count. On the same dataset, the modified YOLOv8 achieved a 3.3% increase in mAP@0.5, a 25% reduction in parameter count, and over 10% decrease in computational complexity.

Table 3. Comparison of the performance with different modules.

Models	mAP@0.5	Parameters (M)	GFLOPs	FPS
YOLOv5 [29]	79.65	13.33	24.6	135
YOLOv5 + SPPF	81.21	13.35	24.6	130
YOLOv7 [41]	84.87	12.56	28.2	105
YOLOv8 + SPP [42]	83.42	12.77	27.3	162
YOLOv8 + SPPF	83.50	12.87	27.3	104
YOLOv8 + SimSPPF [43]	83.18	12.74	27.2	121
YOLOv8 + BiFPN	88.90	9.65	28.8	118
YOLOv8 + SE	89.32	7.40	30.1	97
YOLOv8 + CBAM	90.55	7.39	25.6	90
YOLOv8 + EMA	89.30	8.28	25.2	68
YOLOv8 + Biform [44]	91.3	8.13	25.7	59
YOLOv8 + LSK attention	91.1	10.15	26.0	73
WSDDM	90.13	11.39	25.8	98

Figure 10 shows the validation sample results for model testing. This figure includes the detection results of weld spot samples and the corresponding heat map visualization. In the heat maps, red areas represent regions identified by the model as defective, while blue areas indicate normal regions. This color contrast provides an intuitive way to observe the areas the model focuses on, especially in response to defects. This indicates the model's high accuracy in detecting even minor defect locations. Additionally, the heat maps from different weld spot images demonstrate strong generalization ability, making the model suitable for various weld surface conditions. The middle five heat maps represent different feature extraction stages. This visualization demonstrates that, under a multi-cross layer network feature extraction and fusion approach, as the extracted feature information progresses to higher levels, the confidence increases gradually from low to high. The model successfully identifies and concentrates on weld spot defects, such as the edges of the welds and areas where cracks or depressions might be present. The model effectively captures the location of welding defects through highlighted (green) regions.

In practical production, some welding spots may be overlooked during manual inspection. To assess the model's generalization capability, specific weld point samples were intentionally left unannotated before training. The trained model is subsequently used to predict these samples, and the results demonstrate that it effectively identifies the missed welding spots, as shown in Figure 11.

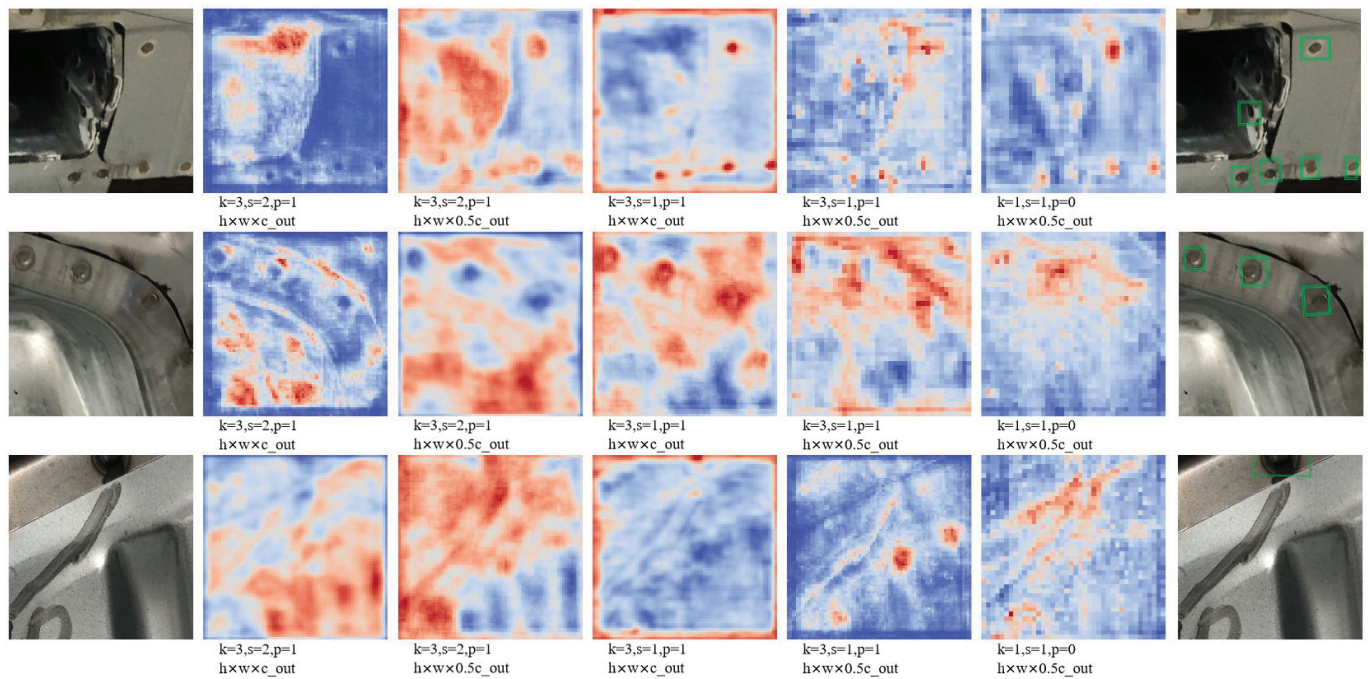


Figure 10. Visualization and validation sample results for model testing. The model effectively captures the location of welding defects through highlighted (green) regions.



Figure 11. Validation sample results for model generalization ability.

4.6. Experimental Pipeline and Detection Systems

Based on the methods and related technologies discussed in previous sections, we finally developed a Body-in-White welding spot quality detection system, integrating sensing technology, deep learning detection techniques, and communication technology. The system's architecture, shown in Figure 12, includes multiple modules: a data acquisition module, a vision inspection module, and a data management module. The system has been deployed on an actual automotive body welding production line, verifying the practicality and effectiveness of each module.

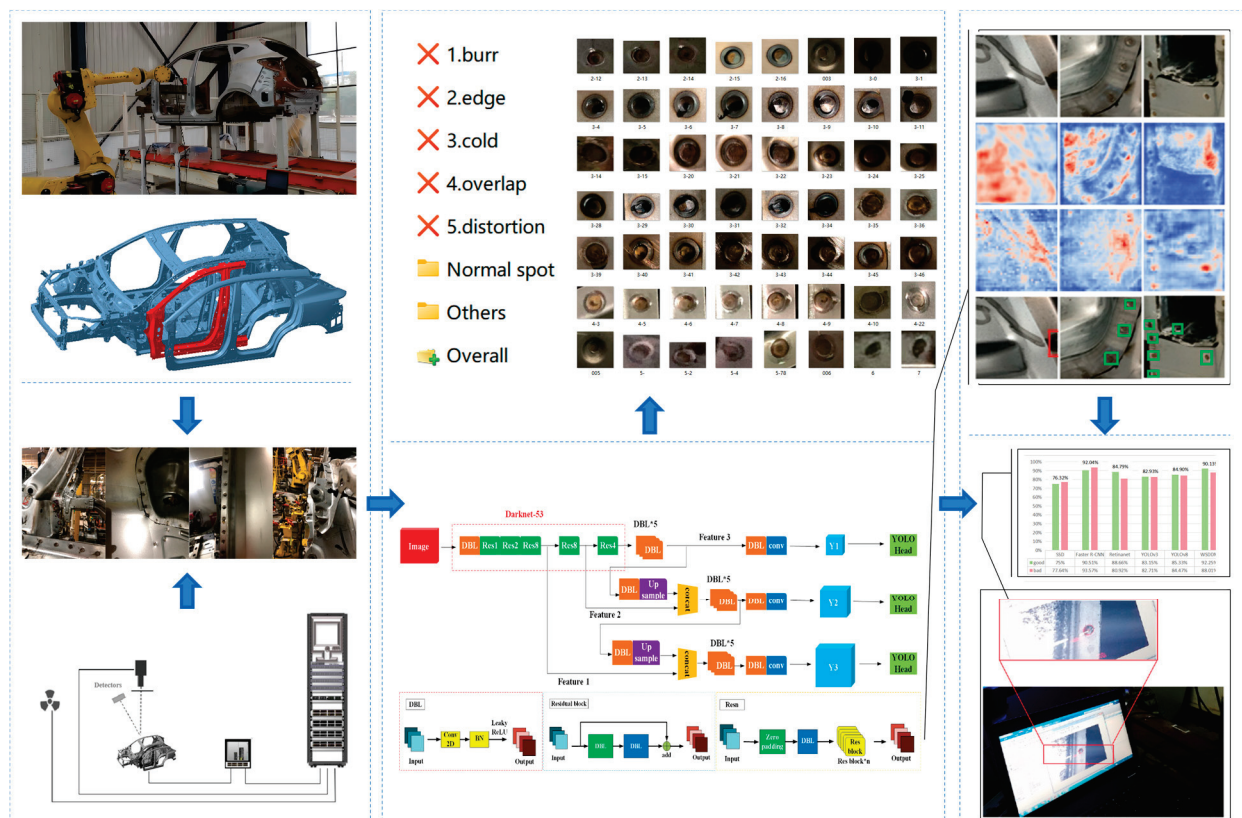


Figure 12. Experimental pipeline and integrated detection systems.

Data Acquisition Module: This system was validated on a production line in the welding workshop of an automotive company. The line includes four welding stations, each equipped with two FANUC robots, with welding guns powered by servo motors. Equipment on the automated line is integrated into the field PLC, ensuring high-reliability welding control.

Vision Inspection Module: This module utilizes pretrained model weights to evaluate welding spot quality. By implementing the vision inspection algorithms proposed in this study and integrating real-time welding parameter monitoring, this setup ensures adaptability under data constraints, leveraging deep learning visual models to enhance the accuracy and efficiency of weld quality inspection.

Data Management Module: Comprising a data visualization module, storage and analysis module, and alert module, the data management component serves as the user interface for interacting with the system. It allows users to query relevant information, including the dynamic visualization of data processing, the real-time display of weld spot data, historical data queries, and quality alert notifications. The system provides feedback to the user on the quality of each weld spot on the body, facilitating the further analysis and optimization of the welding process. Additionally, users can review weld quality based on alerts and optimize model learning through continuous quality feedback.

5. Conclusions and Future Works

In conclusion, this paper introduces a novel TA-YOLOv8-based model, termed the Weld Spot Defect Detection Model (WSDDM), for the detection and classification of welding spot defects in car body production. Our proposed model effectively addresses challenges posed by small sample sizes, class imbalance, and noisy data, often encountered in Body-in-White welding spot datasets. The TA-YOLOv8 network employs a Task-Aligned (TA)

detection head and a cross-scale feature fusion strategy, which significantly enhances sensitivity to fine-grained features, facilitating the detection of small, dense welding spots with high accuracy. The experimental results demonstrate that the WSDDM achieves a detection speed improvement of 2.8% over YOLOv8 and an mAP@50-95 increase of 1.35%, along with a 0.1226 rise in the recall rate. Furthermore, the WSDDM's lightweight DarkNet53 backbone enables efficient, accurate real-time processing suitable for industrial applications. A comparative analysis shows that while Faster R-CNN achieved the highest detection accuracy, it came at a substantial time cost, whereas the WSDDM maintained high detection precision at a fraction of the computational expense. The model's robustness across varying weld spot shapes and lighting conditions underscores its generalization capability, making it a viable and reliable tool for automated welding spot quality inspection in real-world production environments.

To address the issue of the loss of feature information for spot welding during feed forward convolution computations, this paper improves the YOLOv8 model from the perspective of the backbone network, aiming to enhance the detection accuracy and speed of the small-target welding spot. A defective welding spot generation model is established based on autoencoder initialization and the theory of balanced generative adversarial networks, facilitating accurate data augmentation in imbalanced datasets. Finally, the effectiveness of the proposed method is validated using an image dataset collected from the field. The main conclusions are as follows:

- (1) The proposed small-target solder joint detection model outperforms several typical object detection algorithms in terms of mean Average Precision (mAP) and detection speed. The cross-layer feature fusion structure effectively integrates and refines multi-scale features, utilizing robust low-level features to improve the detection accuracy of small-scale targets. The lightweight network DarkNet53 enhances inference speed while maintaining accuracy.
- (2) Qualitative and quantitative comparative analysis with typical GAN models indicates that the proposed defect welding spot generation model can produce high-quality and diverse defect images in highly imbalanced datasets.
- (3) The ablation experiment results demonstrate that the Task-Aligned method, data augmentation, and feature pre-training all contribute to the deep learning model's ability to learn discriminative features from solder joint images. For the common weld spots quality categories on the white body production line, the performance of the proposed TA-YOLOv8 model generally surpasses that of other foundational visual models.

In deep visual models, false positives (FPs) and false negatives (FNs) are common issues when evaluating performance. To analyze and explain the model's failure in such cases, a systematic experimental analysis is often required. The following are steps and methods for analyzing and explaining the reasons behind false positives and false negatives, including Error Classification Analysis, Model Output Analysis and case studies. By following these steps, we are able to conduct an in-depth analysis of false positives and false negatives in the model, understand their root causes, and improve the model's performance. Through dataset analysis, visualization, model architecture optimization, and other methods, the model's weaknesses can be identified, providing a foundation for further improvements. In summary, a thorough analysis of the model's failure cases helps to quantify its limitations and guide subsequent optimization efforts.

Future work to further improve the application of the YOLOv8 algorithm in the field of solder joint defect detection could include the following directions:

- (1) Enhanced Multi-Scale Feature Fusion: This involves the further optimization of cross-layer feature fusion mechanisms, utilizing higher-resolution low-level features to improve the accuracy of small defect detection. Improvements in the selection and fusion strategy of high-level features could also enhance the model's robustness in complex backgrounds.
- (2) Efficient Lightweight Network Architectures: Research into more efficient lightweight network architectures, such as hybrid convolutions or dynamic convolutions, could reduce inference time and computational cost while maintaining detection accuracy, making the model more suitable for resource-constrained, real-time detection tasks.
- (3) Handling Imbalanced and Small Sample Datasets: Industrial applications often face imbalanced datasets. Future work could focus on some other generative networks and transfer learning to augment defect data. By generating diverse defect images, these techniques could alleviate small-sample issues and improve model generalization.

These improvements would help further enhance the applicability and accuracy of YOLOv8 in welding spot defect detection, making it better suited to meet the demands of industrial production environments.

Author Contributions: Conceptualization, W.L. and J.H.; methodology, W.L., M.J. and S.Z. (Shuo Zhang); data collection, W.L., J.Q., and J.H.; writing—original draft preparation, S.Z. (Shuo Zhang) and W.L.; writing—review and editing, J.Q.; supervision, M.J.; validation, W.L. and S.Z. (Siyu Zhu); funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by National Natural Science Foundation of China (52475270, 52035007, U23B20102), and Xie Yubai Design Scientific Research Foundation (XYB-DS-202401).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kah, P.; Rajan, R.; Martikainen, J.; Suoranta, R. Investigation of Weld Defects in Friction-Stir Welding and Fusion Welding of Aluminium Alloys. *J. Mater. Sci. Mater. Eng.* **2015**, *10*, 26. [CrossRef]
2. Den Uijl, N.; Moolevliet, T.; Mennes, A.; Van Der Ellen, A.A.; Smith, S.; Van Der Veldt, T.; Okada, T.; Nishibata, H.; Uchihara, M.; Fukui, K. Performance of Resistance Spot-Welded Joints in Advanced High-Strength Steel in Static and Dynamic Tensile Tests. *Weld. World* **2012**, *56*, 51–63. [CrossRef]
3. Chao, Y.J. Ultimate Strength and Failure Mechanism of Resistance Spot Weld Subjected to Tensile, Shear, or Combined Tensile/Shear Loads. *J. Eng. Mater. Technol.* **2003**, *125*, 125–132. [CrossRef]
4. Duda, R.O.; Hart, P.E. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* **1972**, *15*, 11–15. [CrossRef]
5. Oualid, D.; Khadidja, M.; Kahina, A. Incremental Circle Hough Transform: An Improved Method for Circle Detection. *Optik* **2017**, *133*, 17–31. [CrossRef]
6. Liang, Q.K.; Long, J.Y.; Nan, Y.; Coppola, G.; Zou, K.L.; Zhang, D.; Sun, W. Angle Aided Circle Detection Based on Randomized Hough Transform and Its Application in Welding Spots Detection. *Math. Biosci. Eng. MBE* **2019**, *16*, 1244–1257. [CrossRef]
7. Ronghua, C.; Yoshiaki, O. Numerical Analysis of Freezing Controlled Penetration Behavior of the Molten Core Debris in an Instrument Tube with MPS. *Ann. Nucl. Energy* **2014**, *71*, 322–332. [CrossRef]

8. Cho, W.I.; Woizeschke, P. Analysis of Molten Pool Dynamics in Laser Welding with Beam Oscillation and Filler Wire Feeding. *Int. J. Heat Mass Transf.* **2021**, *164*, 120623. [CrossRef]
9. Ruisz, J.; Biber, J.; Loipetsberger, M. Quality Evaluation in Resistance Spot Welding by Analysing the Weld Fingerprint on Metal Bands by Computer Vision. *Int. J. Adv. Manuf. Technol.* **2007**, *33*, 952–960. [CrossRef]
10. Myrach, P.; Jonietz, F.; Meinel, D.; Suwala, H.; Ziegler, M. Calibration of Thermographic Spot Weld Testing with X-Ray Computed Tomography. *Quant. InfraRed Thermogr. J.* **2017**, *14*, 122–131. [CrossRef]
11. Dell’Avvocato, G.; Palumbo, D. Thermographic Procedure for the Assessment of Resistance Projection Welds (RPW): Investigating Parameters and Mechanical Performances. *J. Adv. Join. Process.* **2024**, *9*, 100177. [CrossRef]
12. Valentin, P.; Kounalakis, T.; Nalpantidis, L. Weld Classification Using Gray Level Co-Occurrence Matrix and Local Binary Patterns. In Proceedings of the 2018 IEEE International Conference on Imaging Systems and Techniques (IST), Krakow, Poland, 16–18 October 2018; pp. 1–6.
13. Javaheri, E.; Kumala, V.; Javaheri, A.; Rawassizadeh, R.; Lubritz, J.; Graf, B.; Rethmeier, M. Quantifying Mechanical Properties of Automotive Steels with Deep Learning Based Computer Vision Algorithms. *Metals* **2020**, *10*, 163. [CrossRef]
14. Alghannam, E.; Lu, H.; Ma, M.; Cheng, Q.; Gonzalez, A.A.; Zang, Y.; Li, S. A Novel Method of Using Vision System and Fuzzy Logic for Quality Estimation of Resistance Spot Welding. *Semantic Scholar. Symmetry* **2019**, *11*, 990. [CrossRef]
15. Younes, D.; Alghannam, E.; Tan, Y.; Lu, H. Enhancement in Quality Estimation of Resistance Spot Welding Using Vision System and Fuzzy Support Vector Machine. *Symmetry* **2020**, *12*, 1380. [CrossRef]
16. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size. *arXiv* **2016**, arXiv:1602.07360.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv*, **2015**, arXiv:1512.03385.
18. Li, Y.; Huang, H.; Xie, Q.; Yao, L.; Chen, Q. Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD. *Appl. Sci.* **2018**, *8*, 1678. [CrossRef]
19. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1440–1448.
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
21. Kim, S.; Hwang, I.; Kim, D.Y.; Kim, Y.M.; Kang, M.; Yu, J. Weld-Quality Prediction Algorithm Based on Multiple Models Using Process Signals in Resistance Spot Welding. *Metals* **2021**, *11*, 1459. [CrossRef]
22. Liu, W.; Hu, J.; Qi, J. Resistance Spot Welding Defect Detection Based on Visual Inspection: Improved Faster R-CNN Model. *Machines* **2025**, *13*, 33. [CrossRef]
23. Wang, Z.; Li, T. A Lightweight CNN Model Based on GhostNet. *Comput. Intell. Neurosci.* **2022**, *2022*, 8396550. [CrossRef]
24. Adarsh, P.; Rath, P.; Kumar, M. YOLO V3-Tiny: Object Detection and Recognition Using One Stage Improved Model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694.
25. Safaldin, M.; Zaghdien, N.; Mejdoub, M. An Improved YOLOv8 to Detect Moving Objects. *IEEE Access* **2024**, *12*, 59782–59806. [CrossRef]
26. Wang, X.; Gao, H.; Jia, Z.; Li, Z. BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. *Sensors* **2023**, *23*, 8361. [CrossRef] [PubMed]
27. Wang, J.; Zhao, H. Improved YOLOv8 Algorithm for Water Surface Object Detection. *Sensors* **2024**, *24*, 5059. [CrossRef] [PubMed]
28. Wu, T.; Dong, Y. YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition. *Appl. Sci.* **2023**, *13*, 12977. [CrossRef]
29. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
30. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021.
31. Neubeck, A.; Van Gool, L. Efficient Non-Maximum Suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR’06), Hong Kong, China, 20–24 August 2006.
32. Wang, X.; Song, J. ICIoU: Improved Loss Based on Complete Intersection Over Union for Bounding Box Regression. *IEEE Access* **2021**, *9*, 105686–105695. [CrossRef]
33. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.

34. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016.
35. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
36. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 318–327. [CrossRef] [PubMed]
37. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
38. Zheng, M.; Li, T.; Zhu, R.; Tang, Y.; Tang, M.; Lin, L.; Ma, Z. Conditional Wasserstein Generative Adversarial Network-Gradient Penalty-Based Approach to Alleviating Imbalanced Data Classification. *Inf. Sci.* **2020**, *512*, 1009–1023. [CrossRef]
39. Xu, Q.; Huang, G.; Yuan, Y.; Guo, C.; Sun, Y.; Wu, F.; Weinberger, K. An Empirical Study on Evaluation Metrics of Generative Adversarial Networks. *arXiv*, 2018; arXiv:1806.07755.
40. Borji, A. Pros and Cons of GAN Evaluation Measures. *arXiv* **2018**, arXiv:1806.07755. [CrossRef]
41. Wang, Y.; Wang, H.; Xin, Z. Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7. *IEEE Access* **2022**, *10*, 133936–133944. [CrossRef]
42. Kaiming, H.; Xiangyu, Z.; Shaoqing, R.; Jian, S. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
43. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
44. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R. BiFormer: Vision Transformer with Bi-Level Routing Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Research on a Point Cloud Registration Method Based on Dynamic Neighborhood Features

Xinrui Liu, Rutao Wang and Zongsheng Wang *

College of Ocean Science and Engineering, Shandong University of Science and Technology,
Qingdao 266590, China; 202283190007@sdust.edu.cn (X.L.); rt_wyyx@163.com (R.W.)

* Correspondence: 13061169159@163.com

Abstract: This paper introduces a method that can enhance the accuracy and efficiency of point cloud data registration. This method selects the centroid of the point cloud as the feature point and uses the projected distance of this feature point within the dynamic neighborhood to other points as the feature information. Through this feature information, it accomplishes the registration of two sets of point cloud data. This method increases the density and integrity of point cloud data, improves the accuracy and robustness of point cloud registration, and the selection of feature points reduces the computational load thereby enhancing processing efficiency. The introduction of the dynamic neighborhood enables the method to flexibly handle point cloud data of different scales and densities. Experimental results show that the proposed method has good performance in terms of accuracy and efficiency for achieving point cloud data registration and dealing with data under various complex conditions and can effectively improve the effect of point cloud data registration and fusion.

Keywords: point cloud registration; feature point; dynamic neighborhood

1. Introduction

In recent years, the rapid development of light detection and ranging (LiDAR) technology has provided strong support for the acquisition and processing of point cloud data, and has promoted technical advancements and expanded applications in related fields [1]. Currently, LiDAR is widely used in autonomous driving, intelligent robotics, drones, surveying and mapping/geographic information systems, construction engineering, environmental monitoring, medical treatment, and other fields [2,3]. LiDAR offers advantages such as high resolution, long detection range, high real-time performance, and strong anti-interference capabilities. Due to factors such as the shape of the object, the scanning angle of the LiDAR system, and the accuracy of the equipment, point cloud data from different poses and angles must be registered and fused in order to obtain more complete data. The main goal of point cloud registration is to combine point clouds acquired from different viewpoints into a unified and complete three-dimensional model [4]. Many researchers have employed different methods and techniques to optimize this process. For example, the iterative closest point (ICP) method, introduced by Chen and Medioni [5] and Besl and McKay [6], minimizes the distance between two point clouds by calculating rotation and translation transformations. This method is simple and efficient, but it requires a good initial pose to avoid registration failure. Xu G X [7] proposed an improved iterative closest point (ICP) method, combining random sample consensus (RANSAC), intrinsic

shape signatures (ISSs), and 3D shape context (3DSC) features. Although this method has improved registration accuracy and efficiency, it is still highly dependent on the initial pose and sensitive to parameters. Dior Arger et al. [8] proposed the four-point congruent sets (4PCSs) method, which utilizes co-planar four-point sets for fast registration. However, the 4PCSs method has high complexity and poor performance when the overlap rates of the two sets of point cloud data are low. Therefore, Pascal Willy Theiler et al. [9] proposed an improved method, key-point-based four-points congruent sets (K4PCSs), which refines the point cloud using key points to address issues such as low registration area density, large registration scanning angles, and low computational and memory efficiency. Peter Biber et al. [10] proposed a three-dimensions-normal distributions transform (3D-NDT) method, introducing the normal distributions transform (NDT) to three-dimensional space for the first time, and validated its performance through registration of real-world mining data. This method is faster and more reliable than the traditional ICP method, but it is still sensitive to the initial pose and more susceptible to noise. To mitigate the significant impact of the initial pose on point cloud registration results, many studies employ a coarse-to-fine strategy. This involves performing a coarse registration to obtain a sufficiently accurate initial pose, followed by a fine registration for further refinement [11,12]. While this approach can lead to higher registration accuracy, it often comes at the cost of increased computational time.

To address the limitations of existing point cloud registration methods, which often suffer from insufficient accuracy and low efficiency, this paper proposes a novel registration method based on single LiDAR dynamic neighborhood features. In this approach, the centroid of the point cloud serves as the feature point, and the distance projection between this feature point and other points within a dynamic neighborhood is extracted as feature information. By calculating the chi-square distance between these feature vectors [13], the registration of two point cloud datasets is achieved. The size of the dynamic neighborhood is adaptively determined based on curvature, density, and curvature-based wavelet transform detail coefficients [14]. This method effectively captures the local geometric characteristics of the point cloud and accurately identifies optimal matching pairs. Compared to existing point cloud registration algorithms, the proposed point cloud registration algorithm in this paper can achieve high-precision point cloud registration more quickly, resulting in significant improvements in both registration accuracy and speed.

2. Methods

The main process of the proposed point cloud data registration method based on dynamic neighborhood features is illustrated in Figure 1. After acquiring the source and target point cloud data, statistical outlier removal is first performed to filter out noise. Subsequently, the point cloud is divided into small, equally sized voxels, and the centroid of the points within each voxel is selected as a feature point. A dynamic neighborhood is determined for each feature point based on its local average curvature, point cloud density, and detail coefficients from a curvature-based wavelet transform. The local feature descriptor for each feature point is then constructed as a cumulative histogram of the projected distances from the remaining points within its dynamic neighborhood onto the feature point along the direction of maximum principal curvature. Feature point pairing is achieved by calculating the chi-squared distance between the cumulative histograms of the projected distances for feature points in the two point clouds. The pair with the minimum chi-squared distance is considered the best match and retained. Matching pairs with chi-squared distances exceeding a predefined threshold are discarded to improve matching accuracy. Finally, the transformation matrix between the source and target point

clouds is computed using the retained best matching pairs. This transformation matrix is then applied to register the source point cloud to the target point cloud [15,16].

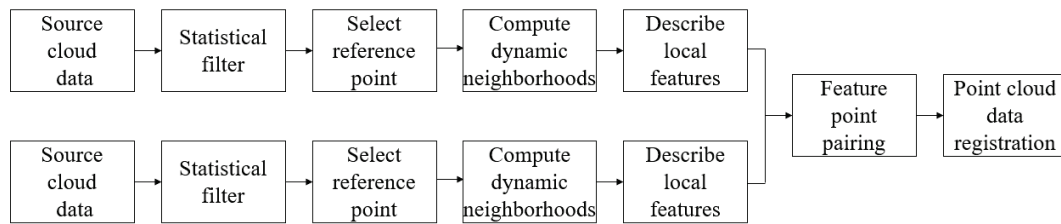


Figure 1. Flowchart of the dynamic neighborhood-based feature point registration method.

2.1. Introduction to LiDAR

In this study, some of the point cloud data were collected using the Livox-Mid360 LiDAR (Shenzhen, China). The physical object and main performance parameters of the Livox-Mid360 LiDAR are shown in Figure 2 and Table 1.



Figure 2. Lidar equipment.

Table 1. Livox-Mid360 LiDAR main specifications.

Specifications	Value
Weight	265 g
Dimensions	65 (L) × 65 (W) × 60 (H) mm
Communication Interface	100 BASE-TX Ethernet
Field of View (FOV)	Horizontal 360°, Vertical −7°~52°
Point Cloud Output	200,000 points/second
Range	0.1~70 m
Power	25 W
Resolution	<0.15°
Frame Rate	10 Hz

2.2. Statistical Filtering

Point cloud data acquired by LiDAR scanning typically exhibit non-uniform density, and measurement errors often introduce sparse outliers. These outliers can lead to inaccuracies in local feature descriptions (such as normal vectors or curvatures at sampling points), which subsequently affect downstream processes like point cloud registration. Exploiting the spatial sparsity of outliers, statistical filtering [17] is employed for their removal. For each point $P_i(x_i, y_i, z_i)$ in the point cloud data, the k nearest points are selected as its

neighborhood set, and the average distance d_i between point P_i and its neighbors is then calculated using the following equation:

$$d_i = \frac{1}{k} \sum_{i=1}^k S_i \quad (1)$$

where S_i is the distance between point P_i and other points $P_m(x_m, y_m, z_m)$ in its neighborhood. To calculate S_i the following equation is used:

$$S_i = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2 + (z_i - z_m)^2} \quad (2)$$

The average value of d_i is $\bar{d}$, which is calculated using the following equation:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (3)$$

The standard deviation is σ , which is calculated using the following equation:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2} \quad (4)$$

where n is the number of points in the point cloud data.

If the average distance d_i of a point satisfies $\bar{d} - \rho \cdot \sigma \leq d_i \leq \bar{d} + \rho \cdot \sigma$, the point is retained; otherwise, the point is considered an outlier and removed. Here, ρ is a multiplier of the standard deviation.

2.3. Local Feature Point Selection

Given an input point cloud dataset, a minimum bounding box (MBB) [18] is constructed based on the maximum and minimum coordinates in the X, Y, and Z directions. Specifically, the MBB is defined by x_{max} , y_{max} , z_{max} , x_{min} , y_{min} , and z_{min} , representing the extrema of the point cloud in each dimension. The resulting MBB tightly encloses the entire point cloud, with edge lengths denoted as L_x , L_y , and L_z . This is represented by the following equation:

$$\begin{cases} L_x = x_{max} - x_{min} \\ L_y = y_{max} - y_{min} \\ L_z = z_{max} - z_{min} \end{cases} \quad (5)$$

The edges of the minimum bounding box (MBB) are divided into n_1 , n_2 , and n_3 equal segments along the X, Y, and Z axes, respectively. Consequently, the MBB is partitioned into $n_1 \cdot n_2 \cdot n_3$ smaller voxels. The edge lengths of each voxel are denoted by m_1 , m_2 , and m_3 , as shown by the following equation:

$$\begin{cases} m_1 = \frac{L_x}{n_1} \\ m_2 = \frac{L_y}{n_2} \\ m_3 = \frac{L_z}{n_3} \end{cases} \quad (6)$$

For each voxel, the centroid $\bar{p}$ of the enclosed point cloud data is computed. The point within each voxel that is nearest to the centroid $\bar{p}$ is then selected as the characteristic point P_c . This is shown in the following equation:

$$\bar{p} = \frac{1}{N_j} \sum_{i=1}^{N_j} p_i \quad (7)$$

where N_j represents the number of points within each voxel, with j denoting the index of the j -th voxel.

2.4. Dynamic Neighborhood Calculation

Following the selection of feature points, it is necessary to determine a dynamic neighborhood for each feature point to characterize its local geometric information [19]. To define the size of this dynamic neighborhood, we introduce three parameters: local point density, local mean curvature, and wavelet transform detail coefficients based on the curvature.

Point cloud density, representing the degree of sparseness or denseness of point cloud data in space, is a crucial parameter for determining the size of the dynamic neighborhood [20]. In regions with high point cloud density a larger number of points are contained within a unit volume, indicating richer point cloud information. In such cases, if the neighborhood size is excessively large it may encompass too many points, leading to a smoothing effect on the computed local features and a loss of critical detail information. Therefore, the dynamic neighborhood in dense point cloud regions should be appropriately reduced to ensure that the calculation of local features can describe finer geometric structures and avoid feature generalization. Smaller neighborhoods can better capture subtle variations in regions with rich surface details (e.g., textures, folds). Conversely, in regions with a sparse point cloud distribution the number of points contained within a unit volume is limited, resulting in relatively insufficient point cloud information. If the neighborhood is too small it may lead to an insufficient number of points to support the calculation of local features, introducing noise or instability [21]. Thus, in sparse regions, the dynamic neighborhood should be appropriately enlarged to ensure that there are enough points within the neighborhood to compute local features. In spacious background areas or regions with missing point clouds a larger neighborhood can compensate for the lack of information caused by point cloud sparsity. In summary, point cloud density indirectly determines the size of the dynamic neighborhood by influencing the number of points within it: dense regions require smaller dynamic neighborhoods to preserve details while sparse regions require larger dynamic neighborhoods to ensure computational stability.

Local mean curvature is an important metric for describing the geometric characteristics of point cloud surfaces, reflecting the average bending of the point cloud surface in the vicinity of a given point. Compared to single curvature measures, mean curvature provides a more comprehensive description of local surface geometry. In regions with high mean curvature (e.g., edges, corners, sharp features), the geometric variations in the point cloud surface are significant, resulting in larger mean curvature values. These regions typically contain rich geometric detail information, such as object contours and sharp edges, which play a crucial role in tasks like point cloud registration, segmentation, and recognition [22,23]. To preserve these detailed features, the dynamic neighborhood should be appropriately reduced to avoid feature smoothing caused by an excessively large neighborhood, preventing the loss of important geometric information. Conversely, in regions with low mean curvature (e.g., planar surfaces, smooth surfaces) the geometric variations in the point cloud surface are gradual, resulting in smaller mean curvature values. The geometric features in these regions are relatively uniform, without significant detail variations. In such cases, the dynamic neighborhood can be appropriately enlarged to capture more local geometric information while enhancing the robustness of feature computation.

Curvature-based wavelet transform detail coefficients provide a further extension of local curvature information, capturing local geometric details and curvature variations in point cloud data through multi-scale analysis. In regions with high curvature, the

geometric changes on the point cloud surface are significant and the wavelet transform can extract prominent high-frequency information, resulting in larger detail coefficient values. These high-frequency components reflect the complexity and detailed variations in local geometric features. In such instances, the dynamic neighborhood should be appropriately reduced to ensure accurate capture and preservation of these detailed features. Conversely, in regions with low curvature, the geometric changes on the point cloud surface are gradual and the wavelet transform extracts less high-frequency information, leading to smaller detail coefficient values [24]. The geometric features in these regions are relatively uniform, with no significant detail variations. In these cases, the dynamic neighborhood can be appropriately enlarged. This approach captures more local geometric information while enhancing the stability of feature computation.

Therefore, the size of the dynamic neighborhood needs to comprehensively consider the influence of point cloud density, local curvature, and detail coefficients. In dense or high-curvature regions, the point cloud information is rich and the geometric features are complex, requiring a smaller dynamic neighborhood to preserve details. In sparse or low-curvature regions, the point cloud information is insufficient and the geometric features are simple, necessitating a larger dynamic neighborhood to ensure computational stability [25]. As a complement to local curvature, the detail coefficients further refine the neighborhood adjustment strategy, ensuring accurate capture of geometric features in point cloud data across multiple scales.

The following details the computational steps involved in determining the dynamic neighborhood.

First, the principal direction is computed for each local feature point P_c within each small grid cell. The covariance matrix T is constructed using the other points within the same grid cell. The principal directions and curvature information at a point can be calculated through the covariance matrix, as shown in Equation (8) which is as follows:

$$T = \frac{1}{N_j} \sum_{i=1}^{N_j} (p_i - \bar{p})(p_i - \bar{p})^T \quad (8)$$

The covariance matrix T is then subjected to eigenvalue decomposition, yielding eigenvalues λ_1 , λ_2 , and λ_3 (assuming $\lambda_1 > \lambda_2 > \lambda_3$), with corresponding eigenvectors v_1 , v_2 , and v_3 . From this, the dominant principal direction v_1 and the secondary principal direction v_2 of the local feature point P_c are obtained.

Similarly, the eigenvalues λ_{i_1} , λ_{i_2} , and λ_{i_3} (assuming $\lambda_{i_1} > \lambda_{i_2} > \lambda_{i_3}$) for each of the other points within the same grid cell can be computed by constructing a covariance matrix T_i . The curvature C_i of the other points within the grid cell is calculated as shown in Equation (9), which is as follows:

$$C_i = \frac{\lambda_{i_3}}{\lambda_{i_1} + \lambda_{i_2} + \lambda_{i_3}} \quad (9)$$

The average curvature $\bar{C}_i$ of the points within the grid cell is obtained from Equation (10), which is as follows:

$$\bar{C}_i = \frac{1}{N_j} \sum_{i=1}^{N_j} C_i \quad (10)$$

The point cloud density ρ_i within a grid cell refers to the number of points per unit volume, as shown in Equation (11) which is as follows:

$$\rho_i = \frac{N_j}{m_1 \cdot m_2 \cdot m_3} \quad (11)$$

The curvature-based wavelet detail coefficient W_i reflects local shape variations. The following outlines the computation of W_i .

The curvature C_i of each point within the grid cell is used as the input signal x for wavelet transformation. The input order of the curvature values is determined based on the sequence of points along the maximum principal direction v_1 and the second maximum principal direction v_2 within the grid cell, which can be shown as follows:

$$x = [C_1 C_2 C_3 \cdots C_n] \quad (12)$$

We choose Daubechies (db4) as the wavelet base for discrete wavelet transform of curvature. The db4 wavelet transform requires two filters: a low-pass filter for extracting low-frequency components and a high-pass filter for extracting high-frequency components.

The input curvature signal is processed using a db4 wavelet filter for low-pass (approximation) and high-pass (detail) filtering. The low-pass filtering yields the approximation coefficients $A_1[n]$, while the high-pass filtering yields the detail coefficients $D_1[n]$. After downsampling both sets of coefficients (taking every other sample), the curvature signal x , after the first level of transformation, is decomposed as shown in Equation (13):

$$x[n] = A_1[n] + D_1[n] \quad (13)$$

where $A_1[n]$ represents the approximation coefficients of the first layer (i.e., the low-frequency components), and $D_1[n]$ represents the detail coefficients of the first layer (i.e., the high-frequency components) [24].

After the first-level transform is completed, the approximation coefficients $A_1[n]$ are further subjected to wavelet transform, resulting in the second-level approximation coefficients $A_2[n]$ and the second-level detail coefficients $D_2[n]$. The equation is as follows:

$$A_1[n] = A_2[n] + D_2[n] \quad (14)$$

The above process is repeated until the desired decomposition level J is reached, resulting in the decomposition of the input curvature signal into the approximation coefficients $A_J[n]$ and the detail coefficients $D_1[n], D_2[n] \dots D_J[n]$. The input signal x is ultimately decomposed as shown in Equation (15):

$$x = A_J[n] + D_1[n] + D_2[n] + \cdots + D_J[n] \quad (15)$$

The detail coefficient components are extracted to calculate the curvature-based wavelet transform detail coefficients, W_i , as shown in Equation (16):

$$W_i = \sum_{j=1}^J \sum_n |D_j[n]|^2 \quad (16)$$

Finally, the dynamic neighborhood L of the feature point P_c within each voxel is determined based on the point cloud data density ρ_i , the average curvature $\bar{C}_i$, and the curvature-based wavelet transform detail coefficients W_i , as defined in Equation (17):

$$L = k \cdot \frac{1}{\sqrt{\rho_i}} \cdot \frac{1}{1 + \alpha \cdot |W_i| + \beta \cdot |\bar{C}_i|} \quad (17)$$

where k is the scaling factor used to control the overall size of the dynamic neighborhood, which can be determined by the average point spacing within the neighborhood, typically ranging from 0.8 to 1.5. α and β are weighting parameters that balance the effects of curvature and detail coefficients on neighborhood size. Increasing α and β will enhance the sensitivity of W_i and $\bar{C}_i$, while decreasing them will suppress this sensitivity. Generally, α can take values from 0.1 to 1, and β typically ranges from 0.8 to 1.5. ρ_i represents the local point cloud density at point i . W_i is the detail coefficient derived from a wavelet transform based on curvature. $\bar{C}_i$ denotes the average curvature of points within the neighborhood. When the local point cloud density is high, or when the curvature variation is complex and the average curvature is relatively high, it will cause L to decrease to avoid losing detail information. Conversely, L will increase in order to capture more detailed information.

2.5. Description of Local Features

Following feature point extraction from the point cloud and computation of dynamic neighborhood radii, a feature descriptor is generated by constructing a cumulative histogram of projection distances between each feature point and its neighbors within the dynamically sized neighborhood.

For each feature point, the Euclidean distance to every point P_i within its dynamic neighborhood L is computed. The resulting distance vector is then projected onto the feature point's principal eigenvector v_1 , corresponding to the largest eigenvalue, yielding a projection length d_p , as shown in Equation (18):

$$d_p = (P_i - P_c) \cdot v_1 \quad (18)$$

Upon obtaining the projected distances for all points within the neighborhood, a cumulative projected distance histogram is constructed for each feature point. The histogram's range is defined by the minimum projected distance, d_{pmin} , and the maximum projected distance, d_{pmax} , which serve as the lower and upper bounds, respectively. This range is then partitioned into a set of equally sized bins, and the frequency of projected distances falling within each bin is accumulated. The resulting histogram represents the distribution of projected distances with respect to the orientation defined by v_1 and is shown in the following equation:

$$H(i) = \text{count}(d \in [d_p, d_{p+1}]) \quad (19)$$

where $H(i)$ represents the number of points within the i -th interval $[d_p, d_{p+1}]$.

The cumulative projected distance histogram provides a local feature descriptor for each feature point.

2.6. Pairing of Feature Points

Following the derivation of cumulative projected distance histograms for feature points in both the source and target point cloud datasets, point correspondence is estab-

lished by minimizing the chi-squared distance, denoted as D_c , between the respective histograms, the formulation of which is detailed in Equation (20):

$$D_c = \chi^2(\hat{H}_1, \hat{H}_2) = \frac{1}{2} \sum_{i=1}^n \frac{(\hat{H}_1(i) - \hat{H}_2(i))^2}{\hat{H}_1(i) + \hat{H}_2(i)} \quad (20)$$

The optimal correspondence is identified when the chi-squared distance, D_c , is minimized. For each feature point in the first dataset, the feature point in the second dataset exhibiting the minimum chi-squared distance is designated as the best matching candidate. To enhance matching accuracy, putative correspondences with a chi-squared distance exceeding a predefined threshold are subsequently discarded.

Employing the chi-squared distance as a metric for quantifying the dissimilarity between cumulative projected distance histograms enables effective registration of source and target point clouds, thereby yielding optimal point correspondences.

2.7. Solve Transformation Matrix

After obtaining several sets of optimal correspondences, a rigid transformation is computed from the source point cloud data to the target point cloud data. This rigid transformation allows for the registration of the source point cloud data into the coordinate system of the target point cloud data [26].

Initially, the centroids of both the source point cloud data (P) and the target point cloud data (Q) are calculated. The centroid calculation is shown in Equation (21):

$$\begin{aligned} C_P &= \frac{1}{N_P} \sum_{i=1}^{N_P} p_i \\ C_Q &= \frac{1}{N_Q} \sum_{i=1}^{N_Q} q_i \end{aligned} \quad (21)$$

where N_P and N_Q represent the number of points in the source point cloud data and the target point cloud data, respectively.

The corresponding pairs in both datasets are then translated to the origin, as shown in Equation (22):

$$\begin{aligned} p'_i &= p_i - C_P \\ q'_i &= q_i - C_Q \end{aligned} \quad (22)$$

The covariance matrix H is then obtained by multiplying the translated matching pairs, as shown in Equation (23):

$$H = \sum_{i=1}^{N'} p'_i \cdot q'^T_i \quad (23)$$

where N' is the number of matching pairs.

Singular value decomposition is performed on the covariance matrix H , as shown in Equation (24):

$$H = U \cdot \Sigma \cdot V^T \quad (24)$$

where U and V are orthogonal matrices and Σ is a diagonal matrix. The rotation matrix R can be calculated using U and V , as shown in Equation (25):

$$R = V \cdot U^T \quad (25)$$

The translation vector t can be calculated from the centroid of the point cloud data and the rotation matrix R , as shown in Equation (26):

$$t = C_Q - R \cdot C_P \quad (26)$$

After calculating the rotation matrix R and the translation vector t , the source point cloud data can be transformed into the target point cloud data using Equation (27):

$$q_i = p_i \cdot R + t \quad (27)$$

where p_i and q_i represent the point clouds in the source and target point cloud data, respectively.

In this way, the registration of point cloud data is completed.

3. Results

The data used in this paper comes from self-collected data from LiDAR equipment and open-source datasets. The equipment used for data collection is the Livox-Mid-360 radar. The scenarios are indoor collections, including indoor point cloud data, storage box point cloud data, and chair point cloud data. The open-source datasets used are the Stanford University point cloud data Dragon and Bunny datasets.

As shown in Figure 3, statistical filtering has been applied. The results indicate that this preprocessing technique effectively smooths edge details within the point cloud data. Furthermore, the filtering process not only mitigates the presence of noise and outlier data points, it also maintains the integrity of detailed features. This enhances the accuracy of subsequent point cloud data registration.

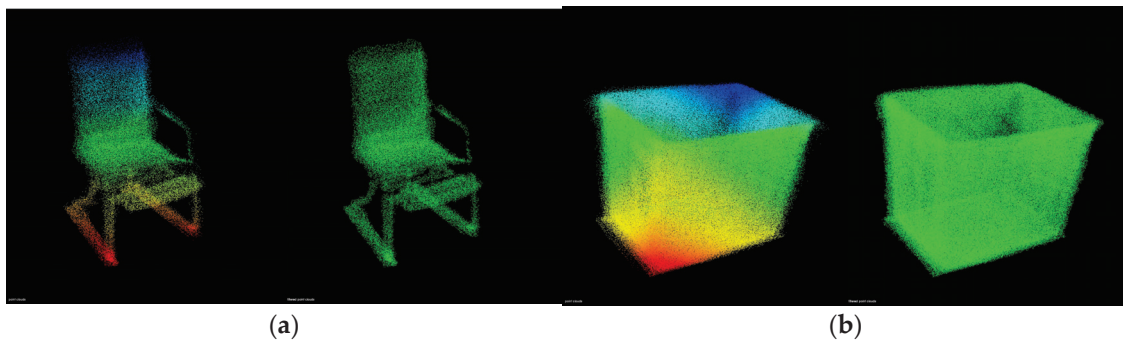


Figure 3. Comparison of the effects of statistical filtering on different point cloud data. (a) Chair data filtering, (b) storage box data filtering.

After applying statistical outlier removal filtering, local feature points were selected from the resulting point cloud data. This feature point selection process was performed on four distinct point cloud datasets: (a) an indoor point cloud captured from varying viewpoints; (b) the Dragon point cloud dataset, with two instances representing the same object but scanned from different positions and orientations; (c) a storage box point cloud dataset, consisting of one partially occluded scan and a more complete scan; and (d) point cloud data of a chair which is captured from different angles and positions resulting in partial occlusions.

Figure 4 illustrates the feature points extracted from the source and target point clouds, depicted in red and green, respectively.

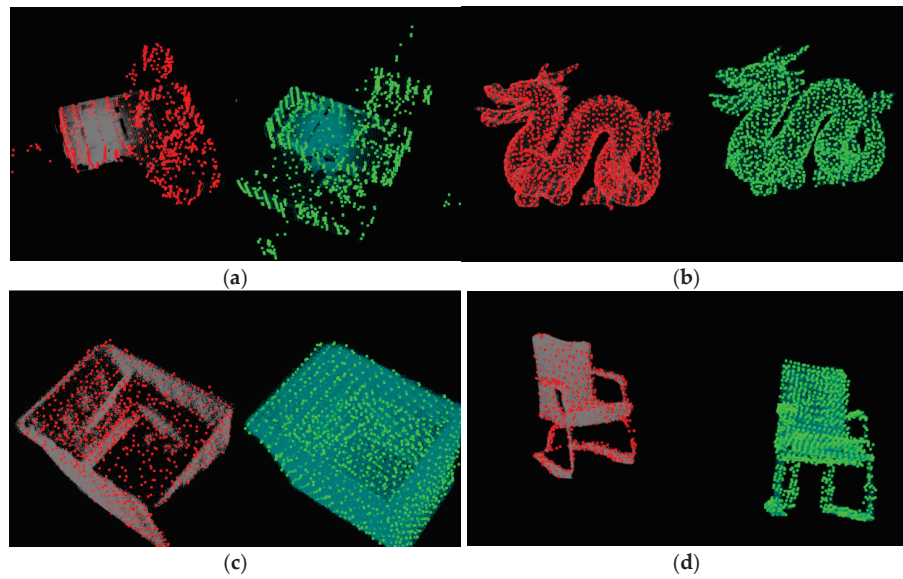


Figure 4. Feature point selection results of different point cloud data. (a) Feature points of indoor data, (b) feature points of the dragon data, (c) feature points of the storage box data, and (d) feature points of the chair data.

Following feature point extraction from the point cloud data, the dynamic neighborhood, L , is computed for each point via Equation (17). A cumulative histogram of projected distances to neighboring points within L then provides a discriminative feature descriptor, facilitating the establishment of feature point correspondences.

Optimal correspondences are determined by minimizing the chi-squared distance, which subsequently allows for the computation of the rotation matrix R and translation vector t . This completes the registration of the point cloud data.

The data used for point cloud data registration are house point cloud data (a), Dragon point cloud data (b), storage box point cloud data (c), and chair point cloud data (d). The registration results are shown in Figure 5.

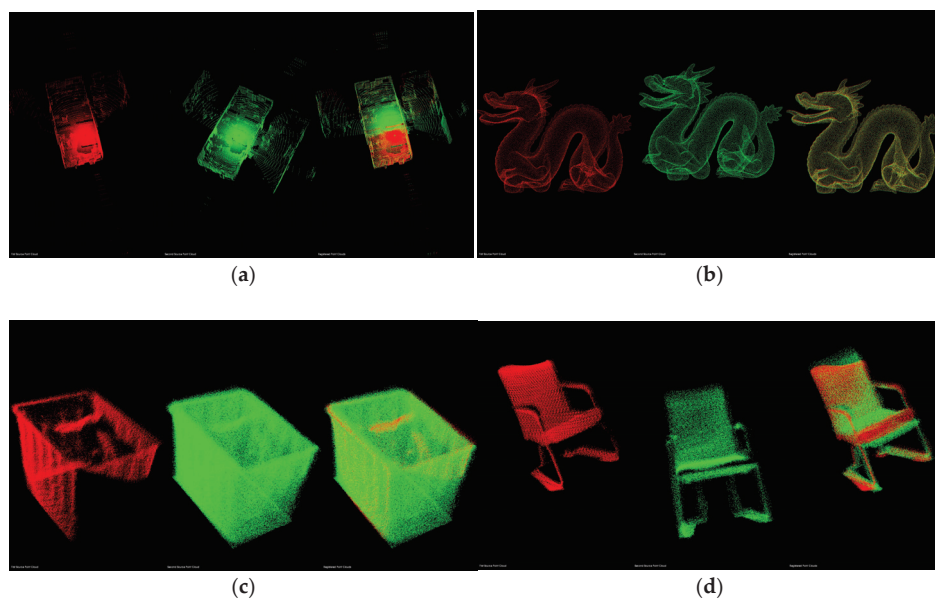


Figure 5. Registration results of different point cloud data. (a) Registration result of indoor data, (b) registration result of the Dragon data, (c) registration result of the storage box data, and (d) registration result of the chair data.

In Figure 5, red point cloud data are target point cloud data and green point cloud data are target point cloud data. After the registration algorithm, the two sets of point cloud data overlap and the data become more complete.

To validate the rationality and effectiveness of the proposed method, we compared its registration performance against K4PCSs, a k-dimensional (KD)-tree-based improved ICP method, and 3D-NDT under the same experimental conditions. Registration quality was evaluated using registration time and root mean square error (RMSE). The comparison experiments utilized three point cloud datasets: the Stanford Bunny and Dragon datasets, and a chair dataset acquired using a Livox-Mid360 LiDAR. The Bunny dataset comprised 35,947 points in both the source and target point clouds. The Dragon dataset contained 29,103 points in both the source and target point clouds. The acquired chair dataset consisted of 76,099 points in the source point cloud and 65,433 points in the target point cloud. Registration results for each method are shown in Figure 6.

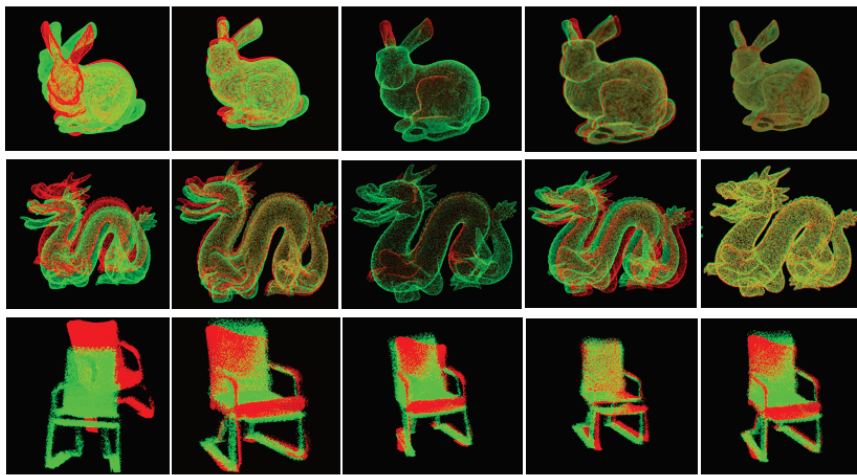


Figure 6. Comparison of point cloud data registration results.

Figure 6 provides a comparative visualization of registration results. Three datasets are used: the Stanford Bunny, the Stanford Dragon, and a chair dataset captured with LiDAR (arranged from top to bottom). The columns display, from left to right: the original, unregistered point clouds, the point clouds registered using K4PCSs, the point clouds registered using a KD-tree-based improved ICP, the point clouds registered using 3D-NDT, and the point clouds registered using the proposed method.

The more overlap of point cloud data, the better the registration effect, which can usually be measured by root mean square error. Root mean square error is the relationship between the distance difference between source point cloud data and target point cloud data after matrix transformation [27], as shown in Equation (28):

$$E = \sqrt{\frac{1}{n} \sum_{i=1}^n (A \cdot p_i - q_i)^2} \quad (28)$$

where A is the transformation matrix; p_i represents source point cloud; q_i is the target point cloud.

The smaller the value of E , the smaller the registration error and the higher the accuracy. The registration effect of each method is measured by the time taken for the registration of point cloud data and the root mean square error, as shown in Tables 2 and 3.

Table 2. Registration time for point cloud data.

Method	Registered Data	Time/s	Standard Deviations
K4PCSs method	Bunny	4.9	6.976
	Dragon	14.11	
	Chair	22.07	
KD-ICP method	Bunny	11.86	10.15
	Dragon	8.64	
	Chair	31.5	
3D-NDT method	Bunny	8.7	2.206
	Dragon	7.61	
	Chair	12.78	
Proposed method	Bunny	5.7	2.46
	Dragon	4.69	
	Chair	10.36	

Table 3. The root mean square error (RMSE) of the method after registration.

Method	Registered Data	RMSE/mm	Standard Deviations
K4PCSs method	Bunny	2.86	0.1028
	Dragon	3.089	
	Chair	2.882	
KD-ICP method	Bunny	0.599	1.520
	Dragon	0.978	
	Chair	4.207	
3D-NDT method	Bunny	1.587	0.7225
	Dragon	1.931	
	Chair	3.264	
Proposed method	Bunny	0.698	0.173
	Dragon	0.964	
	Chair	1.115	

Table 2 shows the registration time for each method. K4PCSs achieved the shortest registration time for the Bunny dataset (4.9 s), while the proposed method had the shortest registration times for both the Dragon (4.69 s) and chair (10.36 s) datasets.

Table 3 shows the root mean square error (RMSE) for the registration of each method. K4PCSs achieved the lowest RMSE for the Bunny dataset (0.599 mm), while the proposed method yielded the lowest RMSE for both the Dragon (0.964 mm) and chair (1.115 mm) datasets.

4. Discussion

Based on the data in Tables 2 and 3, the registration time of the proposed method is only slightly slower than that of K4PCSs for the Bunny dataset; for all other datasets tested, the proposed method achieves the fastest registration time. Regarding registration accuracy, KD-ICP performs slightly better than the proposed method on the Bunny dataset; however, the registration time of KD-ICP is approximately twice that of the proposed method. For the Dragon and chair datasets, the proposed method demonstrates both optimal accuracy and the fastest registration speed. These results indicate that, compared to the improved KD-ICP method [4,5], K4PCSs method [9], and the 3D-NDT method [10], the proposed approach achieves faster and more accurate data registration, significantly enhancing the accuracy and efficiency of point cloud data registration. Noise typically occurs at the edges of the main data, making it easy to filter out the noise that is far from the main data using statistical filters. For unevenly distributed data, the dynamic neighborhood is adjusted appropriately based on data density to include more feature information. In the registration of diverse point cloud datasets the proposed method offers a favorable balance of accuracy and efficiency, enabling the completion of point cloud data registration with high precision and speed.

5. Conclusions

This paper introduces a novel point cloud registration method based on dynamic neighborhood features. The method leverages the projection distance relationship between statistical feature points and other points within the dynamic neighborhood. This approach establishes accurate feature point correspondences and enables robust point cloud registration. By accurately matching feature points and overcoming the limitations of traditional registration techniques that often struggle to simultaneously optimize accuracy and efficiency, this method offers a compelling solution. The incorporation of multiple parameters enhances the robustness of the approach. Furthermore, by using feature points instead of the entire point cloud for registration, the computational burden is reduced, leading to more efficient data processing. The dynamic neighborhood approach also allows the method to adapt to point cloud data with varying scales and densities, enhancing its flexibility and broadening its application potential in complex environments, especially in fields such as autonomous driving, robotic navigation, building reconstruction, and 3D map construction. Integrating this method into practical systems can enhance the capability of processing point cloud data in complex environments. This study was tested only on a limited set of data and did not extend to a wider range of complex environmental data, which presents a limitation. Future work needs to involve testing on more diverse datasets to validate the feasibility of the proposed method and to develop a simpler and more direct guideline for parameter selection.

Author Contributions: Conceptualization, X.L. and Z.W.; methodology, X.L.; software, X.L.; validation, X.L., Z.W. and R.W.; formal analysis, R.W.; investigation, R.W.; resources, Z.W.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, Z.W.; visualization, R.W.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ICP	Iterative Closest Point
LiDAR	Light Detection and Ranging
RANSAC	Random Sample Consensus
ISSs	Intrinsic Shape Signatures
3SDC	3D Shape Context
4PCSs	Four-Point Congruent Sets
K4PCSs	Keypoint-Based Four-Points Congruent Sets
3D-NDT	3 Dimensions-Normal Distributions Transform
NDT	Normal Distributions Transform
KD	K-Dimensional
RMSE	Root Mean Square Error

References

1. Marani, R.; Renò, V.; Nitti, M.; D’Orazio, T.; Stella, E. A modified iterative closest point algorithm for 3D point cloud registration. *Comput.-Aided Civ. Infrastruct. Eng.* **2016**, *31*, 515–534. [CrossRef]

2. Yang, H.; Shi, J.; Carlone, L. Teaser: Fast and certifiable point cloud registration. *IEEE Trans. Robot.* **2020**, *37*, 314–333. [CrossRef]
3. Zou, Z.; Lang, H.; Lou, Y.; Lu, J. Plane-based global registration for pavement 3D reconstruction using hybrid solid-state LiDAR point cloud. *Autom. Constr.* **2023**, *152*, 104907.
4. Yang, J.; Cao, Z.; Zhang, Q. A fast and robust local descriptor for 3D point cloud registration. *Inf. Sci.* **2016**, *346*, 163–179.
5. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155.
6. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*; SPIE: Boston, MA, USA, 1992; Volume 1611, pp. 586–606.
7. Xu, G.; Pang, Y.; Bai, Z.; Wang, Y.; Lu, Z. A fast point clouds registration algorithm for laser scanners. *Appl. Sci.* **2021**, *11*, 3426. [CrossRef]
8. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. In Proceedings of the SIGGRAPH '08: ACM SIGGRAPH 2008 Papers, Los Angeles, CA, USA, 11–15 August 2008; pp. 1–10.
9. Theiler, P.W.; Wegner, J.D.; Schindler, K. Keypoint-based 4-points congruent sets—automated marker-less registration of laser scans. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 149–163.
10. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), Las Vegas, NV, USA, 27 October–1 November 2003; Volume 3, pp. 2743–2748.
11. Xue, S.; Zhang, Z.; Lv, Q.; Meng, X.; Tu, X. Point cloud registration method for pipeline workpieces based on PCA and improved ICP algorithms. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *612*, 032188.
12. Huang, X.; Zhang, J.; Wu, Q.; Fan, L.; Yuan, C. A coarse-to-fine algorithm for matching and registration in 3D cross-source point clouds. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 2965–2977.
13. Sadeghi, H.; Raie, A.A. Approximated Chi-square distance for histogram matching in facial image analysis: Face and expression recognition. In Proceedings of the 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP), Isfahan, Iran, 22–23 November 2017; pp. 188–191.
14. Pittner, S.; Kamarthi, S.V. Feature extraction from wavelet coefficients for pattern recognition tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 83–88.
15. Zheng, Y.; Li, Y.; Yang, S.; Lu, H. Global-PBNet: A novel point cloud registration for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 22312–22319. [CrossRef]
16. Sun, R.; Zhang, E.; Mu, D.; Ji, S.; Zhang, Z.; Liu, H.; Fu, Z. Optimization of the 3D point cloud registration algorithm based on FPFH features. *Appl. Sci.* **2023**, *13*, 3096. [CrossRef]
17. Arce, G.R. Multistage order statistic filters for image sequence processing. *IEEE Trans. Signal Process.* **1991**, *39*, 1146–1163. [CrossRef]
18. Chan, C.K.; Tan, S.T. Determination of the minimum bounding box of an arbitrary solid: An iterative approach. *Comput. Struct.* **2001**, *79*, 1433–1449. [CrossRef]
19. Gressin, A.; Mallet, C.; Demantké, J.; David, N. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS J. Photogramm. Remote Sens.* **2013**, *79*, 240–251. [CrossRef]
20. Biglia, A.; Zaman, S.; Gay, P.; Aimonino, D.R.; Comba, L. 3D point cloud density-based segmentation for vine rows detection and localisation. *Comput. Electron. Agric.* **2022**, *199*, 107166. [CrossRef]
21. Zou, R.; Zhang, Y.; Chen, J.; Li, J.; Dai, W.; Mu, S. Density estimation method of mature wheat based on point cloud segmentation and clustering. *Comput. Electron. Agric.* **2023**, *205*, 107626. [CrossRef]
22. Nurunnabi, A.; West, G.; Belton, D. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognit.* **2015**, *48*, 1404–1419. [CrossRef]
23. Yao, Z.; Zhao, Q.; Li, X.; Bi, Q. Point cloud registration algorithm based on curvature feature similarity. *Measurement* **2021**, *177*, 109274. [CrossRef]
24. Heil, C.E.; Walnut, D.F. Continuous and discrete wavelet transforms. *SIAM Rev.* **1989**, *31*, 628–666. [CrossRef]
25. Makovetskii, A.; Voronin, S.; Kober, V.; Voronin, A. Point cloud registration based on multiparameter functional. *Mathematics* **2021**, *9*, 2589. [CrossRef]
26. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells. *Remote Sens.* **2017**, *9*, 433. [CrossRef]
27. Draper, C.; Reichle, R.; de Jeu, R.; Naeimi, V.; Parinussa, R.; Wagner, W. Estimating root mean square errors in remotely sensed soil moisture over continental scale domains. *Remote Sens. Environ.* **2013**, *137*, 288–298.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Dynamic Obstacle Avoidance for Robotic Arms Using Deep Reinforcement Learning with Adaptive Reward Mechanisms

Sen Yan ¹, Yanping Zhu ^{2,*}, Wenlong Chen ², Jianqiang Zhang ², Chenyang Zhu ¹ and Qi Chen ¹

¹ School of Computer Science and Artificial Intelligence, Changzhou University, Yanzheng West 2468#, Changzhou 213164, China; cczuchenqi@163.com (Q.C.)

² School of Wang Zheng Microelectronics, Changzhou University, Yanzheng West 2468#, Changzhou 213164, China; cczuchenwenlong@163.com (W.C.); 13105506003@163.com (J.Z.)

* Correspondence: zhuyanping@cczu.edu.cn

Abstract: To address the challenges of robotic arm path-planning in dynamic environments, this study proposes a reinforcement learning-based dynamic obstacle avoidance method. The study concerns a robot with six rotational degrees of freedom when moving outside of singular configurations, enabling more flexible and precise motion-planning. First, a dynamic exploration guidance mechanism is designed to enhance learning efficiency and reduce ineffective exploration. Second, an adaptive reward function is developed to enable real-time path-planning while avoiding obstacles. A simulation environment is constructed using CoppeliaSim software, and the experiment is performed with two cylindrical obstacles that move randomly within the workspace. The experimental results demonstrate that the improved method significantly outperforms traditional algorithms in terms of convergence speed, reward value, and success rate.

Keywords: obstacle avoidance; deep reinforcement learning; reward function; robotic arm

1. Introduction

With the advancement of high-performance computing, the Internet of Things, and artificial intelligence technologies, robotic arms have been widely applied in areas such as automated production, urban security and services, and space exploration, continuously progressing toward the intelligent objectives of self-learning and self-regulation [1]. In most cases, robots can effectively avoid the static obstacles present in various environments. Nevertheless, the utilized algorithms exhibit insufficient adaptability to dynamically changing obstacles, making it challenging to quickly and accurately avoid them [2]. Probability sampling-based motion planning algorithms such as Rapid Random Tree (RRT) can swiftly search high-dimensional spaces; however, their limited environmental exploration ability leads to poor performance in handling dynamic obstacles and complex tasks. Therefore, in this context, finding the right training algorithm is important [3,4]. A combination of the Artificial Potential Field (APF) algorithm with numerical methods based on the Jacobian matrix has been employed to address obstacle avoidance-based path-planning in unknown environments [5].

However, due to the aforementioned limitations of traditional algorithms in dynamic environments, academic circles are paying more and more attention to the development of reinforcement learning-based approaches [6].

The rapid advancements in deep learning and computational capabilities have made deep reinforcement learning a reality, which is progressively being employed in domains

such as gaming, robot navigation, and industrial production [7]. DeepMind has developed models such as AlphaZero based on Deep Reinforcement Learning (DRL) [8], which defeated world-class chess players in competitions, demonstrating that this technology can achieve a superhuman performance in complex environments. To address the challenges of non-Markovian tasks in multi-agent systems, one study proposed a synchronized decentralized training paradigm based on GR(1) specifications [9]. By decomposing the synthesized GR(1) strategies into individual reward structures, this approach guides the multi-agent reinforcement learning (MARL) process, thereby effectively preserving the safety and liveness properties of the system. DRL has made significant strides not only in academia, but also in industry. For instance, IBM's artificial intelligence system Watson employs deep reinforcement learning techniques to assist doctors in diagnosing cancer. Furthermore, in areas such as inverted pendulum control and unmanned surface vehicle control [10], DRL is considered one of the most promising cutting-edge artificial intelligence technologies.

Reinforcement learning has similarly garnered the interest of experts and scholars in the field of robotics, demonstrating exceptional performance in path-planning within complex environments in scenarios characterized by a lack of prior knowledge. For example, Liu et al. [11] innovatively integrated digital twin technology with DRL and proposed a multitasking training approach based on the twin synchro-control (TSC) scheme to enhance the control performance of humanoid robot arms. A data acquisition system was developed to collect human joint angle data, which served as a prior knowledge input for DRL, thereby optimizing the reward function of the deep deterministic policy gradient (DDPG) algorithm and improving the learning stability and convergence speed. This approach was successfully applied in the humanoid robot BHR-6, enabling it to perform tasks that are challenging to train using existing methods. Sun et al. [12] proposed an end-to-end navigation strategy based on causal reinforcement learning, which is learned directly from the data, bypassing the traditional mapping and planning steps, thereby improving the responsiveness of drones in obstacle avoidance and navigation tasks. To address the issue of local minima in robotic arm collision avoidance using Artificial Potential Field (APF), a study proposed a method that integrates APF with deep reinforcement learning (DRL) and improves Hindsight Experience Replay (HER) into HER-CA to enhance training efficiency [13]. Wang et al. [14] proposes a decision reconstruction network integrated into the SAC algorithm. This network dynamically reconfigures network layers to enable cross-layer information exchange, effectively mitigating gradient conflicts and addressing the optimization challenges caused by parameter-sharing in robotic multi-task reinforcement learning. Yadav et al. [15] examined regularization-based methods like synaptic intelligence for sequential learning in offline RL, finding they reduce learning time and mitigate forgetting but offer limited knowledge transfer. The study proposed a deep reinforcement learning-based coverage path planning approach, using actor-critic methods to train mobile robots for exploration and full coverage [16]. Zhao et al. [17] introduced the Efficient Progressive Policy Enhancement (EPPE) framework, which integrates sparse and process rewards to improve DRL training efficiency and global optimality in path-planning.

Zhong et al. [18] have used a welding robot path-planning method based on the DDPG algorithm to solve the path-planning problem in a high-dimensional continuous state and action space. Cao et al. [19] proposed a multi-agent reinforcement learning approach for square shaft-hole assembly. The method designs a reward function based on the Decoupled Multi-Agent Deep Deterministic Policy Gradient (DMDDPG) algorithm and models the first three and last three joints of the robotic arm as a multi-agent system. Liu et al. [20] proposed DRL-based path planning method incorporates a safety verification mechanism

to achieve collision-free trajectories. However, it still has limitations such as fixed end-effector orientation and the requirement for a smaller threshold in trajectory merging. Wang Y. et al. [21] introduced a method to control a mobile robotic arm to perform door-opening operations based on an improved Proximal Policy Optimization (PPO) algorithm. Wu et al. [22] achieved the requirements of online trajectory-planning by training a strategy consisting of a feedforward neural network in the simulation environment. Lindner et al. [23] employed the DDPG algorithm to optimize the static obstacle-avoidance performance of robotic arms; however, they did not extend their approach to dynamic obstacle environments. To address high-dimensional path-planning challenges, Prianto et al. [24] proposed a path-planning algorithm based on the Soft Actor-Critic (SAC) framework, utilizing configuration space augmentation to manage the complex configuration spaces associated with multi-arm systems. They also incorporated the Hindsight Experience Replay (HER) technique to enhance the sampling efficiency, enabling the real-time and rapid generation of any end-to-end shortest path. Singla et al. [25] combined recurrent neural networks with temporal attention processes and deep reinforcement learning, allowing agents to remember important contextual data in order to prevent collisions. However, the extensive operational space of robotic arms makes it challenging to quickly obtain successful samples, causing DRL algorithms to expend significant time on ineffective exploration and resulting in slow convergence rates. To mitigate this issue, researchers have proposed leveraging “expert guidance” to accelerate the convergence of DRL algorithms. Xu et al. [26] developed a strategy for axis-hole assembly problems guided by impedance control algorithms assisted by the DDPG for the correction of trajectories. Rahmatizadeh et al. [27] used a deep neural network controller based on a Long Short-Term Memory (LSTM) network to generate trajectories. At the same time, a mixture density network (MDN) was used to calculate the error signal to adapt to multimodal demonstration data. Through imitation learning, the robotic arms were able to reasonably plan paths to complete the picking and placing of objects. Chen et al. [28] designed a comprehensive reward function that combines dynamic obstacle avoidance and target proximity, which was integrated with the priority experience replay (PER) approach to improve the utilization of samples, finally allowing for the realization of real-time path-planning. Li et al. [29] integrated the APF and DDPG algorithms, using approximate poses derived from the APF algorithm as expert guidance. They implemented a phased approach to obstacle-avoidance trajectory planning, where the DDPG algorithm dominated training during the obstacle-avoidance phase and the approximate poses from APF guided DDPG training during the target-planning phase. Additionally, they proposed a guiding mechanism to connect the control outputs of the two phases, thereby efficiently training a policy model tailored to address the research problem at hand.

However, trajectory-planning approaches for robotic arms based on DRL methods still encounter challenges in achieving convergence and conducting efficient explorations, particularly within unstructured work environments that feature both dynamic and static obstacles. The core issue lies in managing a substantial amount of failed explorations and sparse rewards, which result in training failure and make it difficult to learn effective strategies.

In order to solve the problem in which robot arms face difficulties in avoiding obstacles and reaching the target point in environments characterized by dynamic obstacles, we propose a new path-planning method based on reinforcement learning. Through designing a dynamic exploration guidance mechanism and an adaptive obstacle avoidance reward function, we optimize the robotic arm’s performance in terms of path-planning. The overall environmental structure is illustrated in Figure 1. During the training process, the policy network receives the state vector as input and outputs the selected action; the critic network

receives the state vector and action as input and outputs a value function that evaluates the expected reward or value of selecting the action in the state. During the testing process, the state vector obtained from the simulation environment is input into the policy network to obtain the selected action. This study implements a path planning strategy that ensures that the robotic arm remains stable during learning, and evaluates the effectiveness of this strategy in environments featuring dynamic obstacles.

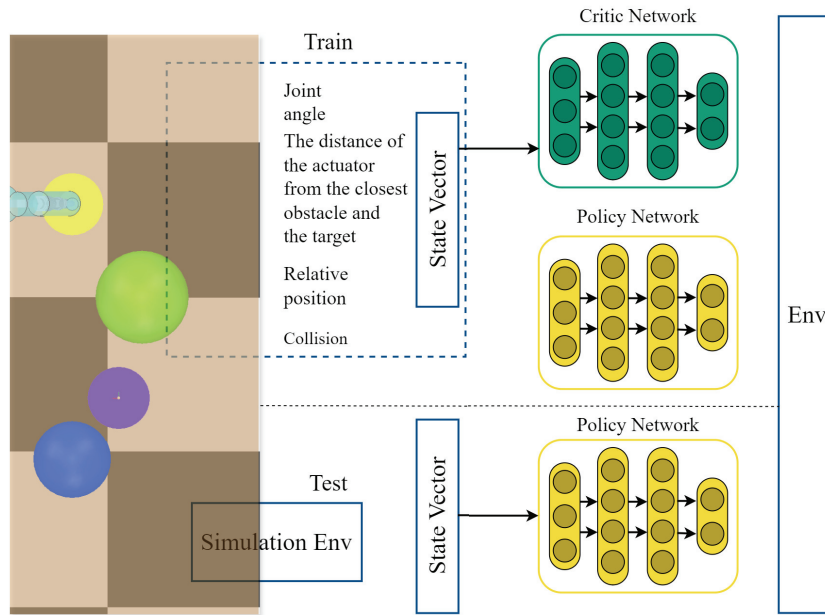


Figure 1. Overall environment architecture.

2. Preliminaries

2.1. Markov Decision Process

Broadly speaking, Reinforcement Learning (RL) addresses sequential decision-making issues by seeking decision sequences that maximize the expected cumulative reward. The Markov decision process (MDP) provides a mathematical framework to model the motion control problem of the robot arm as a reinforcement learning problem. The reinforcement learning algorithm learns the optimal strategy through continuous interaction (trial and error), allowing the robot arm to make autonomous decisions. In RL tasks, MDPs are typically chosen as the mathematical models to describe decision problems, which are represented as tuples (S, A, R, ρ, γ) , where S and A denote finite continuous state and action spaces, respectively; the reward function $R(R_t = r_t | S_t = s_t, A_t = a_t) : S \times A$ specifies the reward received from the environment upon transitioning between states; the transition probability $\rho(S_{t+1} = s_{t+1} | S_t = s_t, A_t = a_t) : S \times A \rightarrow P(S_{t+1})$ defines the probability distribution of moving to state s' after taking action a in state S ; and the discount factor γ adjusts the significance of future rewards. The agent's behavior is defined by the policy $\pi(a | s) : S \rightarrow P(a)$, which maps states to probability distributions across actions.

2.2. Kinematic Modeling

The kinematics of robotic arms primarily reflect the relationships among displacements, velocities, and accelerations between joints. In this study, a six-degrees-of-freedom welding robotic arm is modeled using the Denavit–Hartenberg (D-H) parameter method [30]. As a standard approach for serial robotic arm kinematic modeling, the D-H parameter method is both rigorous and widely applicable. Based on this method, the geometric relationships of the robotic arm's joints are systematically described, and the

standard D-H parameters are presented in Table 1. A picture of the robotic arm with some of the DH parameters marked is shown in Figure 2.

Table 1. D-H parameters of the robotic arm.

Link i	θ_i (°)	d_i (m)	a_i (m)	α_i (°)	θ_i Range (°)
1	q_1	0	0	-90	$[-180, 180]$
2	$q_2 - 90$	-0.1265	0.726	0	$[-85, 85]$
3	$q_3 - 180$	-0.096	0	90	$[-40, 220]$
4	q_4	0.6305	0	90	$[-90, 90]$
5	q_5	0.091	0	-90	$[-90, 90]$
6	q_6	0.122	0	0	$[-360, 360]$

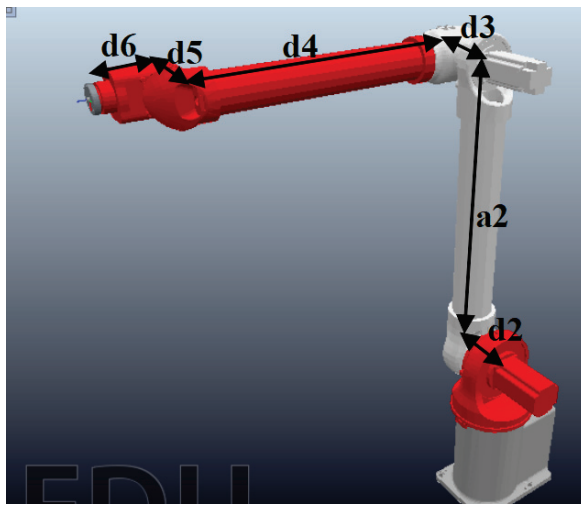


Figure 2. Robotic arm 3D model.

The robotic arm possesses six degrees of freedom, allowing the end-effector's pose and position to be represented by 0_6T . Vectors n , o , and a form a 3×3 matrix that describes the end effector's pose, while the vector p denotes its positional information. Utilizing the D-H parameters presented in Table 1, the link transformation matrices of the robotic arm can be determined. The following is the expression for the transformation matrix 0_6T between the base coordinate frame and the end effector coordinate frame:

$${}^0_6T = \prod_{i=1}^6 {}^{i-1}_iT(q_i) \quad (1)$$

Here, the superscript of the coordinate T denotes the coordinate system being transformed, while the subscript indicates the target coordinate system; (n_x, n_y, n_z) represents the normal vector of the robotic arm's end effector; (o_x, o_y, o_z) indicates the sliding vector; (a_x, a_y, a_z) is the approach vector; and (p_x, p_y, p_z) represents the position vector of the end effector. The homogeneous transformation matrix ${}^{i-1}_iT$ describes the pose of link i 's coordinate system relative to that of link $i-1$, as shown by the following formula:

$${}^{i-1}_iT(q_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The final link is typically referred to as the end effector, to which an effector with a reference frame (e.g., a welding torch) is attached. The homogeneous transformation matrix of the end effector relative to the base link, denoted as ${}^0_eT(q) = {}^0_6T(q){}_e^6T$, can be defined as:

$$x_e = f_{kine}(q) \quad (3)$$

where $x_e = \begin{bmatrix} p_e & \phi_e \end{bmatrix}^T \in R^6$ represents the pose, in which p_e is the three-dimensional position and ϕ_e denotes the orientation in Cartesian space.

As most motions are defined directly in Cartesian space, the robot arm needs to convert Cartesian motion into joint motion in joint space. If the Cartesian velocity of the end effector $\dot{x}_e$ is given, the corresponding joint velocity $\dot{q}$ is expressed as follows:

$$\dot{q} = J^{-1}(q) \dot{x}_e \quad (4)$$

where $J(q)$ represents the 6×6 Jacobian matrix of the six-degree-of-freedom robotic arm.

3. Method

The state space and action space for the dynamic obstacle-avoidance path-planning of robotic arms are explained in detail in this section. We also list the adaptive reward function and the dynamic exploration guidance mechanism. As illustrated in Figure 3, in order to use the MDP to solve the robotic arm path-planning problem discussed in this research, we must specify the interaction process between the algorithm and the simulation environment.

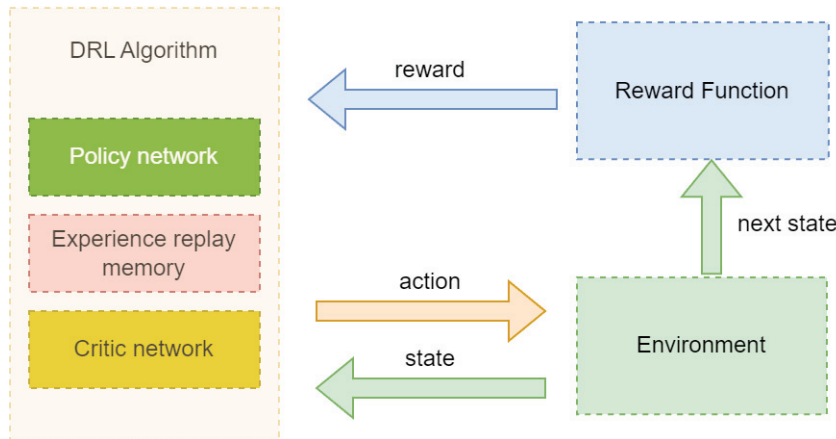


Figure 3. Framework for the DRL algorithm to interact with the environment.

3.1. State Space

In a real working environment, the state information can be vast (and even infinite), making it impractical to enumerate all possible states. Therefore, in practical applications, key features are usually extracted for representation, in order to reduce the computational complexity and improve learning efficiency. In DRL, state information is not only important for the agent to perceive the environment, but also directly affects its decision-making process and the evaluation of long-term rewards. As such, reasonable state design can significantly improve the convergence, convergence speed, and final performance of the algorithm.

This study aims to train an agent to autonomously control the joint motion of a six-axis robot arm, while avoiding collisions and getting as close as possible to a specified target. The state space s_t can be represented by Equation (5):

$$s_t = [q, p_e, d_{tar}, d_{obs}, collision] \quad (5)$$

where $q \in R^6$ denotes the joint angles of the end effector; $p_e \in R^3$ represents the Cartesian position of the end effector's reference coordinate system relative to the target coordinate system; the scalar d_{tar} indicates the Euclidean distance between the end effector and the origin of the target coordinate system; and d_{obs} is the shortest Euclidean distance between the end effector and the two types of obstacles. The final component—collision, which can be obtained via the CoppeliaSim API function—is a Boolean value that indicates whether the manipulator has collided with any environmental obstacles.

3.2. Action Space

The agent in the environment is a 6-DOF welding robot arm. For the motion control of the robot arm, the control strategy of the agent adopts the velocity control method, using the joint velocity as the control input, in order to achieve the real-time adjustment and trajectory optimization of the robot arm in the continuous action space. This method can dynamically adjust the angular velocity of each joint according to the current state, such that the robot arm moves along the optimal trajectory to minimize the distance between the end effector and the target position. Therefore, the action space is defined as follows:

$$a_t = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6] \quad (6)$$

In this context, $\dot{q} \in R^6$ denotes the target angular velocity of each servo motor-driven rotational joint within its allowable range. The dimension of the action space is 6.

3.3. Reward Signal

To address the issue of sparse rewards in DRL-based trajectory-planning for robotic arms within dynamic environments, this study introduces an adaptive reward function that seamlessly integrates existing reward mechanisms with energy function constraints, drawing on the principles of artificial potential field methods and energy functions. This adaptive function facilitates a smooth transition between the two reward mechanisms and carries out dynamic adjustments according to the distance between the end effector and the obstacle to ensure that the path-planning result is both safe and efficient:

$$R_{ada} = \beta \cdot R_t + (1 - \beta) \cdot R_{energy} \quad (7)$$

$$R_{energy} = - \sum_{i=1}^6 \tanh(\|\dot{q}\|^2) \quad (8)$$

Specifically, $\beta \in [0, 1]$ is designed such that, when the end effector is sufficiently distant from obstacles, R_t serves as the dominant factor; when the distance between the end effector and an obstacle falls below the critical distance d_b , the energy function takes precedence. β can be expressed as follows:

$$\beta = \frac{\tanh(x) + 1}{2} \quad (9)$$

where

$$x = \frac{d_{obs} - d_b}{k_o} \quad (10)$$

It is evident that the β function yields a smooth transition from 0 to 1: when d_{obs} is large, β approaches 1; meanwhile, when d_{obs} is small, it approaches 0. Here, k_o serves as a tuning parameter that adjusts the smoothness of the reward mechanism. R_t is designed to prompt the robotic arm to accomplish the task, which mainly includes the following aspects:

$$R_t = R_{\text{target}} + R_{\text{avoid}} + R_{\text{done}} \quad (11)$$

The first term on the right-hand side of Equation (11) is designed to guide the end effector toward a specific target position. It is defined based on the difference in the end effector—target distance between the current step and the previous step:

$$R_{\text{target}} = \begin{cases} d_{\text{tar}} - d'_{\text{tar}} & \text{if } d_{\text{tar}} > \delta \\ (d_{\text{tar}} - d'_{\text{tar}}) \cdot \frac{\delta}{d_{\text{tar}}} & \text{if } d_{\text{tar}} \leq \delta \end{cases} \quad (12)$$

In robotic arm path-planning tasks, we do not necessarily expect the end-effector to reach the designated location in a single step every time; instead, a gradual approach strategy is adopted. Specifically, the end effector is controlled to move incrementally at each step, gradually closing in on the target position. In designing the reward feedback, the distance between the end effector's current position and the target point is introduced as a key parameter. Compared with relying solely on a one-step Euclidean distance, this reward function may slow down the movement process, yet it effectively reduces control complexity and meets the path-planning objective under higher-precision requirements.

Furthermore, in path-planning, precision and efficiency often impose mutual constraints. To enhance the reward mechanism's effectiveness, when the current state nears boundary conditions (e.g., when the distance to the target point is relatively small), each slight movement in the correct direction should receive a stronger positive reward. This approach guides the robotic arm to gradually refine its path, ultimately achieving high-precision control of the target position.

The second term, R_{avoid} , is the obstacle avoidance reward function designed to guide the robotic arm in evading obstacles:

$$R_{\text{avoid}} = -\left(\frac{1}{1 + d_{obs}}\right)^\rho \quad (13)$$

Here, d_{obs} represents the smallest gap between the robotic arm and obstacles, and ρ denotes the decay rate of the reward function.

The third term, R_{done} , is the termination reward function, primarily designed to reward the achievement of the target and collisions with obstacles:

$$R_{\text{done}} = \begin{cases} 30 & d_{\text{tar}} < \eta \\ -15 & \text{collision} = \text{True} \end{cases} \quad (14)$$

Considering the complexity of dynamic environments, the likelihood of collisions during the initial phases of training is substantially higher than that of reaching the target. Consequently, the reward for successfully attaining the target should be several times greater than the penalty imposed for collisions, and the reward is set to zero when the target is not achieved.

3.4. Dynamic Exploration Guidance Mechanism

There are two main challenges regarding path-planning in dynamic obstacle avoidance tasks: one is the need to avoid collisions with obstacles, and the other is the need to successfully reach the specified target point. However, it is very complicated to deduce the relationship between the state space and the action space from the perspective of the agent. This is because the state space is usually large and continuous, especially in a dynamic environment, and the agent must not only deal with the current state, but also predict and react to possible changes. In order to learn how to choose the appropriate action from the current state, the agent usually requires a large number of training iterations and calculations. This process may involve billions of training rounds and calculations, which is very time-consuming and computationally intensive for real-world applications. Therefore, in order to solve this problem, a better approach is to transform the infinite state space into a finite state space. Specifically, by limiting the range of states, the originally very large or continuous state space can be simplified into a finite set.

Consequently, it is difficult to directly start training the robot path-planning model using reinforcement learning algorithms. By restricting the state space, the agent can focus on exploring only those states that are relevant to the path-planning task, which is crucial for ensuring efficient convergence and learning. This approach excludes irrelevant and extraneous states, thereby simplifying the agent's decision-making process. Rather than being overwhelmed by an unlimited and irrelevant state space, the agent concentrates on pertinent states, resulting in more effective decision making and faster learning. Therefore, implementing dynamic policy selection enables quicker convergence during training. Through the combination of a guiding term with stochastically sampled actions that include exploration noise from deep reinforcement learning algorithms, the selected action is dynamically integrated. Due to the presence of obstacles in the working environment, relying solely on the velocity output $\dot{q}_{inv}$ generated by the inverse kinematics (IK) module is insufficient to achieve effective obstacle avoidance. To address this limitation, this paper incorporates an additional action term at produced by a deep reinforcement learning (DRL) policy network. This action serves as a compensatory component to the IK output, guiding the manipulator to avoid obstacles while maintaining goal-directed motion. The resulting hybrid action formulation is given by

$$q_{t+1} = q_t + (\dot{q}_{inv} + a_t) \cdot \Delta t \quad (15)$$

where Δt denotes the time step duration. This formulation enables the manipulator to adaptively adjust its trajectory in response to environmental constraints while efficiently progressing toward the target. We introduced the dynamic scale factor to adjust the proportion of dynamic exploration guidance mechanism:

$$rate = \ln(Current_episode + 1) / \ln(Max_episode + 1) \quad (16)$$

In addition, when operating a robotic arm manually, the speed of operation slows down as the arm approaches the target point to maintain accuracy. Similarly, in robotic arm control, the movement step size should gradually decrease to ensure a certain level of precision. Therefore, the agent's joint angles are controlled as follows:

$$q_{t+1} = \begin{cases} q_t + (\dot{q}_{inv} \cdot (1 - rate) + a_t \cdot rate) \cdot dt & \text{if } d_{tar} > \delta \\ q_t + (\dot{q}_{inv} \cdot (1 - rate) + a_t \cdot rate) \cdot dt \cdot \frac{d_{tar}}{\delta} & \text{if } d_{tar} \leq \delta \end{cases} \quad (17)$$

In this context, rate is a dynamic scaling factor that increases rapidly during the initial phase and slows its growth in the later stages; Current_episode and Max_episode denote the current and maximum number of training episodes, respectively, and a_t refers to a randomly sampled action augmented with exploration noise. The training workflow incorporating the dynamic exploration guidance strategy is illustrated in Figure 4.

By dynamically adjusting the scaling factor, the initial phase of training significantly increases the probability of the robotic arm obtaining positive samples, effectively preventing the waste of training resources caused by ineffective exploration and alleviating the issue of sparse rewards in the early stages. As the number of training iterations increases, the scaling factor gradually grows, diminishing the influence of inverse kinematics, while the proportion of randomly sampled actions markedly rises, resulting in greater randomness in the robotic arm's action selection. Building on the sufficient number of positive samples obtained from the preliminary training, the model achieves efficient convergence optimization, progressively mastering the ability to realize path-planning in dynamic obstacle-avoidance environments.

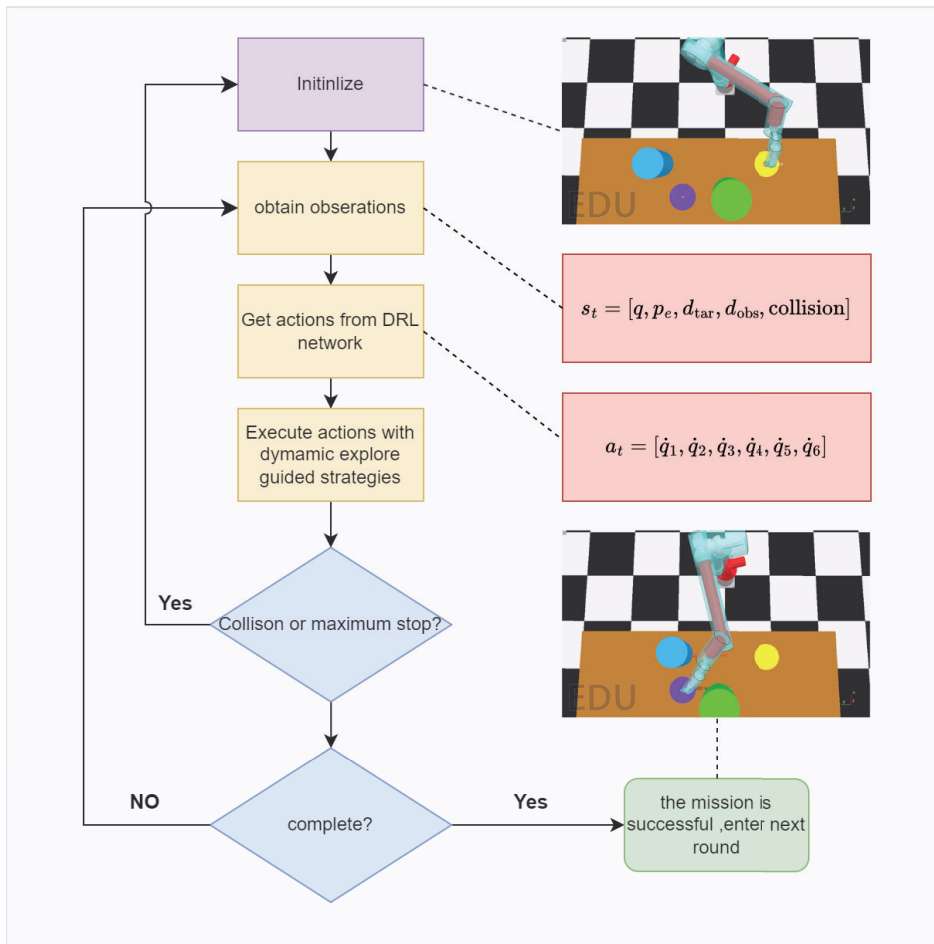


Figure 4. Training flowchart after applying the dynamic exploration guidance mechanism.

4. Experiments And Analysis

4.1. Experimental Setup

The simulation platform employed for the experiments was CoppeliaSim Edu version 4.2. For the implementation of the algorithm, the reinforcement learning frameworks PyTorch and OpenAI's Gym were utilized with the code developed in Python. The inverse kinematics module was provided by MATLAB. With the support of CoppeliaSim's Remote API,

network communication between Python and the CoppeliaSim platform was achieved, as illustrated in Figure 5. The state information of the robotic arm was obtained by parsing the feedback data from the CoppeliaSim Edu simulation platform. Regarding the hardware, a PC (CPU i7-12600H @2.70 GHz, RAM 32 GB, GPU RTX 3070 8 GB) was used to support the training of the algorithms.

As shown in Figure 6, each environment included two cylindrical obstacles with a base diameter of 0.15 m and a height of 0.3 m. In environment 1, the two obstacles remained stationary while the state of the obstacles in environment 2 changed; namely, the obstacles moved randomly at every step. Their maximum linear velocity was set to 150 mm/s to simulate real-time environmental changes. The initial positions of the two obstacles relative to the robot base were set as [900, −400, 300] mm for Obstacle 1 and [1050, −300, 300] mm for Obstacle 2. Similarly, the target point was located at [100, −350, 300] mm relative to the robot base. In both cases, the robot arm had to bypass the obstacles and reach the target point to be considered successful. The location of the target point was randomly generated after the robot arm reached it. To ensure numerical stability and generalizability, all robot configurations during training and testing were selected to be sufficiently far from singularities and joint limits. Training in the first environment was conducted for 5000 episodes.

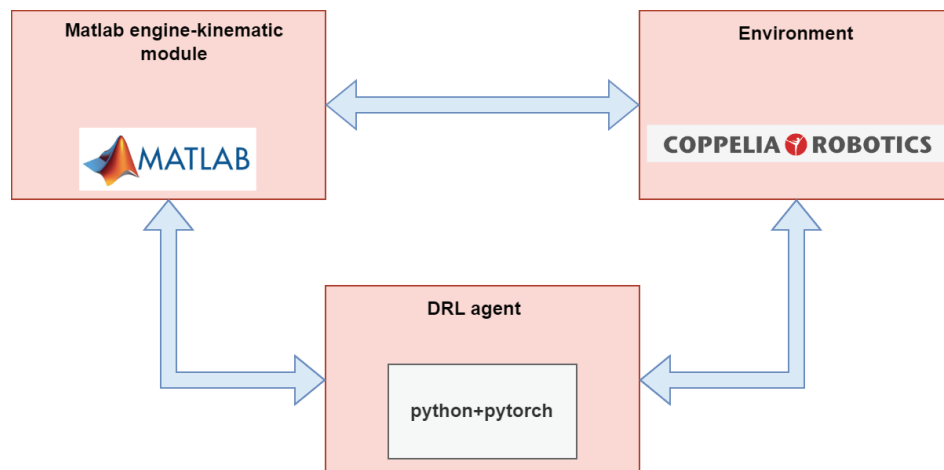


Figure 5. Network connections in the simulation framework.

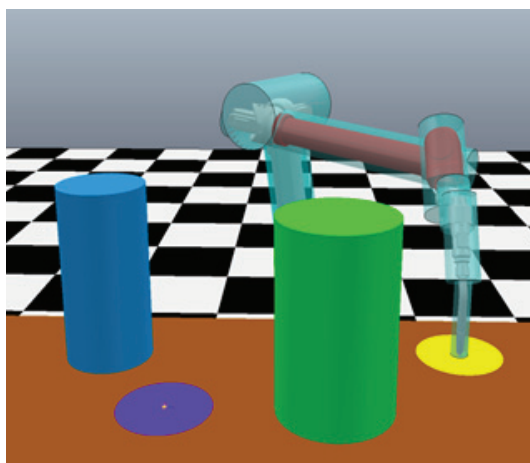


Figure 6. Diagram of the experimental scene.

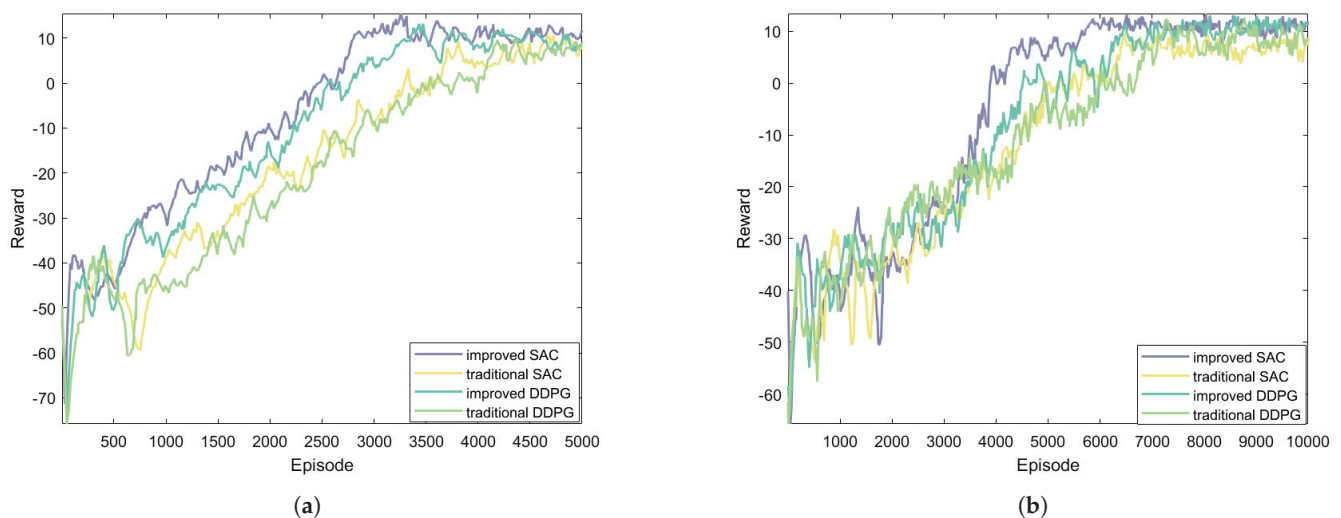
4.2. Algorithm Performance Evaluation

The reward function parameters employed in the experiments are presented in Table 2.

Table 2. Parameters of the reward function.

Parameter	Value	Parameter	Value
δ	0.1	d_b	0.2
ρ	2	k_o	5
η	0.05		

The experiments involved comparing the traditional DDPG and SAC algorithms with two improved versions of these training algorithms, incorporating a dynamic policy selection mechanism and adaptive rewards. Figure 7 illustrates the growth curve of reward values over 5000 episodes for the traditional algorithms, as well as the corresponding performance of the improved algorithms in Environment 1. The improved algorithms typically converged at around 3000 episodes, whereas the traditional algorithms generally took approximately 4000 episodes to converge. In Environment 2, the introduction of randomly moving obstacles complicated the training environment. To ensure that the trained policy models could handle nearly all situations, the number of training episodes was increased to 10,000. The resulting reward curves, as shown in the figure, indicate that the improved SAC algorithm began to converge around 6000 episodes, while the traditional algorithms generally converged at around 7000 episodes. Evidently, the improved algorithms exhibited significant advancements over the traditional algorithms. The results presented in the figure demonstrate that the dynamic exploration guidance mechanism and adaptive rewards contribute positively to the convergence speed and stability of the algorithms.

**Figure 7.** Comparison of four algorithms in (a) Environment 1 and (b) Environment 2.

To evaluate the model's adaptability to complex environments, we conducted tests within a simulated setting, specifically utilizing the more intricate Environment 2. Figure 8 illustrates the complete process of dynamic obstacle avoidance while reaching the target within a single episode.

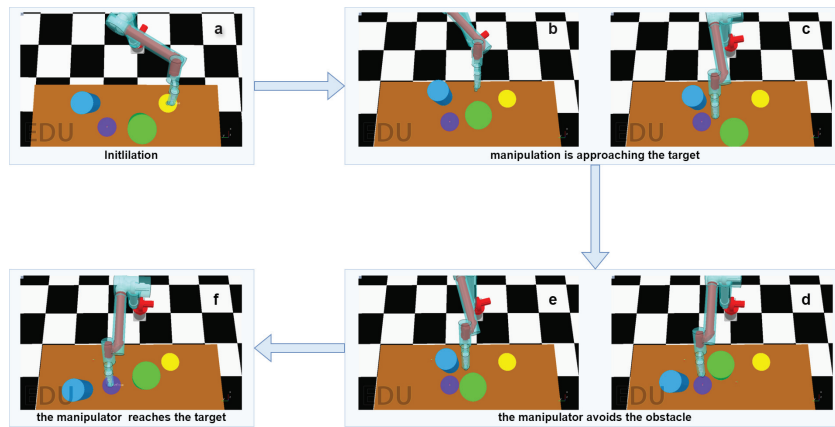


Figure 8. The entire path-planning and dynamic obstacle-avoidance process of the manipulator. (a–f) illustrate key stages of the task: (a) Initialization of the manipulator; (b,c) The manipulator moves toward the target while detecting the surrounding environment; (d,e) The manipulator encounters the dynamic obstacle and adjusts its path to avoid it; (f) The manipulator successfully reaches the target.

In environment 2, although the obstacles were designed to move randomly during the training phase to introduce dynamic complexity, they were configured to move along predefined straight-line trajectories during the testing phase (as shown in Figure 9). This setup was intentionally chosen to control variables and ensure consistent environmental conditions across different algorithms, thereby enabling a fair and direct performance comparison. Since both the initial point and the target point were located on the tabletop surface, and the reward function encourages the robot to complete the task in as few steps as possible, the motion of the robot's end-effector showed only minor variation along the z-axis and was primarily confined to the x-y plane.

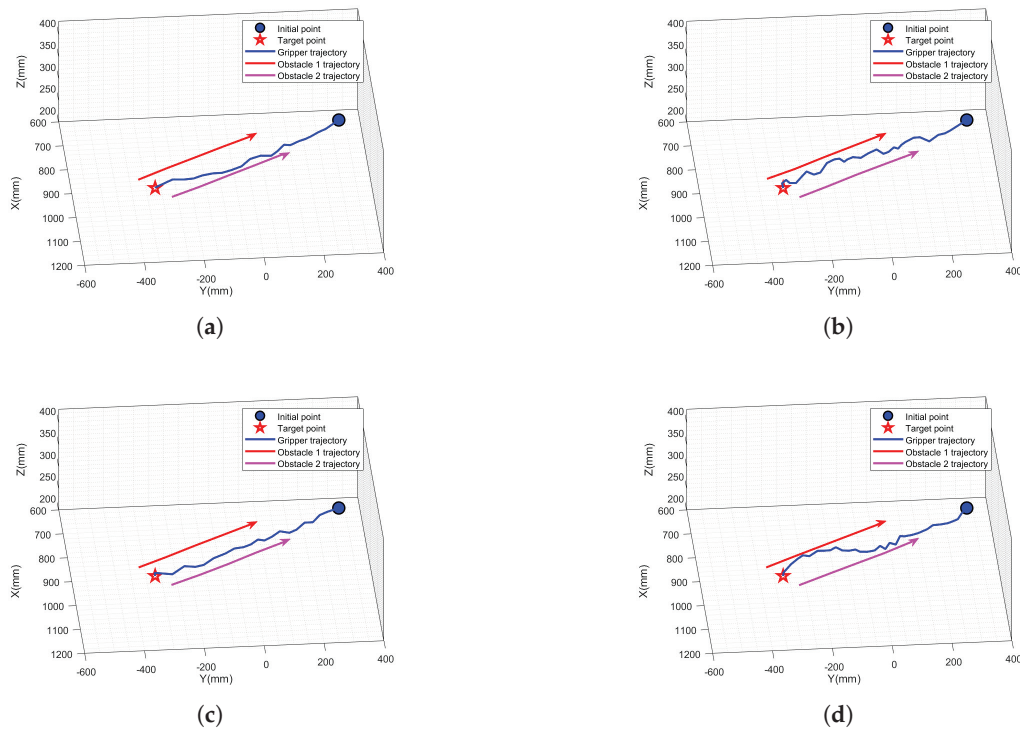


Figure 9. The robot manipulator and obstacle trajectories: (a) improved SAC algorithm; (b) traditional SAC algorithm; (c) improved DDPG algorithm; and (d) traditional DDPG algorithm.

We used the four algorithms to perform multiple simulations in the same dynamic simulation environment. The trajectory diagrams obtained from the experiment are shown in Figure 9. In this figure, a blue sphere is used to denote the starting point, while the red five-pointed star represents the end point. The robot's motion trajectory is depicted in blue, and the motion trajectories of the obstacles are shown in red and pink. It is evident that the robot avoided collision with obstacles throughout its movement. Figure 9a,c shows the trajectory diagrams for the improved algorithm, from which it is obvious that the trajectories of the improved algorithms were shorter and smoother than those of the traditional algorithms.

A total of 500 test trials were performed. The final results are presented in Table 3, which includes the average success rate, average reward, and average number of steps, as well as the reward variance and step variance for both the traditional algorithms and their improved versions under the condition of two randomly moving obstacles. The results indicate that the SAC algorithm is more suitable than the DDPG algorithm for the obstacle-avoidance path-planning of robotic arms in high-dimensional spaces. However, the method proposed in this paper enhanced the performance metrics of both algorithms, thereby validating the stability and robustness of the proposed approach.

In particular, the reward variance and step variance further confirm the improvements in algorithm stability and efficiency. For SAC, the reward variance decreased from 8.47 to 4.13, a reduction of approximately 51.3%, and the step variance dropped from 20.55 to 8.27, a reduction of about 59.8%. For DDPG, the reward variance decreased from 11.87 to 6.63 (a 44.2% reduction), and the step variance was reduced from 27.57 to 11.90 (a 56.8% reduction). These significant decreases in variance clearly demonstrate that the improved methods produce a more stable, consistent performance across multiple trials. In our experiments, the obstacle speed was reasonably set to within a realistic range. The robot joint velocity during obstacle avoidance remained smooth and complied with both the joint position and velocity constraints, making the entire motion process feasible for future real-world implementations.

These findings demonstrate that the newly proposed reinforcement learning-based obstacle-avoidance path-planning method is not only practical but also more robust and reliable. By significantly improving both the average performance and its stability, the method proves effective in accomplishing real-time obstacle avoidance tasks for robotic arms operating in dynamic, high-dimensional environments.

Table 3. Test results.

Policy	Average Success Rate	Average Reward	Reward Variance	Average Step	Step Variance
Traditional SAC	75.25	10.0673	8.4654	28.24	20.55
Improved SAC	86.65	11.6967	4.1266	21.58	8.27
Traditional DDPG	71.33	9.1696	11.8709	30.25	27.57
Improved DDPG	80.56	10.8799	6.6281	24.58	11.90

5. Discussion

In this study, we proposed a reinforcement learning (RL)-based path-planning approach for a robotic manipulator in dynamic obstacle-avoidance tasks, integrating inverse kinematics (IK) with a dynamic exploration guidance mechanism. Path-planning under dynamic conditions involves two fundamental challenges: collision-free navigation around obstacles and successful arrival at designated targets. Due to the large and continuous nature of the state space in dynamic environments, traditional reinforcement learning ap-

proaches often require significant computational resources and extensive training iterations to deduce the state-action relationships effectively.

To overcome these challenges, we introduced a novel hybrid action formulation that combines the deterministic velocity output from an IK module with an adaptive compensatory action generated by a deep reinforcement learning (DRL) policy network. Specifically, this compensatory action dynamically guides the manipulator to effectively avoid obstacles while maintaining goal-directed motion. Furthermore, our method incorporates a dynamic scaling factor to adaptively modulate exploration intensity. This scaling factor rapidly increases at the early training stages, promoting effective exploration and preventing inefficient exploration, and gradually slows down in later stages to refine the manipulator's movements and decision-making precision. Additionally, the designed adaptive reward mechanism derived from an energy function further supports effective policy training by improving the exploration–exploitation balance.

The effectiveness of our proposed method was clearly demonstrated through rigorous comparisons with traditional DRL algorithms. The experimental results indicate that our improved SAC algorithm achieved an average success rate of 86.65%, significantly higher than the 75.25% success rate of traditional SAC. Likewise, our improved DDPG method reached a success rate of 80.56%, markedly surpassing the traditional DDPG's 71.33%. Moreover, the improved algorithms exhibited considerably lower reward and step variances, implying more stable and robust performance. Specifically, the improved SAC and DDPG recorded reward variances of 4.1266 and 6.6281, respectively, compared to their traditional counterparts (8.4654 and 11.8709). Step variances also decreased significantly from 20.55 to 8.27 (SAC) and from 27.57 to 11.90 (DDPG).

The superior performance of the proposed approach compared to traditional methods can be attributed primarily to the underlying strategy of transforming an infinite or extensive state space into a manageable finite subset. Traditional reinforcement learning methods often suffer from inefficient convergence due to their exhaustive exploration of irrelevant states, which contributes to higher variance and reduced success rates. In contrast, our dynamic exploration guidance, combined with adaptive reward-shaping and IK-based policy refinement, significantly narrows the exploration scope. This targeted approach not only accelerates convergence but also ensures more informed and precise decision-making, substantially enhancing the robot's real-time path-planning capabilities.

6. Conclusions

In this study, we addressed the problem of obstacle-avoidance trajectory-planning for robotic arms in dynamic environments by proposing a planning method that integrates a dynamic exploration guidance strategy with adaptive rewards. The algorithm leverages the principles of deep reinforcement learning, modeling the research problem as an MDP. Taking into account the interactive training between the robot and its environment, we introduced and designed the state space, action space, and reward function. To enhance the training convergence efficiency and control performance of the algorithm, the planning process undergoes dynamic adjustments, allowing a dynamic exploration guidance mechanism to be developed. This mechanism is guided by inverse kinematics during the initial phase of training and shifts to random exploration in the later stages. Additionally, an adaptive reward function is incorporated, enabling the robotic arm to focus not only on obstacle avoidance but also on task execution. Ultimately, a policy model that is capable of addressing the research problem can be trained. The effectiveness of the proposed algorithm was validated through simulation experiments. The experimental results demonstrated that, compared to traditional approaches, the proposed method facilitates faster

convergence, achieves higher success rates, and produces robotic arm control strategies with a superior performance.

Author Contributions: Conceptualization, S.Y.; methodology, S.Y. and Y.Z.; validation, S.Y., Y.Z. and C.Z.; investigation, S.Y., Y.Z., C.Z., and Q.C.; writing, S.Y., Y.Z., W.C., and J.Z.; supervision, Y.Z. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Jiangsu Graduate Research Fund Innovation Project (grant number KYCX23_3059), and the APC was funded by Changzhou University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to considerations of privacy protection and ethical principles.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Niloy, M.A.K.; Shama, A.; Chakraborty, R.K.; Ryan, M.J.; Badal, F.R.; Tasneem, Z.; Ahamed, M.H.; Moyeen, S.I.; Das, S.K.; Ali, M.F.; et al. Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review. *IEEE Access* **2021**, *9*, 35338–35370. [CrossRef]
2. Ekrem, Ö.; Aksoy, B. Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106099. [CrossRef]
3. Azizi, M.R.; Rastegarpanah, A.; Stolkin, R. Motion planning and control of an omnidirectional mobile robot in dynamic environments. *Robotics* **2021**, *10*, 48. [CrossRef]
4. Shi, W.; Wang, K.; Zhao, C.; Tian, M. Obstacle avoidance path planning for the dual-arm robot based on an improved RRT algorithm. *Appl. Sci.* **2022**, *12*, 4087. [CrossRef]
5. Park, S.O.; Lee, M.C.; Kim, J. Trajectory planning with collision avoidance for redundant robots using jacobian and artificial potential field-based real-time inverse kinematics. *Int. J. Control Autom. Syst.* **2020**, *18*, 2095–2107. [CrossRef]
6. Semnani, S.H.; Liu, H.; Everett, M.; De Ruiter, A.; How, J.P. Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3221–3226. [CrossRef]
7. Finean, M.N.; Petrović, L.; Merkt, W.; Marković, I.; Havoutis, I. Motion planning in dynamic environments using context-aware human trajectory prediction. *Robot. Auton. Syst.* **2023**, *166*, 104450. [CrossRef]
8. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **2018**, *362*, 1140–1144. [CrossRef]
9. Zhu, C.; Zhu, J.; Si, W.; Wang, X.; Wang, F. Multi-agent reinforcement learning with synchronized and decomposed reward automaton synthesized from reactive temporal logic. *Knowl.-Based Syst.* **2024**, *306*, 112703. [CrossRef]
10. Li, Y.; Li, X.; Wei, X.; Wang, H. Sim-real joint experimental verification for an unmanned surface vehicle formation strategy based on multi-agent deterministic policy gradient and line of sight guidance. *Ocean Eng.* **2023**, *270*, 113661. [CrossRef]
11. Liu, C.; Gao, J.; Bi, Y.; Shi, X.; Tian, D. A multitasking-oriented robot arm motion planning scheme based on deep reinforcement learning and twin synchro-control. *Sensors* **2020**, *20*, 3515. [CrossRef] [PubMed]
12. Sun, T.; Gu, J.; Mou, J. UAV autonomous obstacle avoidance via causal reinforcement learning. *Displays* **2025**, *87*, 102966. [CrossRef]
13. Xu, Q.; Zhang, T.; Zhou, K.; Lin, Y.; Ju, W. Active Collision Avoidance for Robotic Arm Based on Artificial Potential Field and Deep Reinforcement Learning. *Appl. Sci.* **2024**, *14*, 4936. [CrossRef]
14. Wang, T.; Ruan, Z.; Wang, Y.; Chen, C. Control strategy of robotic manipulator based on multi-task reinforcement learning. *Complex Intell. Syst.* **2025**, *11*, 175. [CrossRef]
15. Yadav, S.P.; Nagar, R.; Shah, S.V. Learning vision-based robotic manipulation tasks sequentially in offline reinforcement learning settings. *Robotica* **2024**, *42*, 1715–1730. [CrossRef]
16. Garrido-Castañeda, S.I.; Vasquez, J.I.; Antonio-Cruz, M. Coverage Path Planning Using Actor–Critic Deep Reinforcement Learning. *Sensors* **2025**, *25*, 1592. [CrossRef]

17. Zhao, W.; Zhang, Y.; Xie, Z. EPPE: An Efficient Progressive Policy Enhancement framework of deep reinforcement learning in path planning. *Neurocomputing* **2024**, *596*, 127958. [CrossRef]
18. Zhong, J.; Wang, T.; Cheng, L. Collision-free path planning for welding manipulator via hybrid algorithm of deep reinforcement learning and inverse kinematics. *Complex Intell. Syst.* **2022**, *8*, 1899–1912. [CrossRef]
19. Cao, G.; Bai, J. Multi-agent deep reinforcement learning-based robotic arm assembly research. *PLoS ONE* **2025**, *20*, e0311550. [CrossRef]
20. Liu, J.; Yap, H.J.; Khairuddin, A.S.M. Path Planning for the Robotic Manipulator in Dynamic Environments Based on a Deep Reinforcement Learning Method. *J. Intell. Robot. Syst.* **2025**, *111*, 3. [CrossRef]
21. Wang, Y.; Wang, L.; Zhao, Y. Research on door opening operation of mobile robotic arm based on reinforcement learning. *Appl. Sci.* **2022**, *12*, 5204. [CrossRef]
22. Wu, Y.H.; Yu, Z.C.; Li, C.Y.; He, M.J.; Hua, B.; Chen, Z.M. Reinforcement learning in dual-arm trajectory planning for a free-floating space robot. *Aerosp. Sci. Technol.* **2020**, *98*, 105657. [CrossRef]
23. Lindner, T.; Milecki, A. Reinforcement learning-based algorithm to avoid obstacles by the anthropomorphic robotic arm. *Appl. Sci.* **2022**, *12*, 6629. [CrossRef]
24. Prianto, E.; Kim, M.; Park, J.H.; Bae, J.H.; Kim, J.S. Path planning for multi-arm manipulators using deep reinforcement learning: Soft actor-critic with hindsight experience replay. *Sensors* **2020**, *20*, 5911. [CrossRef] [PubMed]
25. Singla, A.; Padakandla, S.; Bhatnagar, S. Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 107–118. [CrossRef]
26. Xu, J.; Hou, Z.; Wang, W.; Xu, B.; Zhang, K.; Chen, K. Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1658–1667. [CrossRef]
27. Rahmatizadeh, R.; Abolghasemi, P.; Behal, A.; Bölöni, L. From virtual demonstration to real-world manipulation using LSTM and MDN. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
28. Chen, P.; Pei, J.; Lu, W.; Li, M. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. *Neurocomputing* **2022**, *497*, 64–75. [CrossRef]
29. Li, Y.; Zhang, C.; Chai, L. Collaborative obstacle avoidance trajectory planning for mobile robotic arms based on artificial potential field DDPG algorithm. *Comput. Integr. Manuf. Syst.* **2023**, *30*, 4282.
30. Paredis, C.J.; Khosla, P.K. Kinematic design of serial link manipulators from task specifications. *Int. J. Robot. Res.* **1993**, *12*, 274–287. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Multi-Robot Task Allocation with Spatiotemporal Constraints via Edge-Enhanced Attention Networks

Yixiang Hu [†], Daxue Liu ^{*,†}, Jinhong Li, Junxiang Li and Tao Wu

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China; huyixiang19@nudt.edu.cn (Y.H.); lijinhong19@nudt.edu.cn (J.L.); lijunxiang@nudt.edu.cn (J.L.); wutao@nudt.edu.cn (T.W.)

* Correspondence: daxueliu@nudt.edu.cn

[†] These authors contributed equally to this work.

Abstract

Multi-Robot Task Allocation (MRTA) with spatiotemporal constraints presents significant challenges in environmental adaptability. Existing learning-based methods often overlook environmental spatial constraints, leading to spatial information distortion. To address this, we formulate the problem as an asynchronous Markov Decision Process over a directed heterogeneous graph and propose a novel heterogeneous graph neural network named the Edge-Enhanced Attention Network (E2AN). This network integrates a specialized encoder, the Edge-Enhanced Heterogeneous Graph Attention Network (E2HGAT), with an attention-based decoder. By incorporating edge attributes to effectively characterize path costs under spatial constraints, E2HGAT corrects spatial distortion. Furthermore, our approach supports flexible extension to diverse payload scenarios via node attribute adaptation. Extensive experiments conducted in simulated environments with obstructed maps demonstrate that the proposed method outperforms baseline algorithms in task success rate. Remarkably, the model maintains its advantages in generalization tests on unseen maps as well as in scalability tests across varying problem sizes. Ablation studies further validate the critical role of the proposed encoder in capturing spatiotemporal dependencies. Additionally, real-time performance analysis confirms the method's feasibility for online deployment. Overall, this study offers an effective solution for MRTA problems with complex constraints.

Keywords: multi-robot task allocation; reinforcement learning; deep learning; graph neural network

1. Introduction

Multi-Robot Task Allocation (MRTA) stands as a critical challenge in the collaboration of multi-robot systems. Its core objective lies in determining the optimal “task–robot” matching through effective coordination mechanisms, aiming to maximize the overall system utility (e.g., task success rate) while ensuring conflict-free decision-making [1]. With rapid advancements in autonomous technologies, MRTA has demonstrated essential value in fields such as precision agriculture [2], logistics and warehousing [3], unmanned reconnaissance [4,5], hazardous environment monitoring [6,7], and cooperative mapping [8]. In these real-world applications, the complexity of the environment and the diversity of tasks pose severe challenges to the efficiency and robustness of allocation algorithms.

According to the taxonomy proposed by Gerkey [1], MRTA problems are categorized based on three axes: robot type (Single-Task vs. Multi-Task), task type (Single-Robot vs.

Multi-Robot) and assignment type (Instantaneous vs. Time-extended). This study focuses on the allocation problems with single-task robots, single-robot tasks, and time-extended assignments (ST-SR-TA), where a single robot executes tasks serially and independently. When tasks are subject to deadlines and robots are permitted to execute a sequence of tasks, the problem falls into the category of In-schedule Dependencies (ID). Nunes et al. [9] indicate that MRTA with temporal constraints represents the most intensively studied scenario, aligning with the rigorous deadline requirements of most practical applications.

MRTA is a typical NP-hard combinatorial optimization problem that has been addressed by a suite of classic approaches, including exact methods like Mixed Integer Linear Programming (MILP) [10,11] and approximate methods such as clustering algorithms [12,13], heuristics [14,15], meta-heuristics [16–18], and market-based mechanisms [19,20]. Exact methods centrally optimize and thus guarantee global optimality, whereas approximate methods trade optimality for real-time efficiency. Nevertheless, both non-learning paradigms struggle with large-scale or dynamic instances. On the one hand, exact solvers suffer from combinatorial explosion, consequently, their runtime grows prohibitively as problem size increases. On the other hand, heuristics depend on hand-crafted rules that are tightly coupled to specific scenarios, limiting their ability to generalize when the environment changes.

MRTA is a typical NP-hard combinatorial optimization problem that has been addressed by a suite of classic approaches, generally categorized into exact and approximate methods. Exact methods, such as Mixed Integer Linear Programming (MILP) formulated in [11] and the CLIP algorithm in [10], centrally optimize to guarantee global optimality. However, they suffer from combinatorial explosion, causing runtimes to grow prohibitively as problem size increases. To address real-time constraints, approximate methods trade optimality for efficiency. These include clustering algorithms [12,13], heuristics [14,15], meta-heuristics [16–18], market-based mechanisms [19,20], and graph-based approaches like the BiG-MRTA [10]. While effective in their respective domains, these non-learning paradigms encounter significant challenges when applied to the complex spatial environments with obstacles considered in this study. Specifically, graph-based and heuristic methods often rely on simplified metrics (e.g., Euclidean distance) to maintain responsiveness, which fails to capture the non-linear path costs imposed by obstacles. Conversely, incorporating precise path planning into their iterative evaluation processes (e.g., fitness calculation in meta-heuristics or edge weighting in graph matching) would introduce a prohibitive computational bottleneck. Furthermore, rule-based heuristics often depend on scenario-specific designs, limiting their generalization across varying map topologies.

In recent years, the rapid development of deep learning has offered a new paradigm for solving combinatorial optimization problems. Kool et al. [21] proposed modeling the Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP) as sequential decision-making frameworks, achieving near-optimal results using deep reinforcement learning. Inspired by this, Wang et al. [22] incorporated the MRTA problem into similar frameworks, achieving promising results in temporal-constrained scenarios via imitation learning; however, the reliance on expert data limits its generalization ability. To improve scalability, recent studies [23,24] represented the MRTA problem as a graph structure utilizing Capsule Networks and Covariant Compositional Networks for feature extraction.

However, despite retaining the effective sequential decision-making backbone, these methods often overlook the environmental spatial constraints that are inextricably linked to execution time. Existing approaches typically encode the coordinate attributes of robots and tasks directly, implicitly assuming that the distance between nodes is the Euclidean distance. However, in complex real-world environments characterized by high-density obstacles or structural partitions, the Euclidean distance fails to reflect actual path costs,

leading to spatial information distortion. A previous study [25] considered spatial constraints in MRTA, yet its focus remained on avoiding conflicts within robot workspaces rather than addressing the fundamental impact of obstructed maps on path planning and task reachability.

To capture these complex dependencies, Graph Neural Networks (GNNs) incorporating edge attributes, exemplified by architectures such as EGAT [26] and HGT [27], have emerged as a promising direction. In the domain of robotic task scheduling, recent works have adopted these concepts; notably, Gao et al. [28] utilized edge-featured GATs to model transmission delays, while Ma et al. [29] employed heterogeneous graphs to describe collaborative relationships. Nevertheless, a structural limitation persists regarding the integration of physical constraints within these architectures. Most existing approaches incorporate edge attributes primarily to modulate attention weights, effectively treating them as auxiliary scalar biases. This design implies that while edge attributes influence the aggregation mechanism, specific edge information becomes implicitly fused into node features rather than being propagated as a distinct component. In scenarios characterized by strict spatiotemporal coupling, reliance on such implicit fusion potentially compromises the fidelity of physical constraints within the final embeddings, thereby constraining the solver's ability to discern tasks that exhibit logical proximity but remain physically distant.

To address this limitation, this study formulates the MRTA problem as a Markov Decision Process over a directed heterogeneous graph and proposes the Edge-Enhanced Attention Network (E2AN). Central to this framework is the Edge-Enhanced Heterogeneous Graph Attention Network (E2HGAT) encoder. Distinct from generic edge-aware GNNs, E2HGAT is architecturally engineered to specifically preserve the fidelity of physical constraints against implicit fusion. It incorporates an MPNN-inspired [30] spatial pre-injection to ground node features with physical distances and employs a novel dual-path residual mechanism driven by edge attributes. This residual structure establishes a dedicated information channel for physical constraints, ensuring they are explicitly preserved alongside semantic features processed by edge-enhanced aggregation. Furthermore, a dynamic gating mechanism is introduced to adaptively fuse these spatial insights with high-level representations. Finally, we integrate an attention-based decoder to construct an asynchronous sequential decision-making framework. This integrated design not only corrects spatial information distortion in complex environments but also enhances the model's generalization across varying problem scales, with the flexibility to adapt node attributes for diverse payload requirements.

The main contributions of this paper can be summarized as:

1. To address environmental spatial constraints in temporal-constrained MRTA, we propose a Markov Decision Process (MDP) model founded on a directed heterogeneous graph. This model operates within an asynchronous sequential decision-making framework and accommodates extensions for diverse payload requirements. This formulation transforms the high-dimensional combinatorial optimization problem into sequential assignments for deep reinforcement learning. Ultimately, the integrated paradigm intrinsically embeds environmental spatial constraints within the MRTA architecture, ensuring that allocation decisions align with physical realities.
2. To adapt to the directed heterogeneous graph modeling, we propose the E2AN. Built upon an encoder–decoder architecture, the core E2HGAT encoder fuses spatial edge attributes with node features to correct spatial information distortion. Coupled with an attention-based decoder, the network enhances the policy model's capacity to capture spatiotemporal dependencies inherent in the coupled constraints.
3. We constructed a simulated environment considering temporal and environmental spatial constraints. Through multi-dimensional experimental evaluation—covering

in-distribution testing, cross-scenario generalization, and cross-scale scalability testing—we verified the effectiveness of the proposed method in both basic scenarios and extended diverse payload scenarios. Benchmarking against baselines and ablation variants shows higher task success rates across different map environments and problem sizes. Furthermore, the analysis of real-time performance confirms the method’s computational efficiency and feasibility for online deployment.

2. Problem Formulation

This section focuses on the ST-SR-TA class of Multi-Robot Task Allocation problems with temporal and environmental spatial constraints. In this scenario, tasks far outnumber robots, so each robot must navigate through the environment to carry out multiple tasks in sequence. This study centers on ground robots whose motion plans are constrained by regional traversability. Building on this setting, the environmental spatial constraints are encoded directly into the optimization model by representing obstacle-partitioned regions on a grid map. The formulation is further generalized to accommodate diverse payload requirements, ensuring that the robot’s working capacity matches the requirements of the assigned tasks. The subsequent subsections will first provide a detailed description and formal formulation of the MRTA problem, followed by its modeling as a Markov Decision Process over a heterogeneous graph.

2.1. MRTA Problem Description and Formulation

We formulate the problem as a MILP model. Building on temporal-constrained approaches [10,31] and incorporating environmental spatial constraints from the map, we state the model as follows. Let $\mathcal{R} = \{1, \dots, n_R\}$ denote the set of robots, and $\mathcal{T} = \{1, \dots, n_T\}$ denote the set of tasks. Each task i is characterized by a workload w_i and a deadline D_i . The working capacity of robot r is denoted by c_r , and the service time of task i is given by $s_{r,i} = w_i / c_r$. To satisfy spatial constraints, we precompute exact robot-to-task travel distances offline with path planning algorithms like A* instead of using Euclidean estimates on the 2D map. Assuming a uniform and consistent travel speed for the robots, we derive the exact travel time $t_{i,j}$ between locations and the arrival time $t_{r,i}^{\text{start}}$. The decision variables of this problem are defined as follows: the binary variable $y_{r,s,i}$ equals 1 if task i is assigned as the s -th task in robot r ’s sequence, where $S_{\max}$ is the maximum number of tasks per robot. Variable x_i equals 1 if task i is completed successfully. Additionally, the task start time T_i and completion time C_i are defined as continuous variables. The domains of the relevant variables are defined as follows:

$$y_{r,s,i} \in \{0, 1\}, \quad x_i \in \{0, 1\}, \quad T_i \geq 0, \quad C_i \geq 0. \quad (1)$$

The optimization objective is to maximize the number of completed tasks that satisfy deadline constraints. The objective function is formulated as:

$$\max n_{\text{success}} = \sum_{i \in \mathcal{T}} x_i. \quad (2)$$

The constraints for the MILP model are described in Equations (3)–(8). Equation (3) ensures that a task is assigned to a single robot and occupies a unique position in the sequence. To linearize conditional constraints and ensure they apply only to active robot-task assignments, a sufficiently large constant M is introduced. The value of M is set to twice the maximum deadline, i.e., $M = 2 \max_i D_i$. Equation (4) enforces the validity of all assignments, ensuring that completion times do not exceed deadlines. Due to temporal constraints, if a task can be successfully assigned to a robot (satisfying the deadline), it is considered successfully completed, as implied by Equation (5). Equation (6) guarantees

that the completion time of a task is greater than its start time plus the execution duration, where $\mathcal{S} = \{0, \dots, S_{\max}\}$ denotes the range of the task sequence. Equation (7) ensures that the sequential arrangement of tasks adheres to spatial constraints, where $\hat{\mathcal{S}} = \mathcal{S} \setminus \{S_{\max}\}$. Equation (8) guarantees that every robot can reach its first assigned task from its start point.

$$\sum_{r \in \mathcal{R}} \sum_{s=0}^{S_{\max}} y_{r,s,i} = 1 \quad \forall i \in \mathcal{T}, \quad (3)$$

$$C_i \leq D_i + M \cdot (1 - x_i) \quad \forall i \in \mathcal{T}, \quad (4)$$

$$x_i = \sum_{r \in \mathcal{R}} \sum_{s=0}^{S_{\max}} y_{r,s,i} \quad \forall i \in \mathcal{T}, \quad (5)$$

$$C_i \geq T_i + s_{r,i} - M \cdot (1 - y_{r,s,i}) \quad \forall r \in \mathcal{R}, \forall s \in \mathcal{S}, \forall i \in \mathcal{T}, \quad (6)$$

$$T_j \geq C_i + t_{i,j} - M \cdot (2 - y_{r,s,i} - y_{r,s+1,j}) \quad \forall r \in \mathcal{R}, \forall s \in \hat{\mathcal{S}}, \forall i, j \in \mathcal{T}, i \neq j, \quad (7)$$

$$T_i \geq t_{r,i}^{\text{start}} - M \cdot (1 - y_{r,0,i}) \quad \forall r \in \mathcal{R}, \forall i \in \mathcal{T}. \quad (8)$$

To address diverse payload requirements, we extend the robot capacity $\mathbf{c}_r$ and the task workload $\mathbf{w}_i$ to K -dimensional vectors in $\mathbb{R}_+^K$. Assuming each task i requires a specific payload type k_i , $\mathbf{w}_i$ is defined as a sparse vector with a non-zero value exclusively in the k_i -th dimension.

This study defines a binary variable $p_{r,k} \in \{0, 1\}$ to indicate if robot r is equipped with payload k . Robots can carry multiple payloads, resulting in multi-dimensional capacity vectors $\mathbf{c}_r$. Payloads are divided into basic (carried by all robots) and specific. We assume each robot carries exactly one specific payload, which determines its category. Accordingly, task requirements are subdivided into basic requirements and specific requirements: a basic requirement corresponds to a basic payload and can be satisfied by any robot; a specific requirement corresponds to a specific payload and can only be satisfied by the corresponding category of robots. Since robots can execute multiple types of tasks, different types of tasks must compete for the same robotic resources, which increases the coupling and complexity of the problem.

To ensure the matching between capabilities and requirements, task i can only be assigned to a robot equipped with the corresponding payload k_i . This constraint is formalized as:

$$\sum_{s=0}^{S_{\max}} y_{r,s,i} \leq p_{r,k_i} \quad \forall r \in \mathcal{R}, i \in \mathcal{T}. \quad (9)$$

This constraint guarantees that a robot lacking the necessary payload cannot execute the task. Accordingly, the service duration $s_{r,i}$ is determined jointly by the workload in the corresponding dimension and the robot's efficiency in that dimension, calculated as:

$$s_{r,i} = \frac{w_{i,k_i}}{c_{r,k_i}}. \quad (10)$$

2.2. Markov Decision Process over a Heterogeneous Graph

Since the essence of the MRTA problem lies in seeking an optimal matching between two sets of entities, we draw upon the classic bipartite graph matching model to formulate the problem as an MDP over a complete directed heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The node set $\mathcal{V} = \mathcal{V}_R \cup \mathcal{V}_T$ comprises two categories: robot nodes and task nodes. Robot node

attributes consist of real-time position, current time, working capability, and operational status (moving, working, idle). Task node attributes include deadline, coordinates, workload, and task status (pending, completed). In this study, both workload and working capability can be extended to K -dimensional vectors to match the diverse payload requirements of the scenario. The edge set $\mathcal{E}$ describes the spatial correlation between robots and tasks, and its weights ω_{ij} are designed to quantify the connection strength between nodes. In scenarios with diverse payload requirements, edges can only be established between robot and task nodes when the payload matching relationship is satisfied. For these valid connections, to eliminate dimensional discrepancies, a normalized weight function based on reachability is defined:

$$\omega_{ij} = \alpha \left(1 - \frac{d_{ij}}{d_{\max}} \right), \quad (11)$$

where d_{ij} is the precise distance solved based on spatial constraints, $d_{\max}$ is the distance calculated between diagonal endpoints of the environmental map, and α is a tuning coefficient. This formula ensures that spatially proximal nodes possess higher correlation weights.

Based on the aforementioned heterogeneous graph structure, we model the task allocation process as a sequential decision-making problem, formalized as an MDP tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$.

State Space $\mathcal{S}$: This describes the global state of the system at decision moment t . Specifically, state S_t is composed of the current dynamic heterogeneous graph $\mathcal{G}_t$ and the static environmental map. The heterogeneous graph stores the current physical states of robots and attribute features of tasks, serving as the primary input for the policy. The environmental map provides spatial constraint information describing traversable areas and obstacles, serving as the basis for accurate state updates. In this research scenario, static obstacles are known a priori, and communication bandwidth among agents is sufficient; thus, we assume the environment is fully observable, meaning the central scheduler can access complete global state information in real-time, avoiding information loss caused by partial observability.

Action Space $\mathcal{A}$: To avoid conflicts arising from concurrent decisions by multiple agents, the decision step does not equal the time step; instead, an asynchronous event-triggered mechanism is adopted. At each decision step t , the system selects the robot r^* that is currently the first idle as the decision agent. The action set is defined as the index set of optional tasks $\{0, \dots, N_T\}$. The selected action must be valid: candidate tasks must be in the pending state and must not be reassigned. If the scenario involves diverse payload requirements, the robot is required to be equipped with the specific payload required by the candidate task.

State Transition $\mathcal{P}$: We formulate the MRTA problem as a deterministic transition model, where a valid task assignment means the task is deterministically completed, without guaranteeing it finishes before its deadline. The model is described in Algorithm 1. When action a_t is executed—i.e., task i is assigned to robot r^* —the graph structure updates accordingly: task i is marked as completed, its completion time is logged, and it is removed from the heterogeneous graph. Meanwhile, the state of robot r^* is updated based on task i , with its position transitioning to the task's endpoint and its time state set to the task completion moment.

Reward $\mathcal{R}$: As current decisions have long-term impacts on the global schedule, quantifying their immediate effect is difficult. Therefore, we adopt a sparse reward mechanism to evaluate the overall solution only at the end episode. The reward function is defined as follows:

$$R = \begin{cases} -\sum_{i=1}^{n_T} R_i, & \text{if } n_{\text{success}} < n_T \\ e^{-T_{\Psi}}, & \text{if } n_{\text{success}} = n_T \end{cases}, \quad (12)$$

where n_{success} is the number of tasks completed on time, R_i includes the overdue penalty per task, and T_{Ψ} denotes the global time cost. Their specific calculation methods are as follows:

$$R_i = \begin{cases} \frac{C_i}{D_i}, & \text{if } C_i > D_i \\ 0, & \text{otherwise} \end{cases}, \quad (13)$$

$$T_{\Psi} = \frac{1}{D_{\max}} \sum_{r \in \mathcal{R}} C_r^{\text{end}}. \quad (14)$$

In Equation (14), C_r^{end} is the completion time of robot r , and $D_{\max}$ represents the maximum deadline. The reward function is designed to balance feasibility and optimality:

- Penalty based on delay length: During the early training, when the policy fails to complete all tasks, we use a penalty R_i based on delay duration instead of the task failure rate. This metric quantifies the degree of failure. It encourages the agent to reduce the overdue duration even when a deadline violation is inevitable, preventing passive task abandonment.
- Reward based on time efficiency: When the policy completes all tasks before their deadlines, the focus shifts to minimizing execution costs to further improve the policy. At this stage, the reward function becomes positive, adopting an exponential decay form $e^{-T_{\Psi}}$. We utilize the total working duration of the robots as the cost metric and use $D_{\max}$ as a scaling factor for normalization.

Finally, to integrate the defined state, action, transition, and reward components into a coherent operational framework, the complete asynchronous decision-making process is detailed in Algorithm 1. This pseudo-code illustrates the complete interaction cycle, demonstrating how the scheduler samples actions for idle agents, updates the global state according to the transition model, and derives the sparse reward, thereby linking the mathematical definitions to the procedural execution.

Algorithm 1 MDP Allocation Algorithm

```

1: Inputs: Robots  $\mathcal{R}$ , Tasks  $\mathcal{T}$ , Map  $\mathcal{M}$ , Policy  $\pi_{\theta}$ 
2: Initialize:  $\mathcal{R}_{\text{active}} \leftarrow \mathcal{R}$ ,  $\mathcal{T}_{\text{pending}} \leftarrow \mathcal{T}$ ,  $t \leftarrow 0$ , Construct  $\mathcal{G}_0$ 
3: while  $\mathcal{T}_{\text{pending}} \neq \emptyset \wedge \mathcal{R}_{\text{active}} \neq \emptyset$  do
4:    $r^* \leftarrow \arg \min_{r \in \mathcal{R}_{\text{active}}} (r.\text{time})$  ▷ Select first idle robot
5:    $\mathbf{m}_t \leftarrow \text{GetMask}(r^*, \mathcal{T}_{\text{pending}})$ 
6:   if  $\mathbf{m}_t = \mathbf{0}$  then
7:      $\mathcal{R}_{\text{active}} \leftarrow \mathcal{R}_{\text{active}} \setminus \{r^*\}$  ▷ No valid tasks
8:   else
9:      $a_t \sim \pi_{\theta}(\mathcal{G}_t) \odot \mathbf{m}_t$  ▷ Sample valid task  $i^*$ 
10:     $S_{t+1} \leftarrow \text{StateUpdate}(S_t, a_t)$ 
11:     $\mathcal{G}_{t+1} \leftarrow \text{UpdateGraph}(\mathcal{G}_t, S_{t+1})$ 
12:     $\mathcal{T}_{\text{pending}} \leftarrow \mathcal{T}_{\text{pending}} \setminus \{a_t\}$  ▷ Update pending task set
13:     $t \leftarrow t + 1$ 
14:   end if
15: end while
16: if  $n_{\text{success}} = n_T$  then
17:    $R \leftarrow e^{-T_{\Psi}}$  ▷ Reward for global time efficiency
18: else
19:    $R \leftarrow -\sum R_i$  ▷ Penalty for failed tasks
20: end if
21: return  $R$ 

```

3. Methods

Based on the Heterogeneous Graph Markov Decision Process constructed in Section 2.2, the present section implements a deep reinforcement learning method within a heterogeneous graph framework. Specifically, this study proposes the Edge-Enhanced Attention Network (E2AN). This method employs an encoder–decoder architecture as the policy model to generate optimal task allocation schemes through asynchronous sequential decision-making, as illustrated in Figure 1.

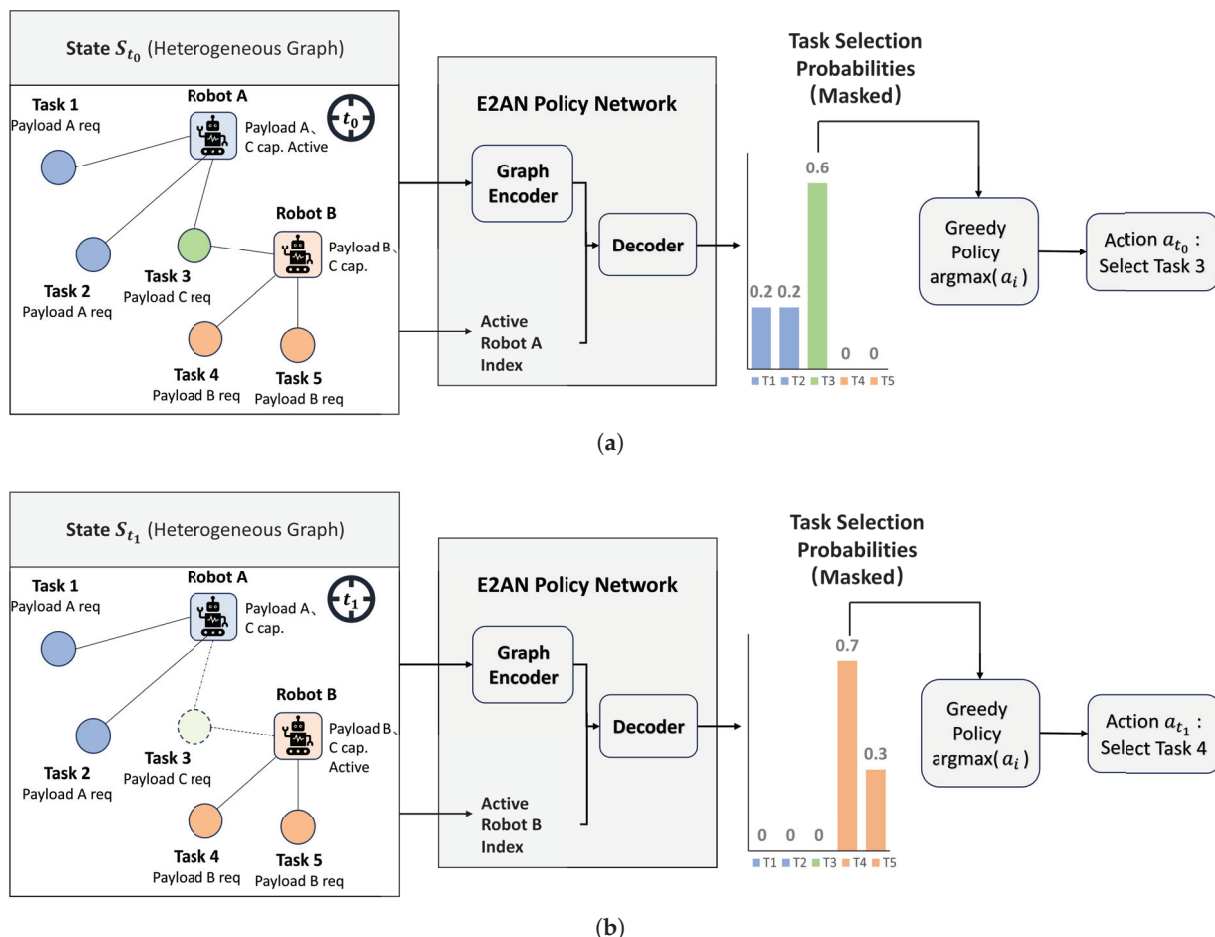


Figure 1. Illustration of an MRTA policy deployment via E2AN. (a) At time t_0 , task 3 is assigned to robot A; (b) at time t_1 , task 4 is assigned to robot B. The E2AN output is reset to zero for the previously assigned task (task 3 in panel (b)) and for tasks that do not satisfy the payload matching relationship due to the lack of corresponding payloads.

Initially, the Edge-Enhanced Heterogeneous Graph Attention Network (E2HGAT) functions as the core encoder and takes the constructed heterogeneous graph as input. It extracts the attributes of robot and task nodes along with their topological connections. The encoder maps all nodes into high-dimensional embeddings containing global spatiotemporal information. These node embeddings are input to the attention decoder. It also receives the index of the robot making decisions, which is the first idle one. The decoder then calculates the selection probability distribution for the robot across all valid tasks. The policy greedily selects actions based on this distribution to interact with the environment, and this process iterates until a complete task sequence is formed. For optimization, the proposed method employs the REINFORCE algorithm with a baseline to reduce variance in gradient estimation and enhance policy convergence.

3.1. Edge-Enhanced Heterogeneous Graph Attention Network Encoder

We propose an Edge-Enhanced Heterogeneous Graph Attention Network encoder as the core encoder module. As shown in Figure 2, taking the raw heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ containing robot and task nodes as input, the encoder maps all nodes into a unified high-dimensional embedding space $\mathbf{H}_{final}$ containing both local attributes and global topological structures through stacked multilayer message passing. Functioning as an “Edge-to-Edge aware” architecture, it distinguishes itself from generic GNN architectures by structurally integrating spatial constraints across all processing stages: initialization, attention modeling, propagation, and fusion.

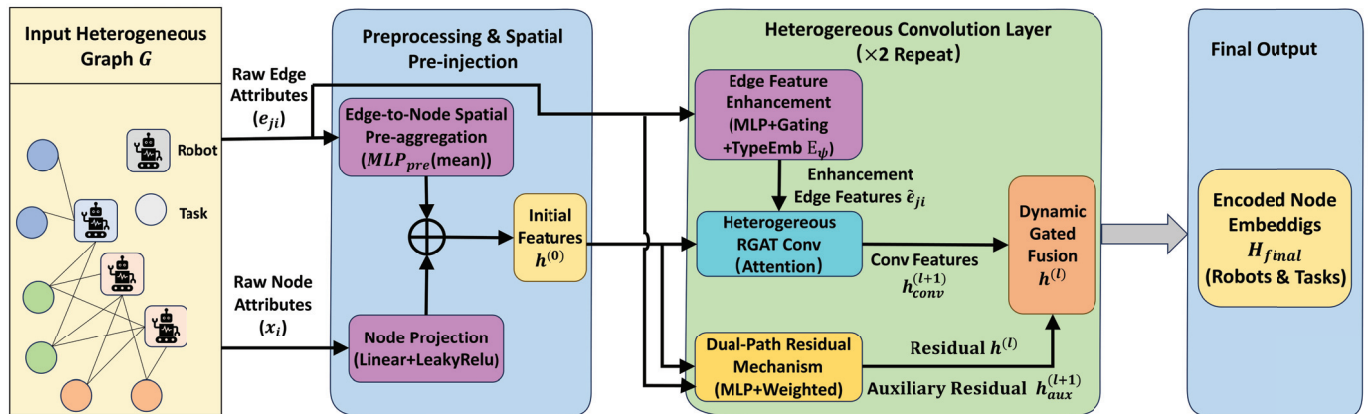


Figure 2. Overview of the E2HGAT Encoder. After initializing $\mathbf{h}^{(0)}$ from raw node attributes $\mathbf{x}_i$ and spatial edge attributes $\mathbf{e}_{ji}$, the model utilizes stacked convolution layers. Within each layer, a Dynamic Gated Fusion module integrates features from the RGAT Convolution (driven by enhanced edges $\tilde{\mathbf{e}}_{ji}$) and the Dual-Path Residual Mechanism. This process iteratively updates embeddings $\mathbf{h}^{(l+1)}$ to generate the final representation $\mathbf{H}_{final}$.

Addressing the MRTA with diverse payloads, the framework projects node attributes of varying lengths, such as k -dimensional capability requirements, into a unified feature space. This mechanism enables the model to adapt to arbitrary input dimensions without altering the core architecture. Unlike homogeneous graph networks, this framework models robots and tasks through type-specific projection layers. This design integrates edge attributes throughout. Therefore, the model captures both the environmental spatial constraints and the highly coupled temporal constraints in MRTA. By leveraging the intrinsic relation-aware characteristics of RGAT, the encoder achieves permutation invariance, so node embeddings are robust to variations in node indexing order. The architecture includes five modules for enhancing spatial awareness: Spatial Information Pre-injection, Edge Feature Enhancement, Heterogeneous RGAT Convolution, Dual-Path Residual Mechanism, and Dynamic Gated Fusion.

Drawing inspiration from the aggregation philosophy of Message Passing Neural Networks (MPNN) [30], we implement a Spatial Pre-injection mechanism. Distinct from standard MPNNs that use edges conditionally during message construction, we position this as a preprocessing step to explicitly embed neighbor physical distances into the initial node representation $\mathbf{h}^{(0)}$ before node attribute interaction. Initially, node features $\mathbf{x}_i$ are converted to the same hidden dimension D using type-specific linear layers.

$$\mathbf{h}_i^{(0)} = \text{LeakyReLU}\left(\mathbf{W}_{\text{proj}}^{\phi(i)} \mathbf{x}_i\right), \quad (15)$$

where $\phi(i)$ represents the node type. The edge attribute $\mathbf{e}_{ji}$ is aggregated through summation operations and then projected into vectors by the multi-layer perceptron (MLP_{pre}), which will be injected into the initial features:

$$\mathbf{h}_i^{(0)} \leftarrow \mathbf{h}_i^{(0)} + \text{MLP}_{pre} \left(\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{e}_{ji} \right), \quad (16)$$

where $\mathcal{N}(i)$ is the set of neighbors for node i .

The encoder consists of two stacked heterogeneous graph attention convolutional layers. This design enables interaction information between indirectly connected nodes, such as tasks competing for the same robot, to propagate through multi-hop message passing.

In each convolutional layer, Edge Feature Enhancement is applied as an initial step to fuse $\mathbf{e}_{ji}$ with its corresponding edge type embedding $\mathbf{E}_{\psi(j,i)}$. Although the topology contains only robot–task connections, directionality is modeled by learning distinct edge type embeddings $\mathbf{E}_{\psi(j,i)}$ conditioned on the start node type. Furthermore, a gating function $\sigma(\cdot)$ is incorporated to suppress long-range or weakly relevant connections, acting as a physical filter before attention calculation. The formulation is defined as

$$\mathbf{e}'_{ji} = \text{LayerNorm}(\text{MLP}_{enc}(\mathbf{e}_{ji})) + \mathbf{E}_{\psi(j,i)}, \quad (17)$$

$$\tilde{\mathbf{e}}_{ji} = \sigma(\mathbf{W}_{gate}[\mathbf{e}'_{ji} \parallel \mathbf{e}_{ji}]) \odot \mathbf{e}'_{ji}, \quad (18)$$

where $\parallel$ is concatenation, $\odot$ denotes element-wise product, and $\tilde{\mathbf{e}}_{ji}$ represents the enhanced edge features.

The RGAT mechanism is used to model graph heterogeneity, allowing it to learn independent parameters and attention patterns for different edge types. Unlike standard RGAT [32] which relies solely on discrete relation types, or EGAT [26] which focuses on continuous attributes, our design implements a hybrid attention. This study introduces enhanced edge features $\tilde{\mathbf{e}}_{ji}$ in the attention coefficient calculation. By concatenating edge and node attributes, we incorporate spatial information into the input, as shown in Equation (19).

$$\alpha_{ji} = \text{softmax} \left(\text{LeakyReLU} \left(\mathbf{a}^T [\mathbf{W}_q \mathbf{h}_i^{(l)} \parallel \mathbf{W}_k \mathbf{h}_j^{(l)} \parallel \mathbf{W}_e \tilde{\mathbf{e}}_{ji}] \right) \right), \quad (19)$$

$$\mathbf{h}_{i,\text{conv}}^{(l+1)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ji} \mathbf{W}_v \mathbf{h}_j^{(l)}, \quad (20)$$

where $\mathbf{h}_i^{(l)}$ and $\mathbf{h}_{i,\text{conv}}^{(l+1)}$ are the l -th layer input features and current layer convolutional output features; $\mathbf{W}_{\{q,k,v,e\}}$ are learnable projection matrices; $\mathbf{a}$ is the attention vector corresponding to the directed edge type; and $\mathcal{N}(i)$ is the neighbor set of node i .

In addition to retaining the standard feature residual connection ($\mathbf{h}^{(l)} + \mathbf{h}_{\text{conv}}^{(l+1)}$), Dual-Path Residual Mechanism introduces an auxiliary Edge-Attribute-Driven residual path. Distinct from standard residual connections that solely preserve node semantic continuity, this mechanism establishes a dedicated channel independent of node semantic similarity. It explicitly transmits edge-conditioned statistical information across layers to prevent spatial constraints from being obscured via implicit fusion with high-dimensional node features. Specifically, raw edge attributes are first mapped to scalar weights $w_{ji} = \sigma(\text{MLP}_{dist}(\mathbf{e}_{ji}))$ via a projector. Subsequently, neighbor features are aggregated based on these weights to

generate the auxiliary residual vector $\mathbf{h}_{i,\text{aux}}^{(l+1)}$. As shown in Equation (21), this mechanism ensures that features of proximal nodes are highlighted and preserved.

$$\mathbf{h}_{i,\text{aux}}^{(l+1)} = \text{MLP}_{\text{proj}}\left(\frac{\sum_j w_{ji} \mathbf{h}_j^{(l)}}{\sum_j w_{ji} + \epsilon}\right), \quad (21)$$

where ϵ is a smoothing term added to prevent the denominator from becoming zero.

At the end of each graph convolutional layer, to adaptively integrate newly extracted topological features with retained residual information, we use Dynamic Gated Fusion. Unlike standard Gated GCNs that rely on node states, E2HGAT introduces an edge-conditioned gating mechanism, enabling the model to adaptively regulate message propagation strength according to edge semantics. Initially, a learnable update gate $\mathbf{z}_i$ is calculated to determine the optimal ratio for feature fusion. Then, the fused features are processed through Layer Normalization and an ELU activation function to generate the final node embedding $\mathbf{h}_i^{(l+1)}$ for the current layer:

$$\mathbf{z}_i = \sigma\left(\mathbf{W}_{\text{update}} \mathbf{h}_{i,\text{conv}}^{(l+1)}\right), \quad (22)$$

$$\mathbf{h}_i^{(l+1)} = \text{LayerNorm}\left(\text{ELU}\left(\mathbf{z}_i \odot \mathbf{h}_{i,\text{conv}}^{(l+1)} + (1 - \mathbf{z}_i) \odot (\mathbf{h}_i^{(l)} + \mathbf{h}_{i,\text{aux}}^{(l+1)})\right)\right). \quad (23)$$

3.2. Multi-Head Attention Mechanism Based Decoder

This study implements a stepwise allocation decoder based on Multi-Head Attention (MHA), which transforms the decoding process into a probability generation task for a specific robot. In each decision-making step, the model selects the first idle robot, calculates the selection probability of the robot for all valid tasks, and then greedily selects the optimal task. This process is repeated until all tasks have been assigned. This point-wise construction mechanism can decompose combinatorial optimization problems into a series of sub-problems, which makes the strategy more robust under different problem scales and agent configurations.

The attention mechanism constructs the Query vector ($\mathbf{q}$) and Key–Value pairs ($\mathbf{K}, \mathbf{V}$). In this model, $\mathbf{K}$ and $\mathbf{V}$ are projections of the task node embeddings $\mathcal{H}_{\text{task}}$ and remain invariant across decision steps. Conversely, the Query vector $\mathbf{q}_t$ is dynamically constructed via a linear projection of the context vector. This context vector integrates the current robot's embedding, the mean embedding of peer robots, and the global mean of task embeddings. This design enables $\mathbf{q}_t$ to act as a probe, selecting the most compatible target from the task feature space.

The decoder performs multi-head scaled dot product attention calculation, using 8 parallel attention heads. In each head h , the dynamic robot Query interacts with the global task Keys to compute compatibility scores. The calculation process is as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \quad (24)$$

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)\mathbf{W}^O, \quad (25)$$

where $\mathbf{W}^O$ is the output projection matrix. The outputs from the multiple heads are concatenated and linearly transformed to generate a latent vector rich in context information for the current decision step.

In the final output stage, to map the latent vector back to the discrete action space, the decoder utilizes a single-head attention mechanism to multiply the MHA output with a Logit $\mathbf{K}$ pointer, obtaining the final pointing score for each task. The Logit $\mathbf{K}$ pointer is derived from the global task node embeddings $\mathbf{H}_{task}$ via a linear transformation. Since the decoding process must ensure the generated task sequence satisfies all constraints and is nonrepetitive, we introduce a hard masking strategy prior to softmax normalization. The system maintains a binary mask based on the current state, forcing the Logit values corresponding to already assigned tasks and tasks incompatible with the current robot's payload capacity to $-\infty$. This operation ensures the probability distribution is defined exclusively over the set of valid tasks. Subsequently, the policy outputs action a_t via greedy decoding based on this distribution, marks the task as completed, and proceeds to the next decision step until N assignments are completed.

3.3. Training Algorithm

Although the MILP formulation constructed in Section 2.1 theoretically provides a mathematical form for optimal solutions, the NP-hard nature of the MRTA problem renders the acquisition of large-scale, high-quality ground truth labels computationally prohibitive as the problem size increases. This computational bottleneck severely restricts the feasibility of supervised learning approaches. Consequently, building on the MDP framework defined in Section 2.2, the proposed method employs the REINFORCE policy gradient algorithm to train the proposed neural network.

To effectively mitigate the variance in gradient estimation, we incorporate a greedy rollout baseline mechanism. The baseline update condition is that the paired t -test shows that the current strategy reward is better than the baseline and p -value is less than the significance level $\alpha = 0.05$. The gradient approximation of the objective function $J(\theta)$ with respect to the model parameters θ is formulated as shown in Equation (26):

$$\nabla_{\theta} J(\theta) \approx \frac{1}{B} \sum_{i=1}^B (R(s_i, a_i) - b(s_i)) \nabla_{\theta} \log p_{\theta}(a_i | s_i), \quad (26)$$

where B denotes the batch size; s_i represents the state of the i -th input instance; a_i is the action generated by the current policy p_{θ} ; and $R(s_i, a_i)$ is the cumulative reward. The term $b(s_i)$ serves as the baseline value, that is, the reward obtained by the baseline policy for the same state s_i .

Independent training and validation sets were generated for each epoch. During training, the decoder employs stochastic sampling instead of greedy strategy to enhance exploration and prevent premature convergence. The validation set evaluates the current policy π_{θ} against the baseline $\pi_{\theta_{BL}}$.

4. Results

To systematically evaluate the generalization and scalability of the proposed method, this study designed the experiments in two progressive stages. In the initial stage, we focused on a basic MRTA scenario involving temporal constraints and environmental spatial constraints. By constructing simulated environments and datasets, we established the performance advantage of our model on basic problems through baseline comparisons, generalization tests, scalability tests and ablation studies. Subsequently, the experimental scenario was extended to include diverse payload requirements, comprehensively verifying the model's adaptability under complex constraints. Finally, we measured the model's inference latency to evaluate the real-time responsiveness of the proposed approach, thereby completing the evaluation pipeline from basic generalization and scalability to online deployability.

4.1. Design of Experiments

This section specifies the simulation setup for the basic MRTA scenario. This study introduced environmental maps to impose spatial constraints, utilizing a benchmark dataset from the field of Multi-Agent Path Finding (MAPF) [33] as the map resource library. The dataset includes mazes, rooms, city, and random obstacles, as shown in Figure 3. To increase the complexity of spatial constraints, we selected structurally benchmark maze map to construct the training environment and used other map types for generalization tests.

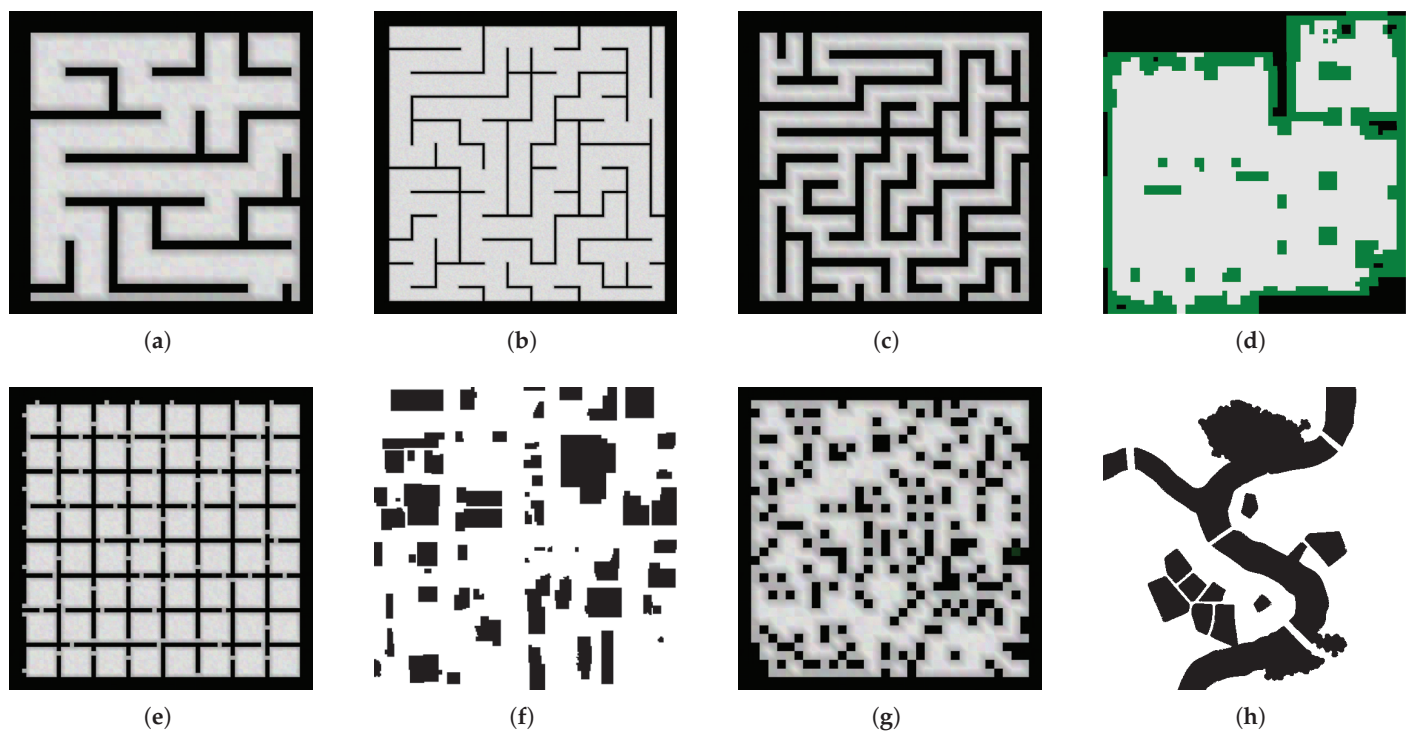


Figure 3. Environment maps selected for this study. (a) Benchmark maze. (b) Normal maze. (c) Complex maze. (d) Simple room. (e) Complex room. (f) City scenario. (g) Random obstacles. (h) Wild scenario. Black and green regions denote obstacles; white and gray regions denote traversable areas.

In the basic scenario configuration, the simulation parameters were adapted from the standard Task Allocation Problem with Time and Capacity (TAPTC) benchmark established by Mitiche et al. [34] and utilized by recent baselines [23]. The map was set as a 2D 100×100 grid to simulate a medium-sized environment characterized by complex spatial obstructions. While the original benchmark considers obstacle-free settings, we incorporated static obstacles to test spatial adaptability. Regarding system scale, we set $n_T = 100$ tasks and $n_R = 7$ robots (the upper bound of the standard dataset). This robot count was selected to provide sufficient total service capacity relative to the task volume in obstructed environments. Temporal and physical parameters were calibrated to maintain consistency with the benchmark while reflecting the dynamics of ground robots: the deadline D_i was uniformly sampled from the interval $[36, 540]$, covering varying urgency levels. A speed of 4 grid units per time step was selected to match the ground robot kinematics with the map scale and environmental navigation constraints. The service capabilities of robots and the workload requirements of tasks were uniformly sampled from the intervals $[1, 2]$ and $[10, 15]$, respectively, modeling realistic variations in operational capacity and task difficulty.

Based on these parameters, we constructed the dataset and trained the model for 100 epochs using the training algorithm described in Section 3.3. Detailed training hyper-

parameters are listed in Table 1. To ensure reproducibility and training stability, specific architectural and optimization parameters were explicitly defined. The E2HGAT encoder utilizes a 2-layer structure to capture sufficient topological depth without over-smoothing; a detailed sensitivity analysis justifying this choice is provided in Appendix A. Meanwhile, while the decoder employs an 8-head attention mechanism to diversify the focus on different task features. Furthermore, a gradient clipping threshold of 1.0 and a tanh clipping of 10.0 were applied to prevent gradient explosion and constrain the logits range during the reinforcement learning process. The test dataset was constructed with same scenario configurations, each of which includes 100 fixed cases.

Table 1. The detailed hyperparameters used for model architecture and training.

Hyperparameter	Value
Epoch	100
Training Dataset	10,000
Validation Dataset	1000
Hidden Embedding Dimension	128
Encoder GNN Layers	2
Attention Heads	8
Optimizer	Adam (PyTorch 2.0.1)
Learning Rate	1×10^{-4}
Gradient Clipping Norm	1.0
Tanh Clipping	10.0
Baseline Significance (α)	0.05
Training Frequency	100

All simulations and algorithm implementations were executed using Python 3.8. The proposed network architecture and training pipeline were developed using PyTorch 2.0.1, while the graph neural network modules within the encoder utilized the PyTorch Geometric (PyG) library (version 2.6.1). Additionally, the SciPy library was employed to accelerate path planning calculations in the simulation. All experiments were conducted on a workstation equipped with an Intel Core i9-10900KF CPU (@ 3.70 GHz) (Intel, Santa Clara, CA, USA), 64GB of RAM, and a single NVIDIA GeForce RTX 3090 GPU with 24 GB VRAM (NVIDIA, Santa Clara, CA, USA).

4.2. Baselines

To comprehensively evaluate the performance of the proposed method, five baselines were selected for comparison, including random selection method, optimization-based method, graph-based method and two state-of-the-art learning-based methods:

1. Random Selection (RD): Based on the asynchronous framework, this method selects actions randomly from valid candidates. To ensure a fair comparison, we applied the masking strategy from Section 3.2 to filter out completed or payload-mismatched tasks, ensuring physical feasibility. This strategy serves to evaluate the difficulty of scenario cases and establishes a performance lower bound. It verifies that the proposed learning algorithm captures actual decision patterns rather than merely relying on constraints.
2. CLIP-MRTA: Adapted from prior work [10], this method formulates the problem as a MILP model using A* distances and environmental spatial constraints. We employed the Gurobi solver (version 11.0.3) [35] with a 10-min runtime cutoff. Due to computational complexity, it yields some feasible rather than optimal solutions within the limited time.

3. Capsule Attention Mechanism (CapAM): This approach [23] models MRTA problem as a homogeneous task graph processed by a graph capsule network. In its decoding stage, the query vector is constructed by concatenating the robot state with temporal and peer information to handle task allocation.
4. Covariant Attention Mechanism (CAM): Distinct from CapAM, this approach [24] employs a covariant compositional network encoder. It enhances representation by extracting local k -nearest neighbor structural features. To ensure a fair comparison, both CapAM and CAM were trained using the identical REINFORCE algorithm and hyperparameter settings (Table 1) as our proposed method.
5. Bipartite Graph-based MRTA (BiG-MRTA): Representing a recent graph-based approach [10], this method maps the allocation problem to a weighted bipartite graph structure. It employs a heuristic incentive model to quantify robot–task suitability as edge weights and subsequently utilizes maximum weighted matching to derive optimal assignments. Designed as an efficient online solver, this approach supports decentralized deployment, serving as a robust baseline for evaluating performance against traditional graph-theoretic algorithms in dynamic environments.

4.3. Generalization and Scalability Analysis

This section evaluates the performance of the E2AN method in basic MRTA scenarios. Initially, this study conducted in-distribution testing to comprehensively compare E2AN with five baseline algorithms, verifying the fundamental advantages of the proposed method. Subsequently, we selected the two learning-based baselines to conduct further generalization and scalability experiments on the trained models.

To clarify the experimental scope, we defined generalization as the model’s adaptability to unseen maps. Although the model was trained only on benchmark maze maps, it was evaluated on unseen maps including mazes and other categories. The probability distributions of instance configuration parameters in the test set, such as task positions, deadlines, and robot attributes, remain consistent with the training set, and the scenario scale (number of tasks and robots) remains constant. Scalability was defined as the model’s ability to handle problems of different scales. Since the model was trained with a fixed scale, we generated evaluation cases of varying scales by changing the number of tasks (increasing or decreasing) while maintaining a constant robot-to-task ratio.

Regarding evaluation metrics, based on the problem definition in Section 2.1, this study adopted the task success rate as the core metric for allocation effectiveness. We visualized the distribution of task success rates across 100 cases in the test set using box plots. This study employed the median to quantify overall performance and the interquartile range (IQR) to characterize stability. Performance differences were validated via paired t -tests ($\alpha = 0.05$).

In-Distribution Testing: We compared the E2AN algorithm with five baseline algorithms on a test set that is equally distributed as the training set. As shown in Figure 4, the median task success rate of E2AN was significantly higher than that of the baseline algorithm. Meanwhile, the IQR of this algorithm was the narrowest, demonstrating the best stability.

After analyzing the performance of each algorithm, it was found that all methods except RD could achieve a relatively high task success rate, which confirms their ability to solve MRTA problems. Constrained by the computational complexity of the problem and the preset runtime, the CLIP method could not converge to the global optimal solution in all test scenarios. Overall, the ranking of each algorithm is as follows: E2AN > CAM > CapAM > BiG $\approx$ CLIP > RD. The results of the paired sample t -test further statistically verified the performance advantages of E2AN.

Generalization Tests: To assess the model’s adaptability in unseen environments, we designed the generalization tests using six map types with completely different structures

(corresponding to Figure 3b–g), which include new mazes, rooms, cities and random obstacles. The experimental results in Figure 5 indicate that, in all test scenarios, E2AN maintains a performance advantage over the baseline algorithms. Notably, the BiG-MRTA method exhibits a significant performance gap compared to other approaches. Further analysis reveals that performance differentiation becomes subtler in simpler maps due to saturation effects. In simple scenarios (such as simple rooms and random obstacles), although the median of E2AN and CAM both reached 100%, the lower quartile (Q1) and minimum value of E2AN are better. This demonstrates that while most methods can handle general cases in simple environments, E2AN exhibits superior stability in handling edge cases.

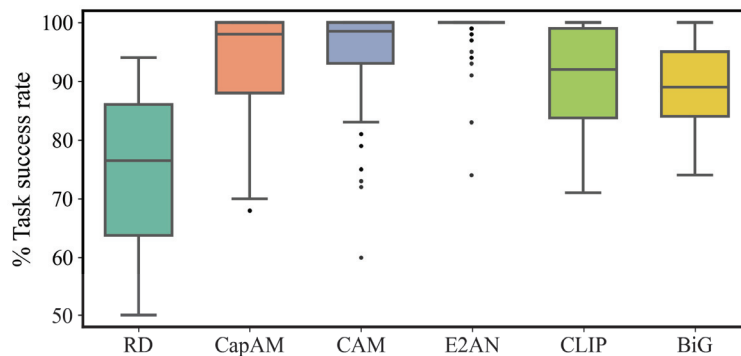


Figure 4. Boxplot of the task success rate for in-distribution testing in basic MRTA scenarios. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

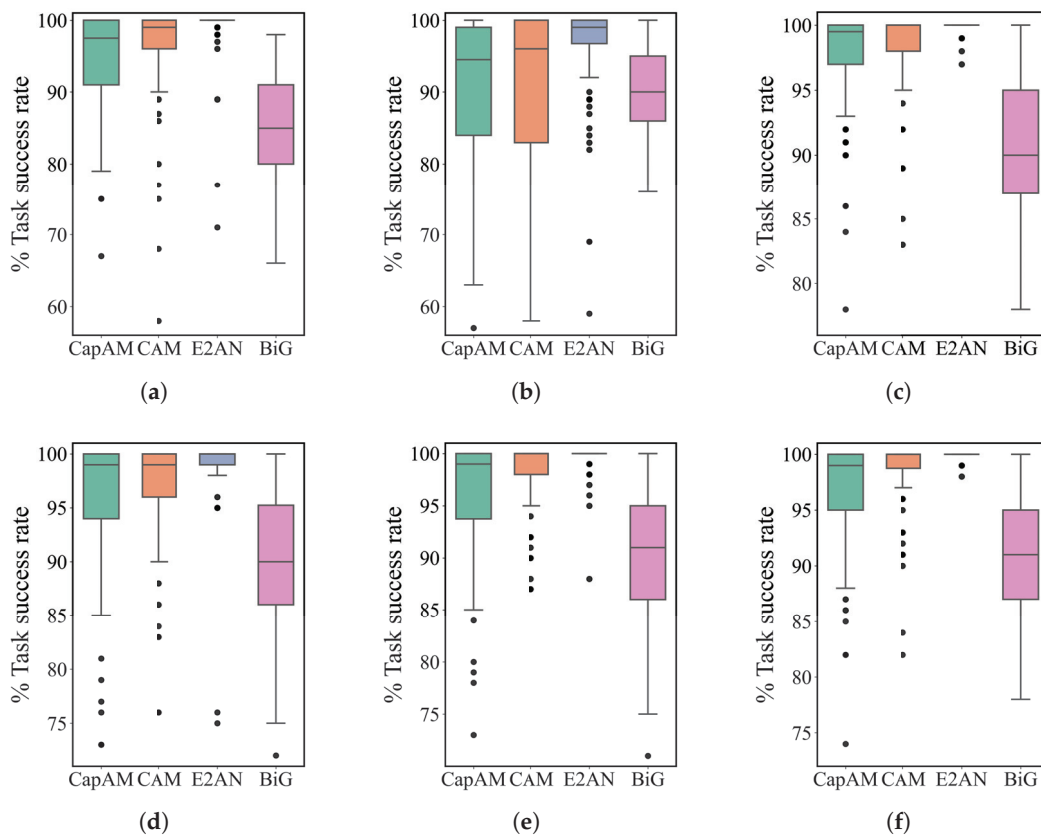


Figure 5. Boxplot of the task success rate for generalization tests in basic MRTA scenarios. (a) Scenarios in normal maze. (b) Scenarios in complex maze. (c) Scenarios in simple room. (d) Scenarios in complex room. (e) Scenarios in city. (f) Scenarios in random obstacles. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

Scalability Tests: We varied the problem scales by adjusting the number of tasks and robots, with configurations set as $(n_T, n_R) = (50, 4), (150, 10), (200, 14)$. As presented in Figure 6, E2AN is in a leading position at every scale. The comparison of experimental results in different scenarios shows that the task success rates of all algorithms increase as the problem scale expands. The underlying reason lies in the fact that when the map size is fixed, increasing the number of robots will enhance the distribution density of service nodes, thereby reducing the difficulty of achieving full coverage of tasks.

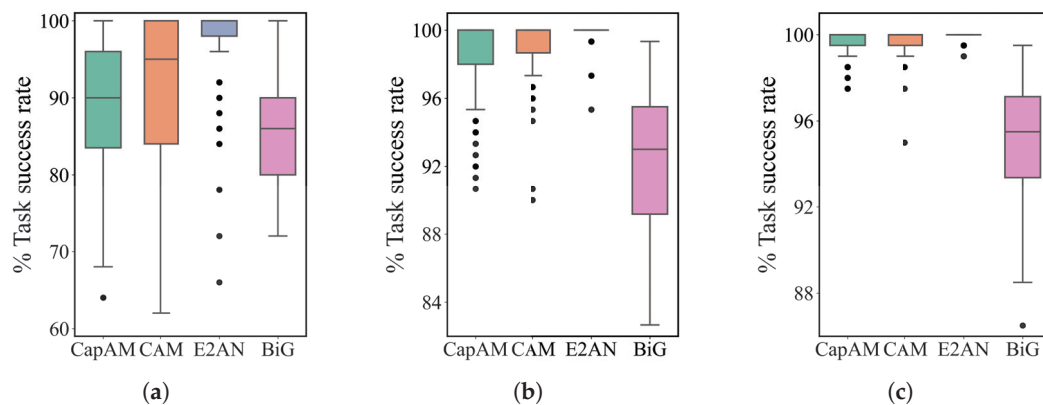


Figure 6. Boxplot of the task success rate for scalability tests in basic MRTA scenarios. (a) Scenarios with 50 tasks and 4 robots. (b) Scenarios with 150 tasks and 10 robots. (c) Scenarios with 200 tasks and 14 robots. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

4.4. MRTA with Diverse Payloads

To test the algorithm under complex constraints, we extended the experiments to scenarios with diverse payload requirements. Following the definitions in Section 2.1 exactly, the payload dimension was set to $K = 4$. The scenario was configured with 15 robots partitioned into 4 types, where the quantity of each type was randomly distributed in the interval $[2, 6]$. The 100 tasks were categorized based on supply-demand scarcity into general demand (25 tasks across 2 types) and specific demand (75 tasks across 4 types). Except for the payload attributes, the generation logic for other spatio-temporal parameters remains consistent with the basic scenario. For training, we selected the wild scenario shown in Figure 3h, as it aligns better with diverse payload requirements. Since the introduction of multi-dimensional payload vectors changes the node feature dimensions, the input layer of the encoder was adapted. The model was then retrained with all other hyperparameters kept unchanged to ensure a fair comparison.

Due to the computational complexity constraints of CLIP in solving high-dimensional problems, only RD, CapAM, and CAM were selected as baselines in this section. As illustrated in Figure 7, a comprehensive comparison shows that E2AN maintains a leading position in key metrics, including median task success rate and distribution variance. The results of the paired sample t -test further confirm that this performance difference is statistically significant.

We selected two unseen map environments (Figure 3a,b) for generalization tests, and adjusted the problem scales (number of tasks and robots) proportionally to $(50, 8)$ and $(200, 30)$ for scalability tests. The results in Figure 8, demonstrate that E2AN consistently achieves higher task success rates and narrower variance across all settings.

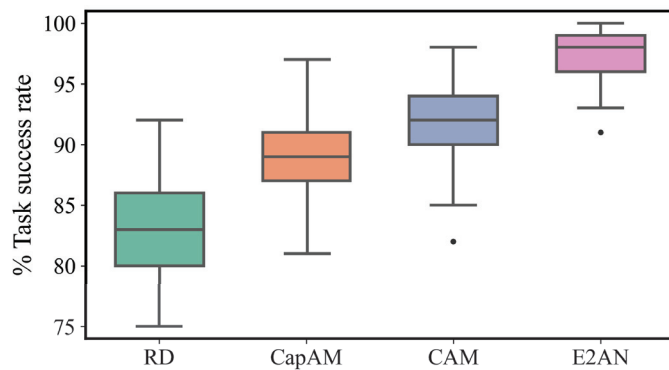


Figure 7. Boxplot of the task success rate for in-distribution testing in diverse payloads MRTA scenarios. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

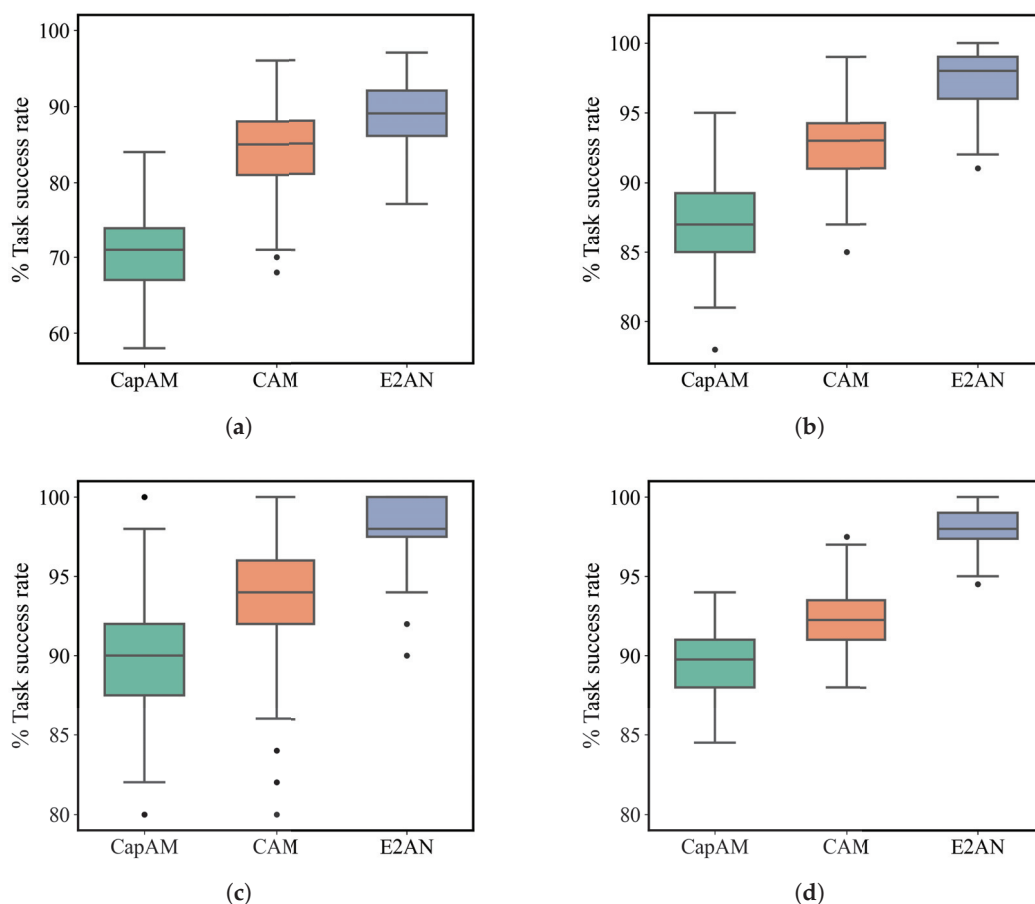


Figure 8. Boxplot of the task success rate for generalization tests and scalability tests in diverse payloads MRTA scenarios. (a) Scenarios in benchmark maze. (b) Scenarios in complex room. (c) Scenarios with 50 tasks and 8 robots. (d) Scenarios with 200 tasks and 30 robots. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

4.5. Ablation Studies

To evaluate the E2HGAT encoder and attention decoder, we designed ablation studies by replacing these components with Feed-Forward Networks (FFN) to construct two ablation variants.

Encoder Ablation (FFE): For both robot and task nodes, independent MLP channels (containing two linear layers) were employed to map raw attributes to the embedding dimension, without performing any message passing between nodes.

Decoder Ablation (FFD): The Multi-Head Attention-based decoder was replaced with a feed-forward decoder. As shown in Equation (27), this module retains the original query vector construction logic. It concatenates the query with the embedding of each task, then reduces the feature dimension from $2D$ to 1 via a linear layer. Finally, it generates the unnormalized score (logit) for the task using a LeakyReLU activation. The subsequent masking strategy and decision-making logic remain strictly consistent with Section 3.2.

$$\text{logit}_i^{\text{task}} = \text{LeakyReLU}\left(\text{FFN}\left(\text{Concat}\left(H_i^{\text{task}}, Q\right)\right)\right), \quad \text{where } i \in V_T. \quad (27)$$

We comprehensively evaluated the performance of E2AN and its ablation models under two scenario settings: basic MRTA and diverse payload MRTA. The evaluation covers in-distribution tests, generalization tests, and scalability tests. The experimental results are presented in Figure 9 (basic scenarios) and Figure 10 (diverse payload scenarios).

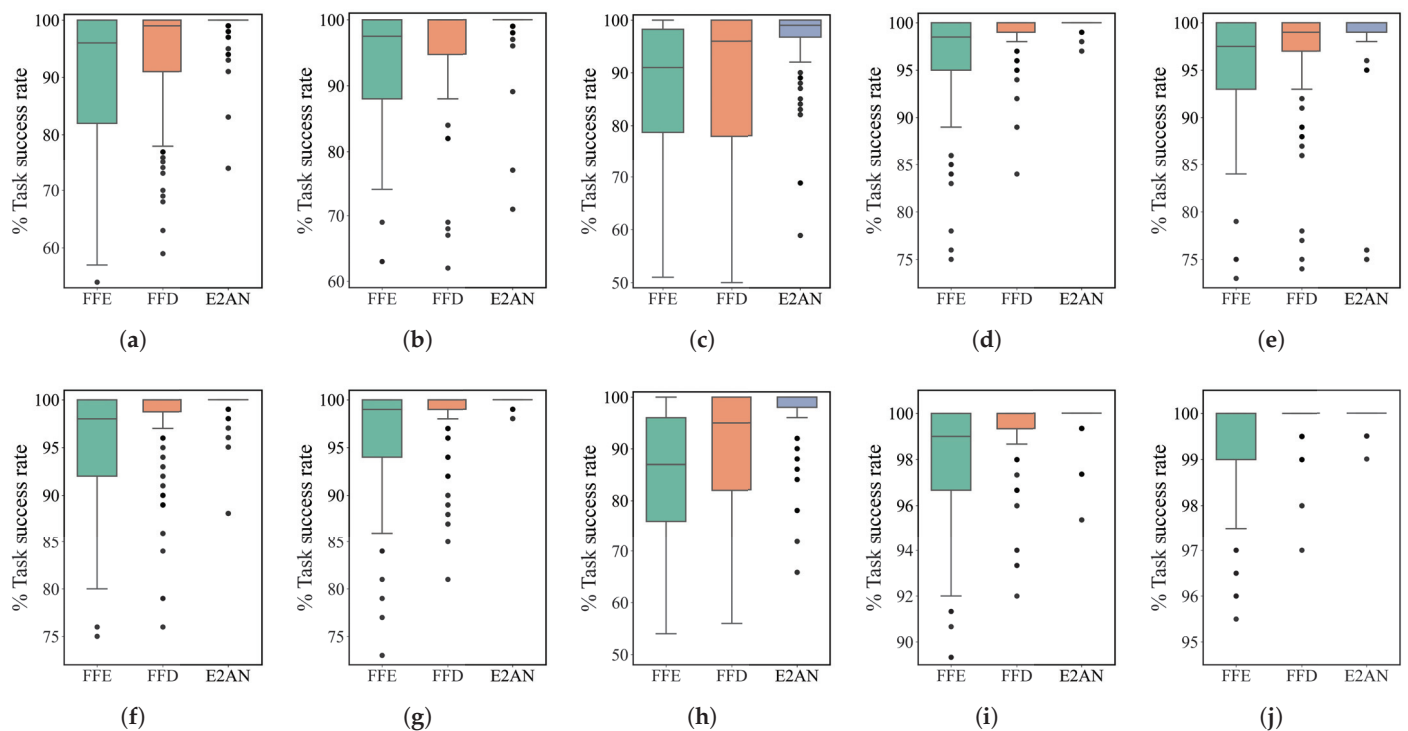


Figure 9. Boxplot of the task success rate for ablation tests in basic MRTA scenarios. (a) Scenarios like training. (b) Scenarios in normal maze. (c) Scenarios in complex maze. (d) Scenarios in simple room. (e) Scenarios in complex room. (f) Scenarios in city. (g) Scenarios in random obstacles. (h) Scenarios with 50 tasks and 4 robots. (i) Scenarios with 150 tasks and 10 robots. (j) Scenarios with 200 tasks and 14 robots. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

In basic scenarios, E2AN achieves optimal results across all evaluation metrics. Although the gap in median success rates narrows in some low-difficulty cases, E2AN still demonstrates robustness in handling the worst cases. Comparing the two ablation variants, FFD outperformed FFE. This observation suggests that the graph encoding mechanism contributes more to model performance than the attention decoding mechanism.

With the increased complexity of diverse payloads, the performance gap widens significantly, highlighting the limitations of the single module. This confirms that the complete framework is essential for capturing intricate matching relationships.

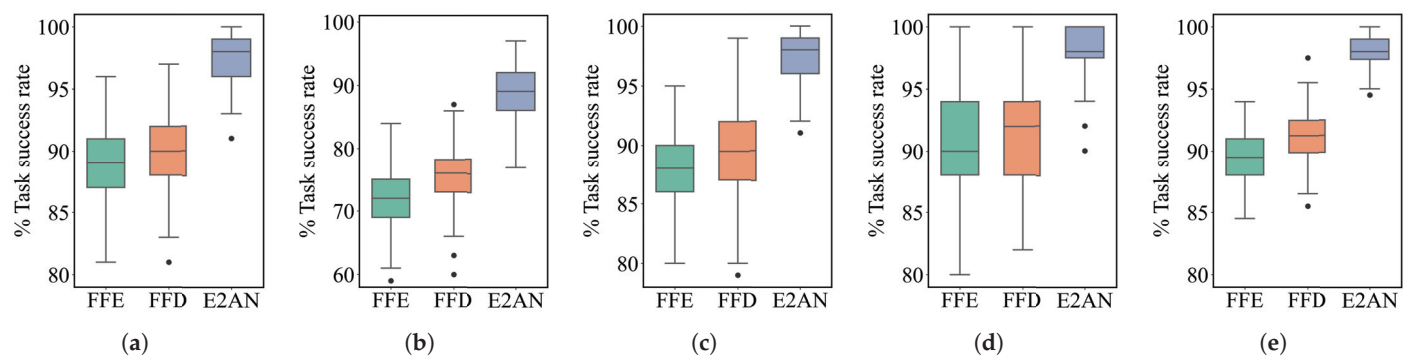


Figure 10. Boxplot of the task success rate for ablation tests in diverse payloads MRTA scenarios. (a) Scenarios like training. (b) Scenarios in benchmark maze. (c) Scenarios in complex room. (d) Scenarios with 50 tasks and 8 robots. (e) Scenarios with 200 tasks and 30 robots. Different colors represent different algorithms as labeled on the x-axis, and the dots denote outliers.

4.6. Analysis of Computational Efficiency and Scalability

Tables 2 and 3 provide a detailed analysis of inference latency under varying problem scales and scenario types. The results from the six distinct data groups indicate that although inference latency trends upward with problem scale, this growth is notably non-linear. For instance, quadrupling the number of tasks from 50 to 200 results in a marginal increase of only approximately 2 ms, confirming that computational complexity does not escalate linearly with input size. Moreover, the transition from basic to diverse payload scenarios incurs minimal computational overhead, demonstrating that E2AN maintained stable performance regardless of problem size or constraint complexity.

Optimization-based methods, such as CLIP, operate as global planners that must solve the entire problem instance before generating a final executable plan. This batch-processing paradigm prevents them from making immediate decisions during execution. In contrast, E2AN employs an asynchronous sequential decision-making framework, modeling task allocation as an event-driven online process where decisions are triggered instantaneously only when a robot becomes idle. Given this operational paradigm, inference latency, which measures the computation time required for a single decision step, was adopted as the primary metric for evaluating real-time capability in dynamic environments.

We evaluated real-time performance by calculating the average inference latency per decision step across the test sets for each scenario configuration. The performance metrics reported in this section were obtained using the workstation configuration detailed in Section 4.1. The results are summarized in Tables 2 and 3.

Table 2. Comparison of average inference latency across different algorithms and ablation variants.

Method	CapAm	CAM	E2AN	FFE	FFD
Inference Latency	2.43 ms	2.23 ms	11.09 ms	0.74 ms	10.64 ms

Table 2 compares the inference latency of E2AN against baselines and ablation variants. Although E2AN (11.09 ms) entails a higher computational cost than simpler homogeneous methods like CapAM and CAM, or the ablation variant FFE, this difference is primarily driven by the E2HGAT encoder, as evidenced by the performance gap between FFE and FFD. However, this latency remains at the millisecond level, which is negligible compared to robot travel time. This indicates that E2AN effectively exploits potential idle time during system operation. Thus, trading a slight increase in computation for significant gains in decision quality is a justifiable strategy.

Table 3. Comprehensive analysis of computational efficiency under varying scenario types and problem scales.

Scenario Configuration	Scale (n_R/n_T)	Latency (ms)	Peak VRAM (MB)
Basic Scenario	4/50	10.57	1105
	7/100	11.09	1141
	14/200	12.34	1249
Diverse Payload	8/50	11.00	1349
	15/100	11.74	1367
	30/200	12.76	1503
Large Scale Extension	35/500	27.51	2079
	53/750	49.95	3255
	70/1000	85.12	4985
	100/1000	119.90	6551

To further investigate the hardware feasibility under extreme conditions, we extended the experimental scope beyond the generalization tests in Section 4.3. While previous results established that the success rate remains stable given the consistent map density, this section focuses on the trends in computational cost as the problem scale of the basic scenario expands up to 100 robots and 1000 tasks. The comprehensive results covering basic scenarios (at standard and large scales) and diverse payload scenarios are summarized in Table 3.

The results indicate that the system exhibits robust scalability with a predictable cost growth pattern. By comparing the Basic and Diverse Payload scenarios, it is evident that increasing constraint complexity incurs minimal computational overhead regarding latency, although it introduces a slight constant increase in memory usage due to the additional feature dimensions. Regarding memory consumption, the data reveals a quadratic growth trend governed by the $O(n_R \times n_T)$ space complexity of the interaction matrix in the graph attention mechanism. It is important to note a fixed base overhead of approximately 1100 MB required for the framework context and model weights; this base cost dominates at smaller scales but becomes less significant as the dynamic portion increases with problem size. Despite this growth, the peak memory usage in the largest tested scenario (100 robots, 1000 tasks) reaches 6.55 GB. Since the algorithm is designed to be deployed on a centralized workstation acting as a global scheduler, this consumption is well within the capacity of standard high-performance GPUs specified in Section 4.1.

Furthermore, while the inference latency rises to approximately 120 ms at the maximum scale, this delay remains operationally viable. Given the asynchronous, event-triggered nature of the proposed framework, decisions are required only at discrete moments when a robot completes a task. Consequently, this millisecond-level computation time is negligible compared to the physical execution time of tasks ranging from seconds to minutes, ensuring seamless operational continuity even in large-scale deployments.

5. Discussion

This study focused on the MRTA problem with coupled constraints of obstructed maps and deadlines while further handling diverse payload requirements. To address this problem, we formulated the task allocation as a Markov Decision Process over a heterogeneous graph and proposed the E2AN method. Tests indicate that E2AN outperforms baselines across basic and diverse-payload scenarios and maintains robustness even as problem complexity increases.

5.1. Mechanism Analysis

Further mechanism analysis reveals how the design of the framework contributes to its performance. The generalization ability of the model in unseen map scenes is related to the input form corresponding to the spatial constraints. This method does not use the raw map image but takes precise path costs as edge attributes to avoid overfitting to the topological structure of a specific map. Heterogeneous graph structures and asynchronous sequential decision-making frameworks are key factors for the scalability of the system under different problem scales. Among them, the heterogeneous graph structure allows nodes to flexibly achieve quantity changes by altering some connections while the asynchronous decision-making mechanism enables the decoder to dynamically adjust the query vector according to the real-time system state. The adaptability of E2AN to diverse payload requirements stems from its heterogeneous-graph encoder. Distinct from homogeneous methods relying on decoders, E2AN utilizes a heterogeneous graph encoder to capture robot–task dependencies and produces more discriminative features for downstream decisions.

5.2. Practical Deployment Framework

To clarify the applicability of the proposed algorithm in physical environments, we delineate its operational positioning within a hierarchical control architecture. In this framework, E2AN functions as a centralized global scheduler deployed on a high-performance workstation. Crucially, this high-level decision-making module is structurally decoupled from the low-level motion control and execution of individual robots. The specific operational workflow is detailed as follows:

- **Input Stream:** The system receives real-time state updates from the robot fleet via a communication network. These updates include robot poses obtained from localization algorithms, current operational status, and dynamic payload attributes. This mechanism allows the system to support real-time updates for scenarios involving payload switching or payload degradation. Simultaneously, the environmental map is maintained globally. If onboard sensors detect new static obstacles, the global map is updated and the edge weights in the heterogeneous graph are recalculated using global path planning algorithms based on the latest map snapshot.
- **Decision Process:** The E2AN policy is triggered asynchronously. Whenever a robot completes a task or a new agent joins the network, the scheduler constructs the current state graph S_t and executes inference. As analyzed in Section 4.6, this inference occurs on the workstation with millisecond-level latency to ensure immediate responsiveness.
- **Output Stream:** The scheduler outputs a target assignment ID to the specific robot. The onboard local planner of the robot then executes the specific navigation. This layer handles sensor noise, localization correction, and immediate avoidance of dynamic obstacles such as pedestrians during execution.

This hierarchical separation ensures robustness. Although the global scheduler assumes a deterministic transition model, the asynchronous event-triggered mechanism prevents error accumulation. Even if sensor noise or dynamic obstacles cause execution delays at the robot level, these deviations simply delay the next decision trigger without invalidating the global logic because the scheduler always acts on the confirmed real-time state.

5.3. Limitations and Future Directions

Despite the promising simulation results and the feasible deployment framework, this study has several limitations that define the scope for future research:

- **Dependence on Global Environmental Models:** The framework relies on a consistent global map and stable communication to compute precise path costs. Consequently, its adaptability may be compromised in partially observable scenarios such as autonomous exploration or in highly dynamic environments where topological structures fluctuate rapidly. Future research could integrate predictive traffic models into the edge attribute construction to enhance robustness against environmental uncertainty.
- **Computational Scalability Limits:** The interaction mechanism within the graph attention network entails a memory complexity that grows quadratically with the system scale. This characteristic imposes an upper bound on the maximum supportable problem size and necessitates the deployment on centralized high-performance workstations. While suitable for standard logistics scenarios, this computational cost restricts infinite scalability for massive-scale swarms.
- **Complex Collaborative Constraints:** While the current model supports diverse payload requirements via node attributes, it assumes a one-to-one mapping between robots and payloads. Complex capacity trade-offs or multi-robot coalition tasks are not explicitly modeled. Future research could extend the action space and graph schema to address these combinatorial optimization challenges.
- **Real-World Validation:** The absence of physical experiments limits the evaluation of the model regarding resilience to non-Gaussian noise and communication instability found in the real world. Future studies are encouraged to deploy E2AN on a physical multi-robot testbed to validate the proposed architecture and explore Sim-to-Real transfer techniques to bridge the gap between simulation training and physical deployment.

6. Conclusions

This study proposes E2AN to address the MRTA problem characterized by temporal and environmental spatial constraints, with the flexibility to accommodate diverse payload requirements. By constructing a heterogeneous graph MDP model that supports asynchronous sequential decision-making, we transform the high-dimensional combinatorial optimization problem into a learnable sequential decision process. Experimental results show that the proposed method consistently outperforms existing baselines across various map topologies and problem scales. In particular, ablation and extended experiments verify that the core E2HGAT encoder mitigates spatial information distortion in complex environments by incorporating spatial edge attributes within a heterogeneous graph framework. Moreover, the analysis of inference latency demonstrates that the method maintains computational efficiency, validating its feasibility for online deployment.

However, specific costs and limitations exist. The graph attention mechanism entails a quadratic growth in memory consumption relative to the system scale, which imposes a theoretical limit on infinite scalability and necessitates high-performance centralized hardware. Additionally, the reliance on global map availability may constrain robustness in partially observable or highly dynamic settings.

This study provides an effective solution method with strong generalization, scalability, and real-time capability for MRTA in complex dynamic environments.

Author Contributions: Conceptualization, Y.H. and D.L.; Data curation, Y.H.; Formal analysis, Y.H. and D.L.; Funding acquisition, D.L.; Investigation, Y.H. and J.L. (Jinhong Li); Methodology, Y.H.; Project administration, D.L.; Resources, Y.H., J.L. (Junxiang Li) and T.W.; Software, Y.H.; Supervision, D.L.; Validation, Y.H. and J.L. (Jinhong Li); Writing—original draft, Y.H.; Writing—review & editing, D.L., J.L. (Junxiang Li) and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. Part of the data are not publicly available due to our laboratory's confidentiality agreement and policies.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MRTA	Multi-Robot Task Allocation
ST-SR-TA	single-task robots, single-robot tasks, and time-extended assignments
ST-MR-TA	single-task robots, multi-robot tasks, and time-extended assignments
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem
RGAT	Relational Graph Attention Network
E2AN	Edge-Enhanced Attention Network
E2HGAT	Edge-Enhanced Heterogeneous Graph Attention Network
MDP	Markov Decision Process
MILP	Mixed Integer Linear Programming
MHA	Multi-Head Attention
IQR	interquartile range
FFN	Feed-Forward Network

Appendix A. Sensitivity Analysis of Encoder Layers

This appendix provides a detailed sensitivity analysis of the number of convolutional layers (L) in the E2HGAT encoder. This hyperparameter is critical as it determines the receptive field of the Graph Neural Network, influencing both the model's ability to capture topological dependencies and its computational efficiency. We evaluated models configured with $L = 1$, $L = 2$, and $L = 3$ on the standard test set.

The impact of layer depth on task success rate is illustrated in Figure A1. The model configured with a single layer ($L = 1$) consistently underperforms compared to deeper architectures, confirming that sufficient depth is essential for robust generalization. This limitation is attributed to the restricted receptive field, which prevents the effective aggregation of information from indirect neighbors (e.g., resolving conflicting relationships between multiple robots targeting the same task). Increasing the depth to $L = 2$ yields a substantial improvement in success rates, indicating that a two-hop propagation is sufficient to capture critical local topological constraints. However, further increasing the depth to $L = 3$ results in negligible performance gains, suggesting a performance saturation where additional layers offer no further benefit and may potentially introduce over-smoothing risks.

In addition to performance, we analyzed the computational cost, as shown in Table A1. The data indicates that peak GPU memory consumption remains consistent across all configurations (≈ 1140 MB), demonstrating that increasing the number of encoder layers has a negligible impact on memory overhead. However, the inference latency increases linearly with the number of layers. The 3-layer model increases the inference time by approximately 35% compared to the 2-layer model (15.00 ms vs. 11.09 ms) without providing a performance gain. Therefore, to balance decision accuracy with real-time responsiveness, we determined $L = 2$ to be the optimal parameter for the E2HGAT encoder.

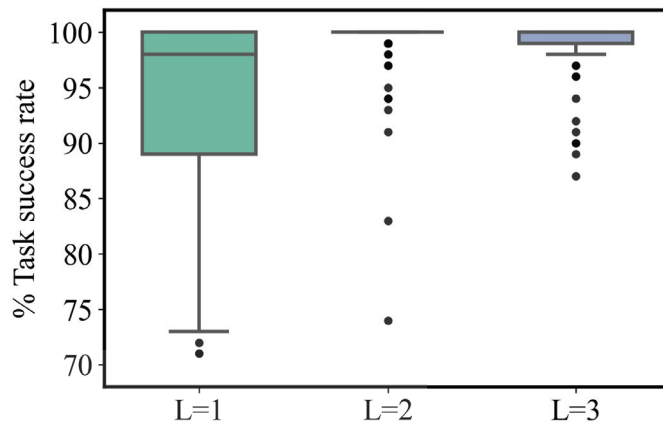


Figure A1. Boxplot of task success rates on the standard test set across different encoder layer counts ($L = 1, 2, 3$). The median success rate saturates at $L = 2$, while $L = 1$ shows significantly lower performance and higher variance. Different colors represent algorithms with different layer counts as labeled on the x-axis, and the dots denote outliers.

Table A1. Computational efficiency analysis with varying encoder layers.

Encoder Layers (L)	Inference Latency (ms)	Peak GPU Memory (MB)
1	6.54	1140
2	11.09	1141
3	15.00	1143

References

- Gerkey, B.P.; Matarić, M.J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [CrossRef]
- Kim, J.; Son, H.I. A Voronoi Diagram-Based Workspace Partition for Weak Cooperation of Multi-Robot System in Orchard. *IEEE Access* **2020**, *8*, 20676–20686. [CrossRef]
- Li, K.; Liu, T.; Ram Kumar, P.N.; Han, X. A Reinforcement Learning-Based Hyper-Heuristic for AGV Task Assignment and Route Planning in Parts-to-Picker Warehouses. *Transp. Res. E Logist. Transp. Rev.* **2024**, *185*, 103518. [CrossRef]
- Gao, S.; Wu, J.; Ai, J. Multi-UAV Reconnaissance Task Allocation for Heterogeneous Targets Using Grouping Ant Colony Optimization Algorithm. *Soft Comput.* **2021**, *25*, 7155–7167. [CrossRef]
- Tian, X.; Xu, T.; Luo, X.; Jia, Y.; Yin, J. Multi-UAV Reconnaissance Task Allocation in 3D Urban Environments. *IEEE Access* **2024**, *12*, 30989–30999. [CrossRef]
- Wang, R.; Zhao, D.; Min, B.-C. Initial Task Allocation for Multi-Human Multi-Robot Teams with Attention-Based Deep Reinforcement Learning. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 7915–7922.
- Wang, R.; Zhao, D.; Gupte, A.; Min, B.-C. Initial Task Allocation in Multi-Human Multi-Robot Teams: An Attention-Enhanced Hierarchical Reinforcement Learning Approach. *IEEE Robot. Autom. Lett.* **2024**, *9*, 3451–3458. [CrossRef]
- Zhou, C.; Li, J.; Shi, M.; Wu, T. Multi-Robot Path Planning Algorithm for Collaborative Mapping under Communication Constraints. *Drones* **2024**, *8*, 493. [CrossRef]
- Nunes, E.; Manner, M.; Mitiche, H.; Gini, M. A Taxonomy for Task Allocation Problems with Temporal and Ordering Constraints. *Robot. Auton. Syst.* **2017**, *90*, 55–70. [CrossRef]
- Ghassemi, P.; Chowdhury, S. Multi-Robot Task Allocation in Disaster Response: Addressing Dynamic Tasks with Deadlines and Robots with Range and Payload Constraints. *Robot. Auton. Syst.* **2022**, *147*, 103905. [CrossRef]
- Wang, H.; Li, Y.; Li, S.; Xu, G. Research on Multiple AUVs Task Allocation with Energy Constraints in Underwater Search Environment. *Electronics* **2024**, *13*, 3852. [CrossRef]
- Martin, J.G.; Muros, F.J.; Maestre, J.M.; Camacho, E.F. Multi-Robot Task Allocation Clustering Based on Game Theory. *Robot. Auton. Syst.* **2023**, *161*, 104314. [CrossRef]
- Hu, J.; Bhowmick, P.; Jang, I.; Arvin, F.; Lanzon, A. A Decentralized Cluster Formation Containment Framework for Multi-robot Systems. *IEEE Trans. Robot.* **2021**, *37*, 1936–1955. [CrossRef]

14. Cui, W.; Li, R.; Feng, Y.; Yang, Y. Distributed Task Allocation for a Multi-UAV System with Time Window Constraints. *Drones* **2022**, *6*, 226. [CrossRef]
15. David, J.; Valencia, R. Simultaneous Path Planning and Task Allocation in Dynamic Environments. *Robotics* **2025**, *14*, 17. [CrossRef]
16. Hu, Y.; Wu, X.; Zhai, J.; Lou, P.; Qian, X.; Xiao, H. Hybrid Task Allocation of an AGV System for Task Groups of an Assembly Line. *Appl. Sci.* **2022**, *12*, 10956. [CrossRef]
17. Liu, F.; Xu, W.; Feng, Z.; Yu, C.; Liang, X.; Su, Q.; Gao, J. Task Allocation and Path Planning Method for Unmanned Underwater Vehicles. *Drones* **2025**, *9*, 411. [CrossRef]
18. Guo, H.; Miao, Z.; Ji, J.; Pan, Q. An Effective Collaboration Evolutionary Algorithm for Multi-Robot Task Allocation and Scheduling in a Smart Farm. *Knowl.-Based Syst.* **2024**, *289*, 111474. [CrossRef]
19. Hwang, N.E.; Kim, H.J.; Kim, J.G. Centralized Task Allocation and Alignment Based on Constraint Table and Alignment Rules. *Appl. Sci.* **2022**, *12*, 6780. [CrossRef]
20. Lin, J.; Li, P.; Dai, J.; Liu, S.; Tian, W.; Li, B.; Wang, C. Multi-Robot Cooperative Assembly Task Planning for Large-Scale Aerospace Structures. In Proceedings of the 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Beijing, China, 19–20 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 795–800.
21. Kool, W.; van Hoof, H.; Welling, M. Attention, Learn to Solve Routing Problems! In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019.
22. Wang, Z.; Gombolay, M. Learning Scheduling Policies for Multi-Robot Coordination with Graph Attention Networks. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4509–4516. [CrossRef]
23. Paul, S.; Ghassemi, P.; Chowdhury, S. Learning Scalable Policies over Graphs for Multi-Robot Task Allocation Using Capsule Attention Networks. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8815–8822.
24. Paul, S.; Chowdhury, S. Learning to Allocate Time-Bound and Dynamic Tasks to Multiple Robots Using Covariant Attention Neural Networks. *J. Comput. Inf. Sci. Eng.* **2024**, *24*, 091005. [CrossRef]
25. Wang, Z.; Liu, C.; Gombolay, M. Heterogeneous Graph Attention Networks for Scalable Multi-Robot Scheduling with Temporal Constraints. *Auton. Robot.* **2022**, *46*, 249–268. [CrossRef]
26. Wang, Z.; Chen, J.; Chen, H. EGAT: Edge-Featured Graph Attention Network. In Proceedings of the 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, 14–17 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 253–264.
27. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous Graph Transformer. In Proceedings of the Web Conference 2020 (WWW '20), Taipei, Taiwan, 20–24 April 2020; ACM: New York, NY, USA, 2020; pp. 2704–2710.
28. Gao, W.; Yu, Z.; Wang, T.; Wang, L.; Cui, H.; Guo, B.; Xiong, H. GNN-Based Deep Reinforcement Learning for Computation Task Scheduling in Autonomous Multi-Robot Systems. *J. Syst. Archit.* **2025**, *168*, 103534. [CrossRef]
29. Ma, Z.; Gong, H.; Xiong, J.; Wang, X. Heterogeneous Multiagent Task Allocation Based on Graph-Based Convolutional Assignment Neural Network. *IEEE Internet Things J.* **2025**, *12*, 17281–17299. [CrossRef]
30. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural Message Passing for Quantum Chemistry. In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia, 6–11 August 2017; PMLR: Cambridge, MA, USA, 2017; pp. 1263–1272.
31. Park, B.; Kang, C.; Choi, J. Cooperative Multi-Robot Task Allocation with Reinforcement Learning. *Appl. Sci.* **2021**, *12*, 272. [CrossRef]
32. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3229–3238.
33. Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T.K.; et al. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. *Proc. Int. Symp. Comb. Search* **2021**, *10*, 151–158. [CrossRef]
34. Mitiche, H.; Boughaci, D.; Gini, M. Iterated Local Search for Time-Extended Multi-Robot Task Allocation with Spatio-Temporal and Capacity Constraints. *J. Intell. Syst.* **2019**, *28*, 347–360. [CrossRef]
35. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi: Beaverton, OR, USA, 2024.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Deep Reinforcement Learning-Based Water Jet Control for Robotic Manipulators Using an Improved Experience Replay Mechanism

Rong Zhang ^{1,2}, Jianjun Qin ^{1,2,3}, Luyang Wang ^{1,2,3,*} and Guotong Li ^{1,2,3}

¹ School of Mechanical-Electronic and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China; 2108020023033@stu.bucea.edu.cn (R.Z.); qinjianjun@bucea.edu.cn (J.Q.); liguotong@bucea.edu.cn (G.L.)

² Embodied Intelligence and Robotics Technology Research Center of the Artificial Intelligence + Research Institute, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

³ Beijing Key Laboratory of Intelligent Construction Equipment for Urban Building Construction, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

* Correspondence: wangluyang@bucea.edu.cn

Abstract

Existing robotic water jet control methods are limited by fixed spray configurations and low adaptability to complex or dynamic environments. These constraints hinder precise targeting in three-dimensional spaces. To overcome this, we propose a reinforcement learning-based water jet control framework that achieves accurate targeting without pose or angle restrictions. Specifically, we introduce Goal-Priority Hindsight Experience Replay (GPHER), a replay strategy that integrates the principles of Hindsight Experience Replay (HER), Prioritized Experience Replay (PER), and curriculum learning. GPHER dynamically adjusts sampling priorities based on goal-space distance, guiding training from simple to complex goals. Combined with Truncated Quantile Critics (TQCs), this approach accelerates convergence and enhances success rates. Both simulation and real-world experiments validate the robustness and adaptability of the proposed method, demonstrating its effectiveness for real-time robotic fluid control.

Keywords: robotic arm control; water-jet control; experience replay

1. Introduction

Intelligent water-spraying robots have attracted increasing attention due to their potential applications in real-world scenarios such as firefighting, agricultural irrigation, and building façade cleaning. In firefighting tasks, robots are required to accurately direct water jets toward flames in complex and cluttered environments while maintaining a safe distance from hazardous areas [1]. In agricultural irrigation, crops are typically distributed irregularly in three-dimensional space, requiring precise control of water jet positions to ensure efficient watering while minimizing water waste [2]. Similarly, in building cleaning and industrial washing tasks, water jets must be directed toward specific surface locations at varying heights and orientations [3]. Achieving accurate water jet targeting in these scenarios is challenging because the landing position of the jet depends on multiple factors, including the nozzle position, orientation, jet velocity, and environmental disturbances such as air resistance. These factors introduce complex nonlinear dynamics that are difficult to model precisely using traditional analytical approaches.

Furthermore, studying robotic fluid manipulation is also important from the perspective of embodied intelligence. As robots are expected to operate in increasingly diverse environments, they must interact with objects with different physical properties, such as rigid bodies, deformable objects, and fluids. Compared with rigid object manipulation, fluid interaction introduces additional challenges due to its complex and dynamic behavior. Therefore, investigating robotic water jet control not only addresses specific application needs but also contributes to improving the general manipulation capabilities of robots in complex environments.

Water jet control therefore represents a complex robotic manipulation problem. Existing robotic fluid manipulation studies have largely focused on liquid pouring tasks [4,5], where fluids mainly flow downward under gravity. However, practical applications such as plant watering, firefighting, and high-altitude cleaning involve more complex spatial target distributions. To simplify the problem, many existing methods restrict certain control variables. For example, some approaches fix the spray angle while adjusting only the nozzle position [6]. In firefighting robotics, this strategy forces the robot to rely on base mobility for coarse positioning, which severely restricts its ability to reach targets in vertically cluttered environments such as multi-story buildings. Similarly, other studies keep the nozzle position fixed and vary only the spray angle to control the landing point [7]. While effective for planar targets, this approach cannot accommodate irregular three-dimensional crop distributions in agricultural settings, where plants vary in both height and horizontal offset. These simplified strategies fundamentally limit the controllable workspace and make it difficult to reach arbitrary target positions in three-dimensional space. Moreover, such control strategies often require significant manual tuning.

The choice of a robotic manipulator for water jet control is motivated by its inherent advantages in dexterity and flexibility. Unlike fixed spraying systems or those with limited mobility, a robotic arm provides multiple degrees of freedom (DOF), enabling simultaneous and continuous adjustment of both the nozzle's position (X, Y, Z) and orientation (pitch, yaw). This capability is essential for directing a water jet to arbitrary points in three-dimensional space, a fundamental requirement in applications such as firefighting, where the robot must maintain a safe distance from hazardous areas while accurately targeting flames, or in agricultural irrigation, where crops are distributed irregularly across varying heights and angles. By leveraging the full pose control of a manipulator, the system can overcome the constrained workspace limitations of prior approaches [6,7] and achieve precise targeting under complex environmental conditions. Furthermore, studying robotic fluid manipulation with high-DOF arms contributes to the broader goal of embodied intelligence: as robots are expected to operate in increasingly diverse real-world scenarios, they must acquire the ability to interact with objects of varying physical properties, including rigid bodies, deformable materials, and fluids. Mastering water jet control thus serves not only a practical application need but also as a foundational step toward enhancing the generalization and adaptability of robotic systems in unstructured environments.

Reinforcement learning (RL) offers a novel research paradigm for water jet control tasks. Unlike traditional control methods that rely on accurate modeling, RL enables adaptive learning and optimization of control policies through interaction with simulation environments, without requiring explicit and complex fluid dynamic models. In contrast, conventional water control methods struggle with high-dimensional control problems due to their inherent limitations in control dimensionality. These methods often operate under constrained conditions, for instance, with fixed spray angles, which leads to limited adaptability and generalization capability. Despite the demonstrated flexibility and adaptability of RL in handling complex control tasks, existing replay mechanisms exhibit significant limitations in environments characterized by sparse rewards and complex goal spaces.

Hindsight Experience Replay (HER) mitigates the sparse reward problem by relabeling failed trajectories as successful experiences relative to achieved states, thereby improving learning efficiency. However, its uniform goal sampling neglects the varying importance of experiences for policy improvement, resulting in suboptimal sample utilization. In contrast, Prioritized Experience Replay (PER) assigns sampling weights based on Temporal Difference (TD) errors [8], enabling faster convergence but still suffering from inefficiency in sparse-reward scenarios due to the lack of valuable feedback samples. Integrating HER and PER has been explored to combine goal diversity with prioritized learning, yet the pseudo-success samples generated by HER distort the TD error distribution. Because these relabeled goals cluster in simpler regions of the task space, PER's prioritization mechanism is weakened, ultimately reducing training effectiveness.

To overcome these limitations, this paper proposes Goal-Priority Hindsight Experience Replay (GPHER), a joint sampling framework that incorporates both goal-space distance and TD error. Adaptive weighting is applied to balance their influence across different training stages. Integrated with the Truncated Quantile Critics (TQC) algorithm, GPHER is evaluated in a simulated water jet control environment with air resistance modeling. Experimental results show that GPHER significantly accelerates convergence and enhances policy stability, achieving higher success rates, especially for high-difficulty targets, thereby improving both learning efficiency and generalization in complex robotic fluid control tasks.

Figure 1 illustrates the overall framework of the proposed robotic water jet control system. The framework includes three main stages: simulation environment construction for modeling the robotic manipulator and fluid dynamics, reinforcement learning-based policy training using the TQC algorithm combined with the proposed GPHER mechanism, and finally sim-to-real deployment with physical validation on a robotic platform. This pipeline illustrates how the water jet control task is formulated, how the learning agent is trained in simulation, and how the learned policy is transferred to the real system for evaluation.

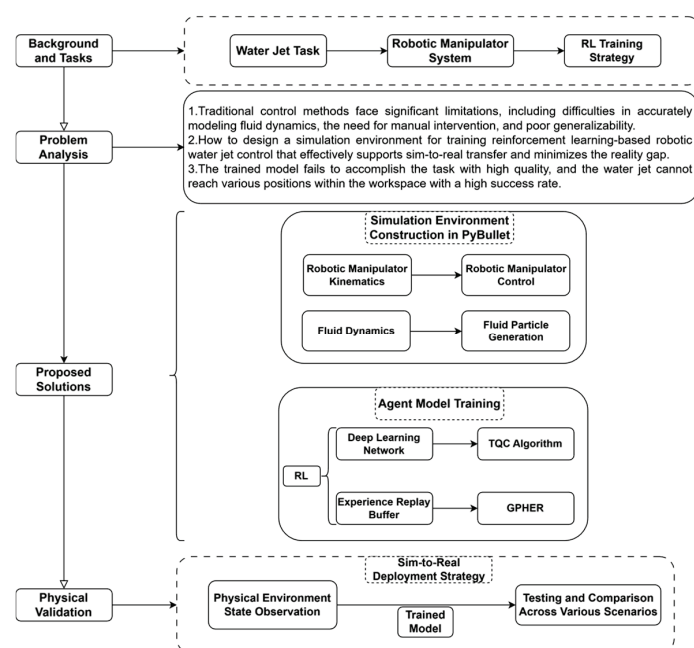


Figure 1. Overall pipeline of the proposed reinforcement learning framework for robotic water jet control, including simulation environment construction, agent training with TQC and GPHER, and sim-to-real deployment.

2. Related Work

2.1. Robotic Water Manipulation Tasks

Robotic water flow manipulation is increasingly important in fields such as agriculture, firefighting, and cleaning. Most existing research, however, focuses on liquid pouring or fixed-point spraying in structured scenarios. For instance, Vatavuk et al. (2022) proposed a vineyard spraying method based on Task Space Model Predictive Control, which performs well in regularly planted vineyards but cannot adapt to complex three-dimensional target spaces [9]. Wei (2024) developed a six-axis robotic arm with a tracked chassis for automated irrigation, using Cartesian resolution and constrained joint optimization to reduce water deviation [10]. Nonetheless, its path planning relies on static modeling and cannot dynamically respond to plant morphology or obstacles. High-dimensional modeling complexity has led most studies to adopt limited control approaches, adjusting only spray angles or nozzle positions. Similarly, firefighting robots typically achieve coverage via base position and manipulator posture adjustments while maintaining fixed nozzle orientation [6], a strategy that is inefficient in cluttered or multi-level environments and prone to water wastage. In summary, traditional spraying methods are constrained by limited control degrees of freedom, poor environmental adaptability, and insufficient generalization.

These limitations highlight a fundamental challenge in robotic water manipulation: the difficulty of obtaining an accurate analytical model of the water jet's behavior. The trajectory of a water jet is influenced by a complex set of parameters, which can be categorized into observable and unobservable factors. Observable parameters include the nozzle's position and orientation, the initial jet velocity, and the target coordinates. These can be measured or controlled directly. However, numerous critical parameters are unobservable or highly uncertain in practice, such as the distribution of droplet sizes (which significantly affects air resistance), real-time fluctuations in flow rate and pressure, ambient air currents and turbulence, temperature-dependent variations in fluid viscosity and air density, surface tension effects, droplet breakup during flight, etc. The combined influence of these unobservable factors makes precise analytical modeling extremely challenging, as they introduce nonlinearities and random variations that are difficult to capture with deterministic equations. Conventional control methods for robotic water jet tasks can be broadly categorized into several approaches. Model-based methods, such as inverse kinematics combined with ballistic trajectory models, compute the required nozzle pose by solving analytical equations of motion under simplifying assumptions (e.g., neglecting air resistance or assuming uniform droplet size) [7]. While computationally efficient, these methods suffer from significant inaccuracies when unobservable factors are present, as their underlying models cannot capture the stochastic nature of real-world fluid behavior. Model Predictive Control (MPC) has been applied in structured agricultural settings [9], where it optimizes control inputs over a receding horizon based on a predefined dynamic model. However, MPC's performance degrades substantially in dynamic environments where disturbances cannot be accurately predicted, and its computational complexity increases with model fidelity. Adaptive control techniques attempt to compensate for modeling errors online by estimating uncertain parameters during operation. Although they offer some robustness, these methods struggle with the high-dimensional, nonlinear dynamics of fluid–structure interaction and often require manual tuning of adaptation laws. More recently, learning-based approaches have emerged, but early attempts still relied on hybrid architectures that combine learned components with fixed analytical models [7], limiting their ability to fully adapt to complex environments. The fundamental limitation shared by these traditional methods is their reliance, whether explicit or implicit, on some form of analytical model that cannot adequately represent the unobservable factors discussed above. This motivates the adoption of a purely data-driven, model-free reinforcement learn-

ing framework that learns control policies directly from interaction experience, without requiring any prior analytical model of the fluid dynamics.

2.2. Experience Replay in Reinforcement Learning

Among existing experience replay mechanisms, HER provides an effective approach for goal-conditioned tasks under sparse rewards. By integrating with Universal Value Function Approximators [11], HER enables generalization to unseen desired goals. In long-horizon reaching tasks, HER demonstrates high sample efficiency [12,13]. To enhance HER's performance, several research directions have been explored. These include improving the diversity of achieved goals through curriculum learning [14], leveraging demonstration data to boost training efficiency [15], generating more valuable desired goals [16], and incorporating curiosity-driven exploration mechanisms [17]. Further extending this line of work, some studies have combined continual reinforcement learning with HER in methods like RHER to enhance training efficiency [18]. Nonetheless, in complex environments such as the reaching sub-task within the water manipulation domain considered in this work, these methods, like the original HER, fail to account for the differential learning utility of various states. To provide a clearer understanding of the HER mechanism, Figure 2 illustrates the process of hindsight goal relabeling. As shown in Figure 2, failed trajectories can be relabeled with alternative goals that were actually achieved later in the episode, enabling the agent to learn from otherwise unsuccessful experiences.

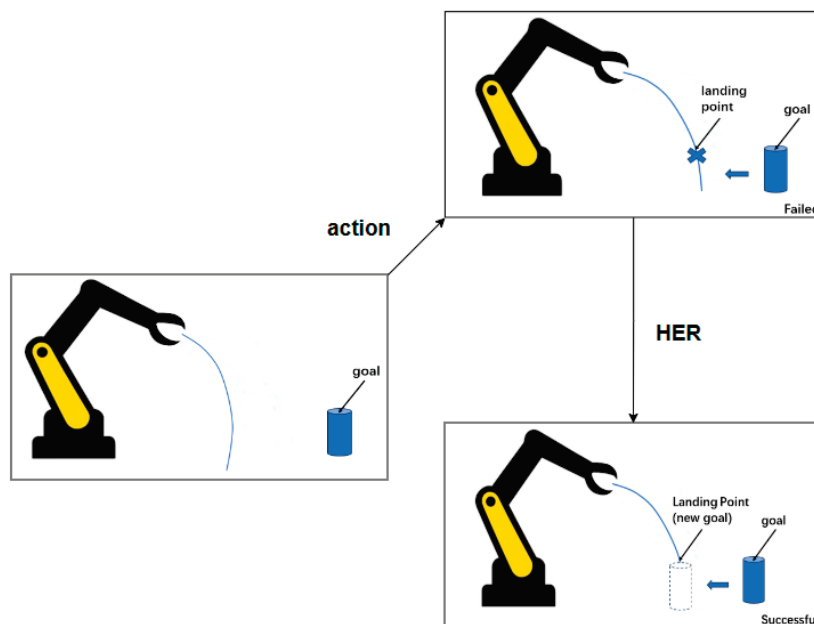


Figure 2. Illustration of the goal relabeling mechanism in HER.

While HER increases sample volume, PER emphasizes prioritized sampling. Recent methods aim to improve prioritization by incorporating informative signals, such as contact energy [19]. For example, Energy-Based Prioritization (EBP) [20] quantifies trajectory importance via an energy function, sampling high-energy episodes more frequently. Maximum Entropy-regularized Prioritization (MEP) [21] promotes exploration by encouraging uniform goal distributions, yet goal selection within trajectories remains uniform, allowing simpler samples to dominate. Studies have shown that directly combining HER and PER can sometimes perform worse than either alone [22], suggesting a mechanistic conflict. In complex water manipulation tasks with intricate dynamics and large goal spaces, such direct integration often fails to balance convergence speed and policy generalization.

3. Preliminaries

This section introduces the theoretical foundations used in this study. First, the Markov Decision Process (MDP) formulation of the reinforcement learning problem is presented. Then, the basic principles of experience replay mechanisms are introduced. Finally, the water jet motion model under air resistance is described, which provides the physical basis for the simulation environment used in the proposed framework.

3.1. Markov Decision Process

The core framework of RL is based on MDP, defined by the tuple $M = (S, A, P, R, \gamma, T)$, where S represents the state space, with $s_t \in S$ denoting the state at time t . $a_t \in A$ represents the action space, denoting the action taken at time t . $P\{\cdot\}$ denotes the state transition probability function. The probability of the state at time t , p_t can be described as:

$$p_t(s'|s, a) = P\{S_t = s' | S_{t-1} = s, A_{t-1} = a\}, \quad (1)$$

the immediate reward r_t can be computed as:

$$r_t = R(s_t, a_t), \quad (2)$$

where R represents the reward function. The ultimate objective of reinforcement learning is to learn an optimal policy π^* that maximizes the expected cumulative discounted reward:

$$J(\pi) = E_p \left[\sum_{t=0}^T \gamma^t r_t(s_t, a_t, s_{t+1}) \right], \quad (3)$$

where $J(\pi)$ denotes the expected return under policy π , where $E_p[\cdot]$ denotes the expectation under the trajectory distribution induced by policy π and the environment dynamics, the scalar $\gamma \in [0, 1]$ represents the discount factor, t is the current number of episodes and T is the maximum episode length. The MDP framework establishes the mathematical basis for representing the robotic water jet control task within the reinforcement learning paradigm.

3.2. Experience Replay Algorithm

The Hindsight Experience Replay [12] introduces a goal relabeling mechanism that enables the agent to generate a large number of informative samples even from failed trajectories, thereby alleviating the inefficiency of learning in sparse-reward environments. HER relies on three key concepts: Desired goal (g^d), achieved goal (g^a) and hindsight goal (g^h).

Specifically, HER samples a subset of transitions from past trajectories and, with a given probability, replaces the original goal g^d with a hindsight goal g^h drawn from future achieved goals within the same episode. The reward is recomputed:

$$(s_t, a_t, r_t, s_{t+1} | g^d) \rightarrow (s_t, a_t, r'_t, s_{t+1} | g^h), \quad (4)$$

the reward r_t for a failed step is the failure reward. After applying HER, those steps are reinterpreted as successful, and their reward value is converted to the success reward r'_t . This process effectively transforms negative samples into informative positive samples without altering the task definition, thus greatly enhancing learning efficiency in sparse-reward environments.

PER assigns higher sampling probabilities to transitions with greater TD errors δ , defined as:

$$\delta = Q_\phi(s_t, a_t) - [r(s_t, a_t) + \gamma Q_\phi(s_{t+1}, a_{t+1})], \quad (5)$$

where $Q_\phi(s_t, a_t)$ denotes the Q -value for the current state s_t and action a_t , with ϕ representing the parameters of the Q -network. The scalar $\gamma \in [0, 1]$ represents the discount factor. The term $r(s_t, a_t)$ signifies the immediate reward received for executing action a_t in state s_t . The sampling probability is calculated by the following formula:

$$p(i) = \frac{p_i^\alpha}{\sum_{j=0}^N p_j^\alpha}, \quad (6)$$

where N is the total number of samples in the buffer. p_i is the priority value $p_i = |\delta(i)| + \epsilon$, and α controls the degree of prioritization. Existing experience replay mechanisms such as HER and PER significantly improve learning efficiency, but their direct combination may still face challenges in complex environments with sparse rewards and large goal spaces, which motivates further improvements in replay strategies.

3.3. Water Jet Motion Modeling Under Air Resistance

In robotic water manipulation simulations, the trajectory of the water jet is influenced not only by gravity but also by aerodynamic drag, which becomes significant in the medium-to-low velocity range (several meters per second). The Reynolds number (Re) of droplets typically lies between 20 and 400, ignoring air resistance would result in substantial trajectory deviations, leading to discrepancies between simulation and reality. To address this issue, this study models air resistance using the classical drag force model [23]:

$$\vec{F}_{\text{drag}} = -\frac{1}{2}\rho_a A C_D |\vec{v}_0| \vec{v}_0 = -\beta C_D |\vec{v}_0| \vec{v}_0, \quad (7)$$

where ρ_a is the air density, A is the droplet cross-sectional area with radius r , and v_0 is the velocity vector of the fluid particle. The parameters β and droplet mass m are expressed as functions of the particle radius r :

$$\beta(r) = \frac{1}{2}\rho_a A = \rho_a \frac{\pi}{2} r^2, \quad (8)$$

$$m(r) = \rho \frac{4\pi}{3} r^3, \quad (9)$$

where the drag coefficient C_D for viscous droplets with dynamic viscosity contrast is obtained from Feng [24] as a function of Reynolds number:

$$Re(|\vec{v}_0|, r) = \frac{2r\rho_a |\vec{v}_0|}{\mu_a}, \quad (10)$$

where μ_a denotes the dynamic viscosity of air. The total force acting on each droplet in air can thus be expressed as:

$$m\dot{\vec{u}} = \vec{F}(t) = \vec{F}_g + \vec{F}_{\text{drag}} = -(m - m_a)g\vec{e}_{WZ} - \beta C_D |\vec{v}_0| \vec{v}_0, \quad (11)$$

where m_a is the mass of air displaced by the droplet (i.e., the buoyant force related mass), and g denotes the gravitational acceleration. Under the conditions specified in this study $T = 20^\circ\text{C}$, the parameter values are as follows: $g = 9.81 \text{ m/s}^2$, $\rho_a = 1.204 \text{ kg/m}^3$, $\rho = 1.204 \text{ kg/m}^3$ [25] and $\mu_a = 1.813 \cdot 10^{-5} \text{ kg/ms}$ [26].

As shown in Figure 3, the proposed water-jet motion model incorporates the effects of air resistance and other physical forces, enabling a more realistic representation of water jet

dynamics in the simulation environment. This physical model provides the foundation for accurately evaluating the control policies learned by the reinforcement learning agent.

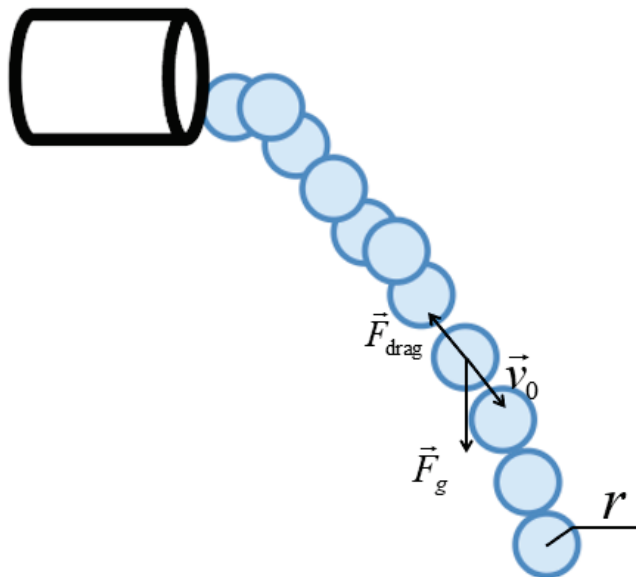


Figure 3. Force analysis of a water droplet under air resistance at each simulation step. The droplet of radius r experiences gravity $\vec{F}_g$, aerodynamic drag $\vec{F}_{\text{drag}}$ opposing the velocity direction $\vec{v}_0$.

4. Methodology

4.1. State and Action Space of the Simulation Environment

The simulation environment for the robotic water jet control task was developed using the PyBullet physics engine, incorporating a particle-based fluid model to simulate the behavior of water droplets. All simulation experiments were conducted using Python 3.11 and PyBullet (version 3.2.7) on Ubuntu 22.04. The kinematic analysis of the water jet trajectory is formulated as follows:

As shown in Figure 4, let the world coordinate system be denoted as $\Sigma_w : O - XYZ$, and the end-effector coordinate system of the robotic arm as $\Sigma_z : o - xyz$. Its pose is represented by a rotation matrix $R \in SO(3)$ and a translation vector $P \in \mathbb{R}^3$. The rotation matrix R satisfies the definition of the Special Orthogonal Group:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \right\}, \quad (12)$$

where $P = [p_x, p_y, p_z]^T$ represents the coordinates of the origin of the end-effector frame Σ_z expressed in the world frame Σ_w . The column vectors of the rotation matrix R correspond to the projections of the basis vectors of Σ_z onto Σ_w , given by:

$$\mathbf{R} = [\mathbf{e}_x \quad \mathbf{e}_y \quad \mathbf{e}_z], \quad (13)$$

here, $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ represent the unit basis vectors of Σ_z projected in Σ_w . The Z-axis of the end effector coordinate system corresponds to the direction of the water jet. The jet direction vector can be obtained from the third column of $\mathbf{R}$:

$$\mathbf{d} = \mathbf{e}_z = \mathbf{R}[:, 3] = [d_x, d_y, d_z]^T, \quad (14)$$

which satisfies the normalization condition:

$$\|\mathbf{d}\| = \sqrt{d_x^2 + d_y^2 + d_z^2} = 1, \quad (15)$$

where the pitch angle θ and yaw angle ϕ are used to describe the orientation of the nozzle relative to the world coordinate:

$$\theta = \arcsin(d_Z), \quad (16)$$

$$\phi = \arctan(d_Y/d_X). \quad (17)$$

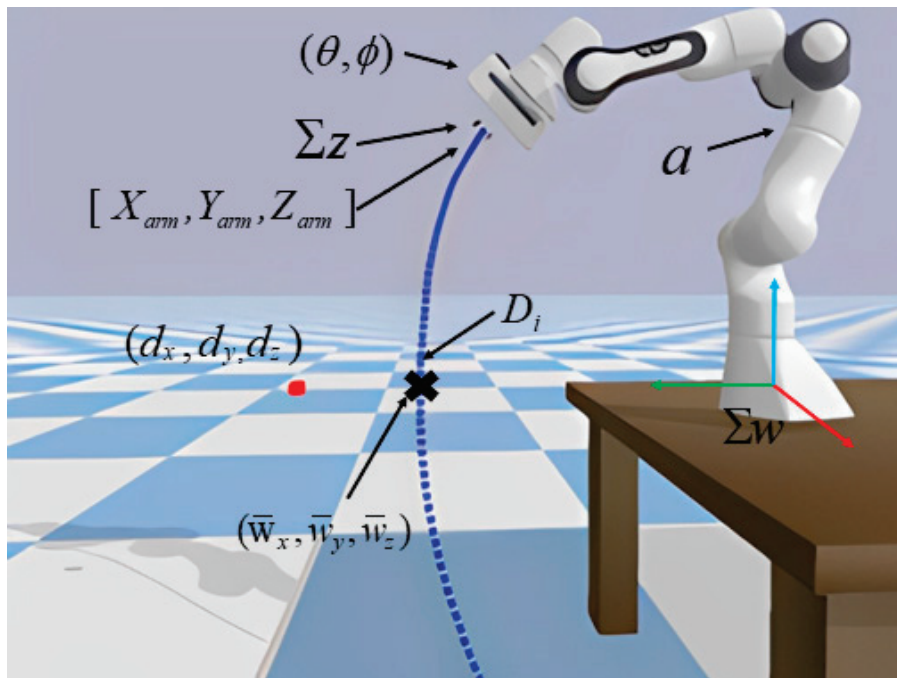


Figure 4. Schematic diagram of coordinate frames and nozzle orientation. The world frame $\{W\}$ and end-effector frame $\{Z\}$ are shown, with the water jet direction aligned with the Z -axis of $\{Z\}$. The pitch angle θ and yaw angle ϕ define the nozzle orientation relative to $\{W\}$. D_i denotes the selection of fluid particles close to the target plane, and its state space and action space are also annotated in the figure.

For a desired target location (d_x, d_y, d_z) , all simulated particles are sorted by their vertical distance from the target plane, and the first k particles with the smallest distances are selected to form a cluster. The centroid of this cluster $\bar{w} = (\bar{w}_x, \bar{w}_y, \bar{w}_z)$ defines the landing position of the water jet:

$$(\bar{w}_x, \bar{w}_y, \bar{w}_z) = \frac{1}{k} \sum_{i \in I_k} D_i. \quad (18)$$

In the proposed robotic water jet control environment, the state space $s \in R^8$ consists of three components:

- (1) The Cartesian position of the end-effector $[X_{arm}, Y_{arm}, Z_{arm}]^T \in R^3$;
- (2) The nozzle orientation described by pitch and yaw angles $(\theta, \phi) \in R^2$;
- (3) The distribution of the landing points of the water droplets on the target plane $(\bar{w}_x, \bar{w}_y, \bar{w}_z) \in R^3$.

The action space $a \in R^7$ corresponds to the continuous joint position commands for the seven joints of the robotic manipulator. Each action represents the target position increment applied to the corresponding joint at each control step. This formulation enables smooth motion control while maintaining compatibility with the physics simulation and facilitating sim-to-real transfer.

In summary, this section defines the state and action representations used in the robotic water jet control environment. The formulation captures the essential geometric relationship between the manipulator pose, jet direction, and landing position of the water particles.

For simplicity, several assumptions are adopted in the simulation environment, including a simplified particle-based representation of the water stream. While these simplifications improve computational efficiency during reinforcement learning training, they may not fully capture all complex fluid dynamics in real-world spraying scenarios. Nevertheless, the adopted model is sufficient for learning stable control policies and provides a reliable platform for evaluating the effectiveness of the proposed reinforcement learning framework.

4.2. Reward Function Design

Given the geometric nature of the water jet control task, the primary objective of the agent is to minimize the distance between the actual water jet landing point and the target location. Thus, a sparse reward function is defined as:

$$R = \begin{cases} 0, & \text{if } \|(w_x, w_y, w_z) - (d_x, d_y, d_z)\| \geq \varepsilon \\ 1, & \text{if } \|(w_x, w_y, w_z) - (d_x, d_y, d_z)\| < \varepsilon \end{cases} \quad (19)$$

where $(\bar{w}_x, \bar{w}_y, \bar{w}_z)$ is the centroid of the selected water particle cluster, (d_x, d_y, d_z) is the target position, and ε represents a small tolerance threshold. In this work, a sparse reward scheme is adopted. The use of sparse rewards reduces the risk of bias introduced by artificially designed dense reward functions [12].

The sparse reward formulation encourages the agent to focus on achieving accurate target reaching without introducing additional heuristic biases. However, this reward design simplifies the task objective by considering only the final landing position of the water jet. Other practical factors, such as water distribution efficiency, obstacle interference, or energy consumption, are not explicitly modeled in this study and may be considered in future work.

4.3. Goal-Priority Hindsight Experience Replay

Although HER has certain advantages in expanding the number of samples, the data it generates are not fully utilized by PER. The core idea of HER is to convert failed experiences into successful ones through goal relabeling, which can significantly increase the number of positive samples. Nevertheless, these ‘hindsight goals’ are essentially virtual successes computed based on substitute goals. Since trajectories rarely reach high-difficulty positions during training, simpler goals are more likely to be resampled by HER. When these samples are incorporated into PER’s priority calculation, the TD error may not accurately reflect their value for policy improvement, and they may even be assigned excessively high priorities, leading the agent to over-rely on pseudo-success samples during training and weakening the exploration of true goals. This phenomenon is particularly prominent in complex water jet control tasks, where the task goal space is large, samples of distant or high-difficulty goals are scarce, and the substitute samples generated by HER dilute the weight of these key experiences in PER, ultimately affecting the overall training effectiveness and policy reliability. Meanwhile, in water jet control task, the system has higher requirements for reaching remote or boundary target positions, and relying solely on HER and PER makes it difficult to balance efficiency and success rate. To address this limitation, this paper introduces an enhanced experience replay sampling strategy named GPHER. This method introduces a goal relevance measure on the basis of traditional PER and thereby optimizes the sample selection strategy to more effectively highlight experiences closely related to

the task goals during sampling. As shown in Figure 5, the GPHER framework designed in this paper further demonstrates its improved mechanism and workflow when combining HER and PER.

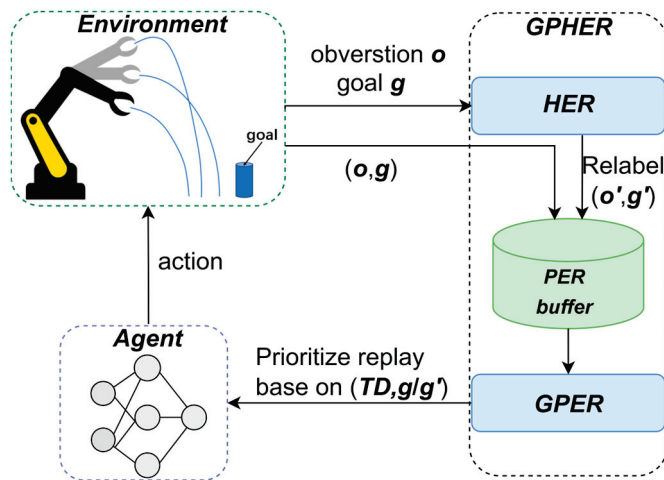


Figure 5. Structure of the GPHER framework.

Within the framework of traditional PER that allocates sampling priorities based on TD error, it further adds the goal-space center distance as a weighting factor, thereby increasing the sampling probability of samples far from the goal center and forcing the agent to focus more on learning in high-difficulty goal regions. Meanwhile, the priority calculation not only depends on the desired goal itself but also incorporates the spatial relationship between the achieved goal and the goal center. Under the same goal conditions, it prioritizes sampling trajectory samples that are closer to difficult goals. The revised priority formula is:

$$p_i = |\delta_i| \cdot (1 - \lambda) + \frac{\|g'_i - g_m\|}{\|g_{\max} - g_m\|} \cdot \lambda + \frac{\|\bar{w} - g_m\|}{\|g_{\max} - g_m\|} \cdot \lambda + \epsilon, \quad (20)$$

where g'_i is the goal of the current state transition tuple, g_m is the center of the goal space, m is the dimensionality of the goal space, $g_{\max}$ is the farthest distance in the goal space. $\bar{w}$ is the achieved position, The change in λ is as follows:

$$l_n = l_0 + (l_N - l_0) \frac{n}{N}, \quad (21)$$

where λ_n represents the parameter value at training step n , the maximum number of steps is N , the initial value is $\lambda_0 = 0$, and the final value is $\lambda_N = 0.4$. This dynamic adjustment mechanism is similar to the idea of Curriculum Learning, but the difference is that it acts on the experience replay mechanism rather than directly changing the task goals. Compared to achieving curriculum-like training by modifying goals, this method can gradually increase the sampling difficulty without introducing environmental mutations, thereby focusing more on learning difficult goals in the later stages of training.

Since the sampling probability of samples is proportional to the priority measure, the experience distribution deviates from the uniform sampling assumption, thereby introducing estimation bias. To eliminate this bias, PER introduces importance sampling weights:

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)} \right)^\beta, \quad (22)$$

where N represents the size of the experience pool, $P(i)$ is the sampling probability of sample i . The parameter $\beta \in [0, 1]$ controls the correction strength. The weights are typically normalized:

$$w_j = \left((N \cdot P(j))^{-\beta} \right) / \max_i (w_i), \quad (23)$$

the pseudocode of the Algorithm 1 is presented as follows:

Algorithm 1. TQC-GPHER

```

Initialize the policy network TQC parameters, total training steps  $N$ , current step  $n$ ,
minibatch size  $k$ , replay period  $K$ , buffer  $D$  and  $p_1 = 1$ 
for episode = 1,  $X$  do
  Sample a goal  $g$  and an initial state  $S_t$ 
  for  $t = 0, T$  do
    Select an action using the policy  $\pi_\theta$  from Algorithm:  $a_t \leftarrow \pi_\theta(S_t \| g)$ 
    Execute action, interact with the environment and observe  $S_{t+1}$ 
  end for
  for  $t = 0, T$  do
    Compute reward  $r(t) = R(s, a, s', g)$ 
    Store transition  $R(s, a, s', g)$  into buffer  $D$  with priority:  $p_t = \max_{i < t} p_i$ 
    Sample a set of goals  $G$  from the current episode states
    for  $g' \in G$  do
      Compute relabeled reward  $r(t) = R(s, a, s', g')$ 
      Store relabeled transition  $(s_t, a_t, s_{t+1}, r'_t, g')$  into replay buffer  $D$ 
    end for
  end for
  for  $t = 1 \bmod K$  then
    for  $j = 1$  to  $k$  do
      Sample transition  $i$  from the buffer with probability  $p(i)$ 
      Compute importance-sampling weight  $w_j$  as defined in Equation (23)
      Compute TD error according to Equation (5)
      Update transition priority  $p_i$  according to Equation (20)
    end for
    Update network parameters
  end for
end for

```

In the subsequent experimental setup, this paper selects the currently well-performing Truncated Quantile Critics (TQC) algorithm as the basic reinforcement learning framework. On this basis, the proposed GPHER mechanism (Algorithm 1) is compared with various mainstream experience replay methods to systematically evaluate its performance improvement in water jet control task. The overall algorithm flow of TQC-GPHER is shown in Figure 6. The robotic manipulator interacts with the simulation environment to generate trajectories, which are stored in the replay buffer and processed by the proposed GPHER sampling strategy before being used to update the policy network.

Overall, the proposed GPHER mechanism improves experience replay efficiency by integrating TD-error prioritization with goal-space difficulty information. This design enables the agent to focus more effectively on challenging goal regions during training. Nevertheless, the current formulation assumes that goal difficulty can be reasonably approximated by the distance from the goal-space center. Although effective in the considered water jet control task, more complex environments may require alternative difficulty met-

rics or adaptive goal representations, which remain an interesting direction for future research.

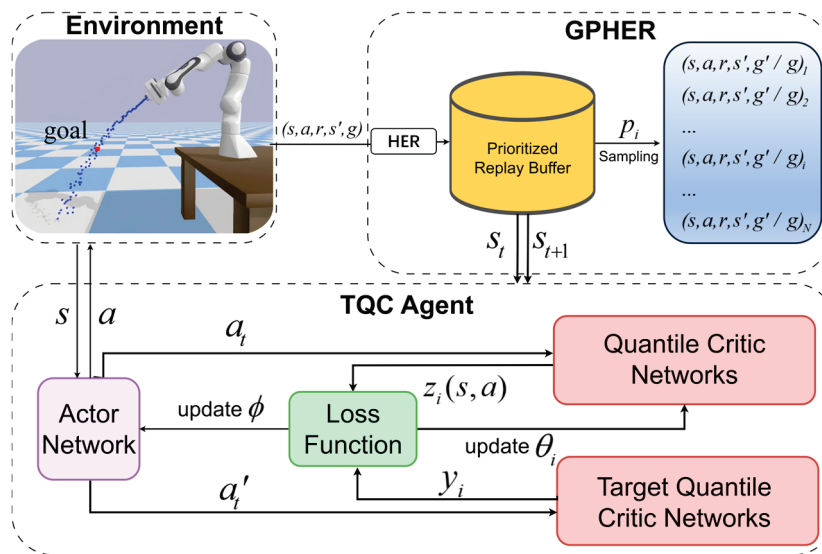


Figure 6. Overall algorithmic framework of the proposed TQC-GPHER.

5. Simulations and Experiments

5.1. Simulation Environment Setup for the Robotic Water Jet Control Task

Direct training in real-world environments often faces issues such as low sample efficiency, high safety risks, and prohibitive experimental costs [14]. Therefore, this paper chooses to train the robotic arm in a simulation environment to accomplish the water jet control task and validate the performance of the proposed TQC-GPHER algorithm. The experimental platform is built based on the PyBullet physics engine combined with the Gymnasium library to construct the reinforcement learning environment, as shown in Figure 7. The figure shows the training process of an episode.

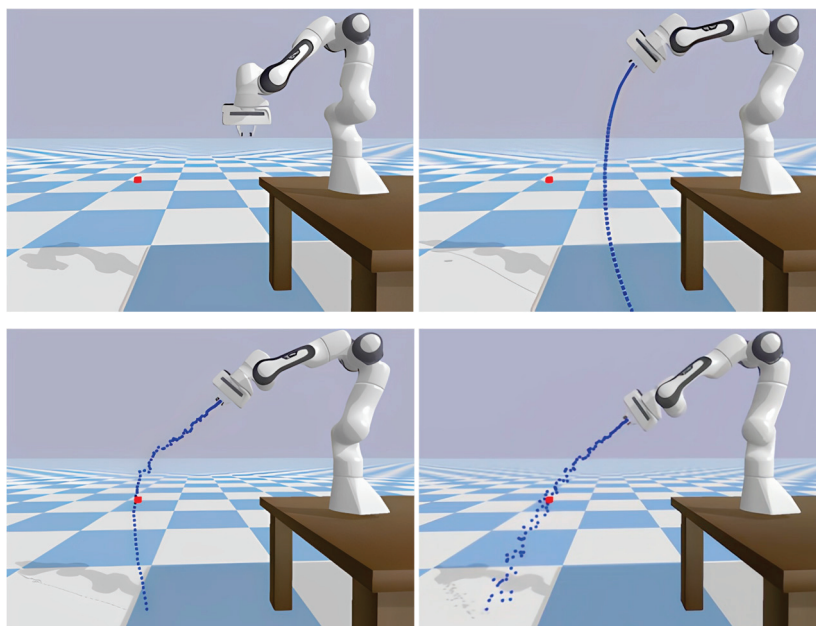


Figure 7. Simulation environment setup for the water jet control task, showing a successful episode during training. The red cube indicates the target position that the water jet needs to reach.

In this environment, the agent adjusts the water outlet position and jet angle by controlling the pose of the robotic arm. To ensure the quantifiability and observability of the experiments while maintaining applicability to real-world scenarios, this paper adopts a design scheme where “the robotic arm gripper holds a water pipe for spraying.” It should be noted that the water flow is modeled and generated by the PyBullet particle system. As this model does not explicitly consider aerodynamic effects, this study further incorporates the modeling of air resistance on fluid particles, adding air resistance to the motion state of each fluid particle at each simulation time step to improve the physical plausibility of the simulation and the credibility of the experimental results. This provides a more reliable foundation for subsequently applying the trained policies to real-world water jet control tasks (such as firefighting, agricultural irrigation, or high-altitude cleaning). The details of the simulation setup are presented in Table 1.

Table 1. Simulation environment parameters.

Parameter	Symbol
End-effector position of the manipulator	$[X_{arm}, Y_{arm}, Z_{arm}]^T$
End-effector orientation of the manipulator	R
Three-dimensional outlet velocity of water jet	v_0
Pitch angle of the nozzle	θ
Yaw angle of the nozzle	ϕ
Desired landing point position	(d_x, d_y, d_z)
Radius of fluid particles	r

First, the range of the water jet control task needs to be analyzed to determine a suitable target scope. The diameter d of natural raindrops is constrained to approximately 1–7 mm. In each simulation time step, fluid particles with a radius $r = 3$ mm are selected for jetting. The reachable space of the fluid is determined through Monte Carlo sampling to establish the set of reachable targets. According to Figure 8, the target range can be taken as $\mathcal{D}$. The initial velocity for particle jetting is set to 4 m/s. Data from the reachable target dataset where $d_x > 0.5$ are used for training:

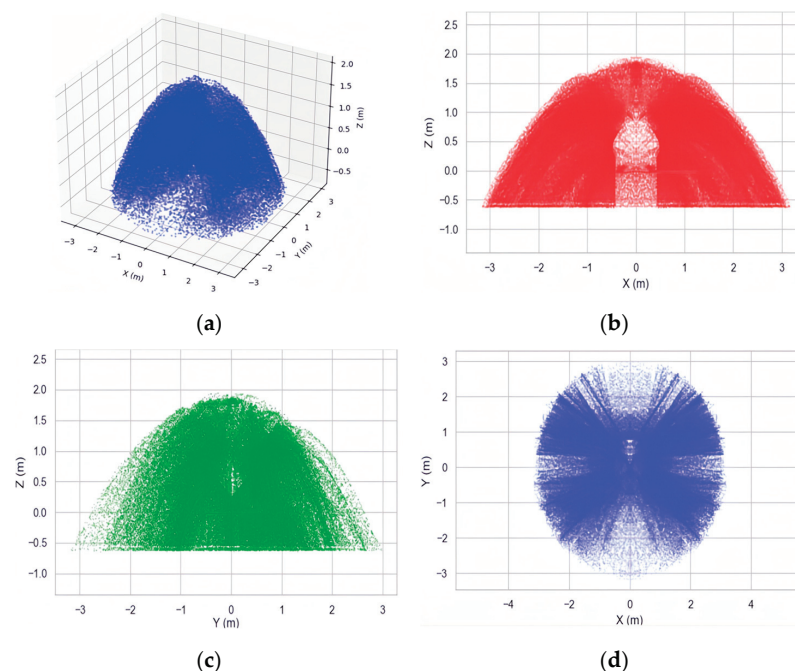


Figure 8. Reachable workspace of water droplets in the simulated environment: Subfigures (a–d) illustrate the reachable workspace of water droplets from different viewpoints. Subfigure (a) shows

the three-dimensional distribution of reachable points, while (b,c) present the front and side views projected onto the X–Z and Y–Z planes, respectively. Subfigure (d) displays the top view on the X–Y plane, highlighting the overall coverage pattern of the water droplets in the workspace.

5.2. Comparison with Baseline

To comprehensively evaluate the proposed method in the water jet control task, comparative experiments were conducted in the simulation environment using TQC as the base RL framework against various baseline experience replay mechanisms. All experiments used the same hyperparameters, listed in Table 2.

Table 2. GPHER-TQC algorithm hyperparameters.

Hyperparameter	Value	Hyperparameter	Value
Learning rate	0.01	Neurons per hidden layer	256
Discount factor	0.98	Number of hindsight relabeled goals	4
Soft update coefficient	0.95	Entropy coefficient	1
Replay buffer capacity	1×10^6	Entropy target	H_0
Batch size	512	Number of quantiles	25
Maximum episode length	200	Huber quantile loss parameter	1
Number of network layers	2	Prioritization exponent	0.6

For fair comparison, all methods were trained for the same total of 750,000 environment steps, and the success rate curves report the average performance over five independent training runs. Since the episode lengths vary across different runs, the success rate data were interpolated to a unified episode interval. A moving average was further applied to smooth the curves and reduce the influence of stochastic fluctuations. Figure 9 shows the learning curves of each method. The results indicate that when HER and PER are simply combined, as PER relies solely on TD error for importance sampling, the virtual goal samples introduced by HER relabeling are not fully utilized, instead weakening the prioritized sampling effect of PER, leading to overall performance even worse than using HER alone. After approximately episode 1500, the success rate of this method shows a significant decline, indicating a lack of policy stability. In contrast, the proposed GPHER exhibits faster convergence speed in the early stages of training and reaches a success rate of about 95% around episode 1500. Subsequently, its success rate remains stable at around 98%, significantly outperforming HER and Prioritized Hindsight Experience Replay (PHER) (85% with large fluctuations).

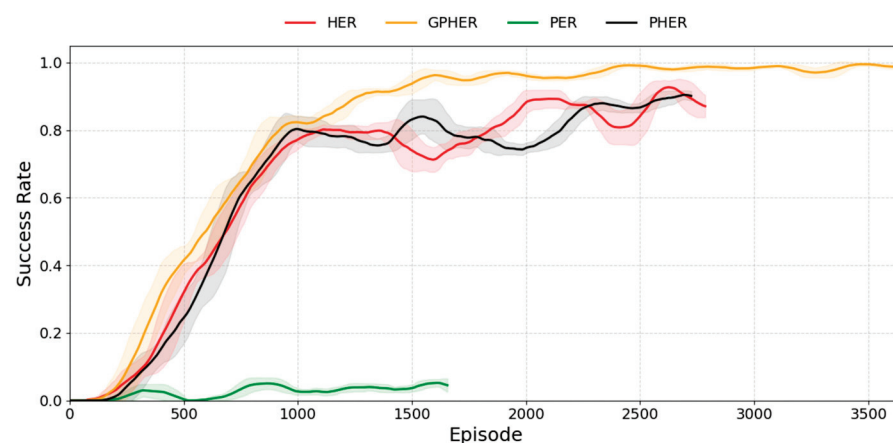


Figure 9. Comparison of training performance between the proposed GPHER and baseline algorithms.

For the water jet control task, stable reaching performance across the entire reachable space is essential. To evaluate performance in challenging goal regions, comparative tests were conducted between PHER and GPHER, both outperforming HER. Each method was tested over 100 episodes on two representative planes: the horizontal plane ($X = 1$) and the vertical plane ($Z = 0$), where the fluid particle landing density is highest. For consistency, all target regions were defined as square areas and discretized into 16 configurations. On the $X = 1$ plane, regions I–IV and A–D corresponded to evenly spaced segments along the Z-axis and Y-axis, respectively, while on $Z = 0$, they represented divisions along the X- and Y-axes. As shown in Figure 10, GPHER achieves an average success rate of 96.8% in the edge and high-difficulty regions across the two evaluated scenarios, which is substantially higher than the 91.46% achieved by PHER. These results indicate that GPHER exhibits markedly improved performance in challenging boundary regions, demonstrating superior exploration efficiency, policy generalization, and robustness compared with PHER.

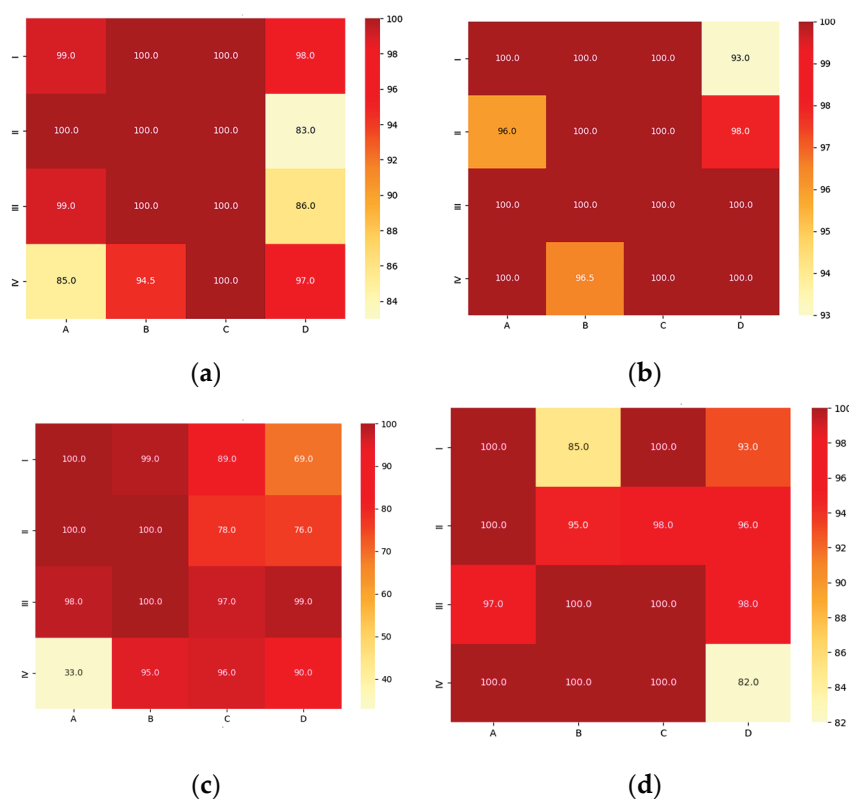


Figure 10. Performance heatmaps of GPHER and PHER in different goal regions of the environment: Subfigures (a,b) show the success rate distributions of GPHER when the target location is fixed at $X = 1$ and $Z = 0$, respectively. Subfigures (c,d) display the corresponding success rate distributions for PHER under the same conditions. Each heatmap reflects the success rate across different goal regions, with darker regions indicating higher success.

5.3. Ablation Experiment

To evaluate the contribution of each component in GPHER, ablation experiments were conducted in the same simulation environment, comparing TQC alone, HER, PER, and GPHER. As shown in Figure 11, HER accelerates early-stage learning, reaching 80% success around episode 1100, but its relabeling mechanism favors simpler goals, limiting exposure to difficult targets and yielding a final success rate of 82%. In contrast, GPHER maintains higher sample efficiency throughout training and achieves a stable final success rate of 98%, significantly outperforming HER. This confirms that the joint prioritization of goal difficulty and TD error in GPHER effectively mitigates HER's bias toward simple samples, enhancing both policy learning efficiency and generalization.

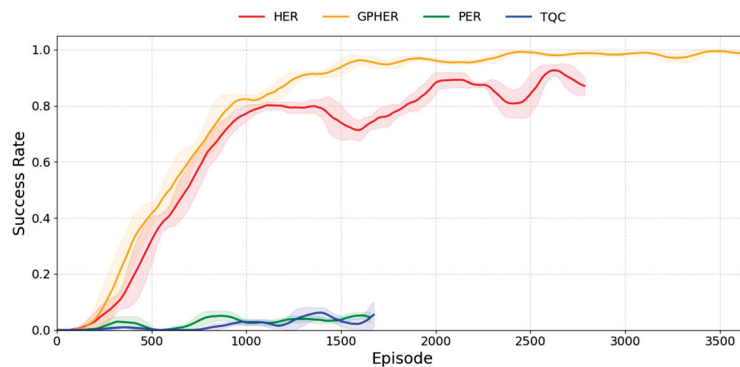


Figure 11. Ablation study results comparing the performance contribution of different components in GPHER.

5.4. Extending PGHER to Other Reaching Tasks

To further verify the general applicability of the proposed GPHER method, this section selects the relatively simple robotic arm Reach task as a supplementary experimental scenario. In the experimental setup, adjustments were made to the previous training details: During the same 250,000 training steps, the minibatch size was reduced from 512 to 256, and the PyTorch-based DDPG algorithm was used as the base learning framework.

In the Reach task, the robot end-effector was required to reach randomly sampled target positions within a predefined three-dimensional workspace. Specifically, the goal positions were uniformly sampled from a cubic region defined in the world coordinate frame, with ranges of $[-1, 1]$ m along the x-axis, $[0, 2]$ m along the y-axis, and $[0.1, 0.3]$ m along the z-axis.

A reaching attempt was considered successful if the Euclidean distance between the end-effector position and the target goal was below a predefined threshold of 0.05 m. The success rate was computed as the ratio of successful episodes over the total number of episodes and was averaged across multiple independent training runs.

The experimental results are shown in Figure 12. As shown in the figure, GPHER achieves a success rate of approximately 0.95 within the first 500 training episodes, significantly outperforming the baseline methods during the early training stage. This indicates substantially higher sample efficiency compared with HER, PER, and their direct combination. Moreover, GPHER maintains a higher and more stable success rate throughout the remainder of the training process, with reduced performance fluctuations relative to the comparison algorithms.

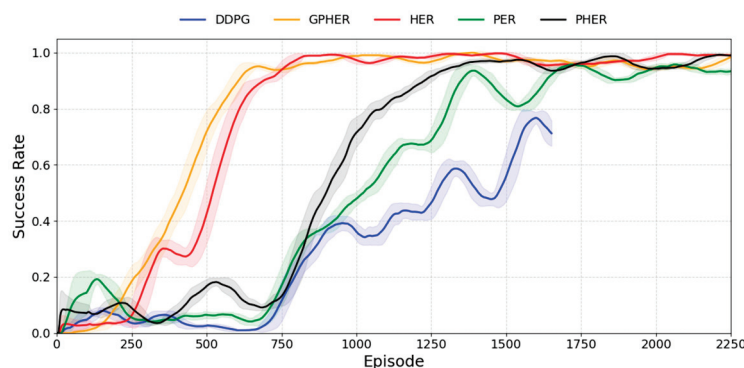


Figure 12. Comparative training results of GPHER and baseline algorithms on the Reach task.

5.5. Real-World Experiments

This paper further validates the effectiveness of the proposed method on a physical platform. The experiment adopts a two-stage deep reinforcement learning training

framework, which separately models visual perception and the control policy. As shown in Figure 13, the visual perception module performs target detection and localization of the watering area and landing points, providing this as structured input to the control policy module for subsequent path planning and action decision training. This design effectively avoids the reliance of end-to-end methods on large-scale fine-grained modeling and high-quality annotated data while enhancing the transferability and generalization of the algorithm.

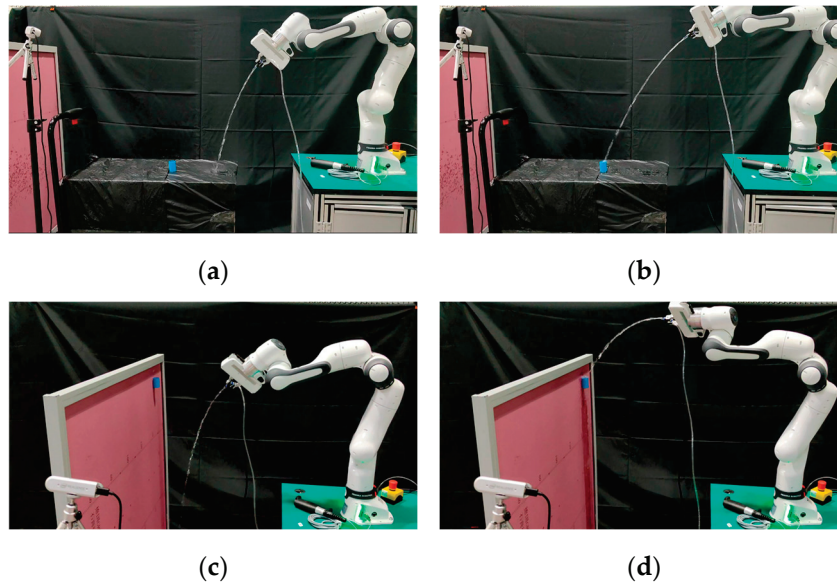


Figure 13. Real-world experimental validation of the proposed GPHER method on a physical robotic arm platform: Subfigures (a,b) show the process of the horizontal target test, and Subfigures (c,d) show the process of the vertical target test.

The real-world experiments were conducted using a Franka Emika Panda robotic manipulator, a widely used 7-degree-of-freedom (7-DOF) collaborative robot designed for research applications. The robot has a maximum reach of approximately 0.85 m and a payload capacity of 3 kg. A water nozzle connected to a pressurized water source was mounted at the end-effector of the robotic arm, allowing the jet direction to be controlled through the end-effector pose.

During the experiments, the robotic arm was used to adjust the position and orientation of the nozzle to direct the water jet toward specified target locations. The distance between the robotic arm base and the target surface was approximately 0–2 m, which lies within the reachable workspace considered in the simulation environment. The water jet velocity was maintained at approximately 4 m/s, consistent with the parameter settings used in the simulation environment.

The control commands were generated by the trained reinforcement learning policy and executed through the robot's position control interface. The control framework was implemented on a workstation running Ubuntu, where the trained neural network model was deployed using PyTorch (version 2.5.1) for real-time inference.

All experiments were conducted in an indoor laboratory environment with relatively stable lighting conditions and minimal airflow to reduce disturbances to the water jet. Although the experimental setup simplifies certain environmental factors such as wind disturbances and water pressure variations, it is sufficient to validate the effectiveness and robustness of the proposed control framework.

For the sim-to-real validation, models trained with three different replay mechanisms (HER, PHER, and the proposed GPHER) were evaluated on a physical robotic platform

under identical experimental conditions. Two representative difficult target regions were considered: the boundary regions of the horizontal plane ($Z = 0$) and the boundary regions of the vertical plane ($X = 1$). For each scenario, 100 target positions were randomly sampled and tested.

As summarized in Table 3, the proposed GPHER method consistently achieves the highest success rates across all evaluation settings. In the difficult target regions on the vertical plane ($X = 1$), GPHER attains a success rate of 0.93, outperforming PHER and HER by 10 percentage points (0.83 for both). Similarly, on the horizontal plane ($Z = 0$), GPHER reaches a success rate of 0.95, compared with 0.84 for PHER and 0.82 for HER, corresponding to a relative improvement of approximately 13.1% and 15.9%, respectively.

Table 3. Average success rates of models in the physical robotic experiments.

Trained Model	Difficult Target Region at $X = 1$	Difficult Target Region at $Z = 0$	Random Target Positions in the Goal Space
GPHER	0.93	0.95	0.96
PHER	0.83	0.84	0.85
HER	0.83	0.82	0.81

When evaluated over randomly sampled target positions across the entire goal space, GPHER maintains a high success rate of 0.96, exceeding PHER (0.85) and HER (0.81) by 11 and 15 percentage points, respectively. These results indicate that GPHER not only improves overall task success but also demonstrates superior robustness and generalization capability in real-world conditions.

Notably, the performance gap between GPHER and the baseline methods is most pronounced in the difficult target regions, which correspond to the boundary areas of the workspace where control precision and trajectory stability are hardest to achieve. This observation aligns with the design motivation of GPHER, which explicitly increases the sampling priority of high-difficulty goals during training, enabling the learned policy to better handle edge cases that are underrepresented in standard replay strategies.

6. Conclusions and Future Works

Addressing the lack of techniques in the field of intelligent water jet control that enables directly hitting targets without imposing pose constraints, this paper proposes, for the first time, the use of reinforcement learning to control the robotic arm for this task. To address the issue where data generated by HER often concentrates within simple goal ranges, which results in an uneven goal distribution in the replay buffer, suboptimal learning efficiency, and low post-training policy success rates, particularly for critical edge targets in water jet control tasks, this paper proposes GPHER, an improved method that integrates HER and PER. Model validation shows that the proposed algorithm achieves a success rate of up to 99% in the simulated water control environment. Experiments conducted on a real robotic arm yield a success rate of 96% in the physical setup. Furthermore, validation performed on a Reach task demonstrates the method's potential for generalization. Nevertheless, several limitations remain in the current study. The experimental setup does not explicitly model environmental disturbances such as wind or water pressure variations, which may influence the water jet trajectory in real-world applications.

Future work will focus on extending the proposed framework to more complex and realistic scenarios. In particular, we plan to incorporate fluid properties such as varying flow velocities and fluid viscosity into the state representation to better capture the physical characteristics of different fluids. Furthermore, the system will be integrated with a mobile robotic platform to expand the operational workspace and improve task flexibility.

Finally, future studies will investigate the robustness of the proposed approach in outdoor environments with dynamic airflow and other environmental disturbances.

Author Contributions: Conceptualization, R.Z. and L.W.; methodology, R.Z. and J.Q.; software, R.Z. and G.L.; simulation environment construction and physical modeling, R.Z. and G.L.; reinforcement learning algorithm design and implementation, R.Z.; validation, R.Z.; formal analysis, R.Z. and J.Q.; investigation, R.Z.; resources, L.W.; data curation, R.Z.; writing—original draft preparation, R.Z.; writing—review and editing, J.Q. and L.W.; visualization, R.Z.; supervision, L.W.; project administration, L.W.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: Research supported by the National Local Co-Built Humanoid Robotics Innovation Center Open Fund, Cultivation Project Special Fund of BUCEA (No. X25014), the Artificial Intelligence + Research Institute “Challenge and Response” program of Beijing University of Civil Engineering and Architecture (Grant 06080925012) and the open project of Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology (Beijing Jiaotong University), Ministry of Education (No.Z26008).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request due to the large size of the experimental data and the hardware-dependent experimental setup.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RL	Reinforcement Learning
MDP	Markov Decision Process
HER	Hindsight Experience Replay
PER	Prioritized Experience Replay
GPHER	Goal-Priority Hindsight Experience Replay
PHER	Prioritized Hindsight Experience Replay
TQC	Truncated Quantile Critics
TD	Temporal Difference
DDPG	Deep Deterministic Policy Gradient
EBP	Energy-Based Prioritization
MEP	Maximum Entropy-regularized Prioritization
RHER	Relay Hindsight Experience Replay
Re	Reynolds Number
DOF	Degree(s) of Freedom

References

1. Dhumatkar, A.; Bhiogade, S.; Rajpal, S.; Renge, D.; Kale, V. Automatic Fire Fighting Robot. *Math. Comput. Sci. Inf. Technol. J.* **2015**, *2*, 42–46.
2. Yuan, J.F. Design of Six DOF Watering Robotic Arm. *Highlights Sci. Eng. Technol.* **2024**, *106*, 224–230. [CrossRef]
3. García-Rodríguez, G.M.; Castillo-Castañeda, E. A Robot for Facade Cleaning Based on a Cartesian Configuration: Kinematic Analysis and Prototype Construction. In *Advances in Service and Industrial Robotics*; Müller, A., Brandstötter, M., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2022.
4. Huang, Y.Q.; Wilches, J.; Sun, Y. Robot Gaining Accurate Pouring Skills through Self-Supervised Learning and Generalization. *Robot. Auton. Syst.* **2021**, *136*, 103692. [CrossRef]

5. Lin, H.T.; Fu, Y.W.; Xue, X.Y. PourIt!: Weakly-Supervised Liquid Perception from a Single Image for Visual Closed-Loop Robotic Pouring. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 241–251.
6. Jindal, K.; Wang, A.; Thakur, D.; Zhou, A.; Spurny, V.; Walter, V.; Broughton, G.; Krajnik, T.; Saska, M.; Loianno, G. Design and Deployment of an Autonomous Unmanned Ground Vehicle for Urban Firefighting Scenarios. *Field Robot.* **2021**, *1*, 1–17. [CrossRef]
7. Lin, Y.H.; Ji, W.L.; He, H.W.; Chen, Y.J. Two-Stage Water Jet Landing Point Prediction Model for Intelligent Water Shooting Robot. *Sensors* **2021**, *21*, 2704. [CrossRef] [PubMed]
8. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. *arXiv* **2015**, arXiv:1511.05952.
9. Vataavuk, I.; Vasiljević, G.; Kovačić, Z. Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator. *Agriculture* **2022**, *12*, 381. [CrossRef]
10. Wei, J.W. Research on the Mechanical Structure of Automatic Watering Robot with Track Design and Six-Axis Robot Arm. In Proceedings of the International Conference on Electronics, Electrical and Information Engineering (ICEEIE 2024), Haikou, China, 11 December 2024; Volume 13445, p. 134451S.
11. Schaul, T.; Horgan, D.; Gregor, K.; Silver, D. Universal Value Function Approximators. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015*; Bach, F., Blei, D., Eds.; PMLR: New York, NY, USA, 2015; Volume 37.
12. Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; Zaremba, W. Hindsight Experience Replay. *arXiv* **2018**, arXiv:1707.01495. [PubMed]
13. Plappert, M.; Andrychowicz, M.; Ray, A.; McGrew, B.; Baker, B.; Powell, G.; Schneider, J.; Tobin, J.; Chociej, M.; Welinder, P.; et al. Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research. *arXiv* **2018**, arXiv:1802.09464. [CrossRef]
14. Fang, M.; Zhou, T.; Du, Y.; Han, L.; Zhang, Z. Curriculum-Guided Hindsight Experience Replay. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
15. Nair, A.; McGrew, B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Overcoming Exploration in Reinforcement Learning with Demonstrations. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 6292–6299.
16. Han, C.; Peng, Z.; Liu, Y.; Tang, J.; Yu, Y.; Zhou, Z. Overfitting-Avoiding Goal-Guided Exploration for Hard-Exploration Multi-Goal Reinforcement Learning. *Neurocomputing* **2023**, *525*, 76–87. [CrossRef]
17. Li, B.; Lu, T.; Li, J.; Lu, N.; Cai, Y.; Wang, S. ACDER: Augmented Curiosity-Driven Experience Replay. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–4 June 2020; pp. 4218–4224.
18. Luo, Y.; Wang, Y.; Dong, K.; Zhang, Q.; Cheng, E.; Sun, Z.; Song, B. Relay Hindsight Experience Replay: Self-Guided Continual Reinforcement Learning for Sequential Object Manipulation Tasks with Sparse Rewards. *Neurocomputing* **2023**, *557*, 126620. [CrossRef]
19. Sayar, E.; Bing, Z.; D’Eramo, C.; Oguz, O.S.; Knoll, A. Contact Energy Based Hindsight Experience Prioritization. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 5434–5440.
20. Zhao, R.; Tresp, V. Energy-Based Hindsight Experience Prioritization. In *Proceedings of the 2nd Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018*; Billard, A., Dragan, A., Peters, J., Morimoto, J., Eds.; Proceedings of Machine Learning Research, PMLR: New York, NY, USA, 2018; Volume 87.
21. Zhao, R.; Sun, X.; Tresp, V. Maximum Entropy-Regularized Multi-Goal Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019*; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research, PMLR: New York, NY, USA, 2019; Volume 97.
22. Zhao, R.; Tresp, V. Curiosity-Driven Experience Prioritization via Density Estimation. *arXiv* **2020**, arXiv:1902.08039. [CrossRef]
23. Morrison, F.A. *An Introduction to Fluid Mechanics*; Cambridge University Press: Cambridge, UK, 2013.
24. Feng, Z.G.; Michaelides, E.E. Drag Coefficients of Viscous Spheres at Intermediate and High Reynolds Numbers. *J. Fluids Eng.* **2001**, *123*, 841–849. [CrossRef]
25. The Engineering ToolBox. Air Density, Specific Weight, and Thermal Expansion Coefficients at Varying Temperatures and Pressures. Available online: https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html (accessed on 25 July 2025).
26. The Engineering ToolBox. Air Viscosity: Dynamic and Kinematic Viscosity at Various Temperatures and Pressures. 2025. Available online: https://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d_601.html (accessed on 29 July 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Manipulation Models for Robotic High-Arc Object Transfer and Their Implementation

Junwoo Lee [†], Seunghwa Oh [†] and Jungwon Seo ^{*}

Department of Electric and Electronic Engineering, Pusan National University, Busan 46241, Republic of Korea; ljw1998@pusan.ac.kr (J.L.); seunghwa9118@pusan.ac.kr (S.O.)

^{*} Correspondence: junseo.kr@pusan.ac.kr

[†] These authors contributed equally to this work.

Abstract

This paper presents robotic manipulation methods for rapid high-arc object transfer using dynamic, non-prehensile interactions. Two complementary techniques are introduced, two-fingered scoop-and-flick and one-fingered topple-and-flick, designed for objects with low and high centers of mass, respectively. Both methods enable a robot to retrieve objects resting on a surface and launch them into controlled projectile trajectories without requiring stable grasp formation. To support these maneuvers, we develop physics-based models of object acquisition and release, and combine them with a data-driven framework. While analytical modeling guides the acquisition phase, the highly nonlinear flicking dynamics are captured using learned predictive models that enable accurate selection of control parameters for desired trajectories. The proposed techniques enable dynamic object transfer, reduced grasp planning complexity, and adaptability to environmental constraints. Experiments conducted on a custom robotic platform demonstrate reliable and accurate high-arc object transfer, in which the majority of object displacement is achieved through projectile motion.

Keywords: grasping; dexterous manipulation; grippers; end-effectors

1. Introduction

This work focuses on establishing a high-speed framework for robotic pick-and-place operations. In particular, we seek to generate controlled projectile trajectories of objects through high-speed, non-prehensile contact interactions. As an initial step toward this objective, we introduce two dynamic manipulation strategies termed scoop-and-flick and topple-and-flick. As depicted in Figure 1, the robot first acquires an object resting on a supporting surface using a swift action, and then promptly transitions to a flicking action that propels the object toward a desired trajectory. The proposed methods offer several advantages. Unlike traditional grasp-and-throw pipelines that require grasp synthesis and stable grasp formation before release, our approach uses non-prehensile interactions to acquire and launch objects directly. This reduces planning complexity and enables faster pick-and-place operations.

To accommodate objects with different geometric and inertial characteristics, we introduce two complementary manipulation strategies. The scoop-and-flick approach targets objects with relatively low centers of mass, whereas the topple-and-flick approach is suited for objects with relatively high centers of mass. The key distinction between the two methods lies in how the object is initially acquired. In the scoop-and-flick strategy,

a robotic digit dynamically inserts beneath the object to lift and capture it. In contrast, the topple-and-flick strategy deliberately destabilizes the object, causing it to topple and settle onto the digit before being launched.

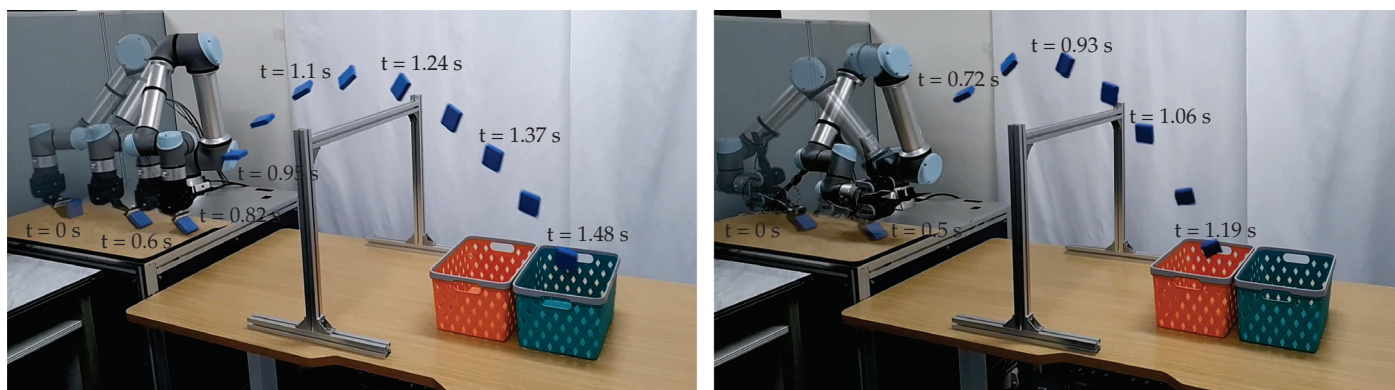


Figure 1. Our two robotic manipulation techniques for dynamic high-arc object transfer, shown in time-lapse: **(left)** one-fingered topple-and-flick and **(right)** two-fingered scoop-and-flick.

Although this manipulation approach enables rapid operation and reduces the need for complex grasp synthesis, it also presents control challenges due to the inherently dynamic and non-prehensile nature of the interaction. To overcome these difficulties, the proposed techniques combine analytical physical modeling with data-driven learning approaches. The methods can be realized using relatively simple hardware architectures and can effectively utilize environmental contacts during manipulation. The practicality of the approach is demonstrated through extensive experimental validation.

The remainder of this paper is structured as follows. Section 2 surveys relevant prior work. Section 3 introduces the conceptual manipulation models. Section 4 describes the system implementation. Section 5 presents the experimental evaluation.

2. Related Work

This presented work is situated within the area of dynamic manipulation, where dynamic physical interactions are deliberately exploited as part of the control policy rather than mitigated as undesired disturbances. Early demonstrations of this idea include object reorientation via controlled tilting of a tray [1] and some of the first robotic juggling systems [2]. These efforts were later followed by systematic treatments of planar dynamic manipulation that addressed both planning and control aspects [3], and by extensions to three-dimensional settings [4], showing that complex dynamic behaviors can emerge from relatively simple robotic mechanisms. To cope with the inherent modeling and computational difficulties, many studies have relied on simplified representations, such as the acrobot model [5]. In more recent years, dynamic manipulation methods have been applied to problems including within-hand object rearrangement [6] and the handling of large or heavy objects [7]. The importance of dynamics is similarly underscored in robotic locomotion research, particularly in investigations of passive dynamic walking [8], which frame bipedal locomotion as a largely self-stabilizing, gravity-driven dynamic process with minimal actuation.

Within the broader class of dynamic manipulation, robotic throwing is especially relevant to the work presented here. Owing to both its methodological complexity and its practical utility, throwing has been the subject of extensive investigation. Early studies emphasized hardware-centric solutions, such as high-speed manipulators developed specifically for ball throwing [9], dexterous planar robots capable of executing accurate throwing

motions [10], and hybrid grippers that integrate grasping and throwing functionalities [11]. Throwing behaviors have also been demonstrated on alternative actuation platforms, including soft robotic arms [12] and hydraulic manipulators [13], which exploit centrifugal forces to achieve high release velocities.

A central difficulty in robotic throwing is its high sensitivity to modeling inaccuracies, external disturbances, and uncertainty in the timing of object release, any of which can lead to large deviations in the final outcome. To address this sensitivity, prior studies have demonstrated that throwing motions can be deliberately shaped to improve robustness against release-time uncertainty [14]. Other approaches have leveraged tracking error prediction in conjunction with model predictive control to enhance execution precision [15]. In parallel, learning-based methods have gained traction as an alternative means of handling uncertainty in throwing tasks. Representative examples include data-driven systems that enable reliable grasp-then-throw behaviors for a wide variety of objects [16]. In contrast to such grasp-then-throw strategies, the proposed method does not require stable grasp formation and instead centers on impulse-driven acquisition through non-prehensile interactions. Other examples include collaborative learning formulations that jointly address grasping and throwing [17], methods for learning object-dependent tossing dynamics [18], and control strategies capable of achieving strong performance with relatively small training datasets [19].

3. Manipulation Models

This section presents robotic manipulation models that enable dynamic, rapid, non-prehensile object transfer along high-arc trajectories while accommodating environmental constraints.

3.1. Two-Fingered Scoop-and-Flick

The first manipulation model introduced in this study is the two-fingered scoop-and-flick technique, which enables a minimal two-fingered gripper to rapidly secure an object resting on an environmental surface and subsequently impart sufficient momentum to launch it into projectile motion. The method targets relatively thin objects, for which conventional grasping is challenging, and exploits brief, well-timed contact interactions between the fingers, the object, and the supporting surface.

The first stage of the manipulation is to secure the object within the workspace of the gripper. Figure 2a shows the initial configuration, in which the object rests on the ground defined as the xy -plane. One digit of the gripper providing a flat contact surface, referred to as the thumb, is also shown. We adopt the following assumptions:

- The object and the gripper, including the thumb, move exclusively within the xz -plane, which is perpendicular to the ground. This simplification is based on the observation that, once airborne, the object motion is largely confined to a vertical plane determined by the initial impulse and gravity.
- Any out-of- xz -plane motion of the object-gripper system is neglected and assumed to be effectively constrained.

Under this assumption, the manipulation task is modeled in the xz -plane and analyzed using rigid-body mechanics.

In Figure 2a, the object is about to be struck by the thumb moving in the x -direction with a sufficiently high velocity v_0 , such that the interaction is dominated by dynamic effects. In the reference frame fixed to the thumb, the object appears to approach the thumb's contact surface, as shown in the figure. When the object is struck by the thumb in Figure 2b, a substantial impulse $\mathbf{C}_{\text{avg}}\Delta t$ —defined as the product of the average contact force $\mathbf{C}_{\text{avg}}$ and the short impact duration Δt —is applied at the contact point C (the tip of

the thumb), while non-impulsive forces such as gravity $\mathbf{W}$ and ground reaction force $\mathbf{G}$ at the distal end of the object are neglected over the impact duration. This approximation follows the treatment of impulsive forces in classical mechanics [20], where forces that can be regarded as non-impulsive include the weight of a body, $\mathbf{W}$. In addition, the objects are placed on a compliant pad during the experiments, so that the ground reaction $\mathbf{G}$ behaves similarly to a spring force, which can likewise be regarded as non-impulsive [20].

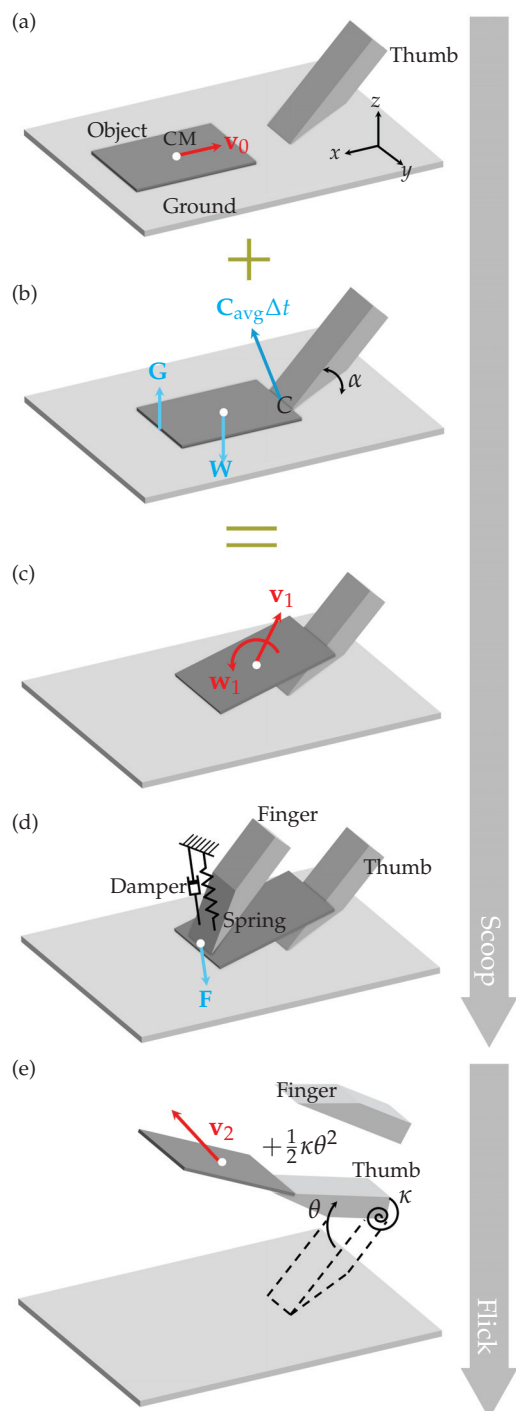


Figure 2. Two-fingered scoop-and-flick manipulation. (a–d) Scoop phase and (e) flick phase. (a) Pre-impact configuration with relative velocity $\mathbf{v}_0$; CM denotes the object center of mass. (b) Impulse $\mathbf{C}_{avg} \Delta t$ applied at C, with gravity $\mathbf{W}$ and ground reaction $\mathbf{G}$ treated as non-impulsive (shown in a lighter color). (c) Resulting post-impact linear and angular momenta of the object. (d) Low-impedance finger, shown as a mass-spring-damper system, for object capture. (e) Flicking via release of elastic energy (for example, by torsional spring stiffness κ), launching the object with velocity $\mathbf{v}_2$.

The post-impact motion of the object can be analyzed using conservation of angular momentum about the contact point C [21]. The total angular momentum about C is zero immediately before impact, since the initial velocity $\mathbf{v}_0$ is directed toward C , and remains zero immediately after impact, as the sole impulsive contact force $\mathbf{C}_{\text{avg}}$ acts through C . This constraint admits the post-impact motion illustrated in Figure 2c, in which a counterclockwise rotation $\mathbf{w}_1$ of the object is balanced by a clockwise angular momentum contribution about C from the translational motion $\mathbf{v}_1$ of the center of mass, yielding zero net angular momentum:

$$I\mathbf{w}_1 + \mathbf{r} \times m\mathbf{v}_1 = \mathbf{0} \quad (1)$$

where I is the object's moment of inertia, $\mathbf{r}$ is the vector to $\mathbf{v}_1$ originating from C , and m refers to the object's mass. As a result, the object moves past the thumb and enters the workspace of the gripper, as illustrated in Figure 2c. This behavior is facilitated by a sufficiently shallow angle of attack of the thumb, denoted α in Figure 2b, since a shallow approach allows the horizontal component of the impulsive force to be kept small.

We next consider a mechanism to securely retain the object following the thumb impact, namely post-strike caging by the gripper. As shown in Figure 2d, the second digit, referred to as the finger, is introduced to make contact with the top of the object. The finger is modeled as a mass–spring–damper system, representing a low-impedance interaction (i.e., small variations in contact force in response to motion disturbances [22]). When the mechanical impedance of the finger is sufficiently low, the resulting contact force $\mathbf{F}$ is non-impulsive and therefore does not alter the impact analysis described above. Consequently, dynamic thumb penetration remains feasible even in the presence of the additional low-impedance finger.

With both the thumb and finger engaged, the object's mobility within the workspace (the xz -plane) can be effectively constrained, resulting in a caging configuration similar to that described in [21]. Specifically, the object is caged in the following sense. First, escape through the gap between the finger and the ground is prohibited by first-order geometric constraints, assuming the finger and the thumb are fixed relative to the ground. Second, although escape through the gap between the finger and the thumb is kinematically possible, it requires the injection of a positive amount of work against gravity. As a result, the object remains effectively caged unless sufficient energy is supplied to overcome this gravitational barrier.

The stage following successful scooping is the launch into projectile motion. Once the object is securely caged, it is propelled into a ballistic trajectory by executing a precisely timed flick of the thumb within the workspace—the xz -plane. Prior to the thumb flick, the finger is retracted to avoid interference with the object and the thumb during release. This separation is essential to ensure that the subsequent motion is governed solely by the interaction between the thumb and the object. The flicking action is then executed by the thumb and can be realized by exploiting elastic energy stored in a compliant element, analogous to a slingshot mechanism. As shown in Figure 2e, a torsional spring can be used to accumulate elastic energy of $\frac{1}{2}\kappa\theta^2$, which is rapidly released, with a portion converted into the kinetic energy of the object at launch, with velocity $\mathbf{v}_2$.

Overall, the scoop-and-flick maneuver is inherently dynamic, with object motion governed by impulsive and elastic energy transfer rather than quasi-static force regulation. Successful execution therefore requires a high-speed manipulation platform with sufficient compliance to store and release energy over short time scales. These requirements motivate the gripper design to be described in Section 4.1.

3.2. Topple-and-Flick

The scoop-and-flick strategy described above primarily target relatively thin objects, for which dynamic scooping enables rapid entry of the thumb into the space underneath the object's bottom face. We now consider a complementary technique designed for taller objects, whose height-to-width ratio makes scoop-based acquisition ineffective. This method, referred to as topple-and-flick, exploits deliberate object toppling induced by an impulsive contact, followed by a flicking action to launch the object into projectile motion.

Similarly to the two-fingered scoop-and-flick technique, the manipulation is initiated by striking the object with the thumb, as illustrated in Figure 3. When the object's center of mass is positioned sufficiently high relative to the contact point C , the impulsive contact generates a clockwise rotation of the object, in contrast to the counterclockwise rotation observed for thin objects with a low center of mass. As a result, the object topples and falls onto the flat contact surface of the thumb.

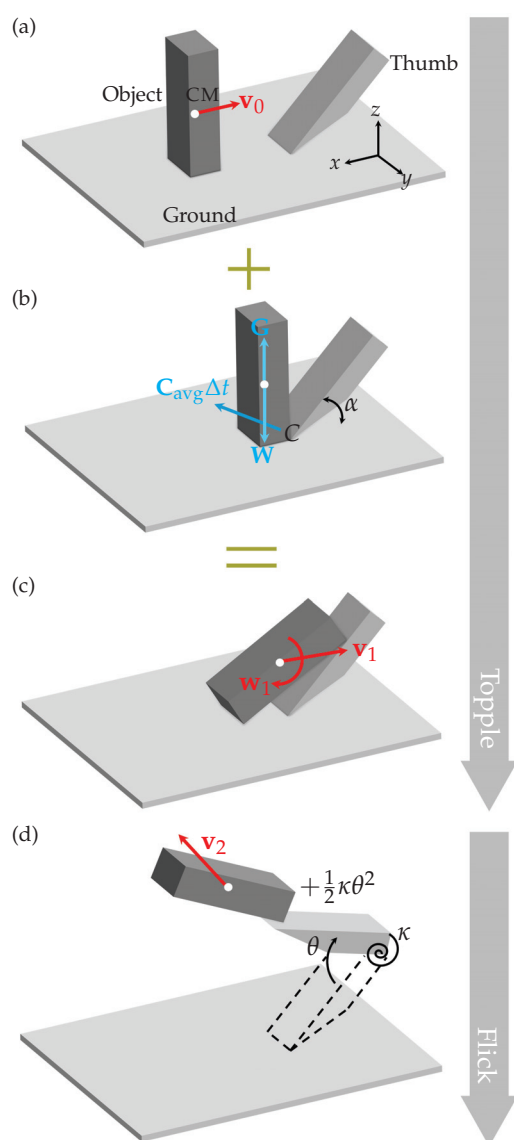


Figure 3. Topple-and-flick manipulation. (a–c) Topple phase and (d) flick phase. (a) Pre-impact configuration with relative velocity v_0 ; CM denotes the object center of mass. (b) Impulse $C_{avg}\Delta t$ applied at C , with gravity W and ground reaction G treated as non-impulsive (shown in a lighter color). (c) Resulting post-impact linear and angular momenta of the object. (d) Flicking.

This behavior can also be understood through conservation of angular momentum about the contact point C. Immediately before impact (Figure 3a), the object possesses a clockwise angular momentum about C, as its center of mass approaches the thumb. Since the impulsive contact force acts through C (Figure 3b), the angular momentum about this point is conserved during impact. Consequently, the post-impact motion must also exhibit clockwise angular momentum, which manifests as a toppling motion toward the thumb (Figure 3c). This analysis highlights the range of applicability of the topple-and-flick strategy: The object's center of mass must be sufficiently elevated to induce clockwise rotation.

Following successful toppling, the final stage mirrors that of the previous methods. Once the object has fallen onto and is supported by the thumb, it is launched into projectile motion by executing a flicking action (Figure 3d). Notably, the entire process can be executed using only a single digit.

4. Implementation

This section presents the hardware and software systems developed to implement the proposed manipulation models for high-arc transfer.

4.1. Hardware System

The hardware platform used in this study consists of a conventional robot arm equipped with a custom gripper featuring up to two digits (Figure 4a,b). For the two-fingered scoop-and-flick experiments, the gripper is configured with two digits, consisting of a thumb and a finger (Figure 4a). For the topple-and-flick experiments, a single-digit configuration is used (Figure 4b), in which the thumb alone executes the dynamic interaction with the object.

The gripper design follows key principles dictated by the manipulation models described in Section 3, as summarized below:

- **High-speed motion:** Rapid digit motion is essential to generate the large contact impulses required for scoop-and-flick and topple-and-flick manipulation. High peak velocities and accelerations enable momentum transfer over short contact durations, ensuring that impact dynamics dominate over non-impulsive effects such as gravity and ground reaction forces, as assumed in the models of Section 3.
- **Low-impedance actuation:** Conventional motion-controlled robotic fingers with high mechanical impedance are not well suited for the dynamic energy exchange process inherent to the manipulation models described in Section 3. In contrast, a low-impedance mechanism, characterized by compliance and underactuation, allows the finger to naturally transmit energy and respond to the object's motion without requiring explicit control intervention. The passive mechanical behavior also assists in safely negotiating physical interactions and contact events with the ground, and reduces the need for precise force modulation.

We adopt direct-drive electric motor actuation to satisfy these design requirements. As shown in Figure 4c, each digit is implemented as a five-bar closed-loop linkage with two degrees of freedom, and is therefore actuated by two motors co-axially located at the base of the linkage. No gear train is used, enabling high-speed operation while allowing for the mechanical impedance to be tuned.

A wide thumb is employed to provide a stable and sufficiently large contact surface for impulsive interactions during scooping, toppling, and flicking, as shown in Figure 4d. In addition, a wall is incorporated to prevent the object from translating backward as the thumb and robot advance forward. This wall also integrates infrared (IR) proximity sensors (Figure 4d) that detect whether the object has been successfully positioned at the corner

of the wall on the thumb's contact surface. Specifically, if the object is detected within the sensor's effective range of 4–11 mm from the wall, the scooping (and toppling) action is considered successful, and the system proceeds to the subsequent flicking phase. The sensor is covered with a 7 mm-thick sponge layer, whose thickness is chosen to match the sensor's detection range; this layer additionally serves as a passive damper that mitigates impact forces on the object during contact with the wall.

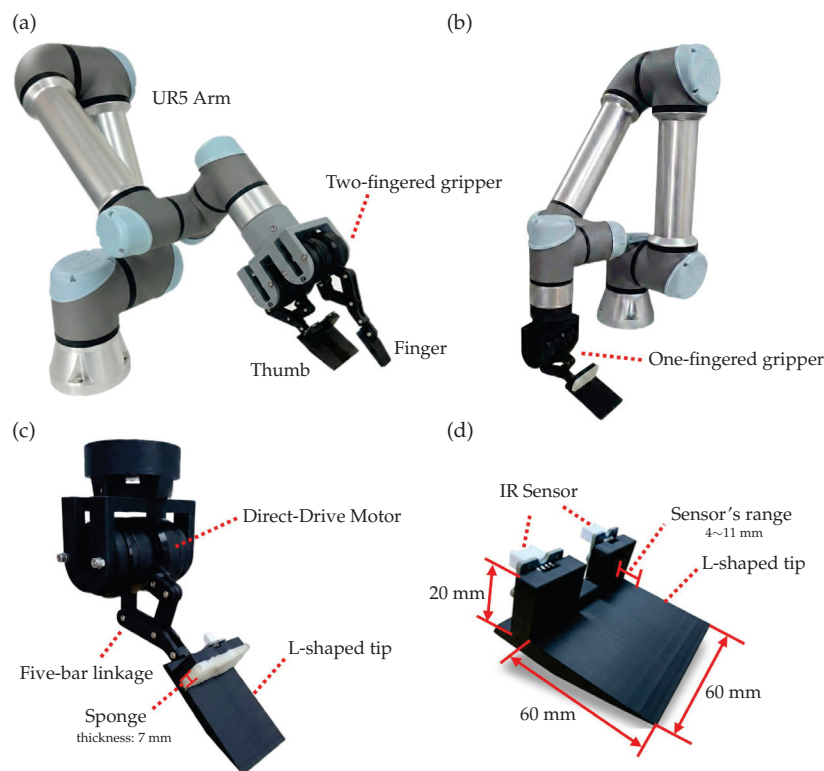


Figure 4. Hardware system. (a) Two-fingered configuration for the scoop-and-flick maneuver. (b) One-fingered configuration for the topple-and-flick maneuver. (c) Digit implemented as a two-degree-of-freedom five-bar closed linkage. (d) Thumb tip featuring a wide contact geometry and an instrumented wall with IR proximity sensors. A notch is cut around the center of the wall to prevent collisions with the other digit.

4.2. Software System for Flicking Control: Data Collection

The central component of the software system is the control strategy for flicking. Flicking involves highly transient and nonlinear dynamics, including complex frictional moment exchange between the robot and the object, which are difficult to model accurately using first principles. Rather than attempting to explicitly derive and regulate these dynamics, we pursue a data-driven approach to obtain a holistic understanding. This perspective enables flicking motions to be executed in an open-loop manner, despite the underlying complexity of the contact interactions.

We collected experimental data to characterize the dynamic flicking process illustrated in Figures 2e and 3d. The configuration of the two-DOF digit (Figure 4c) is parameterized by two variables, as shown in Figure 5a: α , which denotes the fingertip angle of attack relative to the ground surface, and δ , which represents the relative orientation between the two internal links. The digit is operated by varying its initial and final configurations, and the resulting object trajectories are recorded.

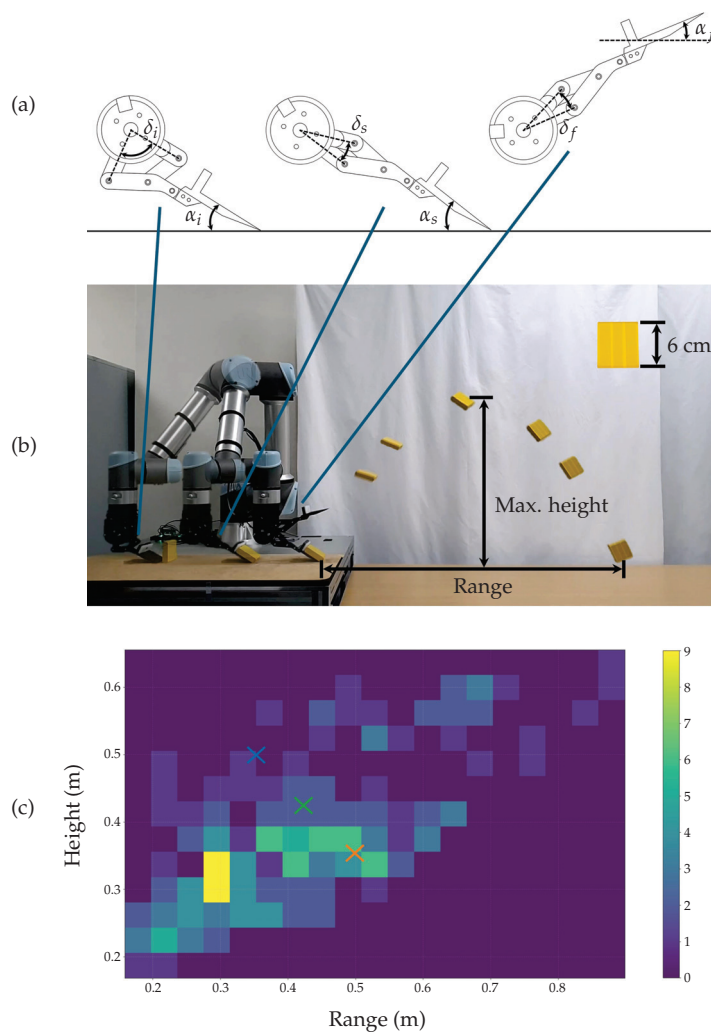


Figure 5. (a) The initial, intermediate, and final configurations of the thumb during the flick phase: (α_i, δ_i) , (α_s, δ_s) , and (α_f, δ_f) , respectively. (b) Topple-and-flick maneuver. (c) Distribution of the 200 recorded data samples, where color represents the frequency of occurrences. The $\times$ symbols indicate the query points used for generating the learning curves. These query points are selected from regions with high–short arc (blue), mid–mid arc (green), and low–long arc (orange).

Specifically, topple-and-flick scenarios are considered (Figure 5b). Note that the focus is on the flick phase; therefore, scoop-and-flick could also be employed. A tall 3D-printed object, with an essentially even mass distribution, is initially positioned on a flat ground surface, after which a topple-and-flick maneuver is executed. Throughout the maneuver, the tool center point (TCP) at the arm wrist is commanded to move horizontally forward at a constant speed of 0.1 m/s. The thumb begins at an initial configuration (α_i, δ_i) and is driven to a final configuration (α_f, δ_f) using PD setpoint motion control, during which the object is propelled away from the contact region. The commanded motion of the thumb passes through an intermediate configuration (α_s, δ_s) that extends the digit to facilitate more effective object acquisition—scooping and toppling. Among the configuration parameters, the initial and final angles of attack, α_i and α_f , are varied. Additional trials are conducted using objects of the same geometry but with different masses, while all other parameters remain unchanged. The specific parameter values used in the experiments are summarized as follows:

- Object mass m_{obj} : 14, 24, 33, 43 g (the maximum value constrained by the electric motor capacity)
- δ_i : 83° (fixed).

- α_i : 20° (minimum) to 40° (maximum).
- δ_s : 25° (fixed).
- α_s : Determined such that the fingertip remains in contact with the ground at the value of δ_s . Accordingly, it varies between 29.7° (minimum) and 43.3° (maximum), dependently on δ_s and α_i .
- δ_f : 25° (fixed).
- α_f : from −10° (minimum) to 5° (maximum). At negative values, the fingertip points upward.
- P-gain: Set to 12 initially and increased to 48 past the intermediate configuration (α_s, δ_s). The value refers to the ratio between motor torque and shaft position error.
- D-gain: Set to 0.2 initially and increased to 0.6 past the intermediate configuration (α_s, δ_s). The value represents the ratio between motor torque and shaft angular velocity error.

The resulting object trajectories are captured using the video analysis software Tracker. Each collected data sample is represented as $(m_{\text{obj}}, \alpha_i, \alpha_f, h, r)$, where h and r correspond to the measured maximum height and horizontal range of the resulting trajectory, respectively. The parameters α_i , α_f , and m_{obj} represent the initial and final angles of attack and the object mass described previously. To isolate the effect of the thumb-driven flicking motion, the additional horizontal displacement introduced by the arm motion (0.1 m/s) is subtracted from the measured range. The aggregated dataset is summarized in Figure 5c, which presents the results in terms of the achieved range r and maximum height h of the flicked objects.

4.3. Software System for Flicking Control: Learning

From the collected data $\{(m_{\text{obj}}, \alpha_i, \alpha_f, h, r)\}$, we seek to estimate a function $\mathbf{f}$,

$$\mathbf{f} : (m_{\text{obj}}, h_{\text{des}}, r_{\text{des}}) \mapsto (\alpha_i^*, \alpha_f^*), \quad (2)$$

which maps an input three-tuple $(m_{\text{obj}}, h_{\text{des}}, r_{\text{des}})$ —representing the object to be handled and a target trajectory specified by the desired maximum height h_{des} and range r_{des} —to the optimal finger control parameters (α_i^*, α_f^*) . During analysis of the collected data, we observed that multiple distinct control parameter pairs (α_i, α_f) can yield object trajectories (h, r) that are nearly indistinguishable from one another. This nonuniqueness makes it difficult to approximate $\mathbf{f}$ directly, as direct regression tends to average multiple valid solutions, resulting in estimates that fail to achieve the desired trajectory. Consequently, instead of directly learning $\mathbf{f}$, which we refer to as the inverse model, we first train a forward model $\mathbf{g}$,

$$\mathbf{g} : (m_{\text{obj}}, \alpha_i, \alpha_f) \mapsto (h, r), \quad (3)$$

using the collected data, and then recover $\mathbf{f}$ through optimization. Specifically, the forward model $\mathbf{g}$ is trained using Gaussian Process Regression (GPR) [23]. Given a desired trajectory $(h_{\text{des}}, r_{\text{des}})$, the optimal control parameters (α_i^*, α_f^*) are subsequently obtained by solving a constrained optimization problem using the L-BFGS-B algorithm [24]. The optimization minimizes the loss function $\mathcal{L}$,

$$\mathcal{L} = |h_{\text{pred}} - h_{\text{des}}| + |r_{\text{pred}} - r_{\text{des}}|, \quad (4)$$

which is defined as the sum of absolute errors between the desired trajectory $(h_{\text{des}}, r_{\text{des}})$ and the predicted trajectory $(h_{\text{pred}}, r_{\text{pred}})$ produced by the forward model.

Figure 6 presents the learning curves, illustrating the prediction performance of the learned mapping $\mathbf{f}$ in terms of the maximum height and range of the resulting projec-

tile motion. Each plot contains three curves corresponding to queries selected from the high–short, mid–mid, and low–long arc regions of the histogram shown in Figure 5c. The height and range errors are defined as the mean absolute error computed over five trials using an object whose mass was not included in the training dataset. The results show a rapid initial reduction in error for all three queries, with performance gradually saturating at approximately 100 training samples. Overall, the prediction errors in maximum height are smaller than those in range. For comparison, we also evaluated an alternative learning approach based on a three-layer neural network following [12]. In this case, the convergence was slower, with the prediction performance saturating after approximately 200 training samples.

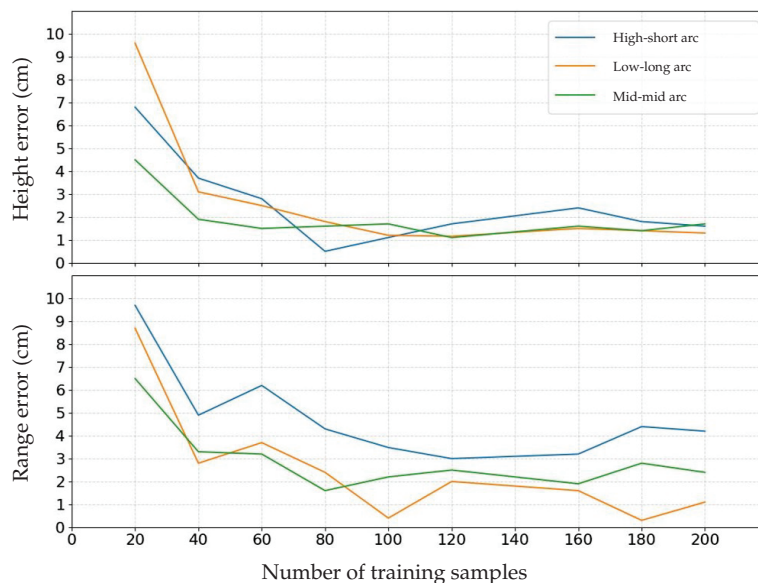


Figure 6. Learning curves for maximum height errors (**top**) and range errors (**bottom**). The colors—blue, green, and orange—correspond to the selected query points indicated in Figure 5.

5. Experiments

This section presents a series of experiments to demonstrate the manipulation models for high-arc object transfer.

5.1. End-to-End Control Protocol

The end-to-end control protocol used in the experiments, including the flicking strategy described in Section 4.3, is implemented as an open-loop control scheme. This design choice is motivated by the difficulty of reliably detecting and interpreting contact events in the presence of compliance, friction, and rapid contact transitions such as sliding and separation, as encountered in the tasks considered in this study. Rather than relying on complex sensing and feedback, the motion profile is predefined and executed in a feedforward manner. Interaction forces are instead regulated implicitly through the mechanical properties of the system, eliminating the need for explicit contact perception or contact-state estimation.

Specifically, the two techniques—two-fingered scoop-and-flick and topple-and-flick—are executed as follows. Since the topple-and-flick technique is simpler, it is described first:

- **Topple-and-flick:**

1. Initially, the thumb is positioned in front of the target object, as shown in Figure 7a. At this stage, the thumb is set to the configuration (α_i, δ_i) , as specified in Section 4.2.

2. As shown in Figure 7b, the arm is commanded to move horizontally forward, while the thumb is simultaneously extended to strike the object, reaching the configuration (α_s, δ_s) . This interaction causes the object to topple and land on the thumb's tip.
3. When the thumb's tip reaches a predefined horizontal position, the flick command is triggered to reconfigure the thumb to the final configuration (α_f, δ_f) . Figure 7c illustrates this moment. The tip position is tracked at a frequency of 500 Hz by combining the TCP position at the arm wrist with the thumb's forward kinematics. Based on this tracking information, the flick command is issued at essentially the same spatial location across trials.
4. After the flick command is issued, the object is launched (Figure 7d). The arm follows through by maintaining the commanded horizontal velocity for a finite duration of approximately 0.2 s, after which deceleration is initiated.

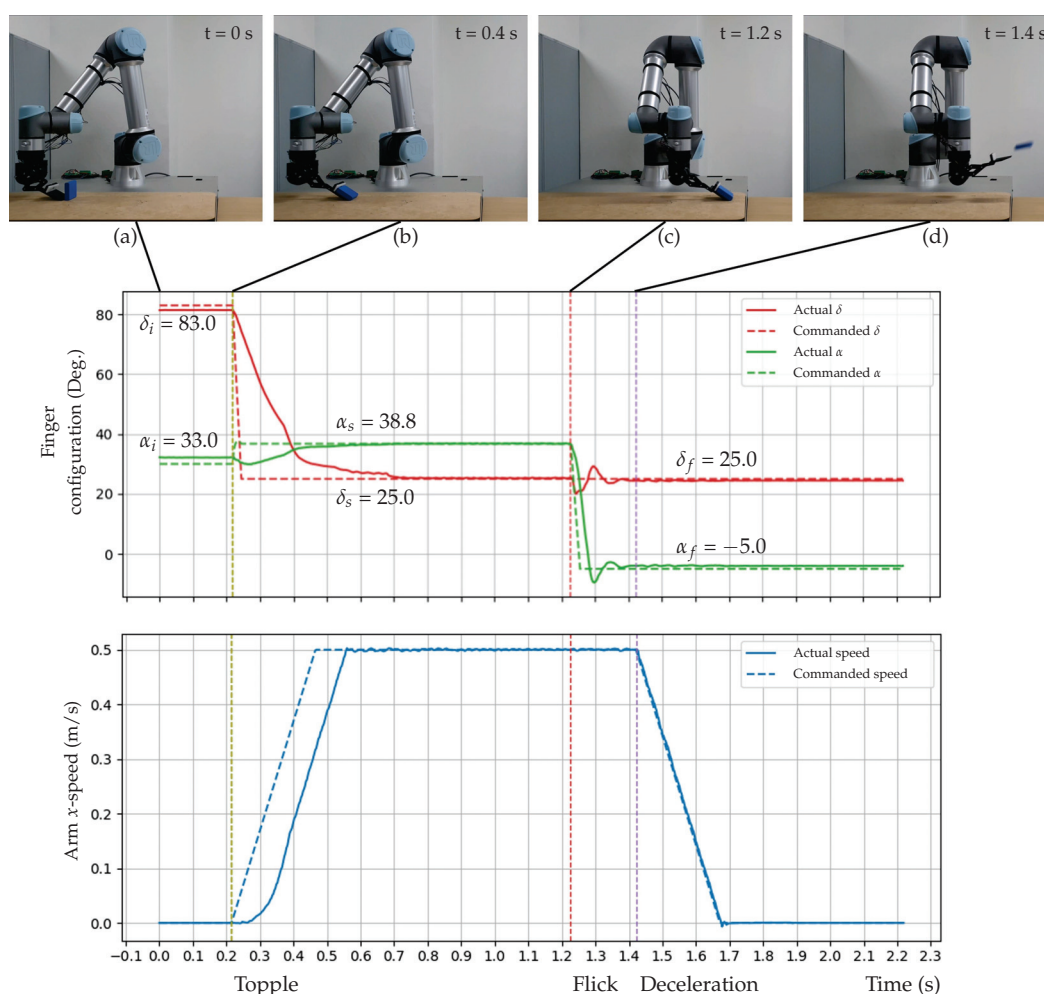


Figure 7. Thumb configuration (α, δ) and arm speed profiles during the execution of a topple-and-flick maneuver: (a) The thumb is set to the initial configuration (α_i, δ_i) . (b) The arm moves forward while the thumb extends to the intermediate configuration (α_s, δ_s) . (c) When the thumb's tip reaches a predefined position, the thumb is reconfigured to the final configuration (α_f, δ_f) . (d) The arm follows through, after which deceleration is initiated.

- **Scoop-and-flick:** The manipulation is initiated by scooping the object with two digits. The arm is first commanded to move vertically downward, while the two-fingered gripper performs a coordinated scooping motion to secure the object within the workspace (Figure 8a). The corresponding control parameters of the gripper are

selected according to the method described in [21]. After successful scooping, the finger is opened to provide sufficient clearance for the object's free flight. The object is then launched by executing a flicking action, which is performed in the same manner as described above for the topple-and-flick technique (Figure 8b).

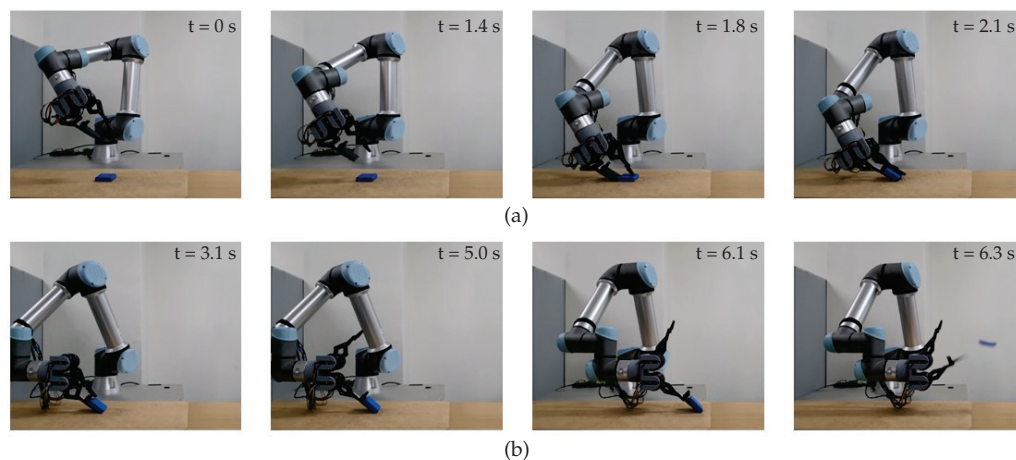


Figure 8. Two-fingered scoop-and-flick: (a) the scoop phase and (b) the flick phase.

5.2. End-to-End Experiments

To verify the soundness of the integrated hardware–software system, we conducted end-to-end experiments. First, with the arm speed fixed at 0.1 m/s, we evaluated the performance of the two-fingered scoop-and-flick maneuver (Figure 8). As an illustrative example, consider a flick trial commanded to achieve a horizontal range of 30 cm and a maximum height of 30 cm, corresponding to a nominal flight duration of 0.495 s. When incorporating the horizontal arm motion at 0.1 m/s, the final range and maximum height are predicted to be approximately 35 cm and 30 cm, respectively. The measured range differed from the prediction by 2.1 cm, corresponding to 6% of the predicted value.

Table 1 summarizes the results across all trials, conducted with varying flick commands and arm speeds. Among these trials, the 6% error reported above represents the worst case deviation. These results indicate that the system is capable of achieving sufficient accuracy for practical manipulation tasks. Moreover, since the two-fingered scoop-and-flick and topple-and-flick techniques share an identical flick phase, the demonstrated performance extends to both manipulation strategies.

Table 1. End-to-end experimental results for two-fingered scoop-and-flick.

Arm Speed (m/s)	Flight Duration (s)	Predicted Height (cm)	Predicted Range (cm)	Height Error (cm)	Range Error (cm)
0.1	0.495	30.0	35.0	0.5	2.1
0.1	0.571	40.0	45.7	2.0	1.7
0.1	0.639	50.0	56.9	1.1	0.1
0.1	0.571	40.0	45.7	2.0	1.7
0.2	0.571	40.0	51.4	0.8	2.4
0.4	0.571	40.0	62.8	1.6	2.3
0.8	0.571	40.0	85.7	1.8	1.3

Each experiment was conducted five times. The error values represent the average.

5.3. Application Experiments: Bucket Delivery

Encouraged by the strong end-to-end experimental performance, we further examined an application task: bucket delivery (Figure 9a)—transferring objects into target buckets located at horizontal ranges of 100 cm and 120 cm while clearing a horizontal bar set at 40 cm using the two proposed techniques. The evaluation employed both 3D-printed and everyday objects. The 3D-printed items differed from those used in training in terms of mass and geometry, while the real-world objects included a battery case and a carton box. To facilitate controlled evaluation, prism-shaped objects were primarily adopted. In particular, a single object can be configured in two ways, resting on different faces, to produce either a relatively low or high center of mass, corresponding to the two manipulation strategies presented in this study. Note that the presented manipulation models in Section 3 are primarily based on the relative height of the object's center of mass and are therefore not constrained by specific global object shapes.

The finger control parameters (α_i, α_f) for the flick phase and the arm's horizontal velocity v_{arm} during execution of the maneuver (see Figure 7), which are common to both techniques, are determined as follows. First, the finger parameters (α_i, α_f) are selected using f in Equation (2) to generate a trajectory whose peak height is sufficient to clear the horizontal bar, set at 40 cm (Figure 9a). Second, based on the resulting flight duration, the arm velocity v_{arm} is computed such that the object's landing position aligns with the center of the buckets, located at horizontal ranges of 100 cm and 120 cm (Figure 9a).

5.3.1. Topple-and-Flick Results

A bucket delivery trial is considered successful only if the object clears the bar and lands in one of the target buckets. The results are summarized in Table 2 and illustrated in Figure 9. The findings are as follows:

- Compared with the results obtained using an object included in the training set in Section 4.2 (Table 2, column 1), similar 3D-printed objects with different masses (column 2) and increased thicknesses (column 3) exhibit comparably high success rates.
- These favorable outcomes generalize to real-world objects, namely a carton box (column 4) and a battery case (column 5), which is likely attributable to their geometric and inertial similarity to the objects used during training. In the failed trials involving the carton box, the object consistently overshot the target bucket, suggesting a systematic bias in the predicted range for that object.
- The method also performs well with non-cuboid objects, as shown in columns 6–8. This suggests that the limited variation in training variables such as geometry is not a critical limitation in the current framework, since the training objects are abstracted primarily by their mass values (Equation (2)). In the cases shown in columns 6 and 7, contact between the object and the fingertip wall may occur at a point rather than along an edge, as is typical for cuboid-shaped objects (or at two contact points in column 8). Although such contact configurations could be inherently less stable, successful transfers were still achieved, which may be attributed to the stabilizing effect of the fingertip wall's compliance.
- Additionally, an object with a conspicuously uneven mass distribution (column 9) and an object with substantially different friction properties (column 10) proved ineffective with the current implementation. This outcome can be attributed to the fact that mass distribution and friction are not explicitly considered in the current control learning framework.

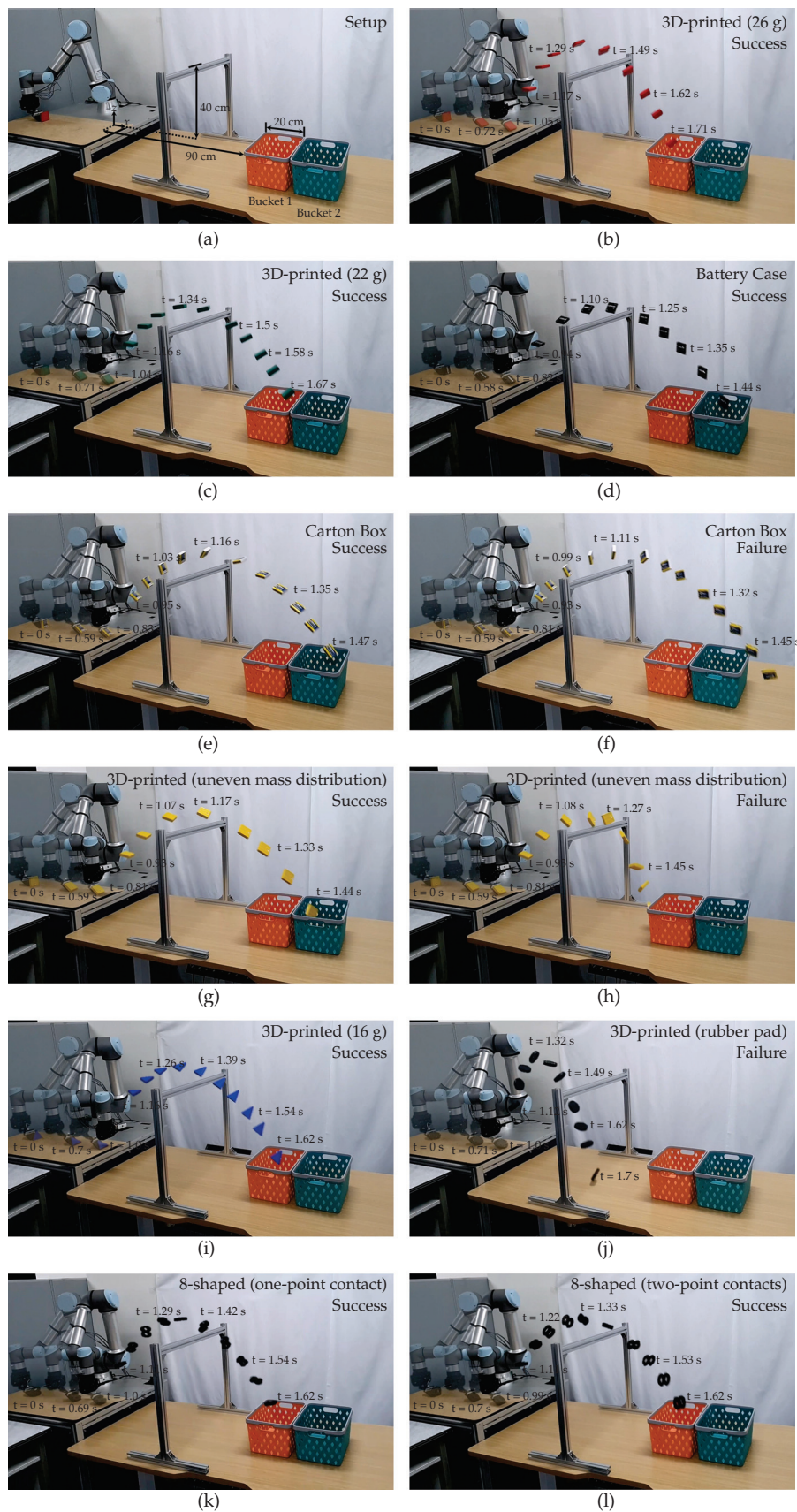


Figure 9. Bucket delivery experiments using topper-and-flick, shown in time-lapse. Initially, the objects are placed on the ground with a high center of mass. (a) experimental setup; (b,c,i) 3D-printed objects with different masses, successful transfers; (d–f) real-world objects (battery case and carton box); (g,h) objects with uneven mass distribution; (j) object with rubber pad, failure; (k,l) 8-shaped objects with one- and two-point contacts.

Table 2. Results of bucket delivery experiments with topple-and-flick.

Test Objects	3D-Printed (Used for Training)	3D-Printed (Not Used for Training)	3D-Printed (Not Used for Training)	Carton Box	Battery Case	3D-Printed (Not Used for Training)	8-Shaped (One-Point Contact)	8-Shaped (Two-Point Contacts)	3D-Printed (With Uneven Mass Distribution ¹)	Cylinder (With Rubber Pad)
Dimensions (Length × Thickness)	 76 × 15 mm	 76 × 15 mm	 76 × 20 mm	 66 × 14 mm	 74 × 13 mm	 77 × 15 mm	 66 × 15 mm	 66 × 15 mm	 76 × 15 mm	 66 × 14 mm
Mass (g)	24	26	22	20	22	16	17	17	24	23
Success rate ² (Bucket 1)	10/10	10/10	9/10	8/10	9/10	8/10	9/10	9/10	4/10	0/10
Success rate ² (Bucket 2)	10/10	10/10	10/10	9/10	9/10	9/10	9/10	10/10	3/10	0/10

¹ Point mass (nut) attached off-center. ² Ratio of successful trials to the total number of attempts.

5.3.2. Two-Fingered Scoop-and-Flick Results

We also evaluated bucket delivery using two-fingered scoop-and-flick with objects configured to have a low center of mass, specifically the same objects resting on their wide faces. The flick phase was executed using the same arm speed and flick parameters as in the topple-and-flick trials. The results are summarized in Table 3 and illustrated in Figure 10. Overall, similar performance trends were observed, although the success rate was slightly lower. This reduction is mainly attributed to the greater difficulty of achieving consistent scooping compared to toppling, which occasionally caused variations in object configuration prior to flicking. Objects with uneven mass distribution and different friction properties (Table 3, columns 9 and 10) remained particularly challenging, resulting in collisions with the bar, insufficient launch height to clear it, or shortened range preventing the object from reaching the target bucket.

Table 3. Results of bucket delivery experiments with scoop-and-flick.

Test Objects	3D-Printed (Used for Training)	3D-Printed (Not Used for Training)	3D-Printed (Not Used for Training)	Carton Box	Battery Case	3D-Printed (Not Used for Training)	8-Shaped (One-Point Contact)	8-Shaped (Two-Point Contacts)	3D-Printed (With Uneven Mass Distribution ¹)	Cylinder (With Rubber Pad)
Dimensions (Length × Thickness)	 76 × 15 mm	 76 × 15 mm	 76 × 20 mm	 66 × 14 mm	 74 × 13 mm	 77 × 15 mm	 66 × 15 mm	 66 × 15 mm	 76 × 15 mm	 66 × 14 mm
Mass (g)	24	26	22	20	22	16	17	17	24	23
Success rate ² (Bucket 1)	8/10	9/10	8/10	9/10	9/10	9/10	9/10	8/10	4/10	0/10
Success rate ² (Bucket 2)	10/10	10/10	10/10	9/10	9/10	9/10	9/10	10/10	4/10	0/10

¹ Point mass (nut) attached off-center. ² Ratio of successful trials to the total number of attempts.

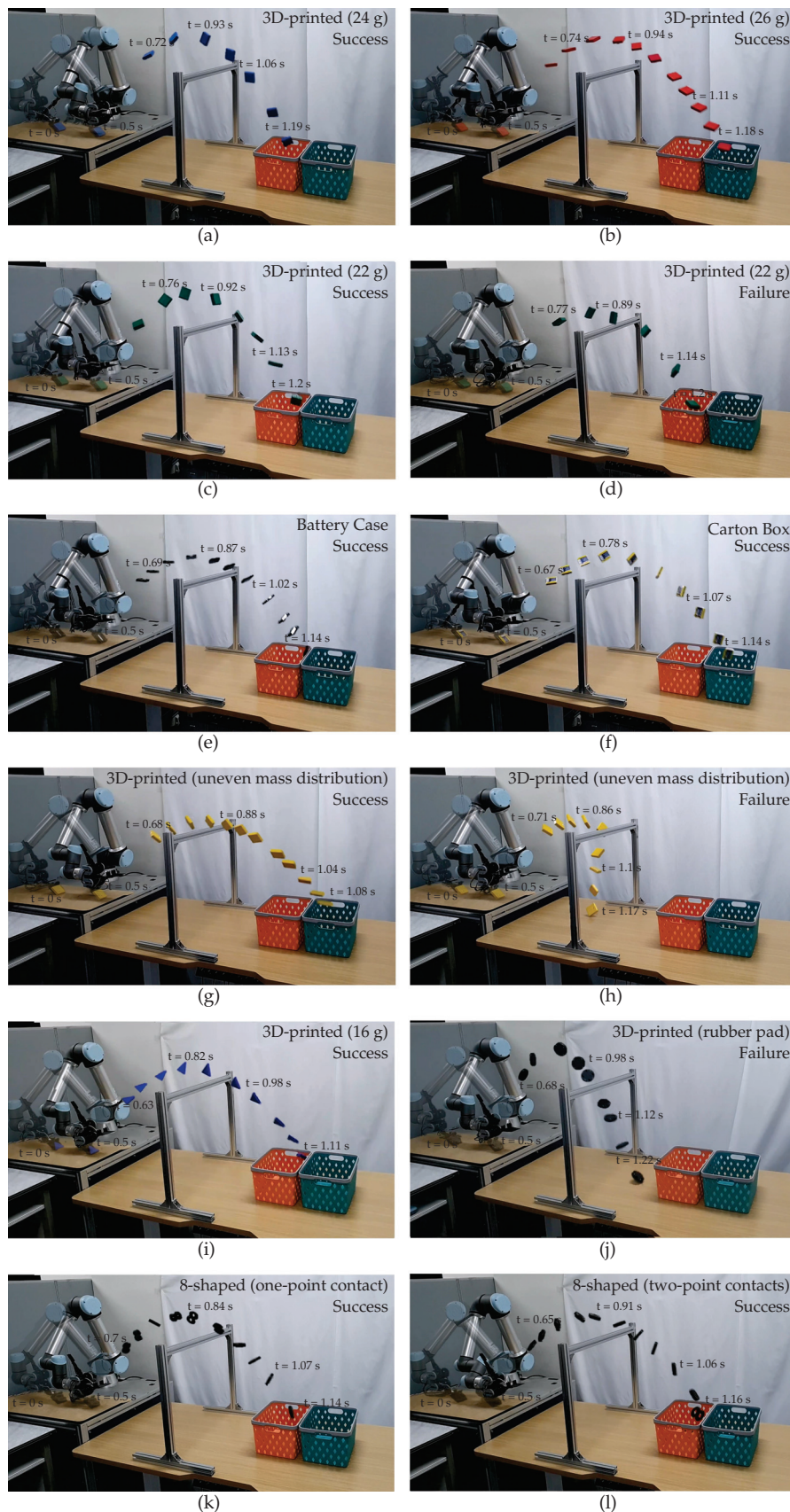


Figure 10. Bucket delivery experiments using scoop-and-flick, shown in time-lapse. Initially, the objects are placed on the ground with a low center of mass. (a–c,i) 3D-printed objects (16–26 g), successful transfers; (d,h,j) failure cases; (e,f) real-world objects (battery case and carton box), successful transfers; (g,h) objects with uneven mass distribution; (k,l) 8-shaped objects with one- and two-point contacts.

6. Conclusions

This paper presents robotic manipulation techniques for dynamic high-arc object transfer using non-prehensile physical interactions. Two complementary methods, two-fingered scoop-and-flick and one-fingered topple-and-flick, were introduced to handle objects with low and high centers of mass, respectively. By exploiting impulsive contact interactions, the proposed techniques enable robots to rapidly acquire objects resting on environmental surfaces and transfer them through controlled projectile motion without requiring stable grasp formation.

To support these maneuvers, we developed physics-based models describing object acquisition mechanisms and combined them with a data-driven framework for flicking control. The hybrid formulation addresses the difficulty of analytically modeling highly transient and nonlinear contact dynamics while enabling reliable selection of control parameters for desired object trajectories. The proposed methods were implemented on a custom robotic platform featuring high-speed, low-impedance digits.

Extensive experiments demonstrated accurate and repeatable high-arc object transfer across objects with varying geometries and masses. The results further show that the learned flicking model generalizes to unseen objects and supports application-level tasks, such as obstacle-clearing bucket delivery. These findings highlight the potential of dynamic non-prehensile manipulation as a fast and efficient alternative to conventional grasp-based pick-and-place operations.

Future work will focus on extending the approach to broader object classes with complex mass distributions and varied friction properties by incorporating perception-guided estimation of object properties and closed-loop adaptation during contact phases. Extending the framework to explicitly account for three-dimensional dynamics and trajectory control is another interesting avenue for future work. More sophisticated arm swing motions may also be incorporated, where guaranteed tracking performance under disturbances [25] will be essential.

Author Contributions: Conceptualization, J.S.; methodology, J.L. and S.O.; software, J.L. and S.O.; validation, J.L. and S.O.; formal analysis, J.S.; investigation, J.L. and S.O.; resources, J.S.; data curation, J.L. and S.O.; writing—original draft preparation, J.S.; writing—review and editing, J.L., S.O. and J.S.; visualization, J.L. and S.O.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the BK21FOUR program, Creative HR Education and Research Programs for ICT Convergence in the 4th Industrial Revolution.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Erdmann, M.A.; Mason, M.T. An exploration of sensorless manipulation. *IEEE J. Robot. Autom.* **1988**, *4*, 369–379. [CrossRef]
2. Rizzi, A.; Koditschek, D. Progress in spatial robot juggling. In Proceedings of the 1992 IEEE International Conference on Robotics and Automation (ICRA), Nice, France, 12–14 May 1992; pp. 775–780. [CrossRef]
3. Lynch, K.M.; Mason, M.T. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *Int. J. Robot. Res.* **1999**, *18*, 64–92. [CrossRef]
4. Woodruff, J.Z.; Lynch, K.M. Robotic Contact Juggling. *IEEE Trans. Robot.* **2023**, *39*, 1964–1981. [CrossRef]

5. Donner, P.; Buss, M. Cooperative Swinging of Complex Pendulum-Like Objects: Experimental Evaluation. *IEEE Trans. Robot.* **2016**, *32*, 744–753. [CrossRef]
6. Shi, J.; Woodruff, J.Z.; Umbanhowar, P.B.; Lynch, K.M. Dynamic In-Hand Sliding Manipulation. *IEEE Trans. Robot.* **2017**, *33*, 778–795. [CrossRef]
7. Nazir, A.; Xu, P.; Seo, J. Rock-and-Walk Manipulation: Object Locomotion by Passive Rolling Dynamics and Periodic Active Control. *IEEE Trans. Robot.* **2022**, *38*, 2354–2369. [CrossRef]
8. McGeer, T. Passive dynamic walking. *Int. J. Robot. Res.* **1990**, *9*, 62–82. [CrossRef]
9. Senoo, T.; Namiki, A.; Ishikawa, M. High-speed throwing motion based on kinetic chain approach. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3206–3211. [CrossRef]
10. Mori, W.; Ueda, J.; Ogasawara, T. 1-DOF dynamic pitching robot that independently controls velocity, Angular velocity, and direction of a ball: Contact models and motion planning. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 1655–1661. [CrossRef]
11. Govindan, N.; Ramachandran, B.; Sai, P.H.V.; Krishna, K.M. A Novel Hybrid Gripper Capable of Grasping and Throwing Manipulation. *IEEE/ASME Trans. Mechat.* **2023**, *28*, 3317–3328. [CrossRef]
12. Bianchi, D.; Campinoti, G.; Comitini, C.; Laschi, C.; Rizzo, A.; Sabatini, A.M.; Falotico, E. SoftSling: A Soft Robotic Arm Control Strategy to Throw Objects with Circular Run-Ups. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8250–8257. [CrossRef]
13. Werner, L.; Nan, F.; Eyschen, P.; Spinelli, F.A.; Yang, H.; Hutter, M. Dynamic Throwing with Robotic Material Handling Machines. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 98–104. [CrossRef]
14. Liu, Y.; Billard, A. Tube Acceleration: Robust Dexterous Throwing Against Release Uncertainty. *IEEE Trans. Robot.* **2024**, *40*, 2831–2849. [CrossRef]
15. Takahashi, A.; Sato, M.; Namiki, A. Dynamic Compensation in Throwing Motion with High-Speed Robot Hand-Arm. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 6287–6292. [CrossRef]
16. Zeng, A.; Song, S.; Lee, J.; Rodriguez, A.; Funkhouser, T. TossingBot: Learning to Throw Arbitrary Objects with Residual Physics. *IEEE Trans. Robot.* **2020**, *36*, 1307–1319. [CrossRef]
17. Hou, Y.; Fang, Z.; Li, J. Self-Supervised Multi-Modal Learning for Collaborative Robotic Grasp-Throw. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4250–4256. [CrossRef]
18. Chen, L.; Lu, W.; Zhang, K.; Zhang, Y.; Zhao, L.; Zheng, Y. TossNet: Learning to Accurately Measure and Predict Robot Throwing of Arbitrary Objects in Real Time with Proprioceptive Sensing. *IEEE Trans. Robot.* **2024**, *40*, 3232–3251. [CrossRef]
19. Monastirsky, M.; Azulay, O.; Sintov, A. Learning to Throw with a Handful of Samples Using Decision Transformers. *IEEE Robot. Autom. Lett.* **2023**, *8*, 576–583. [CrossRef]
20. Beer, F.; Johnston, E.; Mazurek, D.; Cornwell, P.; Self, B. *Vector Mechanics for Engineers: Statics And Dynamics, SI*; McGraw Hill: Columbus, OH, USA, 2025.
21. Cha, H.; Lee, I.; Seo, J. High-Speed Scooping Through Dynamic Manipulation: Model and Practice. *IEEE Robot. Autom. Lett.* **2025**, *10*, 1377–1384. [CrossRef]
22. Lynch, K.M.; Park, F.C. *Modern Robotics*; Cambridge University Press: Cambridge, UK, 2017.
23. Williams, C.; Rasmussen, C. Gaussian processes for regression. *Adv. Neural Inf. Process. Syst.* **1995**, *8*, 514–520.
24. Byrd, R.H.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [CrossRef]
25. Liu, Z.; Zhang, O.; Zhao, Y.; Zhu, Q.; Liu, J. Adaptive neural network-based fixed-time control for robots with input saturation and prescribed performance. *Nonlinear Dyn.* **2025**, *113*, 18229–18241. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

FES Control of a Finger MP Joint with a Proxy-Based Super-Twisting Algorithm

Hua Chen ¹, Xiaogang Xiong ^{2,*}, Koki Honda ³, Shouta Okunami ¹ and Motoji Yamamoto ^{1,*}

¹ Mechanical Engineering Department, Kyushu University, Ito Campus, Fukuoka 8190395, Japan; tinnka@outlook.com (H.C.); okunami.shouta.091@s.kyushu-u.ac.jp (S.O.)

² School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China

³ Graduate School of Frontier Sciences, Tokyo University, Kashiwa Campus, Tokyo 2778563, Japan; khonda@edu.k.u-tokyo.ac.jp

* Correspondence: xiongsg@hit.edu.cn (X.X.); yama@mech.kyushu-u.ac.jp (M.Y.); Tel.: +81-092-802-2700 (M.Y.)

Abstract: To improve motion accuracy through functional electrical stimulation (FES) of forearm muscles, feedback control laws are applied to the index finger's metacarpophalangeal (MP) joint. This paper introduces a proxy-based super-twisting algorithm (PSTA) for precise servo control of MP joints via FES. The PSTA combines first-order sliding mode control with a second-order super-twisting algorithm, effectively preventing windup during FES saturation and ensuring robust, accurate control. An implicit Euler method minimizes numerical chattering in the digital implementation. Experiments with Arduino and volunteers confirm the algorithm's effectiveness.

Keywords: functional electrical stimulation (FES); finger metacarpophalangeal (MP) joint; sliding mode control (SMC); super-twisting algorithm (STA)

1. Introduction

The functional electrical stimulation (FES) is a type of neuromuscular electrical stimulation that induces muscle contraction by passing an electric current through motor nerves. FES is clinically applied to assist in the rehabilitation training of hemiplegic patients due to strokes. After a stroke, patients often face upper limb impairments like muscle weakness and loss of coordination. Functional Electrical Stimulation (FES) can assist in restoring these functions by stimulating the neuromuscular system electrically [1]. In recent years, it has become popular as an interface for Augmented Reality (AR) [2] and Brain Computer Interface (BCI) systems [3]. Most FES rehabilitation systems use open-loop control schemes, limiting them to simple movements only. However, closed-loop feedback control with joint angle measurement is necessary to achieve complex and accurate movements such as the extension–flexion symmetrical exercise, which effectively reconstructs motor functions [4,5].

There are numerous closed-loop feedback control strategies available for FES control in the literature, but when applied to the case of low-computation such as Arduino boards, each one presents unique challenges. The backstepping control method [6], the model predictive control (MPC) [7] and the control law based on the recurrent neural network (RNN) [8] were also proposed for the FES control. However, they require an accurate model of muscle or large amount of data, which are difficult to obtain or difficult to implement in embedded systems. In recent years, due to the robustness to disturbances and model uncertainties, the sliding mode control (SMC) has been widely employed for the control of joint angles through the FES [8–11]. These SMC methods have the robustness to disturbances and finite-time convergence rate, but also fall into the trouble of numerical chattering problem, which deteriorates the control accuracy [12]. The higher-order SMC (HOSMC) attenuate the magnitude of chattering, but it requires the model of muscle and

the knowledge of the upper bound of the uncertainties and disturbances. It also requires the unknown disturbance to be differentiable, which limits the scope of its applications. Furthermore, even in the noise-free cases, the conventional discrete-time implementation of HOSMC also requires a high sampling rate and powerful real-time computation platform to avoid numerical chattering [13,14].

In these designed control schemes for FES, an accurate muscle model is crucial for controlling joint angles. For instance, Sakaino et al. developed a second-order linear FES model and proposed a full-state feedback control scheme based on it [15]. However, the challenge lies in the fact that the muscle model varies with time and differs among individuals, making accuracy difficult to achieve. Developing a high-fidelity muscle model of finger metacarpophalangeal (MP) joint, which is only a single joint with one degree of freedom (DoF), for different individuals is still cumbersome due to the nonlinearity, time-varying, and individual-depending characteristics of the musculoskeletal system. Furthermore, some of the control strategies, e.g., the MPC [7], the RNN [8], and the HOSMC [11], all require a high-frequency sampling rate, large amount of data, or powerful computation resource to achieve a high performance of the FES control, which is not suitable for embedded devices of MP joint control with limited computation power and sampling rate.

Among the various control strategies, the super-twisting algorithm (STA) is a popular method within higher-order sliding mode control (HOSMC). It achieves second-order sliding mode accuracy while only requiring first-order information of the sliding variable. STA has been widely used in mechatronic systems such as fish robots [16] and exoskeleton systems [17]. To reduce chattering effects while maintaining robustness, an adaptive version of STA was developed and is commonly employed for real-time control in practical systems [18–20]. However, conventional and adaptive STAs are not suitable for the input saturation cases and they have no anti-windup effects due to the integration action in the STA. To further strengthen disturbance rejection, recent studies have shown significant advancements. Zhang et al. proposed a robust finite-time command-filtered backstepping control for flexible-joint robots using only position measurements [21]. Zhang et al. also developed a rapid swing control strategy for 5-DOF tower cranes [22]. Additionally, Qian et al. introduced a neural network-based sliding mode control for dual ship-mounted cranes, effectively suppressing sea wave disturbances [23].

To achieve safer control with anti-windup capabilities, Kikuuwe et al. [24] proposed proxy-based sliding mode control (PSMC). This notable scheme combines first-order SMC with conventional PID control to provide robust control and damped motion without needing a plant model. PSMC has been widely applied in various scenarios, including exoskeleton suits [25], due to its desirable characteristics like the anti-windup effect and nonovershooting behavior when actuators are saturated. As noted in [24], when the input is not saturated or the closed-loop system remains on the sliding surface, PSMC functions equivalently to a conventional PID controller. One limitation of PSMC is that its equivalent PID controller's asymptotic accuracy depends on proportional, integral, and derivative gains that require careful tuning to achieve satisfactory performance.

This paper intends to solve the control problem of FES for the MP joints with a new method of SMC by exploiting its robustness and its anti-windup property. It has a high control accuracy as well as a low computation complexity, such that it can be even implemented in embedded devices such as the Arduio platform in a real-time manner. Meanwhile, the model uncertainties of the MP joints and actuation saturation are also taken into account to design the new control law. The main contributions are as follows:

- **Robustness and High Control Accuracy:** Introducing the proxy-based super-twisting algorithm (PSTA) for FES control of MP joints, combining the robustness of Sliding Mode Control (SMC) and the high control accuracy of the Super-Twisting Algorithm (STA).
- **Avoiding Numerical Chattering:** Proposing an implicit Euler discretization method to avoid numerical chattering while maintaining high control accuracy and robustness.

- **Experimental Validation:** Validating the PSTA and its implicit discretization through experiments on MP joints using FES. Achieving high accuracy on a low-cost Arduino Mega board in real-time, demonstrating robustness and low computational complexity without detailed muscle model information.

To the best of the authors' knowledge, this is the first time to achieve such a high control accuracy for the finger MP joint through the FES with such a low-cost hardware platform and without requiring the knowledge of the muscle model of the MP joint.

Notations: Let $\mathcal{A}$ be a closed interval of real numbers defined as $\mathcal{A} := [A, B]$ with $A < B \in \mathbb{R}$. This paper uses the following projection function $\text{proj}(\cdot)$:

$$\text{proj}(\mathcal{A}, x) = \max(\min(B, x), A), \forall x \in \mathbb{R} \quad (1)$$

where $x \in \mathbb{R}$. This paper also uses the following definition of signum functions, where the function $\text{sgn}(\cdot)$ is the set-valued version of the signum function that differs from the single-valued version $\text{sign}(\cdot)$. This kind of set-valued definition, instead of the conventional single-value definition, has been employed by many previous papers [12,26,27]. To distinguish the difference, $\text{sign}(\cdot)$ is defined as the single-value function: $\text{sign}(x) := \text{sgn}(x)$ for $x \neq 0$ and $\text{sign}(x) := 0$ for $x = 0$.

With a nonnegative scalar $F \geq 0$, the following relation is satisfied:

$$x \in F\text{sgn}(y - x) \iff x = \text{proj}(\mathcal{F}, y), \quad (2)$$

with $\mathcal{F} = [-F, F]$, of which the proof can be found in previous papers [24,26]. The following mathematical equivalence is strictly satisfied [28]:

$$z \in -\alpha_1 \text{sgn}(z - x) - \alpha_2 \text{sgn}(z - y) \iff z = \text{proj}(\mathcal{B}, x) \quad (3)$$

where α_1, α_2 are parameters, $0 < \alpha_2 < \alpha_1$ and $x, y, z \in \mathbb{R}$, $\mathcal{B} := [\text{proj}(-\mathcal{C}, y), \text{proj}(\mathcal{C}, y)]$, $\mathcal{C} := [\alpha_1 - \alpha_2, \alpha_1 + \alpha_2]$ is a compact set, and $\text{proj}(\cdot)$ is defined as in (1).

The structure of this paper is arranged as follows. The problem statement is introduced in Section 2. The proposed proxy-based super-twisting algorithm (PSTA) and its stability analysis are given in Section 3. Subsequently, Section 4 presents the proposed implicit Euler implementation of the proposed PSTA. The proposed approach is evaluated through experimental results of MP joint control in Section 5. Finally, conclusions and some future work are given in Section 6.

2. Problem Statement

In muscle modeling literature, the Hill-type model is a well-known approach [29]. This model represents the muscle as a second-order system with two poles and one zero. However, M. Ferrarin and A. Pedotti [30] studied the relationship between pulse and torque in knee joint muscle models. They found that in many cases, a first-order model with only one pole is more optimal than a second-order system model. Therefore, assuming that this simplified first-order system can also be applied to the finger MP joint, we can define the angle-torque model using an actuation saturation first-order system:

$$\dot{x} = a(t, x) + b(t, x)\mu^*, \quad (4a)$$

$$\mu^* = \text{proj}(\mathcal{F}, \mu), \sigma = \sigma(t, x) \quad (4b)$$

where $x \in \mathbb{R}$ is the measurable angle of the finger MP joint, as shown in Figure 1, μ is the FES signal, i.e., the time duration of a fixed magnitude of voltage, determined by a control law μ to be designed, μ^* is the maximum FES signal, i.e., the projections of the control input μ to the set $\mathcal{F} = [-F, F]$ with the maximum duration of time $F > 0$, $a(t, x)$ and $b(t, x) > 0$ are smooth functions, σ is the sliding variable to be defined [31], and $\text{proj}(\cdot)$ is the projection function defined as in (1). The object is to develop a control law μ that

can be realized within the single board computer (SBC) of Arduino such that the sliding variable converges to zero, i.e., $\sigma(t, x) = 0$.

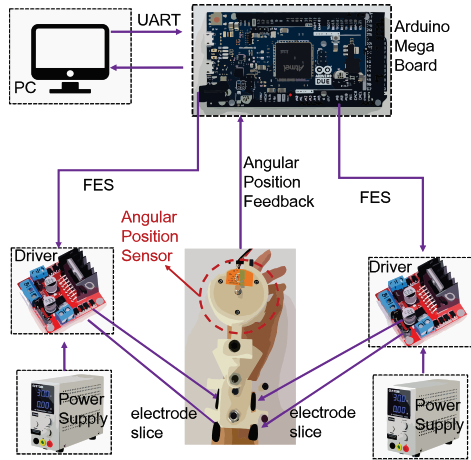


Figure 1. The hardware components and connections of the experimental setup for the servo control of the MP joint.

The challenge lies in the fact that the values of terms $a(t, x)$ and $b(t, x)$ vary over time and across individuals due to the model representing the muscle of the MP joint in humans. This variability makes it difficult to obtain an accurate model. Despite this uncertainty, it is established that the finger MP joint model (4) is a first-order system with a relative degree of 1. Then, as noted by Levant [31], let us define the dynamics of the sliding variable σ :

$$\dot{\sigma} = f(t, x) + g(t, x)\mu^* \quad (5)$$

where $f(t, x)$ and $g(t, x)$ are the Lie derivatives derived from (4), i.e., $f(t, x) = \frac{\partial \sigma}{\partial x} a(t, x)$, $g(t, x) = \frac{\partial \sigma}{\partial x} b(t, x)$. The two functions are unknown smooth but bounded functions and let us assume [31], $\forall t > 0, x \in \mathbb{R}$:

$$|f(t, x)| \leq F_0, |\dot{f}(t, x)| \leq F_1, 0 < g_m \leq g(t, x) \leq g_M, \quad (6)$$

that is, $f(t, x)$ and its derivative are bounded by constants F_0 and F_1 , respectively, while the input gain $g(t, x)$ is a constant or slowly varying unknown value that is bounded by two constants: g_m and g_M . The objective is to design a control law μ that drives the state variable, denoted as x , to a desired trajectory x_d , i.e., $\sigma = 0$, even when under actuation saturation $\mu^* = \text{proj}(\mathcal{F}, \mu)$. This must be achieved without prior knowledge of the functions $f(t, x)$, $g(t, x)$ mentioned above or existing but unknown constants F_0 , F_1 , g_m , and g_M . Previous works in SMC [14,31] have made similar assumptions to (6). Instead of focusing on the finger MP joint model (4), it would be reasonable to design an SMC controller with respect to (5) using (6), which varies for different individuals, but remains within a certain range.

With these bounded assumptions, it is straightforward to design some strategies of SMC to stabilize the system (4), because the problem of stabilizing the systems like (4) has been solved by different SMC methods under the appropriate assumptions, such as terminal SMC, adaptive SMC, HOSMC, and implicit-Euler SMC [32]. However, there are three challenges to design the strategies of SMC here:

- Without the knowledge of boundness parameters F_0 , F_1 , g_m , and g_M and only the actuation saturation level F , adaptive SMC methods can be designed here, while their implementations require high-frequency sampling and switching rate of actuation, which is not suitable for the embedded platform of a low computation resource and sampling rate.

- One strategy is to set the gains of the gains of terminal SMC and HOSMC strategies as large as possible such that the stability of the uncertain closed-loop systems (5) and (6) is guaranteed. The problem is that the magnitude of chattering is proportional to the size of gains and the sampling period, which deteriorates the control accuracy, especially for the embedded system with a low sampling frequency.
- Even with large gains, the implicit-Euler discretizations of SMC can remove the numerical chattering in the absence of noises, but the method requires the knowledge of unavailable term $g(t, x)$, making it difficult to be implemented in such cases.

3. Proposed Proxy-Based Super-Twisting Algorithm (PSTA)

Continuous-Time Expression of PSTA

To improve the accuracy of PID control while retaining the anti-windup property of PSMC [24], we propose incorporating first-order sliding mode control (SMC) with the super-twisting algorithm (STA) [33,34], one of second-order SMC strategies. This combination exploits the set-valueness of extended SMC in the Filippov sense and is expected to enhance second-order control accuracy when saturation is inactive. It can be described as follows:

$$\sigma = e + \kappa_3 \int_0^t |e(\tau)|^\zeta d\tau, \quad \mu^* = \text{proj}(\mathcal{F}, \mu) \quad (7a)$$

$$\mu \in -F \text{sgn}(s_2), \quad s_2 = \sigma - s_1, \quad (7b)$$

$$\mu = -\kappa_1 |s_1|^{1/2} + v, \quad \dot{v} \in -\kappa_2 \text{sgn}(s_1), \quad (7c)$$

where “ $\in$ ” means that the signum function $\text{sgn}(\cdot)$ is set-valued instead of the conventional single-valued version, $e = x - x_d$, σ is the defined sliding variable, and when $\zeta = 1$, it changes to the linear one $\sigma = e + \kappa_3 \int_0^t e(\tau) d\tau$, μ is defined in (4) and (5), while μ^* is the saturated input, $\kappa_3 > 0$ is a parameter of determining the convergence rate of e , s_2 is an intermediate variable and can be viewed as the sliding variable of the first-order SMC, s_1 is an another intermediate variable and can be viewed as the sliding variable of the STA, and their summation always satisfies $s_1 + s_2 = \sigma$, $F > 0$ is the gain of the sliding mode control (SMC) and is also the saturation level of actuator, v is also an intermediate variable of the STA, $\kappa_1, \kappa_2 > 0$ are the gains of the conventional STA and its selection can refer to the work in [35], and the control input μ is determined by the set-valued first-order SMC (7b) and the set-valued STA (7c) simultaneously. The notation $|\cdot|^\zeta$ is defined as $\forall x \in \mathbb{R}, |x|^\zeta = |x|^\zeta \text{sign}(x); \forall \zeta \in (0, 1], |0|^\zeta = 0; |x|^0 = \text{sign}(x)$.

It should be noted that μ appearing in both (7b) and (7c) means that it is subjected to the constraints of (7b) and (7c) simultaneously, which is possible due to the set-valued mappings of (7b) and (7c). Due to the special structure of the above controller (7), it is referred as the proxy super-twisting algorithm (PSTA) thereafter. Additionally, from (7), one has $\mu = \mu^*$, that is, $\mu \in \mathcal{F} = [-F, F]$ is always satisfied, which is not the case of the conventional STA.

One desired property of the PSTA (7) is that, due to the saturation constraint by (7b), it has a similar property of the PSMC [24] and the conditioned STA [36], i.e., anti-windup effect with the saturation level $F > 0$, which is inherited from the PSMC. However, the differences is that, when $|\mu| < F$, the control input μ is determined by the STA (7c), which enables it to achieve a second-order asymptotical accuracy and higher than that of the PSMC with the same three number of parameters. Moreover, comparing to the conventional STA [33], it has the properties of anti-windup property and can deal with nondifferentiable and differentiable disturbances due to the combination of (7b) and (7c).

Let us see the dynamics (5) interconnected by the PSTA (7), by first checking the derivative of s_2 in (7) with $\mu = \mu^*$, that is, $\dot{s}_2 = \dot{\sigma} - \dot{s}_1$ and:

$$\begin{aligned} \dot{s}_2 &= f(t, x) + g(x, u)\mu^* - \dot{s}_1 = \dot{e} + \kappa_3 |e|^\zeta - \dot{s}_1 \\ &\in [-F_0, F_0] + [g_m, g_M]\mu - \dot{s}_1, \end{aligned} \quad (8)$$

from which one can see the closed-loop system with the control input μ subjected to the constraints $\mu \in -F \operatorname{sgn}(s_2)$, $s_2 = \sigma - s_1$, and $\mu = -\kappa_1 [s_1]^{1/2} + v$, $\dot{v} \in -\kappa_2 \operatorname{sgn}(s_1)$. Since μ is always and simulatively subjected to (7b) and (7c), (8) can be equivalently rewritten as the following system:

$$\dot{s}_2 \in [-F_0, F_0] + [g_m, g_M]\mu - \dot{s}_1, \mu \in -F \operatorname{sgn}(s_2), \quad (9a)$$

$$\mu = -\kappa_1 [s_1]^{1/2} + v, \dot{v} \in -\kappa_2 \operatorname{sgn}(s_1), \quad (9b)$$

from which one can see that the control input μ is simultaneously subjected to constraints of the first-order SMC and the second-order STA and is independent on the unknown model (4). Alternatively, one can say that the input μ computed by the STA always satisfies $|\mu| \leq F$, which is constraint by the first-order SMC. The closed-loop system (9) has the stability property shown in Theorem 1.

Theorem 1. Consider the dynamical system (5) with the uncertain smooth functions $f(t, x)$ and $g(t, x)$ satisfying $\sup |f(t, x)| \leq F_0$ and $0 < g_m \leq g(t, x) \leq g_M$, $\forall t \geq t_0 \geq 0, x \in \mathbb{R}$. If $F > F_0/g_m$, $\kappa_2 \geq F_1/g_m$, $\kappa_1 > \sqrt{\kappa_2 + F_1}/g_m$, and $\kappa_3 > 0$ are satisfied for all $t \geq 0$, then, with bounded initial condition $s_2(t_0) + s_1(t_0) = \sigma(t_0)$, the equilibrium of the sliding variable $\sigma = s_1 + s_2 = 0$ of the closed-loop system (5) interconnected with (7) is asymptotically attained.

Proof. Here, only the sketch of the proof is provided, because both the stabilities of first-order SMC and the STA are well studied in the literature. Because the closed-loop system (5) interconnected with (7) can be equivalently expressed as (9), one can only check the stability of (9). Let us first check the stability of (9a):

$$\dot{s}_2 \in [-F_0, F_0] + [g_m, g_M]\mu - \dot{s}_1, \mu \in -F \operatorname{sgn}(s_2), \quad (10)$$

of which the stability can be easily recognizable when $F > F_0/g_m$ and $|\dot{s}_1|$ remains bounded, because it is the standard first-order SMC. The asymptotical satiability of (10) is straightforward to show by defining the Lyapunov function $V_1(s_2) = s_2^2/2$ according to ([26], Prop. 1), when only (10) is considered and $\dot{s}_1$ is considered as a bounded disturbance. If the assumptions in (6) and $F > F_0/g_m$ hold, there exists a finite time $t_1 \geq 0$ (depending on initial conditions $s_2(t_0)$), $\forall t \geq t_1, \dot{s}_2 = s_2 = 0$ ([26], Prop. 1). This convergence period $t \in [t_0, t_1]$ of (10) can be considered as the first stage of (9). Therefore, in the first stage, both $|s_2|$ and $|\dot{s}_2|$ are shrinking if $|\dot{s}_1|$ is bounded. Then, for $t \geq t_1, s_2 = \dot{s}_2 = 0$, i.e., in the second stage, from (9), due to $\sigma = s_1 + s_2$ and $\dot{\sigma} = \dot{s}_1 + \dot{s}_2$, one has the following expressions:

$$\dot{s}_1 \in [-F_0, F_0] + [g_m, g_M]\mu = f(t, x) + g(t, x)\mu \quad (11a)$$

$$\mu = -\kappa_1 [s_1]^{1/2} + v, \dot{v} \in -\kappa_2 \operatorname{sgn}(s_1), \quad (11b)$$

which can be further rewritten as follows:

$$\dot{s}_1 \in -[g_m, g_M]\kappa_1 [s_1]^{1/2} + z_1 \quad (12a)$$

$$\dot{z}_1 \in -[g_m, g_M]\kappa_2 \operatorname{sgn}(s_1) + [-F_1, F_1], \quad (12b)$$

with definition $z_1 := f(t, x) + g(t, x)v$. Note from (11) to (12), $\dot{g}(t, x) = 0$ has been used, because the input gain $0 < g_m \leq g(t, x) \leq g_M$ in (6) assumes to be a constant or slowly varying but bounded parameter. The system (12) is finite-time stable if the above gain conditions satisfied, if $\kappa_2 \geq F_1/g_m$, $\kappa_1 > \sqrt{\kappa_2 + F_1}/g_m$, and $\kappa_3 > 0$ hold [37].

The above proof is based on the assumption that $|\dot{s}_1|$ is bounded. To demonstrate that $|\dot{s}_1|$ and $|s_1|$ are shrinking and bounded, one can equivalently rewrite (10) as the dynamics about s_1 with the control law μ replaced by the STA (7c) and consider $\dot{s}_2$ as the disturbance. Then, by following previous works of STA [33,37], one can see the assumption holds that $|\dot{s}_1|$ and $|s_1|$ are shrinking and bounded. From the above analysis of (10), $|\dot{s}_2|$ is decreasing,

and thus, the size of bound of $|\dot{s}_1|$ is decreasing. Therefore, $\sigma = s_1 + s_2 = 0$ is attained for a finite time $t \geq t_s = \max(t_1, t_2)$. $\square$

Remark 1. From Theorem 1, it is evident that even if the models $f(t, x)$ and $g(t, x)$ are unknown, and the bounds g_m, g_M, F_0 , and F_1 are not available in practice, setting large enough gains such as F, κ_1 , and κ_2 satisfies the conditions of Theorem 1. This guarantees stability of the closed-loop system (5) interconnected with (7). However, due to these large gains, numerical chattering may occur, which needs to be resolved using numerical methods. As mentioned in Section 2, conventional SMC strategies face three challenges in this situation. However, these challenges can be overcome by using PSTA (7) and its implicit-Euler discretization method without depending on $g(t, x)$. Further details are provided in subsequent sections.

4. Discretization Scheme of PSTA

In the first stage, i.e., $s_2 \neq 0$ or $s_1 \neq \sigma$, the dynamics (5) interconnected by the control law (7) is actually governed by the system (9), which is a perturbed first-order dynamical system controlled by the first-order SMC.

One can see that the PSTA (7) is an algebraic inclusion and cannot be directly implemented due to the set-valued signum functions. To obtain the discretization scheme (7), let us first assume that x is available and x_d is known. Then, let us eliminate μ by rewriting the system (7) as follows:

$$-\kappa_1 |s_1|^{1/2} + v \in -F \operatorname{sgn}(\sigma - s_1), \quad (13a)$$

$$\dot{v} \in -\kappa_2 \operatorname{sgn}(s_1), \quad (13b)$$

which is an algebraic inclusion about the sliding variable s_1 . Let us discretize (13) with the implicit Euler method:

$$v_{k+1} \in F \operatorname{sgn}(s_{1,k+1} - \sigma_{k+1}) + \kappa_1 |s_{1,k+1}|^{1/2} \operatorname{sgn}(s_{1,k+1}) \quad (14a)$$

$$v_{k+1} \in v_k - h\kappa_2 \operatorname{sgn}(s_{1,k+1}), \quad (14b)$$

where $h > 0$ represents the sampling period of time and the subscript index $k \in \{\mathbb{N} \cup 0\}$ is the index of time-stepping, i.e., $t_k = hk$ and $t_k - t_{k-1} = h$. Substituting v_k in (14a) with the expression in (14b) leads to the following expression:

$$v_k - \kappa_1 |s_{1,k+1}|^{1/2} \operatorname{sgn}(s_{1,k+1}) \in F \operatorname{sgn}(s_{1,k+1} - \sigma_{k+1}) + h\kappa_2 \operatorname{sgn}(s_{1,k+1}), \quad (15)$$

which is equivalent to the following one:

$$v_k - \kappa_1 |s_{1,k+1}|^{1/2} \operatorname{sgn}(s_{1,k+1}) \in h\kappa_2 \operatorname{sgn}(|s_{1,k+1}|^{1/2} \operatorname{sgn}(s_{1,k+1})) \\ F \operatorname{sgn}(|s_{1,k+1}|^{1/2} \operatorname{sgn}(s_{1,k+1}) - |\sigma_{k+1}|^{1/2} \operatorname{sgn}(\sigma_{k+1})) \quad (16)$$

due to the property $\operatorname{sgn}(x) = \operatorname{sgn}(\alpha x), \forall \alpha > 0, x \in \mathbb{R}$. If κ_2 and h are properly selected such that $F > h\kappa_2$ is satisfied, inspired by the work in [38], the mathematical equivalence (3) can be applicable to (16), leading to:

$$v_k - \kappa_1 |s_{1,k+1}|^{1/2} \operatorname{sgn}(s_{1,k+1}) = \operatorname{proj}(\mathcal{D}_k, y_k) \quad (17)$$

where $y_k := v_k + \kappa_1 |\sigma_{k+1}|^{1/2} \operatorname{sgn}(\sigma_{k+1})$, $\mathcal{D}_k := [\mathcal{C}_{1,k}, \mathcal{C}_{2,k}]$ with $\mathcal{C}_{1,k} := \operatorname{proj}(-\mathcal{F}, v_{k-1})$, $\mathcal{C}_{2,k} := \operatorname{proj}(\mathcal{F}, v_{k-1})$, and $\mathcal{F} := [F - h\kappa_2, F + h\kappa_2]$. From (17), one has the updated law of $a_{2,k}$:

$$s_{1,k+1} = \frac{1}{\kappa_1^2} (\operatorname{proj}(\mathcal{D}_k, y_k) - v_k)^2 \operatorname{sign}(v_k - \operatorname{proj}(\mathcal{D}_k, y_k))$$

where $\text{sign}()$ is the single-valued signum function. Then, from (17), because of $\mu = -\kappa_1 |s_1|^{1/2} + v$, the first-stage of the proposed implicit-Euler digital implementation scheme for (7) is as follows:

$$\mu_k = \text{proj}(\mathcal{D}_k, y_k) - v_k + v_{k+1} \quad (18a)$$

$$v_{k+1} \in v_k - h\kappa_2 \text{sgn}(s_{1,k+1}). \quad (18b)$$

One can see that “ $\in$ ” still exists in (18b) and one straightforward way is to approximate it with the single-valued signum function $\text{sign}()$. Here, to avoid the numerical chattering caused by the single-valued signum function, an alternative way is to reuse (17): if $\text{proj}(\mathcal{D}_k, y_k) - v_k \neq 0$:

$$\text{sgn}(s_{1,k+1}) = \frac{v_k - \text{proj}(\mathcal{D}_k, y_k)}{|v_k - \text{proj}(\mathcal{D}_k, y_k)|}, \quad (19)$$

due to $\kappa_1 |s_{1,k+1}|^{1/2} = |\text{proj}(\mathcal{D}_k, y_k) - v_k|$ from (17). Finally, one has the proposed digital implementation scheme of (7) without any signum function:

$$v_{k+1} = v_k - h\kappa_2 \frac{v_k - \text{proj}(\mathcal{D}_k, y_k)}{|v_k - \text{proj}(\mathcal{D}_k, y_k)|} \quad (20a)$$

$$\mu_k = \text{proj}(\mathcal{D}_k, y_k) - h\kappa_2 \frac{v_k - \text{proj}(\mathcal{D}_k, y_k)}{|v_k - \text{proj}(\mathcal{D}_k, y_k)|}. \quad (20b)$$

Note that the variable y_k contains the prediction value of σ_{k+1} as in [12], which can be approximated with $\sigma_{k+1} \simeq \sigma_k - hF\text{proj}(\mathcal{I}, \sigma_k / (Fh))$ with $\mathcal{I} = [-1, 1]$ [14].

Remark 2. From the digital implementation algorithm (20), one can see that it is different from the discrete-time realizations of the conventional STA in the work [12]. The discrete-time updating laws of both u_k and v_k have some mathematical similarities to the quasi-continuous SMC [39] and the discrete-time equivalents of STA in [40]. In all these cases, the numerical chattering have been significantly reduced.

In the second stage, i.e., when $s_2 = 0$ and $\sigma = s_1$, the dynamics (5) interconnected by the control law (7) is actually governed by the system (12), which is a perturbed first-order dynamical system controlled by the STA with uncertain input gain $0 < g_m \leq g(t, x) \leq g_M$. That means during the second stage, the properties of the proposed control law (7), such as the control accuracy and convergence, are finally determined by the STA.

There are some works related to the digital realizations of the STA in the literature [12,40]. However, as illustrated and compared in [32], these works have different properties, such as some have lost the asymptotical second-order control accuracy, and some are sensitive the overestimation of gains. To obtain the discrete-time implementation algorithm during the second stage, for simplicity, let us assume that $g_t = g(t, x)$ $k_1 = g_t \kappa_1$, $k_2 = g_t \kappa_2$, $\phi(t) = f(t, x)$, $\triangle(t) = \dot{f}(t, x)$ and consider the discretization and realization of (12) with the implicit Euler method:

$$s_{1,k+1} = s_{1,k} + h\mu_k + h\phi_{k+1}, \phi_{k+1} = \phi_k + h\triangle_{k+1}, \quad (21a)$$

$$v_{k+1} \in v_k - hk_2 \text{sgn}(s_{1,k+1}), \quad (21b)$$

$$\mu_k = -k_1 |s_{1,k+1}|^{1/2} + v_{k+1} \quad (21c)$$

where one should be noted that the discretizations $\phi_k = \phi(t_k)$ and $\triangle_k = \triangle(t_k)$ do not mean that ϕ_k is a piece-wise constant, but refers to the fact that it has no effect on e_k within the period (t_k, t_{k+1}) after sampled at t_k by analog-to-digital (ADC) devices in practice and (21) is an implicit Euler discretization of (12), while other schemes such as the simple explicit

Euler and more accurate zero-order-hold (ZOH) are also possible. Substituting (21c) to (21a) leads to:

$$s_{1,k+1} = s_{1,k} - h\mu_k + hz_{2,k} + h^2\Delta_{k+1} \quad (22a)$$

$$\mu_k \in \left(k_1 \sqrt{|s_{1,k+1}|} + hk_2 \right) \text{sgn}(s_{1,k+1}) \quad (22b)$$

where $z_{2,k} := \phi_k + v_k$, Δ_{k+1} and $s_{2,k}$ are unknown to us because $s_{2,k} := \phi_k + v_k$ contains the external unknown disturbance ϕ_k . By following the work [41], one has the following results:

$$s_{1,k+1} = s_{1,k} - h\tilde{\mu}_k - h\phi_k - h^2\Delta_{k+1} \quad (23a)$$

$$\tilde{\mu}_k = \tilde{u}_{1,k} + v_{k+1}; \tilde{u}_{1,k} = \frac{1}{h} \text{proj}(C_k, s_{1,k}) \quad (23b)$$

$$v_{k+1} = v_k + h\tilde{u}_{2,k}; \tilde{u}_{2,k} = \frac{1}{h^2} \text{proj}(C_{2,k}, s_{1,k}) \quad (23c)$$

where $C_{2,k} := [-D_{2,k}, D_{2,k}]$ while C_k is defined as follows:

$$C_k := \begin{cases} [-D_{1,k}, D_{1,k}] & \text{if } |s_{1,k}| > D_{1,k}, \\ [-D_{2,k}, D_{2,k}] & \text{else} \end{cases} \quad (24)$$

with $D_{1,k} := hk_1 \sqrt{|s_{1,k}|} + h^2k_2$ and $D_{2,k} := h^2k_2$.

To make the proposed discrete-time realization algorithm (23) easy to be implemented, the condition $s_2 = 0$ or $\sigma = s_1$ is relaxed as $|\sigma - s_1| \leq \varepsilon$, where the parameter $\varepsilon > 0$ is a small real number as the threshold of the practical sliding motion. If $|\sigma_k - s_{1,k}| > \varepsilon$, it means that the discrete-time system (23) is still within the first stage, while if $|\sigma_k - s_{1,k}| < \varepsilon$, then it shows that (23) has attained the sliding motion of the first stage, i.e., $\sigma_k \approx s_{1,k}$. The input gain g_t is unknown, and thus, $k_j, j = 1, 2$ are replaced with κ_j in practice.

5. Experiments

5.1. Experiment Setup

The Ethics Committee of the School of Engineering at Kyushu University approved the experiments (Approval no. 2021-02), which were meticulously conducted with strict adherence to protocols. The experiment setup is illustrated in Figure 1, which includes an Arduino microprocessor (Arduino Mega, the manufacturer is Laplace, located in Miyagi, Japan), a driver (L298n), electrodes (IVES ELR-6211), and a joint angle acquisition unit (CP45 Single-turn Potentiometer). Six volunteers, *A, B, C, D, E, F*, four females and two males ranging in age from 18 to 60 years, participated in the study. Preliminary experiments revealed that electrical stimulation elicited a more sensitive response in the MP joint of the index finger than other finger joints; therefore, this joint was selected for further investigation to enable clearer comparison between different control methods. The location of each participant's Flexor digitorum superficialis muscle, responsible for flexion movement, and Extensor digitorum muscle, responsible for extension movement, were identified on their arm's surface, as these target muscles are located relatively superficially. Surface electrode patches provided by OGwellness (model ELR-615) were used after simple adjustments were made to their positions upon application; if no motion response was observed upon current application, electrodes were slightly relocated within close proximity until a motion response was achieved in the finger.

As studied in our previous research [42], the biphasic wave can effectively stimulate the nerve-muscles, and thus, leads to joint movements. Therefore, a stimulation signal is employed in experiments with alternating positive and negative square waves. The magnitude of the stimulation voltage remains constant, i.e., V_{max} or V_{min} , followed a zero voltage of with constant duration of 50 μ s. By changing the duration of the square wave with the input μ , the actual average voltage is changed and its magnitude of the actual

average voltage determines the amount of joint motion and the sign of u determines the motion direction. In the experiments, the output μ of the PSTA is as the actuation input, i.e., μ μ s with the saturation level $F = 150$ μ s, i.e., $|\mu| \leq F = 150$ μ s, and the feedback of the joint angle $x = \theta$ as the output is sampled by an analog angle sensor, one analog potentiometer, with a period of 5 ms from the Analog Pin of the Arduino equipped with ADC device. As shown in (7), we have adjusted the gain $F = 150$ μ s according to practical actuation saturation levels. For κ_1 and κ_2 , we set $\kappa_1 = 1.5\sqrt{L}$ and $\kappa_2 = 1.1L$ based on literature about STA. Due to its insensitivity to gain overestimation, we increased L until tracking error convergence was achieved. Regarding κ_3 in (7), we tuned it through trial and error to ensure there was no overshooting in experiments.

The proposed PSTA for the MP joint control is shown in Figure 1, where the PSTA implemented in the Arduino board and μ is calculated in a real-time manner with $h = 25$ ms. Then, the stimulation signal with a duration of μ μ s is generated by the Arduino board to the motor driver connected with an external power supply. The outputs of the two motor drivers are transmitted to electrode patches that are applied to the index finger extensor muscle group, i.e., Channel 1, and the index finger flexor muscle group, i.e., Channel 2, respectively. One circular potentiometer, CP45 Single-turn Potentiometer, as the encoder with a resolution of 0.35° is placed at the MP joint of the index finger. To prevent the wrist from rotating with the index finger, a fixation device is also placed on the wrist, as shown in Figure 1.

5.2. Experimental Results

The experimental results are shown in Figures 2 and 3 for $x_d = \theta_r = 30^\circ$. During the experiments, six different volunteers, A, B, C, D, E, F , were required to keep their index finger relaxed without tension throughout the experiments and the index MP joint was driven by the proposed algorithm “PSTA”, the STA of second-order SMC realized with the explicit-Euler discretization, denoted as “Explicit”, and the PID, respectively, to the target angle θ_r . With the usage of PID, we set its gains through trial and error to achieve the optimal performance. To show the performances of closed-loop control, the MP joint is also controlled by the self-conscious movement of the healthy volunteers, denoted as “Voluntary”, that is, each volunteer tries to control their MP joint to move from zero degree to 30° as quick as possible by observing a marker of 30° at the angle sensor with their own eyes. During the experiments of the six volunteers, the parameters of the proposed “PSTA” are kept as the same, i.e., $\kappa_1 = 2.2\sqrt{\kappa_2}$, $\kappa_2 = 0.08$, $\kappa_3 = 50$, $\xi = 0.6$. One can see that, without the information of $f(t, x)$ and $g(t, x)$ in (5), with the proposed algorithm PSTA for different experimenters, the controlled joint of MP finger quickly and exactly converges to θ_r with little overshooting, while other methods have lower accuracy, as shown in Figure 2 and Table 1.

Table 1. RMSE and SSE of six volunteers under different control methods in Figure 3.

Volunteer	Metric	PSTA	Explicit	PID	Voluntary
A	RMSE	0.39	2.48	0.47	0.67
	SSE	0.33	0.33	0.33	0.67
B	RMSE	0.35	1.84	1.54	0.38
	SSE	0.02	4.16	0.02	0.02
C	RMSE	0.33	1.04	0.59	0.95
	SSE	0.08	0.02	0.08	1.34
D	RMSE	0.33	2.76	1.06	0.70
	SSE	0.02	1.04	0.08	1.04
E	RMSE	0.35	1.74	1.35	0.83
	SSE	0.02	8.46	1.36	0.44
F	RMSE	0.38	1.74	1.76	0.42
	SSE	0.08	0.08	0.08	0.08

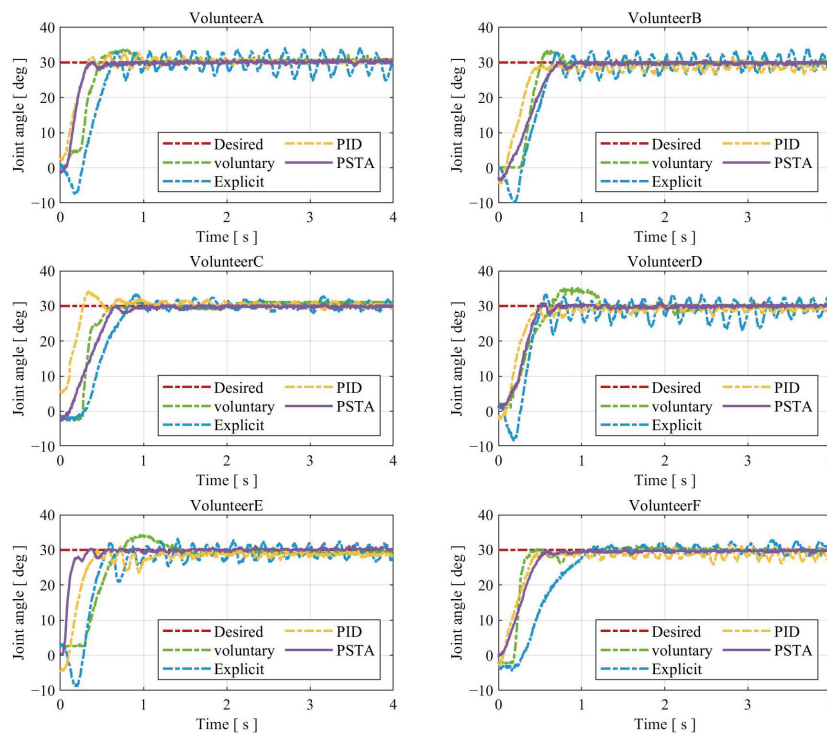


Figure 2. Experiment results of six different volunteers, *A, B, C, D, E, F*, for control of the MP joint with the low-cost Arduino Mega and an encoder of a resolution around 0.1° . For each volunteer *A~F*, the PID gains have tuned by trial and error method to achieve the best performance. The PSMC [24] has the anti-windup effect as the PSTA, but has the same accuracy of PID.

The results presented in Figure 2 indicate that the conventional explicit-Euler discretization of the STA, referred to as “Explicit”, causes numerical chattering–oscillation around the target degree. On the other hand, the proposed PSTA effectively reduces chattering and leads to accurate tracking results. The tracking errors of four strategies are displayed in Figure 3, which show fluctuations and small magnitude of chattering around zero during steady-state due to sensor noise and varying muscle strength.

Increasing PID or SMC gains (or force in “Voluntary”) to improve control accuracy can result in overshooting and oscillations. However, thanks to its set-valued extension and implicit-Euler discretization, even with high gains, the proposed PSTA demonstrates high robustness when controlling MP joints across different volunteers without any overshooting or oscillations. The corresponding results for six volunteers are shown in Table 1, with statistical data presented in Figure 4. It shows that the proposed PSTA has the smallest average tracking error and variance for the six volunteers, even better than the deliberate “Voluntary” movement.

In Table 2, we compare the computational load of the proposed PSTA method, the Explicit method, and the PID method. The implicit Euler discretization of PSTA increases the computational load to average 420 μ s, compared to the explicit Euler discretization of STA and PID. However, given a sampling period of 5 ms and an update period of 25 ms for control input μ_k , this increase is relatively minor. Despite higher computational complexity, the proposed PSTA offers benefits such as low chattering, anti-windup capabilities, and high control accuracy.

To further validate the robustness and effectiveness of the proposed method, we conducted additional experiments using a more complex reference trajectory, changing the target angle from 30° to 45° . The results of these experiments are shown in Figure 5. Similar to the previous experiments, the volunteers were instructed to keep their index finger relaxed throughout the experiments.

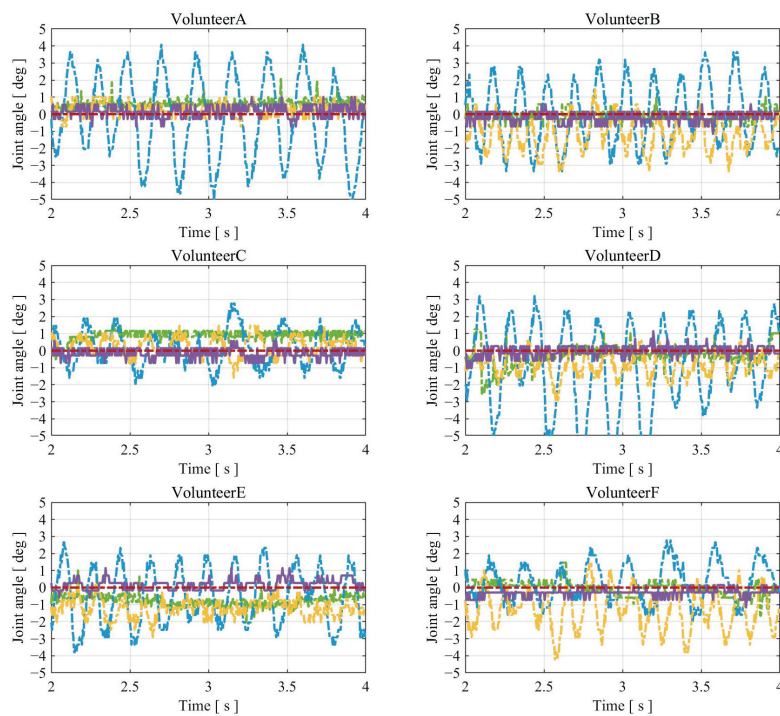


Figure 3. The tracking errors (red lines represent Desired, yellow lines represent PID, green lines represent Voluntary, purple lines represent PSTA and blue lines represent Explicit) of six different volunteers, *A, B, C, D, E, F*, in Figure 2.

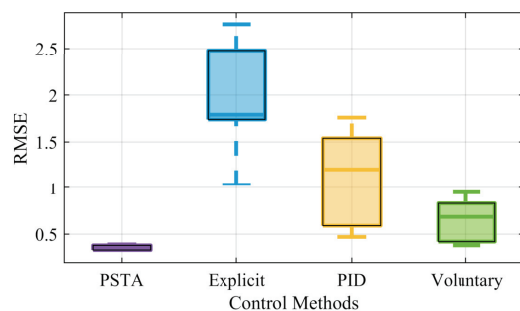


Figure 4. The average and variance of the RMSE of steady-state tracking errors for the data of six volunteers in Table 1 under different control methods.

Table 2. Computational load of PSTA, Explicit, and PID controllers.

	PSTA	Explicit	PID
Average value [μ s]	420	199	265

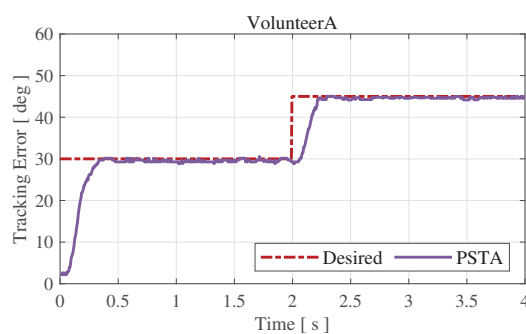


Figure 5. Tracking performance of the PSTA with changing target angles

Even with the changing target angle, the PSTA was able to maintain a low level of chattering and accurate tracking performance. These findings further confirm the superiority of the PSTA in achieving precise control of MP joints under different conditions.

6. Conclusions

This manuscript presents the proxy-based super-twisting algorithm (PSTA) and an implicit Euler realization of the PSTA for highly accurate control of MP joint through FES. The proposed method maintains the accuracy and robustness of SMC without requiring an accurate model of a MP joint, while eliminating numerical chattering. Experimental results from different volunteers demonstrate that the PSTA is more effective than other control schemes, including conventional STA, PID, and voluntary movement. Qualitative analysis shows that this method is one of the most effective and accurate in the experiments. With limited computation resources and sampling rates, the PSTA still performs well in terms of response speed and control accuracy, which are sufficient for controlling MP joints. Future investigations should include more experiments on time-varying angles of MP joints through FES as well as other muscles, such as triceps brachii, using PSTA.

Future research includes a strict Lyapunov stability check and extending the work to control multiple finger joints simultaneously. This will involve developing advanced algorithms to manage increased complexity and ensure robust, accurate control across all joints. Addressing these challenges will enhance the applicability and effectiveness of the proposed method in practical scenarios.

Author Contributions: M.Y. and K.H.; methodology, X.X.; software, H.C. and S.O.; validation, H.C.; formal analysis, X.X.; data validation, H.C.; writing—original draft preparation, H.C. and X.X.; writing—review and editing, H.C. and X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by Shenzhen Science and Technology Program under Grant KQTD20190929172545139, GXWD 20231130153844002, Guangdong Basic and Applied Basic Research Foundation, the Open Fund of Laboratory of Aerospace Servo Actuation and Transmission, and Guangdong Provincial Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Ethics Committee of the School of Engineering at Kyushu University approved the experiments (Approval no. 2021-02). The date of approval is 3 August 2021.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We thank all the volunteers who participated in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FES	Functional electrical stimulation
MP joint	Finger metacarpophalangeal joint
SMC	Sliding mode control
STA	Super-twisting algorithm
PSTA	Proxy-based super-twisting algorithm

References

1. Furlan, J.C.; Pakosh, M.; Craven, B.C.; Popovic, M.R. Insights on the Potential Mechanisms of Action of Functional Electrical Stimulation Therapy in Combination With Task-Specific Training: A Scoping Review. *Neuromodul. J. Int. Neuromodul. Soc.* **2022**, *25*, 1280–1288. [CrossRef] [PubMed]
2. Lopes, P.; Ion, A.; Baudisch, P. Impacto: Simulating Physical Impact by Combining Tactile Stimulation with Electrical Muscle Stimulation. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15, Charlotte, NC, USA, 8–11 November 2015; ACM Digital Library : New York, NY, USA, 2015; pp. 11–19. [CrossRef]
3. Remsik, A.B.; van Kan, P.L.E.; Gloe, S.; Gjini, K.; Williams, L., Jr.; Nair, V.; Caldera, K.; Williams, J.C.; Prabhakaran, V. BCI-FES With Multimodal Feedback for Motor Recovery Poststroke. *Front. Hum. Neurosci.* **2022**, *16*, 725715. [CrossRef] [PubMed]
4. Ikura, I.; Katsura, S. Servo Control of Finger MP Joint by Functional Electrical Stimulation of Lumbrical and Extensor Digitorum Muscle. In Proceedings of the 2022 15th International Conference on Human System Interaction (HSI), Melbourne, Australia, 28–31 July 2022; pp. 1–6. [CrossRef]
5. Honda, K.; Nakashima, Y.; Hua, C.; Yamamoto, M. Influence of Combined Vibration and Electrical Stimulation on Latency of Kinesthetic Illusion. *J. Robot. Mechatronics* **2023**, *35*, 823–833. [CrossRef]
6. Sharma, N.; Kirsch, N.A.; Alibeji, N.A.; Dixon, W.E. A Non-Linear Control Method to Compensate for Muscle Fatigue during Neuromuscular Electrical Stimulation. *Front. Robot.* **2017**, *4*, 68. [CrossRef]
7. Bao, X.; Sheng, Z.; Dicianno, B.E.; Sharma, N. A Tube-Based Model Predictive Control Method to Regulate a Knee Joint With Functional Electrical Stimulation and Electric Motor Assist. *IEEE Trans. Control. Syst. Technol.* **2021**, *29*, 2180–2191. [CrossRef]
8. Ajoudani, A.; Erfanian, A. A Neuro-Sliding-Mode Control With Adaptive Modeling of Uncertainty for Control of Movement in Paralyzed Limbs Using Functional Electrical Stimulation. *IEEE Trans. Biomed. Eng.* **2009**, *56*, 1771–1780. [CrossRef]
9. Sharma, N.; Stegath, K.; Gregory, C.M.; Dixon, W.E. Nonlinear Neuromuscular Electrical Stimulation Tracking Control of a Human Limb. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2009**, *17*, 576–584. [CrossRef] [PubMed]
10. Kobravi, H.R.; Erfanian, A. Decentralized adaptive robust control based on sliding mode and nonlinear compensator for the control of ankle movement using functional electrical stimulation of agonist-antagonist muscles. *J. Neural Eng.* **2009**, *6*, 046007. [CrossRef] [PubMed]
11. Nekoukar, V.; Erfanian, A. A Decentralized Modular Control Framework for Robust Control of FES-Activated Walker-Assisted Paraplegic Walking Using Terminal Sliding Mode and Fuzzy Logic Control. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 2818–2827. [CrossRef]
12. Brogliato, B.; Polyakov, A.; Efimov, D. The Implicit Discretization of the Supertwisting Sliding-Mode Control Algorithm. *IEEE Trans. Autom. Control* **2020**, *65*, 3707–3713. [CrossRef]
13. Hanan, A.; Levant, A.; Jbara, A. Low-Chattering Discretization of Homogeneous Differentiators. *IEEE Trans. Autom. Control* **2022**, *67*, 2946–2956. [CrossRef]
14. Hanan, A.; Jbara, A.; Levant, A. Homogeneous low-chattering sliding mode discretization. In Proceedings of the 2022 IEEE 61st Conference on Decision and Control (CDC), Cancun, Mexico, 6–9 December 2022; pp. 4276–4281. [CrossRef]
15. Sakaino, S.; Kitamura, T.; Mizukami, N.; Tsuji, T. High-Precision Control for Functional Electrical Stimulation Utilizing a High-Resolution Encoder. *IEEJ J. Ind. Appl.* **2021**, *10*, 124–133. [CrossRef]
16. Wang, H.; Mi, C.; Cao, Z.; Zheng, J.; Man, Z.; Jin, X.; Tang, H. Precise Discrete-Time Steering Control for Robotic Fish Based on Data-Assisted Technique and Super-Twisting-Like Algorithm. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10587–10599. [CrossRef]
17. Rakhtala, S.M. Adaptive gain super twisting algorithm to control a knee exoskeleton disturbed by unknown bounds. *Int. J. Dyn. Control* **2021**, *9*, 711–726. [CrossRef]
18. Fazli, E.; Rakhtala, S.M.; Mirrashid, N.; Karimi, H.R. Real-time implementation of a super twisting control algorithm for an upper limb wearable robot. *Mechatronics* **2022**, *84*, 102808. [CrossRef]
19. Tiaiba, H.; Daachi, M.E.H.; Madani, T. Real-time adaptive super twisting algorithm based on PSO algorithm: Application for an exoskeleton robot. In *Robotica*; Cambridge University Press: Cambridge, UK, 2024; pp. 1–26. [CrossRef]
20. Rakhtala, S.M.; Ghayebi, R. Real time control and fabrication of a soft robotic glove by two parallel sensors with MBD approach. *Med. Eng. Phys.* **2022**, *100*, 103743. [CrossRef] [PubMed]
21. Zhang, Y.; Zhang, M.; Du, F. Robust Finite-Time Command-Filtered Backstepping Control for Flexible-Joint Robots With Only Position Measurements. *IEEE Trans. Syst. Man. Cybern. Syst.* **2024**, *54*, 1263–1275. [CrossRef]
22. Zhang, M.; Jing, X.; Zhou, Z.; Sun, M. Rapid and restricted swing control via adaptive output feedback for 5-DOF tower crane systems. *Mech. Syst. Signal Process.* **2024**, *212*, 111283. [CrossRef]
23. Qian, Y.; Zhang, H.; Hu, D. Finite-time neural network-based hierarchical sliding mode antiswing control for underactuated dual ship-mounted cranes with unmatched sea wave disturbances suppression. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, in press.
24. Kikuuwe, R.; Yasukouchi, S.; Fujimoto, H.; Yamamoto, M. Proxy-based sliding mode control: A safer extension of PID position control. *IEEE Trans. Robot.* **2010**, *26*, 670–683. [CrossRef]
25. Kikuuwe, R. Sliding motion accuracy of proxy-based sliding mode control subjected to measurement noise and disturbance. *Eur. J. Control* **2021**, *58*, 114–122. [CrossRef]
26. Acary, V.; Brogliato, B.; Orlov, Y.V. Chattering-Free Digital Sliding-Mode Control with State Observer and Disturbance Rejection. *IEEE Trans. Autom. Control* **2012**, *57*, 1087–1101. [CrossRef]

27. Acary, V.; Brogliato, B. Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems. *Syst. Control Lett.* **2010**, *59*, 284–293. [CrossRef]
28. Jin, S.; Kikuuwe, R.; Yamamoto, M. Improving Velocity Feedback for Position Control by Using a Discrete-Time Sliding Mode Filtering with Adaptive Windowing. *Adv. Robot.* **2014**, *28*, 943–953. [CrossRef]
29. Haeufle, D.; Günther, M.; Bayer, A.; Schmitt, S. Hill-type muscle model with serial damping and eccentric force–velocity relation. *J. Biomech.* **2014**, *47*, 1531–1536. [CrossRef] [PubMed]
30. Ferrarin, M.; Pedotti, A. The relationship between electrical stimulus and joint torque: A dynamic model. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 342–352. [CrossRef]
31. Levant, A. Higher-order sliding modes, differentiation and output-feedback control. *Int. J. Control* **2003**, *76*, 924–941. [CrossRef]
32. Xiong, X.; Chen, G.; Lou, Y.; Huang, R.; Kamal, S. Discrete-Time Implementation of Super-Twisting Control With Semi-Implicit Euler Method. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 99–103. [CrossRef]
33. Moreno, J.A.; Osorio, M. Strict Lyapunov Functions for the Super-Twisting Algorithm. *IEEE Trans. Autom. Control* **2012**, *57*, 1035–1040. [CrossRef]
34. Moreno, J.A. On strict Lyapunov Functions for some non-homogeneous super-Twisting algorithms. *J. Frankl. Inst.* **2014**, *351*, 1902–1919. [CrossRef]
35. Liu, L.; Zheng, W.X.; Ding, S. An Adaptive SOSM Controller Design by Using a Sliding-Mode-Based Filter and its Application to Buck Converter. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 2409–2418. [CrossRef]
36. Seeber, R.; Reichhartinger, M. Conditioned Super-Twisting Algorithm for systems with saturated control action. *Automatica* **2020**, *116*, 108921. [CrossRef]
37. Seeber, R.; Horn, M. Stability proof for a well-established super-twisting parameter setting. *Automatica* **2017**, *84*, 241–243. [CrossRef]
38. Xiong, X.; Bai, Y.; Shi, R.; Kamal, S.; Wang, Y.; Lou, Y. Discrete-Time Twisting Algorithm Implementation with Implicit–Euler ZOH Discretization Method. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 3435–3439. [CrossRef]
39. Levant, A. Quasi-continuous high-order sliding-mode controllers. *IEEE Trans. Autom. Control* **2005**, *50*, 1812–1816. [CrossRef]
40. Koch, S.; Reichhartinger, M. Discrete-time equivalents of the super-twisting algorithm. *Automatica* **2019**, *107*, 190–199. [CrossRef]
41. Xiong, X.; Liu, Z.; Kamal, S.; Jin, S. Discrete-Time Super-Twisting Observer with Implicit Euler Method. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 1288–1292. [CrossRef]
42. Hua, C.; Nakashima, Y.; Yamamoto, M. On a Precise Control for MP Joint of Human Index Finger Using Functional Electrical Stimulation—Basic Characteristics of the Joint Movement. In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; pp. 1324–1327. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Applied Sciences Editorial Office
E-mail: appls@mdpi.com
www.mdpi.com/journal/appls



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-7656-3