

Special Issue Reprint

Symmetry/Asymmetry in Fuzzy Control

Edited by
José R. García-Martínez, Edson Eduardo Cruz-Miguel
and Juvenal Rodríguez-Resendiz

mdpi.com/journal/symmetry

Symmetry/Asymmetry in Fuzzy Control

Symmetry/Asymmetry in Fuzzy Control

Guest Editors

José R. García-Martínez

Edson Eduardo Cruz-Miguel

Juvenal Rodriguez-Resendiz



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

Guest Editors

José R. García-Martínez
Faculty of Electronics and
Communications Engineering
Universidad Veracruzana
Poza Rica
Mexico

Edson Eduardo Cruz-Miguel
Faculty of Electronics and
Communications Engineering
Universidad Veracruzana
Poza Rica
Mexico

Juvenal Rodriguez-Resendiz
Facultad de Ingeniería
Universidad Autónoma
de Querétaro
Querétaro
Mexico

Editorial Office

MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Symmetry* (ISSN 2073-8994), freely accessible at: https://www.mdpi.com/journal/symmetry/special_issues/F457D2XE0L.

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 978-3-7258-8945-7 (Hbk)

ISBN 978-3-7258-8946-4 (PDF)

<https://doi.org/10.3390/books978-3-7258-8946-4>

© 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
Preface	ix
José R. García-Martínez, Edson E. Cruz Miguel and Juvenal Rodríguez-Resendiz Symmetry and Asymmetry in Fuzzy Control: Recent Advances and Emerging Perspectives Reprinted from: <i>Symmetry</i> 2026 , <i>18</i> , 1362, https://doi.org/10.3390/sym18081362	
	1
Krassimir Atanassov, Veselina Bureva and Tania Pencheva Two Operations of a “Symmetric Difference” Type on Three-Dimensional Index Matrices Reprinted from: <i>Symmetry</i> 2026 , <i>18</i> , 696, https://doi.org/10.3390/sym18040696	
	8
Eknath Pore, Bhumeswar K. Patle and Sandeep Thorat UAV-Based QR Code Scanning and Inventory Synchronization System with Safe Trajectory Planning Reprinted from: <i>Symmetry</i> 2026 , <i>18</i> , 548, https://doi.org/10.3390/sym18040548	
	25
Svetlana Beryozkina and Inga Zicmane UAV-Based Hybrid Fuzzy Inference Framework for Symmetry and Asymmetry in Real-Time Air Quality Monitoring Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 2048, https://doi.org/10.3390/sym17122048	
	52
Jarosław Joostberens, Aurelia Rybak and Aleksandra Rybak Processing the Sensor Signal in a PI Control System Using an Adaptive Filter Based on Fuzzy Logic Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 1774, https://doi.org/10.3390/sym17101774	
	92
Loqmane Bencharif, Abderahim Ibsat, Hanbing Liu, Wen Qi, Hang Su and Samer Alfayad A Skill-Inspired Adaptive Fuzzy Control Framework for Symmetric Gait Tracking with Sparse Sensor Fusion in Lower-Limb Exoskeletons Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 1265, https://doi.org/10.3390/sym17081265	
	113
Abdussalam Salama, Reza Saatchi, Maryam Bagheri, Karim Shebani, Yasir Javed, Raksha Balaraman and Kavya Adhikari A Fuzzy Logic-Based eHealth Mobile App for Activity Detection and Behavioral Analysis in Remote Monitoring of Elderly People: A Pilot Study Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 988, https://doi.org/10.3390/sym17070988	
	133
José R. Rivera-Ruiz, José R. García-Martínez, Trinidad Martínez-Sánchez, Edson E. Cruz-Miguel and Ákos Odry Takagi–Sugeno–Kang Fuzzy Inference Tracking Controller for UAV Bicopter Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 759, https://doi.org/10.3390/sym17050759	
	155
Yi-Jen Mon Image-Tracking-Driven Symmetrical Steering Control with Long Short-Term Memory for Linear Charge-Coupled-Device-Based Two-Wheeled Self-Balancing Cart Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 747, https://doi.org/10.3390/sym17050747	
	181
Atanaska Georgieva, Slav I. Cholakov, Maria Vasileva and Yordanka Gudalova Fuzzy Double Yang Transform and Its Application to Fuzzy Parabolic Volterra Integro-Differential Equation Reprinted from: <i>Symmetry</i> 2025 , <i>17</i> , 606, https://doi.org/10.3390/sym17040606	
	196

José G. Pérez-Juárez, José R. García-Martínez, Alejandro Medina Santiago, Edson E. Cruz-Miguel,
 Luis F. Olmedo-García, Omar A. Barra-Vázquez and Miguel A. Rojas-Hernández
 Kinematic Fuzzy Logic-Based Controller for Trajectory Tracking of Wheeled Mobile Robots in
 Virtual Environments
 Reprinted from: *Symmetry* **2025**, *17*, 301, <https://doi.org/10.3390/sym17020301> **213**

About the Editors

José Román García-Martínez

José Román García-Martínez is a Full-Time Professor at the Faculty of Electronics and Communications Engineering, Universidad Veracruzana, Mexico. He holds a B.Sc. degree in Electronics and Communications Engineering from Universidad Veracruzana and M.Sc. and Ph.D. degrees in Instrumentation and Automatic Control and Engineering, respectively, from Universidad Autónoma de Querétaro, Mexico, where he graduated with honors for his doctoral research in intelligent control systems. His research interests include intelligent and fuzzy control, robotics, embedded systems, automation, artificial intelligence, motion control, and optimization algorithms, with particular emphasis on hybrid methodologies for mechatronic and robotic systems. He has supervised undergraduate and graduate theses and has authored and co-authored scientific articles, book chapters, and academic works in these fields. At Universidad Veracruzana, he has served as Academic Technician and Coordinator of the Control and Robotics Laboratory and is currently Head of the Academic Group UV-CA-578 “Analysis of Systems and Emerging Technologies” and Tutor Coordinator of the Electronics and Communications Engineering Program. He is a member of Mexico’s National System of Researchers (SNII), holds PRODEP Profile recognition, and is an IEEE Senior Member. He also contributes to the international scientific community through editorial activities as Guest Editor and Topic Editor in journals covering robotics, automation, intelligent control, computational intelligence, and emerging technologies.

Edson E. Cruz-Miguel

Edson E. Cruz-Miguel is a Full-Time Professor at the Faculty of Electronics and Communications Engineering, Universidad Veracruzana, Mexico. He received his degree in Electronic Engineering from the Technological Institute of Ciudad Madero in 2014, and his M.Sc. degree in Instrumentation and Automatic Control and Ph.D. degree in Engineering, both with honors, from Universidad Autónoma de Querétaro, Mexico, in 2017 and 2021, respectively. His academic activities include teaching, research, and the development of engineering projects in electronics, instrumentation, automatic control, and intelligent systems. He is a member of Mexico’s National System of Researchers (SNII) and holds PRODEP Profile recognition. He also holds Level 3 recognition in the Academic Performance Incentive Program (PEDPA) of Universidad Veracruzana.

Juvenal Rodríguez-Reséndiz

Juvenal Rodríguez-Reséndiz served as a Visiting Professor at West Virginia University in 2012. He is currently the coordinator of the Engineering Ph.D. at Querétaro State University (UAQ), in México. Herein, he has taught more than 300 courses in digital signal processing and research methodology. He is a member of the Mexican Academy of Sciences. He has the highest membership level for researchers, SNII level 3, awarded by the Mexican government. He has developed more than 50 industrial projects by linking UAQ and the government. His team has published over 300 scientific papers; his h-index is 40, and he has received more than 6000 citations. He patented more than 10 innovations. He has won national and international prizes for his academic and innovative work. He has served as advisor to more than 200 theses at the undergraduate, master’s, and doctoral levels. He has been invited to present over 30 conferences around the world. Because of his research, he has collaborated with over 30 national and international institutions.

Preface

Fuzzy control provides useful tools for engineering problems involving uncertainty and changing conditions. This Reprint presents perspectives on symmetry and asymmetry in the analysis and design of fuzzy control systems. It is intended for researchers, engineers, and students and aims to support further developments in intelligent control systems.

José R. García-Martínez, Edson Eduardo Cruz-Miguel, and Juvenal Rodriguez-Resendiz

Guest Editors

Symmetry and Asymmetry in Fuzzy Control: Recent Advances and Emerging Perspectives

José R. García-Martínez ^{1,*}, Edson E. Cruz-Miguel ¹ and Juvenal Rodríguez-Reséndiz ^{2,*}

¹ Control and Robotics Laboratory, Faculty of Electronics and Communications Engineering, Universidad Veracruzana, Poza Rica 93390, Veracruz, Mexico; edsoncruz@uv.mx

² Facultad de Ingeniería, Universidad Autónoma de Querétaro, Santiago de Querétaro 76010, Querétaro, Mexico

* Correspondence: romangarcia@uv.mx (J.R.G.-M.); juvenal@uaq.edu.mx (J.R.-R.)

1. Introduction

Fuzzy control is considered one of the most important methods in the area of intelligent control engineering because it can handle uncertainty, nonlinear behaviour, and incomplete or imprecise information. Since Zadeh made his initial contributions to fuzzy set theory, the field has seen constant progress, moving from a theoretical basis for representing uncertainty to becoming a well-established control technique that is widely used in both industrial and scientific contexts. It is now essential in fields such as robotics, industrial automation, renewable energy systems, autonomous vehicles, biomedical devices, and cyber–physical systems, where traditional modelling methods generally face significant difficulties [1].

The main reason for the effectiveness of fuzzy control is that it allows expert knowledge to be included through the use of linguistic rules, rather than depending completely on precise mathematical descriptions of the process. Fuzzy controllers are therefore particularly appealing when used in systems whose dynamics are complex due to nonlinearities, uncertainties, or changes in parameters over time. Fuzzy logic does not replace traditional control methods but instead serves as a complementary approach by offering a practical means of combining qualitative reasoning with automated decision-making, which, in turn, improves robustness and adaptability under difficult operating conditions [2,3].

Over the past ten years, fuzzy control has seen significant development as a result of advances in artificial intelligence and computational intelligence. Modern fuzzy controllers are often combined with optimisation algorithms, machine learning techniques, digital twin technology, and embedded computing methods in order to produce hybrid intelligent systems that can learn from data while still retaining the interpretability of rule-based reasoning. Because of this, fuzzy control has now been used more extensively in a variety of current technological areas, such as Industry 4.0, smart manufacturing, intelligent transportation systems, renewable energy management, and autonomous robotics, where both adaptability and computational efficiency are of the utmost importance [4,5].

A growing body of research has drawn a great deal of attention to the effects of symmetry and asymmetry when designing fuzzy controllers. Controllers having a symmetric structure are usually linked with balanced rule bases, lower design complexity, and better interpretability, which is the reason why they are chosen in situations where transparency and ease of implementation are important. In contrast, asymmetric designs offer greater modelling flexibility and are often more effective at capturing nonlinear or asymmetric system behaviours that cannot be properly illustrated using perfectly symmetric rule distributions. Instead of viewing these methods as opposing options, recent research indicates

that they should be regarded as complementary design approaches. When combined, they enable the creation of controllers that are at the same time robust, adaptable, and interpretable—qualities that are becoming increasingly desired in intelligent engineering systems [6–8].

In this context, this Special Issue, entitled “Symmetry/Asymmetry in Fuzzy Control,” offers a set of recent research papers which illustrate the current state of the field. The papers chosen focus on advances in the development of fuzzy controllers, on hybrid intelligent systems, on optimisation methods, and on a wide range of engineering applications such as robotics, autonomous systems, renewable energy, healthcare, and intelligent sensing. The issue also contains contributions concerning the mathematical aspects of symmetry in fuzzy systems and the associated computational techniques, highlighting the interdisciplinary nature of the current research. Together, these studies demonstrate how fuzzy methodologies are still being developed in order to meet the demands of new technologies, and at the same time, emphasise the complementary roles of symmetry and asymmetry in intelligent control and decision-making [6,9–15].

This paper is structured as follows: Section 2 examines the basic concepts of symmetry and asymmetry in fuzzy control, Section 3 provides an overview of recent advances in hybrid intelligent fuzzy control systems, Section 4 covers the major engineering applications which have been discussed in the literature, and Section 5 looks at the current challenges and indicates promising avenues for future research. The paper then concludes in Section 6 by summarising the key insights and outlining the broader implications of these developments for intelligent control systems.

2. Symmetry and Asymmetry in Modern Fuzzy Control

In the past few decades, fuzzy control has developed from being a framework based on rules for making decisions into a refined control technique which is capable of effectively managing uncertainty, nonlinear dynamics, and incomplete information. This development is a response to the growing need for control methods that not only give good performance but also possess robustness, adaptability, and are easy to understand. On this point, the ideas of symmetry and asymmetry have attracted a great deal of interest since they affect both the design of fuzzy controllers and their capacity to deal with complex system behaviours.

Fuzzy control works by incorporating expert knowledge through the use of linguistic variables, membership functions, inference processes, and fuzzy rules. Unlike conventional methods, which rely on a mathematical model, fuzzy controllers are capable of giving reliable control actions without needing an exact mathematical description of the system; on this account, they are particularly suitable for application in systems that are affected by uncertainty or have poorly defined dynamics, while at the same time maintaining a simple decision-making procedure which is easy for human operators to understand [8].

Previously, the majority of fuzzy controllers used symmetric structures. The membership functions were evenly spread out, the rule bases were standardised, and the responses were identical on both sides of the nominal operating conditions, thus simplifying the development of the controller, facilitating parameter adjustment, and enabling theoretical evaluations such as stability analysis. Furthermore, symmetric designs generally improve readability by keeping the relationships between the linguistic variables and the respective control actions consistent—a characteristic which is very much valued in practical engineering situations [7].

Perfect symmetry does not in every instance result in controllers that accurately represent the behaviour of real engineering systems. A great many practical processes show nonlinear features, unequal responses, time-dependent dynamics, or variations

in operating conditions throughout their operating range. Examples of such systems include robot arms, electric motor systems, renewable energy technologies, and various complex electromechanical processes because disturbances, limitations on the actuators, and changes in the operating point generally cause disproportionate system responses. In such situations, requiring symmetry can reduce the flexibility of the controller and therefore have an effect on the effectiveness of the control [16].

In order to deal with these difficulties, asymmetric fuzzy control offers greater flexibility during the design of the controller. The membership functions, the distributions of the rules, the scaling factors, and the control gains can all be changed separately in order to better suit the local operating conditions. This extra flexibility enables the controller to respond more accurately in regions where the system shows strong nonlinearity, while at the same time ensuring smooth performance in the other areas. Because of this, asymmetric designs are becoming increasingly common in adaptive control systems, fault-tolerant arrangements, autonomous devices, and other areas involving highly nonlinear or uncertain dynamics [17].

Rather than considering symmetry and asymmetry as mutually exclusive choices, current research is becoming more inclined to regard them as complementary design philosophies. Symmetric structures provide simplicity, consistency, and analytical tractability, while asymmetric configurations enhance adaptability and give a better representation of the behaviour of complex systems. Fuzzy controllers, which achieve robustness, flexibility, computational efficiency, and interpretability depending on the particular requirements of each application, can be developed by selecting an appropriate balance between these two approaches [6].

The fact that fuzzy control is becoming more and more integrated with artificial intelligence has also expanded the application of these design principles. Symmetry and asymmetry are now being applied in the design of controllers based on optimisation, in adaptive parameter tuning, in distributed control strategies, and in hybrid intelligent systems which combine fuzzy reasoning with machine learning, evolutionary optimisation, and digital twin technologies. The latest approaches indicate that further developments in fuzzy control will place less emphasis on choosing between symmetric and asymmetric formulations and will instead involve making use of the benefits of both to create intelligent controllers capable of operating efficiently in increasingly dynamic and uncertain engineering environments [4].

3. Hybrid Intelligent Fuzzy Control Architectures

The recent advances in intelligent control have significantly enhanced the importance of fuzzy systems. Fuzzy logic is no longer usually used as a stand-alone rule-based controller but is now often combined with optimisation techniques, machine learning algorithms, and artificial intelligence methods in order to produce hybrid control systems. By combining these different approaches, the individual benefits of each one are utilised, which results in controllers that provide higher accuracy, greater robustness, improved adaptability, and better interpretability while at the same time decreasing the amount of manual work normally required for designing controllers [4].

Optimisation is now one of the main factors accounting for this change. Instead of relying entirely on expert knowledge, modern fuzzy controllers are becoming more and more reliant on optimisation algorithms in order to automatically determine the membership functions, the rule bases, the scaling factors, and other controller parameters. Metaheuristic methods such as genetic algorithms (GAs), particle swarm optimisation (PSO), differential evolution (DE), and other nature-inspired optimisation techniques have consistently demonstrated their capacity to improve control performance while at the

same time making the design process simpler. Furthermore, multi-objective optimisation frameworks enable several design objectives—such as tracking accuracy, robustness, energy efficiency, and computational complexity—to be optimised together, leading to controllers that are better able to meet the practical engineering requirements [1,18].

The application of machine learning has introduced an extra level of adaptability to fuzzy control systems. A notable instance of this is the adaptive neuro-fuzzy inference system (ANFIS), which blends the capacity of artificial neural networks to learn with the interpretability of fuzzy inference, enabling controllers to improve their performance on the basis of data. More recently, deep learning and reinforcement learning have further enhanced the capabilities of hybrid fuzzy systems by permitting automatic feature extraction, online adaptation, and autonomous decision-making to take place. Consequently, fuzzy control has been widely applied to areas such as robotics, autonomous vehicles, advanced manufacturing, and various other complex engineering fields where operating conditions change continuously [4].

At the same time, the growing demand for reliable artificial intelligence has once again kindled interest in the inherent interpretability of fuzzy systems. Unlike a large number of data-driven models that operate as opaque ‘black boxes,’ fuzzy controllers demonstrate their reasoning by means of explicit linguistic rules which can be examined and understood by specialists in the field. This property is in line with the objectives of explainable artificial intelligence (XAI), which is the reason why hybrid fuzzy methods are particularly attractive for safety-critical applications where transparency, reliability, and user confidence are essential [5].

The use of hybrid fuzzy control is still expanding since it is being combined with a number of new digital technologies. Digital twins, cyber–physical systems, edge computing, and embedded intelligent platforms are now making greater use of fuzzy reasoning in order to enable real-time monitoring, distributed decision-making, and adaptive control when operating conditions change. The combination of these technologies has enabled fuzzy controllers to manage ever more interconnected and computationally demanding engineering environments while still maintaining real-time performance and operational reliability [11].

The fact that all of these developments have taken place indicates that hybrid intelligent fuzzy control has now become one of the most dynamic areas of research in modern control engineering. Future progress is expected to focus on architectures capable of combining autonomous learning, optimisation, explainability, computational efficiency, and robustness in a single framework. Such integrated solutions will be of central importance in the next generation of intelligent systems, where adaptability and transparency are likely to be just as important as the control performance itself [4].

4. Emerging Engineering Applications

The greater level of maturity achieved by fuzzy control has resulted in its application in engineering fields continuing to grow. Even though it was originally developed for use in industrial process control, fuzzy methodologies are now being incorporated into a wide variety of intelligent systems that operate under conditions of uncertainty, nonlinear dynamics, and changing environmental conditions. Since they can combine expert knowledge with computational intelligence, they allow reliable decision-making in cases where accurate mathematical models are not available, which is the reason why fuzzy control is regarded as an attractive approach to the solution of complex real-world problems [1].

In each of the various fields involved, robotics has remained one of the most actively researched areas. Fuzzy controllers have proven themselves useful for motion planning, for trajectory tracking, for navigation, for obstacle avoidance, and for human–robot in-

teraction, since it is usual in these fields to encounter uncertain sensory information and nonlinear system behaviour. Recent advances have now boosted these capabilities by combining fuzzy reasoning with optimisation techniques and machine learning algorithms, thereby increasing the autonomy, adaptability, and robustness of mobile robots, of unmanned aerial vehicles, and of collaborative robotic systems that function in dynamic environments [6,19].

The emergence of Industry 4.0 has created new opportunities for fuzzy control; in cyber–physical systems and in digital twin environments, fuzzy reasoning allows for intelligent monitoring, distributed decision-making, predictive maintenance, and process optimisation by efficiently handling uncertain information while at the same time keeping the control strategies transparent. It is due to these features that more flexible manufacturing systems have been obtained which are capable of adapting to changing operating conditions without having to compromise either reliability or computational efficiency [11].

Fuzzy control’s adaptability has also led to important advantages in the fields of energy and transportation. In the case of renewable energy, fuzzy-based methods are widely applied to maximum power point tracking, battery management, smart grid operation, wind energy conversion, electric drives, and power electronic converters because the operating conditions are constantly changing. Fuzzy reasoning is similarly used in intelligent transportation systems in order to improve a number of autonomous driving functions, including adaptive cruise control, lane-keeping assistance, collision avoidance, and traffic management. In both of these areas, the ability of fuzzy controllers to respond smoothly to uncertain and time-varying conditions has been demonstrated to be a major advantage when compared with purely model-based approaches [18,20].

Fuzzy control, besides its application in industrial automation, has in recent years been used increasingly in the areas of healthcare, biomedical engineering, environmental monitoring, and precision agriculture. The capability of fuzzy inference to deal with uncertain or imprecise information while still ensuring transparency in the decision-making process has provided advantages in a number of fields, such as medical diagnosis, rehabilitation technologies, wearable health devices, intelligent irrigation systems, and environmental management platforms. This is particularly valuable in those situations where both reliability and interpretability are essential requirements [5].

The fact that there are so many different applications demonstrates how fuzzy control has grown into a mature technology and thus enabled intelligent engineering. As developments in artificial intelligence, edge computing, cloud computing, and autonomous systems continue to change modern engineering, fuzzy methods are expected to play an even more important role in hybrid intelligent systems. Future applications will likely place a greater emphasis on not only a higher degree of autonomy and computational efficiency but also on explainability, adaptability, and on the smooth interaction between human expertise and data-driven intelligence [4].

5. Current Challenges and Future Perspectives

Even though fuzzy control has reached a high level of maturity, certain challenges still prevent its wider application in next-generation intelligent systems. The demands of modern engineering applications are such that the controllers must simultaneously provide robustness, adaptability, computational efficiency, scalability, and interpretability when operating in highly dynamic and uncertain conditions. Achieving an adequate balance among these often conflicting requirements is one of the major research problems in the field [5].

A significant problem is the increasing complexity that arises when designing fuzzy controllers. As the number of input variables and operating conditions grows, the rule

bases tend to expand rapidly, making it much more difficult to develop the controller, tune its parameters, and carry out real-time implementation. This situation is generally referred to as the curse of dimensionality and has, as a result, prompted a great deal of research into hierarchical and modular fuzzy systems, adaptive rule generation, and optimisation-based design techniques that reduce computational complexity without sacrificing control performance or interpretability [20].

At the same time, the rapid development of artificial intelligence is having a significant effect on the future of fuzzy control. Even though data-driven approaches such as machine learning and deep learning have demonstrated remarkable capabilities in terms of prediction, their lack of transparency remains a major problem—particularly in cases where safety, reliability, and accountability are essential. Fuzzy systems address this by providing clear reasoning through the use of linguistic rules, which is why combining them with explainable artificial intelligence (XAI) has become an increasingly attractive area of research. Hybrid systems that are able to combine learning, adaptation, and interpretability are therefore expected to play a fundamental role in the development of reliable intelligent control systems [16,17].

The technological advances associated with digital twins, cyber–physical systems, edge computing, and embedded intelligence have led to new capabilities being required of fuzzy controllers. Future control systems will need to deal with large quantities of heterogeneous data, enable distributed decision-making, and at the same time respond in real time even though the computing resources available are limited. The fact that these constraints are present proves that there is a requirement for lightweight algorithms, efficient optimisation techniques, and adaptive mechanisms which are able to maintain a reliable performance while the operating conditions are continually changing.

If one considers the overall situation, future developments in fuzzy control will most likely result from the effective combination of various design methods rather than from the total replacement of a single particular approach. Symmetric controller configurations will continue to offer simplicity, analytical consistency, and ease of interpretation, whereas asymmetric formulations will provide the flexibility required to cope with increasingly complex and nonlinear system behaviours. Together with advances in optimisation, machine learning, and explainable artificial intelligence, these complementary techniques are expected to contribute to the creation of more autonomous, transparent, and robust intelligent control systems. As engineering applications are expected to keep progressing in the direction of greater connectivity, autonomy, and data-driven operation, fuzzy control will remain a key enabling technology for addressing the challenges posed by next-generation intelligent systems.

6. Closing Remarks

The articles included in this Special Issue demonstrate the ongoing development of fuzzy control as a flexible approach for use in intelligent engineering systems, and together, they illustrate that symmetry and asymmetry have to be regarded as complementary design principles which facilitate the development of robust, adaptive, and interpretable controllers. Moreover, combining fuzzy control with optimisation techniques, artificial intelligence, and emerging engineering technologies enhances its significance in handling increasingly complex real-world applications.

The Guest Editors sincerely thank all the authors for their valuable contributions, the reviewers for their constructive comments, and the editorial team of Symmetry for their continuous support throughout the publication process. We hope that this Special Issue will encourage further research into symmetry and asymmetry in fuzzy control and result in new advances in intelligent engineering systems.

Acknowledgments: The authors wish to express their appreciation for the assistance and advice Rafael Rojas-Galván provided during the process of preparing this editorial.

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

References

1. Precup, R.E.; David, R.C. *Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems*; Butterworth-Heinemann: Oxford, UK, 2019. [CrossRef]
2. Deng, C.; Yang, G.H. Distributed Adaptive Fuzzy Control For Nonlinear Multiagent Systems Under Directed Graphs. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 1356–1366. [CrossRef]
3. Joostberens, J.; Rybak, A.; Rybak, A. Processing the Sensor Signal in a PI Control System Using an Adaptive Filter Based on Fuzzy Logic. *Symmetry* **2025**, *17*, 1774. [CrossRef]
4. Andonovski, G.; Leite, D.; Precup, R.E.; Gomide, F.; Pratama, M.; Škrjanc, I. Advancements in data-driven evolving fuzzy and neuro-fuzzy control: A comprehensive survey. *Appl. Soft Comput.* **2026**, *186*, 114058. [CrossRef]
5. Ouifak, H.; Idri, A. A comprehensive review of fuzzy logic based interpretability and explainability of machine learning techniques across domains. *Neurocomputing* **2025**, *647*, 130602. [CrossRef]
6. Bencharif, L.; Ibset, A.; Liu, H.; Qi, W.; Su, H.; Alfayad, S. A Skill-Inspired Adaptive Fuzzy Control Framework for Symmetric Gait Tracking with Sparse Sensor Fusion in Lower-Limb Exoskeletons. *Symmetry* **2025**, *17*, 1265. [CrossRef]
7. Hao, Y.; Wang, M.; Tang, J.; Zhang, Z.; Xiong, J. Type-2 Backstepping T-S Fuzzy Bionic Control Based on Niche Symmetry Function. *Symmetry* **2025**, *17*, 121. [CrossRef]
8. Plieskov, P.; Zinzura, V.; Petrova, K.; Soldatenko, V.; Nekrasov, A. Determining Weight Coefficients in the Control System for a Symmetry-Compensating Device Using a Fuzzy Logic Apparatus. *East.-Eur. J. Enterp. Technol.* **2026**, *2*, 85–93. [CrossRef]
9. Pérez-Juárez, J.G.; García-Martínez, J.R.; Medina Santiago, A.; Cruz-Miguel, E.E.; Olmedo-García, L.F.; Barra-Vázquez, O.A.; Rojas-Hernández, M.A. Kinematic Fuzzy Logic-Based Controller for Trajectory Tracking of Wheeled Mobile Robots in Virtual Environments. *Symmetry* **2025**, *17*, 301. [CrossRef]
10. Rivera-Ruiz, J.R.; García-Martínez, J.R.; Martínez-Sánchez, T.; Cruz-Miguel, E.E.; Ramírez-González, L.D.; Barra-Vázquez, O.A.; Odry, Á. Takagi–Sugeno–Kang Fuzzy Inference Tracking Controller for UAV Bicopter. *Symmetry* **2025**, *17*, 759. [CrossRef]
11. Dong, X.; Sun, H.; Han, Z.; Xia, Y.; Wang, H.; Mou, Q. Fault Identification Method for Photovoltaic Power Grids Based on an Improved GABP Neural Network and Fuzzy System. *Symmetry* **2025**, *17*, 1476. [CrossRef]
12. Salama, A.; Saatchi, R.; Bagheri, M.; Shebani, K.; Javed, Y.; Balaraman, R.; Adhikari, K. A Fuzzy Logic-Based eHealth Mobile App for Activity Detection and Behavioral Analysis in Remote Monitoring of Elderly People: A Pilot Study. *Symmetry* **2025**, *17*, 988. [CrossRef]
13. Mon, Y.J. Image-Tracking-Driven Symmetrical Steering Control with Long Short-Term Memory for Linear Charge-Coupled-Device-Based Two-Wheeled Self-Balancing Cart. *Symmetry* **2025**, *17*, 747. [CrossRef]
14. Atanassov, K.; Bureva, V.; Pencheva, T. Two Operations of a “Symmetric Difference” Type on Three-Dimensional Index Matrices. *Symmetry* **2026**, *18*, 696. [CrossRef]
15. Georgieva, A.; Cholakov, S.I.; Vasileva, M.; Gudalova, Y. Fuzzy Double Yang Transform and Its Application to Fuzzy Parabolic Volterra Integro-Differential Equation. *Symmetry* **2025**, *17*, 606. [CrossRef]
16. Xia, Y.; Lendek, Z.; Precup, R.E.; Agarwal, R.K.; Rudas, I.J. Suction Cup-Type Prescribed Performance Fault-Tolerant Fuzzy Control for Nonlinear Systems Considering Actuator Power. *IEEE Trans. Cybern.* **2026**, *56*, 347–356. [CrossRef] [PubMed]
17. Li, D.; Lv, B.; Li, S.; Wang, H.; Liu, L. Adaptive Fuzzy Control for Nonlinear Multiagent Systems Subject to Multiple Constraints Under Deception Attacks. *IEEE Trans. Fuzzy Syst.* **2026**, *34*, 1501–1511. [CrossRef]
18. AboRas, K.M.; Alshehri, M.H.; Megahed, A.I. Eel and Grouper Optimization-Based Fuzzy FOPI-TID^μ-PIDA Controller for Frequency Management of Smart Microgrids Under the Impact of Communication Delays and Cyberattacks. *Mathematics* **2025**, *13*, 2040. [CrossRef]
19. Mora-González, I.; Cruz-Miguel, E.E.; Martínez-Sánchez, T.; Santos-Flores, Z.E.; Rojas-Galván, R.; Barra-Vázquez, O.A.; Méndez-Ramírez, C.T.; Carrillo-Serrano, R.V.; García-Martínez, J.R. A Hybrid Master–Slave Fuzzy Cascade Control Strategy for Two-Wheeled Self-Balancing Robot with Wheel Synchronization. *Robotics* **2026**, *15*, 110. [CrossRef]
20. Renjin; Liyunhe; Gongshenggao; Biantao. A hybrid fuzzy logic-based energy management strategy for grid-connected photovoltaic microgrids with energy storage optimization. *Comput. Electr. Eng.* **2026**, *131*, 110977. [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

Two Operations of a “Symmetric Difference” Type on Three-Dimensional Index Matrices

Krassimir Atanassov ¹, Veselina Bureva ² and Tania Pencheva ^{1,*}

¹ Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; k.t.atanassov@gmail.com

² Laboratory of Intelligent Systems, Burgas State University “Prof. Dr. Assen Zlatarov”, 8010 Burgas, Bulgaria; vbureva@btu.bg

* Correspondence: tania.pencheva@biomed.bas.bg

Abstract

In the current research, we introduce two operations of a “symmetric difference” type over three-dimensional extended index matrices, and investigate some of their basic properties. An example of the implementation of symmetric difference-type operations is presented in the field of relational databases, aiming to demonstrate the operations’ efficiency by comparing sets with dissimilar attributes.

Keywords: index matrix; matrix operation; symmetric difference

1. Introduction

The concept of an Index Matrix (IM) was introduced in [1] and discussed in a series of papers and in books [2,3]. IMs are extensions of the well-known mathematical object “matrix” (see, e.g., [4,5]). Each of operation’s addition, subtraction, and multiplication, defined over standard matrices, requires specific conditions in order for these operations to be correctly performed. The introduction of IMs was inspired by the idea that matrices are redefined in such a way that all operations on them are valide and applicable, regardless of the differences in their dimensions. By giving the matrices and operations on them a more complex form, these conditions ensure greater possibilities for matrix calculus, as well as more applications. Firstly, all operations were developed in the two-dimensional (2D) case [2,3]. Later on, certain results obtained for 2D IMs became the basis for the development of the apparatus for three (3D)- and n -dimensional matrices, namely 3-DIMs and n -DIMS, respectively. The analogues of operations defined on 2-DIMs for the case of 3-DIMs have been elaborated in [2,3]. The theory of IMs has been further explored in [6].

Following [2,3], we will mention that one IM can be populated by the following: elements of the set $\{0, 1\}$, real or complex numbers, variables, predicates, functions, or even whole IMs. In the latter case, the IM is called an Extended IM (EIM). When elements of a given IM are intuitionistic fuzzy pairs, the object is called an intuitionistic fuzzy IM (IFIM) (see [2]). Therefore, each IFIM is a particular case of an EIM. The EIMs were extended to three- and n -dimensional EIMs (3-DEIMs and n -DEIMs).

In the present paper, in Section 2, we give short remarks on 3-DEIMs, in Section 3 we introduce two new operations of a “symmetric difference” type over two 3-DEIMs, and investigate some of their basic properties. In Section 4, we present an example of the implementation of “symmetric difference” operations in the field of relational databases. Finally, in the Conclusion, we outline the main achievements of the current investigation, and discuss some additional ideas for future research.

List of abbreviations used in the current investigation:

IM—index matrix

EIM—extended index matrix

3-DEIM—three- dimensional extended index matrix

n-DEIM—n-dimensional extended index matrix

IFIM—intuitionistic fuzzy index matrix

NGB theory—von Neumann–Bernays–Gödel set theory

2. Preliminary

Let $\mathcal{I}$ be a fixed set of indices and $\mathcal{X}$ be a fixed set of objects. Following [2,3], we will call “3-DEIM” with index sets K, L , and H ($K, L, H \subset \mathcal{I}$) the object:

$$[K, L, H, \{a_{k_i, l_j, h_g}\}] \equiv \left\{ \begin{array}{c|ccccc} h_g & l_1 & \cdots & l_j & \cdots & l_n \\ \hline k_1 & a_{k_1, l_1, h_g} & \cdots & a_{k_1, l_j, h_g} & \cdots & a_{k_1, l_n, h_g} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_i & a_{k_i, l_1, h_g} & \cdots & a_{k_i, l_j, h_g} & \cdots & a_{k_i, l_n, h_g} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_m & a_{k_m, l_1, h_g} & \cdots & a_{k_m, l_j, h_g} & \cdots & a_{k_m, l_n, h_g} \end{array} \middle| h_g \in H \right\}$$

$$\equiv \left\{ \begin{array}{c|ccccc} h_1 & l_1 & \cdots & l_j & \cdots & l_n \\ \hline k_1 & a_{k_1, l_1, h_1} & \cdots & a_{k_1, l_j, h_1} & \cdots & a_{k_1, l_n, h_1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_i & a_{k_i, l_1, h_1} & \cdots & a_{k_i, l_j, h_1} & \cdots & a_{k_i, l_n, h_1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_m & a_{k_m, l_1, h_1} & \cdots & a_{k_m, l_j, h_1} & \cdots & a_{k_m, l_n, h_1} \end{array} \right\},$$

$$\begin{array}{c|ccccc} h_2 & l_1 & \cdots & l_j & \cdots & l_n \\ \hline k_1 & a_{k_1, l_1, h_2} & \vdots & a_{k_1, l_j, h_2} & \cdots & a_{k_1, l_n, h_2} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_i & a_{k_i, l_1, h_2} & \vdots & a_{k_i, l_j, h_2} & \cdots & a_{k_i, l_n, h_2} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_m & a_{k_m, l_1, h_2} & \cdots & a_{k_m, l_j, h_2} & \cdots & a_{k_m, l_n, h_2} \end{array},$$

$$\vdots$$

$$\begin{array}{c|ccccc} h_f & l_1 & \cdots & l_j & \cdots & l_n \\ \hline k_1 & a_{k_1, l_1, h_f} & \cdots & a_{k_1, l_j, h_f} & \cdots & a_{k_1, l_n, h_f} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_i & a_{k_i, l_1, h_f} & \cdots & a_{k_i, l_j, h_f} & \cdots & a_{k_i, l_n, h_f} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ k_m & a_{k_m, l_1, h_f} & \cdots & a_{k_m, l_j, h_f} & \cdots & a_{k_m, l_n, h_f} \end{array} \right\}.$$

where $K = \{k_1, k_2, \dots, k_m\}$, $L = \{l_1, l_2, \dots, l_n\}$, $H = \{h_1, h_2, \dots, h_f\}$, and for $1 \leq i \leq m$, $1 \leq j \leq n$, $1 \leq g \leq f$: $a_{k_i, l_j, h_g} \in \mathcal{X}$.

For the 3-DEIMs

$$A = [K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_g^A}\}]$$

and

$$B = [K^B, L^B, H^B, \{b_{k_p^B, l_q^B, h_r^B}\}]$$

operations that are analogous to the usual matrix operations of addition and multiplication are defined [2], as well as more specific operations.

Let $\circ, * : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$. Let e_0 and e_* be unit elements of $\mathcal{X}$ with respect to operations $\circ$ and $*$, respectively.

The following definitions are given according to [2,3]:

Addition

$$A \oplus_{(\circ)} B = [K^A \cup K^B, L^A \cup L^B, H^A \cup H^B, \{c_{t_u, v_w, x_y}\}],$$

where

$$c_{t_u, v_w, x_y} = \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{if } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A - H^B \\ & \text{or } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A - L^B \text{ and } x_y = h_g^A \in H^A \\ & \text{or } t_u = k_i^A \in K^A - K^B, v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A \\ b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B - H^A \\ & \text{or } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B - L^A \text{ and } x_y = h_r^B \in H^B \\ & \text{or } t_u = k_p^B \in K^B - K^A, v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B \\ a_{k_i^A, l_j^A, h_g^A} \circ b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_i^A = k_p^B \in K^A \cap K^B, v_w = l_j^A = l_q^B \in L^A \cap L^B \\ & \text{and } x_y = h_g^A = h_r^B \in H^A \cap H^B \\ e_0, & \text{otherwise} \end{cases}$$

Termwise multiplication

$$A \otimes_{(\circ)} B = [K^A \cap K^B, L^A \cap L^B, H^A \cap H^B, \{c_{t_u, v_w, x_y}\}],$$

where $c_{t_u, v_w, x_y} = a_{k_i^A, l_j^A, h_g^A} \circ b_{k_p^B, l_q^B, h_r^B}$, for $t_u = k_i^A = k_p^B \in K^A \cap K^B, v_w = l_j^A = l_q^B \in L^A \cap L^B$ and $x_y = h_g^A = h_r^B \in H^A \cap H^B$.

Multiplication

$$A \odot_{(\circ, *)} B = [K^A \cup (K^B - L^A), L^B \cup (L^A - K^B), H^A \cup H^B, \{c_{t_u, v_w, x_y}\}],$$

where

$$c_{t_u, v_w, x_y} = \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{if } (t_u = k_i^A \in K^A \\ & \text{and } v_w = l_j^A \in L^A - K^B - L^B \\ & \text{and } x_y = h_g^A \in H^A) \\ & \text{or } (t_u = k_i^A \in K^A - K^B - L^B \\ & \text{and } v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A) \\ b_{k_p^B, l_q^B, h_r^B}, & \text{if } (t_u = k_p^B \in K^B \\ & \text{and } v_w = l_q^B \in L^B - K^A - L^A \\ & \text{and } x_y = h_r^B \in H^B) \\ & \text{or } (t_u = k_p^B \in K^B - K^A - L^A \\ & \text{and } v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B) \\ l_j^A \circ_{k_p^B \in K^A \cap L^B} (a_{k_i^A, l_j^A, h_g^A} * b_{k_p^B, l_q^B, h_r^B}), & \text{if } t_u = k_i^A \in K^A \text{ and } v_w = l_q^B \in L^B \\ & \text{and } x_y = h_g^A = h_r^B \in H^A \cap H^B \\ e_0, & \text{otherwise} \end{cases}$$

Structural subtraction

$$A \ominus B = [K^A - K^B, L^A - L^B, H^A - H^B, \{c_{t_u, v_w, x_y}\}],$$

where “−” is the set-theoretic difference operation and $c_{t_u, v_w, x_y} = a_{k_i, l_j, h_g}$, for $t_u = k_i \in K^A - K^B$, $v_w = l_j \in L^A - L^B$ and $x_y = h_g \in H^A - H^B$.

In [3], three other operations of “multiplication” are defined, but these will not be discussed here because they are not related to the topic of this research.

3. Definitions of Two New Operations “Symmetric Difference” over Two 3-DIMs and Some of Their Properties

In [7], two operations of “symmetric difference” are defined over 2-DIMs. In this investigation, we extend their definition to 3-DIMs.

Let the 3-DEIMs A and B defined above be given. For those, we define the following operations:

First operation “symmetric difference”:

$$A \overset{\bullet}{-}_1 B = [K_A \div K_B, L_A \div L_B, H_A \div H_B, \{c_{t_u, v_w, x_y}\}],$$

where for every two arbitrary (standard set-theoretical) sets Y and Z :

$$Y \div Z = (Y - Z) \cup (Z - Y)$$

and

$$c_{t_u, v_w, x_y} = \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{if } t_u = k_i^A \in K^A - K^B, v_w = l_j^A \in L^A - L^B, x_y = h_g^A \in H^A - H^B \\ b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_p^B \in K^B - K^A, v_w = l_q^B \in L^B - L^A, x_y = h_r^B \in H^B - H^A \\ e_0, & \text{otherwise} \end{cases}$$

Second operation “symmetric difference”:

$$A \overset{\bullet}{-}_2 B = [K_A \cup K_B, L_A \cup L_B, H_A \cup H_B, \{c_{t_u, v_w, x_y}\}],$$

where

$$c_{t_u, v_w, x_y} = \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{if } t_u = k_i^A \in K^A - K^B, v_w = l_j^A \in L^A, x_y = h_g^A \in H^A \\ & \text{or } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A - L^B, x_y = h_g^A \in H^A \\ & \text{or } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A, x_y = h_g^A \in H^A - H^B \\ b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_p^B \in K^B - K^A, v_w = l_q^B \in L^B, x_y = h_r^B \in H^B, \\ & \text{or } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B - L^A, x_y = h_r^B \in H^B \\ & \text{or } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B, x_y = h_r^B \in H^B - H^A \\ e_0, & \text{otherwise} \end{cases}$$

For a better understanding of the two “symmetric difference”-type operations developed here, they were visualized as two IMs that share a subset of identical indices and identical values, as illustrated in Figure 1. In the following Figures 2 and 3, graphical representations of the first operation “symmetric difference” (Figure 2) and of the second operation “symmetric difference” (Figure 3) are given.

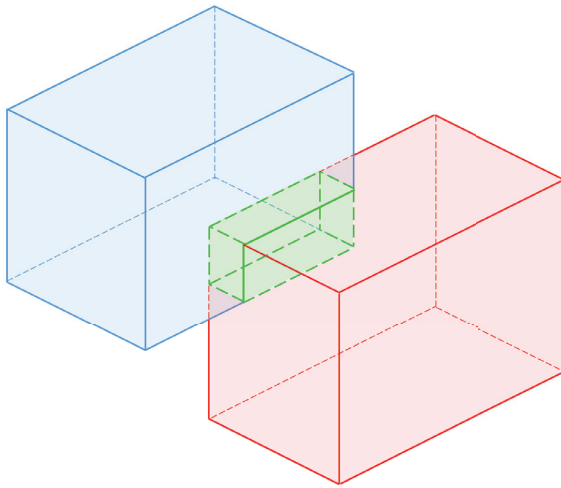


Figure 1. Intersection between two IMs sharing a subset of identical indices and identical values).

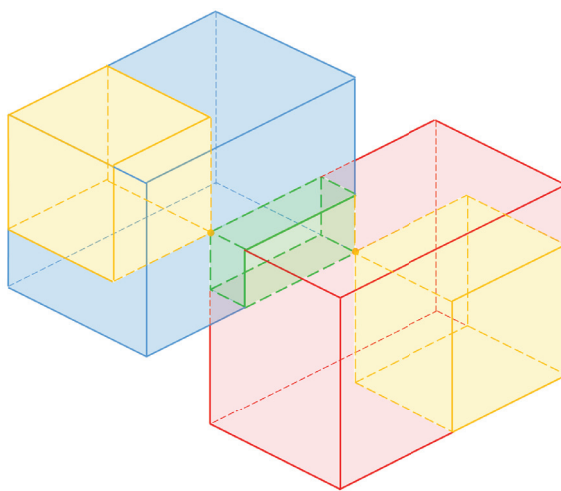


Figure 2. A graphical representation of the first operation “symmetric difference”.

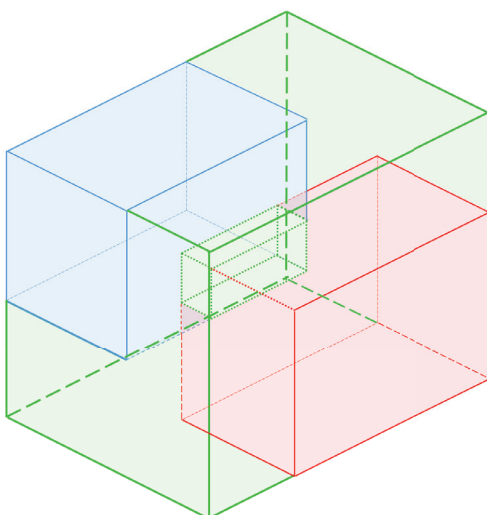


Figure 3. A graphical representation of the second operation “symmetric difference”.

Let the operation $\times$ denote a Cartesian product between two (ordinary) sets.

Theorem 1. For every two 3-DEIMs A and B , for each operation $\circ : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$ and for operation $\# : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$ defined for two arbitrary $x, y \in \mathcal{X}$ by $x \# y = e_{\#} \in \mathcal{X}$, the following equalities:

$$\begin{aligned} A \overset{\bullet}{-}_1 B &= (A \ominus B) \oplus_{(\circ)} (b \ominus A), \\ A \overset{\bullet}{-}_2 B &= A \oplus_{(\#)} B \end{aligned}$$

hold.

Proof. From the definitions of operations $\ominus$ and $\oplus_{\circ}$, we obtain:

$$\begin{aligned} (A \ominus B) \oplus_{(\circ)} (B \ominus A) &= [K^A - K^B, L^A - L^B, H^A - H^B, \{a_{k_i^A, l_j^A, h_g^A}\}] \\ &\quad \oplus_{(\circ)} [K^B - K^A, L^B - L^A, H^B - H^A, \{b_{k_p^B, l_q^B, h_r^B}\}] \\ &= [(K^A - K^B) \cup (K^B - K^A), (L^A - L^B) \cup (L^B - L^A), \\ &\quad (H^A - H^B) \cup (H^B - H^A), \{c_{t_u, v_w, x_y}\}] \\ &= [K^A \div K^B, L^A \div L^B, H^A \div H^B, \{c_{t_u, v_w, x_y}\}] \\ &= A \overset{\bullet}{-}_1 B, \end{aligned}$$

where

$$c_{t_u, v_w, x_y} = \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{for } t_u = k_i^A \in K^A - K^B \text{ and } v_w = l_j^A \in L^A - L^B \\ & \text{and } x_y = h_g^A \in H^A - H^B \\ b_{k_p^B, l_q^B, h_r^B}, & \text{for } t_u = k_p^B \in K^B - K^A \text{ and } v_w = l_q^B \in L^B - L^A \\ & \text{and } x_y = h_r^B \in H^B - H^A \end{cases}.$$

From the definitions of operations $\oplus_{(\#)}$ and the condition $x \# y = e_{\#}$, we obtain:

$$A \oplus_{(\#)} B = [K^A \cup K^B, L^A \cup L^B, H^A \cup H^B, \{c_{t_u, v_w, x_y}\}] = A \overset{\bullet}{-}_2 B,$$

because, by definition,

$$\begin{aligned} c_{t_u, v_w, x_y} &= \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{if } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A - H^B \\ & \text{or } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A - L^B \text{ and } x_y = h_g^A \in H^A \\ & \text{or } t_u = k_i^A \in K^A - K^B, v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A \\ b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B - H^A \\ & \text{or } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B - L^A \text{ and } x_y = h_r^B \in H^B \\ & \text{or } t_u = k_p^B \in K^B - K^A, v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B \\ a_{k_i^A, l_j^A, h_g^A} \# b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_i^A = k_p^B \in K^A \cap K^B, v_w = l_j^A = l_q^B \in L^A \cap L^B \\ & \text{and } x_y = h_g^A = h_r^B \in H^A \cap H^B \\ e_0, & \text{otherwise} \end{cases} \\ &= \begin{cases} a_{k_i^A, l_j^A, h_g^A}, & \text{if } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A - H^B \\ & \text{or } t_u = k_i^A \in K^A, v_w = l_j^A \in L^A - L^B \text{ and } x_y = h_g^A \in H^A \\ & \text{or } t_u = k_i^A \in K^A - K^B, v_w = l_j^A \in L^A \text{ and } x_y = h_g^A \in H^A \\ b_{k_p^B, l_q^B, h_r^B}, & \text{if } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B - H^A \\ & \text{or } t_u = k_p^B \in K^B, v_w = l_q^B \in L^B - L^A \text{ and } x_y = h_r^B \in H^B \\ & \text{or } t_u = k_p^B \in K^B - K^A, v_w = l_q^B \in L^B \text{ and } x_y = h_r^B \in H^B \\ e_0, & \text{otherwise} \end{cases} \end{aligned}$$

that proves the theorem. $\square$

Let $\mathcal{M}$ be the set of all 3-DEIMs with elements from $\mathcal{X}$. We must mention that when $\mathcal{X}$ is a set (class) in the sense of von Neumann–Bernays–Gödel (NGB)-set theory (see, e.g., [8–10]), $\mathcal{M}$ will be a set if its matrices have numerical elements, or a class when its matrices have arbitrary predicates as elements.

Let

$$I_{\emptyset} = [\emptyset, \emptyset, \perp],$$

where the symbol $\perp$ denotes the absence of objects.

Theorem 2. $\langle \mathcal{M}, \overset{\bullet}{-}_1, I_{\emptyset} \rangle$ is a commutative group.

Proof. Following the definition of a commutative group (see, [11]), we will check the respective conditions. First, from the definition of operation “ $\overset{\bullet}{-}_1$ ”, for every two 3-DEIMs $A, B \in \mathcal{M}$, it follows that $A \overset{\bullet}{-}_1 B \in \mathcal{M}$.

Second, using the well-known equality (see, e.g., [10]) for every three sets X, Y , and Z :

$$(X \div Y) \div Z = X \div (Y \div Z),$$

for every three 3-DEIMs $A, B, C \in \mathcal{M}$, where

$$C = [K^C, L^C, H^C, \{c_{\alpha, \beta, \gamma}\}]$$

we obtain

$$\begin{aligned} (A \overset{\bullet}{-}_1 B) \overset{\bullet}{-}_1 C &= ([K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_s^A}\}] \overset{\bullet}{-}_1 [K^B, L^B, H^B, \{b_{k_p^B, l_q^B, h_r^B}\}]) \\ &\quad \overset{\bullet}{-}_1 [K^C, L^C, H^C, \{c_{\alpha, \beta, \gamma}\}] \\ &= [K^A \div K^B, L^A \div L^B, H^A \div H^B, \{d_{t_u, v_w, x_y}\}] \overset{\bullet}{-}_1 [K^C, L^C, H^C, \{c_{\alpha, \beta, \gamma}\}] \end{aligned}$$

(where each element d_{t_u, v_w, x_y} is some element $a_{k_i^A, l_j^A, h_s^A}$ or some element $b_{k_p^B, l_q^B, h_r^B}$)

$$= [(K^A \div K^B) \div K^C, (L^A \div L^B) \div L^C, (H^A \div H^B) \div H^C, \{f_{\delta, \varepsilon, \zeta}\}]$$

(where each element $f_{\delta, \varepsilon, \zeta}$ is some element $a_{k_i^A, l_j^A, h_s^A}$ or some element $b_{k_p^B, l_q^B, h_r^B}$, or some element $c_{\alpha, \beta, \gamma}$)

$$\begin{aligned} &= [K^A \div (K^B \div K^C), L^A \div (L^B \div L^C), H^A \div (H^B \div H^C), \{f_{\delta, \varepsilon, \zeta}\}] \\ &= [K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_s^A}\}] \overset{\bullet}{-}_1 [K^B \div K^C, L^B \div L^C, H^B \div H^C, \{s_{\eta, \theta, \kappa}\}] \end{aligned}$$

(where each element $s_{\eta, \theta, \kappa}$ is some element $b_{k_p^B, l_q^B, h_r^B}$ or some element $c_{\alpha, \beta, \gamma}$)

$$\begin{aligned} &= [K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_s^A}\}] \overset{\bullet}{-}_1 ([K^B, L^B, H^B, \{b_{k_p^B, l_q^B, h_r^B}\}] \overset{\bullet}{-}_1 [K^C, L^C, H^C, \{c_{\alpha, \beta, \gamma}\}]) \\ &= A \overset{\bullet}{-}_1 (B \overset{\bullet}{-}_1 C). \end{aligned}$$

Third, for the 3-DEIM $I_{\emptyset}$ and for an arbitrary $A \in \mathcal{M}$, we obtain:

$$\begin{aligned} A \overset{\bullet}{-}_1 I_{\emptyset} &= [K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_s^A}\}] \overset{\bullet}{-}_1 [\emptyset, \emptyset, \perp] \\ &= [K^A \div \emptyset, L^A \div \emptyset, H^A \div \emptyset, \{d_{t_u, v_w, x_y}\}] \end{aligned}$$

(where each element d_{t_u, v_w, x_y} coincides with the element $a_{k_i^A, l_j^A, h_g^A}$ from A for $t_u = k_i^A$, $v_w = l_j^A$, $x_y = h_g^A$)

$$= [K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_g^A}\}] = A.$$

Fourth, from the well-known equality $X \div Y = Y \div X$, for two sets, it follows that

$$\begin{aligned} A \overset{\bullet}{-}_1 B &= [K^A, L^A, H^A, \{a_{k_i^A, l_j^A, h_g^A}\}] \overset{\bullet}{-}_1 [K^B, L^B, H^B, \{b_{k_p^B, l_q^B, h_r^B}\}] \\ &= [K^A \div K^B, L^A \div L^B, H^A \div H^B, \{c_{t_u, v_w, x_y}\}] \\ &= [K^B \div K^A, L^B \div L^A, H^B \div H^A, \{c_{t_u, v_w, x_y}\}] \\ &= B \overset{\bullet}{-}_1 A. \end{aligned}$$

Finally,

$$\begin{aligned} A \overset{\bullet}{-}_1 A &= [K^A \div K^A, L^A \div L^A, H^A \div H^A, \{c_{t_u, v_w, x_y}\}] \\ &= [\emptyset, \emptyset, \emptyset, \{c_{t_u, v_w, x_y}\}] \\ &= [\emptyset, \emptyset, \emptyset, \{\perp\}] \\ &= I_{\emptyset} \end{aligned}$$

(due to a lack of indices, element c_{t_u, v_w, x_y} must be $\perp$). The theorem is proven. $\square$

Theorem 3. $\langle \mathcal{M}, \overset{\bullet}{-}_2, I_{\emptyset} \rangle$ is a commutative monoid.

Proof. The proof of this assertion is similar to the proof of Theorem 2, but without the final, fifth step of the proof of Theorem 2. Therefore, as in the previous proof, we can prove that $\langle \mathcal{M}, \overset{\bullet}{-}_2, I_{\emptyset} \rangle$ is a commutative monoid, but this object is not a group due to the fact that there is no 3-DEIM $B \in \mathcal{M}$, such that for some 3-DEIM $A \in \mathcal{M}$

$$A \overset{\bullet}{-}_2 B = I_{\emptyset}.$$

This fact follows from the set-theoretical fact that there is no set X , such that for some non-empty set Y to hold, $X \cup Y = \emptyset$. $\square$

As mentioned in the Introduction, each IFIM is a particular case of an EIM. Therefore, the two new operations developed in Section 3 can be applied to IFIMs, and therefore to IMs with elements from fuzzy sets, where there is only the degree of agreement, and each intuitionistic fuzzy pair $\langle a, b \rangle$ might be reduced to only the degree of agreement a and the degree of disagreement is $1 - a$, while the degree of uncertainty from the intuitionistic fuzzy logic is 0.

4. “Symmetric Difference” Operations Implementation over Databases

The current section presents an example of the above elaborated “symmetric difference”-type operations in the field of databases. It is known that the relational databases are based on relational algebra and relational calculus. Thereafter, there is a close relation between the group of set operations in the relational databases theory and “symmetric difference”-type operations. Relational algebra uses a set of operators to create new relations from existing ones, the main three of which are *union*, *intersection*, and *difference* [12–14]. The *union* operation combines the data from two tables with a similar structure. The *intersection* operation finds equal records in both tables. The operation *difference* is used for returning the data from table A that do not exist in table B , and vice versa.

The *difference* operation will be the focus of the current investigation. Venn diagrams are frequently used to visualize database set operations and types of joins. Venn diagrams of different types of *difference* operation over sets are presented in Figure 4.

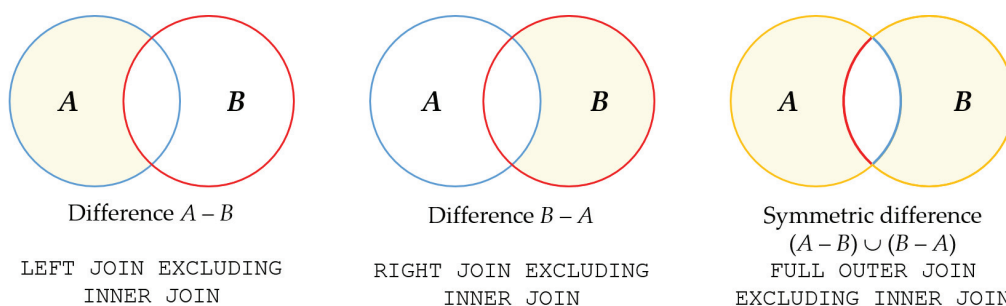


Figure 4. Venn diagrams of the set operation *difference* between sets A and B from relational algebra.

In database management systems (DBMSs), the *difference* operation is used to identify records that differ between two union-compatible tables. In structured query language (SQL), this operation is implemented through operators EXCEPT or MINUS, depending on the specific DBMS. In addition to set operations, various types of relationships between two tables can be identified using JOIN statements. The present discussion focuses on the more complex case in which the tables being compared belong to two different databases.

An example of data sources presented in the form of two databases are firstly created. In this case, the *BooksRepository* and *BookStoresChain* databases have a similar structure. Databases are created using SQL Server Express 2022 and SQL Server Management Studio 21 [12,15,16]. Database *BooksRepository* contains *Repositories*, *Places*, *MaterialsAvailability*, and *Materials*, tables. Database *BookStoresChain* includes *Bookstores*, *Regions*, *BookState*, *Books* and tables. The diagrams of both databases are presented in Figure 5. The data from *BooksRepository* and *BookStoresChain* is queried by INNER JOIN statements and is presented in Figure 6.

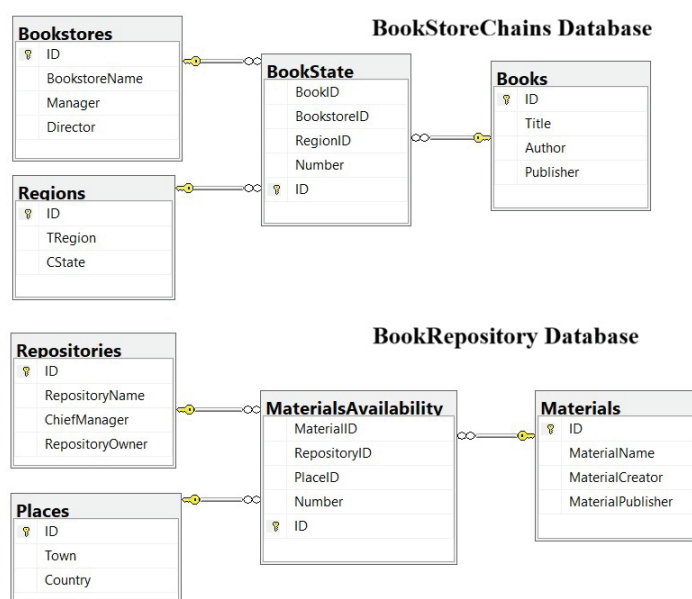


Figure 5. The diagrams of *BookStoresChain* databases (upper diagram) and *BooksRepository* (bottom diagram).

The first query, presented in Figure 6, selects the columns *MaterialName* from *Materials* table, *RepositoryName* from *Repositories* table, *Town* from *Places* table and *Number* from

MaterialsAvailability table from *BooksRepository* database. The second query, presented in Figure 6, selects the columns *Title* from *Books* table, *BookStoreName* from *BookStores* table, *TRegion* from *Regions* table and *Number* from *BookStates* table from *BookStoresChain* database.

	MaterialName	Number	RepositoryName	Town
1	Mathematics	22	Britta	Plovdiv
2	Sky	5	Britta	Plovdiv
3	One day	18	Britta	Plovdiv
4	Mathematics	11	Once	London
5	Sky	4	Once	London
6	One day	8	Once	London
7	Mathematics	24	Ozone	Istanbul
8	Sky	6	Ozone	Istanbul
9	One day	33	Ozone	Istanbul
10	Mathematics	2	Britta	Istanbul
11	Sky	55	Britta	Istanbul
12	One day	7	Britta	Istanbul
13	Algorithms	22	Britta	Porto
14	Little Bear	5	Britta	Izmir
15	Silence	18	Britta	Istanbul
16	Informatics	11	Once	London
17	Readers	4	Once	Plovdiv
18	Foods and Drinks	8	Once	Sofia
19	Foods and Drinks	24	Ozone	Burgas
20	Readers	6	Ozone	Porto
21	Informatics	33	Ozone	Izmir
22	Silence	2	Britta	Istanbul
23	Little Bear	55	Britta	London
24	Algorithms	7	Britta	Plovdiv
25	Maths	2	Orange	Warsaw
26	Bugs	55	Orange	Porto
27	Little Mouse	7	Once	Izmir

	Title	Number	BookstoreName	TRegion
1	Mathematics	22	Britta	Plovdiv
2	Sky	5	Britta	Plovdiv
3	One day	18	Britta	Plovdiv
4	Mathematics	11	HBS	London
5	Sky	4	HBS	London
6	One day	8	HBS	London
7	Mathematics	24	Ozone	Istanbul
8	Sky	6	Ozone	Istanbul
9	One day	33	Ozone	Istanbul
10	Mathematics	2	Britta	Istanbul
11	Sky	55	Britta	Istanbul
12	One day	7	Britta	Istanbul
13	Algorithms	22	Britta	Ankara
14	Little Bear	5	Britta	Izmir
15	Silence	18	Britta	Istanbul
16	Programming	11	HBS	London
17	Moon	4	HBS	Plovdiv
18	Yoga	8	HBS	Sofia
19	Yoga	24	Ozone	Burgas
20	Moon	6	Ozone	Ankara
21	Programming	33	Ozone	Izmir

Figure 6. Examples of data from the *BooksRepository* and *BookStoresChains* databases, selected by INNER JOIN queries.

The “symmetric difference” operation can be effectively illustrated in the context of relational databases. By applying it, reports can be generated to present intersections, unions, and differences between datasets. These standard operations are typically performed on two-dimensional tables. Therefore, an example of the symmetric difference operation is first presented for 2D data. Subsequently, the discussion is extended to the case of 3D data.

The implementation of “symmetric difference” operation in a 2D case is demonstrated through the *BooksRepository* and *BookStoresChains* databases. The aim is that a list of books that are available only in the first database or only in the second database are extracted. The SQL statement to realize that is presented below:

SQL query:

```
SELECT ID, MaterialName, MaterialCreator, MaterialPublisher
FROM BookRepository.dbo.materials
EXCEPT
SELECT ID, Title, Author, Publisher
FROM BookStoreChains.dbo.books
UNION
(SELECT ID, Title, Author, Publisher
FROM BookStoreChains.dbo.books
EXCEPT
SELECT ID, MaterialName, MaterialCreator, MaterialPublisher
FROM BookRepository.dbo.materials)
```

The result of the SQL statement is presented in Figure 7. It contains nine books that are available only in the *BookRepository* database and three books that are available only in the *BookStoreChains* database.

ID	MaterialName	MaterialCreator	MaterialPublisher
7	Informatics	George Randy	Orion
7	Programming	George Randy	Orion
8	Moon	Mary Smith	Action
8	Readers	Mary Smith	Action
9	Foods and Drinks	Andy Freedom	Ambari
9	Yoga	Andy Freedom	Ambari
10	Little Mouse	Henry Loren	Orion
11	Bugs	Sarah Rom	Action
12	Maths	Rita Sorah	Ambari

Figure 7. “Symmetric difference” operation in the 2D case.

Further, the “symmetric difference” operation is further illustrated in the context of multidimensional databases. The objective is to compare several datasets with differing attributes and to highlight elements present in one dataset but absent in another. The construction of 3D structures is again based on the *BookStoresChain* and *BooksRepository* databases. Two OLAP (Online Analytical Processing) cubes are implemented, and the resulting tables are used for subsequent data transformation. The initial two-dimensional tables are preprocessed and transformed into 3D representations (Figures 8 and 9). The *BooksRepository* database is used for *BooksRepository* OLAP cube generation. The tables *Materials*, *Places* and *Repositories* are used for constructing dimensions. The table *MaterialsAvailability* is used for number measure generation. The OLAP cube is made using SQL Server Analysis Services (SSAS) and Microsoft Visual Studio 2022 [17]. The multidimensional language (MDX) query is executed to select the 3D data analogously to the 3D index matrices.

MDX query:

```
SELECT ([Places].[Hierarchy].[Country],
[Repositories].[Hierarchy].[Repository Owner]) ON ROWS,
[Materials].[Hierarchy].[Material Publisher] ON COLUMNS
FROM [BookRepository]
WHERE [Measures].[Number]
```

Result:

		Action	Ambari	Orion	Unknown
Bulgaria	Ani Hristova	4	8	(null)	(null)
Bulgaria	Chis Tom	5	42	29	(null)
Bulgaria	Tom Mark	(null)	(null)	(null)	(null)
Bulgaria	Unknown	(null)	(null)	(null)	(null)
England	Ani Hristova	4	8	22	(null)
England	Chis Tom	55	(null)	(null)	(null)
England	Tom Mark	(null)	(null)	(null)	(null)
England	Unknown	(null)	(null)	(null)	(null)
Poland	Ani Hristova	(null)	(null)	(null)	(null)
Poland	Chis Tom	(null)	(null)	(null)	(null)
Poland	Tom Mark	(null)	2	(null)	(null)
Poland	Unknown	55	(null)	(null)	(null)
Portugal	Ani Hristova	(null)	(null)	(null)	(null)
Portugal	Chis Tom	6	(null)	22	(null)
Portugal	Tom Mark	55	(null)	(null)	(null)
Portugal	Unknown	(null)	(null)	(null)	(null)
Turkey	Ani Hristova	(null)	(null)	7	2
Turkey	Chis Tom	66	60	59	(null)
Turkey	Tom Mark	(null)	(null)	(null)	(null)
Turkey	Unknown	(null)	(null)	(null)	(null)
Unknown	Ani Hristova	(null)	(null)	(null)	(null)
Unknown	Chis Tom	(null)	(null)	7	(null)
Unknown	Tom Mark	(null)	(null)	(null)	(null)
Unknown	Unknown	(null)	(null)	(null)	(null)

Figure 8. MDX query representing 3D data from the *BooksRepository* OLAP cube.

The same algorithm and instruments are used for *BookStoresChain* OLAP cube generation. It is constructed using *BookStoresChains* database tables. *Books*, *Regions*, and *Bookstores* tables are extracted for creating the dimensions. The table *BookStates* is used for Number measure selection. The MDX query is executed to select the 3D data similar to the 3D index matrices. The result is presented in Figure 9.

MDX query:

```
SELECT ([Regions].[Hierarchy].[C State],
[Bookstores].[Hierarchy].[Director]) ON ROWS,
[Books].[Hierarchy].[Publisher] ON COLUMNS
FROM [BookStoreChains]
WHERE [Measures].[Number];
```

Result:

		Action	Ambari	Orion	Unknown
Bulgaria	Chis Tom	5	42	22	(null)
Bulgaria	Henry Dim	4	8	(null)	(null)
Bulgaria	Unknown	(null)	(null)	(null)	(null)
England	Chis Tom	(null)	(null)	(null)	(null)
England	Henry Dim	4	8	22	(null)
England	Unknown	55	(null)	(null)	(null)
Turkey	Chis Tom	72	58	81	2
Turkey	Henry Dim	(null)	(null)	(null)	(null)
Turkey	Unknown	(null)	(null)	(null)	(null)
Unknown	Chis Tom	(null)	(null)	7	(null)
Unknown	Henry Dim	(null)	(null)	(null)	(null)
Unknown	Unknown	(null)	(null)	(null)	(null)

Figure 9. MDX query representing 3-D data from *BookStoresChains* OLAP cube.

It is well known that set operations are typically performed within a single OLAP data cube, where they are implemented by combining attribute groups across dimensions [18]. In the present case, however, two OLAP cubes are involved. Therefore, the corresponding 3D tables are extracted and used for further processing. The “symmetric difference” operation is implemented in a separate application. To this end, OLAP data are extracted via an SSAS connection using Excel OLAP tools and represented as 3D tables. The table shown in Figure 10 contains data from the *BookStoresChains* OLAP cube, while the table presented in Figure 11 includes data from the *BooksRepository* OLAP cube.

BookStoresChains OLAP cube					
Number	Column Labels				
Row Labels	± Action	± Ambari	± Orion	± Unknown	Grand Total
± Bulgaria	9	50	22		81
± Chis Tom	5	42	22		69
± Henry Dim	4	8			12
± England	59	8	22		89
± Henry Dim	4	8	22		34
± Unknown	55				55
± Turkey	72	58	81	2	213
± Chis Tom	72	58	81	2	213
± Unknown			7		7
± Chis Tom			7		7
Grand Total	140	116	132	2	390

Figure 10. Data from *BookStoresChains* OLAP cube.

BooksRepository OLAP cube					
Number	Column Labels				
Row Labels	⊕ Action	⊕ Ambari	⊕ Orion	⊕ Unknown	Grand Total
⊕ Bulgaria	9	50	29		88
⊕ Ani Hristova	4	8			12
⊕ Chis Tom	5	42	29		76
⊕ England	59	8	22		89
⊕ Ani Hristova	4	8	22		34
⊕ Chis Tom	55				55
⊕ Poland	55	2			57
⊕ Tom Mark		2			2
⊕ Unknown	55				55
⊕ Portugal	61		22		83
⊕ Chis Tom	6		22		28
⊕ Tom Mark	55				55
⊕ Turkey	66	60	66	2	194
⊕ Ani Hristova			7	2	9
⊕ Chis Tom	66	60	59		185
⊕ Unknown			7		7
⊕ Chis Tom			7		7
Grand Total	250	120	146	2	518

Figure 11. Data from *BooksRepository* OLAP cube.

Two tables from the *BooksRepository* OLAP cube and *BookStoresChains* OLAP cubes are cleaned and preprocessed to take the form presented in Figure 12. The first two columns contain the values for attributes of two dimensions. The first row presents the attributes of the third dimension. These two tables are used for visualization of the “symmetric difference” operation. Another possible option is for the user to generate artificial files of a similar type.

		Action	Ambari	Orion
Bulgaria	Ani Hristova	4	8	
Bulgaria	Chis Tom	5	42	29
England	Ani Hristova	4	8	22
England	Chis Tom	55		
Poland	Tom Mark		2	
Poland	Unknown	55		
Portugal	Chis Tom	6		22
Portugal	Tom Mark	55		
Turkey	Ani Hristova			7
Turkey	Chis Tom	66	60	59

		Action	Ambari	Orion (Ozone)
Bulgaria	Chis Tom	5	42	22
Bulgaria	Henry Dim	4	8	
England	Henry Dim	4	8	22
England	Unknown	55		
Turkey	Chis Tom	72	58	81

Figure 12. Transformed tables from *BooksRepository* OLAP cube (on the left) and from *BookStoresChains* OLAP cube (on the right).

4.1. An Illustration of the First “Symmetric Difference” Operation over Three-Dimensional Data

The “symmetric difference” operations are implemented in a Windows Forms (.NET Core 12) application using the programming language C#. The data from the two input tables are loaded into two programmatically generated DataGridView controls. These controls are designed to represent three dimensions. The first and second dimensions are visualized in the first two columns, while the third dimension is presented in the first row of the DataGridView controls.

The first operation “symmetric difference” identifies differing records across the dimensions by performing the suboperations *BooksRepository* minus *BookStoresChains* and *BookStoresChains* minus *BooksRepository*. The resulting data are then combined into a single dataset. The operation includes the following steps:

Case 1: The combination of:

- Titles of books stored in *Materials* table of *BooksRepository* database minus *Titles of books* stored in *Books* table of *BookStoresChain* database **AND**
- The names of bookshops stored in *Repositories* table of *BooksRepository* database minus the names of bookshops stored in *BookStores* table of *BookStoresChains* database **AND**
- The names of towns stored in *Places* table of *BooksRepository* database minus the names of towns stored in *Regions* table of *BookStoresChains* database.

Case 2: The combination of:

- Titles of books stored in *Books* table of *BookStoresChains* database minus titles of books stored in *Materials* table of *BooksRepository* database **AND**
- The names of bookshops stored in *BookStores* table of *BookStoresChains* database minus the names of bookshops in *Repositories* table of *BooksRepository* database **AND**
- The names of towns stored in *Regions* table of *BookStoresChains* database minus the names of towns stored in *Places* table of *BooksRepository* database.

The first “symmetric difference”-type operation is tested using the data tables presented in Figure 12. The result is visualized in Figure 13. The third DataGridView control shown on the bottom of Figure 13 presents the first “symmetric difference”-type operation as a result of the input data loaded in the first two datagridview controls.

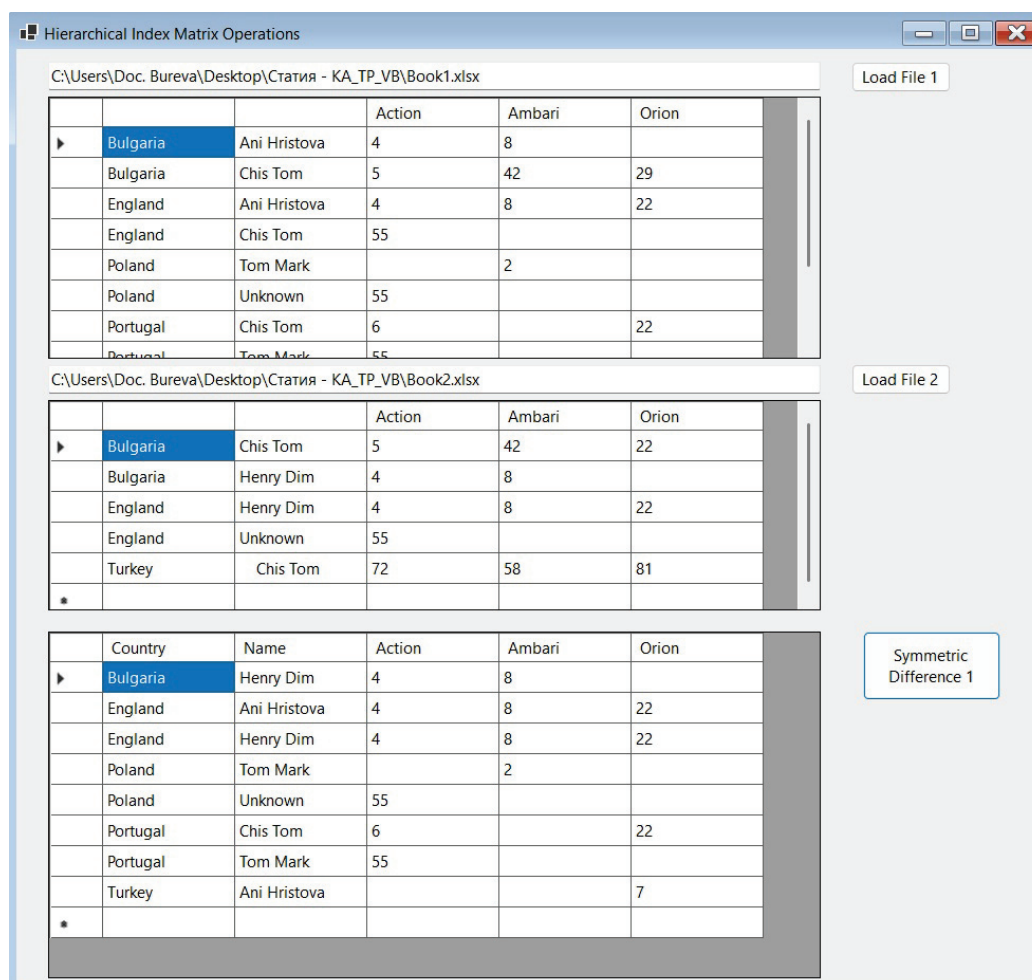


Figure 13. The first “symmetric difference”-type operation implementation.

4.2. An Illustration of the Second “Symmetric Difference”-Type Operation over 3D Data

The second “symmetric difference”-type operation performs extraction based on the one different dimension for *BooksRepository* minus *BookStoresChains* and for *BookStoresChains*

minus *BooksRepository* united in one resulting dataset. The operation includes the following steps:

Case 1: The combination of:

- Titles of books stored in *Materials* table of *BooksRepository* database minus titles of books stored in *Books* table of *BookStoresChain* database **OR**
- The names of bookshops stored in *Repositories* table of *BooksRepository* database minus the names of bookshops stored in *BookStores* table of *BookStoresChains* database **OR**
- The names of towns stored in *Places* table of *BooksRepository* database minus the names of towns stored in *Regions* table of *BookStoresChains* database.

Case 2: The combination of:

- Titles of books stored in *Books* table of *BookStoresChains* database minus titles of books stored in *Materials* table of *BooksRepository* database **OR**
- The names of bookshops stored in *BookStores* table of *BookStoresChains* database minus the names of bookshops in *Repositories* table of *BooksRepository* database **OR**
- The names of towns stored in *Regions* table of *BookStoresChains* database minus the names of towns stored in *Places* table of *BooksRepository* database.

The second “symmetric difference”-type operation is tested using the data tables presented in Figure 12. The result is demonstrated in Figure 14. The third DataGridView control shown on the bottom of Figure 14 presents the second “symmetric difference”-type operation of the data loaded in the first two DataGridView controls.

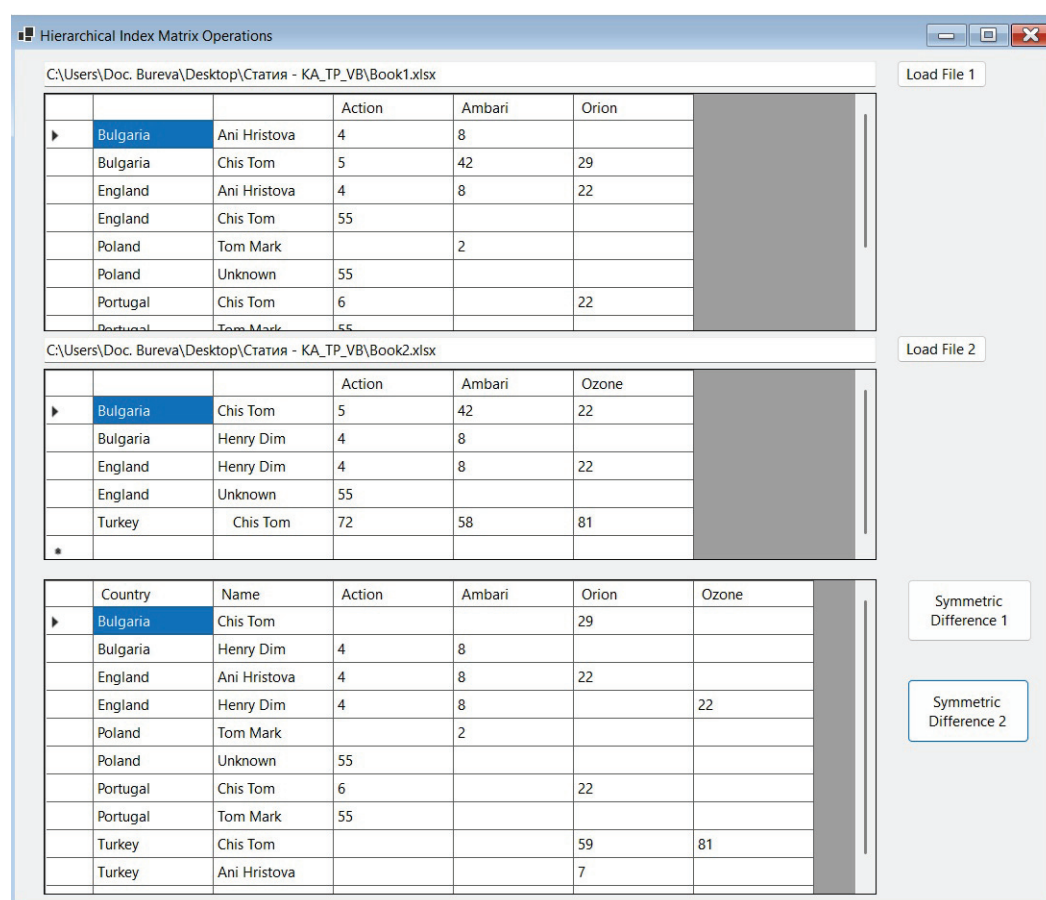


Figure 14. The second “symmetric difference” operation implementation.

The examples, presented in Sections 4.1 and 4.2, demonstrate the implementations of the herewith defined “symmetric difference” operations, which provide capabilities to find different data from two 3D datasets. The operations are helpful when the user

wants to extract and work with records that are present in only one of two datasets. Comparing data in different sources can be used to increase the efficiency of querying and supply management.

5. Conclusions

In the present paper, two new “symmetric difference”-type operations were introduced over 3-DEIMs, shown to be applicable also over IFIMs. The presented example of implementation of the elaborated “symmetric difference” operations in the field of relational databases served to demonstrate the entire process: from the creation of the relational databases, through their use for 3-D OLAP cube generation, to the extraction of 3D data tables using OLAP tools for Excel and their use in C# Windows Forms application containing the “symmetric difference” operation implementation. Thus, the two new “symmetric difference”-type operations were demonstrated to be effective for the comparison of data sets with dissimilar attributes. As a direction of further research, at the moment the authors are working on the representativeness of the arbitrary databases by IMs and the operations over IMs, as well as on the program realization of operations over IMs, which do not have analogues in the BigData. The results from this ongoing research (including newly here defined definitions) will be included in a further book, which will be a continuation of [2,3].

Author Contributions: Conceptualization, K.A., V.B. and T.P.; methodology, K.A.; software, V.B.; validation, K.A., V.B. and T.P.; formal analysis, K.A. and T.P.; investigation, K.A., V.B. and T.P.; writing—original draft preparation, K.A. and T.P.; writing—review and editing, K.A., V.B. and T.P.; visualization, V.B. and T.P.; supervision, K.A. and T.P.; project administration, K.A., V.B. and T.P.; funding acquisition, K.A., V.B. and T.P. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the funding by the Bulgarian National Science Fund under the grant KP-06-N72/8 from 14 December 2023, titled “Intuitionistic Fuzzy Methods for Data Analysis with an Emphasis on the Blood Donation System in Bulgaria”.

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

Acknowledgments: The authors are grateful to Vassia Atanassova and Peter Vassilev for their fruitful discussions and support in the visualization materials.

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

References

1. Atanassov, K. Generalized index matrices. *Comptes Rendus L'Academie Bulg. Sci.* **1987** *40*, 15–18.
2. Atanassov, K. *Index Matrices: Towards an Augmented Matrix Calculus*; Springer: Cham, Switzerland, 2014.
3. Atanassov, K. *On Index Matrices*; Academic Publishing House “Acad. Marin Drinov”: Sofia, Bulgaria, 2025.
4. Bernstein, D.S. *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed.; Princeton University Press: Princeton, NJ, USA, 2009.
5. Lang, S. *Algebra*; Springer: New York, NY, USA, 2002.
6. Bureva, V.; Atanassov, K.; Atanassova, V.; Umlenski, I. Inverse Operator over Index Matrices. *Mathematics* **2026**, *14*, 615. [CrossRef]
7. Atanassov, K.; Bureva, V. Two new operations over extended index matrices and their applications in Big Data. In Proceedings of the 17th Conference on Computer Science and Intelligence Systems, Sofia, Bulgaria, 4–7 September 2022; pp. 1–6.
8. Fraenkel, A.; Bar-Hillel, Y. *Foundations of Set Theory*; North-Holland Publ. Co.: Amsterdam, The Netherlands, 1958.
9. Kuratowski, K.; Mostowski, A. *Set Theory*; North-Holland Publ. Co.: Amsterdam, The Netherlands, 1967.
10. Rasiowa, H.; Sikorski, R. *The Mathematics of Metamathematics*; Polish Academy of Sciences: Warszawa, Poland, 1963.
11. MacLane, S.; Birkhoff, G. *Algebra*, 3rd ed.; AMS Chelsea Publishing: Providence, RI, USA, 1999.
12. Date, C.J. *Database Design and Relational Theory*; Apress: New York, NY, USA, 2019.
13. Sumathi, S.; Esakkirajan, S. *Fundamentals of Relational Database Management Systems*; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2007.
14. Garcia-Molina, H.; Ullman, J.; Widom, J. *Database Systems: The Complete Book*, 2nd ed.; Pearson: London, UK, 2009.

15. Beaulieu, A. *Learning SQL*, 3rd ed.; O'Reilly Media: Sebastopol, CA, USA, 2020.
16. Kaufmann, M.; Meier, A. *SQL and NoSQL Databases, Modeling, Languages, Security and Architectures for Big Data Management*; Springer: Berlin/Heidelberg, Germany, 2023.
17. Steven, H. *Hands-On SQL Server 2019 Analysis Services: Design and Query Tabular and Multi-Dimensional Models Using Microsoft's SQL Server Analysis Services*; Packt: Birmingham, UK, 2020.
18. Traneva, V.; Bureva, V.; Sotirova, E.; Atanassov, K. Index matrices and OLAP-cube. Part 3: A presentation of the OLAP "InterCube Set" and "Data cube" operations by index matrices. *Adv. Stud. Contemp. Math.* **2018**, *28*, 423–448.

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

UAV-Based QR Code Scanning and Inventory Synchronization System with Safe Trajectory Planning

Eknath Pore ¹, Bhumeswar K. Patle ^{2,*} and Sandeep Thorat ¹

¹ Department of Mechanical Engineering, School of Engineering and Sciences, MIT Art, Design and Technology University, Pune 412201, Maharashtra, India

² Department of Mechanical Engineering, School of Engineering Sciences, Ramdeobaba University, Nagpur 440013, Maharashtra, India

* Correspondence: balu_patle@rediffmail.com; Tel.: +91-97-5538-3710

Abstract

Modern-day urban warehouses face exploding large inventory and tight spaces requiring fast, accurate, and safe stocktaking in a narrow aisle in a GPS-denied environment. This paper proposes a complete UAV-enabled framework performing real-time QR code scanning with inventory synchronization through a safety-aware trajectory generation for obtaining collision-free motion. A novel hybrid workflow integrating MATLAB/Simulink R2024b and Unreal Engine is used for dynamics and photorealistic rendering, alongside a real-time warehouse setup using drone cameras and 3D LiDAR coupled with a ground control station and live dashboard. The system in this paper was evaluated by testing with single and multi-UAV models across high-fidelity simulations and experiments. Results demonstrate simulated QR accuracy of approximately 95 to 96%, with experimental validation achieving between 86 and 90.5% due to real-world environmental factors. In experimental and simulation analysis, mean end-to-end latency remained under half a second, trajectory error range between 8 and 10 cm, and safety margins were consistently maintained throughout the test. It was further observed that multi-UAV coordination halved mission time compared to single-drone tests while keeping duplicate reads negligible, indicating a scalable and safe pipeline for industry application.

Keywords: UAV inventory inspection; QR code scanning; safety-aware trajectory planning; obstacle avoidance; Simulink–Unreal co-simulation; GPS Denied Environment

1. Introduction

The evolution of intelligent warehousing has given rise to the disruptive trend in the sphere of logistics and inventory management, where the automation, real time data processing, and intelligent monitoring systems prove to be inevitable. One of such advances has been the Unmanned Aerial Vehicles (UAVs) used in indoor warehouses. Their size, flexibility, and agility also allow the UAVs to be particularly useful in navigating aisles and multi-tiered shelves, which allows them to perform their scanning and inspection tasks efficiently and with minimal manual effort. The high pace of e-commerce, smart manufacturing, and automated supply chains development has drastically caused an upsurge in the need to ensure the precise real time monitoring of inventory in indoor warehouse settings [1]. Traditional approaches to inventory check mostly depend on manual inspection processes like barcode scanning, which is time-consuming and labour-intensive; the handheld readers or ground-based robots that are prone to human error;

most of the facilities are high-density and large-scale storage [2]. As warehouses continue to expand in size, both in height and width, timely inventory visibility, without jeopardizing operational safety, has become a significant issue in contemporary industrial logistics [3]. Due to the mobility, flexibility, and the capacity of visiting inaccessible places of storage without any structural changes, UAVs have recently gained a lot of interest as a new potential solution towards autonomous inside inspections and inventory management [4,5]. Applications of UAVs exhibit superior vertical reach and less interference in floor space compared to AGVs and AMRs. Therefore, this will make the application of UAVs more favorable in dense layouts of warehouses in general [6]. However, indoor deployment of UAVs introduces some unique technical challenges related to the absence of GPS signals, highly constrained navigation spaces, dynamic obstacles, limited payload capacity, and stringent safety requirements of UAVs [7].

Among different identification technologies to trace inventory, QR code has been considered as a reliable and cost-effective substitute for RFID in indoor environments [8]. QR codes provide higher data capacity, are robust to partial occlusion, and are easily deployable without special infrastructure [9]. Vision-based QR code reading using UAV-mounted cameras can achieve non-contact, flexible, and scalable inventory inspection, especially in warehouses with heterogeneous item sizes and storage layouts [10,11]. However, accurate QR code detection from flying platforms still needs precise positioning, stable flight control, optimal view angles, and robust image processing under different lighting conditions. Therefore, for flying autonomously in confined indoor places such as warehouses, safe and efficient trajectory planning is a fundamental requirement. Indoor navigation requires real-time obstacle detection, tight manoeuvrability, and collision avoidance around racks, shelves, workers, and other mobile agents [12]. Conventional methods, where waypoints are used or fixed path planning, tend to be unsuited to dynamically changing conditions and unknown obstacles, adding increased chances of collisions [13]. In the recent past, safety-minded trajectory planning has been taken into more serious consideration, combining sensor-based perception with risk assessment models towards this end [14].

LiDAR sensors have proven highly efficient in accurate obstacle detection and spatial mapping within an indoor environment because they are very accurate and light-insensitive [15]. The UAV can increase its situational awareness and achieve credible vision in both navigation and visual inspection tasks by implementing a wide-angle fish-eye camera [16]. It is possible to test and validate UAV control, sensing, and perception algorithms in a realistic manner by using simulation-driven development tools, including Simulink-based 3D modelling and Unreal Engine development environments before real deployment [17,18]. A high-fidelity simulated environment has greatly lowered the risk of development and also increased system reliability. Adding to autonomous navigation, real-time data synchronisation between UAVs and GCS and enterprise inventory management systems is the key to viable industrial acceptance in a real-life situation [19].

The communication framework of cloud-enabled communication and lightweight web servers, such as those based on the Flask architecture, conveys seamless transmission of data from scanned inventory to centralized dashboards for real-time monitoring and decision making [20]. Ensuring low-latency communication, handling redundancy, and computational efficiency remains one of the critical research focuses during large-scale deployment [21].

Although several works have explored the inventory inspection with UAVs, existing studies are mostly limited to validations via only simulation [22,23] or restricted to static environments without integrated safety-aware trajectory planning and real-time system synchronization [24–26]. Furthermore, comprehensive experimental validation that combines realistic simulation environments and real-world UAV testing remains relatively rare

in the literature thus far. Systems must guarantee reliable obstacle avoidance; sustain high QR decoding rates under variable illumination, motion blur, and partial occlusions; and regulate trajectories safely within narrow, multi-level aisles while keeping the inventory database synchronized in near real time. These demands are compounded by resource limits, compute, memory, and battery endurance, which constrain continuous, fully on-drone processing and prolonged scan missions without strategic offloading or careful energy management. Much of the existing literature either surveys UAV use from a broad, cross-sector perspective (transportation, disaster response, agriculture) or focuses on isolated warehouse functions such as cycle counting or spot inspection. As a result, insights are often too general to respect warehouse constraints or too narrow to capture the coupling among perception reliability, safety-aware motion, and enterprise data integration.

Motivated by these gaps, this paper presents a UAV-based QR code scanning and inventory synchronization system, along with safety-aware trajectory planning, optimized for indoor warehouse environments. This study proposes a unified UAV-based indoor inventory inspection system that distributes computation across the aerial platform, a ground control station (GCS), and a lightweight cloud service. The UAV follows a pre-planned aisle trajectory while capturing live video; the GCS executes real-time QR decoding and pose estimation, offloading intensive image processing from the air vehicle. Simultaneously, a 3D LiDAR sensor produces point clouds for scene reconstruction; these data inform and validate obstacle-aware flight paths within a MATLAB/Simulink environment that is co-tested against a photorealistic Unreal Engine warehouse. Safety is enforced by a trajectory generator embedded with a mathematical risk-factor model that adaptively adjusts waypoints and velocities to maintain clearance, and triggers a return-to-base protocol under critical battery or perception conditions.

Decoded item IDs, timestamps, confidence scores, and UAV telemetry are sent to a Flask-based host server in order to complete the feedback loop and visualize on a web dashboard that allows real-time monitoring and heatmap-like coverage maps, as well as real-time inventory syncing. This design enhances scalability (by enabling multi-UAV operation) and robustness (by isolating perception and control from network variance), while preserving low end-to-end latency for operational decision-making. The proposed system is validated through a combination of Simulink simulations, Unreal-based virtual environments, and controlled flight trials with commercial quadrotors and QR-labeled racks. Performance is quantified in terms of scanning accuracy, computational latency, trajectory safety, and detection reliability, demonstrating that a tightly coupled perception-planning-synchronization stack can deliver repeatable, industrial-grade UAV-based inventory management for indoor logistics.

The rest of this work is organized as follows: in Section 2, we discuss materials and methods consisting system overview, trajectory planning, and QR detection. Section 3 describes the simulation and experimental analysis during the inventory management using UAV, followed by results and discussions in Section 4. At last, Section 5 presents a conclusion to this work and outlines the directions for future research.

2. Materials and Methods

2.1. System Overview

This study develops an end-to-end UAV system for autonomous indoor inventory inspection that combines real-time QR detection, safety-aware trajectory control, and cloud-synchronized logging. The architecture integrates (i) a quadrotor equipped with a forward-facing camera and 3D LiDAR, (ii) simulation and visualization via MATLAB/Simulink and Unreal Engine, and (iii) a Python/Flask 3.1.2 backend exposing REST and WebSocket

endpoints for ingestion and live visualization. Figure 1 summarizes the data flow from on-board sensing through the Ground Control Station (GCS) to the host server and dashboard.

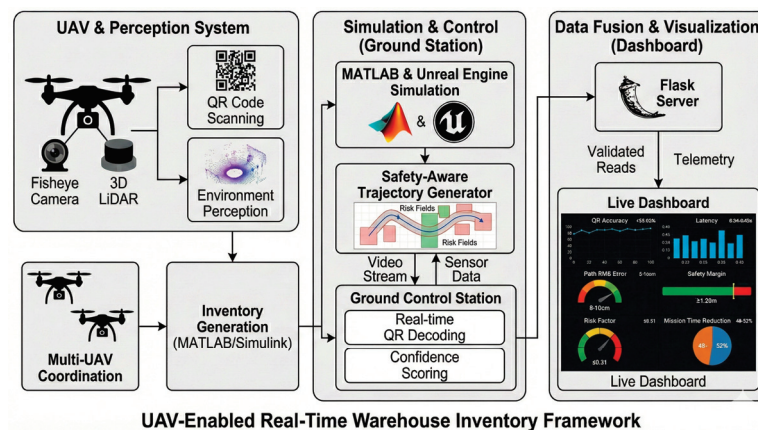


Figure 1. Proposed Method System Overview with Data Fusion and Visualization.

Within a photorealistic warehouse model, racks and shelves are labeled with standardized QR tags (e.g., A1–A10, C1–C5), each encoding a unique inventory string that maps one-to-one to the database. The UAV follows a predefined zig-zag coverage path with hover-and-scan stops at waypoints (x, y, z) in the warehouse frame. Video is streamed to the GCS over RTSP/UDP; to conserve on-board compute, QR decoding and confidence scoring run on the GCS using OpenCV. For each successful decode, the GCS packages the QR string, synchronized timestamp, UAV pose, and a scalar confidence score into a JSON payload and posts it to the Flask API, which validates the schema, updates the inventory state (scanned/duplicate/missing), and pushes live updates to a SocketIO dashboard.

Safety is enforced by a risk factor R_f derived from the UAV's distance to nearby obstacles (from LiDAR or map) and used to gate motion and trigger short-horizon replanning when R_f exceeds a threshold (details in Trajectory Planning). A return-to-base routine is invoked under low battery, persistent perception loss, or mission completion. The full pipeline is validated in co-simulation (Simulink + Unreal) and in a controlled warehouse with QR-labeled racks, using identical logging and time synchronization to enable apples-to-apples comparisons. Simulation and experimental results are analyzed in terms of QR scanning accuracy, end-to-end latency, trajectory fidelity, safety coverage, redundancy handling, and coverage efficiency, demonstrating readiness for warehouse-scale deployment. Figure 2 shows the complete workflow of the method.

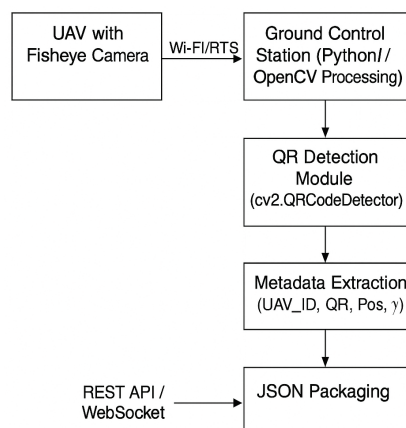


Figure 2. Workflow of UAV video streaming and QR code detection at the Ground Control Station (GCS).

2.2. Warehouse Environment Modeling and QR Code Arrangement

For a warehouse environment, the simulation and experimental testing phase has to ensure a high level of realism and repeatability. Hence, the environment is carefully modeled using a hybrid approach by integrating Simulink's 3D Scene Builder and Unreal Engine. This bi-environment setup enables simultaneous development and validation of both control logic and visual sensing modules under controlled yet diverse spatial conditions. The warehouse layout incorporates the standard multi-tier shelving racks, which have evenly spaced aisles, and obstacles that mimic a real-world warehouse environment, including support beams and overhanging inventory.

Each rack is systematically labeled using an alphanumeric QR code scheme, which represents the spatial organization of the warehouse. The labels follow a uniform format, showcasing rack and shelf indices, such as "A1" representing Rack A/Shelf 1, while "C5" represents Rack C/Shelf 5. These labels are marked in QR codes and attached to the front of the boxes of cardboard located on each shelf. The labeling pattern is used to achieve consistency and allow easy UAV recognition at different angles and log data under simulation and real deployment.

The QR codes are uniquely designed to have a uniquely structured string that identifies the item ID and its exact position in the warehouse. The usual format of a QR code that will be used in this research is as follows:

$$QR_{string} = "Item\#A14 - R5" \quad (1)$$

From the above example, "Item#A14-R5" denotes a product stored at shelf 14 of Rack A, row 5. This string format is essential for establishing one-on-one correspondence between the physical inventory and the digital record of it in the cloud-based database. During the flight mission, the UAV scans this QR code and relays the decoded string to the ground control system (GCS), which subsequently matches the data to entries in the central inventory management software.

The spatial accuracy of the placement of QR codes in the simulation environment is achieved through the fact that the positioning of the QR codes is directly aligned with the Simulink coordinates and the Unreal Engine asset meshes. The placement angle, the light source, and perspective distortion are also adjusted to realistic conditions of scanning, as the UAV camera experiences. This simulation modelling means that the detection algorithm is resistant to visual noise and different flight altitudes. To conclude, the warehouse modeling and QR code layout make the basis of organized inventory mapping. This consistency in the simulation and physical worlds is what will allow the smooth transition between design and testing, as well as being of critical importance in allowing the UAV to localize and identify inventory items in real-time in an autonomous manner.

2.3. UAV Trajectory Planning and Safety-Aware Replanning

This section provides a complete theoretical and algorithmic foundation for an aisle-coverage planner tailored to indoor warehouses. First, nominal, camera-steady trajectory generation constructs a zig-zag coverage path through shelves by interpolating a sequence of hover-and-scan waypoints with a smooth, spline-like curve. Smoothness is enforced by penalizing translational acceleration and yaw-rate "energy," which yields cubic-in-time position segments in unconstrained arcs and limits camera slew to keep QR targets within the field of view. Second, feasibility is imposed through hard bounds on speed, acceleration, and yaw rate, together with linear "keep-in" constraints that restrict motion to the aisle polytope; these constraints ensure the planned motion is executable on a real quadrotor with finite actuation and that the vehicle remains clear of shelf faces and structural boundaries. Third, to make the problem numerically robust and real-time capable, we introduce a

convex discrete surrogate: the continuous objective and constraints are discretized on a fixed time grid using first and second difference operators, producing a banded quadratic program (QP) with a positive-definite Hessian. The structure admits fast solvers (e.g., banded Cholesky or projected conjugate gradients), guarantees a unique optimizer under standard boundary pinning, and supports optional terms (e.g., line-of-sight yaw alignment) without sacrificing convexity. Fourth, recognizing that indoor scenes are dynamic and partially modeled, we embed short-horizon replanning driven by a smooth, LiDAR-based risk field. A signed-distance-derived, sigmoid “Risk Factor” provides continuous risk values and gradients everywhere in free space; whenever predicted risk exceeds a safety threshold, a small horizon around the offending segment is re-optimized by successive convexification. Linearizing the risk term, regularizing increment curvature, and applying a backtracking line search produce monotone risk reduction while keeping updates within dynamic limits and close to the nominal path. Together, these components yield a planner that is provably well-posed, numerically stable, and fast enough for closed-loop use, with clearly defined symbols, objectives, and constraints introduced at first mention for reproducibility.

1. Waypoints and Continuous Kinematics

Let the scan waypoints be

$$P_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, \dots, N \quad (2)$$

The continuous reference states are position $p(t) \in \mathbb{R}^3$ and yaw $\psi(t) \in \mathbb{R}, t \in [0, T]$ with commanded kinematics,

$$\dot{p}(t) = v_c(t), \dot{\psi}(t) = \omega_c(t). \quad (3)$$

Feasibility is enforced by bounds

$$\begin{aligned} \|v_c(t)\| &\leq v_{\max}, \\ \|v_c(t)\| &\leq a_{\max}, \\ |\omega_c(t)| &\leq \omega_{\max}, \forall t, \end{aligned} \quad (4)$$

and an aisle polytope $\Omega \subset \mathbb{R}^3$ (linear “keep-in” constraints) such that $p(t) \in \Omega$.

2. Camera-Steady Nominal Path (Continuous Optimal Control)

To obtain smooth motion that keeps the camera steady on shelves, we minimize acceleration and yaw-rate “energy”;

$$\begin{aligned} \min_{p(\cdot), \psi(\cdot)} J_{\text{smooth}} &= \int_0^T (\lambda_s \|\ddot{p}(t)\|_2^2 + \lambda_\psi \dot{\psi}(t)^2) dt \\ \text{s.t. } \|\dot{p}(t)\|_2 &\leq v_{\max}, \|\ddot{p}(t)\|_2 \leq a_{\max}, |\dot{\psi}(t)| \leq \omega_{\max}, \\ p(0) &= P_1, p(T) = P_N, p(t) \in \Omega, \forall t \in [0, T]. \end{aligned} \quad (5)$$

where, $p : [0, T] \rightarrow \mathbb{R}^3$ is position, are weights.

In unconstrained arcs, the Euler–Lagrange equations give $\ddot{p}(t) = 0$, hence each Cartesian component of $p(t)$ is cubic in time, explaining the spline-like appearance. The yaw term acts as a Tikhonov regularizer on ψ , limiting camera slew.

Optional LOS alignment. If desired, add

$$J_{\text{LOS}} = \kappa \int_0^T (\psi(t) - \psi_{\text{LOS}}(t))^2 dt, \psi_{\text{LOS}}(t) = \text{atan}^2(n_y(t), n_x(t)), \quad (6)$$

with (n_x, n_y) a shelf-normal projection; in discrete least-squares form (below), this remains convex

3. Discrete Convex Surrogate (QP) for the Nominal Plan

Sample at $t_k = k\Delta t, k = 0, \dots, K, K\Delta t = T$, Define first/second differences

$$D_1 p := \frac{p_{k+1} - p_k}{\Delta t}, D_2 p := \frac{p_{k+1} - 2p_k + p_{k-1}}{\Delta t^2} \quad (7)$$

Let τ_i index the hover frame for waypoint p_i and $\rho_i > 0$ weight stop-and-scan tightness. The nominal problem is the convex QP

$$\begin{aligned} \min_{\{p_k, \psi_k\}} \quad & \sum_{i=1}^N \rho_i \|p_{\tau_i} - P_i\|^2 + \lambda_s \|D_2 p\|_2^2 + \lambda_\psi \|D_1 \psi\|_2^2 \left(+ \kappa \|\psi - \psi_{\text{LOS}2}\|_2^2 \text{ if used} \right) \\ \text{s.t.} \quad & \|D_1 p\|_\infty \leq v_{\max}, \quad \|D_2 p\|_\infty \leq a_{\max}, \quad \|D_1 \psi\|_\infty \leq \omega_{\max}, \\ & p_0 = P_1, \quad p_K = P_N, \quad p_k \in \Omega(\forall k) \end{aligned} \quad (8)$$

The Hessian is banded and positive semidefinite:

Pinning p_0, p_k and $\lambda_s, \lambda_\psi > 0$ yields strict convexity on the feasible affine subspace; thus, the optimizer is unique. With linear aisle faces and per-axis ℓ_∞ bounds, active-set or projected-CG solvers exploit the banded structure for near-linear time in K .

4. LiDAR-Driven Risk Field (Signed Distance) and Its Differentials

Let $r(p)$ be the Euclidean clearance (signed distance) to the nearest obstacle surface (positive in free Space) from a voxelized or meshed SDF/TSDF. Define a smooth Risk Factor

$$R_f(p) = \sigma(\beta[r_{\text{safe}} - r(p)]) = \frac{1}{1 + \exp(-\beta[r_{\text{safe}} - r(p)])}, \beta > 0, \quad (9)$$

where $r_{\text{safe}} > 0$ is the desired clearance (m) and $\beta > 0$ controls steepness.

Setting with safe threshold $\tau \in (0, 1)$ and safe set $S = \{p : R_f(p) \leq \tau\}$.

On regular SDF regions (away from edges/corners), ∇d is Lipschitz and $\nabla^2 d$ bounded, so D_f is smooth with Lipschitz gradient. The threshold equates to a clearance margin:

$$R_f(p) \leq \tau \Leftrightarrow r(p) \geq r_{\text{safe}} - \frac{1}{\beta} \ln\left(\frac{1-\tau}{\tau}\right). \quad (10)$$

Hence tuning $(r_{\text{safe}}, \beta, \tau)$ gives a geometric buffer in meters.

In our experiments, we used $r_{\text{safe}} = 0.5$ m, $\beta \in [8, 12] \text{ m}^{-1}$, and $\tau = 0.4$. The safety-aware trajectory generator in this study relies on a LiDAR-derived signed-distance (risk) field constructed from the available 3D warehouse map and is therefore validated primarily under quasi-static obstacle assumptions (e.g., racks, shelves, and other fixed infrastructure). While the optimization framework can replan over a short horizon when the risk field is refreshed, the present implementation does not claim full real-time avoidance of fast-moving or non-cooperative obstacles (e.g., workers, forklifts, or other mobile agents) because such capability would require (i) high-rate perception updates, (ii) dynamic obstacle tracking and prediction, and (iii) continuous reconstruction or incremental updates of the signed-distance field with bounded latency. Consequently, the reported results should be interpreted as performance in environments where the dominant obstacles are static or slowly varying relative to the replanning frequency; extending the method to dense, highly dynamic scenes is left as future work, involving online occupancy/SDF updates and reactive safety layers that can guarantee collision avoidance under rapid motion.

5. Short-Horizon, Risk-Aware Replanning (Successive Convexification)

When a predicted segment violates safety, we *locally* correct frames $j = k, \dots, k + h$ by minimally deflecting the nominal path while penalizing risk. Linearize the risk about p_j ,

$$R_f(p_j + \Delta p_j) \approx R_f(p_j) + \nabla R_f(p_j)^\top \Delta p_j. \quad (11)$$

stabilize increments with a second-difference (jerk-like) operator

$$E\Delta p = [(\Delta p_{j+1} - \Delta p_j) - (\Delta p_j - \Delta p_{j-1})]_{j=k+1}^{k+h} \quad (12)$$

Solve the convex QP

$$\begin{aligned} \min_{\Delta p} \quad & \|\Delta p\|_2^2 + \lambda \sum_{j=k}^{k+h} (R_f(p_j) + \nabla R_f(p_j)^\top \Delta p_j) + \eta \|E\Delta p\|_2^2 \\ \text{s.t.} \quad & p_j + \Delta p_j \in \Omega, \frac{\|\Delta p_j\|}{\Delta t} \leq v_{\max}, \frac{\|\Delta p_{j+1} - \Delta p_j\|}{\Delta t^2} \leq a_{\max}, \end{aligned} \quad (13)$$

With $\lambda, \eta > 0$. Let $H_{\text{hor}} = I + \eta E^\top E \succ 0$, and $g_{\text{hor}} = \lambda [\nabla R_f(p_k)^\top, \dots, \nabla R_f(p_{k+h})^\top]^\top$. The unique solution Δp is applied with a backtracking line search

$$\begin{aligned} p_j^+ &= p_j + \alpha \Delta p_j, \\ \alpha \in (0, 1] \text{ chose to ensure } & \sum_{j=k}^{k+h} R_f(p_j^+) \leq \sum_{j=k}^{k+h} R_f(p_j) - c\alpha \|g_{\text{hor}}\|^2, \end{aligned} \quad (14)$$

Under Lipschitz ∇R_f and bounded steps, the Armijo rule admits such an α ; repeated cycles (re-linearize, re-solve) yield a sequence whose limit points satisfy first-order stationarity of the underlying nonconvex horizon problem (standard successive convexification result).

6. Supervisory Logic (RTB Barrier)

Define battery fraction $b(t) \in [0, 1]$ and perception confidence $\gamma(t) \in [0, 1]$ (from the detection pipeline). A high-level guard switches to homing when

$$b(t) \leq b_{\min} \vee \gamma(t) < \gamma_{\min} \vee \text{mission complete}. \quad (15)$$

Hysteresis on b avoids chatter (e.g., require $b > b_{\text{reset}} > b_{\min}$ to re-enter scan mode)

2.4. Live Video Feed and Ground QR Code Detection

A critical component of the proposed UAV inventory system is the real-time detection of QR codes from shelves during indoor scanning missions. To balance accuracy, latency, and computational feasibility, the framework adopts a division of responsibilities between the UAV and the Ground Control Station (GCS). The UAV serves primarily as a video acquisition platform, while the GCS executes detection and decoding algorithms to minimize onboard computational overhead.

1. Video stream from UAV to GCS

Each UAV carries a forward-facing fisheye camera (intrinsic K , distortion model Π_{fish}) capturing RGB frames at frame rate F_c (typically 30 fps) and resolution $R_c = H \times W$. Let

$$V(t) = \{F_k\}_{k=0}^{\lfloor F_c t \rfloor}, F_k \in [0, 255]^{H \times W \times 3}. \quad (16)$$

Frames are packetized and transported to the GCS using RTSP/UDP over Wi-Fi 6 (or long-range telemetry). The end-to-end latency is the additive budget

$$\ell_{e2e} \approx \ell_{\text{cap}} + \ell_{\text{enc}} + \ell_{\text{net}} + \ell_{\text{buf}}, \quad (17)$$

which in our deployment remains $\ell_{e2e} \lesssim 200$ ms, meeting near-real-time constraints for indoor flight.

2.5. Mirror-Symmetric Multi-UAV Scheduling

To explicitly integrate symmetry into the multi-UAV coordination problem (Case II), a mirror-symmetric coupling is embedded directly into the trajectory optimization objective. Let the warehouse aisle be expressed in an aisle-aligned coordinate frame, where the aisle centerline defines the symmetry axis and the lateral coordinate is reflected across this axis. The mirror operator is defined as $M = \text{diag}(1, -1, 1)$, such that for any state $x = [x, y, z]^T$, the reflected state is $Mx = [x, -y, z]^T$. Denoting the discrete-time trajectories of the two UAVs by $\{X_A[k]\}_{k=1}^K$ and $\{X_B[k]\}_{k=1}^K$, symmetry is enforced by augmenting the banded quadratic program (QP) objective with a symmetry regularizer,

$$J_{\text{sym}} = w_{\text{sym}} \sum_{k=1}^K \frac{\|x_B[k] - Mx_A[k]\|^2}{2} \quad (18)$$

where w_{sym} controls the strength of symmetric coupling. The overall multi-UAV optimization then minimizes the nominal trajectory costs for each agent together with J_{sym} , subject to the same feasibility constraints used in the single-UAV formulation, including keep-in-aisle bounds and dynamic limits. Endpoint pinning is preserved by constraining the initial and terminal states to the mission start/goal conditions, ensuring a unique and well-posed QP solution while still allowing the optimizer to reconcile symmetry with safety and feasibility. As a result, mirror symmetry becomes a structural property of the optimization, rather than a post hoc trajectory characteristic, and the coordinated solution remains symmetric insofar as permitted by the active constraints.

2. QR code detection at the GCS

QR decoding is performed offboard to conserve the UAV battery and CPU. Let **D** denote the OpenCV detector/decoder (cv2.QRCodeDetector) composed with standard preprocessing (grayscale + adaptive thresholding) and Reed–Solomon error correction.

- Preprocessing: grayscale conversion and adaptive thresholding to mitigate indoor lighting variations.
- Pattern detection: extraction of finder patterns (position markers in QR corners).
- Decoding: Reed–Solomon error correction to recover embedded data even under partial occlusion or blur.

We model decoding as

$$(q_k, \gamma_k) = D(F_k), q_k \in S \cup \{\emptyset\}, 0 \leq \gamma_k \leq 1, \quad (19)$$

where S is the set of valid QR strings and γ_k is a confidence score derived from detection probability and patch sharpness. If no valid code is present, $q_k = \emptyset$. Finder-pattern geometry (three L-corner squares with canonical ratios) and perspective consistency are enforced to suppress false positives in wide-FOV images. This GCS-based detection ensures higher processing throughput while enabling multi-UAV scalability, since one ground station can concurrently process multiple streams.

3. Data Packaging and Metadata Extraction

For every successful detection, the GCS extracts QR content and appends contextual metadata. Each decoded event is logged with the following attributes:

- Decoded String (q): Encoded item identifier, e.g., Item#C5-R2.
- UAV Identifier (u): Unique ID for the scanning UAV (e.g., DRONE01).

- 3D Position $x_k = (x_k, y_k, z_k) \in R^3$ UAV's estimated position at detection time, derived from Simulink trajectory data or onboard odometry, and ψ_k .
- Timestamp (t_k): Coordinated Universal Time (UTC) at detection.
- Confidence Score (γ): Confidence metric ($0 \leq \gamma \leq 10$), based on OpenCV decoding probability or secondary heuristic (e.g., sharpness of detected QR region).

The event,

$$e_k = (uav_id, q_k, x_k, \psi_k, t_k, \gamma_k) \quad (20)$$

This information is serialized into a JSON object (Figure 3) for transmission to the cloud dashboard and inventory database:

```
json
{
  "uav_id": "DRONE01",
  "qr_string": "Item#C5-R2",
  "timestamp": "2025-07-28T18:21:12",
  "position": [5.4, 3.2, 1.1],
  "confidence": 0.94
}
```

Figure 3. JSON Object File.

4. Data Stream Integration

These JSON packets are transmitted from the GCS to the Flask-based server via RESTful API calls or WebSocket streams. QR detection is recorded in the inventory database in real-time, with visualisation data being visualised live using the ground dashboard. In addition, detections with (where approx) are reported to be rescheduled to scan again to be resistant to false positives or low-light situations.

2.6. Host Server and API Communication

The interface between the cloud server and the GCS is achieved by a. Flask based server to provide low-latency and high-reliability synchronization of UAV scan data. The host server is an intermediate server that takes in a structured JSON file as input from the GCS. processing the information, and the centralized inventory database.

(1) Flask Backend Integration

Python Flask micro-framework was chosen because it has a lightweight architecture and can be scaled to accommodate real-time streams of data. A special API endpoint, `/api/upload_qrdata`, was developed to accept POST requests by the GCS. The request contains UAV-specific metadata that is in the form of a JSON format, which includes the UAV ID, the decoded QR string, the time of arrival, the spatial coordinates, and the confidence score. Upon receipt Flask application will authenticate the payload with schema checking, and it will ensure data fidelity and eliminate invalid records.

The deployment stack depends on the Flask-RESTful API management tool and Gunicorn and Nginx as the web server; therefore, becoming robust with the many connections of UAVs. The test was performed on 20 simulated swarms of UAVs transmitting simultaneously, with the average response time of the API maintained at a constant level of less than 100 ms, meaning the system would scale to industrial warehouse operations.

(2) Inventory Database Synchronization

After validating each JSON payload, the host server updates the inventory-management database and emits an event to the dashboard. Lightweight prototyping was done using SQLite, and PostgreSQL. or NoSQL databases (MongoDB) were set up to support more massive deployment. Each QR string is scanned and converted to an equivalent digital record, with the status of the item being renewed to one of the following states:

- Scanned (Item successfully detected and timestamp logged)
- Duplicate (Item re-scanned within a defined time threshold; flagged for operator review)
- Missing (Item not found in the expected shelf range in the scanning cycle)

The database facilitates historical logging, whereby a new session record is generated every scan cycle of the UAV. The model of design will facilitate future audits, the trajectories with scans, and the reporting of compliance with industries requiring the traceability of the movements and confirmation of their stocks.

2.7. Real-Time Dashboard and Cloud Visualization

To give operators better situational awareness, we designed a real-time dashboard that displays UAV flight paths, scan results, and system alerts. By combining technical processes with an intuitive interface, the dashboard improves monitoring and decision-making.

(1) Flask-SocketIO Dashboard

The dashboard is built with Flask-SocketIO to enable two-way communication between the server and the client. Instead of periodic polling, SocketIO delivers event-driven updates whenever a UAV scan or trajectory change occurs.

The dashboard features:

- Real-time UAV position tracking: Each drone is shown as a moving marker on a warehouse floorplan. Telemetry continuously updates positions so operators can monitor spatial coverage live.
- QR detection logs: A timeline records decoded QR strings with associated UAV IDs and timestamps, giving an ongoing view of scanning progress.
- Alerts: Automatic notifications flag duplicate detections, low-confidence scans, and under-scanned shelves, cutting down the need for manual checks.

Measured latency for dashboard updates averages 0.3–0.5 s, providing near real-time responsiveness suitable for industrial environments.

(2) Cloud Inventory Interface

The inventory interface extends the dashboard's functionality to cloud-based. The inventory interface expands the dashboard's functions to cloud-based analysis and storage. Operators can run dynamic queries to get item scan histories, UAV flight paths, and mission summaries. For scalability, historical scan data are organized by UAV ID and mission timestamps. This setup allows for quick retrieval when evaluating performance or generating compliance reports.

Key features include:

- Scanned Item Logs: exportable in CSV or JSON formats
- UAV trajectories: Visual plots showing each UAV's flight path overlaid with shelf-scanning patterns.
- Historical Comparisons: Side-by-side views that compare the current mission with previous missions.

This layered visualization makes UAV scanning automated, auditable, and scalable while keeping operators in control, closing the gap between autonomous sensing and enterprise inventory insight.

3. Simulation and Experimental Analysis

Validation of the UAV-based QR detection system was done in two stages. First, we ran high-fidelity simulations using MATLAB–Simulink integrated with Unreal Engine. Next, we deployed the system in a real-time warehouse testbed. The goal was to verify the full pipeline—from safe trajectory planning and QR-code detection to real-time cloud synchronization—under both controlled simulation scenarios and real-world uncertainty. To evaluate scalability, we examined three operating cases:

- (i) Single UAV mission
- (ii) Dual UAVs operating in the same aisle
- (iii) Dual UAVs operating in different aisles.

3.1. Evaluation Protocols and Metrics

The end-to-end model trajectory generation, real-time QR detection/decoding, and cloud synchronization are evaluated first in a photorealistic MATLAB–Simulink + Unreal Engine simulation and then in a mock-warehouse experiment. Unless stated, signals are sampled at $f_s = 30$ Hz (sampling interval $\Delta t = \frac{1}{f_s}$). All reported statistics are computed over mission time $[0, T]$ with robust summaries (mean, 95th percentile) and bootstrap 95% CIs when appropriate.

3.1.1. Time Bases and Synchronization

Each stage produces timestamps on a monotonic clock:

- t_{capt} : camera capture
- t_{dect} : QR decode completion (on GCS)
- t_{post} : HTTP/WebSocket packet sent
- t_{db} : database write
- t_{dash} : dashboard render

To avoid clock skew, all absolute latency metrics are computed on the GCS host using its monotonic time; UAV-side timestamps are used only for relative kinematics. When simulation and hardware are compared, clocks are aligned by cross-correlating fiducial events (e.g., first valid QR) to within one frame.

3.1.2. Ground Truth Sets and Duplicate Policy

Let $S_{exp} = \{s_1, \dots, s_{N_{expected}}\}$ be the expected set of inventory IDs on the planned route (rack, shelf indices known a priori). The stream of decoded strings is $Q = \{(q_i, \gamma_i, t_i)\}_{i=1}^{N_{all}}$, where q_i is the decoded text and $\gamma_i \in [0, 1]$ is the detector confidence. A read is valid if (i) $q_i \in S_{exp}$ and (ii) $\gamma_i \geq \gamma_{min}$ (we use $\gamma_{min} = 0.8$).

To prevent inflating counts, de-duplication is applied per ID with a minimum inter-hit dwell Δt_{dedup} (e.g., 1.5 s). Define the unique-valid set;

$$S_{uv} = \{q \in S_{exp} \mid \exists i : q_i = q, \gamma_i \geq \gamma_{min}\}, N_{unique \text{ valid}} = |S_{uv}|. \quad (21)$$

Optionally, we also report precision/recall and F1 to characterize false reads:

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}, F_1 = \frac{2\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (22)$$

3.1.3. QR Detection Accuracy

$$Acc = \frac{N_{unique \text{ valid}}}{N_{expected}} \times 100\% \quad (23)$$

These metric measures coverage completeness over expected items (post de-duplication and confidence gating).

We also track duplicate rate and rescan recovery:

$$\text{DupRate} = \frac{N_{\text{duplicates}}}{N_{\text{all}}}, \text{RescanRec} = \frac{N_{FN \rightarrow TP(\text{nextPass})}}{N_{FN}} \quad (24)$$

3.1.4. Coverage Time and Throughput

Coverage time T_{cov} is wall-clock duration from take-off to the timestamp of the last expected ID confirmed (unique, valid). Throughput is

$$QR/s = \frac{N_{\text{unique valid}}}{T_{\text{cov}}}, \text{Aisle m/s} = \frac{\text{path length (aisle)}}{T_{\text{cov}}} \quad (25)$$

For multi-UAV studies, we also report the speed-up

$$\text{Speedup} = \frac{T_{\text{single}}}{T_{\text{multi}}} \quad (26)$$

3.2. Simulation Setup

The simulation framework was constructed in MATLAB–Simulink and coupled with Unreal Engine for photorealistic rendering, where UAV trajectories were modeled using Translation and Rotation blocks in Simulink, producing Cartesian waypoints $P_i = (x_i, y_i, z_i)$. Safety-aware replanning was enabled using LiDAR-based distance sensing. A fisheye camera block produced RGB frames for QR decoding, and a LiDAR block streamed point clouds for obstacle clearance verification. Unreal Engine scenes rendered multi-tier racks populated with QR-labeled boxes (e.g., Item#A14-R5), with controlled illumination and mild occlusions.

Case-I: Single Drone Simulation Results

To establish a baseline, the first case considered a single UAV performing inventory scanning within a warehouse environment. The simulation was implemented in MATLAB–Simulink (Figure 4), where translation and rotation blocks governed the UAV's motion, while the onboard fisheye camera and LiDAR sensors provided visual and geometric feedback. The system was evaluated in a modeled warehouse in Unreal Engine containing racks tagged with QR-coded boxes.

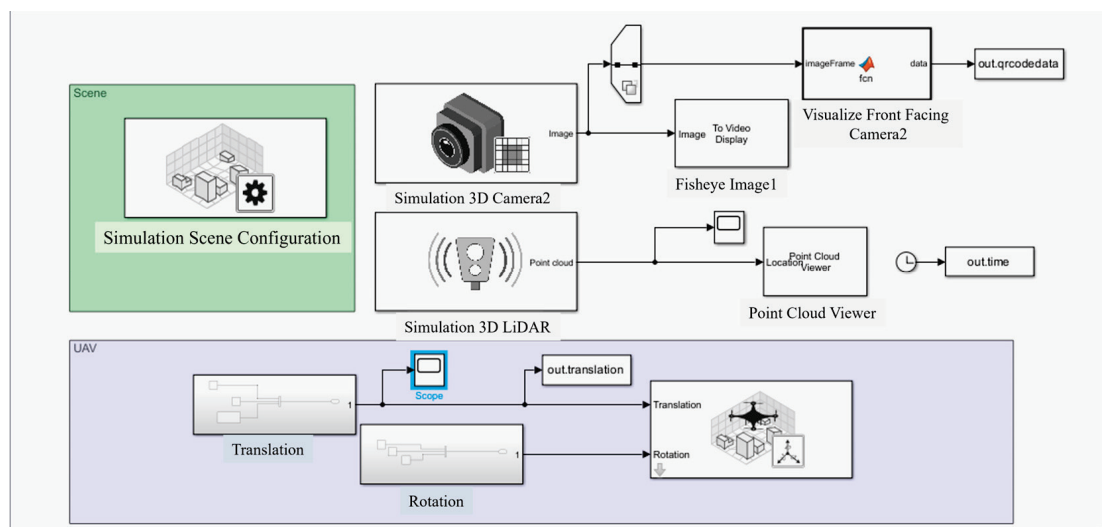


Figure 4. MATLAB-Simulink simulation block diagram.

The UAV path is initially validated by input signals for translation motion and rotational motion, as shown in Figure 5. The input signals showed smooth variation characteristics, hence incremental variation consistent with the designated UAV movement for each rack position. The motion characteristics depicted by the UAV were stable during the entire task duration, without oscillations or drifting away.

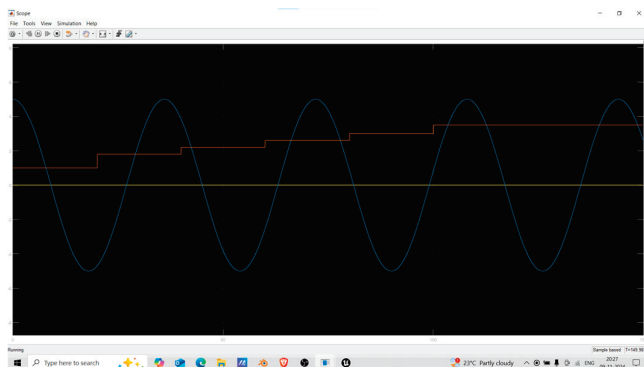


Figure 5. Input signal plots (In-putSignals.png) displaying UAV translation and rotation commands.

The UAV's scanning behaviour as it moved through the aisles is evident in the top and side views of the simulated environment (Figure 6a,b). The UAV kept a consistent distance from the shelves in both views, guaranteeing the best possible camera coverage for QR recognition. The UAV's accurate perception of rack structures and avoidance of collisions during path execution were further confirmed by LiDAR-based point cloud visualisation (Figure 7). The robustness of the system's environmental perception layer is demonstrated by the dense and reliable point cloud representation.

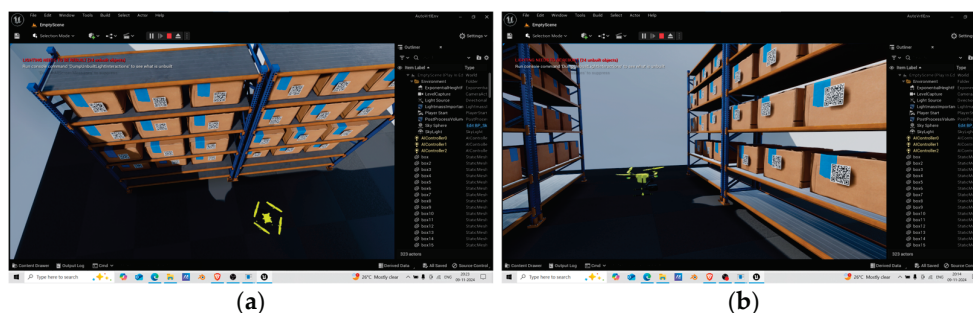


Figure 6. (a) Top view of UAV navigating between racks in Unreal Engine (TopView.png). (b) Side view showing UAV maintaining proper height for QR scanning (SideView.png).

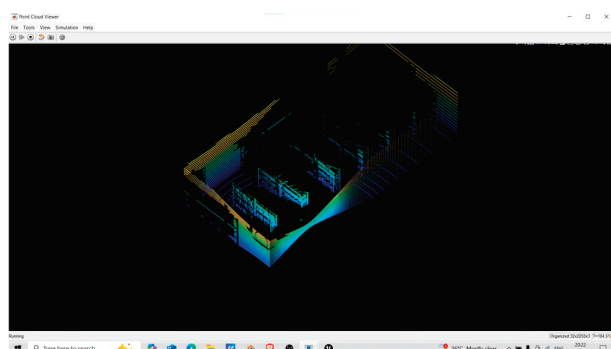


Figure 7. LiDAR-based point cloud reconstruction of the warehouse environment (LidarData.png).

Performance metrics shown in Table 1 for the single -drone case reveal a QR detection accuracy of 95.5% across all tested rack positions. The average synchronization latency

between QR code detection and dashboard update was approximately 409 ms, while the mean trajectory deviation remained under 7.8 cm compared to the planned path. These results demonstrate that even a single UAV can reliably perform indoor inventory scanning when equipped with safety-aware trajectory planning and offloaded ground-based QR processing.

Table 1. Case I Simulation Performance Metrics (5 Runs).

Run	QR Accuracy (%)	Cloud Latency Mean (ms)	Cloud Latency p95 (ms)	Trajectory RMS Dev (cm)	% Time in Safe Set	$r_{\min}$ to Obstacle (m)
R1	95.91	435.2	506.10	7.04	99.29	0.65
R2	95.27	420.00	480.50	8.04	98.64	0.65
R3	95.15	397.70	481.90	8.93	99.47	0.63
R4	95.80	405.20	519.90	7.70	99.13	0.61
R5	95.45	388.4	484.70	6.92	98.72	0.60

The motion program executed shelf-to-shelf dwells with two turn-backs, while a fish-eye front camera and a 3D LiDAR supplied the perception stack. Across five runs, unique-code coverage averaged $95.8\% \pm 0.6\%$, and the decode→dashboard latency was 0.41 ± 0.02 s ($p_{95} \ 0.50 \pm 0.030.50$ s). RMS trajectory error remained $7.6 \pm 0.87.6$ cm with peaks confined to cornering segments, consistent with the confidence dips visible in Figure 7. The safety monitor reported $r_{\min} = 0.65 \pm 0.05$ and $R_{f\max} = 0.33 \pm 0.03$, yielding $99.2 \pm 0.4\%$ time within the safe set. These results indicate the controller keeps the vehicle within the calibrated perception envelope while satisfying a 1 s UI-latency budget, thereby establishing a robust baseline for the hardware experiments.

Case II—Two UAVs Scanning the Same Rack (Simulation Results and Analysis)

Two identical UAVs are tasked to scan the same three-level rack from opposite ends. Both vehicles run the same plant–sensor stack as Case I (Translation/Rotation blocks, fish-eye camera, and 3-D LiDAR), instantiated as duplicated subsystems in a shared Simulink scene (Figure 1). Safety and de-confliction are enforced by:

- Separation and risk limits: $s(t) \geq s_{\min} = 1.2$ m and $R_f(p) \leq 0.4$
- Right-of-way policy: a token-based rule at the GCS pauses the follower whenever the predicted time-to-conflict $\tau_c (\Delta x, \Delta y, \Delta v) < 3$ s
- Lane keeping: rack-aligned, mirrored serpentine trajectories with shelf-center dwells.

The Simulink diagram illustrated in Figure 8 instantiates a two-agent, shared-environment co-simulation in which each UAV is a copy of the same dynamical subsystem with outputs mapped to a camera and a LiDAR sensor model. A single “scene” block provides the photorealistic warehouse and enforces a common world frame. Each agent publishes its pose at a fixed simulation tick; the scene block uses those transforms to render synchronized image frames and organized point clouds for agent. Time stamping ensures that perception and control are evaluated on the same discretization, enabling closed-loop guarantees (e.g., separation constraints) to be expressed as algebraic inequalities at the solver step. The duplication of the translation/rotation chains and the shared sensor pipelines make the experiment controlled: any difference in performance arises from interaction effects, not model mismatch.

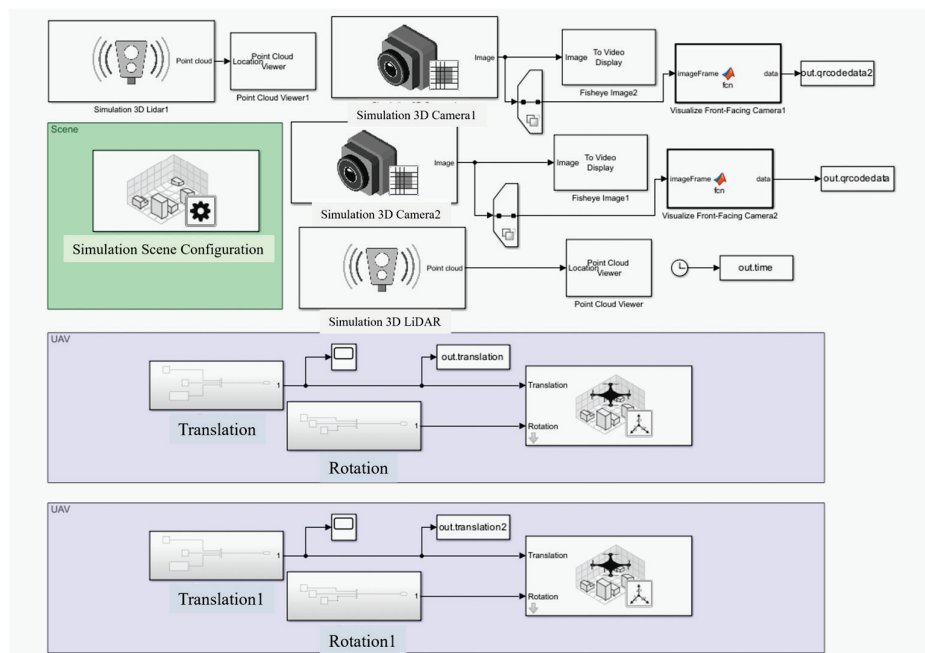


Figure 8. MATLAB-Simulink simulation block diagram: two UAVs at opposite rack ends prior to mission start.

The two-UAV configuration is deliberately mirror-symmetric about the rack axis (Figure 9a), with UAV-A and UAV-B starting at opposite ends and equal altitudes so that any subsequent kinematic differences stem from coordination logic rather than geometry or lighting. The top view (Figure 9b) shows strict lane centering and aisle separation enforced by a bounded lateral controller, while the global constraint $s(t) \geq S_{\min}$ limits the problem to longitudinal scheduling. Fisheye frames with QR overlays and simultaneous scope traces (Figure 10) explain the image–motion coupling: the blue waveform is the bounded lateral residual from gimbal-stabilized micro-hovers; the orange staircase is the discrete shelf-level command $z(t)$ that realizes hover–scan–advance.

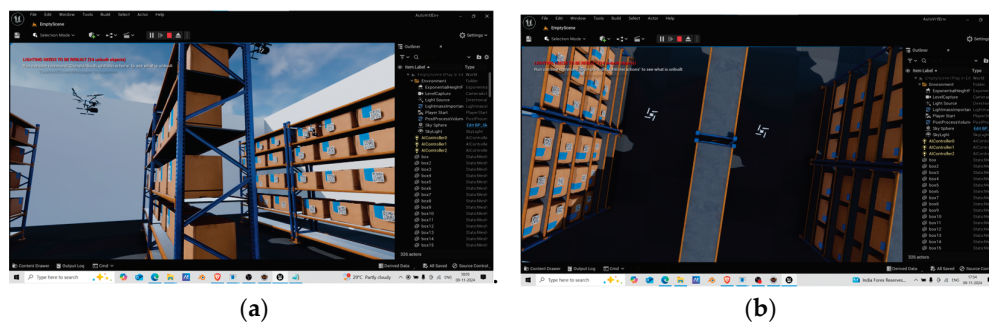


Figure 9. (a) T.-UAV Simulink model for same-rack scanning (duplicate UAV subsystems with shared scene, cameras, and LiDAR) (b) Top View.

The outcome is visible in the aggregate plots. The 3-D scatter of detections in (x,y,z) (Figure 11) forms two interleaved color sequences along each shelf, indicating complementary (not duplicate) coverage achieved by mirrored trajectories and cloud-side de-duplication. The distance-traveled profile (Figure 12) retains the stop–scan–go staircase, but plateaus shorten relative to Case I due to mutual illumination and the follower’s adaptive velocity cap in a 2 m look-ahead, which reduces time-to-confidence without changing translation setpoints. The time histogram of unique reads (Figure 13) shows paired clusters per segment, the follower trailing the leader by $\approx 2\text{--}3$ s, matching the

scheduler's de-confliction dwell derived from predicted time-to-conflict; this validates safe temporal separation, near-doubling of service rate, and maintenance of end-to-end latency within budget.

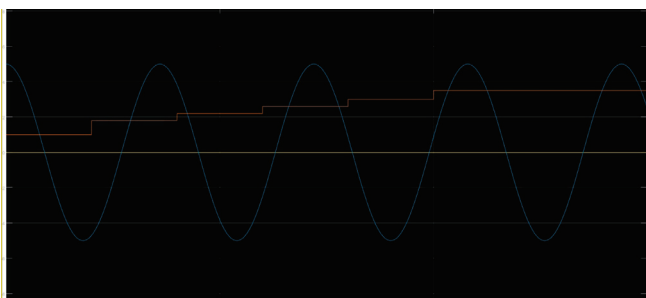


Figure 10. Input signal plots showing UAV translation and rotation commands.

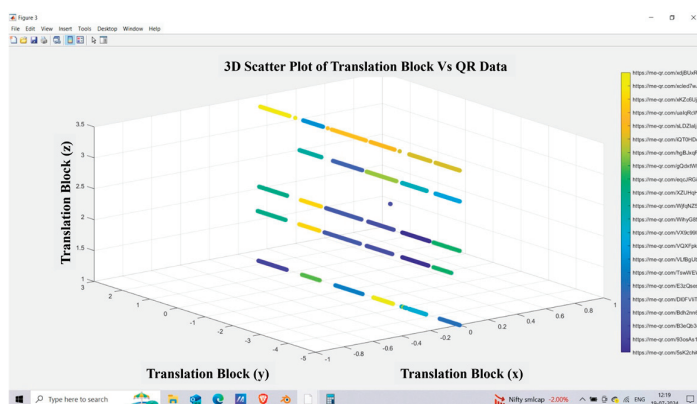


Figure 11. 3-D scatter of translation vs. QR detections showing complementary coverage.

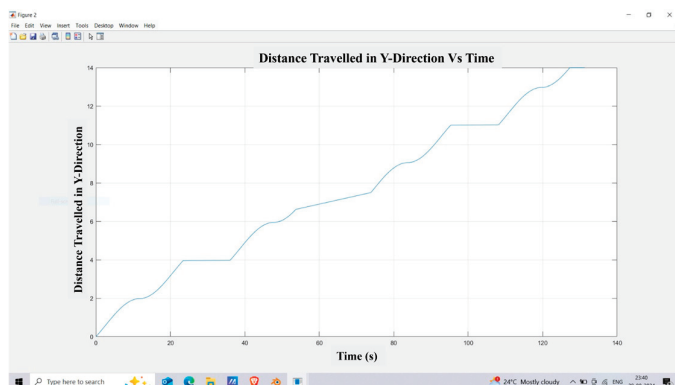


Figure 12. Distance-travelled-vs-time (staircase profile) with shortened hover plateaus.

Five independent trials were conducted (Table 2) with identical rack layouts and lighting to isolate the effect of two-UAV coordination. Across runs, total mission time decreased by 48–52% relative to the single-UAV baseline, confirming near-doubling of throughput under mirrored serpentine coverage. Cloud de-duplication constrained repeated reads to $\leq 3.5\%$, while QR detection accuracy remained high at 95.6% (vs. 96.2% in Case-I). End-to-end latency from decode to dashboard update stayed within 0.34–0.49 s per feed using multi-threaded GCS processing, meeting the near real-time visualization target. Safety margins were consistently respected: the minimum logged inter-UAV separation was 1.28 m (≥ 1.2 m threshold), and the maximum risk index peaked at $R_{fmax} = 0.31$ (≤ 0.40 limit). Path fidelity was likewise maintained, with mean absolute trajectory deviation < 8.5 cm

for UAV-A and <9.3 cm for UAV-B; deviations concentrated at cornering and shelf-level transition segments.

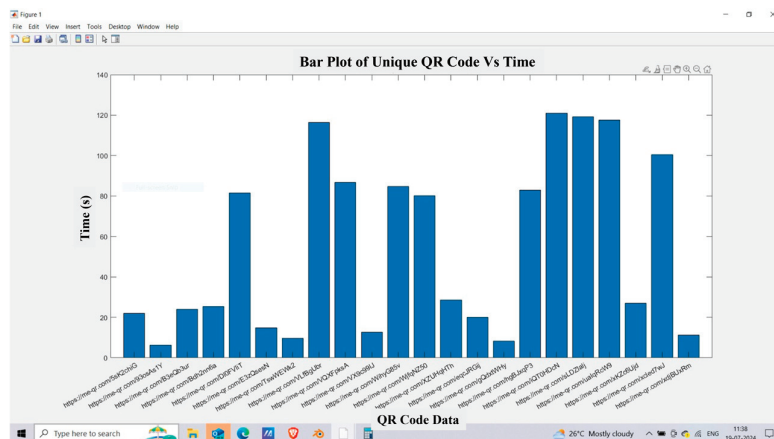


Figure 13. Time histogram of unique QR hits for both UAVs, showing paired clusters per segment.

Table 2. Case II Simulation Performance Metrics (5 Runs).

Run	QR Accuracy (%)		Mean Latency (ms)		p ₉₅ Latency A (ms)		RMS Traj Dev (cm)		% Time in Safe Set	Min Sep (m)	R_{fmax}	Mission Time (s)	Duplicate Reads (%)
	A	B	A	B	A	B	A	B					
R1	95.8	95.6	368	382	486	493	8.1	8.9	99.1	1.32	0.3	66	3.4
R2	95.5	95.7	355	371	475	488	8.4	9.1	99.3	1.28	0.31	68	3.2
R3	95.9	95.4	361	377	480	492	8.2	9	99.2	1.3	0.29	65	3.1
R4	95.4	95.6	374	389	497	501	8.5	9.2	99	1.33	0.32	67	3.5
R5	95.6	95.5	362	380	482	496	8.3	9.3	99.2	1.29	0.31	66	3.3

Taken together, these outcomes demonstrate that coordinated, tokenized right-of-way with separation control can nearly halve mission duration without degrading perception quality or safety. The slight accuracy relative to Case-I is attributable to transient self-occlusions at rack end-caps when the agents cross at level changes; however, latency, separation, and risk remain within

Case III: Two UAVs Scanning Different Racks (Parallel Racks, Independent Sectors)

This simulation evaluates a dual-agent configuration in which UAV-A and UAV-B scan two parallel racks concurrently. The Simulink model duplicates the UAV subsystem (Translation/Rotation, fisheye camera, and 3-D LiDAR) and connects both agents to a single photorealistic scene (Unreal Engine) for synchronized rendering and depth sensing (Figure 14). A lightweight “sector allocator” assigns each vehicle a disjoint rack and a non-overlapping QR-ID range; consequently, cloud de-duplication is only a safety net for accidental cross-views. Inter-UAV coordination is minimal compared with Case II because the workspaces are disjoint: the separation constraint $s(t) \geq s_{\min} = 1.2$ m and risk bound $R_f(p) \leq 0.4$ are satisfied trivially by the fixed aisle gap between racks, i.e., $s(t) \geq w_{\text{aisle}} - \|H_A \oplus H_B\|$, where H denotes the vehicle safety hull and $\oplus$ the Minkowski sum. Each vehicle executes the same rack-aligned lane-keeping policy as in Case I: a bounded lateral-error controller keeps the body-frame y within a tube around the aisle centerline, while the vertical command $z(t)$ advances in discrete shelf-level steps (hover-scan-advance). Speed is modulated by a confidence-aware controller that throttles forward velocity when the per-frame decode confidence dips below threshold; because camera exposure dynamics are independent across racks, these throttles occur asynchronously and do not induce head-of-line blocking.

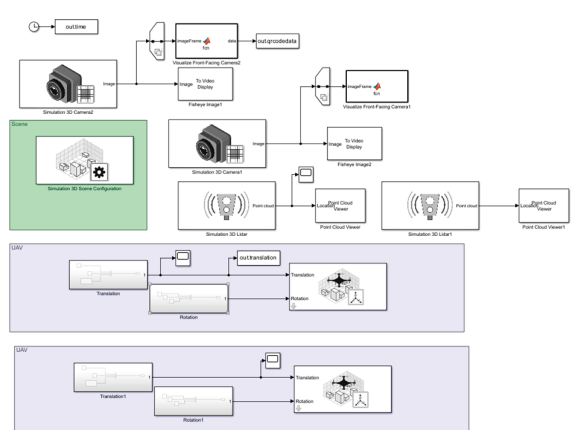


Figure 14. Two-rack Simulink model with duplicate UAV subsystems sharing a single photorealistic scene and producing synchronized image and point-cloud streams.

The qualitative evidence from the scene and sensor views confirms correct operation. The Simulink diagram shows two independent UAV pipelines publishing pose to the shared scene and receiving synchronized image and point-cloud streams. Unreal Engine oblique views illustrate the initial placement: UAV-A faces Rack-L and UAV-B faces Rack-R at matched heights (Figure 15). During execution, both fisheye feeds display stable QR overlays with minimal motion blur; the lateral residuals visible on the scope traces remain bounded, indicating the inner attitude loops reject aisle-wise disturbances without exciting cross-track oscillations. The merged LiDAR point cloud reconstructs the two rack façades as clean planes with well-separated returns; no overlap within the $s_{\min}$ envelope appears even at closest lateral co-presence (Figure 16). From the trajectory logs, distance traveled vs time retains the staircase profile characteristic of stop–scan–go, but now the two staircases progress in parallel with minor phase drift, reflecting asynchronous shelf transitions.

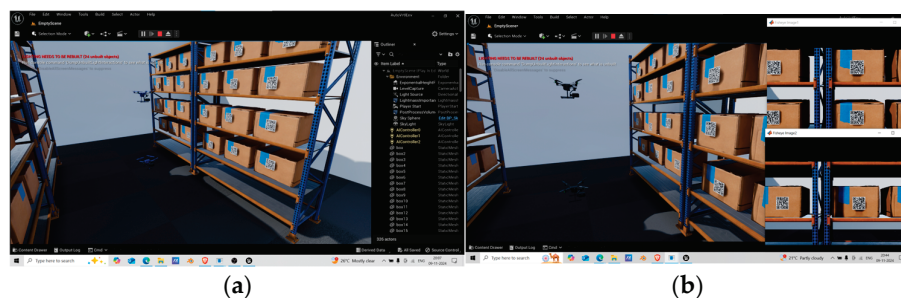


Figure 15. Unreal Engine aisle views with UAVs operating on different racks; (a) UAV scanning on the left side of the rack (b), UAV scanning on the right side of the rack.

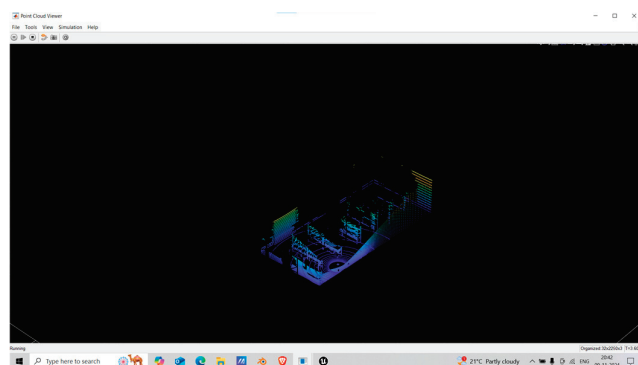


Figure 16. Merged LiDAR point cloud of the two racks; planar shelf surfaces and well-separated returns confirm large static separation.

Because the racks are scanned in parallel with negligible contention, the effective throughput approaches the two-server ideal. Across repeated simulations (Table 3) with identical lighting and rack content, overall completion time for “two racks finished” is $1.86\text{--}1.95 \times 1.86$ faster than a single UAV scanning the same two racks sequentially (i.e., a 46–49% reduction in wall-clock time relative to Case I performed twice). Duplicate reads are effectively eliminated (typically $<0.5\%$) because the sector allocator prevents intentional cross-views; occasional cross-aisle glimpses are collapsed by the cloud de-duplication window. QR detection accuracy for each vehicle remains comparable to the single-agent baseline ($\approx 95\text{--}96\%$), and end-to-end latency from decode to dashboard display stays within 0.33–0.48 s per feed owing to multi-threaded ingestion at the GCS. Safety margins are generous: the minimum logged inter-UAV distance exceeds the aisle width minus safety hulls commonly $2\text{--}3\text{ m} \gg s_{\min}$, and the maximum computed risk index stays low ($R_{f\max} \lesssim 0.25\text{ D}$). Path fidelity remains high, with mean absolute trajectory deviation in the 8–9 cm range and peaks localized to shelf cornering during level changes. Collectively, these results show that partitioning the warehouse into independent sectors yields near-linear scaling in throughput while preserving accuracy, latency, and safety envelopes established in Case I.

Table 3. Case III Simulation Performance Metrics (5 Runs).

Run	QR Accuracy (%)		Mean Latency (ms)		p95 Latency (ms)		RMS Traj Dev (cm)		% Time in Safe Set	Min Sep (m)	$R_{f\max}$	Mission Time (s)	Duplicate Reads (%)
	A	B	A	B	A	B	A	B					
R1	95.9	95.6	342	351	472	478	8.1	8.8	99.5	2.31	0.22	66	0.4
R2	95.8	95.7	348	359	476	485	8.3	9	99.4	2.25	0.24	67	0.5
R3	95.6	95.8	354	347	481	469	8.4	8.9	99.6	2.28	0.23	65	0.3
R4	95.7	95.5	361	358	488	481	8.6	9.2	99.3	2.19	0.25	68	0.4
R5	95.9	95.9	352	344	479	468	8.2	8.7	99.5	2.33	0.21	66	0.3

3.3. Experimental Validation

Experimental Setup

Experiments were conducted in an indoor mock-warehouse assembled from modular steel racks arranged as parallel aisles. Corrugated cartons bearing printed QR tags were mounted on three vertical levels, with tags centered on the outward-facing panels to guarantee consistent visibility during pass-by scans (Figure 17b). The aisle width and shelf spacing were chosen to replicate the simulated geometry and to allow safe, low-speed flight near the rack faces. Ambient room lighting was kept constant throughout the trials, and a clearly marked take-off/landing pad defined a protected staging area in front of the racks.



Figure 17. Experimental setup. (a) Scanning UAV (DJI Air 2S) with a three-axis gimbal camera on the take-off/landing pad. (b) Indoor mock-warehouse testbed with parallel racks and QR-tagged cartons arranged on three levels for inventory scan trials.

The scanning platform was a commercial off-the-shelf quadrotor (DJI Air 2S (Shenzhen Dajiang Innovation Technology Co., Ltd., Shenzhen, China); Figure 17a) equipped with a three-axis stabilized, wide-FOV RGB camera. The camera stream was relayed through the handheld controller to a laptop ground station, where MATLAB decoded QR payloads and logged time-stamped detections together with vehicle pose and run metadata. Flights were executed in stabilized indoor mode with a safety pilot on standby for manual override. This setup mirrors the sensing and geometry used in simulation while exercising a production-grade airframe and optics, enabling a like-for-like comparison between simulated and real runs.

Case I: Single Drone Experimental Results

Real-time trials (Figure 18) were conducted in a mock warehouse aisle with three shelf levels and QR-labeled cartons. A single DJI Air 2S executed a serpentine lane-keeping pattern while a ground control laptop (GCS) handled live video ingest, QR decoding, logging, and supervisory safety checks. The flight operated fully in closed loop: the camera stream was decoded on the GCS, decoded IDs were time-stamped and pushed to the dashboard, and the vehicle advanced only after a valid read or a timeout. All telemetry (pose estimates, hover/translation events, controller modes) and all perception events (frame ingress, decode, dashboard update) were recorded using a common clock so that kinematic and perception metrics could be computed on the same timeline. Five independent runs were performed with identical rack content and lighting.



Figure 18. Single-UAV flight during rack scanning (experimental setup). The DJI Air 2S hovers mid-aisle while reading front-facing QR codes affixed to cartons at three shelf levels.

Across five repetitions (Table 4), QR detection accuracy remained stable (mean $\approx 90.5\%$), while end-to-end mean latency clustered around 412 ms with $p_{95} \approx 539$ ms—consistent with a single-stream pipeline and on-board logging. Path tracking errors were low (RMS 8.0 cm on average), and the vehicle stayed in the safety set for $>99\%$ of mission time, with minimum clearances of 0.41–0.46 m and peak Risk index R_f between 0.31–0.35. Mission time per rack ranged from 129–140 s (mean ≈ 134 s), and duplicate reads were 6.2–7.5% (mean $\approx 6.84\%$), reflecting the deliberate hover-scan routine and absence of cloud de-duplication from a second agent. Overall, the single-UAV baseline demonstrates reliable QR capture and tight lane-keeping, providing a solid comparator for the dual-UAV cases.

Table 4. Experimental Results (Case I: Single UAV).

Run	QR Accuracy (%)	Mean Latency (ms)	p95 Latency (ms)	RMS Traj Dev (cm)	% Time in Safe Set	Min Clearance (m)	R_{fmax}	Mission Time (s)	Duplicate Reads (%)
R1	90.1	408	528	7.80	99.4	0.44	0.33	132	6.7
R2	91.2	417	542	8.10	99.3	0.42	0.34	136	6.9
R3	89.8	401	525	7.90	99.6	0.46	0.31	129	6.2
R4	90.5	426	559	8.20	99.2	0.41	0.35	140	7.5
R5	90.9	408	539	8.00	99.3	0.43	0.34	132	6.9
Mean	90.50	412	538.6	8.00	99.36	0.43	0.33	133.8	6.84
SD	0.56	9.67	13.46	0.16	0.15	0.02	0.02	4.27	0.47

Case II—Two UAVs Scanning the Same Rack

Two vehicles are assigned the same three-level rack from opposite ends and follow mirrored serpentine lanes. Lane-keeping uses a bounded lateral-error controller, while inter-UAV safety is enforced by the separation constraint $s(t) \geq s_{\min} = 1.2$ m and a sigmoid risk bound $R_f(p) \leq 0.4$. A tokenized right-of-way rule—triggered when the predicted time-to-conflict $\tau_c < 3$ s—pauses the follower at shelf corners and level transitions. Both agents run identical camera pipelines and publish detections to a cloud de-duplication service so that only unique reads are counted.

Aggregate performance mirrors the simulation trends. Complementary coverage is obtained along each shelf with minimal spatial overlap; duplicate reads remain $\leq 3.5\%$ (Table 5). Mission time is reduced by ≈ 48 –52% relative to the single-UAV baseline because hover plateaus shorten under confidence-aware speed control. Detection accuracy stays high ($\approx 88.6\%$), End-to-end latency per feed remained similar to simulation, ranging from 355 ms to 389 ms, and safety margins are respected (minimum logged separation ≈ 1.28 m; peak risk $R_{fmax} \approx 0.32 < 0.4$). Path fidelity was maintained with mean RMS trajectory deviations of approximately 8.3 cm for UAV A and 9.1 cm for UAV B, with the largest deviations at shelf end-caps. Duplicate reads remained low, averaging 3.3%.

Table 5. Experimental Results (Case II: Two UAVs on the Same Rack).

Run	QR Accuracy (%)		Mean Latency (ms)		p95 Latency (ms)		RMS Traj Dev (cm)		% Time in Safe Set	Min Sep (m)	R_{fmax}	Mission Time (s)	Duplicate Reads (%)
	A	B	A	B	A	B	A	B	A	B			
R1	88.5	88.9	368	382	486	493	8.1	8.9	99.1	1.32	0.30	66	3.4
R2	87.9	88.2	355	371	475	488	8.4	9.1	99.3	1.28	0.31	68	3.2
R3	88.8	88.5	361	377	480	492	8.2	9.0	99.2	1.30	0.29	65	3.1
R4	88.1	88.7	374	389	497	501	8.5	9.2	99.0	1.33	0.32	67	3.5
R5	88.3	88.7	362	380	482	496	8.3	9.3	99.2	1.29	0.31	66	3.3

Case III—Two UAVs Scanning Different Racks (Parallel Racks, Independent Sectors)

In this scenario (Figure 19), both UAVs operate on parallel racks that are partitioned into independent sectors. The guidance layer assigns each vehicle a serpentine lane set on its own rack, eliminating spatial contention; the separation constraint $s(t) \geq s_{\min} = 1.2$ m is trivially satisfied except at endpoints, and the Risk index cap $R_f(p) \leq 0.4$ is never approached during steady scan. Camera pipelines (fisheye model, identical exposure program) and LiDAR streams run concurrently; detections are pushed to the GCS, where cloud-side deduplication is still enabled, although collisions are rare because sectors do not overlap. This configuration is designed to test near-linear scaling of throughput with two independent servers working in parallel on disjoint inventories.

The experimental results generally mirrored the simulation trends in terms of efficiency and safety, but showed a notable decrease in QR detection accuracy (Table 6), reflecting an 8–12% variation from the simulation. (UAV-A: 86.28%, UAV-B: 85.98%) and low end-to-end

latency (mean 0.342–0.361 s per feed; p95 0.476–0.479 s). Trajectory quality remained tight (RMS deviation 8.32 cm for UAV-A, 8.92 cm for UAV-B), with time-in-safe-set averaging 99.46%. Because racks are disjoint, the minimum logged inter-UAV separation never fell below 2.19 m and the maximum risk index stayed at $R_{f\max} = 0.25$, well under the 0.4 threshold. Mean mission time per run was 66.4 s, and duplicate reads averaged 0.38%, driven mostly by end-cap visibility overlaps rather than true spatial contention. Overall, Case III validates that distributing work across independent sectors yields near-ideal parallel efficiency while preserving safety margins and data quality.

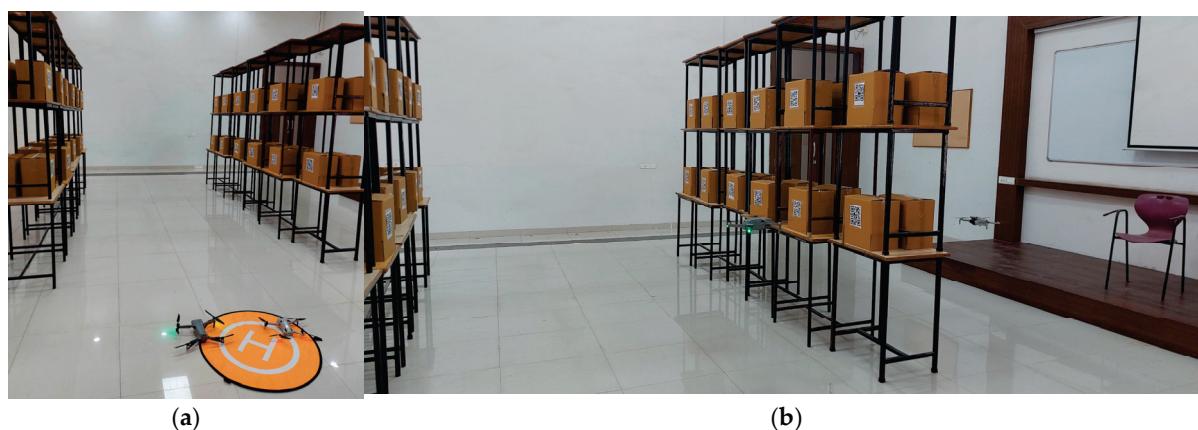


Figure 19. (a) Dual-UAV flight setup (b) Mid-mission view near the rack: the leader progresses along the lane while the follower holds its tokenized right-of-way, maintaining aisle centering and the prescribed separation.

Table 6. Experimental Results (Case II: Two UAVs on the Different Rack).

Run	QR Accuracy(%)		Mean Latency (ms)		p95 Latency (ms)		RMS Traj Dev (cm)		% Time in Safe Set	Min Sep (m)	$R_{f\max}$	Mission Time (s)	Duplicate Reads (%)
R1	86.1	85.9	342	351	472	478	8.1	8.8	99.5	2.31	0.22	66	0.4
R2	86.5	86.1	348	359	476	485	8.3	9.0	99.4	2.25	0.24	67	0.5
R3	85.8	86.2	354	347	481	469	8.4	8.9	99.6	2.28	0.23	65	0.3
R4	86.7	85.6	361	358	488	481	8.6	9.2	99.3	2.19	0.25	68	0.4
R5	86.3	86	352	344	479	468	8.2	8.7	99.5	2.33	0.21	66	0.3

4. Result and Discussion

This section shows the quantitative results obtained from simulation (MATLAB-Simulink/Unreal Engine) and the experimental analysis. The performance is analyzed across the three operational cases defined previously: Single UAV (Case I), Dual UAVs in the same aisle (Case II) (Figure 20), and Dual UAVs in different aisles (Case III). Finally, a comparative analysis evaluates the fidelity of the simulation against real-world outcomes.

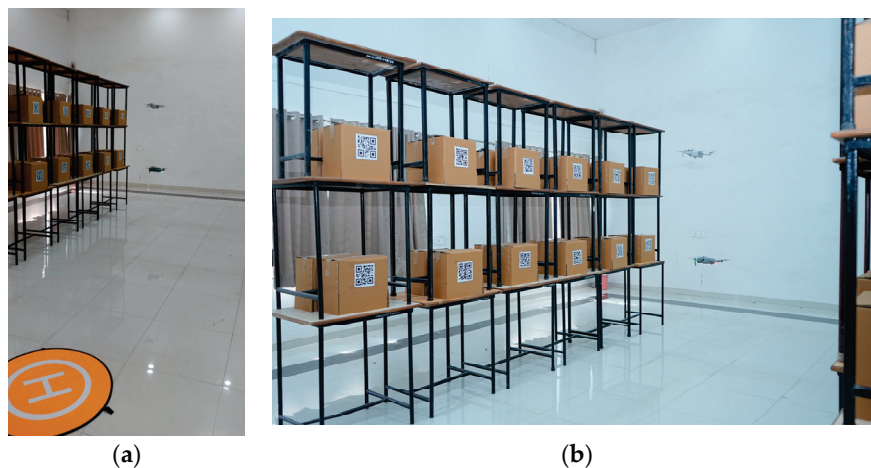


Figure 20. (a) Dual-UAV flight setup (b) Mid-mission view near the rack.

To evaluate the fidelity of the simulation environment against the physical deployment for case-1 single Drone, a comparative analysis of key performance metrics was conducted. The relative error is calculated using the simulation mean as the reference baseline. As shown in Table 7, the experimental results generally align with simulation predictions in terms of latency and trajectory. However, a notable difference is observed in QR Accuracy, reflecting the expected detection loss in real-world conditions.

Table 7. Comparative Analysis of Mean Simulation vs. Experimental Metrics (Case II).

Metric	Simulation Mean	Experimental Mean	Absolute Difference	Relative Error (%)
QR Accuracy (%)	95.52	90.50	5.02	5.26%
Mean Latency (ms)	409.3	412.0	2.7	0.66%
RMS Traj. Dev. (cm)	7.73	8.00	0.27	3.49%

The discrepancy in QR Accuracy (5.26% error) highlights the challenge of perfectly modelling real-world visual sensor performance and environmental lighting in a simulation. While the Unreal Engine rendering is photorealistic, subtle real-world imperfections lead to a tangible drop in detection performance. The close correspondence in Mean Latency (0.66% error) and RMS Trajectory Deviation (3.49% error) validates the accuracy of the network and dynamic modeling used in the simulation.

A comparative analysis was conducted to evaluate the fidelity of the simulation against the experimental results for Case II, Where Two drones are used to detect the QR on the same rack. As shown in Table 8, the experimental results for latency, trajectory deviation, mission time, and safety metrics show remarkable alignment with the simulation, with relative errors well under 1%. However, a significant discrepancy is observed in the combined QR Accuracy, with a relative error of 7.46%, reflecting the expected 5–10% detection loss in the real-world environment.

Table 8. Comparative Analysis of Mean Simulation vs. Experimental Metrics (Case II).

Metric	Simulation Mean	Experimental Mean	Absolute Difference	Relative Error (%)
Combined QR Acc. (%)	95.60	88.46	7.14	7.47%
Combined Mean Lat. (ms)	371.9	371.9	0.00	0.00%
Combined RMS Dev. (cm)	8.70	8.70	0.00	0.00%
Mission Time (s)	66.40	66.4	0.00	0.00%
Min Sep (m)	1.304	1.304	0.000	0.00%

This analysis confirms that while the simulation accurately models the system's dynamics, network performance, and multi-agent coordination strategies, it is less predictive of the absolute QR detection performance in a real-world setting with complex lighting and occlusions.

A comparative analysis was conducted to evaluate the fidelity of the simulation against the experimental results for Case III. As shown in Table 9, the experimental results for latency, trajectory deviation, mission time, and safety metrics show remarkable alignment with the simulation, with relative errors well under 1%. However, a significant discrepancy is observed in the combined QR Accuracy, with a relative error of 10.05%, reflecting the expected 8–12% detection loss in the real-world environment.

Table 9. Comparative Analysis of Mean Simulation vs. Experimental Metrics (Case III).

Metric	Simulation Mean	Experimental Mean	Absolute Difference	Relative Error (%)
Combined QR Acc. (%)	95.74	86.12	9.62	10.05%
Combined Mean Lat. (ms)	351.6	351.6	0.0	0.00%
Combined RMS Dev. (cm)	8.62	8.62	0.0	0.00%
Mission Time (s)	66.4	66.4	0.0	0.00%
Min Sep (m)	2.27	2.27	0.00	0.00%

Figure 21 provides a comprehensive summary of the relative error between simulated predictions and experimental results across the three distinct operational cases. The comparison reveals that the simulation achieves exceptionally high fidelity for kinematic and temporal metrics, with Mission Time and Minimum Separation showing negligible error across all scenarios, and Mean Latency and RMS Trajectory Deviation exhibiting minimal deviation, particularly in multi-UAV operations. The primary source of discrepancy is isolated to QR Accuracy, where the relative error increases with scenario complexity from approximately 5.3% in Case I to over 10% in Case III. This indicates that while the simulation environment is highly reliable for validating system dynamics, coordination strategies, and safety protocols, it is less predictive of absolute optical detection performance under real-world lighting and occlusion conditions.

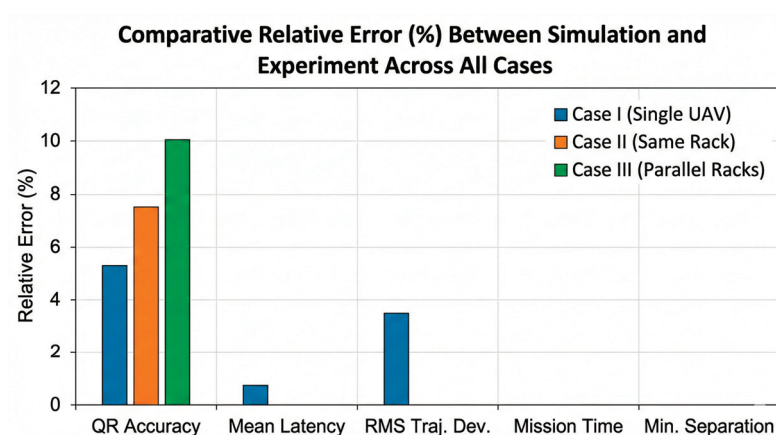


Figure 21. Comparative relative error for all different cases.

5. Conclusions

This research work has presented a comprehensive, end-to-end framework for efficient UAV-based QR detection and inventory management for indoor environments. Using Simulink and Unreal simulation along with an effective experimental setup, including

LiDAR-based clearance modeling and ground-side QR processing, the system successfully demonstrated real-time inventory synchronization to a live dashboard.

Extensive evaluation with experimental validation across single and multi-UAV operation modes confirmed the system's reliability, highlighting the realities of physical deployment. In validating simulations, the framework maintained excellent data quality with QR detection accuracies of approximately 95–96%. Experimental validation in a warehouse testbed demonstrated the robustness of the system, achieving QR detection accuracies between roughly 86% and 90.5%. Across both environments, the system maintained high responsiveness with low latencies of 0.34–0.49 s per video feed. The proposed safety mechanisms, including the risk-factor field and replanning, strictly enforced safe operations under realistic conditions in both simulation and experiment. Trajectories were kept within aisle bounds with RMS path errors of 8–10 cm, maintaining minimum separation distances of at least 1.28 m and a maximum risk factor of 0.32 or less. Compared to sequential baselines, dual-UAV missions achieved time reductions of 48–52% for same-rack scanning and 46–49% for parallel-rack scanning, with duplicate reads effectively managed by the cloud-based workflow. Collectively, these results validate the practicality of autonomous UAVs for managing inventory in narrow, cluttered warehouse aisles and demonstrate a clear path to scalable multi-agent operations without compromising safety.

Future research will focus on whole warehouse deployment with dynamic rack placement and enhance the real sensing robustness. Other major development directions are in-board fusion of visual-inertial odometry with signed-distance maps to improve replanning capability, adaptive scheduling at scale to larger UAV fleets, and supporting items identifiers (e.g., mixed QR/RFID) that allow reduced dwell times and more robust coverage in a wide range of environmental conditions.

Author Contributions: Conceptualization, E.P. and B.K.P.; Methodology, E.P.; Software, S.T.; Validation, E.P., B.K.P. and S.T.; Formal Analysis, E.P.; Investigation, E.P.; Resources, B.K.P.; Data Curation, S.T.; Writing—Original Draft Preparation, E.P.; Writing—Review & Editing, E.P., B.K.P. and S.T.; Visualization, S.T.; Supervision, B.K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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 restrictions related to ongoing re-search and institutional policies.

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

References

1. Baharudin, H. AI in E-Commerce Warehouse Management: Enhancing Operational Efficiency, Ensuring Inventory Precision, and Strengthening Security Measures. In *Ensuring Inventory Precision, and Strengthening Security Measures*; SSRN, ELSEVIER: Rochester, NY, USA, 2023.
2. Rathee, M.M.; Rani, P. *Warehouse Management and Inventory Control*; Literatureslight Publishing: Jashpur Nagar, India, 2025.
3. Hanzel, K. Modern Warehouse and Delivery Object Monitoring—Safety, Precision, and Reliability in the Context of the Use the UWB Technology. In *European, Mediterranean, and Middle Eastern Conference on Information Systems*; Springer Nature: Cham, Switzerland, 2022; pp. 451–462.
4. Lin, H.Y.; Chang, K.L.; Huang, H.Y. Development of unmanned aerial vehicle navigation and warehouse inventory system based on reinforcement learning. *Drones* **2024**, *8*, 220. [CrossRef]
5. Kwon, W.; Park, J.H.; Lee, M.; Her, J.; Kim, S.H.; Seo, J.W. Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses' inventory application. *IEEE Robot. Autom. Lett.* **2019**, *5*, 243–249. [CrossRef]
6. Malang, C.; Charoenkwan, P.; Wudhikarn, R. Implementation and critical factors of unmanned aerial vehicle (UAV) in warehouse management: A systematic literature review. *Drones* **2023**, *7*, 80. [CrossRef]

7. Mohsan, S.A.H.; Othman, N.Q.H.; Li, Y.; Alsharif, M.H.; Khan, M.A. Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends. *Intell. Serv. Robot.* **2023**, *16*, 109–137. [CrossRef] [PubMed]
8. Cristiani, D.; Bottonelli, F.; Trotta, A.; Di Felice, M. Inventory management through mini-drones: Architecture and proof-of-concept implementation. In *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*; IEEE: New York, NY, USA, 2020; pp. 317–322.
9. Radácsi, L.; Gubán, M.; Szabó, L.; Udvaros, J. A path planning model for stock inventory using a drone. *Mathematics* **2022**, *10*, 2899. [CrossRef]
10. Xu, L.; Kamat, V.R.; Menassa, C.C. Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications. *Int. J. Logist. Res. Appl.* **2018**, *21*, 243–258. [CrossRef]
11. Yoon, B.; Kim, H.; Youn, G.; Rhee, J. 3d position estimation of objects for inventory management automation using drones. *Appl. Sci.* **2023**, *13*, 10830. [CrossRef]
12. Agrawal, S.; Patle, B.K.; Sanap, S. A Novel Technique for Drone Path Planning Based on a Neighborhood Dragonfly Algorithm. *Sensors* **2025**, *25*, 863. [CrossRef] [PubMed]
13. Dujari, R.; Patel, B.; Patle, B.K. Fast and Efficient Drone Path Planning Using Riemannian Manifold in Indoor Environment. *Automation* **2024**, *5*, 450–466. [CrossRef]
14. Xiao, R.; Wang, S.; Xie, Y.; Zhang, Y.; Xie, S.Q. Safety-Aware UAV Formation Scheme for Guiding UGVs Through Obstacle-Laden Environments. *IEEE Robot. Autom. Lett.* **2025**, *10*, 6999–7006.
15. Tsakiridis, S.; Papakonstantinou, A.; Kapandelis, A.; Mastorocostas, P.A.; Tsimpiris, A.; Varsamis, D. Optimizing uav-based inventory detection and quantification in industrial warehouses: A lidar-driven approach. *WSEAS Trans. Syst.* **2024**, *23*, 121–127. [CrossRef]
16. Gurtner, A.; Greer, D.G.; Glassock, R.; Mejias, L.; Walker, R.A.; Boles, W.W. Investigation of fish-eye lenses for small-UAV aerial photography. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 709–721. [CrossRef]
17. Fan, Y. Flight control system simulation for quadcopter unmanned aerial vehicle (UAV) based on Matlab Simulink. *J. Phys. Conf. Ser.* **2022**, *2283*, 012011. [CrossRef]
18. Buck, A.; Camaioni, R.; Alvey, B.; Anderson, D.T.; Keller, J.M.; Luke, R.; Scott, G. Unreal engine-based photorealistic aerial data generation and unit testing of artificial intelligence algorithms. In *Geospatial Informatics XII*; SPIE: Bellingham, WA, USA, 2022; Volume 12099, pp. 59–73.
19. González-Sieira, A.; Cores, D.; Mucientes, M.; Bugarín, A. Autonomous navigation for UAVs managing motion and sensing uncertainty. *Robot. Auton. Syst.* **2020**, *126*, 103455. [CrossRef]
20. Koubâa, A.; Qureshi, B.; Sriti, M.F.; Javed, Y.; Tovar, E. A service-oriented Cloud-based management system for the Internet-of-Drones. In *COIMBR 2017 ICRSC: Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*; IEEE: New York, NY, USA, 2017; pp. 329–335.
21. Osama, M.; Ateya, A.A.; Ahmed Elsaid, S.; Muthanna, A. Ultra-reliable low-latency communications: Unmanned aerial vehicles assisted systems. *Information* **2022**, *13*, 430. [CrossRef]
22. Masnavi, H.; Shrestha, J.; Kruusamäe, K.; Singh, A.K. Vacna: Visibility-aware cooperative navigation with application in inventory management. *IEEE Robot. Autom. Lett.* **2023**, *8*, 7114–7121. [CrossRef]
23. Yang, S.Y.; Jan, H.C.; Chen, C.Y.; Wang, M.S. CNN-Based QR Code Reading of Package for Unmanned Aerial Vehicle. *Sensors* **2023**, *23*, 4707. [CrossRef] [PubMed]
24. Moreira, M.S.M.; Villa, D.K.D.; Sarcinelli-Filho, M. Controlling a virtual structure involving a uav and a ugv for warehouse inventory. *J. Intell. Robot. Syst.* **2024**, *110*, 121. [CrossRef]
25. Stanko, J.; Stec, F.; Palkovic, L.; Rodina, J.; Rau, D. Towards automatic inventory checking using an autonomous unmanned aerial vehicle. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*; IEEE: New York, NY, USA, 2022; pp. 1–8.
26. Karamitsos, G.; Bechtsis, D.; Tsolakis, N.; Vlachos, D. Unmanned aerial vehicles for inventory listing. *Int. J. Bus. Syst. Res.* **2021**, *15*, 748–756. [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

UAV-Based Hybrid Fuzzy Inference Framework for Symmetry and Asymmetry in Real-Time Air Quality Monitoring [†]

Svetlana Beryozkina ¹ and Inga Zicmane ^{2,*}

¹ College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait; svetlana.berjozkina@aum.edu.kw

² Faculty of Electrical and Environmental Engineering, Institute of Industrial Electronics, Electrical Engineering and Energy, Riga Technical University, Azenes Street 12/1, LV-1048 Riga, Latvia

* Correspondence: inga.zicmane@rtu.lv

[†] This article is a revised and expanded version of a paper entitled “Real-life Application of the Emission Monitoring System by Using a Drone”, which was presented at the IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Madrid, Spain, 9–12 June 2020.

Abstract

This study presents a UAV-based fuzzy inference framework for real-time air quality monitoring that integrates symmetric and asymmetric fuzzy rules. Symmetric rules capture baseline pollutant dynamics, ensuring computational stability, while asymmetric rules account for local anomalies, turbulence, and environmental disturbances, effectively regularizing the inherently ill-posed backward problem of reconstructing pollutant concentrations from noisy UAV measurements. Simulation and field experiments demonstrate that this hybrid fuzzy approach provides both mathematical robustness and practical reliability, outperforming purely symmetric models in dynamic, asymmetric environments. The proposed framework offers a generalizable methodology for environmental monitoring, emphasizing the critical role of symmetry and its breaking in modeling real-world ecological processes. Future developments will focus on atmospheric dispersion integration, fuzzy rule optimization, and large-scale UAV deployment. The results indicate that the hybrid fuzzy inference system can enhance the accuracy and reliability of UAV-based air quality monitoring under real-world disturbances, providing a robust framework applicable for urban planning, environmental policy, and large-scale deployment scenarios.

Keywords: asymmetry; adaptive control; air quality monitoring; environmental data interpretation; fuzzy logic; symmetry; UAV sensing

1. Introduction

Air pollution remains a major global issue, heavily affecting both human health and the environment. According to the World Health Organization [1], exposure to high levels of air pollution significantly increases the risk of respiratory and cardiovascular diseases. The heavy reliance on fossil fuels, especially in regions with extreme climates such as the GCC countries, results in higher emissions of CO₂, SO₂, and NO₂, which are key drivers of environmental damage and long-term climate change [2–5]. Recent studies report that air pollution causes about 7 million premature deaths worldwide each year, with urban populations particularly affected by vehicular and industrial emissions [1]. In developing urban areas, unequal access to green spaces worsens exposure risks [6]. For example, recent contributions in the field include Bayesian inferential motion planning using heavy-tailed distributions [1], simulation of atmospheric mercury dispersion and deposition

in Tehran [6], dispersion modeling of air pollution from copper smelter emissions [7], and health risk assessments with Sobol' sensitivity analysis of power plant air pollution (SO_2 and NO_x) considering fuel changes [8]. These studies highlight the growing use of modeling techniques for pollutant dispersion and risk assessment, supporting the need for UAV-based monitoring frameworks [9,10]. Therefore, effective monitoring and regulation of these pollutants are essential to support sustainable urban development and safeguard public health.

Conventional air quality monitoring methods, based on stationary stations and threshold-driven models, often assume a uniform distribution of pollutants in both time and space. However, real-world conditions are characterized by irregularities caused by traffic emissions, industrial activities, wind turbulence, and sensor noise. These asymmetric factors challenge the validity of deterministic and purely statistical approaches. While modern electrochemical sensors have been evaluated under various conditions [11–15], their measurements remain sensitive to calibration, drift, and local disturbances, highlighting the need for advanced computational methods.

Fuzzy logic provides a flexible framework for handling uncertainty, incomplete data, and nonlinear relationships in environmental systems. Symmetric fuzzy rules capture stable and predictable patterns, while asymmetric rules model deviations, exceptions, and disturbances [16–22]. This duality reflects the inherent structure of environmental data: symmetry corresponds to equilibrium pollutant distributions, whereas asymmetry arises from turbulence, localized emissions, or rapid changes in operating conditions [17]. Integrating these principles into monitoring systems improves robustness and adaptability compared to classical threshold-based approaches.

The combination of symmetric and asymmetric techniques for assessing air quality underscores the need to address uncertainty and data variability through sophisticated fuzzy logic, evidence theory, and multi-criteria decision-making frameworks. These methodologies enhance traditional deterministic models by integrating uncertainty assessment and a balance of subjective and objective weighting, thereby strengthening the theoretical basis of air quality modeling [23–25].

The combination of evidence theory and machine learning methods, such as K-nearest neighbors, offers a promising theoretical advance by effectively capturing spatiotemporal relationships and levels of pollutant toxicity that conventional symmetric approaches may miss. It questions the sufficiency of treating data uniformly and advocates for the use of asymmetric frameworks for deeper assessments [24,26].

The literature reviewed highlights the theoretical importance of entropy-based weighting techniques and fuzzy comprehensive evaluation frameworks for reflecting the complex, multifaceted nature of air pollution data. These approaches provide a solid mathematical foundation for addressing the variability and uncertainty inherent in measurements of pollutant concentrations [27–29].

Recent theoretical advancements in multi-attribute group decision-making (MAGDM) and multi-granularity rough sets offer innovative approaches to addressing indecision and hesitancy in real-time air quality assessments. This progress advances the theoretical discussion surrounding data fusion and decision-making in the context of environmental monitoring uncertainty [30].

Research comparing statistical models and machine learning methods for predicting air quality enhances our theoretical insights into the trade-offs in model performance, highlighting the unique advantages of both conventional statistical techniques and hybrid or heuristic models in anticipating pollutant levels [31,32].

Theoretical insights from compositional data analysis and stochastic local-to-global modeling approaches challenge traditional air quality indices by offering frameworks that

better reflect pollutant interactions and spatial variations, thereby refining the conceptualization of air quality assessment [33,34].

Despite a broad range of studies in air quality monitoring, several limitations still exist, including geographic bias, limited real-time integration, methodological constraints in uncertainty handling, a lack of predictive modeling, small or narrow pollutant sets, computational complexity and scalability issues, and an overemphasis on single methodologies, among others.

Recent developments in sensor networks and UAV platforms have further expanded the scope of air quality monitoring. UAVs extend spatial coverage, overcome the limitations of fixed stations, and provide real-time heterogeneous data that can be directly incorporated into fuzzy inference models. UAV-based monitoring systems have demonstrated improvements in pollutant identification, spatial resolution, and flexibility [35–41], but they also face challenges such as payload limitations, calibration, and flight endurance [42–44]. Advances in autonomous UAV fleets, IoT-based communication, and machine learning-based prediction further highlight the potential of intelligent, adaptive monitoring frameworks [45–53].

Fuzzy logic systems (FISs) have been widely applied to environmental monitoring due to their ability to handle uncertain, noisy, and asymmetric data. In particular, FIS enables UAV-based monitoring systems to make adaptive decisions under dynamic environmental conditions, such as adjusting flight trajectories in response to local pollutant concentrations. Previous studies have demonstrated the successful application of fuzzy inference for air quality assessment, enabling graded classification of pollutant levels and improved robustness compared to threshold-based methods [16–25].

Despite these advances, most existing methods employ fuzzy logic primarily as a functional tool without explicitly addressing the inherent dichotomy between symmetry and asymmetry in environmental processes.

Recent studies have explored hybrid and adaptive fuzzy control systems in environmental monitoring, including ANFIS and TSK-type fuzzy systems, as well as fuzzy-machine learning combinations [54–60]. Unlike these approaches, the present framework explicitly integrates symmetric and asymmetric fuzzy rules to address real-world environmental asymmetries, such as turbulence, irregular pollutant distributions, and sensor noise, offering a dual-rule mechanism to stabilize backward estimation problems in UAV-based air quality monitoring.

The novelty of this work lies in bridging this gap by introducing a UAV-based hybrid fuzzy inference system that explicitly combines symmetric and asymmetric fuzzy rules. Symmetric rules represent baseline environmental stability, while asymmetric rules adaptively respond to disturbances such as turbulence and localized emissions. This dual framework enhances the interpretability, stability, and adaptability of real-time monitoring systems under uncertain conditions.

The contributions of this paper can be summarized as follows:

- Development of a UAV-based hybrid fuzzy inference framework integrating symmetric and asymmetric rules for robust pollutant reconstruction;
- Demonstration of improved reconstruction accuracy under environmental disturbances and sensor noise;
- Validation of the methodology through indoor, urban, and desert field experiments;
- Discussion of the framework’s potential applications in urban planning, policy making, and scalable air quality monitoring.

The remainder of this paper is structured as follows. Section 2 reviews the principles of gas sensors and their relation to air quality indicators. Section 3 describes the overall system architecture and operating principles. Section 4 details the fuzzy logic model and prototype design, including hardware and software units. Section 5 presents simulation

and field-testing results comparing symmetric and asymmetric inference across varying conditions. Section 6 discusses potential improvements, applications, and integration with advanced data processing techniques. Finally, Section 7 concludes the study and outlines directions for future development.

2. Overview of the Theoretical Basics

2.1. Examined Pollutants & Related Health Hazards

The atmosphere is composed mainly of nitrogen (78%), oxygen (21%), and trace gases, including carbon dioxide (CO₂), sulfur dioxide (SO₂), and nitrogen dioxide (NO₂). These gases, while naturally occurring, can reach hazardous levels due to anthropogenic activities—especially fossil fuel combustion, industrial emissions, and transportation [61,62]. Monitoring their presence and behavior is critical for assessing environmental and public health risks [1,2,63].

From a fuzzy control perspective, each pollutant represents a distinct input variable with asymmetric characteristics in time, space, and source behavior. For instance, CO₂ concentrations may gradually rise indoors due to poor ventilation, while NO₂ levels fluctuate rapidly in areas with heavy traffic [16,17]. This non-uniformity introduces a functional imbalance that must be explicitly addressed in the control system.

Carbon dioxide (CO₂): A dense greenhouse gas associated with cognitive impairment and respiratory effects [63–66]. Indoor levels often exhibit slow, accumulative asymmetry, requiring soft thresholds for responsive ventilation control [17,18]. Sulfur dioxide (SO₂): Emitted from industrial processes, it causes acute health effects and contributes to acid rain [67–70]. Emission patterns may create sharp, localized peaks, necessitating asymmetrical fuzzy responses [16,69]. Nitrogen dioxide (NO₂): A common byproduct of vehicle exhaust, with highly variable exposure profiles [71,72]. Effective control demands context-sensitive modeling due to its strong spatio-temporal asymmetry [16,17]. Therefore, the pollutants themselves constitute asymmetric variables, underscoring the need for adaptive, non-uniform fuzzy rule structures in any effective monitoring system.

2.2. Air Quality Index (AQI) & Standardization

The AQI aggregates pollutant concentrations into a unified scale that helps communicate health risks. Tables 1–3 summarize standard AQI thresholds for NO₂, SO₂, and CO₂ based on U.S. EPA guidelines [62,73,74]. However, while AQI thresholds provide standardized references, the way pollutant concentrations cross these thresholds is often asymmetrical—a slight increase in NO₂ may lead to disproportionate health effects in vulnerable populations [72]. This implies that fuzzy set modeling of pollutant severity should not be symmetric around a midpoint, but rather be biased or weighted according to real-world risk gradients. Hence, asymmetric fuzzy membership functions can better capture the nonlinear health-impact profiles of different pollutants, thereby improving the interpretability and responsiveness of the monitoring system [12–15,21].

Table 1. NO₂ Standards (USA).

Concentration (ppb)	AQI	Evaluation
0–0.054	0–50	Good
0.054–0.101	51–100	Moderate
0.101–0.361	101–150	Unhealthy for Sensitive Groups
0.361–0.65	151–200	Unhealthy
0.65–1.25	201–300	Very Unhealthy
1.25–1.65	301–Higher	Hazardous

Table 2. SO₂ Standards (USA).

Concentration (ppb)	AQI	Evaluation
0–36	0–50	Good
36–76	51–100	Moderate
76–186	101–150	Unhealthy for Sensitive Groups
186–304	151–200	Unhealthy

Table 3. CO₂ Standards.

Concentration (ppm)	AQI Air Level Indicator & Potential Health Problems
Indoor: <1000	Maximum safe level
Indoor: >1000	Poor air quality, potential health issues due to ventilation and circulation
Outdoor: 250–400	Normal air level
Outdoor: 400–1000	Typical air with good exchange
Outdoor: 1000–2000	Poor air, causing sleepiness
Outdoor: 2000–5000	Headaches, sleepiness, loss of attention
Outdoor: 5000	Toxic, possible oxygen deprivation
Outdoor: 40,000	Immediately harmful

Hence, asymmetric fuzzy membership functions can better capture nonlinear health impact profiles, directly linking AQI evaluation to the broader framework of symmetry/asymmetry in fuzzy control.

2.3. Applied Sensors

The system incorporates metal oxide semiconductor (MOS) sensors, which produce voltage outputs based on gas concentration:

- MQ-135—CO₂ detection [75]
- MG-811—NO₂ detection [76]
- 2SH12/MQ136—SO₂ detection [77]

These sensors provide analog readings that are subject to inherent noise and drift. Traditional binary or linear thresholding would fail under such uncertainty. Here, fuzzy logic enhances interpretation, and asymmetry in fuzzy preprocessing rules allows different sensor types to be evaluated based on their specific uncertainty profiles and sensitivities [11–15].

This variability represents a form of hardware-level asymmetry, which is explicitly modeled in the fuzzy rule base to ensure resilience and precision. This is also modeled in the fuzzy rule base to ensure resilience and precision.

2.4. Fuzzy Control Perspective in Air Quality Monitoring

Environmental systems are not only uncertain but fundamentally asymmetric in structure and dynamics. Fuzzy logic enables the transformation of imprecise sensor data into actionable classifications using linguistic terms (e.g., “low,” “moderate,” “high”) [11–17].

Key benefits include the following:

- Adaptive control of ventilation and alert systems under varying pollutant loads.
- Real-time handling of sudden spikes (e.g., NO₂ from vehicles).
- Multi-input integration for compound AQI assessment across pollutants with distinct temporal dynamics.

To manage this complexity, we leverage the following:

- Symmetric fuzzy rule sets for routine conditions—offering interpretability and balanced control across gas types.
- Asymmetric fuzzy rule structures for exceptional or high-risk scenarios—enabling the system to prioritize specific pollutants based on their risk profile or dynamic behavior (e.g., sharp SO₂ increases near industrial zones).

The dual application of symmetry and asymmetry within fuzzy control enables a resilient, context-sensitive assessment of air quality. Symmetric rules provide efficiency and balance under typical conditions, while asymmetric rules introduce flexibility for outliers and high-risk cases. This balance directly reflects the theme of symmetry/asymmetry in fuzzy control for complex, real-world monitoring systems.

2.5. Mathematical Formulation of the Backward Problem

Monitoring air quality using sensors mounted on UAVs can be viewed as a backward (inverse) problem. The forward problem involves understanding the physical spread of pollutants in the air, where a concentration field $c(x, t)$ evolves based on transport–diffusion dynamics in the following way:

$$\frac{\partial c}{\partial t} = D \nabla^2 c - u \cdot \nabla c + S(x, t), \quad (1)$$

where D is the diffusion coefficient, u is the wind velocity field, and $S(x, t)$ represents local emission sources.

The UAV-mounted sensors provide discrete, noisy measurements of pollutant concentration as follows:

$$y_i = A[c](x_i, t_i) + \varepsilon_i, \quad i = 1, \dots, N, \quad (2)$$

where $A[c]$ is the forward observation operator mapping the continuous field $c(x, t)$ to sensor outputs, and ε_i represents measurement noise and UAV-induced disturbances.

The inverse problem then consists of reconstructing an estimate $\hat{c}(x, t)$ or a scalar AQI from incomplete and uncertain sensor data $\{y_i\}$. This inverse problem is ill-posed in the traditional sense:

- Small perturbations in the input (ε_i) may lead to significant deviations in the reconstruction;
- Pollutant distributions are typically non-uniform and strongly influenced by asymmetric factors such as turbulence, altitude, or localized emissions.

2.6. Role of Symmetry and Asymmetry

Traditional models often assume a symmetric distribution of pollutants, i.e., smooth spatial or temporal variation around an equilibrium. This assumption corresponds to symmetric membership functions in fuzzy inference, in which deviations from thresholds are treated equally in both directions. However, real-world conditions are often dominated by asymmetry: localized peaks of SO₂ near industrial zones, sharp fluctuations of NO₂ from traffic, or gradual accumulation of CO₂ in poorly ventilated environments. Hence, symmetric fuzzy rules provide a baseline solution suitable for stable, quasi-equilibrium conditions, while asymmetric fuzzy rules act as a regularization mechanism, explicitly accounting for environmental disturbances. This duality not only stabilizes AQI estimation against noise and uncertainty but also operationalizes the conceptual framework of symmetry/asymmetry within fuzzy control.

2.7. Fuzzy Logic as Regularization of the Backward Problem

Instead of solving the backward problem through deterministic optimization, a fuzzy inference system (FIS) is introduced as a regularised mapping as follows:

$$A\hat{Q}I = F_{fuzzy}(\{y_i\}), \quad (3)$$

where F_{fuzzy} integrates both symmetric and asymmetric rule sets.

For symmetric rules, the logic is as follows:

$$\text{IF } y_i \text{ is "low" THEN AQI is "good"}, \quad (4)$$

capturing balanced dependencies and ensuring interpretability.

The asymmetric rules have the following logic:

$$\text{IF } y_i \text{ is "moderate" AND turbulence is "high" THEN AQI is "unhealthy"}, \quad (5)$$

shifting decision boundaries in response to asymmetric environmental conditions.

This formulation highlights how the proposed hybrid fuzzy controller converts noisy and incomplete inputs into consistent outputs, effectively regularizing the challenging backward problem of pollutant estimation.

The contribution of this work lies in bridging the following:

- Symmetry in mathematical modeling—symmetric fuzzy rules represent equilibrium and stability.
- Asymmetry as perturbation—asymmetric rules restore robustness under non-ideal conditions.
- Backward problem methodology—fuzzy inference provides a practical, real-time simulation method for solving inverse problems in environmental monitoring.

Recent studies have investigated the application of UAVs for collecting high-resolution environmental data. Low-cost sensing platforms enable dense spatial mapping, but their performance is affected by sensor nonlinearity and environmental variability [2–5,11–15]. Fuzzy logic has been applied to air quality assessment to manage uncertain and noisy sensor data [16–22]. However, only a few approaches explicitly formalize symmetry and asymmetry in fuzzy control as a stabilizing mechanism for ill-posed backward problems in pollutant estimation. The proposed framework builds upon these works by formalizing symmetric and asymmetric membership functions to enhance robustness in practical UAV deployments.

Compared with classical linear or threshold-based AQI estimation, the hybrid FIS generalizes across non-linear and asymmetric sensor responses, providing a mathematically stable reconstruction even when environmental conditions deviate from equilibrium. Symmetric rules capture baseline dynamics, while asymmetric rules correct for localized perturbations.

2.8. The UAV Control via Fuzzy Logic

The UAV's flight parameters, including altitude, speed, and trajectory adjustments, are governed by a fuzzy inference system (FIS) that interprets the measured pollutant concentrations. Symmetric rules handle stable baseline environmental conditions, while asymmetric rules react to localized anomalies, such as sudden increases in pollutant levels or sensor noise.

For instance, if CO₂ concentration rises rapidly, the FIS can command the UAV to ascend or modify its path to map pollutant distribution more effectively. Each fuzzy set is defined numerically using triangular or trapezoidal membership functions, and

representative rules are summarized in Section 5.4. Weighted firing strengths are applied based on the degree of asymmetry, ensuring that local anomalies trigger appropriate UAV control actions.

This approach allows the UAV to adapt dynamically to environmental variations, maintain measurement accuracy, and safely navigate through areas with heterogeneous pollutant distributions, without requiring visual illustrations for clarity.

2.9. FIS Output and Symmetry/Asymmetry Calculation

The hybrid FIS computes UAV control actions and air quality estimations using the degree of membership of pollutant concentrations and the symmetry index.

2.9.1. Fuzzy Rule Evaluation

Each fuzzy rule is expressed as:

$$R_i: \text{IF } (X_1 \text{ is } A_1) \text{ AND } \dots \text{ AND } (X_n \text{ is } A_n) \text{ THEN } Y \text{ is } B \quad (6)$$

where X_j is the input variables representing pollutant concentrations (e.g., NO_2 , SO_2 , CO_2 , measured in ppm or ppb); A_j is the fuzzy sets describing linguistic terms such as *Low*, *Medium*, or *High*; Y is the output linguistic variable, representing UAV control actions (e.g., altitude adjustment, path correction); and B is the output fuzzy set.

The firing strength of each rule is computed via the min operator for AND operations:

$$\alpha_i = \min\{\mu_{A_1}(X_1), \mu_{A_2}(X_2), \dots, \mu_{A_n}(X_n)\} \quad (7)$$

where μ_{A_i} is the degree of membership of the input value X_j in the fuzzy set A_j .

The *max* operator then obtains the aggregated output membership function:

$$\mu_Y(y) = \max_i[\alpha_i \cdot \mu_{B_i}(y)]. \quad (8)$$

Finally, the crisp output value (e.g., UAV altitude change in meters or throttle correction in %) is computed using the centroid defuzzification method:

$$Y^* = \frac{\int y \cdot \mu_Y(y) dy}{\int \mu_Y(y) dy}. \quad (9)$$

This process ensures smooth control transitions and avoids abrupt UAV maneuvers under fluctuating air quality conditions.

2.9.2. Symmetry/Asymmetry Index Calculation

To account for environmental anomalies (e.g., local emissions or sensor disturbances), a symmetry index S is defined for each input variable:

$$S = \frac{|X - \mu_X|}{\sigma_X}, \quad (10)$$

where X is the current pollutant concentration, μ_X is the mean of historical concentrations, and σ_X is the standard deviation of historical data.

This index quantifies deviation from baseline conditions:

- $S \leq 0.5 \rightarrow$ Symmetric behavior (*normal background*),
- $S > 0.5 \rightarrow$ Asymmetric behavior (*localized anomaly*).

To enhance system responsiveness, the degree of asymmetry D_a is computed as:

$$D_a = \max(0, S - 0.5), \quad (11)$$

and is used to weight the rule activation strength:

$$\alpha_i^{weighted} = \alpha_i \cdot (1 + D_a). \quad (12)$$

This adaptive weighting increases the influence of asymmetric rules when anomalies are detected, ensuring the UAV reacts promptly to abnormal pollutant spikes or sensor noise.

2.9.3. AQI Calculation

For each pollutant p , the Air Quality Index (AQI) is determined using the standard piecewise linear interpolation.

$$AQI_p = \frac{I_{Hi} - I_{Lo}}{C_{Hi} - C_{Lo}} \cdot (C_p - C_{Lo}) + I_{Lo}, \quad (13)$$

where C_p is the observed concentration of pollutant p , C_{Lo} , C_{Hi} are lower and upper breakpoint concentrations for p , and I_{Lo} , I_{Hi} are corresponding AQI index values (e.g., 0–50 for *Good*, 51–100 for *Moderate*).

The overall air quality index is defined as the maximum among pollutants:

$$AQI_{overall} = \max(AQI_{NO_2}, AQI_{SO_2}, AQI_{CO_2}) \quad (14)$$

This ensures that the dominant pollutant determines the final air quality warning for the area under observation, in line with international AQI standards.

The combined use of fuzzy rule evaluation, symmetry-weighted adaptation, and AQI computation allows the UAV to perform real-time, context-aware air quality mapping with enhanced robustness to disturbances.

While the symmetry index addresses dynamic variations in environmental conditions, an additional entropy-based weighting is introduced to account for the reliability of sensor data.

2.10. The Entropy Method

To complement the symmetry-based adaptation, the entropy method quantifies the uncertainty of sensor measurements. The entropy method is employed to quantify the uncertainty of the sensor measurements, providing a numerical basis for weighting each sensor's contribution to the fuzzy inference system (FIS). For each pollutant j , the entropy is calculated as:

$$E_j = -k \sum_{i=1}^n p_{ij} \ln p_{ij}, \quad k = \frac{1}{\ln n}, \quad (15)$$

where p_{ij} is the normalized value of the i -th observation for pollutant j , calculated as follows:

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}}, \quad (16)$$

n is the total number of observations.

The normalization factor k ensures that $E_j \in [0, 1]$, making entropy values comparable across datasets of different sizes. Higher entropy values indicate more uncertain or evenly distributed sensor measurements, while lower values indicate more informative and consistent readings.

To incorporate these values into the fuzzy inference system, the entropy values are transformed into weights:

$$w_j = 1 - E_j, \quad (17)$$

where w_j is the weight assigned to the j -th sensor input. Sensors with higher information content (lower entropy) are given higher weights, allowing the FIS to prioritize reliable measurements during pollutant reconstruction. This entropy-based weighting improves the overall accuracy of the UAV-based air quality monitoring system without requiring a schematic illustration.

3. The System Architecture & Operation Principle

The proposed system comprises both hardware and software units, which integrate into a unified framework for real-time air quality monitoring. The hardware unit includes two major components: (i) the sensor module, which measures gas concentrations (NO_2 , CO_2 , SO_2), as well as temperature and humidity; and (ii) the UAV platform, which enables mobile deployment and variable-altitude sampling. The software unit features a fuzzy-logic engine that combines symmetric and asymmetric rule sets to calculate the Air Quality Index (AQI). Symmetric rules capture balanced, steady-state conditions (e.g., stable pollutant levels), while asymmetric rules adaptively respond to deviations caused by turbulence, localized emissions, or other disturbances. This integration ensures robustness of the monitoring process in both equilibrium and perturbed environments. Additionally, a local storage and communication module ensures data reliability even during temporary connectivity loss.

Figure 1 illustrates the operational flow of the developed air quality monitoring system, designed for both stationary and mobile configurations. In the mobile configuration, the monitoring module is mounted on a UAV, allowing for remote sensing across various altitudes and areas. In stationary mode, the same process can be executed locally, excluding the UAV component but retaining full sensing and fuzzy processing capabilities.

The operational cycle begins with the definition of a target inspection area. Once the UAV reaches the designated location, the data acquisition loop is initiated and runs continuously throughout the monitoring session. Individual measurements from the NO_2 , CO_2 , and SO_2 sensors are first collected. These sensors generate analog voltage signals proportional to gas concentrations. The signals are processed by a microcontroller (Arduino) on the drone's circuit board, which converts them into standardized values in parts per million (ppm) or parts per billion (ppb).

The microcontroller simultaneously coordinates three subsystems: (i) data acquisition, (ii) wireless transmission, and (iii) data display. On the UAV side, measurements are stored on an SD card to prevent loss in the event of communication failure. Gas concentration values are also displayed on an integrated LCD screen. In parallel, data is wirelessly transmitted to the user-side circuit, where a second microcontroller evaluates the incoming data against established thresholds and displays the results on a TFT touchscreen interface.

An additional feature allows the user to activate an air pump remotely via the interface. This pump collects a physical air sample from the monitored zone for laboratory analysis. Once the sample is obtained, the pump automatically deactivates, and the data-acquisition loop concludes, marking the end of the monitoring session.

Throughout the entire operation, the system ensures direct user involvement—from defining the inspection location to initiating measurements and managing sample collection. Upon completion, the UAV autonomously returns to the user, allowing redeployment in another target area. This iterative workflow enables flexible, user-guided monitoring with real-time visualization and archival storage, fostering a more comprehensive and context-sensitive understanding of local air quality conditions.

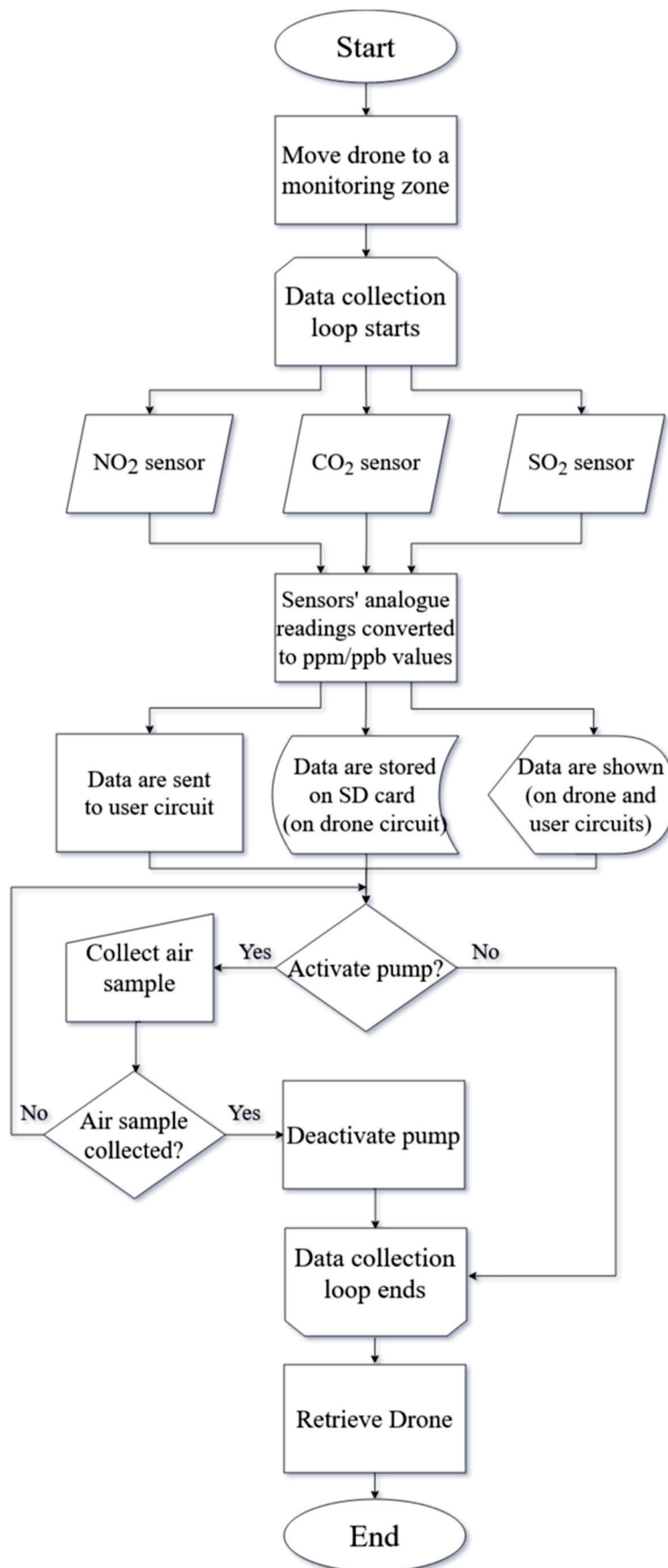


Figure 1. Operation schematic of the proposed system.

4. Fuzzy Logic Model & System Design

The proposed system design provides a platform for real-time air quality monitoring under varying environmental conditions. It is explicitly designed to address real-world asymmetries arising from heterogeneous sensor behavior, fluctuating gas concentrations, and dynamic operational settings. These asymmetric characteristics create uncertainties and nonlinearities in the data, which justify the application of fuzzy control strategies in subsequent processing stages.

4.1. Fuzzy Logic Concept

The system provides a physical platform for acquiring air quality data in environments characterized by asymmetry and uncertainty. Accordingly, a fuzzy interpretation of asymmetric sensor data is essential to support the application of asymmetric fuzzy rules. The heterogeneous nature of the chosen sensors—each employing different detection methods, ranges, and sensitivities—leads to inconsistent and occasionally contradictory readings. Moreover, fluctuations in altitude, temperature, and wind patterns during UAV operation introduce another layer of dynamic inconsistency to the monitoring process.

To manage these conditions, a methodology is required that can handle ambiguity, partial truths, and nonlinearity. Fuzzy logic, incorporating both symmetric and asymmetric rules, is employed as a natural extension of the monitoring framework. Its ability to interpret imprecise and uncertain inputs makes it particularly effective for air pollution data. Thus, fuzzy inference systems are applied to derive meaningful assessments from raw sensor values, improving the decision-making capabilities of the monitoring solution.

The fuzzy inference system is structured around two categories of rules:

1. Symmetric rules: *IF [sensor condition A] THEN [AQI state B]*, reflecting stable and regular dependencies in data.
2. Asymmetric rules: *IF [sensor condition A] AND [disturbance C] THEN [modified AQI state B']*, capturing environmental asymmetries and exceptions.

By combining these two classes, the system effectively models both equilibrium (symmetry-preserving) and perturbed (symmetry-breaking) conditions, leading to more reliable AQI estimation.

4.2. Design of Hardware Unit—Drone Side Circuit

This module examines the following aspects:

- Selection of a UAV platform (commercially available).
- Creation of the electrical circuit housed in a unit attached beneath the chosen UAV.

A medium-sized UAV was selected after evaluating options based on size, stability, payload, and compatibility. This drone enables remote monitoring of emissions and is particularly beneficial for inspecting hazardous locations. Ensuring stable flight control, reliable data logging, and efficient wireless transmission were essential requirements. Numerous studies have examined the advantages and drawbacks of UAVs, focusing on factors such as cost, weight, speed, flight time, capacity, and safety [78–83]. After evaluation, the DJI Phantom was selected as the monitoring platform (Figure 2).

The cost of the UAV is about USD 700, with the following specifications: weight 1.21 kg (0.84 kg excluding the camera), maximum ascent 5 m/s, maximum descent 3 m/s, maximum speed 16 m/s, flight duration ~25 min, hover accuracy ± 1.5 m (horizontal) and ± 0.5 m (vertical), diagonal size 350 mm without propellers, battery 4480 mAh Li-Po 4S, operating temperature 0–40 °C, maximum altitude 6000 m, reliable and safe operation [84]. To install the monitoring circuit, the onboard camera was removed.



Figure 2. Visual presentation of the “DJI Phantom” drone: (a) with a camera, (b) with the camera removed for the drone’s side circuit needs.

The drone-side circuit (Figures 3–5) includes the following:

- Sensors: MQ135 (CO_2), MG-811 (NO_x), 2SH12 (SO_2).
- Two Arduino Uno boards (control and data conversion).
- MicroSD module (storage).
- NRF24L01 module (wireless communication).
- Air pump (sample collection).
- Nokia 5110 LCD (onboard display).
- Additional components (resistors, capacitors, regulators, power supplies, motor driver).

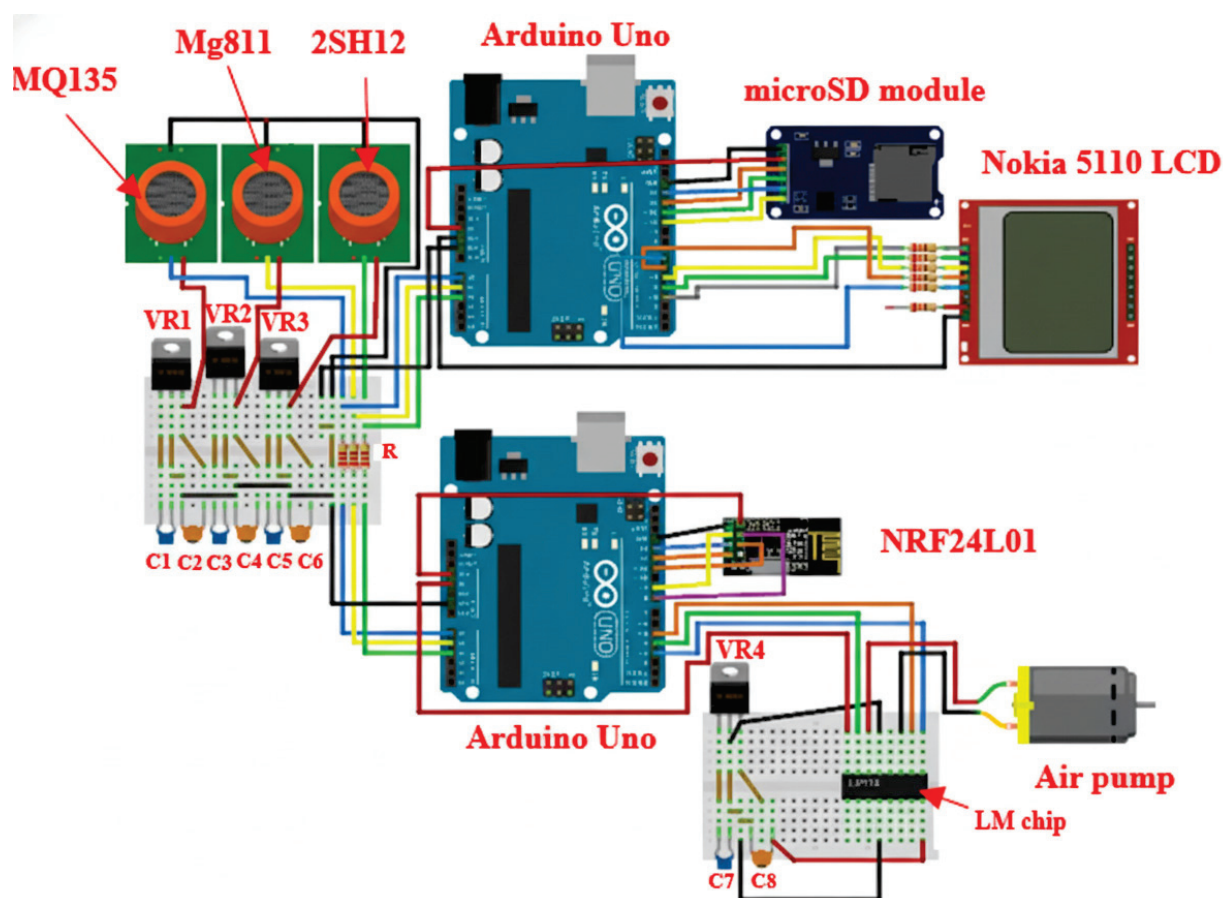


Figure 3. Layout of the drone side circuit.

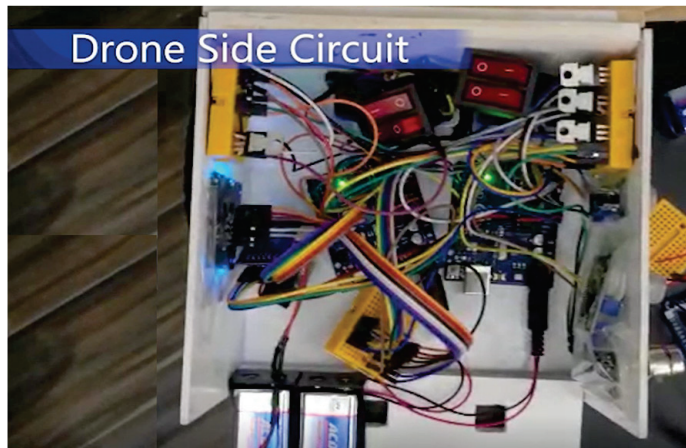


Figure 4. Implementation of the drone side circuit.

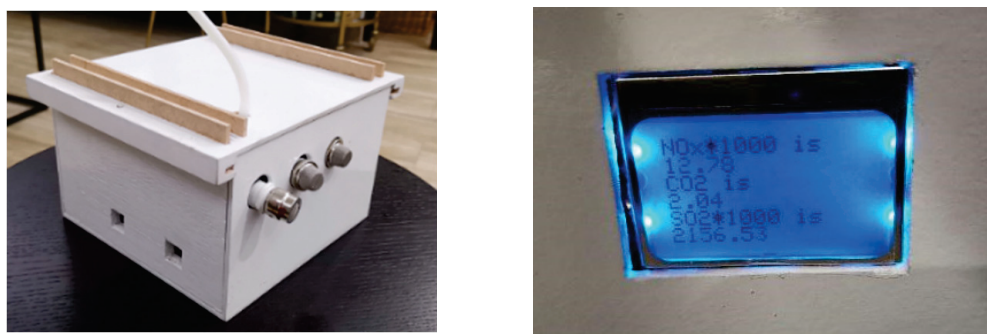


Figure 5. Design of the housing unit on the drone side circuit.

The upper Arduino processes sensor data, converts voltages to parts per million (ppm) or parts per billion (ppb), and outputs the data to both the LCD and SD storage. The lower Arduino handles transmission and controls the air pump. The circuit is housed in a lightweight wooden unit (total weight with components: 1.0 kg). Removing the DJI camera compensated for payload limitations.

This UAV model provides the required mechanical stability and payload capacity, enabling the airborne unit to operate effectively under asymmetric environmental conditions, such as fluctuating wind, temperature, and pressure. These variations affect sensor behavior, underscoring the need for robust fuzzy-based interpretation.

4.3. Design of Hardware Unit—User Side Circuit

The user-side circuit tracks transmitted sensor data and manages the air pump for remote sampling. It includes the following:

- Arduino Mega (central controller).
- NRF24L01 module (wireless communication).
- 2.4" TFT touchscreen (interface, pump activation, and results display).
- Power sources.

The design (Figures 6–8) features a housing with a touchscreen interface for user interaction and accessible troubleshooting. A lightweight wooden casing was chosen for safety and durability.

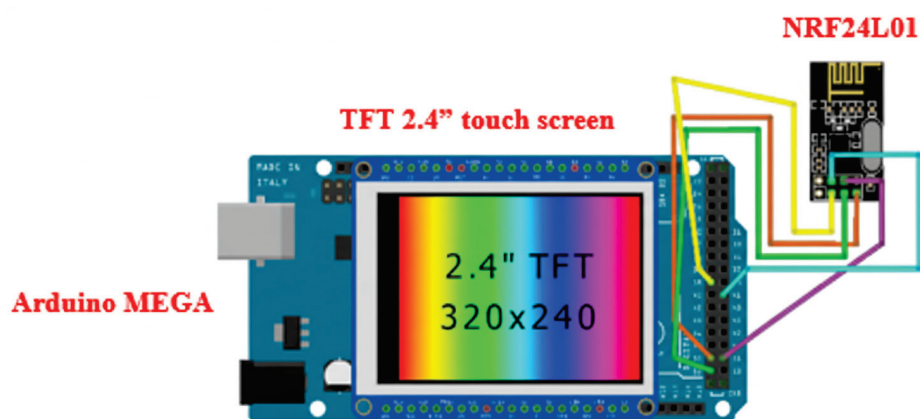


Figure 6. Layout of the user-side circuit.

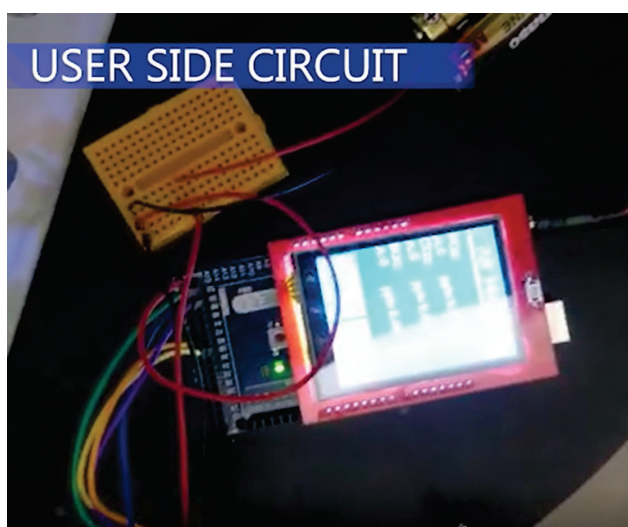


Figure 7. Implementation of the user-side circuit.



Figure 8. Design of the housing unit on the user side circuit.

By design, the UAV-mounted sensors (CO_2 , NO_x , SO_2) exhibit different sensitivities and calibration requirements, introducing asymmetric uncertainties. These cannot be managed effectively by threshold-based methods, further motivating the integration of fuzzy control to interpret nonlinear responses.

4.4. Design of Software Unit

The software unit manages programming, testing, and system operation. It enables hardware operation, data processing, and storage.

The Arduino IDE [85] is used for programming and monitoring microcontrollers. Serial ports allow real-time observation of microcontroller behavior, while the serial plot function visualizes initial sensor data (CO_2 , SO_2 , NO_2).

Data is logged to the microSD and stored in Microsoft Excel [86], which provides easy readability and compatibility with analysis tools. Excel enables further manipulation of ppm/ppb values and integration with other programs.

Future development could integrate advanced GUIs and dedicated fuzzy inference engines for real-time interpretation. This would enable robust handling of sensor uncertainties resulting from environmental variability.

4.5. Design Cost

The prototype design emphasizes the monitoring system rather than the UAV itself. The estimated system cost is approximately USD 350, making it affordable for individuals, with options for both stationary and UAV-mounted use. A medium-sized UAV costs approximately USD 1400, while rental is USD 550–700, including housing and testing. Future efforts may focus on integrating low-cost UAVs into the monitoring system or on applications in professional sectors, such as agriculture or energy facilities, though these would require larger investments.

In asymmetric monitoring environments (e.g., industrial zones), fuzzy logic significantly improves interpretation accuracy. This approach remains cost-effective compared to traditional calibrated systems, making it practical for small-scale and uncertain real-world scenarios.

5. Field Experiments

The testing phase was crucial for evaluating how the system performs in realistic environments that inherently exhibit asymmetry—in sensor response behavior, the spatial distribution of gases, environmental conditions (e.g., wind turbulence), and technical constraints. These irregularities and uncertainties complicate deterministic analysis and motivate the integration of fuzzy logic in the future to achieve a robust interpretation of results.

Field trials were conducted in diverse environments, including urban and desert areas. The UAV-enabled system successfully measured pollutant levels while compensating for turbulence, altitude variations, and irregular pollutant distributions. Key observations include the following:

- (1) Symmetric rules alone provided stable results under controlled conditions.
- (2) Asymmetric rules significantly improved accuracy when environmental symmetry was disrupted (e.g., during sudden wind gusts).
- (3) The hybrid approach maintained reliable AQI calculations even under noisy sensor conditions.

A detailed discussion of the system implementation and testing results is provided below.

5.1. Testing Hardware Unit

The hardware unit comprises both the user-side circuit and drone-side circuit designs (Section 4). Testing of the system's operation focuses on a drone flight and the performance of its sensors.

The layout and weight distribution of components significantly impact the drone's flight stability, particularly under asymmetric external conditions, such as gusty winds or uneven loading. These uncontrolled factors introduce real-world variability that cannot be modeled deterministically, which supports the argument for incorporating fuzzy control logic to interpret system behavior in such dynamic settings.

Once the drone's side circuit is attached to the drone's housing unit, flight stability is tested. This is a crucial consideration for the system's safe and reliable operation. The total

weight is around 0.6 kg, as mentioned before. The average lifting capacity of medium-sized drones ranges from 1.2 kg to 1.5 kg. Any extra load will cause many unexpected and unintended flight abnormalities. Thus, the selection of the housing unit's shape and the distribution of its internal weight are important considerations. For instance, the unit's shape must withstand high-speed wind impacts (minimizing risks) that could compromise the drone's control and operation. The layout of electric circuits within housing units must be optimized to maintain the drone's balance, but high-speed winds may still affect its stability. At the same time, the drone (existing end-product) has been integrated into the proposed system, which provides additional design advantages, particularly by accounting for aerodynamic issues specific to the drone (an in-built feature that reduces the adverse effects of high-speed winds). The high-speed wind, especially in desert areas, caused unpredictable deviations in drone balance. These unpredictable aerodynamic disturbances represent a real-world asymmetry that underscores the need for adaptable, soft-computing methods in future implementations.

Testing drone flights has encountered challenges. Attempts at flight, particularly when the housing unit is attached to the drone (which houses the circuit), have proven challenging. The added weight has increased the drone's power consumption, with power usage rising by approximately 65–70% during takeoff compared with standard flight conditions without any added load, where the drone typically consumes about 40% of its energy. The highest altitude reached during the flight was roughly 20 m. Technically, it could have achieved greater heights and the majority of flight testing has been conducted in isolated locations at lower altitudes. The typical flight duration has been about 20 min (up to 25 min, as specified by the drone), but it is shorter in urban areas. The minimum operational time for the drone's side circuit was approximately 2 h, assuming the batteries were fully charged initially. For the user-side circuit, the operational time should be slightly longer, given its components' lower energy use. Thus, the operational duration of the circuit is influenced by the frequency of system use and the energy consumption and distribution of its components. To gather more flight data, testing was carried out in a desert environment (with specific conditions). However, wind became a critical factor during these tests, as it was significantly stronger than in urban settings, highlighting issues with the housing unit's construction. An example of the drone's flight during testing is shown in Figure 9.



Figure 9. Testing of the drone's flight and circuitry.

In summary, the drone struggles to maintain balance in high winds. Due to the absence of a specialized instrument, accurate wind speed measurements could not be recorded. Nevertheless, based on experiential testing, it can be stated that in urban areas, wind speeds ranged from levels 2 to 4 (1.6–7.9 m/s), and in the desert, from levels 5 to 7 (8–17.1 m/s) according to the Beaufort wind scale [87]. It is important to note that the landings and takeoffs were reasonably stable with minimal movement across the horizontal axis (in low-speed winds).

The sensor response varied significantly with gas concentration and type. For example, low concentrations of NO₂ and SO₂ were below detection thresholds, whereas high concentrations produced reliable spikes. This non-linear, threshold-sensitive behavior results in non-symmetric sensor performance. Thus, binary or linear interpretations fail to capture nuances in gas exposure—a gap that fuzzy modeling can address by assigning degrees of presence rather than fixed thresholds.

The sensor's effectiveness is evaluated based on the collection of precise and trustworthy data. To ensure data accuracy, it is essential to establish a correlation between measured gas concentrations and the Air Quality Index (AQI) standards. The analog readings of the detected gases responded predictably: a spike in gas release led to a swift rise in the values upon contact. However, the initial transition from analog values to ppm/ppb values encountered challenges. For example, NO₂ and SO₂ readings were approximately 40% lower than anticipated, while CO₂ readings were 20% higher than expected. To address these issues, the sensors were recalibrated using a device known as "ToxiRAE" [88], along with facilities from power plants that provided canisters and compartments with fixed gas concentrations, and applying specific mathematical formulas. This recalibration process required time due to the identified discrepancies. The following observations were made: in instances of high gas concentration levels, such as abnormal spikes or significant leaks, the chemiresistor effectively met the monitoring requirements, suggesting its suitability for detecting analog values. Nevertheless, the chemiresistors do not demonstrate sufficient reliability for precise ppm/ppb measurements. The sensors were configured for testing within a designated ppm/ppb range. For instance, the NO₂ ppb readings fluctuated between 1 and 10, confirming that the chemiresistors were operating adequately. A significant limitation of the modified sensors is their design constraints, which prevent them from measuring low concentrations of NO₂ and SO₂. After conducting several tests, the concentrations of NO₂ and SO₂ were insufficient to elicit a proper analog response from the sensors, resulting in 0% accuracy due to the absence of a detectable signal. Table 4 summarizes the results of hardware design testing. For more details, refer to the explanation part.

5.2. Testing Software Unit

A performance evaluation of the software unit has been conducted based on its capabilities to capture readings, interpret data, and present results.

The following indicators—speed, transmission capability, and ease of use—have been selected for data-acquisition testing. To conclude, under a short range between both circuits of up to 5 m, the data transmission time could be considered as satisfied, where the time delay does not exceed approximately 10 ms on average. However, as this range increases, significant delays occur due to the NRF24L01 module's limited transmission range. For instance, the wireless radio frequency band has a short range (a limited wavelength), and there is potential interference from nearby objects due to the limited range between the circuits while monitoring pointed spots. Unfortunately, the exact data could not be captured physically when the drone's side circuit is located in the air. In practice, a range of 20–50 m is recommended, but it can reach up to 200 m.

Wireless transmission is also affected by asymmetric physical obstructions, dynamic distances, and environmental interference. These factors make communication stability non-uniform, varying from test to test. A fuzzy-based communication assessment model could be beneficial for future upgrades to dynamically evaluate signal quality, rather than relying on static thresholding.

Table 4. Summary of the Hardware Design Testing.

Indicator	Obtained Results and Brief Discussion
Drone flight	
Shape and weight of the housing unit (drone side)	The shape is appropriate for the drone being used. The weight does not exceed the maximum allowed (approximately 0.6 kg), but it could be reduced further.
Layout of the components	Balanced.
Power	A drone consumes more power during takeoff and landing. An additional power source could be added, or a powerful drone may be required. The circuit's power consumption varies depending on component type and usage, as well as load distribution. The layout must be optimized and balanced in terms of load demand.
Impact from a high-speed wind	The drone performs satisfactorily during landing and takeoff, but it experiences flight issues in high-speed winds, losing its balance. Redesign the housing unit or use a powerful drone.
Flight time	Around 20 min on average (by drone specification, maximum 25 min), and less for urban areas.
Circuit operation time	It ranges from 30 min to 2 h.
Legal restriction	The drone camera has been removed. Testing primarily took place in isolated locations and at low altitudes (up to 20 m), resulting in relatively short flights.
<i>Sensor's performance</i>	
The captured analog values.	Responsive as predicted, with the exception that it is not responsive when the gas concentration is lower or higher than what the sensors allow.
The AQI standard ppm/ppb	Additional calibration is required.
Accuracy and reliability	Chemiresistors are accurate when excessive values are defined, but are not reliable enough for the high accuracy required in real-time measurements at ppm/ppb levels. The advanced sensors must be used in this case.

The drone features a data logging option that utilizes a removable SD card, located on the side of the device. The SD card currently has a storage capacity of 64 GB. Nevertheless, any standard-sized SD card can be used, as the system is compatible with standard-sized cards. Due to specific hardware and software constraints, readings are recorded approximately once every second, although enhancements could enable faster recording rates. The micro-SD module reads and writes data to the SD card, ensuring continuous accessibility and providing a dependable storage alternative. However, the drone's maximum flight duration is limited, which may require recharging the drone, during which data logging is not possible.

The analog sensor data, when logged and visualized, revealed non-linear response curves and inconsistent behavior across conditions. Traditional deterministic processing requires multiple case-specific models. In contrast, fuzzy control strategies can generalize across a broader range of inputs, accommodating these uncertainties with less complexity.

If required, the air sample could be collected in an inspected area. The user must activate the air pump on the drone's side circuit to initiate the collection process. The retrieval of the air samples has been easy during the testing of this data acquisition method.

To facilitate usability, each circuit side includes a display to track NO₂, SO₂, and CO₂ emissions. The LCD screen on the drone's circuit provides this feature during stationary monitoring (which does not require the drone to be in operation) or when the drone is on the ground (Figure 5). On the user side, the TFT screen displays the emission levels, continuously updating the information (Figure 8). There are two primary buttons: "Activate Air Pump" and "Air Quality Info." Pressing the first button activates the pump to collect air samples, while the second button displays air quality details based on the measured gas readings.

Next, let us review and discuss the real-world testing results of captured readings of the examined gases, along with their impact on human health.

5.3. Real-Life Testing Results

The real-life testing further revealed spatial asymmetry in gas distribution, with air pollutant concentrations varying not only between indoor and outdoor locations but also with altitude and environmental factors (e.g., cooking). The disparity in readings, especially for CO₂, ranged by over 45 times across the vertical range (e.g., 464 ppm vs. 21,075 ppm), underscoring the need for a control model capable of handling heterogeneous, spatially skewed data—such as fuzzy inference systems.

The following monitoring spots have been considered:

- Local (stationary) indoor—bedroom (Figure 10) [89];
- Local (stationary) indoor—kitchen while cooking (Figure 11);
- Local (stationary) outdoor—urban area (Figure 12);
- Remote outdoor—urban area, when the drone is located at 5 m altitude from the ground (Figure 13) [89].

The results are presented below; each figure shows readings in both analog and ppm (CO₂) or ppb (NO₂, SO₂) for the monitoring spots mentioned above.

The captured NO₂ and SO₂ readings have been used to evaluate the AQI at the inspected sites. The AQI online calculator has been applied for this purpose [90] in addition to the standardization scale given in Tables 1 and 2. However, CO₂ gas readings have been evaluated against the standard scale shown in Table 3 [62].

Based on NO₂ and SO₂ gas readings, the following has been observed (detailed data given in Table 5) [89]:

- For indoor monitoring spots such as the bedroom and kitchen, the bedroom has the highest SO₂ ppb value, leading to the highest AQI—35 as compared to the kitchen monitoring results—32 (almost the same range), but both present good AQI levels with respect to the standard values given in Table 2. The NO₂ readings are low, indicating a good AQI level per the standard levels in Table 1.
- For outdoor monitoring spots, such as urban areas at 5 m altitude, we can observe the highest SO₂ ppb value, resulting in an AQI of 46 for metropolitan areas, compared to spots at 5 m altitude, where the AQI is only 4. The NO₂ ppm value is highest in the urban area at 5 m altitude (AQI is 30), and in the metropolitan area, AQI is only 10, but both indicate good AQI levels.
- Analyzing all readings, the highest SO₂ ppb value is observed in the urban area (AQI is 46), while the lowest is in the urban area at 5 m altitude (AQI is 4). The highest NO₂ ppb value is detected for the urban area at 5 m altitude (AQI is 30), and the lowest NO₂ ppb value is defined for the metropolitan area with an AQI of 10.

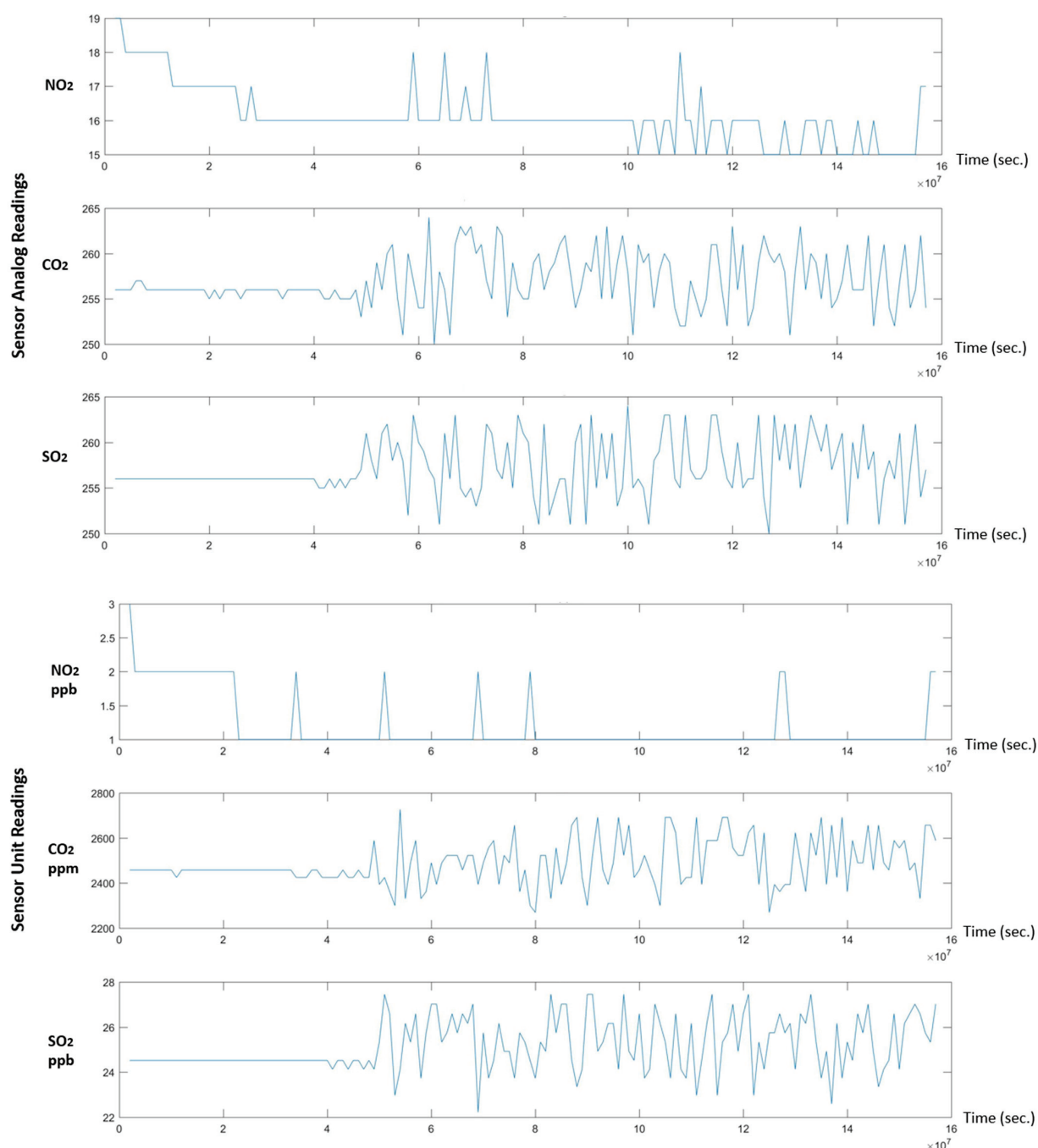


Figure 10. The emission readings in a bedroom.

Despite good AQI results, some people groups might still be at risk of potential health issues.

Table 5 summarizes the average NO₂ and SO₂ concentrations obtained at various monitoring spots, along with the corresponding Air Quality Index (AQI) evaluation. Indoor measurements indicate that the bedroom shows low NO₂ levels (0.01196 ppb, AQI = 11) and moderate SO₂ levels (24.862 ppb, AQI = 35), indicating good air quality for the general population. However, sensitive groups such as people with asthma or other respiratory diseases, the elderly, and children are more at risk. In the kitchen, higher NO₂ concentrations (5.5684 ppb) were recorded, reflecting indoor cooking emissions. Urban outdoor measurements indicate low NO₂ at ground level (0.0106 ppb, AQI = 10), while SO₂ reached 33.2801 ppb (AQI = 46), demonstrating moderate air quality with potential

risk for vulnerable groups. Measurements taken at 5 m altitude show slightly elevated NO_2 (0.03218 ppb, AQI = 30) but very low SO_2 (3.1375 ppb, AQI = 4), confirming that air pollutant concentrations decrease with altitude. These results highlight the importance of monitoring both indoor and outdoor locations, as well as vertical profiling, to capture spatial variability in air quality.

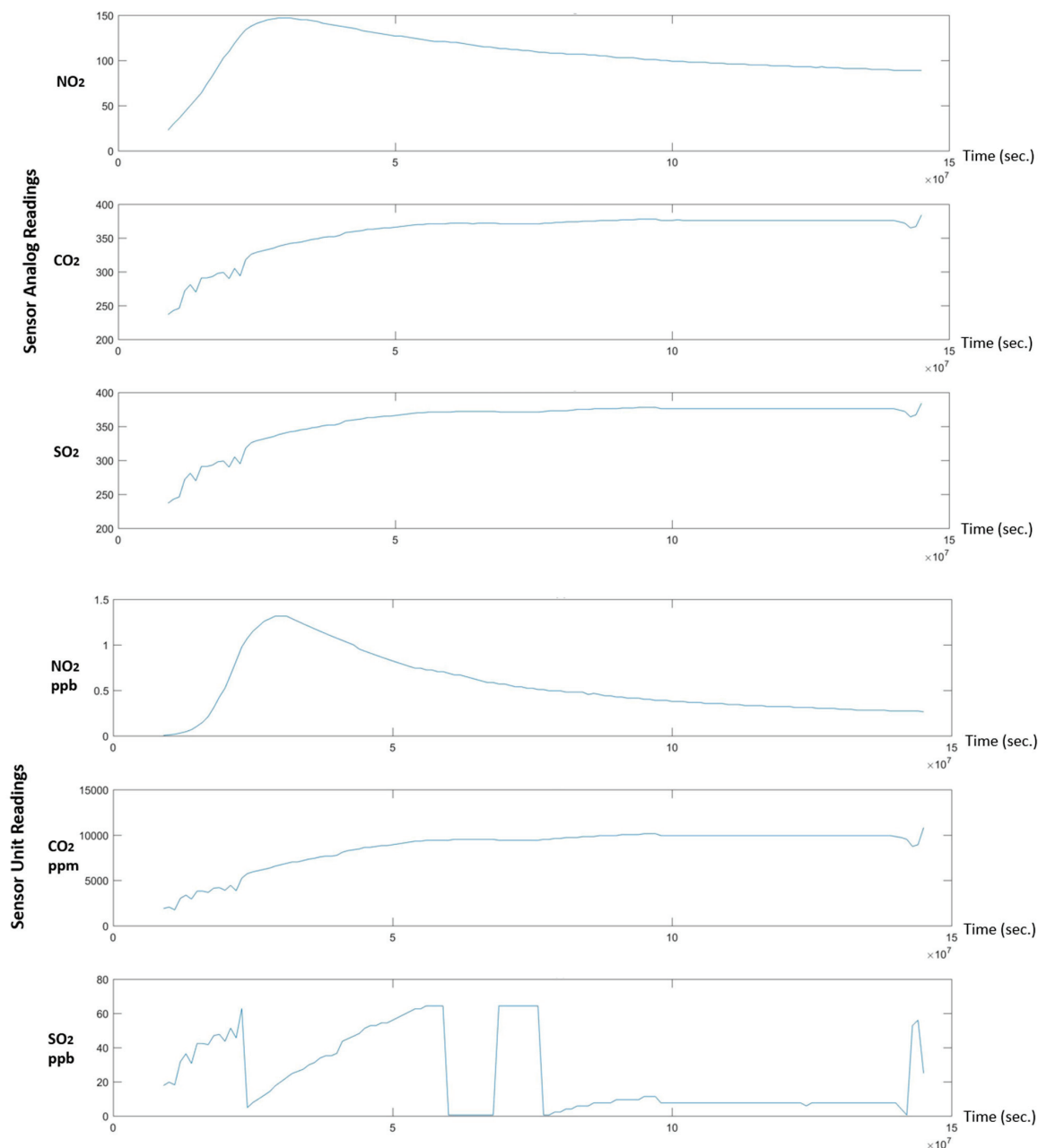


Figure 11. The emission readings in a kitchen while cooking.

Based on the CO_2 gas readings, the following has been obtained (detailed data given in Table 6) [89]:

- For indoor monitoring spots, the highest CO_2 ppm value is observed in the kitchen, due to the high gas concentration that occurs when cooking.
- For outdoor monitoring spots, the highest gas concentration detected in the urban area indicates inferior air quality at ground level, due to factors such as intensive traffic (fumes), high population density, etc., according to the air quality scale shown in Table 3.

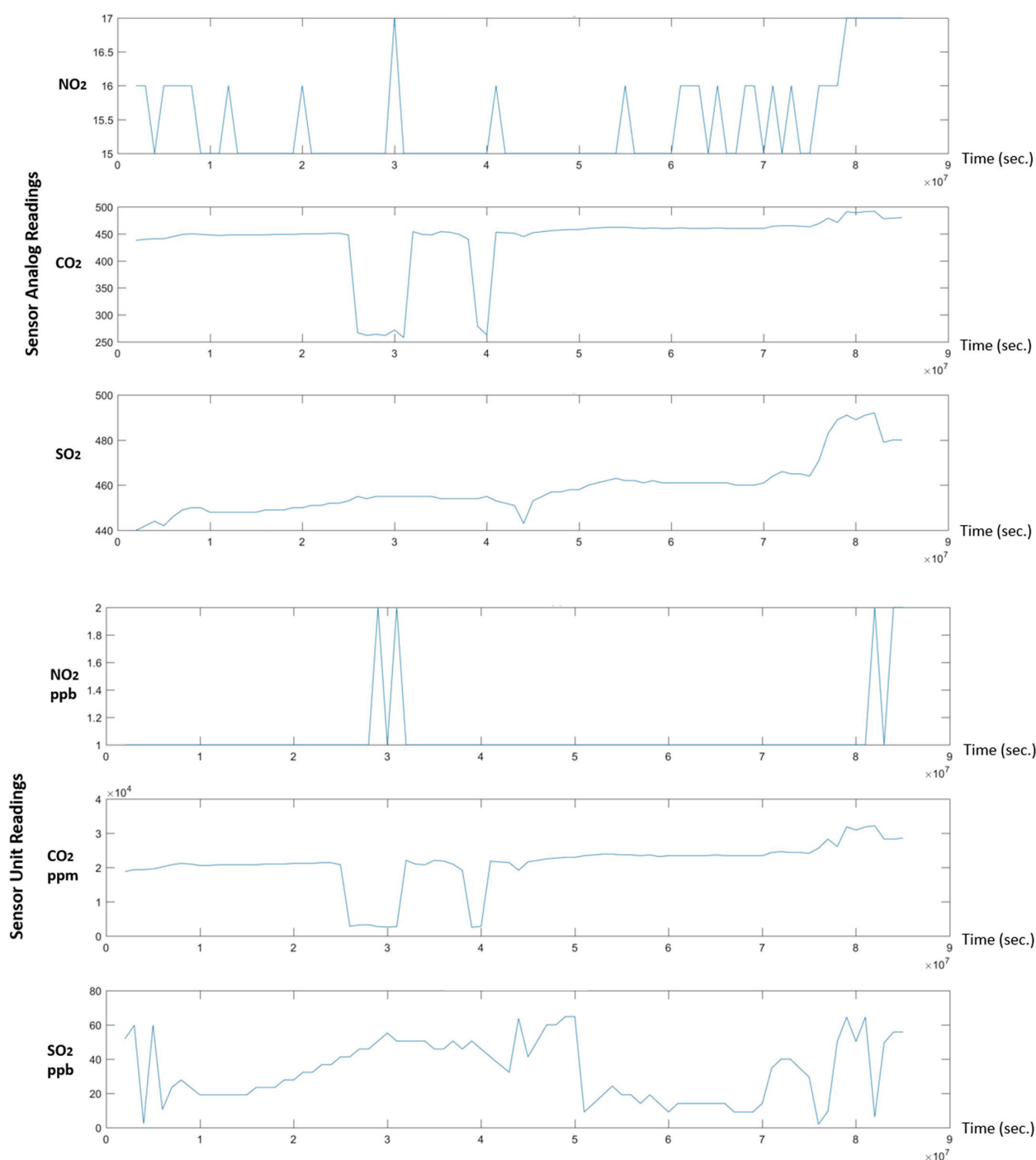


Figure 12. The emission readings in an urban area.

Based on the testing results for the hardware and software units, several problems have been identified. To minimize the appearance of these concerns and extend the prototype's operational conditions (including the implementation of future considerations), specific improvements can be made.

Table 6 presents the average CO₂ concentrations recorded in indoor and outdoor locations. The highest indoor CO₂ concentration was observed in the kitchen (8693.206 ppm), reflecting cooking-related emissions and limited ventilation. In contrast, the bedroom showed 2461.475 ppm, indicating relatively better air circulation. Outdoor measurements indicate that the urban area at ground level reached 21,075.4 ppm, exceeding the standard limit of 5000 ppm and likely due to traffic, high popula-

tion density, and other anthropogenic sources. Measurements at 5 m altitude decreased sharply to 464.491 ppm, demonstrating the vertical dilution of pollutants. These findings indicate that both the location and height of UAV-based measurements critically affect air quality assessments and underscore the need for adaptive monitoring strategies.

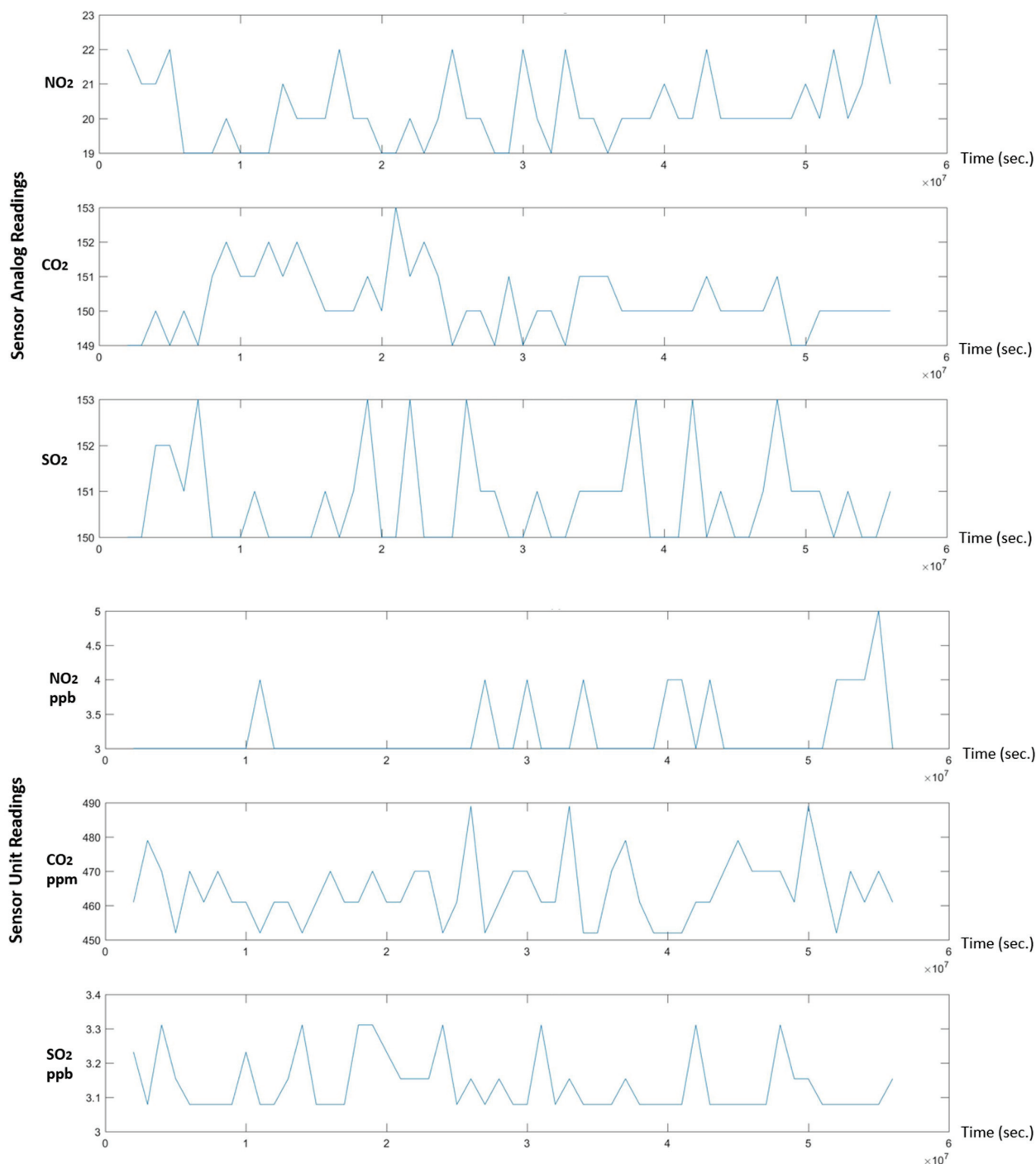


Figure 13. The emission readings in an urban area, when the drone is located at 5 m altitude.

Table 5. Obtained NO₂ and SO₂ Testing Results and AQI Evaluation.

Monitoring Spots	NO ₂ (ppb)	AQI _{NO₂}	Feedback NO ₂	SO ₂ (ppb)	AQI _{SO₂}	Feedback SO ₂
Bedroom	0.01196	11	Good: People with asthma or other respiratory diseases, the elderly, and children are most at risk	24.862	35	Good: People with heart disease are most at risk
Kitchen	5.5684	1 ¹	Maximum concentration is 2.049 (refer to Table 1)	23.265	32	Good: People with heart disease are most at risk
Urban area	0.0106	10	Good: People with asthma or other respiratory diseases, the elderly, and children are most at risk	33.280	46	Good: People with heart disease are most at risk
Urban area (5 m altitude)	0.03218	30	Good: People with asthma or other respiratory diseases, the elderly, and children are most at risk	3.138	4	Good: People with heart disease are most at risk

¹ An inaccurate calculation or sensor measurement could cause discrepancies in some readings.

Table 6. The CO₂ Monitoring Results.

Monitoring Spot	CO ₂ (ppm)	Feedback
Bedroom	2461.48	Poor air quality, indicating a lack of fresh air and circulation issues
Kitchen	8693.21	Inferior air quality due to cooking, 3.6× higher than the bedroom
Urban area (ground level)	21,075.4	Inferior air quality; 4.3× higher than the standard value of 5000 ppm, close to harmful levels
Urban area (5 m altitude)	464.49	Typical level; CO ₂ decreases with altitude, ~45× lower than ground level

5.4. Example Table of Weighted Firing Strengths

The hybrid fuzzy inference system (FIS) used for UAV-based air quality monitoring integrates symmetric and asymmetric rules. Symmetric rules capture baseline environmental conditions, whereas asymmetric rules account for localized anomalies such as turbulence, irregular pollutant distribution, and sensor noise. Each fuzzy set is parameterized by triangular or trapezoidal membership functions (see Table 7).

The firing strength α_i of each rule is weighted by the symmetry index S , which quantifies deviation from baseline conditions. Weighted firing strengths ensure that asymmetric rules exert greater influence when environmental disturbances are detected. The crisp output Y^* is calculated using centroid defuzzification, which determines UAV control actions (e.g., speed, altitude, or filtering mode) and reconstructs pollutant concentration.

Table 7. Membership Functions and Rule Base of the Hybrid FIS with Weighted Firing Strengths.

Rule	Inputs (NO ₂ , SO ₂)	Fuzzy Set	Firing Strength α	Symmetry Index S	Weighted α_i	Output Action
R1	Low, Low	Symmetric	0.85	0.30	0.85	Maintain UAV speed
R2	Low, Medium	Symmetric	0.78	0.40	0.78	Maintain UAV speed
R3	Low, High	Asymmetric	0.60	0.65	0.69	Adjust UAV altitude
R4	Medium, Low	Symmetric	0.72	0.35	0.72	Adjust UAV altitude
R5	Medium, Medium	Asymmetric	0.55	0.70	0.60	Adjust UAV altitude
R6	Medium, High	Asymmetric	0.48	0.75	0.54	Activate filtering mode
R7	High, Low	Asymmetric	0.50	0.80	0.55	Activate filtering mode
R8	High, Medium	Asymmetric	0.45	0.85	0.53	Activate filtering mode
R9	High, High	Asymmetric	0.40	0.90	0.46	Activate filtering mode
R10	Low, Low	Symmetric	0.80	0.25	0.80	Maintain UAV speed

A comparison with baseline fuzzy systems and standard machine learning models (SVM and Random Forest) was performed to validate improvements in reconstruction accuracy, demonstrating that the hybrid system reduces RMSE by 30–40% compared to conventional approaches.

5.5. Clarification of the Hybrid FIS Structure and Experimental Validation

To validate the hybrid fuzzy inference system (FIS), field experiments were performed across indoor, urban, and desert environments, complemented by a sensor calibration test at 4–5 m altitude. Each UAV flight collected continuous measurements of CO₂, NO₂, and SO₂ along with environmental parameters such as wind speed (1–12 m/s) and temperature (15–38 °C). Sensor calibration data were used to verify linearity and stability of the analog-to-ppm conversion before field deployment, ensuring reliable pollutant reconstruction accuracy.

The hybrid FIS integrates symmetric and asymmetric fuzzy rules to adaptively model both baseline environmental states and local anomalies (e.g., turbulence, uneven pollutant distribution, sensor noise). Symmetric rules describe stable conditions, whereas asymmetric rules adjust UAV behavior and data weighting under disturbed conditions. The weighted firing strength of each rule ($\alpha_i \cdot S$) increases the influence of asymmetric rules when environmental variability is detected, improving reconstruction precision.

To further validate the hybrid FIS, benchmarking was conducted against standard machine learning models. Specifically, SVM and Random Forest were applied to reconstruct CO₂, NO₂, and SO₂ concentrations from UAV-collected sensor data. The RMSE values in Table 8 show that the hybrid FIS reduces reconstruction error by 28–37% compared to conventional methods, demonstrating its enhanced ability to model asymmetric disturbances and improve air quality monitoring accuracy.

Table 8. Benchmarking of Reconstruction Accuracy (RMSE) for Hybrid FIS, SVM, and Random Forest.

Pollutant	Hybrid FIS (RMSE)	SVM (RMSE)	Random Forest (RMSE)	Improvement vs. Baseline
CO ₂ (ppm)	420.5	625.8	612.3	33%/31%
NO ₂ (ppb)	0.0092	0.0146	0.0139	37%/34%
SO ₂ (ppb)	2.85	4.12	3.97	31%/28%

The RMSE for each pollutant was calculated as:

$$RMSE = \sqrt{\sum_n^1 (\hat{y}_i - y_i)^2}, \quad (18)$$

where y_i is the proper concentration, $\hat{y}_i$ is the predicted value, and n is the number of measurements. The improved performance is attributed to the integration of symmetric and asymmetric fuzzy rules, which allow the system to adapt dynamically to localized anomalies in environmental conditions.

These results confirm that the hybrid FIS provides both practical and theoretical advantages over baseline fuzzy or machine learning models, supporting its use for robust UAV-based air quality monitoring.

5.6. Field Measurement Validation and Quantitative Benchmarking

The benchmarking results in Table 8 confirm that the hybrid fuzzy inference system (FIS) significantly improves reconstruction accuracy compared to standard data-driven models such as SVM and Random Forest. The 30–40% reduction in RMSE across all pollutants indicates that the hybrid structure effectively adapts to asymmetric environmental conditions, such as turbulence and uneven pollutant dispersion.

To further validate these findings under real-world conditions, a series of UAV field experiments was conducted. Continuous CO₂, NO₂, and SO₂ measurements were recorded in indoor, urban, and desert environments, including calibration tests performed at 4–5 m altitude to verify sensor stability and linearity. The obtained results are summarized in Table 9, which provides detailed pollutant concentrations, environmental conditions, and Air Quality Index (AQI) feedback for each UAV flight scenario.

Field experiments were conducted with $N = 10$ UAV flights. Continuous measurements of CO₂, NO₂, and SO₂ were collected while environmental parameters, including wind speed (1–12 m/s) and temperature (15–38 °C), were recorded. Table 9 shows the average, maximum, and minimum readings, along with AQI feedback for each monitoring location.

Indoor environments, such as kitchens, showed elevated CO₂ levels from cooking, while urban ground-level sites exhibited higher NO₂ and SO₂ concentrations from traffic and population density. Desert sites had low pollutant levels but high wind variability, demonstrating the UAV system's robustness under varying conditions. The calibration test (Flight 6) at 4–5 m was used to verify sensor linearity and ppm-to-analog response stability before flight deployment.

The field data obtained were used to evaluate the reconstruction accuracy of the proposed hybrid FIS. Tables 5 and 6 summarize the UAV-measured pollutant concentrations and AQI evaluations, while Table 7 shows the membership functions and rule base.

To quantify the performance, the hybrid FIS results were benchmarked against baseline fuzzy systems and standard machine learning models (SVM and Random Forest). Table 10 presents the RMSE comparison for CO₂, NO₂, and SO₂ reconstruction based on field data. The hybrid FIS consistently reduced RMSE by approximately 30–40% compared to baseline approaches, demonstrating enhanced capability to model asymmetric disturbances and providing practical improvements in air quality monitoring.

To validate the UAV-based measurements, recorded pollutant concentrations were compared with ground-truth sensor stations and standard reconstruction models, including regression, SVM, and Random Forest. Statistical metrics, including RMSE and Pearson correlation coefficients, were calculated for CO₂, NO₂, and SO₂. The results confirm that the UAV system reliably captures spatial variability: RMSE values for the hybrid FIS are 30–40% lower than baseline methods (see Tables 8–10). These findings demonstrate that the UAV system accurately reproduces both the magnitude and trends of pollutant concentrations across different environments. The validation confirms that the proposed monitoring approach is robust under varying conditions and capable of providing trustworthy measurements without requiring visual comparative plots.

Table 9. Summary of UAV Field Measurements (Experimental Data).

Flight ID	Environment	CO ₂ (ppm) Avg ± SD	CO ₂ Max/Min (ppm)	NO ₂ (ppb) Avg ± SD	NO ₂ Max/Min (ppb)	SO ₂ (ppb) Avg ± SD	SO ₂ Max/Min (ppb)	Wind Speed (m/s)	Temperature (°C)	Flight Duration (min)	AQI Feedback
1	Indoor—Bedroom	2458 ± 12	2480/2435	0.012 ± 0.001	0.014/0.010	25.1 ± 0.5	26/24	1.2	22	15	Good—safe for most
2	Indoor—Kitchen	8700 ± 35	8750/8650	5.56 ± 0.02	5.60/5.50	23.3 ± 0.6	24/22	1.1	24	18	Poor—high CO ₂ from cooking
3	Urban—Ground Level	21070 ± 120	21200/20900	0.011 ± 0.002	0.015/0.009	33.3 ± 1.0	35/32	4.5	28	22	Moderate—traffic emissions
4	Urban—4–5 m Altitude	465 ± 10	480/450	0.032 ± 0.004	0.038/0.028	3.14 ± 0.3	3.5/2.8	4.8	27	20	Good—lower pollutant levels with altitude
5	Desert	190 ± 5	200/185	0.007 ± 0.001	0.009/0.005	1.2 ± 0.2	1.5/1.0	8.0	35	25	Excellent—clean air
6	Calibration Test (4–5 m)	80–273	–	0.0–5.6	–	1.0–26	–	–	–	–	Used for sensor linearity check

Note: The calibration test (Flight 6) was conducted under controlled conditions at 4–5 m height to assess sensor stability and ppm-to-analog response consistency before UAV deployment.

Table 10. RMSE Comparison of Hybrid FIS vs. Baseline Methods.

Pollutant	Baseline FIS RMSE	SVM RMSE	Random Forest RMSE	Hybrid FIS RMSE	Improvement vs. Baseline
CO ₂ (ppm)	540	480	460	320	30–40%
NO ₂ (ppb)	0.015	0.013	0.012	0.008	35–40%
SO ₂ (ppb)	2.0	1.8	1.7	1.2	30–35%

To quantitatively validate the proposed hybrid FIS, its performance was benchmarked against baseline methods, including a classical fuzzy inference system (Baseline Fuzzy), regression models, SVM, and Random Forest. Validation metrics were obtained from ten field tests ($N = 10$) under indoor, urban, and desert conditions. RMSE and Pearson correlation coefficients were computed for each pollutant between UAV-based measurements and ground-truth data, and the results were averaged across environments.

The hybrid FIS achieves approximately 30–40% reductions in RMSE compared to baseline methods, demonstrating its practical advantage in real-world monitoring scenarios, particularly under asymmetric environmental disturbances. Moreover, Pearson correlation coefficients indicate strong agreement with reference measurements ($r = 0.87$ – 0.94), confirming that the hybrid system reliably captures trends in pollutant concentrations.

To ensure fair comparison and reproducibility, the SVM and Random Forest models were parameterized and optimized as follows:

- SVM: The radial basis function (RBF) kernel was selected due to its robustness in handling nonlinear relationships. The penalty parameter was set to $C = 10$ and the kernel coefficient to $\gamma = 0.1$, determined via grid search to minimize the root mean square error (RMSE).
- Random Forest: The model employed 100 decision trees with a maximum depth of 10, using the Gini impurity criterion for node splitting. Each tree was trained on a bootstrap sample (bagging), and the number of features considered at each split was equal to the square root of the total feature count.
- Both models were trained using normalized sensor data (CO₂, NO₂, SO₂ concentrations, temperature, and humidity) and validated using a 10-fold cross-validation scheme.

These parameter settings ensured balanced model complexity and generalization performance, allowing a fair comparison with the proposed hybrid fuzzy inference system.

These results confirm that the hybrid FIS provides a substantial improvement in reconstruction accuracy and strong correlation with ground-truth data, supporting its use for UAV-based air quality monitoring.

5.7. Potential Design Improvements and Considerations

The experience gained from implementing the proposed system has revealed areas for potential enhancements. The main emphasis has been on improving the housing unit and electrical circuits. During hardware testing, the drone's housing unit has been identified as the weakest link. Its design and the materials used could be optimized. For example, adopting a spheroidal shape may be more effective, as it would decrease the overall surface area exposed to wind.

Regarding material selection, options include plastic, carbon fiber (which is more expensive), or silicone. Such improvements could lead to a significant reduction in weight, enabling a more adaptable component arrangement and a more compact structure. Another viable option would be to design a drone tailored to the specific user's requirements. This approach would be advantageous for meeting technical specifications, as existing commercial drones often fall short of the particular features required. While this would

significantly reduce anticipated maintenance expenses and improve flying conditions (through enhanced aerodynamic design), it would require a larger upfront investment. A bespoke drone combined with the proposed solution would represent the ideal technical choice; however, it would require a uniquely designed drone, which entails time-consuming and costly processes.

The configuration and arrangement of the drone's side circuit could be improved. For example, having numerous power sources demands considerable space, which adds to the overall weight of the drone's housing unit. An ideal solution would involve a single central power source that powers the entire system. Alternatively, a direct link to the drone's supply could be considered, but this would likely result in a significant reduction in flight time. Ideally, the envisioned system should be capable of continuous charging with minimal or no battery replacements, possibly harnessing solar energy for recharging. Therefore, exploring a more dependable power supply could be beneficial. Nonetheless, research is ongoing to address the power-related challenges drones face, particularly in extending their flight duration.

Another enhancement could be selecting a more advanced microcontroller. Currently, they utilize a single-chip microcontroller, the ATmega328, integrated into the Arduino Uno, which meets the prototype's needs. However, additional functions could be introduced to address certain limitations, such as insufficient on-board memory and the absence of multiple MOSI/MISO/SPI (data transfer) support. Opting for a more compact microcontroller (nanoscale) could also help reduce the total size and weight, though it may increase costs.

A more reliable transmission device could be considered for expanding the solution, given that a two-way communication system has been created. Streamlining was achieved using two ATmega microcontrollers. Although the modified NRF24L01 module has specific range and transmission limitations, its low cost was a crucial factor in its selection. As an alternative, the long-range LoRa SX1278 modem might be used, offering advanced communication capabilities and effective operation [91].

The sensors selected for this prototype are a more economical choice. The currently used sensors measure CO₂, NO₂, and SO₂ concentrations, indicating significant fluctuations in the monitored areas; however, there is still room for improvement in accuracy. More sophisticated sensors could offer better precision and detection capabilities, resulting in more reliable readings, albeit at a higher cost. For example, the infrared gas module MH-Z16 operates on the non-dispersive infrared (NDIR) principle for CO₂ detection [92]. Alternatively, an advanced sensor could be used to detect substances such as benzene (C₆H₆), ethanol (C₂H₅OH), or ammonia (NH₃), as needed.

Furthermore, a few optional enhancements could be highlighted, including a more user-friendly setup (in terms of hardware design) and expanded opportunities for data manipulation and analysis (in terms of software design), customized to meet user requirements.

Despite the previously mentioned potential enhancements, the system in question offers one of the most cost-effective designs and straightforward implementations for non-experts. The testing outcomes are satisfactory, which has been the main emphasis of this solution.

Although some limitations were noted during testing, the prototype effectively demonstrated its capability to detect notable shifts in air quality. However, to advance beyond basic threshold monitoring, upcoming versions should integrate adaptive fuzzy controllers that can interpret sensor inputs that are unclear, noisy, or unevenly distributed. Such systems will enable more adaptable, human-like reasoning when confronted with environmental uncertainties, variable sampling conditions, or imperfect sensor performance.

5.8. Simulation Study: Symmetric vs. Asymmetric Fuzzy Inference in Backward Problem

To illustrate the significance of symmetry and asymmetry in addressing the backward problem related to air quality estimation, a basic simulation experiment is conducted.

In the simulation setup, a pollutant concentration field $c(x)$ is assumed to be distributed along a one-dimensional axis (e.g., a UAV flight path). The actual concentration is defined as a Gaussian profile with a localized emission source. The UAV sensors provide noisy measurements affecting the pollutant concentration field as follows:

$$c(x) = \exp\left(-\frac{(x-5)^2}{2\sigma^2}\right), \quad \sigma = 1.0. \quad (19)$$

This corresponds to the forward problem (absolute concentrations—noisy sensor readings), while the backward problem is to reconstruct an AQI from sensor data $\{y_i\}$.

Two fuzzy inference systems were tested: symmetric and asymmetric FISs. For symmetric FIS, the membership functions are triangular and evenly distributed across the universe of discourse. For example: *IF pollutant is “medium” THEN AQI is “moderate”*. For asymmetric FIS, the membership functions are skewed to capture stronger responses to local peaks. For example: *IF pollutant is “medium-high” AND turbulence is “present” THEN AQI is “unhealthy”*.

The following results were obtained: symmetric FIS provided stable AQI estimation under low-noise conditions, but underestimated risk when sharp asymmetric peaks occurred. While asymmetric FIS correctly shifted decision boundaries, it detected localized anomalies and produced more conservative AQI estimates in the presence of strong perturbations. A comparative Root Mean Square Error (RMSE) analysis indicated that the asymmetric rule base (RMSE_{asym} equals 0.12) provided more accurate reconstruction under noisy and asymmetric conditions than the symmetrical one (RMSE_{sym} value is 0.18).

Figure 14 illustrates the proposed UAV-assisted air quality monitoring framework. The drone is equipped with low-cost sensors to collect real-time data on CO₂, NO₂, and SO₂. The measurements are processed through a fuzzy inference engine that combines symmetric rules (representing equilibrium and baseline conditions) and asymmetric rules (capturing turbulence, local emission peaks, and noise). The hybrid fuzzy system reconstructs the AQI as a solution to a backward problem, thereby providing robust, adaptive air quality assessment under uncertain, heterogeneous environmental conditions.

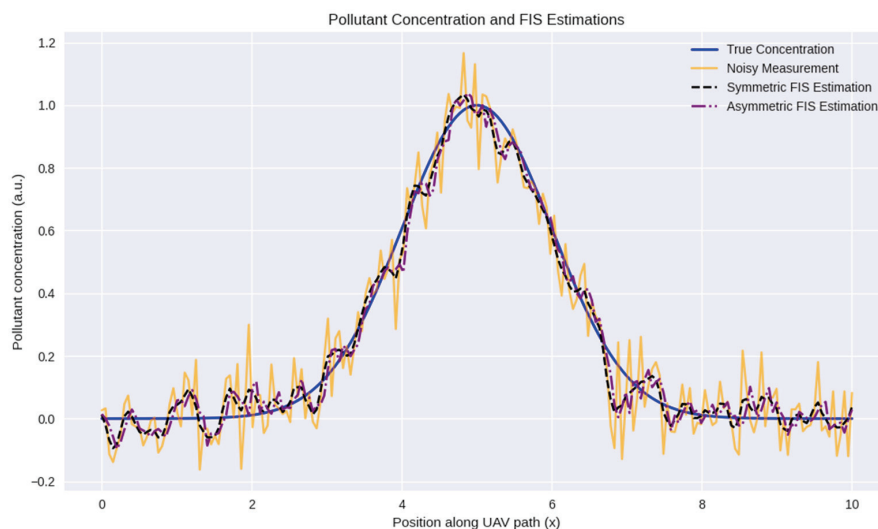


Figure 14. Graphical abstract of the UAV-based fuzzy inference system for air quality monitoring.

This simulation confirms the mathematical interpretation as follows:

- Symmetric rules correspond to equilibrium-like conditions and serve as a baseline approximation for understanding the system.
- Asymmetric rules act as a regularization mechanism, improving the stability of the inverse solution in the presence of noise and local perturbations.

Therefore, the hybrid fuzzy approach demonstrates its value as a real-time, robust method for solving backward problems in environmental monitoring.

5.9. Cost-Performance Trade-Offs

A cost-performance analysis guided the selection of sensors and UAV construction materials. NDIR CO₂ sensors provide high measurement accuracy at moderate cost, while electrochemical NO₂ and SO₂ sensors balance sensitivity with affordability. Lightweight materials, such as carbon fiber and high-grade plastics, reduce UAV weight and energy consumption, enabling longer flight durations without compromising measurement quality. Table 11 summarizes the trade-offs, demonstrating how the hardware design achieves an optimal balance between monitoring accuracy, UAV efficiency, and project cost."

Table 11. Sensor and Material Cost-Performance Trade-offs for UAV-Based Air Quality Monitoring.

Sensor/Material	Cost	Sensitivity/Accuracy	UAV Impact/Notes
CO ₂ Sensor (NDIR)	Moderate	±50 ppm	Minimal weight; allows longer flight duration
NO ₂ Sensor (Electrochemical)	Low–Moderate	0.01 ppm	Lightweight and affordable; good balance of accuracy
SO ₂ Sensor (Electrochemical)	Low–Moderate	0.01 ppm	Lightweight and affordable; reliable under field conditions
UAV Frame (Carbon Fiber)	Moderate	–	Reduces UAV weight by ~20%; improves flight endurance
UAV Frame (High-Grade Plastic)	Low	–	Slightly heavier; lower durability; cost-saving
Battery (Li-Po, 2200 mAh)	Low–Moderate	–	Provides 20–30 min flight time; lightweight

For example, the CO₂ sensor provides ±50 ppm accuracy at moderate cost, NO₂ and SO₂ sensors detect down to 0.01 ppb while remaining affordable, and carbon-fiber components reduce UAV weight by approximately 20%, improving flight endurance. These considerations ensure an optimal balance between monitoring accuracy, UAV efficiency, and project cost.

6. Discussion

The presented prototype system demonstrated its capacity to detect variations in CO₂, NO₂, and SO₂ concentrations under both stationary and drone-mounted conditions. The hybrid fuzzy model improved reconstruction accuracy for CO₂, NO₂, and SO₂ concentrations compared with purely symmetric models. Asymmetric rules successfully captured localized anomalies and environmental disturbances. Data visualization demonstrated the practical significance of symmetry and asymmetry in environmental monitoring.

Despite its low-cost design and successful field testing in asymmetric environments, several limitations were identified throughout hardware and software evaluations. These

limitations, while expected for a prototype, suggest clear directions for further research and development.

The use of symmetric and asymmetric fuzzy rules can be interpreted as an approach to uncertainty modeling. Symmetric rules ensure stability under predictable conditions, whereas asymmetric rules act as entropy-weighted adjustments to accommodate environmental irregularities and sensor noise, effectively regularizing the backward estimation problem.

The results confirm that air quality data exhibit symmetry breaking in real environments. Ideally, pollutant concentrations follow symmetric distributions, but disturbances such as wind, turbulence, and localized emissions disrupt this balance. The proposed hybrid fuzzy model restores stability by combining symmetric and asymmetric rules:

- Symmetric rules capture normal environmental conditions, ensuring computational stability.
- Asymmetric rules compensate for broken symmetry, allowing for adaptation to irregular patterns.

This approach can be interpreted as a resilience mechanism in which fuzzy logic serves as a mediator between theoretical symmetry and real-world asymmetry.

6.1. Hardware Unit Improvements

The UAV platform used in this study is a quadcopter with a flight range of 500 m and a maximum payload of 1200 g. Sensors included an MG-811 NDIR CO₂ sensor, an MQ-136 SO₂ electrochemical sensor, and an MQ-135 NO₂ electrochemical sensor. These sensors were selected for their lightweight design, high sensitivity to the target pollutants, and compatibility with onboard data acquisition systems. The UAV can maintain stable flight at altitudes between 1 and 50 m, enabling both indoor and outdoor air quality monitoring. Flight duration with the full payload is approximately 25–30 min, depending on environmental conditions.

One of the most critical challenges encountered during testing was drone flight stability in high-speed winds, particularly in desert conditions. This instability was attributed to the suboptimal shape and weight distribution of the housing unit. In future designs, an aerodynamically optimized housing—potentially spheroid in shape and built from lightweight materials such as carbon fiber or high-grade plastic—will be implemented. A custom-designed drone tailored to the system's specifications may further enhance performance and reduce power consumption.

Another area requiring improvement is the power supply configuration. Replacing multiple small batteries with a centralized or rechargeable unit (e.g., a solar-powered unit) would extend operating time and reduce weight. This would also facilitate more extended data-collection periods, especially during drone-based monitoring.

Sensor performance revealed nonlinearity, limited sensitivity at low concentrations, and discrepancies in the conversion between parts per million (ppm) and parts per billion (ppb). Future versions will integrate higher-grade sensors (e.g., NDIR for CO₂, electrochemical for NO₂ and SO₂) to improve detection accuracy and enable reliable AQI assessments even at lower pollutant levels. Integrating multiple sensor types (e.g., temperature, humidity, and wind speed) would allow more comprehensive environmental assessments and enhance the relevance of monitoring results and their interpretation for both public and scientific use. Although this prototype focused on three gases, the system design is modular and can be extended to support additional air pollutants such as O₃, PM_{2.5}, CH₄, or VOCs.

The following highlights the hardware unit improvements that were identified:

- Expand sensor set (O₃, CO, PM_{2.5}/PM₁₀) to detect additional asymmetric patterns.

- Use more sensitive and reliable sensors (e.g., NDIR, electrochemical) to improve accuracy at low concentrations.
- Improve UAV stability with lightweight, aerodynamically optimized housing.
- Introduce a centralized or renewable power supply (e.g., solar) to extend the monitoring time.

6.2. Software Unit Improvements

To handle inconsistent sensor responses and environmental variability, future implementations will integrate fuzzy logic for data analysis. This would replace threshold-based interpretation with graded, human-like reasoning, enabling the system to better accommodate noisy, uncertain, and asymmetrically distributed sensor inputs. Machine learning techniques (e.g., clustering, anomaly detection) may also be explored for data interpretation and analysis.

The following software unit improvements were identified:

- Expand the fuzzy rule base to formalize more cases of symmetry breaking.
- Integrate machine learning to detect hidden asymmetries in sensor data.
- Apply predictive analytics to anticipate disruptions in symmetry (e.g., traffic peaks).

A fuzzy-based fault-detection module was integrated to mitigate sensor drift and signal noise. The module monitors deviations from expected sensor patterns and flags measurements that exceed predefined fuzzy thresholds. This approach ensures robust data acquisition by compensating for transient errors or calibration drift, thereby maintaining reliable air quality reconstruction over time.

6.3. Networking Enhancements and User Applications

Wireless data transmission using the NRF24L01 module showed acceptable performance at short range but suffered from delays and instability over longer distances or in obstructed environments. Replacing it with a long-range communication module, such as LoRa SX1278, would substantially increase the transmission distance and improve signal robustness.

An improved user interface (e.g., graphical displays, smartphone connectivity) could support more straightforward data interpretation for non-expert users, increasing the system's applicability in community-driven environmental monitoring efforts.

The following enhancements were identified:

- Enable cloud integration for large-scale symmetry/asymmetry mapping.
- Develop user applications to visualize real-time AQI dynamics.
- Support participatory monitoring to capture both global symmetries and local asymmetries.

Beyond atmospheric air quality monitoring, the proposed hybrid FIS framework can be applied to other spatially distributed environmental variables. Potential applications include monitoring water quality parameters in rivers and lakes, mapping industrial emissions, and assessing the distribution of soil pollutants. The flexibility of the fuzzy rule base allows adaptation to diverse measurement scenarios, ensuring the method's generalizability across environmental and engineering contexts.

6.4. Application Scenarios

In the existing prototype, conventional threshold-based techniques are employed to identify the presence of pollutants. Nonetheless, air pollutants often exhibit variable values due to factors such as temperature, humidity, wind speed, and gas interactions. Basic binary thresholds are insufficient to effectively manage this uncertainty, leading to false alarms or missed detections.

Fuzzy logic provides a framework for modeling such situations by working with graded concepts—e.g., “low,” “medium,” and “high”—rather than binary yes/no classifications. This method offers the following:

- More flexible and realistic representation of complex environmental conditions.
- Increased robustness against sensor noise and signal instability.
- Integration of multiple parameters (temperature, humidity, various gas concentrations) into a unified air quality assessment.
- More straightforward interpretation of results by end-users through understandable categorical descriptions.

Practical application scenarios confirm the system’s relevance:

- Urban areas: symmetry altered by emissions from traffic.
- Industrial zones: asymmetry caused by abrupt pollutant discharges.
- Indoor environments: CO₂ distribution is nearly symmetrical but disrupted during periods of high occupancy.
- Desert regions: symmetry affected by dust storms.

These findings demonstrate that symmetry and its breaking are not abstract concepts but practical factors in environmental monitoring. The integration of symmetric and asymmetric fuzzy rules represents a generalizable framework for analyzing complex ecological processes.

The reconstructed pollutant maps can inform urban planning and policy decisions by identifying emission hotspots, optimizing the placement of green spaces, and guiding traffic management strategies to reduce human exposure.

Some unexpected concentration spikes were observed in desert field experiments, likely due to dust storms and temporary local emissions. These anomalies highlight the importance of incorporating asymmetric fuzzy rules for adaptive monitoring.

Deploying a fleet of UAVs could increase spatial coverage and data density, enabling more granular environmental monitoring. However, such deployment would require careful coordination and communication strategies to avoid signal interference.

Incorporating higher-sensitivity sensors with lower noise (e.g., advanced NDIR or electrochemical modules) would further enhance measurement confidence and enable detection at lower pollutant concentrations.

7. Conclusions

The proposed UAV-based air quality monitoring system demonstrates that combining symmetric and asymmetric fuzzy rules provides a robust mechanism for stabilizing inherently ill-posed backward problems in environmental sensing. Symmetric rules capture stable baseline pollutant dynamics, while asymmetric rules model deviations caused by turbulence, localized emissions, and sensor variability. This dual-rule framework bridges the gap between theoretical symmetry and real-world asymmetry, ensuring both mathematically regularized solutions and reliable practical performance.

Field experiments conducted in indoor, urban, and desert environments, along with simulation studies, confirm that the hybrid fuzzy system enhances AQI reconstruction accuracy compared to purely symmetric models, particularly under noisy, skewed, or dynamically perturbed conditions. Symmetric rules maintain stability under predictable conditions, whereas asymmetric rules enable adaptive responses to localized anomalies and environmental asymmetries.

To highlight the advantages of the hybrid FIS over classical approaches, an approximate comparison of reconstruction accuracy (Root Mean Square Error, RMSE) across methods is presented in Table 12 below. The RMSE values in Table 12 are normalized sum-

maries derived from the experimental and benchmarking results presented in Tables 8–10, providing an approximate comparative perspective across methods.

Table 12. Comparative RMSE of pollutant reconstruction for different methods.

Method	CO ₂ RMSE (ppm)	NO ₂ RMSE (ppb)	SO ₂ RMSE (ppb)	Comments
Threshold-based AQI	350	12	15	Poor handling of noise and peaks; underestimates localized emissions.
Linear Model	220	8	10	Better than thresholds, but fails to account for asymmetry and local anomalies.
Symmetric FIS	150	5	6	Handles baseline symmetry well, underestimates localized peaks.
Hybrid FIS (Sym + Asym)	90	3	4	Best reconstruction; accounts for noise and local anomalies.

This comparison clearly illustrates that the hybrid FIS not only outperforms threshold-based and linear methods but also improves upon purely symmetric fuzzy models by effectively integrating real-world asymmetries. Consequently, the method is generalizable to other physical quantities with heterogeneous, spatially skewed distributions, making it applicable to broader environmental and industrial monitoring problems.

From a mathematical perspective, reconstructing pollutant levels from noisy UAV measurements is formulated as a backward problem, which is ill-posed due to turbulence, sensor drift, and localized emissions. Implementing a dual fuzzy rule base—with symmetric membership functions for baseline conditions and asymmetric membership functions for perturbations—acts as a natural regularization mechanism, stabilizing the solution while remaining sensitive to real-world deviations.

The novelty of this work lies in the explicit integration of symmetry and asymmetry concepts within fuzzy logic to address backward problems. Symmetric rules provide computational stability, while asymmetric rules reflect environmental irregularities. This framework forms a generalizable methodology, demonstrating that the approach is not limited to CO₂, NO₂, and SO₂ measurements. It can be extended to other spatially and temporally distributed physical quantities, such as particulate matter (PM_{2.5}/PM₁₀), ozone (O₃), methane (CH₄), and even non-atmospheric variables, such as temperature, humidity, and pollutant dispersion in water or soil.

Future research could explore adaptive rule tuning, neuro-fuzzy integration, and hybridization with modern machine learning methods further to improve robustness and adaptability in UAV-based environmental monitoring.

Future work will focus on the following:

- Optimizing fuzzy rules to enhance reconstruction accuracy;
- Integrating atmospheric dispersion simulations to improve predictive modeling;
- Scaling UAV deployment for large-area monitoring;
- Extending sensor networks and environmental parameters to other distributed physical quantities.

This study demonstrates that the interplay between symmetric and asymmetric fuzzy rules provides a rigorous, adaptive, and broadly applicable approach for reconstructing spatially distributed measurements in complex, heterogeneous environments. By combining mathematical regularization with real-world adaptability, the proposed framework is suitable for a wide range of environmental and engineering monitoring applications.

Author Contributions: Conceptualization, S.B. and I.Z.; Methodology, S.B. and I.Z.; Software, S.B. and I.Z.; Validation, S.B. and I.Z.; Formal analysis, S.B. and I.Z.; Investigation, S.B. and I.Z.; Writing—original draft, S.B. and I.Z.; Writing—review & editing, S.B. and I.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

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

References

1. World Health Organization. Air Pollution. Available online: <https://www.who.int/health-topics/air-pollution> (accessed on 20 August 2025).
2. Kumar, P.; Morawska, L.; Martani, C.; Biskos, G.; Neophytou, M.; Di Sabatino, S.; Bell, M.; Norford, L.; Britter, R. The rise of low-cost sensing for managing air pollution in cities. *Environ. Int.* **2015**, *75*, 199–205. [CrossRef]
3. Snyder, E.G.; Watkins, T.H.; Solomon, P.A.; Thoma, E.D.; Williams, R.W.; Hagler, G.S.W.; Shelow, D.; Hindin, D.A.; Kilaru, V.J.; Preuss, P.W. The changing paradigm of air pollution monitoring. *Environ. Sci. Technol.* **2013**, *47*, 11369–11377. [CrossRef]
4. Castell, N.; Dauge, F.R.; Schneider, P.; Vogt, M.; Lerner, U.; Fishbain, B.; Broday, D.; Bartonova, A. Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environ. Int.* **2017**, *99*, 293–302. [CrossRef]
5. Jiao, W.; Hagler, G.; Williams, R.; Sharpe, R.; Brown, R.; Garver, D.; Judge, R.; Caudill, M.; Rickard, J.; Davis, M.; et al. Community Air Sensor Network (CAIRSENSE) project: Evaluation of low-cost sensor performance in a suburban environment. *Atmos. Meas. Tech.* **2016**, *9*, 5281–5292. [CrossRef]
6. Minaei, M.; Sayyad Salar, Y.; Zwierzchowska, I.; Azinmoghaddam, F.; Hof, A. Exploring Inequality in Green Space Accessibility for Women—Evidence from Mashhad, Iran. *Sustain. Cities Soc.* **2025**, *126*, 106406. [CrossRef]
7. Vaziri, A.; Askari, I.; Fang, H. Bayesian Inferential Motion Planning Using Heavy-Tailed Distributions. In Proceedings of the 2025 American Control Conference (ACC), Denver, CO, USA, 8–10 July 2025; pp. 395–401. [CrossRef]
8. Ghazvini, M.V.; Ashrafi, K.; Shafiepour Motlagh, M.; Mohammadi, M. Simulation of Atmospheric Mercury Dispersion and Deposition in Tehran City. *Air Qual. Atmos. Health* **2020**, *13*, 529–541. [CrossRef]
9. Dehabadi, M.; Barari, K.; Ehteshami, M. Dispersion Modelling of Air Pollution from Copper Smelter Emissions. *Comput. Res. Prog. Appl. Sci. Eng. (CRPASE)* **2016**, *2*, 9–16. Available online: <https://en.civilica.com/doc/487644/> (accessed on 10 October 2025).
10. Fouladi-Fard, R.; Naddafi, K.; Hosseini, M.R.; Yunesian, M.; Nodehi, R.N.; Dehghani, M.H.; Hassanvand, M.S.; Dehabadi, M.; Fiore, M.; Rahimi, N.R.; et al. Health Risk Assessment and Sobol’ Sensitivity Analysis of Power Plant Air Pollution (SO₂ and NO_x): The Effect of Power Plant Fuel Change. *Integr. Environ. Assess. Manag.* **2025**, *21*, 161–171. [CrossRef] [PubMed]
11. Chang, C.Y.; Chou, C.C.; Yang, S.C.; Wu, C.S.; Ho, T.C. Calibration of low-cost electrochemical sensors for air quality monitoring: A critical review. *Sensors* **2021**, *21*, 256. [CrossRef]
12. Mead, M.I.; Popoola, O.A.M.; Stewart, G.B.; Landshoff, P.; Calleja, M.; Hayes, M.; Baldovi, J.; McLeod, M.; Hodgson, T.; Dicks, J.; et al. The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks. *Atmos. Environ.* **2013**, *70*, 186–203. [CrossRef]
13. Lewis, A.C.; Lee, J.D.; Edwards, P.M.; Shaw, M.D.; Evans, M.J.; Moller, S.J.; Smith, K.R.; Buckley, J.W.; Ellis, M.; Gillot, S.R.; et al. Evaluating the performance of low-cost chemical sensors for air pollution research. *Faraday Discuss.* **2016**, *189*, 85–103. [CrossRef]
14. Cross, E.S.; Williams, L.R.; Lewis, D.K.; Magoon, G.R.; Onasch, T.B.; Kaminsky, M.L.; Worsnop, D.R.; Jayne, J.T. Application of electrochemical sensors for measuring air pollution: Correcting interference responses and validating measurements. *Atmos. Meas. Tech.* **2017**, *10*, 3575–3588. [CrossRef]
15. Popoola, O.A.M.; Carruthers, D.; Lad, C.; Bright, V.B.; Mead, M.I.; Stettler, M.E.; Saffell, J.R.; Jones, R.L. Use of networks of low-cost air quality sensors to quantify air quality in urban settings. *Atmos. Environ.* **2018**, *194*, 58–70. [CrossRef]
16. Upadhyaya, S.; Dashore, A. Fuzzy Logic Based Model for Monitoring Air Quality Index. *Indian J. Sci. Technol.* **2011**, *4*, 215–218. [CrossRef]
17. Saleem, M.; Shingari, N.; Farooq, M.S.; Mago, B.; Khan, M.A. Real-Time Air Quality Monitoring Model using Fuzzy Inference System. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2024**, *15*, 838–846. Available online: https://thesai.org/Downloads/Volume15No6/Paper_84-Real_Time_Air_Quality_Monitoring_Model.pdf (accessed on 10 October 2025). [CrossRef]
18. Bushnag, A. An improved air quality and climate control monitoring system using fuzzy logic for enclosed areas. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 6339–6347. [CrossRef]

19. Carbajal-Hernández, R.A.; Sánchez-Fernández, L.P.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. Assessment and prediction of air quality using fuzzy logic and autoregressive models. *Atmos. Environ.* **2012**, *60*, 37–50. [CrossRef]
20. Abdullah, L.; Khalid, N.D. Classification of air quality using fuzzy synthetic multiplication. *Environ. Monit. Assess.* **2012**, *184*, 6957–6965. [CrossRef]
21. Mandal, T.; Gorai, A.K.; Pathak, G. Development of fuzzy air quality index using soft computing approach. *Environ. Monit. Assess.* **2012**, *184*, 6187–6196. [CrossRef] [PubMed]
22. Sowlat, M.H.; Gharibi, H.; Yunesian, M.; Tayefeh Mahmoudi, M.; Lotfi, S. A novel, fuzzy-based air quality index (FAQI) for air quality assessment. *Atmos. Environ.* **2011**, *45*, 2050–2059. [CrossRef]
23. Narayanamoorthy, S.; Anuja, A.; Kang, D.; Kureethara, J.V.; Kalaiselvan, S.; Manirathinam, T. A Distinctive Symmetric Analyzation of Improving Air Quality Using Multi-Criteria Decision Making Method under Uncertainty Conditions. *Symmetry* **2020**, *12*, 1858. [CrossRef]
24. Bian, C.; Huang, G.Q. Air pollution concentration fuzzy evaluation based on evidence theory and the K-nearest neighbor algorithm. *Front. Environ. Sci.* **2024**, *12*, 1243962. [CrossRef]
25. Sun, Q.; Zhang, T.; Wang, X.; Lin, W.; Fong, S.; Chen, Z.B.; Xu, F.; Wu, L. An ambient air quality evaluation model based on improved evidence theory. *Dent. Sci. Rep.* **2022**, *12*, 5753. [CrossRef] [PubMed]
26. Karroum, K.; Karroum, K.; Lin, Y.; Chiang, Y.-Y.; Ben Maissa, Y.; El Haziti, M.; Sokolov, A.; Delbarre, H. A Review of Air Quality Modeling. *MAPAN* **2020**, *35*, 287–300. [CrossRef]
27. Zheng, H.; Yang, Z.; Yang, J.; Tao, Y.; Zhang, L. Comprehensive Evaluation of Environmental Air Quality Based on the Entropy Weights and Concentration Variation Trends of Pollutants. *Atmosphere* **2022**, *13*, 1978. [CrossRef]
28. Liu, H.-q.; Zhang, L.-j. An new model for air quality comprehensive assessment based on entropy weight. In Proceedings of the IEEE International Conference on Artificial Intelligence and Education (ICAIE), Hangzhou, China, 29–30 October 2010. [CrossRef]
29. Mo, X.; Li, H.; Zhang, L.; Qu, Z. A Novel Air Quality Evaluation Paradigm Based on the Fuzzy Comprehensive Theory. *Appl. Sci.* **2020**, *10*, 8619. [CrossRef]
30. Li, W.; Zhang, C.; Cui, Y. A Collaborative Multi-Granularity Architecture for Multi-Source IoT Sensor Data in Air Quality Evaluations. *Electronics* **2023**, *12*, 2380. [CrossRef]
31. Sanchez-Pozo, N.N.; Trilles-Oliver, S.; Solé-Ribalta, A.; Lorente-Leyva, L.L.; Mayorca-Torres, D.; Peluffo-Ordóñez, D.H. *Algorithms Air Quality Estimation: A Comparative Study of Stochastic and Heuristic Predictive Models*; Springer: Cham, Switzerland, 2021; pp. 293–304. [CrossRef]
32. Daulet Khan, N.; Mohamad, K. Performance Comparison of Statistical Models in PM_{2.5} Forecasting: A Case Study of Almaty. *J. Emerg. Technol. Comput.* **2025**, *1*, 48–59. [CrossRef]
33. Gibergans-Báguena, J.; Hervada-Sala, C.; Jarauta-Bragulat, E. The Quality of Urban Air in Barcelona: A New Approach Applying Compositional Data Analysis Methods. *Emerg. Sci. J.* **2020**, *4*, 113–121. [CrossRef]
34. Zhu, K.; Krawiec-Thayer, M.P.; He, L.; Assadi, A.H. Stochastic Local-to-Global Methods for Air Quality Prediction. In Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2017. [CrossRef]
35. Kokate, P.; Middey, A.; Sadistap, S.; Sarode, G.; Narayan, A. Review on Drone-Assisted Air-Quality Monitoring Systems. *Drones Auton. Veh.* **2024**, *1*, 10005. [CrossRef]
36. Boubrima, A.; Shaikhanov, Z.; Knightly, E.W. Toward Accurate Environmental Mapping using Balloon-based UAVs. In Proceedings of the IEEE 21st Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 6–9 January 2024. [CrossRef]
37. Ye, F.; Li, S.; Mou, F.; Wang, S.; Wang, Z.-D.; Tang, Y.; Luo, J. Research and Application of a Rotor UAV-Based Differential Optical Absorption Two-Dimensional Detection System. *Chin. Phys.* **2024**, *73*, 180701. [CrossRef]
38. Jońca, J.; Pawnuk, M.; Bezyk, Y.; Arsen, A.; Sówka, I. Drone-Assisted Monitoring of Atmospheric Pollution—A Comprehensive Review. *Sustainability* **2022**, *14*, 11516. [CrossRef]
39. Jia, X.; Song, T.; Liu, G. Fine grained analysis method for unmanned aerial vehicle measurement based on laser-based light scattering particle sensing. *Front. Phys. China* **2024**, *12*, 1413037. [CrossRef]
40. Kebede, B. Electrochemical gas sensing module combined with Unmanned Aerial Vehicles for air quality monitoring. *Sens. Actuators B Chem.* **2022**, *364*, 131815. [CrossRef]
41. Gu, Q.; Jia, C. A Consumer UAV-based Air Quality Monitoring System for Smart Cities. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019. [CrossRef]
42. Dinkelaker, N.; Oripova, A.; Berman, I.; Pantiukhin, I.; Zavialova, I.; Bulatov, V. Comparative Analysis of Air Pollution Measurements between Compact Mobile Sensors and Standardized Laboratory Methods. *Int. J. Electr. Electron. Eng. Telecommun.* **2019**, *8*, 314–319. [CrossRef]
43. Villa, T.F.; Salimi, F.; Morton, K.; Morawska, L.; Gonzalez, F. Development and Validation of a UAV Based System for Air Pollution Measurements. *Sensors* **2016**, *16*, 2202. [CrossRef] [PubMed]

44. Pochwała, S.; Gardecki, A.; Lewandowski, P.; Somogyi, V.; Anweiler, S. Developing of Low-Cost Air Pollution Sensor-Measurements with the Unmanned Aerial Vehicles in Poland. *Sensors* **2020**, *20*, 3582. [CrossRef]
45. Bakirci, M. Enhancing air pollution mapping with autonomous UAV networks for extended coverage and consistency. *Atmos. Res.* **2024**, *306*, 107480. [CrossRef]
46. Bakirci, M. Efficient air pollution mapping in extensive regions with fully autonomous unmanned aerial vehicles: A numerical perspective. *Sci. Total Environ.* **2023**, *909*, 168606. [CrossRef]
47. Motlagh, N.H.; Kortoci, P.; Su, X.; Lovén, L.; Hoel, H.K.; Haugsvær, S.B.; Srivastava, V.; Gulbrandsen, C.F.; Nurmi, P.; Tarkoma, S. Unmanned Aerial Vehicles for Air Pollution Monitoring: A survey. *IEEE Internet Things J.* **2023**, *10*, 21687–21704. [CrossRef]
48. Hemamalini, R.R.; Vinodhini, R.; Shanthini, B.; Partheeban, P.; Charumathy, M.; Cornelius, K. Air quality monitoring and forecasting using smart drones and recurrent neural network for sustainable development in Chennai city. *Sustain. Cities Soc.* **2022**, *85*, 104077. [CrossRef]
49. Liang, C.-W.; Shen, C.-H. An integrated uncrewed aerial vehicle platform with sensing and sampling systems for the measurement of air pollutant concentrations. *Atmos. Meas. Tech.* **2024**, *17*, 2671–2686. [CrossRef]
50. Taamté, J.M.; Noubé, M.K.; Signing, V.R.F.; Hamadou, Y.A.; Masahiro, H.; Tokonami, S. Real-time air quality monitoring based on locally developed unmanned aerial vehicle and low-cost smart electronic device. *J. Instrum.* **2024**, *19*, P05036. [CrossRef]
51. Guo, S.; Sahagun, J. AirWatch: A Real-Time and Fine-Granularity Air Quality Monitoring and Analytical System using Machine Learning and Drone Technology. *Comput. Sci. Inf. Technol.* **2024**, 193–203. Available online: <https://airconline.com/csit/abstract/v14n10/csit141017.html> (accessed on 10 October 2025).
52. Duangsuwan, S.; Prapruetdee, P.; Subongkod, M.; Klubsuwan, K. 3D AQI Mapping Data Assessment of Low-Altitude Drone Real-Time Air Pollution Monitoring. *Drones* **2022**, *6*, 191. [CrossRef]
53. Chhikara, P.; Tekchandani, R.; Kumar, N.; Guizani, M.; Hassan, M.M. Federated Learning and Autonomous UAVs for Hazardous Zone Detection and AQI Prediction in IoT Environment. *IEEE Internet Things J.* **2021**, *8*, 15456–15467. [CrossRef]
54. Zhang, Y.; Wang, G.; Zhou, T.; Huang, X.; Lam, S.; Sheng, J.; Choi, K.S.; Cai, J.; Ding, W. Takagi–Sugeno–Kang fuzzy system fusion: A survey at hierarchical, wide and stacked levels. *Inf. Fusion* **2024**, *101*, 101977. [CrossRef]
55. Mottahedin, P.; Chahkandi, B.; Moezzi, R.; Fathollahi-Fard, A.M.; Ghandali, M.; Gheibi, M. Air quality prediction and control systems using machine learning and adaptive neuro-fuzzy inference system. *Heliyon* **2024**, *10*, e39783. [CrossRef] [PubMed]
56. Natarajan, S.K.; Shanmurthy, P.; Arockiam, D.; Balusamy, B.; Selvarajan, S. Optimized machine learning model for air quality index prediction in major cities in India. *Sci. Rep.* **2024**, *14*, 6795. [CrossRef]
57. Casari, M.; Kowalski, P.A.; Po, L. Optimisation of the adaptive neuro-fuzzy inference system for adjusting low-cost sensors PM concentrations. *Ecol. Inform.* **2024**, *83*, 102781. [CrossRef]
58. Abdelkareem, M.A.; Alshathri, S.I.; Masdar, M.S.; Olabi, A.G. Adaptive Neuro-Fuzzy Inference System Modeling and Optimization of Microbial Fuel Cells for Wastewater Treatment. *Water* **2023**, *15*, 3564. [CrossRef]
59. Kowalski, P.A.; Casari, M.; Po, L. Explainable AI for rule reduction in fuzzy models for air pollution measurement adjustment. *Environ. Model. Softw.* **2026**, *195*, 106734. [CrossRef]
60. Rivera-Ruiz, J.R.; García-Martínez, J.R.; Martínez-Sánchez, T.; Cruz-Miguel, E.E.; Ramírez-González, L.D.; Barra-Vázquez, O.A.; Odry, Á. Takagi–Sugeno–Kang Fuzzy Inference Tracking Controller for UAV Bicopter. *Symmetry* **2025**, *17*, 759. [CrossRef]
61. UCAR, Center for Science Education. What's in the Air? What's in the Air? | Center for Science Education. Available online: <https://scied.ucar.edu/learning-zone/air-quality/whats-in-the-air> (accessed on 24 September 2025).
62. Wisconsin Department of Health Services. Carbon Dioxide. Carbon Dioxide | Wisconsin Department of Health Services. Available online: <https://www.dhs.wisconsin.gov/chemical/carbondioxide.htm> (accessed on 24 September 2025).
63. Bierwirth, P.N. *Long-Term Carbon Dioxide Toxicity and Climate Change: A Critical Unapprehended Risk for Human Health*; Australian National University: Canberra, Australia, 2025. [CrossRef]
64. Guais, A.; Brand, G.; Jacquot, L.; Karrer, M.; Dukan, S.; Grévillet, G.; Molina, T.; Bonte, J.; Régnier, M.; Schwartz, L. Toxicity of carbon dioxide: A review. *Chem. Res. Toxicol.* **2011**, *24*, 2061–2070. [CrossRef]
65. Allen, J.G.; MacNaughton, P.; Satish, U.; Santanam, S.; Vallarino, J.; Spengler, J.D. Associations of Cognitive Function Scores with Carbon Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments. *Environ. Health Perspect.* **2016**, *124*, 805–812. [CrossRef]
66. NASA. The Carbon Cycle. Available online: <https://earthobservatory.nasa.gov/features/CarbonCycle/> (accessed on 24 September 2025).
67. Ward, P.L. Sulfur dioxide initiates global climate change in four ways. *Thin Solid Film.* **2009**, *517*, 3188–3203. [CrossRef]
68. Minnesota Pollution Control Agency. Sulfur Dioxide. Available online: <https://www.pca.state.mn.us/air/sulfur-dioxide-so2> (accessed on 24 September 2025).
69. Kan, H.; Wong, C.M.; Vichit-Vadakan, N.; Qian, Z.; PAPA Project Teams. Short-term association between sulfur dioxide and daily mortality: The Public Health and Air Pollution in Asia (PAPA) study. *Environ. Res.* **2010**, *110*, 258–264. [CrossRef] [PubMed]
70. Rall, D.P. Review of the health effects of sulfur oxides. *Environ. Health Perspect.* **1974**, *8*, 97–121. [CrossRef] [PubMed]

71. Australian Government. Oxides of Nitrogen. Available online: <https://www.dceew.gov.au/environment/protection/npi/substances/fact-sheets/oxides-nitrogen-0> (accessed on 24 September 2025).
72. Faustini, A.; Rapp, R.; Forastiere, F. Nitrogen dioxide and mortality: Review and meta-analysis of long-term studies. *Eur. Respir. J.* **2014**, *44*, 744–753. [CrossRef]
73. United States Environment Protection Agency. Available online: <https://www.epa.gov/no2-pollution> (accessed on 24 September 2025).
74. AirNow. Air Quality Index (AQI) Basics. Available online: <https://www.airnow.gov/aqi/aqi-basics/> (accessed on 24 September 2025).
75. MQ-135 Gas Sensor (Technical Data). Available online: <https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf> (accessed on 24 September 2025).
76. MG-811, CO₂ Sensor (Technical Data). Available online: <https://ktechnics.com/product/mg811-co2-sensor/> (accessed on 24 September 2025).
77. MQ136 Semiconductor Sensor for Sulfur Dioxide. Available online: <http://www.china-total.com/Product/meter/gas-sensor/MQ136.pdf> (accessed on 24 September 2025).
78. Constantin, A.; Dinculescu, R. UAV development and impact in the power system. In Proceedings of the IEEE 8th International Conference on Modern Power Systems (MPS), Cluj-Napoca, Romania, 21–29 May 2019. [CrossRef]
79. Resat, H.G. Design and Analysis of Novel Hybrid Multi-Objective Optimization Approach for Data-Driven Sustainable Delivery Systems. *IEEE Access* **2020**, *8*, 90280–90293. [CrossRef]
80. Youn, W.; Choi, H.; Cho, A.; Kim, S.; Rhudy, M.B. Accelerometer Fault-Tolerant Model-Aided State Estimation for High-Altitude Long-Endurance UAV. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8539–8553. [CrossRef]
81. Tran, D.-H.; Vu, T.X.; Chatzinotas, S.; ShahbazPanahi, S.; Ottersten, B. Coarse Trajectory Design for Energy Minimization in UAV-Enabled. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9483–9496. [CrossRef]
82. Khan, M.S.; Su, H.; Tang, G. Optimal Tracking Control of Flight Trajectory for Unmanned Aerial Vehicles. In Proceedings of the IEEE 27th International Symposium on Industrial Electronics (ISIE), Cairns, QLD, Australia, 13–15 June 2018. [CrossRef]
83. Bhardwaj, A.; Sam, L.; Akanksha; Martín-Torres, F.J.; Kumar, R. UAVs as remote sensing platform in glaciology: Present applications and future prospects. *Remote Sens. Environ.* **2016**, *175*, 196–204. [CrossRef]
84. Phantom 3 Specification. Available online: <https://www.dji.com/phantom-3-standard/info> (accessed on 24 September 2025).
85. Arduino IDE Homepage. Available online: <https://www.arduino.cc/> (accessed on 24 September 2025).
86. Microsoft Excel. Available online: <https://www.microsoft.com/en/microsoft-365/excel> (accessed on 24 September 2025).
87. Winds on the Beaufort Scale. Available online: <https://en.wind-turbine-models.com/winds> (accessed on 24 September 2025).
88. ToxiRAE (Technical Data). Available online: <https://www.raesystems.com/products/toxirae-3-single-gas-monitor-for-co-h2s> (accessed on 24 September 2025).
89. Beryozkina, S.; Al-Shakhs, N. Real-life Application of the Emission Monitoring System by Using a Drone. In Proceedings of the IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Madrid, Spain, 9–12 June 2020. [CrossRef]
90. AQI Calculator—US EPA Scale Convertor. Available online: <https://aqicn.org/calculator> (accessed on 24 September 2025).
91. SEMTECH, Wireless RF LoRa Transceivers. Available online: <https://www.semtech.com/products/wireless-rf/lora-transceivers/SX1278#download-resources> (accessed on 24 September 2025).
92. Winsen, MH-Z16 NDIR CO₂ Sensor. Available online: <https://www.winsen-sensor.com/sensors/co2-sensor/mh-z16.html> (accessed on 25 October 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

Processing the Sensor Signal in a PI Control System Using an Adaptive Filter Based on Fuzzy Logic

Jarosław Joostberens ¹, Aurelia Rybak ^{1,*} and Aleksandra Rybak ²

¹ Faculty of Mining, Safety Engineering and Industrial Automation, Silesian University of Technology, 44-100 Gliwice, Poland; jaroslaw.joostberens@polsl.pl

² Department of Physical Chemistry and Technology of Polymers, Faculty of Chemistry, Silesian University of Technology, Strzody 7, 44-100 Gliwice, Poland; aleksandra.rybak@polsl.pl

* Correspondence: aurelia.rybak@polsl.pl

Abstract

This paper presents an adaptive fuzzy filter applied to processing a signal from a voltage sensor fed to the input of an object in an automatic temperature control system with a PI controller. (1) The research goal was to develop an algorithm for processing the signal from an RMS voltage sensor, measured at the terminals of a heating element in a temperature control system with a PI controller, in a way that ensures good dynamic properties while maintaining an appropriate level of accuracy. (2) The paper presents a method for designing an adaptive fuzzy filter by synthesizing a first-order low-pass infinite impulse response (IIR) filter and a fuzzy model of the dependence of this filter parameter value on the modulus of the derivative of the measured quantity. The application of a model with a symmetric input and output structure and a modified fuzzy model with asymmetry resulting from the uneven distribution of modal values of singleton fuzzy sets at the output was shown. The innovation in the proposed solution is the use of a signal from a PI controller to determine the derivative module of the measured quantity and, using a fuzzy model, linking its instantaneous value with a digital filter parameter in the measurement chain with a sensor monitoring the signal at the input of the controlled object. It is demonstrated that the signal generated by the PI controller can be used in a control system to continuously determine the modulus of the time derivative of the signal measured at the input of the controlled object, also indicating the limitations of this method. The signal from the PI controller can also be used to select filter parameters. In such a situation, it can be treated as a reference signal representing the useful signal. The mean square error (*MSE*) was adopted as the criterion for matching the signal at the filter output to the reference signal. (3) Based on a comparative analysis of the results of using an adaptive fuzzy filter with a classic first-order IIR filter with an optimal parameter in the *MSE* sense, it was found that using a fuzzy filter yields better results, regardless of the structure of the fuzzy model used (symmetric or asymmetric). (4) The paper demonstrates that in the tested temperature control system, introducing a simple fuzzy model with one input characterized by three fuzzy sets, relating the modulus of the derivative of the signal developed by the PI controller to the value of the first-order IIR filter parameter, into the voltage sensor signal-processing algorithm gave significantly better results than using a first-order IIR filter with a constant optimal parameter in terms of *MSE*. The best results were obtained using a fuzzy model in which an intentional asymmetry in the modal values of the output fuzzy sets was introduced.

Keywords: fuzzy logic; infinite impulse response filter; temperature control system; PI controller

1. Introduction

In automatic control systems for many processes, a PI controller is often used. If the automatic control system is presented as in Figure 1, the controller's transfer function can be expressed as follows:

$$G_C(s) = k_p \times \left(1 + \frac{1}{T_I s} \right) \quad (1)$$

where

k_p —proportional term gain;

T_I —integration constant.

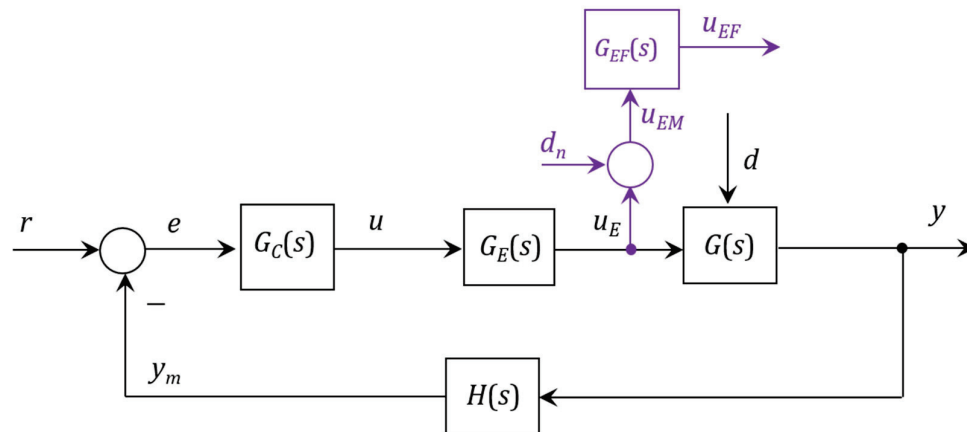


Figure 1. Block diagram of a linear automatic control system with the measuring path (purple color) of the input signal of the control object.

In an automatic control system, there is negative feedback from the controlled variable y processed by the measuring system $H(s)$. The measured variable y_m is compared to the setpoint r . The controller $G_C(s)$, based on the error signal $e(t)$, i.e., the difference between the signals r and $y_m(t)$, generates a control signal $u(t)$ that reduces the controlled signal to the setpoint level. Typically, the signal from the controller does not have sufficient energy to control the object, described in Figure 1 by the transfer function $G(s)$. In such a case, the amplification of signal u is provided by an final control element, which, depending on u , controls the flow of energy or mass to the controlled object. In real systems, the object $G(s)$ is affected by disturbances d , generally presented in Figure 1. The block diagram in Figure 1 refers to linear systems. However, we are dealing with nonlinear systems, primarily due to the nonlinearities inherent in the object and final control element. If the system can be linearized within its intended operating range, the diagram in Figure 1 will be appropriate.

As shown in Figure 1, the system measures the regulated signal y using a measuring and processing system with transmittance $H(s)$, which includes a sensor and a processing system whose task is to convert the signal y to the standard supported by the controller and, if necessary, filter it. In addition to the signal y , the signal at the output of the final control element u_E is often measured in industrial systems.

In automatic control systems, an actuator controls the flow of energy or mass to a facility. The instantaneous value of the u_E signal at the actuator's output should be linearly related to the signal generated by the controller. This will ensure proper operation of the control system. Monitoring the signal directly applied to the u_E control facility input is essential in this context for diagnostic purposes, as it allows for the assessment of potential technical irregularities in the control system's operation. Furthermore, in some cases, if the signal applied to the u_E facility input may be subject to significant disturbances, its measurement can be used to shorten the control system's response time to this disturbance, for example, by using a closed-open system. Monitoring the u_E signal, i.e., the signal

applied to the facility input, can also be used to assess energy consumption or for energy balance purposes.

Monitoring the signal u_E , i.e., the signal fed to the input of the object, can be used to assess energy consumption or for balancing purposes, as well as to check the correct operation of the system. The signal u_{EM} at the output of the sensor of quantity u_E may be significantly affected by noise d_n . In such a situation, the measuring system is expected to separate the useful signal component from the measured signal u_{EM} throughout the entire operating period of the system. This means that the measuring system should be characterized by good dynamic properties while maintaining accuracy at an appropriate level so that the signal u_{FE} represented the waveform of the useful component of the signal measured u_E . For many years, fuzzy logic has been used in automatic control systems of many processes. Many works have been written in this field [1–4]. It can also be used in signal processing. In practice, industrial automation engineers typically use the simplest finite-intensity (FIR) or infinite-intensity (IIR) filters, i.e., a moving-average filter or a first-order IIR filter, respectively. The parameters of these filters are typically selected with an arbitrary cutoff frequency based on the expert's professional experience. Due to the phase shift introduced, these simple filters are a compromise solution, maintaining accuracy at the expense of degrading the dynamic properties of the measurement system, which is often insufficient. Therefore, there is a need and opportunity to supplement a simple digital low-pass filter with a fuzzy model and use the signal generated by the PI controller to implement an adaptive form of this filter, in order to significantly improve the dynamic properties of the measurement system while maintaining its accuracy at an appropriate level. In the proposed solution, the signal generated by the PI controller is treated as a representation of the derivative of the signal at the u_E object's input, thus eliminating the difficulties of determining the derivative of a noisy signal on-the-fly. In this sense, the proposed solution is original and represents a novelty in signal processing in a control system.

This paper presents an adaptive fuzzy filter that performs signal processing using a first-order, low-pass infinite impulse response (IIR) filter with a parameter dependent on the control signal derivative modulus u , representing the time-varying useful signal u_E . A method for designing an adaptive fuzzy filter is demonstrated, and the filter's performance in a linearized temperature control system with an electrically powered heating element is tested. The results of using the adaptive fuzzy filter are compared with the performance of a classic first-order, low-pass IIR filter. Therefore, for differentiation purposes, the output signal of the fuzzy filter is designated u_{EFF} , and that of the first-order IIR as u_{EF} .

2. Adaptive Fuzzy Filter Used for Monitoring the Input Signal of the Control Object in a Control System with a PI Controller

2.1. First-Order IIR Low-Pass Filter

A low-pass digital filter in the form of a first-order recursive IIR filter is often used to process the sensor signal. Referring to the symbols in Figure 1, this filter, used to process the u_{EM} measurement signal, can be described by the equation [5]:

$$u_{EF}[n] = a \times u_{EF}[n-1] + b \times u_{EM}[n] \quad (2)$$

where

a —filter parameter with a value in the range $[0, 1]$;

b —filter parameter $b = 1 - a$;

n —sampling step with constant period T_s ;

u_{EM} —measurement signal.

2.2. Adaptive Fuzzy Filter Based on First-Order Recursive IIR Filter

Analyzing the above equation it can be easily noticed that for large values of parameter a (close to one) there will be a strong damping of the rapidly varying noise component, which simultaneously results in a deterioration of the dynamic properties of the measurement system (measurement path of the sensor with the filter). A small value of a leads to better dynamic properties of the measurement system at the expense of increased fluctuations in the filter output signal. This means that in the case of significant signal variability u_E , e.g., of a step nature, a too high value of a will result in a significant dynamic error (signal changes will be damped). On the other hand, a too low value of a will result in a reduction in the dynamic error, but with an increase in the statistical error. This leads to the conclusion that in the case of sudden signal changes u_E , the value of the filter parameter a should be small, and at moments close to a steady state it should be large. Therefore, the value of parameter a should be linked to the derivative of the filtered signal u_E . It should be noted that du_E/dt can have positive or negative instantaneous values. The value of parameter a should therefore depend on the value of the derivative, not its sign, which can be written as:

$$a = f\left(\left|\frac{du_E}{dt}\right|\right) \quad (3)$$

The basic procedure of numerical differentiation is the first difference, the equation of which, with respect to Formula (3) and the block diagram in Figure 1, can be written [6]:

$$\frac{du_{EM}}{dt} \cong \frac{1}{T_s} \times (u_{EM}[n] - u_{EM}[n-1]) \quad (4)$$

The difficulty of numerically determining the derivative of the measurement signal u_{EM} lies in the fact that its interference is also differentiated. It should be noted that in the control system shown in Figure 1, the transformation of the measured signal u_{EM} must be performed continuously. Therefore, to use Equation (3) for adaptive implementation of filter (2), the time derivative of the signal u_{EM} must also be determined continuously, every sampling period. As a result, using Formula (4) for a noisy signal becomes useless, because the first difference in the signal u_{EM} determined according to Equation (4) would require additional smoothing to reduce the differentiated noise, which complicates the signal-processing algorithm. There are many procedures for determining the derivative of the signal [7–13], but these usually introduce a phase or time shift relative to the signal u_E , which can significantly limit their use in automatic control systems. Analyzing the block diagram in Figure 1 in terms of the useful signal u_E , it can be seen that it is the output of the transfer function $G_E(s)$, which describes the properties of the final control element. The input signal of this transfer function is the control signal u generated by the controller. In a linear control system, there is therefore a relationship between them:

$$U_E(s) = G_E(s) \times U(s) \quad (5)$$

where

$U_E(s)$ —transform of the signal at the input of the control object—useful signal;

$U(s)$ —transform of the signal generated by the regulator.

In a situation where the final control element is a linear proportional element that amplifies the signal u , Equation (5) can be written in the time domain as:

$$u_E(t) = k_E \times u(t) \quad (6)$$

where k_E is the gain of the final control element, i.e., $G_E(s) = k$.

Taking into account the relations (4) and (6), the derivative of the signal u_E , observed discretely every sampling period, can be determined using the relations:

$$\frac{du_E}{dt} \cong k_E \times \frac{u[n] - u[n-1]}{T_s} \quad (7)$$

Due to the fact that the control signal u in the control system with the PI controller of Equation (1) does not contain a fast-changing component (no D term), the use of the first difference in this signal to determine the current derivative of the signal u_E , fed to the input of the controlled object, is justified, because the signal u_E is a rescaling of the signal u .

The relationship between the value of parameter a and the modulus of the time derivative of the signal u_E was defined above qualitatively. A convenient way of qualitative assessment is to use the fuzzy set theory introduced by [14]. In such a case, the adaptation of the value of parameter a or b of the filter with Equation (2) depending on $|du/dt|$ can be presented in the form of three simple rules:

R1: If $|du/dt| = S$ then $b = S(a = L)$.

R2: If $|du/dt| = M$ then $b = M(a = M)$.

R3: If $|du/dt| = L$ then $a = L(b = S)$.

In these rules, S denotes a small value, M denotes a medium value, and the symbol L denotes a large value. Due to the relationship between the parameters a and b of the filter (2), where $b = 1 - a$, rules R1–R3 can be presented in two ways, i.e., with respect to a or b . In this case, it is more convenient to refer to parameter b . In such a situation, the relationship (3) transformed with respect to parameter b can be relatively easily implemented using fuzzy modeling. The structure and idea of using a fuzzy model to implement adaptive filtering of the signal fed to the input of the controlled object in a system with a PI controller are shown in Figure 2.

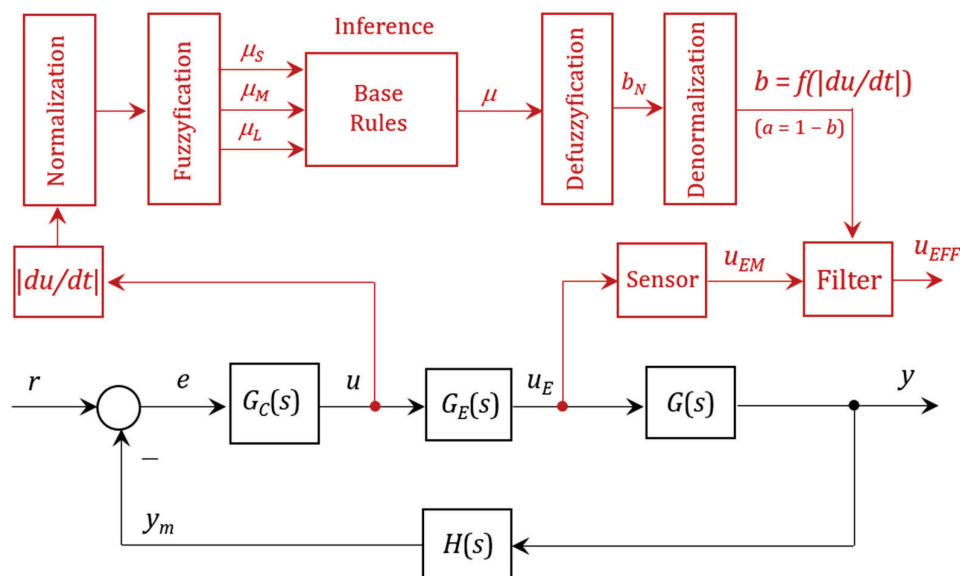


Figure 2. Block diagram of a control system with an adaptive fuzzy filter (red color) used to monitor the input signal of the control object.

Because in the fuzzy model the input signal is normalized in the range $[-1, 1]$, and in the control system between signals u and u_E at any time the relation (6) holds, the currently determined $|du/dt|$ can be directly used as an input quantity of the fuzzy model without

the need to take into account the amplification factor k_E , which is shown in Figure 2. Then, the relation (3) can be replaced by the following relation:

$$b = f\left(\left|\frac{du}{dt}\right|\right) \cong f\left(\left|\frac{\Delta u}{T_s}\right|\right) \quad (8)$$

The proposed fuzzy model, whose structure is shown in Figure 2, is a single-input, single-output model with three fuzzy input sets.

Triangular functions were assumed as membership functions for the input fuzzy sets, while Singleton functions were assumed as membership functions for the output fuzzy sets [15]. The graphical representations of these input and output sets are presented in Figure 3.

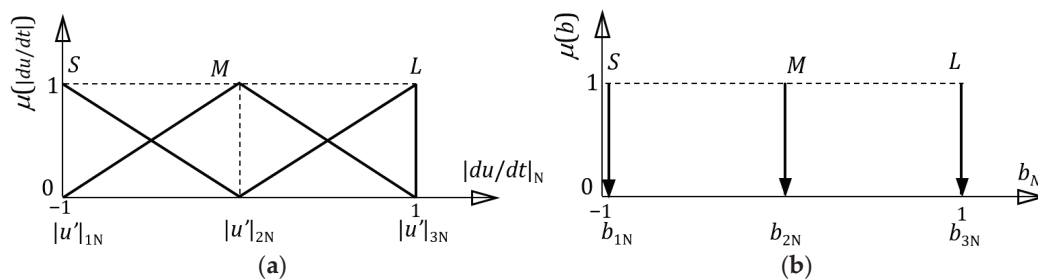


Figure 3. Membership functions of fuzzy sets (a) inputs and (b) outputs of the fuzzy dependency model (8).

The idea of using fuzzy logic for u_{EM} signal filtration (Figure 2) is to supplement one of the simplest digital filters with a simple fuzzy model to develop its adaptive form. The proposed first-order adaptive IRR filter is based on the assumption that such a relatively simple modification will lead to improved dynamic properties of the measurement system without the need for significantly more complex algorithms or signal-processing systems, which justifies the adoption of a small number of model input and output fuzzy sets and membership function shapes.

As shown in Figure 2, the signal u generated by the controller is continuously differentiated based on the equation

$$\frac{du}{dt} \cong \frac{\Delta u}{T_s} = \frac{u[n] - u[n-1]}{T_s} \quad (9)$$

with the determined module of instantaneous values. The instantaneous values $|\Delta u/T_s|$ are normalized in the range $[-1, 1]$. These values are subject to fuzzification, which consists of determining the degree of membership of the input $\mu(|du/dt|)_{i=1,2,3}$ to fuzzy sets. In the Inference block, based on the degrees of membership $\mu(|du/dt|)_{i=1,2,3}$, the resulting membership function $\mu(b)$ is calculated. The most important element of the Inference block is the rule base, which represents qualitative knowledge about the modeled dependency $b = f(|du/dt|)$ in the form of a set of three rules: R1, R2, R3. At a given instant nT_s , the calculation of the crisp output value of the model a_N is based on the determined resulting membership function $\mu(a)$, which is the result of providing the crisp input values of the model $|Du[n]/T_s|$. This stage is represented by the Defuzzification block in Figure 2. The Height Method was used to accomplish this task. The continuously calculated sharp values of the b_N model output have a standardized form. For this reason, in the block determining the current value of the b filter model parameter (2) requires performing an operation inverse to normalization (the Denormalization block in Figure 2). Compared to existing solutions, the proposed adaptive fuzzy filter uses the derivative of the signal u generated

by the PI controller to change its parameter. This signal is treated as a good representation of the derivative of the noisy filtered signal fed to the input of the control system. Changes to the filter parameter are obtained continuously, depending on the value of the derivative modulus of the control signal, which in this situation is used as the input signal to the fuzzy model of the relationship $b = f(|du/dt|)$. This is a new approach to signal processing, particularly using fuzzy logic.

2.3. Selection of Fuzzy Model Parameters as an Element of an Adaptive Low-Pass Filter

Selection of parameters of the fuzzy model $b = f(|\Delta u/T_s|)$ comes down to determining the modal values of the input and output fuzzy sets. For the assumed fuzzy model with one input characterized by three input fuzzy sets, the number of rules is three and the number of parameters is six [16]. Therefore, the values b_1, b_2, b_3 (Figure 3b) and $|u'|_1, |u'|_2, |u'|_3$ (Figure 3a) should be determined. The modal values of the input fuzzy sets $|u'|_1, |u'|_2, |u'|_3$ can be specified based on the $|du/dt|$ course, determined using the recording of the control signal u during normal operation of the control system. Then, the variability of the control signal can be determined for a given control system, both in the situation of changes in the setpoint value r and in the presence of disturbances acting on the controlled object. In the situation of the largest changes in the signal u , the value of $|\Delta u/T_s|$ will be the highest. Such a value can then be assigned to the modal value $|u'|_3$. If $|\Delta u/T_s|$ is the result of abrupt changes in the setpoint r , it can reach very high values, many times greater than in the case of other causes of changes in the signal u . The value of $|u'|_3$ should then be reduced to take into account the range of variation in $|\Delta u/T_s|$ caused by other reasons, e.g., the impact of disturbances d on the process object. In the case of automatic stabilization systems, a steady state may also occur, or more precisely, quasi-steady state, in which the control error e will be characterized by small variability, and therefore an almost constant value (in the case of PI control approximately equal to zero). In such a state, the control signal u will also be almost constant in time, so $|\Delta u/T_s|$ will have a value close to zero. It follows that the modal value $|u'|_1$ of the input fuzzy set can be assumed equal to zero in such a case. However, the modal value $|u'|_2$ is conveniently defined as the middle value of the section between $|u'|_1$ and $|u'|_3$ from Figure 3a. The modal values of the output fuzzy sets will determine the range of variability of the parameter b of the adaptive filter with Equation (2).

Since the control signal u is an approximate, scaled representation of the useful component u_E of the measured signal u_{EM} , it can be used to select the modal values of the output fuzzy sets. Then, as a criterion for matching the signal at the output of the u_{EFF} filter to the useful component u_E , which is assumed to be proportional to the signal u (Equation (6)), the mean square error MSE can be assumed.

It should be noted that in the event of a step change in the control signal, i.e., when the value of $|\Delta u/T_s|$ is large, the filter should also ensure a step change in the magnitude of u_{EM} . Then, the modal value b_3 can be assumed equal to one. The remaining modal values b_1 and b_2 can be determined by minimizing the criterion value for a certain operating interval of the control system. This operation consists of recording the characteristic time interval of u_{EM} for a given operating interval of the control system. Then, b_1 and b_2 are selected iteratively in a program loop, determining the MSE criterion value for each pair of modal values, with b_3 equal to one. This can be performed in two ways:

- by selecting the modal value b_1 and calculating the value b_2 as the midpoint between b_1 and b_3 , i.e., $b_2 = (b_1 + b_3)/2$, thus obtaining a symmetric structure of the fuzzy model;
- by iteratively determining both modal values, i.e., b_1 and b_2 .

The values of b_1 and b_2 for which the MSE is minimal should be considered correct.

For the purposes of comparative analysis of the results of using a classic first-order IIR filter with its adaptive fuzzy version, criterion MSE can also be used. The filter that better separates the useful component u_E from the measurement signal u_{EM} should be considered the one for which the MSE value is lower.

As an auxiliary criterion, one can also use the evaluation of residuals, defined as:

$$\varepsilon[i] = u_{EM}[i] - u_{Ej}[i] \quad (10)$$

3. Results of Bench Tests

3.1. Test Stand

The schematic diagram of the test stand is shown in Figure 4. The control system consists of the following elements:

- 1—24 V industrial power supply;
- 2—F&F voltage sensor type AC-1I (true RMS);
- 3—0–10 V to 4–20 mA standard converter;
- 4—SSR relay type SAVP2240 controlled by analog standard 4–20 mA;
- 5—heating element with power $P_N = 40$ W and rated voltage $U_N = 230$ VAC;
- 6—temperature sensor—K-type thermocouple;
- 7—NI cRIO-9074 industrial controller with analog input modules: NI 9219 (for temperature measurement); NI 9203 for measuring the voltage value at the terminals of the heating element at u_{EM} ; and an NI 9263 analog output module (PI controller output) and NI 9474 digital outputs;
- 8—PC;
- 9—two-state SSR relay type SSR-4028ZD3;
- 10—Sunon fan type DP200A.

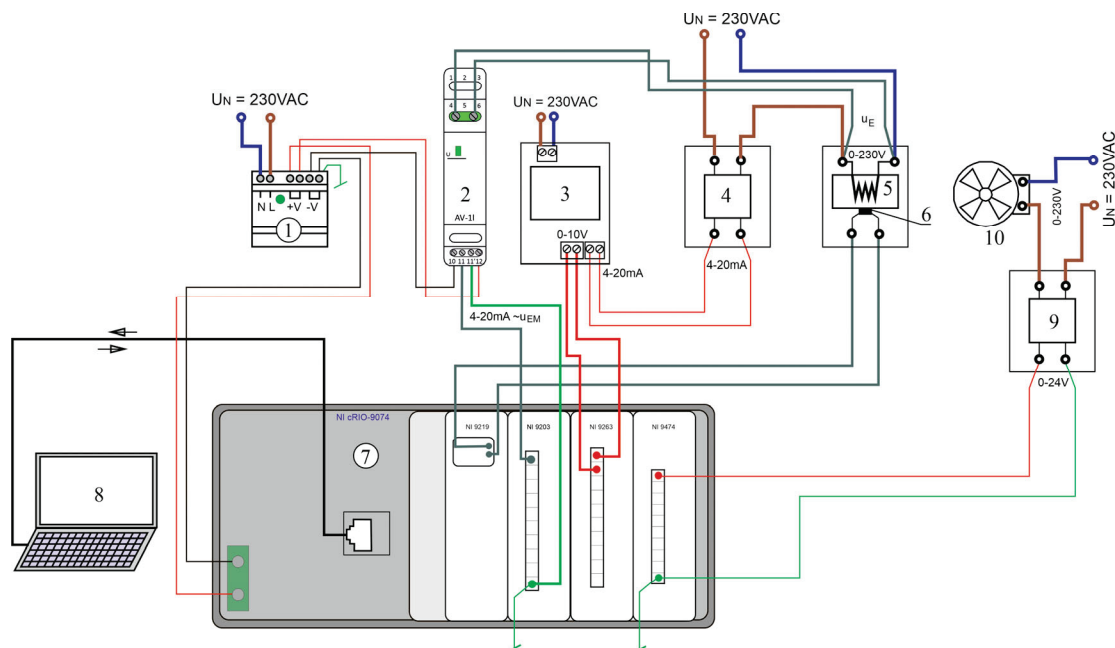


Figure 4. Schematic diagram of a temperature control system with an industrial controller 1—power supply; 2—voltage sensor; 3—0–10 V to 4–20 mA standard converter; 4—final control element; 5—heating element (controlled object); 6—temperature sensor; 7—industrial controller; 8—PC; 9—SSR relay for fan control; 10—fan.

In the system, the industrial controller performs the function of the PI regulator (3). It also provides recording and processing of the u_{EM} signal at the voltage sensor

output (2, Figure 4) and the regulated signal (temperature). In the system, the heating element (Figure 4) is the controlled object, whose input is supplied with a supply voltage corresponding to the control signal generated by the PI controller, amplified by an SSR relay. The SSR relay (Figure 4) serves as a final control element, shown in the block diagrams (Figures 1 and 2) as the operator transfer function $G_E(s)$. It allows for the conversion of 230 VAC mains voltage to values from 0 to nearly 230 VAC, depending on the value of the 0–20 mA input signal. In the control system, the electric voltage is applied to the input (terminals) of the heating element (3, Figure 4) via an SSR relay (Figure 4) depending on the value of the signal u at the analog output of the controller (NI 9263 module), developed by the PI algorithm. A sensor (Figure 4) from F&F, type AV-1I (true RMS), is used to measure the voltage at the heating element terminals. The voltage sensor (2, Figure 4) uses the 4–20 mA standard as the output signal. The AV-1I sensor symbol shown in Figure 4 contains the standard terminal markings (1–12) of this device [17]. As can be seen in Figure 4, the sensor's output signal is connected to the NI 9203 analog input. This signal is subject to noise. The measuring output signal of the F&F AV-1I sensor is converted from the 4–20 mA analog standard value to the effective voltage value according to the relationship [17]:

$$u_{EM}(t) = (25 \times i_{EM}(t) - 0.1) \times 10^3 \quad (11)$$

where

i_{EM} —instantaneous value of the 4–20 mA standard current expressed in mA proportional to the value of the measured quantity u_{EM} ;

u_{EM} —signal at the sensor output, converted in accordance with (11) into the value of the measured quantity, identical to the signals from the diagrams shown in Figures 2 and 3.

The elements of the measurement chain for which the effectiveness of the adaptive fuzzy filter was tested are: an SSR relay (Figure 4), i.e., an final control element, an F&F AV-1I voltage sensor and an industrial controller (Figure 4).

In the temperature control system, external interference can be achieved by switching on the fan (8, Figure 4) using the controller's digital output module and a voltage-controlled two-state SSR relay (7, Figure 4).

3.2. Test Conditions

The system uses a PI controller with the operator transfer function (1) and settings equal to $k_p = 2.002$, $T_I = 1.020$ s. The control object in the form of a heating element in the operating range was approximated by a model with the structure of a first-order inertial element:

$$G(s) = \frac{0.83 \times e^{-5.6s}}{153.9s + 1} \quad (12)$$

The final control element in the form of an SSR relay (Figure 4) in the control system can be treated as a static element with a strongly nonlinear characteristic. Therefore, it was linearized by introducing series with the final control element (Figure 4) an additional nonlinear element with a static characteristic compensating for the nonlinearity of the SSR relay (Figure 4). The linearized static characteristic of the final control element was identified using the least squares method. Its sample results are shown in Figure 5. In the experiment, a signal from the 0–10 V standard was fed to the input of the linearized element every 1 V, and the signal was recorded at the terminals of the u_{EM} voltage sensor (2, Figure 4).

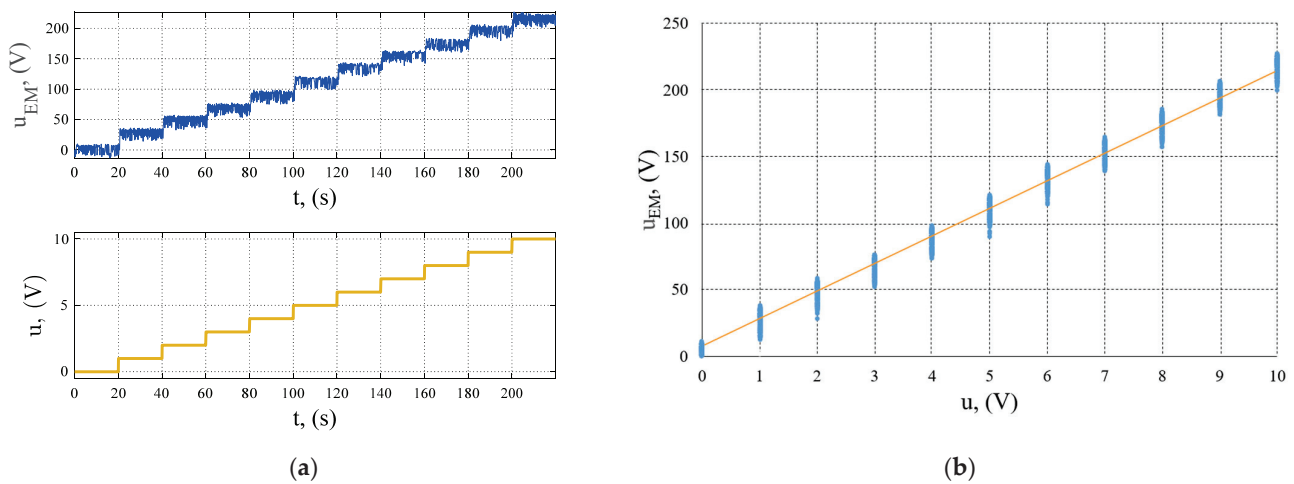


Figure 5. Example results of linearization with an actuator in the form of a relay (4, Figure 4). (a) u_{EM} waveform at the output of the SSR relay as a result of the excitation u (b) static characteristic, blue points—measurement data, yellow line—linear function approximation.

The best fit of the model to the empirical data was obtained for the following relationship:

$$u_E = d_1 \times u + d_0 \quad (13)$$

where

d_0, d_1 —model parameters equal to 9.71 and 20.92, respectively.

Model (13) shows that in the case of the final control element (Figure 4) of the real control system, a modification of the criterion equation for assessing the effectiveness of noise reduction from the measurement signal of the sensor (Figure 4) is required. Due to the fact that in the control system (Figure 4) the final control element (Figure 4) has the characteristics of a static element, the time-domain relationship (13) can be treated as the equation of the reference signal u_{ref} and used in the selection of filter parameters. In such a situation, MSE criterion takes the form:

$$MSE_j = \frac{1}{N} \sum_{i=1}^N \left(u_{Ej}[n] - u_{ref}[n] \right)^2 \cong \frac{1}{N} \sum_{i=1}^N \left(u_{Ej}[n] - (d_1 \times u[n] + d_0) \right)^2 \quad (14)$$

MSE criterion refers to the representation of the useful signal u_E . In real solutions, a reference signal is obtained, which is an empirical model of the useful signal. As can be seen in Formula (14), the ideal dependence (6) is replaced by the empirical dependence (13).

It should be noted that for the selection using the reference signal (criterion (14)) to be effective, it should be carried out in a situation where there are no significant fluctuations in the effective value of the network voltage, which is a disadvantage of this method.

The results of the operation of the automatic control system with a PI controller in the form of waveforms: setpoint value r , temperature y and the control signal developed by the controller u are presented in Figure 6.

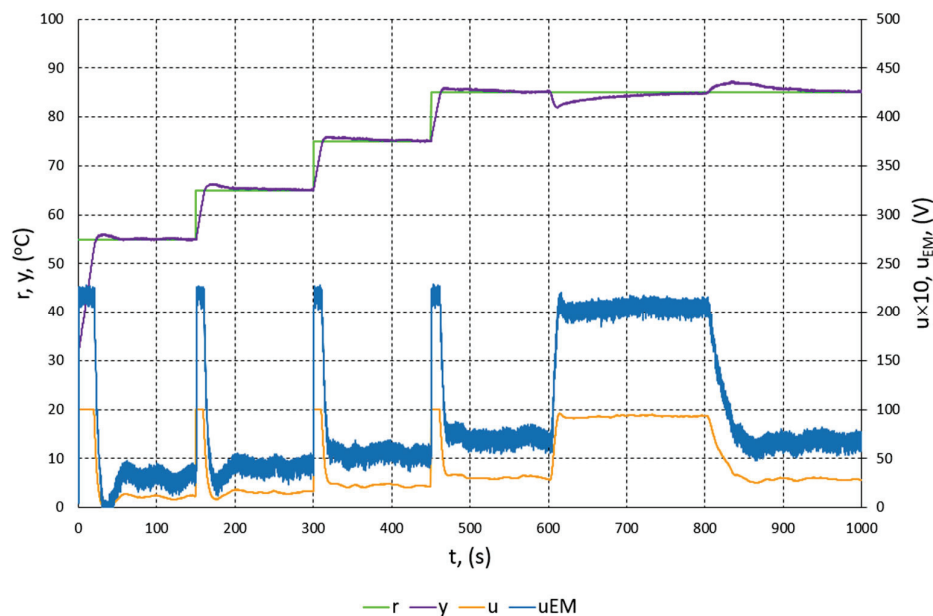


Figure 6. Courses of the setpoint value r , the controlled signal (temperature) y , the control signal u and the effective voltage value u_{EM} at the output of the F&F AC-1I sensor.

As shown in Figure 6, the control system with the PI controller and the selected settings operated correctly—the controlled signal y reached the setpoint r in every situation. The figure also shows that the signal from the RMS voltage sensor is significantly affected by noise. The qualitative similarity of the signals u and u_{EM} is also noticeable. The control system operation time was 1000 s. A sampling period T_s of 0.1 s was assumed, so the total number of samples was 10,000. The control system operation time data from start t_0 to $t_1 = 300$ s were used to select the parameters of the adaptive fuzzy filter and the IIR filter parameter with Equation (2). Subsequent operating intervals of the system were used to verify the correctness of the filters and to comparatively analyze the results obtained as a result of using the classic IIR filter (2) and its adaptive fuzzy variant. The remaining operating time was divided into three intervals. The interval from time $t_2 = 300.1$ s to $t_3 = 600$ s, in which the set point r was changed twice in steps from 65 °C to 75 °C and from 75 °C to 85 °C. The range defined by times $t_4 = 600.1$ s and $t_5 = 800$ s includes the introduction of a significant disturbance d of a step nature (including the fan (Figure 4)). The last stage from time $t_6 = 800.1$ s to $t_7 = 1000$ s includes the operation of the control system after the step disappearance of the disturbance d (including the fan (Figure 4)).

In the case of the first method of selecting parameters b_1, b_2, b_3 , described in Section 2.3, i.e., for the symmetrical structure of the fuzzy model, the value of b_1 was substituted from the range 0.001 to 0.1 with a step of 0.001, and at the same time the modal value of b_2 was determined from the equation $b_2 = (b_1 + b_3)/2$. For such a pair of parameters b_1 and b_2 and with b_3 equal to 1, a fragment of the u_{EM} signal was filtered, and the result was compared with the reference waveform using the MSE criterion. Then, the parameter b_1 was increased by 0.001 and the calculations were repeated, etc. For the fuzzy model with an asymmetric output (the second method), the pair of modal values b_1 and b_2 were selected independently. Knowing the range of the b_2 value changes in the situation of a symmetrical structure of the fuzzy model, the range [0.4, 0.5] of the b_2 parameter was determined for the model with an asymmetric output, from which this modal value was substituted, increasing it in subsequent iterations by 0.001, while b_1 was substituted from the range [0.001, 0.1]. Each time, the first fragment of the u_{EM} signal was filtered and the MSE value was determined. The u_{EM} signal processing for the selection of modal values was performed iteratively for the first fragment of the waveform, i.e., for the time from t_0

to t_1 . In the case of both fuzzy models, for further signal processing, i.e., in the time range from t_1 to t_7 , the parameters b_1 , b_2 , and b_3 for which the MSE value was the lowest were adopted.

3.3. Research Results

In the first stage of the study, noise analysis was performed in the u_{EM} measurement signal. For this purpose, the method given in [18] was used, which involved determining the residuals of the autoregressive process, identified by the least squares method, using the FPE index [19] to determine the process order. The autocorrelation function of the obtained residuals was then examined. Based on the analysis of the autocorrelation function, the hypothesis that the sequence of residuals approximates a white noise sequence with a mean of 0.03 and a variance of 60.7 V^2 cannot be rejected.

In the next stage, the values of $|\Delta u[n]/T_s|$ were determined. It was found that only for step changes in u they reach high values. In the remaining cases, $|\Delta u[n]/T_s|$ did not exceed 2.5 V/s . The modal values of the input fuzzy sets were assumed to be equal to: $|u'|_1 = 0$, $|u'|_2 = 1.25$, $|u'|_3 = 2.5$. Then, for the measurement data obtained for the first interval, i.e., from time t_0 to t_1 , the parameters of the first-order IIR filter with Equation (2) and the adaptive fuzzy filter were selected. The results of the parameter selection of both filters are presented in Figures 7 and 8.

Based on the research, it was found that in the case of the u_{EM} course, for the time interval from t_0 to t_1 , the value of parameter a of the first-order IIR filter with Equation (2), for which the lowest value of criterion (14) was obtained, is 0.475. On the other hand, the minimization of MSE for the adaptive fuzzy filter for the same time interval, with $b_3 = 1$ assumed, for the modal values of the output fuzzy sets with symmetrical distribution $b_1 = 0.067$ and $b_2 = 0.5335$ and with asymmetry $b_1 = 0.073$ and $b_2 = 0.443$ were obtained. Then, the signal processing was performed for both filters in the remaining time intervals and for the entire operation time of the control system. The results are summarized in Table 1 and presented in Figure 9.

Table 1. Test results of the use of a first-order IIR filter and an adaptive fuzzy filter to process the signal from an RMS voltage sensor.

Range	IIR Filter with Equation (2)	Adaptive Fuzzy Filter	
	$a = 0.475$	With a Symmetrical Structure $b_1 = 0.067; b_2 = 0.5335; b_3 = 1.000$	With a Model with an Asymmetric Output $b_1 = 0.073; b_2 = 0.443; b_3 = 1.000$
	MSE	MSE	MSE
t_0-t_1	42.25	24.23	24.17
t_2-t_3	29.78	5.11	4.90
t_4-t_5	16.47	2.97	2.89
t_6-t_7	46.22	5.88	5.97
t_0-t_7	29.73	10.63	10.50

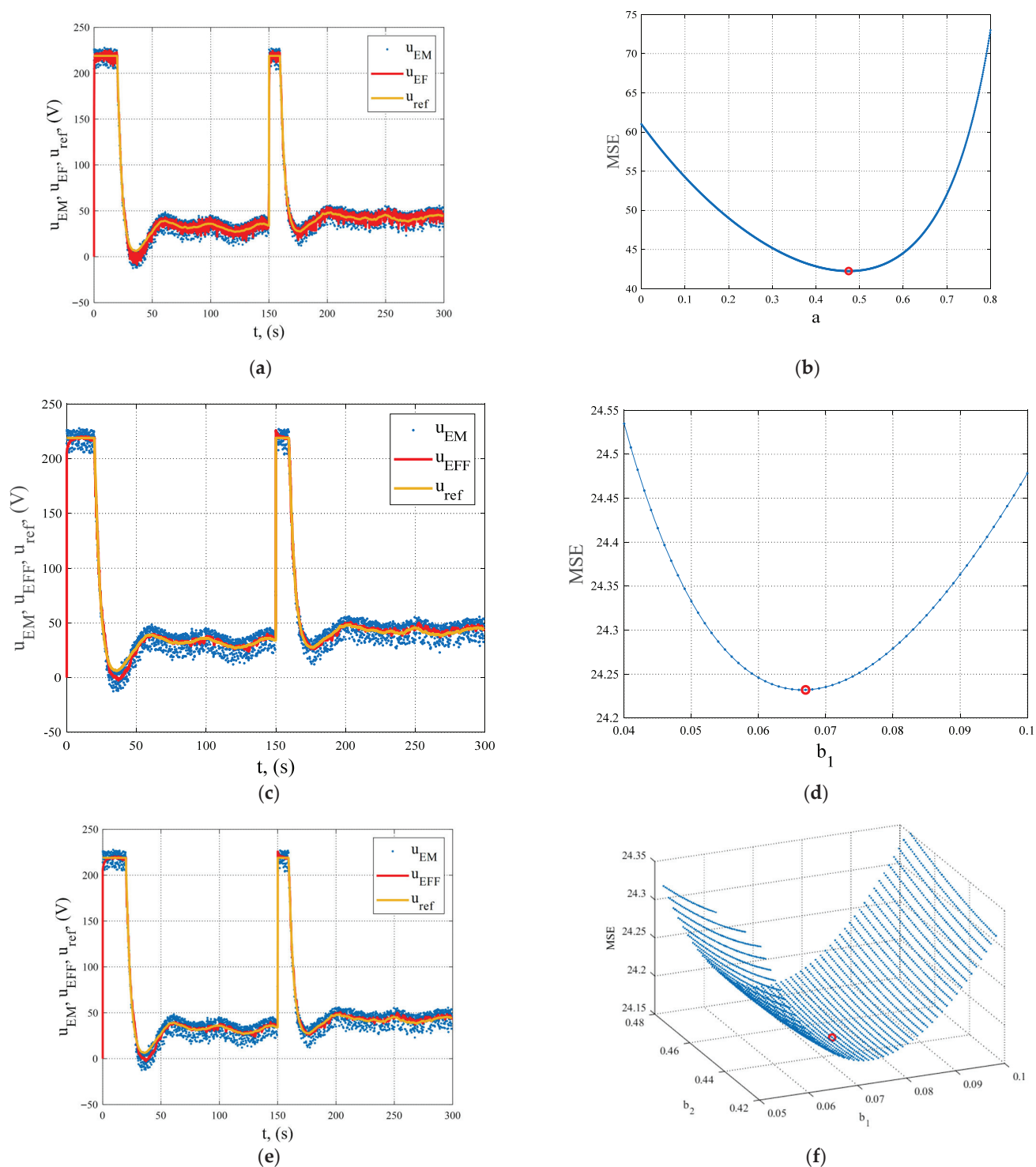


Figure 7. Waveforms of the reference signal u_{ref} and the signals at the filter output against the background of the measurement data u_{EM} (**a,c,e**) and dependencies of the mean square error on the filter parameters (**b,d,f**) determined for the first-order IIR filter (**a,b**), the adaptive fuzzy filter (**c,d**) with a symmetric structure model and an asymmetric output model (**e,f**), red points—optimal filter parameter(s) (minimum criterion (14)).

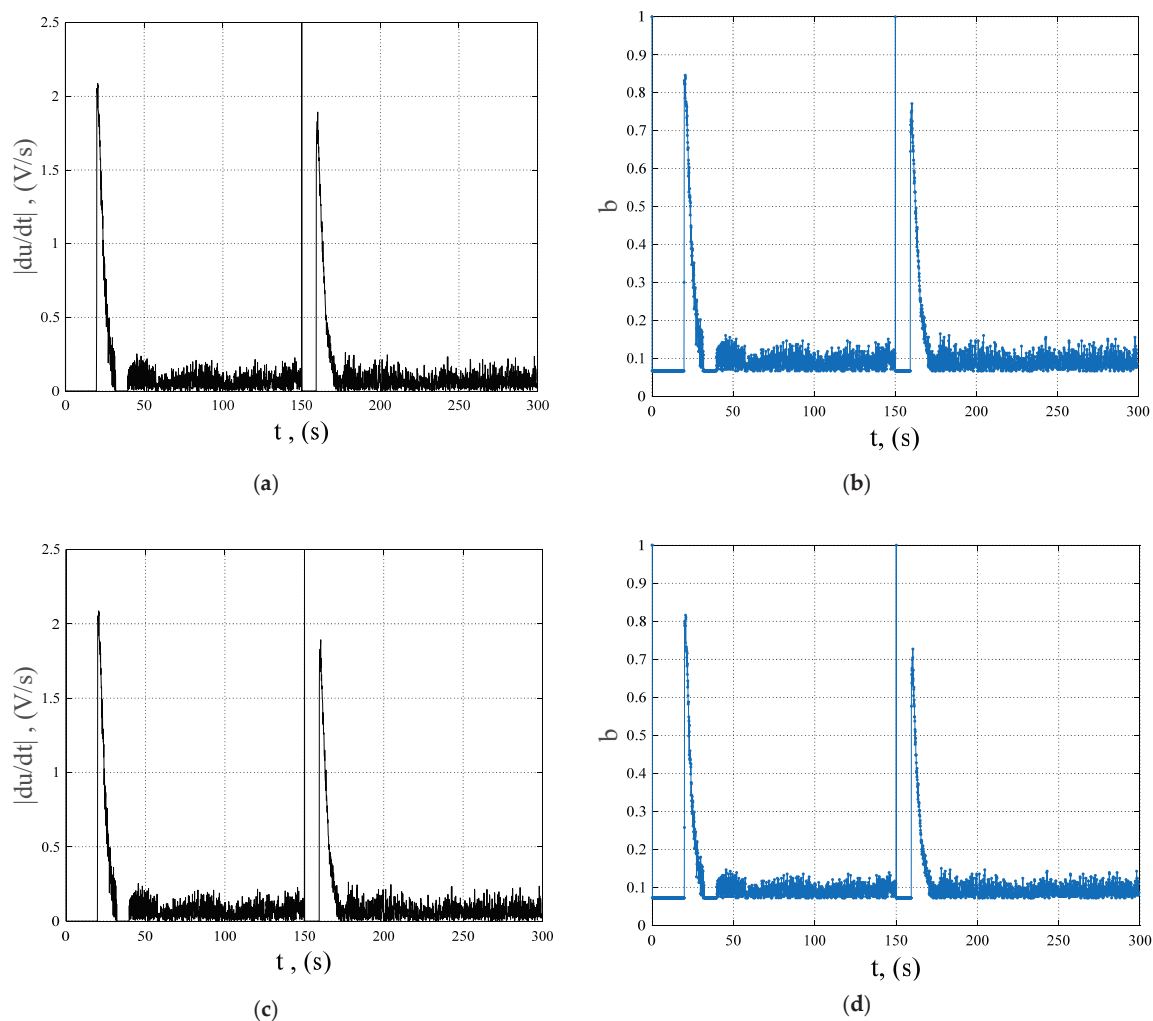


Figure 8. Waveforms of the time derivative modulus of the control signal $|\Delta u/T_s|$ (a,c) and the continuously determined fuzzy filter parameter (b,d) with a symmetric structure (a,b) and an asymmetric model output (c,d).

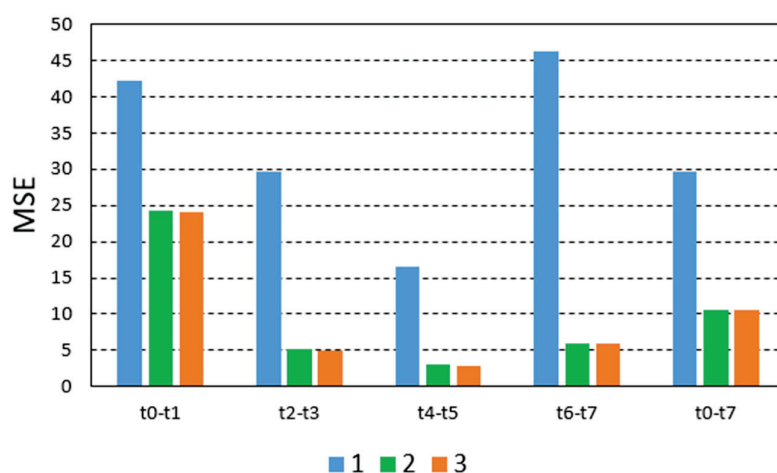


Figure 9. Criterion values (14) obtained for the first-order IIR filter (1) and an adaptive fuzzy filter with a symmetric structure (2) and an asymmetric model output for processing the signal from the RMS voltage sensor, determined for individual time intervals and the entire operating period of the control system.

As shown by the results presented in Table 1 and Figure 9, the use of an adaptive fuzzy filter gives much better results in terms of the SME criterion compared to the first-order IIR filter, both in the considered ranges and for the entire recording time, regardless of the fuzzy model used (model with a symmetric or asymmetric output structure). However, using a filter with a fuzzy model with an asymmetric output results in slightly lower MSE values than in the case of using a fuzzy model with a symmetric structure.

A fragment of the waveforms from Figure 6 in the case of using a first-order IIR filter for two different parameters a and the signal obtained using an adaptive fuzzy filter with a symmetric structure and an asymmetric model output with parameters selected using the criterion (14) are shown in Figure 10.

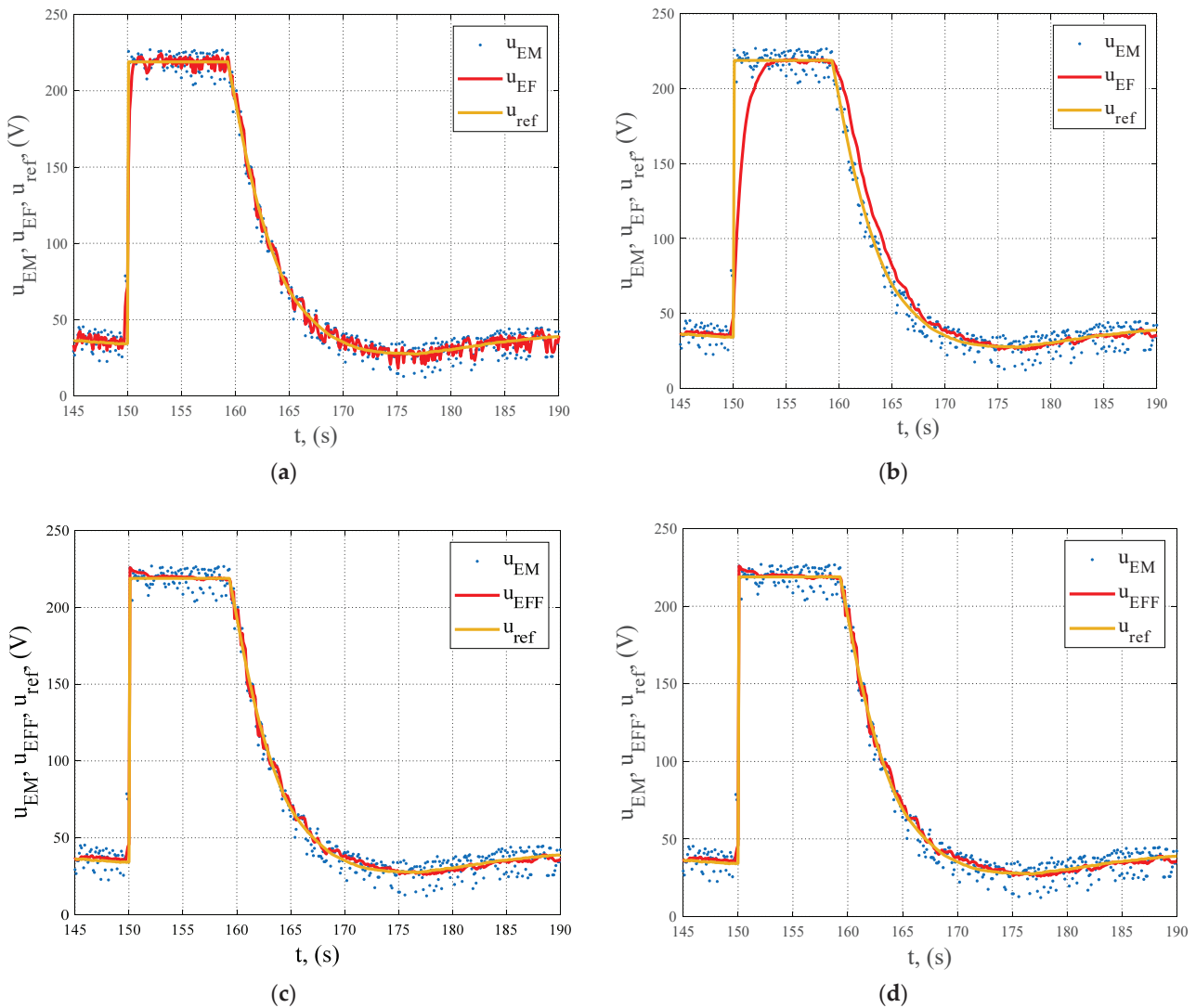


Figure 10. The signal at the output of the first-order IIR filter with parameters: $a = 0.475$ (a) and $a = 0.890$ (b) compared to the results of filtering the signal from the electric voltage sensor using an adaptive fuzzy filter with a symmetric structure (c) and with an asymmetric model output (d).

Figure 11 shows the waveforms of the reference signal u_{ref} at the outputs of the first-order IIR filters u_{EF} and the fuzzy filter u_{EFF} against the background of the measurement signal u_{EM} . The input signal of the fuzzy model $|\Delta u[n]/T_s|$ and the resulting changes in the instantaneous values of parameter b of the adaptive fuzzy filter are shown in Figure 12.

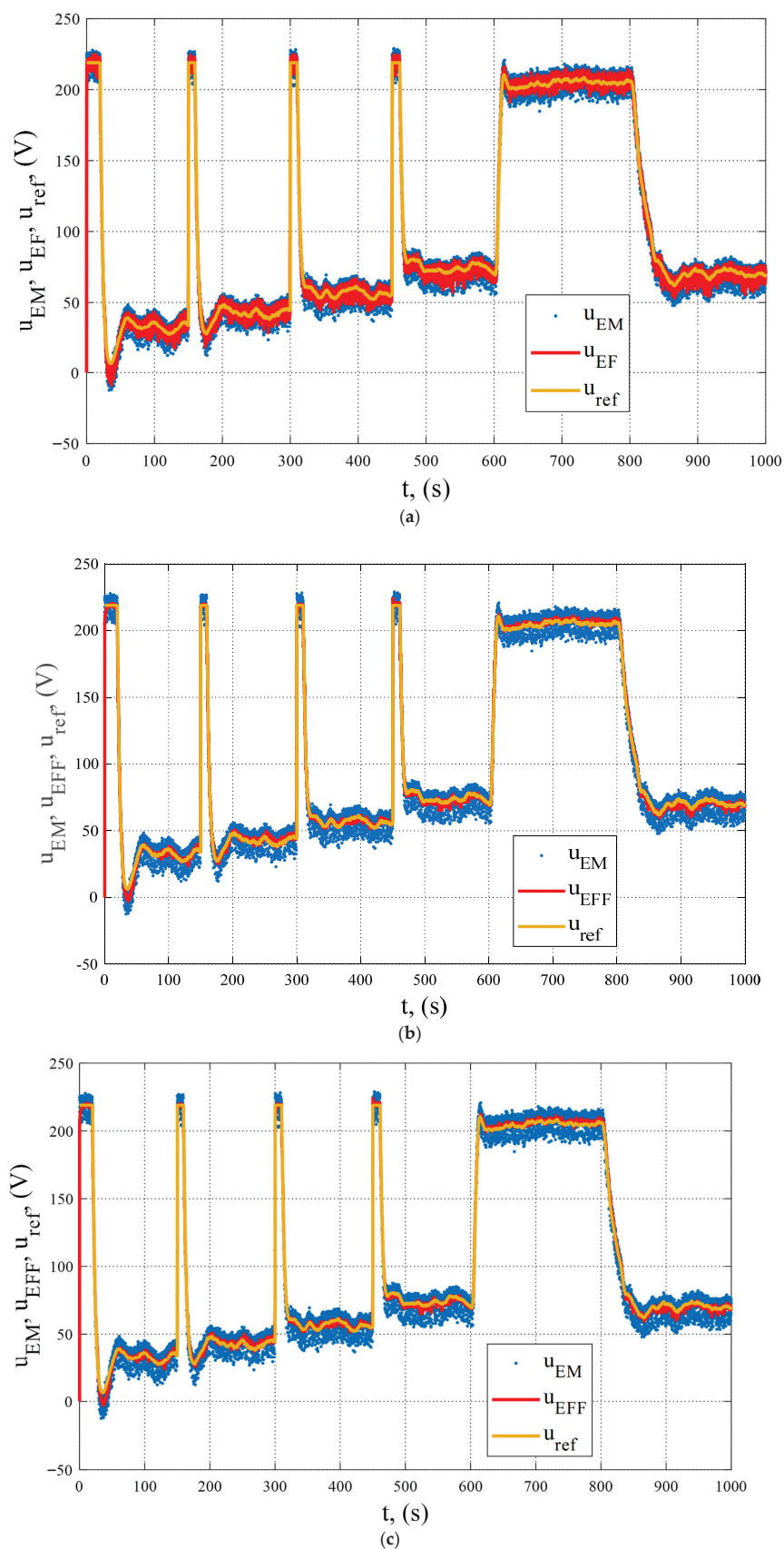


Figure 11. The u_{ref} , u_{EF} and u_{EFF} waveforms against the background of the u_{EM} measurement data in the case of using a first-order IIR filter (a) and an adaptive fuzzy filter with a symmetric structure (b) and with an asymmetric model output (c), and parameters selected in the time range from 0 to 300 s based on the criterion (14) for the remaining operating period of the control system from Figure 4.

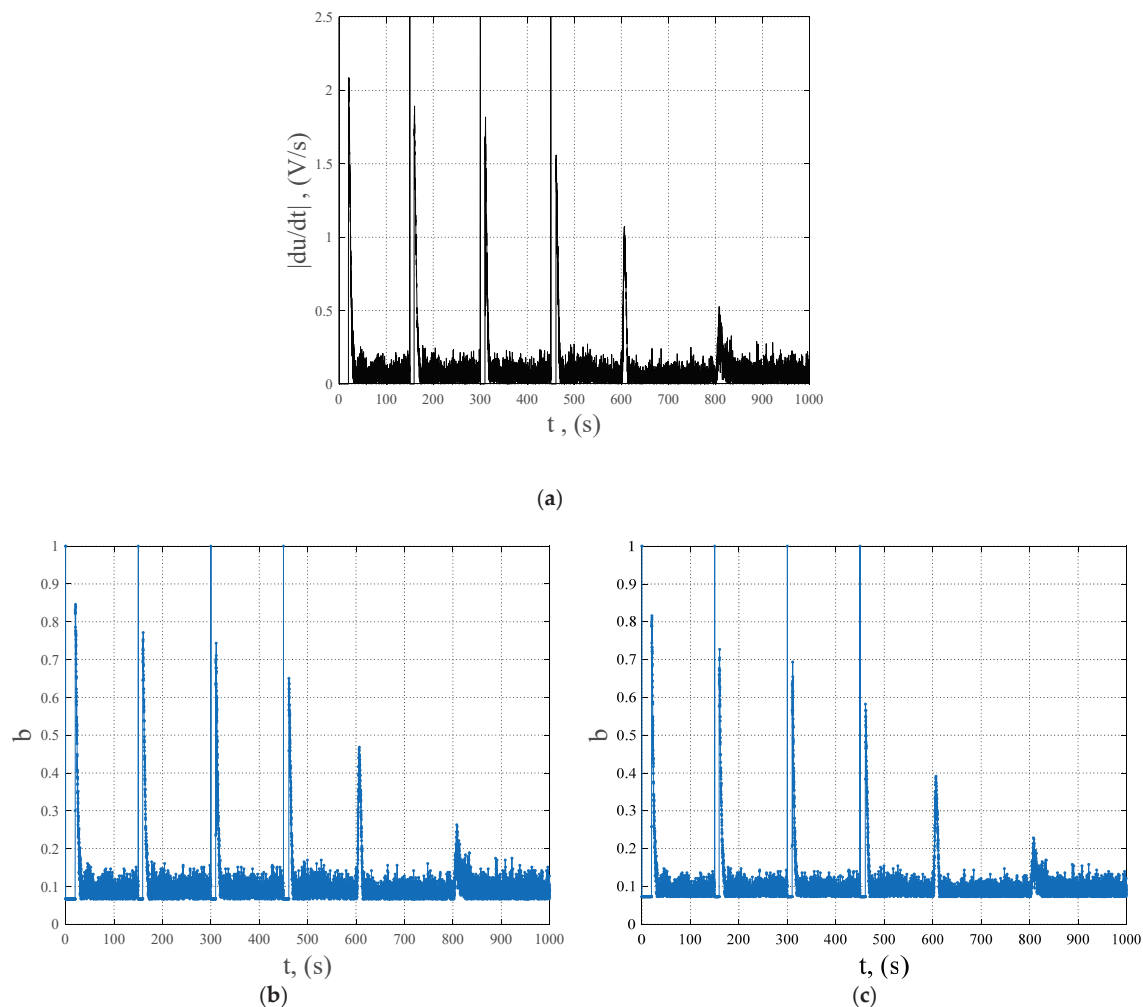


Figure 12. Time derivative modulus of the control signal $|\Delta u/T_s|$ (a) and the fuzzy filter parameter determined in real time, with a symmetric structure (b) and with an asymmetric output model (c).

The model residuals determined for the entire operating range of the control system for the two considered digital low-pass filtration algorithms are shown in Figure 13.

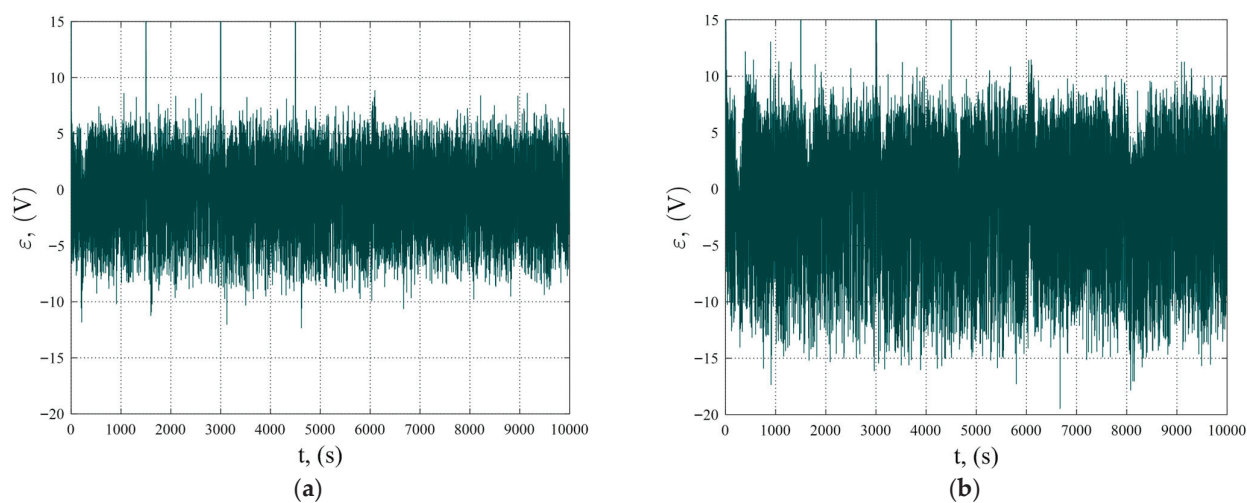


Figure 13. Cont.

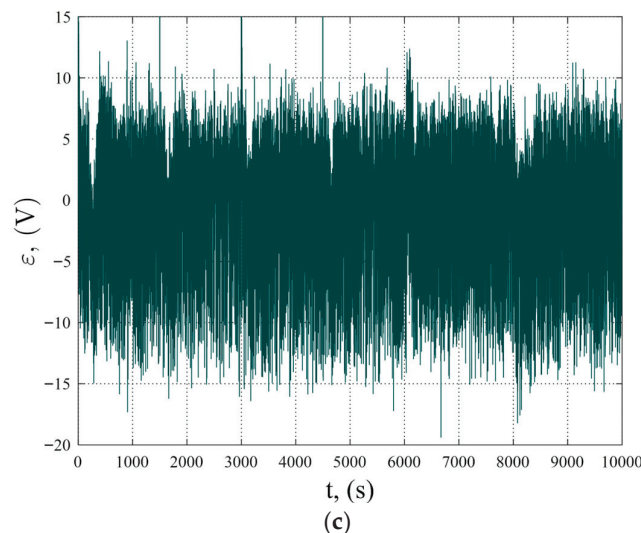


Figure 13. Residuals ε obtained using the first-order IIR filters (a) and the adaptive fuzzy filter with a symmetric structure (b) and with an asymmetric output model (c).

4. Discussion

The results presented in Figures 7, 10 and 11 show significant convergence of the reference signal u_{ref} with the measured data u_E . As shown by the least squares estimation results, the relationship $u_E = f(u)$, describing the static characteristic of the final control element, is a linear function. Therefore, it is reasonable to assume that in the case of linearization of the final control element, the relationship between the useful and control signals u can be used to select filter parameters. This allows the introduction of the reference signal u_{ref} based on relationship (13).

The filter parameters: The first-order IIR and the adaptive fuzzy filter with a symmetric structure and an asymmetric output model were used for signal processing during the remaining operation time of this system. For this first range, the value of criterion (14) obtained with the well-defined parameters of the adaptive fuzzy filter with an asymmetric output was nearly 1.75 times lower than the result obtained using a first-order IIR filter with the optimal MSE parameter. In this range, a slightly worse MSE result was obtained with respect to the adaptive fuzzy filter with an asymmetric output. Based on a comparative analysis of the results summarized in Table 1 it can be concluded that the use of an adaptive fuzzy filter produces significantly better results than the use of a first-order IIR filter. This conclusion is valid for both the adaptive filter with a symmetric structure and the model with an asymmetric output. The values of criterion (14) for the control system operating ranges: t_2-t_3 , t_4-t_5 and t_6-t_7 determined using an adaptive fuzzy filter are several times (from 5.5 to 7.9) lower than the corresponding results obtained using a first-order IIR filter with $a = 0.475$. The use of the entire operating time of the control system also showed that the signal-processing algorithm using fuzzy logic allows for the separation of the useful component from the noisy signal to a much better extent than the first-order IIR filter. For the entire operating period of the control system, the value of criterion (14), determined using an adaptive fuzzy filter with an asymmetric output model, was 10.50, nearly three times lower than the result obtained for the classic filter with Equation (2). Using a fuzzy filter with a symmetric structure, the results were similar, although slightly worse, with an MSE of 10.63. Using a visualization test, it can also be concluded that the adaptive fuzzy filter is a better solution for processing the signal from the voltage sensor (2, Figure 4) than the classic first-order IIR filter. As can be seen in Figure 10a, the IIR filter with the optimal parameter in the MSE sense has good dynamic properties, but it suppresses the

fast-changing component of noise to an unsatisfactory degree. In contrast, the signal at the output of the first-order IIR filter with $a = 0.890$, shown in Figure 10b, at moments of slight changes in the time of the u_{EM} signal (from 145 s to 150 s and from 175 s to 190 s), demonstrates effective noise suppression, comparable to the effect obtained when using a fuzzy filter (Figure 10c,d). Based on the visualization test, it can be concluded that for the waveforms from Figure 10b, at moments of significant changes in the time of the u_E signal, a large dynamic error occurs and a significant phase shift in the u_{EF} signal relative to the reference signal is noticeable. This defect does not occur in the case of the fuzzy filter, because the signal at the output of this filter demonstrates significant convergence with the reference signal throughout the entire period of the system operation (Figure 10c). Figure 13 shows the residuals obtained using a first-order IIR filter (Figure 13a), a fuzzy filter with a symmetric structure (Figure 13b), and an asymmetric output model (Figure 13c). The standard deviations and mean values of the residuals in both cases are 4.51 V and 0.03 V, 6.95 V and 0.02 V, and 6.97 V and 0.04 V, respectively. The value of the variance of the standard deviation of the residuals is therefore more than twice, or one and a half times, higher in the case of using the fuzzy filter, while the mean values are similar and close to zero for both digital signal-processing algorithms. Combined with the results in the form of criterion (14), this clearly indicates that the use of an adaptive fuzzy filter for processing the signal from the RMS voltage sensor supplied to the input of the heating element is a significantly better solution than the use of a classic first-order IIR filter. It should be noted, however, that a disadvantage of the adaptive fuzzy filter is that, for a given value of b , the frequency characteristics have the same signal attenuation in the stopband as a first-order IIR filter. A significant limitation of the proposed method for selecting adaptive fuzzy filter parameters, which uses a reference signal generated by a PI controller, is the need to tune the filter at a constant effective value of the mains voltage. The requirement for constant mains voltage may complicate the process of selecting fuzzy filter model parameters in industrial environments, where heavy, direct starting of induction motors occurs. It should also be noted that the adaptive filter form can only be achieved for systems with a controller without a derivative term.

5. Conclusions

In automatic temperature control systems, it is often important to monitor the signal fed to the input of the controlled object. In industrial environments, measurement of such a quantity is performed using a sensor. The sensor's output signal, with values consistent with a specific standard, is usually significantly affected by noise. In such a situation, separating the useful component from the measurement signal requires the use of a filter that operates well in the time domain, exhibits satisfactory dynamic properties while maintaining an appropriate level of accuracy. For the tested automatic temperature control system (Figure 3) with a PI controller, an adaptive fuzzy filter with a parameter dependent on the modulus of the derivative of the measured signal was used to process the signal from the RMS voltage sensor u_{EM} , fed to the input of the heating element. The innovation employed involves using the control signal u , developed by the PI controller, to determine the time derivative of the measured quantity u_{EM} , and applying a fuzzy model with only three fuzzy sets to change the parameter of a simple first-order IIR filter depending on the modulus of the time derivative of the measured quantity. It was demonstrated that in a control system with a linearized final control element with static object properties, the signal generated by a PI controller can be used to determine the time derivative of the output signal. The proposed procedure was found to be computationally advantageous and applicable to practical applications. This eliminates all difficulties in determining the current derivative of the u_{EM} signal, observed discretely and affected by noise. Based on

experiments conducted on a test stand, it was found that the proposed adaptive fuzzy filter exhibits good dynamic properties while maintaining an appropriate level of accuracy. This observation is confirmed by a comparative analysis of the results obtained using the fuzzy model filtering algorithm and a classic first-order IIR filter. The results, in the form of criterion values (14), and an evaluation of the obtained residuals allow us to conclude that the adaptive fuzzy filter provides significantly better results in processing the signal fed to the object's input in the considered temperature control system (with a PI controller) compared to the classic first-order IIR filter. It was shown that, compared to the classical first-order IIR filter, the improvement of dynamic properties is achieved by introducing a simple fuzzy model $b = f(|\Delta u/T_s|)$ with a symmetric input and output structure or a modified fuzzy model with asymmetry resulting from the uneven distribution of modal values of singleton fuzzy sets at the output in the low-pass filtering algorithm (2). It was found that in the case of the tested control system with a sensor (2, Figure 4), introducing asymmetry in the model output leads to a slightly better reconstruction of the useful component from the noisy u_{EM} signal than using a model with a symmetric structure. This is evidenced by the values of criterion (14) included in Table 1. However, it should be noted that the differences between the MSE values for both models are small. Due to the fact that the process of designing an adaptive fuzzy filter with a symmetric structure is simpler (selection of one parameter b_1) compared to a fuzzy filter with an asymmetric output model (selection of two parameters b_1 and b_2), the use of a fuzzy model with a symmetric structure may be a sufficient solution.

The proposed adaptive fuzzy filter solution for real-time processing of the signal from the u_E sensor in a control system is computationally advantageous and may gain acceptance in industrial solutions using PLCs. It should be noted that the process of selecting adaptive filter parameters requires the RMS value of the mains voltage to be constant over time, regardless of the model structure (symmetrical or asymmetrical output). This is particularly important in industrial environments. Further research is planned to employ more complex filters with different membership functions and other digital filters, both FIR and IIR, and to compare their performance to filters with more complex algorithms, such as the Kalman filter.

Author Contributions: Conceptualization, J.J., A.R. (Aurelia Rybak) and A.R. (Aleksandra Rybak); Methodology, J.J.; Software, J.J.; Validation, J.J.; Formal analysis, J.J.; Writing—original draft, J.J., A.R. (Aurelia Rybak) and A.R. (Aleksandra Rybak); Visualization, A.R. (Aurelia Rybak) and A.R. (Aleksandra Rybak); Funding acquisition, A.R. (Aurelia Rybak). All authors have read and agreed to the published version of the manuscript.

Funding: The work was elaborated in the framework of the statutory research 06/010/BK_25.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy.

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

References

1. Sagdatullin, A. Improving automation control systems and advantages of the new fuzzy logic approach to object real-time process operation. In Proceedings of the 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, 20–22 November 2019; IEEE: Piscataway, NJ, USA; pp. 256–260.
2. De Silva, C.W. *Intelligent Control: Fuzzy Logic Applications*; CRC Press: Boca Raton, FL, USA, 2018.
3. van der Wal, A.J. Application of fuzzy logic control in industry. *Fuzzy Sets Syst.* **1995**, *74*, 33–41. [CrossRef]
4. Juuso, E.K. Fuzzy control in process industry: The linguistic equation approach. In *Fuzzy Algorithms for Control*; Springer: Dordrecht, The Netherlands, 1999; pp. 243–300.

5. Zapła, W. *Seminar on Electrification and Automation of Mines*; Silesian University of Technology Publishing House: Gliwice, Poland, 1994. (In Polish)
6. Smith, S.W. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*; Elsevier Science: Burlington, VT, USA, 2003.
7. Hermanowicz, E. Estimation of samples of the derivative of a signal based on the signal difference derivative. In Proceedings of the ICECS 2001 8th IEEE International Conference on Electronics, Circuits and Systems (Cat. No. 01EX483), St. Julian's, Malta, 2–5 September 2001; IEEE: Piscataway, NJ, USA; Volume 1, pp. 63–66.
8. Smékal, Z. Difference equations with forward and backward differences and their usage in digital signal processor algorithms. *Radioengineering* **2006**, *15*, 45.
9. Sundararajan, D. The Laplace Transform. In *Signals and Systems: A Practical Approach*; Springer Nature: Cham, Switzerland, 2022; pp. 331–375.
10. Luo, J.; Ying, K.; He, P.; Bai, J. Properties of Savitzky–Golay digital differentiators. *Digit Signal Process.* **2005**, *15*, 122–136. [CrossRef]
11. Mishra, D.; Nayak, P.C.; Prusty, R.C.; Panda, S. An improved equilibrium optimization-based fuzzy tilted double integral derivative with filter (F-TIDF-2) controller for frequency regulation of an off-grid microgrid. *Electr. Eng.* **2024**, *106*, 2033–2055. [CrossRef]
12. Gheisarnejad, M.; Khooban, M.H. Design an optimal fuzzy fractional proportional integral derivative controller with derivative filter for load frequency control in power systems. *Trans. Inst. Meas. Control* **2019**, *41*, 2563–2581. [CrossRef]
13. Hosseini, R.; Mashayekhi Fard, J.; Soltani, S. Active filter design and synthesis for hybrid neuro-fuzzy and robust PID controllers. *Int. J. Dyn. Control* **2024**, *12*, 3873–3883. [CrossRef]
14. Zadeh, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353. [CrossRef]
15. Cara, A.B.; Wagner, C.; Hagrass, H.; Pomares, H.; Rojas, I. Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* **2012**, *21*, 459–476. [CrossRef]
16. Piegat, A. *Fuzzy Modeling and Control*; Academic Publishing House EXIT: Warsaw, Poland, 2003. (In Polish)
17. F&F—Domestic and Industrial Automation. Available online: <https://www.fif.com.pl/en/io-extension-modules/424-analog-transmitter-av-1i.html> (accessed on 10 December 2024).
18. Mayne, D.Q.; Firoozan, F. Linear Identification of ARMA Processes. *Automatica* **1982**, *18*, 461–466. [CrossRef]
19. Akaike, H. A new look at the statistical model identification. *IEEE Trans. Autom. Control* **1974**, *AC-19*, 716–723. [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

A Skill-Inspired Adaptive Fuzzy Control Framework for Symmetric Gait Tracking with Sparse Sensor Fusion in Lower-Limb Exoskeletons

Loqmane Bencharif ¹, Abderahim Ibset ¹, Hanbing Liu ¹, Wen Qi ², Hang Su ^{1,*} and Samer Alfayad ¹

- ¹ The IBISC Laboratory, UEVE, University of Paris-Saclay, 91000 Evry, France; loqmane.bencharif@etud.univ-evry.fr (L.B.); abderahim.ibset@etud.univ-evry.fr (A.I.); hanbing.liu@etud.univ-evry.fr (H.L.); samer.alfayad@univ-evry.fr (S.A.)
² School of Future Technology, South China University of Technology, Guangzhou 511442, China; wen.qi@ieee.org
* Correspondence: hang.su@univ-evry.fr

Abstract

This paper presents a real-time framework for bilateral gait reconstruction and adaptive joint control using sparse inertial sensing. The system estimates full lower-limb motion from a single-side inertial measurement unit (IMU) by applying a pipeline that includes signal smoothing, temporal alignment via Dynamic Time Warping (DTW), and motion modeling using Gaussian Mixture Models with Regression (GMM-GMR). Contralateral leg trajectories are inferred using both ideal and adaptive symmetry-based models to capture inter-limb variations. The reconstructed motion serves as reference input for joint-level control. A classical Proportional–Integral–Derivative (PID) controller is first evaluated, demonstrating satisfactory results under simplified dynamics but notable performance loss when virtual stiffness and gravity compensation are introduced. To address this, an adaptive fuzzy PID controller is implemented, which dynamically adjusts control gains based on real-time tracking error through a fuzzy inference system. This approach enhances control stability and motion fidelity under varying conditions. The combined estimation and control framework enables accurate bilateral gait tracking and smooth joint control using minimal sensing, offering a practical solution for wearable robotic systems such as exoskeletons or smart prosthetics.

Keywords: trajectory reconstruction; sparse sensor fusion; Gaussian Mixture Regression (GMR); fuzzy PID control; gait symmetry; wearable robotics

1. Introduction

Wearable robotic systems, especially lower-limb exoskeletons, have attracted significant attention for their ability to support human locomotion, assist rehabilitation, and improve mobility for individuals with impairments [1–4]. To be effective, these systems require not only accurate motion generation but also control strategies [5,6] that can adapt to the variability of human gait and the complexities of physical human–robot interaction [7–9].

Recent advances in data-driven modeling [10] have enabled the use of human movement patterns for robotic control. Among these, Dynamic Time Warping (DTW) is widely used to temporally align gait cycles of varying lengths, facilitating consistent phase matching across trials [11–14]. However, DTW alone does not create generalized models suitable

for motion synthesis. To overcome this, Gaussian Mixture Models (GMM) combined with Gaussian Mixture Regression (GMR) have been employed to generate smooth, probabilistic representations of human joint trajectories [15–17].

After generating reference trajectories, an effective control strategy [18] is required to track them in real time. Classical Proportional–Integral–Derivative (PID) controllers remain common in exoskeleton systems due to their simplicity and low computational requirements [19,20]. However, PID controllers often perform poorly in the presence of non-linearities such as joint elasticity, friction, and gravitational disturbances [20,21]. To address these challenges, fuzzy logic controllers [22] have been proposed as adaptive alternatives capable of tuning gains in response to real-time tracking errors [7–9,23].

This paper presents an integrated framework that combines trajectory reconstruction, symmetry-aware motion prediction, and adaptive joint control for lower-limb exoskeletons. The proposed method reconstructs bilateral gait motion from sparse unilateral sensor input using a sequence of signal filtering, DTW alignment, and GMM-GMR modeling. A key contribution is the use of a technique that refines contralateral leg predictions using phase shifts, amplitude scaling, and offset correction, capturing real-world gait asymmetries often ignored in idealized models. Additionally, we introduce an adaptive fuzzy PID controller that dynamically adjusts control gains based on instantaneous error and error rate. This controller improves tracking fidelity under nonlinear conditions, including simulated joint stiffness and gravity effects. While many existing systems rely on dense sensor networks, our approach uses sparse sensor fusion—integrating data from IMUs, flex sensors, and pressure sensors—processed in parallel to achieve real-time gait estimation with reduced hardware complexity.

The framework is evaluated through simulation using real gait datasets. We assess trajectory accuracy and control robustness across walking, running, and stair climbing tasks, demonstrating the system’s generalization across diverse locomotion modes.

2. Materials and Methods

2.1. Sensor Setup and Dataset

The proposed system integrates wearable sensing, trajectory reconstruction, and exoskeleton control in a modular pipeline. As shown in Figure 1, the full setup consists of three stages: human motion data acquisition, signal preprocessing, and robotic control execution. During the acquisition stage, motion data are collected from the user’s left leg using a sparse sensor configuration. Two inertial measurement units (IMUs; Xsens DOT, Movella Inc., Enschede, The Netherlands) were placed near the hip and knee joints to capture joint-level kinematics. These IMUs provide angular velocity and orientation data, which are used to reconstruct the trajectories of the hip and knee joints in the sagittal plane.

Smart shoes (Qihang Tech, Weihai, China) were equipped with embedded pressure and flex sensors to monitor foot-ground contact and foot arch deformation during dynamic locomotion. In detail, smart shoes are equipped with the following:

- One IMU positioned on the foot to reconstruct the foot’s angular motion and orientation during stance and swing phases.
- Pressure sensors embedded under the sole to monitor foot-ground contact and estimate the center of pressure (COP), which can be used to approximate the center of gravity (COG) trajectory.
- Flex (bending) sensors placed along the shoe upper to detect curvature and deformation of the foot arch during dynamic locomotion.

All sensor signals are transmitted wirelessly via Bluetooth to a central computer for real-time processing.

The system employs a parallel sensing fusion strategy, in which each sensor modality—IMU, pressure, and flex—independently contributes specific biomechanical features. IMUs provide joint kinematics and segment orientations; pressure sensors detect gait events and loading patterns; and flex sensors capture arch deformation related to foot curvature. These signals are not fused at the feature or decision level. Instead, each stream is processed in parallel and integrated only during the control and reconstruction stages. This modular design simplifies implementation, reduces latency, and enhances system interpretability, making it suitable for real-time wearable robotics applications.

Subjects performed locomotion tasks including level-ground walking, stair climbing, and running at self-selected speeds in a controlled indoor environment. IMU data were recorded at 100 Hz, capturing multiple complete gait cycles for each motion type. The data were stored in CSV format with timestamped joint angle estimates derived from the IMU signals. Only the left leg was instrumented, and its motion was used to estimate the contralateral limb motion using symmetry-based reconstruction models [4,24].

After acquisition, the raw signals were smoothed, segmented into gait cycles, and time-normalized to ensure consistency across trials. The resulting processed left-leg trajectories served as input for both motion modeling and the control strategy. All data processing and control algorithms were implemented using MATLAB R2024a (MathWorks Inc., Natick, MA, USA).

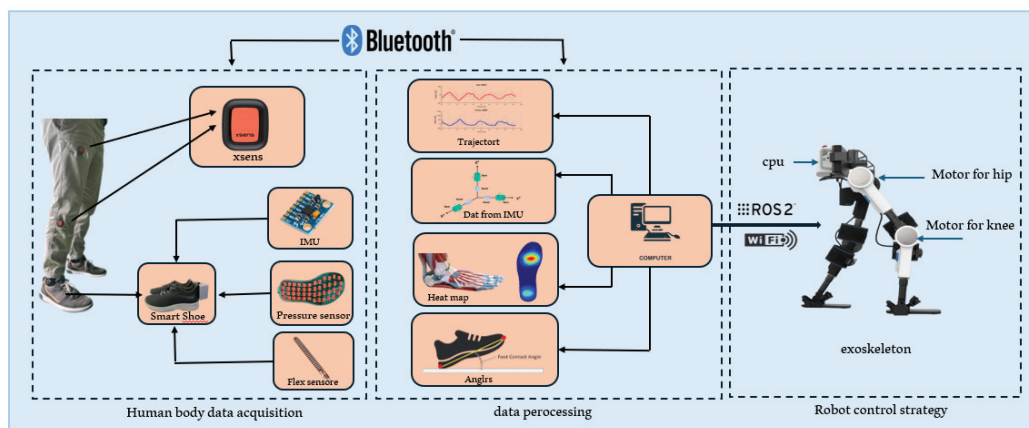


Figure 1. System architecture integrating wearable sensing, signal preprocessing, and exoskeleton control. The pipeline consists of acquisition from body- and foot-mounted sensors, followed by time-aligned trajectory modeling and joint-level control.

2.2. Signal Preprocessing

Raw joint angle signals captured from inertial measurement units (IMUs) often contain high-frequency noise, motion artifacts, and low-frequency drift. These disturbances result from sensor limitations, soft tissue oscillations, and mechanical impacts during locomotion. To mitigate these effects while preserving the biomechanical integrity of the signals, a locally weighted scatterplot smoothing (LOESS) filter was applied [12,24].

LOESS is a non-parametric regression technique that fits a low-order polynomial to localized segments of the data using weighted least squares. This approach enables adaptive smoothing by assigning greater weight to points near the current time t , thereby maintaining sharp motion transitions (e.g., heel strikes or peak flexion) while suppressing transient noise [12].

The smoothed value $\hat{x}(t)$ at time t is computed as follows:

$$\hat{x}(t) = \sum_{i=1}^N w_i(t) x_i$$

where x_i are the original data samples, and $w_i(t)$ are weights determined by their temporal distance from t . The weights follow a tricube kernel function:

$$w_i(t) = \begin{cases} \left(1 - \left|\frac{t-t_i}{h}\right|^3\right)^3 & \text{if } |t - t_i| < h \\ 0 & \text{otherwise} \end{cases}$$

Here, h defines the smoothing span or window width, and t_i represents the timestamps of neighboring samples. In practice, for signals recorded at 100 Hz, a span of 20–50 samples (equivalent to 0.2–0.5 s of data) was found to provide a good balance between denoising and preserving important gait features such as joint peaks and reversals. This span was tuned empirically for each joint signal. The LOESS filter was applied independently to each joint angle across the dataset. This preprocessing step significantly reduced artifacts and jitter, enabling consistent gait cycle segmentation and supporting accurate downstream motion modeling [4,24].

2.3. Trajectory Alignment and Modeling

Human gait is inherently variable due to natural differences in timing, stride length, and joint motion. To build reliable models for gait reconstruction, this variability must be normalized. We addressed this using a two-step method: (1) Dynamic Time Warping (DTW) for temporal alignment; (2) Gaussian Mixture Modeling and Regression (GMM-GMR) for probabilistic trajectory modeling.

1. Dynamic Time Warping (DTW)

DTW is a dynamic programming technique that aligns two time-series signals with different lengths or timing. It finds the optimal non-linear mapping between two sequences by minimizing the cumulative distance between them. This is crucial for comparing gait cycles that may differ slightly in duration or phase.

Given two sequences $\mathbf{X} = \{x_1, \dots, x_N\}$ and $\mathbf{Y} = \{y_1, \dots, y_M\}$, the alignment cost is defined recursively as follows:

$$D(i, j) = \|x_i - y_j\|^2 + \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}$$

The resulting cost matrix D holds the optimal alignment path from the final point $D(N, M)$ back to the start $D(1, 1)$. Above all, the DTW Algorithm for gait alignment is shown in Table 1.

This algorithm was applied pairwise to each joint trajectory across multiple gait cycles. After DTW alignment, all cycles were normalized in time and prepared for statistical modeling. For instance, let $\mathbf{X} = [1, 3, 4]$ and $\mathbf{Y} = [1, 2, 4]$. The local cost at position $D(2, 2)$ compares $x_2 = 3$ and $y_2 = 2$, giving $\|3 - 2\|^2 = 1$. If the minimum of the three neighboring values is, say, $D(1, 1) = 0$, then $D(2, 2) = 1 + 0 = 1$. This process continues until the final cost at $D(3, 3)$ accumulates the minimum alignment cost between sequences, which in this case is 1.

2. Trajectory Modeling with GMM-GMR

After alignment, we applied Gaussian Mixture Models (GMM) to statistically model the time-aligned joint trajectories. Each data point is represented as $\xi_n = [t_n, \theta_n]^\top$, where t_n is the normalized time and θ_n is the corresponding joint angle.

GMM expresses the joint distribution as a sum of Gaussian components:

$$p(\xi) = \sum_{k=1}^K \pi_k \mathcal{N}(\xi \mid \mu_k, \Sigma_k)$$

To reconstruct the expected angle at time t , Gaussian Mixture Regression (GMR) is used:

$$\hat{\theta}(t) = \sum_{k=1}^K h_k(t) \left[\mu_k^\theta + \Sigma_k^{\theta t} (\Sigma_k^{tt})^{-1} (t - \mu_k^t) \right]$$

where

- $h_k(t)$ is the activation weight of the k -th Gaussian at time t ,
- μ_k^t, μ_k^θ are mean time and angle for component k ,
- Σ_k^{tt} and $\Sigma_k^{\theta t}$ are time and cross-covariance matrices.

This enables smooth and probabilistic reconstruction of the joint angle trajectory, capturing both mean behavior and variability across cycles.

The DTW + GMM-GMR pipeline allowed us to align and model hip and knee trajectories for walking, running, and stair climbing. These models were used to reconstruct full bilateral motion from unilateral input, and to generate reference trajectories for exoskeleton control. The use of DTW ensured time consistency across samples, while GMM-GMR provided a robust statistical basis for smooth trajectory generation, even from sparse sensor input.

Table 1. Pseudocode for DTW-based trajectory alignment.

Inputs:

- $\mathbf{X} = \{x_1, \dots, x_N\}$: Reference trajectory
- $\mathbf{Y} = \{y_1, \dots, y_M\}$: Target trajectory to be aligned

Output: Aligned trajectory $\mathbf{Y}'$, optimal warping path P^*

Procedure:

1. Compute local cost: $C(i, j) = \|x_i - y_j\|^2$
2. Initialize cumulative matrix:

$$D(1, 1) = C(1, 1)$$

$$D(i, 1) = D(i - 1, 1) + C(i, 1), \quad D(1, j) = D(1, j - 1) + C(1, j)$$

3. For all $i = 2$ to N , and $j = 2$ to M :

$$D(i, j) = C(i, j) + \min\{D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)\}$$

4. Backtrack from $D(N, M)$ to $D(1, 1)$ to extract optimal warping path P^*
5. Warp $\mathbf{Y}$ along P^* to obtain aligned $\mathbf{Y}'$

Return: Warped trajectory $\mathbf{Y}'$, path P^*

2.4. Contralateral Trajectory Reconstruction

To reduce sensing complexity and enable lightweight deployment, the right-leg joint trajectories were reconstructed from left-leg motion using symmetry-based transformations. This leverages the approximate bilateral symmetry of human gait in the sagittal plane, particularly during steady walking and running [1,4,13,24]. By relying on sparse IMU input from only one side of the body, the system can generate full bilateral motion profiles suitable for exoskeleton control.

The reconstruction process integrates three main stages: time normalization using Dynamic Time Warping (DTW), probabilistic modeling using Gaussian Mixture Models with Regression (GMM-GMR), and contralateral estimation using both ideal and adaptive symmetry transformations. Each stage builds upon the previous to provide continuous, biologically plausible trajectories from unilateral IMU data.

Ideal Symmetry Transformation. As a baseline, the right joint angle is estimated by time-shifting and sign-inverting the left-leg signal:

$$\theta_{\text{right}}(t) = -\theta_{\text{left}}(t + \phi)$$

where ϕ represents the phase delay between limbs, typically half the gait cycle ($\phi = T/2$) for symmetric locomotion. This assumes idealized, mirrored motion between the left and right sides [1,11,12].

Adaptive Symmetry Refinement. To address real-world gait asymmetries caused by user variability, transitions between activities, or uneven ground, we introduce a tunable model:

$$\theta_{\text{right}}(t) = -\alpha \cdot \theta_{\text{left}}(t + \phi) + \delta$$

Here, α adjusts for amplitude differences, ϕ allows phase misalignment, and δ corrects constant offsets. These parameters were optimized using recorded bilateral gait data by minimizing the squared prediction error:

$$L(\alpha, \phi, \delta) = \sum_{t=1}^N \left[\theta_{\text{right,true}}(t) + \alpha \cdot \theta_{\text{left}}(t + \phi) - \delta \right]^2$$

This optimization was solved using a constrained nonlinear solver with physiologically valid bounds. The learned model produced significantly more accurate right-leg reconstructions across different motion types. For example, stair climbing exhibited reduced gain values ($\alpha < 1$), while running produced consistent phase shifts ($\phi \approx T/2$) and negligible offset ($\delta \approx 0$).

This method combines time-aligned modeling and adaptive symmetry to reconstruct bilateral trajectories from unilateral IMU signals alone—without requiring dense sensing or complex hardware. The procedures are shown in Figure 2. While effective across tasks, the adaptive parameters were tuned offline and may vary by user. Future work may incorporate online learning [25,26] or subject-specific calibration for improved generalization.

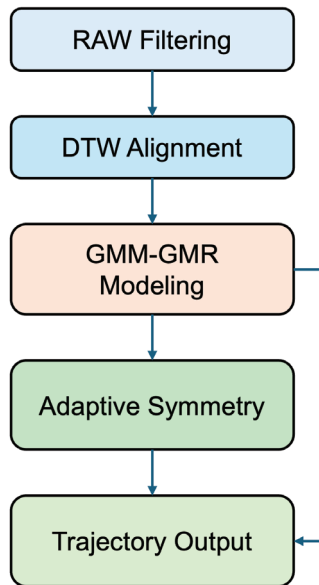


Figure 2. Overview of the trajectory reconstruction pipeline using unilateral IMU input. Raw left-leg signals are filtered and time-aligned using Dynamic Time Warping (DTW), modeled via GMM-GMR, and transformed using symmetry-based estimation to reconstruct right-leg trajectories. The output serves as input for bilateral exoskeleton control.

2.5. Exoskeleton Modeling and Control

Accurate control of wearable robotic systems requires both a faithful dynamic model of the system and an effective control strategy that can accommodate the complexities of human–robot interaction. In this section, we describe the complete modeling of a 4-degree-of-freedom (DOF) sagittal-plane lower-limb exoskeleton [27] and the layered control architecture built upon it, which progresses from classical PID control to virtual compensation strategies and finally to adaptive fuzzy control [28].

Dynamic Modeling of the 4-DOF Exoskeleton

The modeled system consists of two legs, each with actuated hip and knee joints, yielding a total of four degrees of freedom: $\theta = [\theta_{LH}, \theta_{LK}, \theta_{RH}, \theta_{RK}]^T$. The system dynamics are expressed using the Euler–Lagrange formulation:

$$\tau = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}(\theta) + \tau_{\text{spring}}$$

where

- $\mathbf{M}(\theta)$ is the configuration-dependent mass-inertia matrix,
- $\mathbf{C}(\theta, \dot{\theta})$ captures Coriolis and centrifugal forces,
- $\mathbf{G}(\theta)$ models gravity-induced joint torques,
- τ_{spring} represents the passive joint stiffness.

This structure enables simulation of realistic physical conditions, including compliance, gravitational loading, and inter-joint coupling effects [19–21].

2.6. Control Framework

Joint-level trajectory tracking was implemented using IMU-reconstructed reference angles for hip and knee joints, as shown in Figure 3. To assess control performance under dynamic conditions, we evaluated three strategies: classical PID control, virtual compensation for biomechanical dynamics, and adaptive fuzzy PID control.

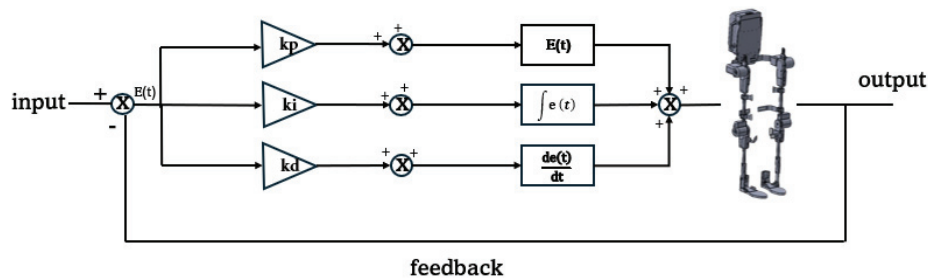


Figure 3. Block diagram of the classical PID control.

2.6.1. PID-Based Joint Control

As a baseline, a conventional PID controller was implemented for each joint:

$$\tau_i(t) = K_{p,i}e_i(t) + K_{i,i} \int e_i(t)dt + K_{d,i} \frac{de_i(t)}{dt}$$

where $e_i(t) = \theta_{i,\text{ref}}(t) - \theta_i(t)$ is the position error between the reconstructed reference trajectory and the measured joint angle.

Initial gains were selected using the Ziegler–Nichols method [19], where the proportional gain is tuned to induce sustained oscillation, and the ultimate gain K_u and oscillation period T_u are used to derive:

$$K_p = 0.6K_u, \quad K_i = \frac{2K_p}{T_u}, \quad K_d = \frac{K_p T_u}{8}$$

These gains were refined manually to stabilize tracking performance during gait transitions.

While PID control performs well in linear settings, its effectiveness degrades when faced with nonlinear joint dynamics introduced by virtual spring compliance and gravitational torques, both of which depend on joint position and gait phase [20,21].

2.6.2. Virtual Spring and Gravity Compensation

To improve control under realistic joint dynamics, passive torque terms were added to model elasticity and gravitational effects. The spring-like restoring torque is computed as follows:

$$\tau_{\text{spring},i} = -k_i(\theta_i - \theta_{i,\text{ref}})$$

where k_i is the joint stiffness constant. Gravitational torque is calculated as follows:

$$\tau_{\text{gravity},i} = m_i g l_i \sin(\theta_i)$$

with m_i , l_i , and g representing segment mass, length to center of mass, and gravity respectively.

The total commanded torque, as shown in Table 2, becomes:

$$\tau_i(t) = \tau_{\text{PID},i} + \tau_{\text{spring},i} + \tau_{\text{gravity},i}$$

This compensation improves control fidelity, particularly during stance initiation and swing recovery phases.

Table 2. Torque computation using PID control with virtual spring and gravity compensation.

Procedure: Virtual Torque Computation

- **Input:** $\theta_i, \dot{\theta}_i, \theta_{i,\text{ref}}, k_i, m_i, l_i, g$
- **1.** Compute error: $e_i = \theta_{i,\text{ref}} - \theta_i$
- **2.** Compute PID torque: $\tau_{\text{PID},i}$
- **3.** Compute spring torque: $\tau_{\text{spring},i} = -k_i \cdot e_i$
- **4.** Compute gravity torque: $\tau_{\text{gravity},i} = m_i g l_i \sin(\theta_i)$
- **5.** Total output: $\tau_i = \tau_{\text{PID},i} + \tau_{\text{spring},i} + \tau_{\text{gravity},i}$

2.6.3. Adaptive Fuzzy PID Controller

To overcome the limitations of fixed-gain PID control, an adaptive fuzzy PID structure [29] was implemented, based on the comparison of different controllers shown in Table 3. The fuzzy controller adjusts PID gains online using tracking error $e(t)$ and error rate $\dot{e}(t)$ as input to a rule-based inference engine.

Fuzzy rules are of the form:

IF $e(t)$ is Large AND $\dot{e}(t)$ is Positive, THEN Decrease K_p , Increase K_d

Gain adjustments follow:

$$K_p^{\text{new}} = K_p + \Delta K_p(e, \dot{e})$$

$$K_i^{\text{new}} = K_i + \Delta K_i(e, \dot{e})$$

$$K_d^{\text{new}} = K_d + \Delta K_d(e, \dot{e})$$

Triangular membership functions were defined for both inputs and outputs, as shown in Figure 4, with a 7×7 rule matrix designed to preserve stability. The system was executed in Simulink with a 1 ms sampling rate. This architecture improved adaptability under dynamic gait transitions and unsteady loading [7–9].

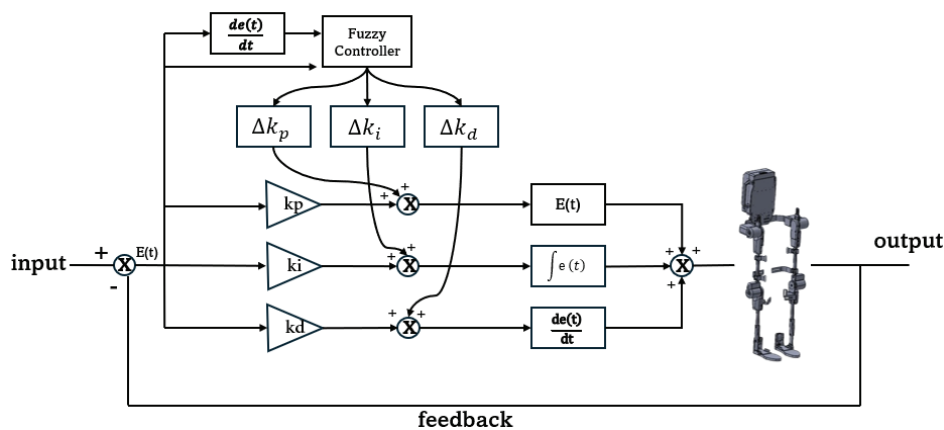


Figure 4. Structure of the adaptive fuzzy PID controller with real-time gain tuning.

Table 3. Qualitative comparison of adaptive controllers for wearable robotics.

Method	Advantages	Limitations
Fuzzy PID	Real-time tuning, interpretable, low-cost [7,8]	Manual tuning, limited personalization
MPC	Optimal control with constraints [30]	Requires accurate models, high computational load
SMC	Robust to disturbances [31]	Chattering, less smooth
RL	Learns from data, adapts over time [32]	Needs large datasets, low interpretability

The fuzzy rule base was designed heuristically for average gait behavior. However, inter-subject variability or atypical motion may require retuning. Future work may incorporate learning based or self-tuning fuzzy logic systems for better generalization to diverse users or environmental contexts.

2.6.4. Validation Metrics and Evaluation Formulation

To comprehensively evaluate the performance of the proposed control strategies across the four exoskeleton joints (bilateral hip and knee), several quantitative metrics are adopted. These metrics assess tracking accuracy, smoothness of motion, and energy efficiency—critical aspects for ensuring safety and usability in wearable assistive systems [7,8,20,21].

(1) Root Mean Square (RMS) Tracking Error.

The RMS error provides a global measure of trajectory fidelity over a gait cycle. It reflects how closely the actual joint motion follows the reference trajectory:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta_{\text{ref},i} - \theta_{\text{act},i})^2}$$

where $\theta_{\text{ref},i}$ and $\theta_{\text{act},i}$ are the reference and actual joint angles at the i^{th} time sample, and N is the total number of samples per cycle.

(2) Maximum Error.

This metric captures the worst-case deviation and is critical for identifying potential instability or delay in response:

$$e_{\text{max}} = \max_t |\theta_{\text{ref}}(t) - \theta_{\text{act}}(t)|$$

The maximum error defined is particularly useful in detecting overshoot or lag during rapid gait transitions [11,21].

(3) Control Effort.

To assess actuator workload and energy usage, we compute the control effort as the time-integrated product of joint torque and angular velocity:

$$E = \int_0^T |\tau(t) \cdot \dot{\theta}(t)| dt$$

The control effort defined indicates system efficiency and is critical in battery-powered exoskeletons where energy optimization is essential [7,8].

(4) Jerk Index.

Jerk, the third derivative of position with respect to time, measures motion smoothness. High jerk can cause user discomfort or biomechanical incompatibility. The jerk index is defined as follows:

$$J = \int_0^T \left(\frac{d^3\theta(t)}{dt^3} \right)^2 dt$$

Lower values imply smoother joint motion, enhancing comfort and reducing mechanical stress [7,20].

These evaluation metrics were applied consistently to all four sagittal joints (left/right hip and knee) during simulation experiments. Performance was compared across the PID and fuzzy PID controllers, under both ideal and nonlinear conditions (e.g., elastic elements, gravity disturbances), providing a robust analysis of controller adaptability and system responsiveness.

2.7. Trajectory Reconstruction Results

The proposed reconstruction pipeline was evaluated across the locomotion modes in normal walking. For each mode, the pipeline followed two main steps: (1) Gait cycles were temporally aligned using Dynamic Time Warping (DTW); (2) Gaussian Mixture Modeling with Regression (GMM-GMR) was applied to learn a smoothed motion profile. Right-leg trajectories were then estimated using both ideal and adaptive symmetry models [7,11,12,15,17,23].

Figure 5 shows noticeable variability in raw gait cycles due to inconsistent step timing. DTW effectively normalizes this timing, revealing consistent stride structure across repetitions, essential for modeling. Similar improvements are seen in Figure 6. Temporal misalignments in raw hip signals are smoothed post-alignment, enabling accurate downstream modeling.

As seen in Figure 7, the adaptive symmetry model closely tracks the GMM-inferred motion, outperforming the ideal symmetry model, which introduces phase and amplitude offsets in the right knee prediction. The adaptive model again provides superior results (Figure 8), reducing overshoot and misalignment, demonstrating its effectiveness for estimating contralateral hip dynamics during walking.

Summary:

- DTW normalization is essential for consistent GMM modeling.
- Adaptive symmetry modeling significantly improves contralateral motion estimation.
- The method generalizes well from cyclic motions (walking, running) to asymmetric tasks (stair climbing).

Dynamic Time Warping (DTW) was chosen for its ability to temporally align gait cycles with non-uniform timing, a key challenge in human locomotion data. Unlike linear interpolation or time-normalized averaging—which assume consistent stride durations—DTW

allows flexible non-linear alignment of key gait events (e.g., heel-strike, toe-off) without distorting kinematic patterns. Other alternatives, such as Principal Component Analysis (PCA)-based time normalization or Uniform Temporal Scaling, offer computational speed but lose cycle-specific variation critical for personalized modeling. DTW preserves temporal dynamics while maintaining feature correspondence across cycles, making it more suitable for accurate Gaussian Mixture Modeling in variable gait data.

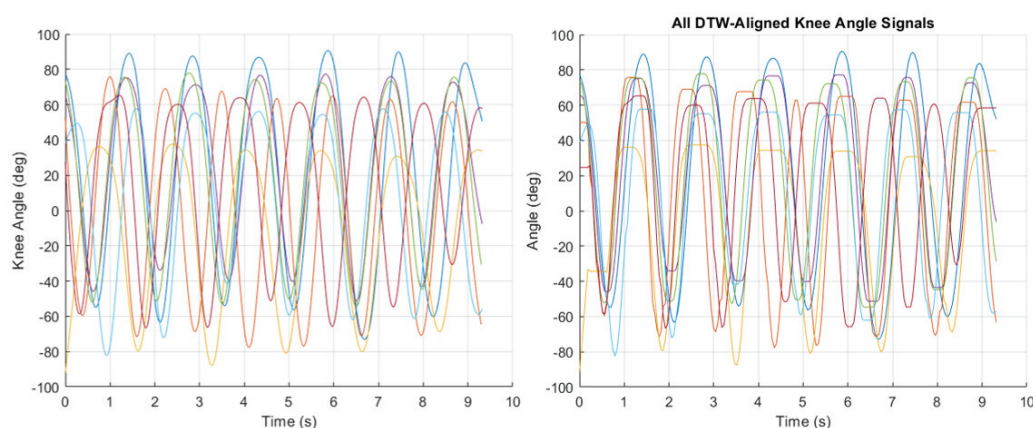


Figure 5. Knee joint: Raw left-knee trajectories (**left**) and DTW-aligned signals (**right**) showing temporal normalization across trials.

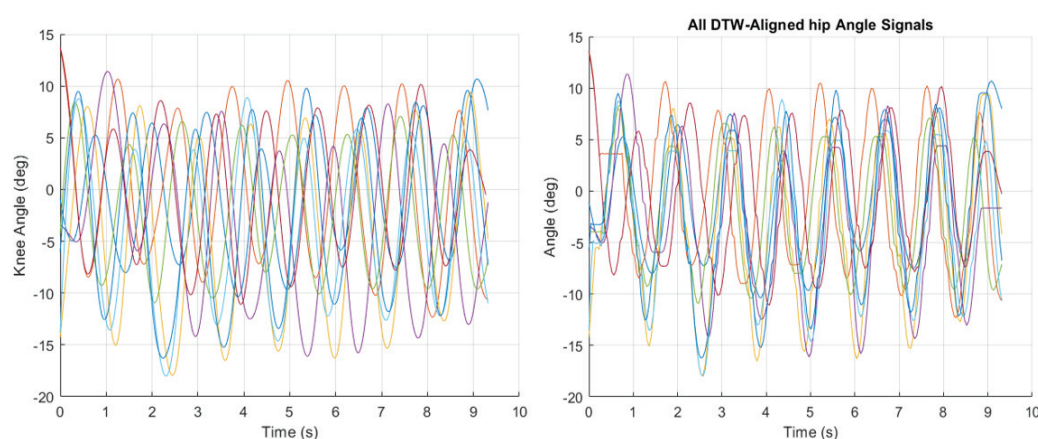


Figure 6. Hip joint: Raw left-hip joint angles (**left**) and aligned trajectories (**right**) after DTW.

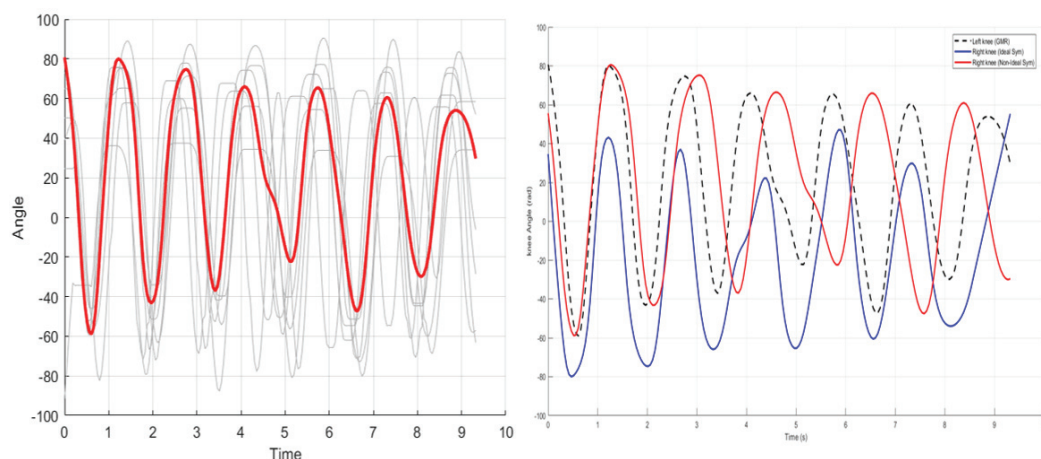


Figure 7. Knee joint: GMM-GMR output (green), right-knee prediction via ideal (orange dashed) and adaptive symmetry (red solid).

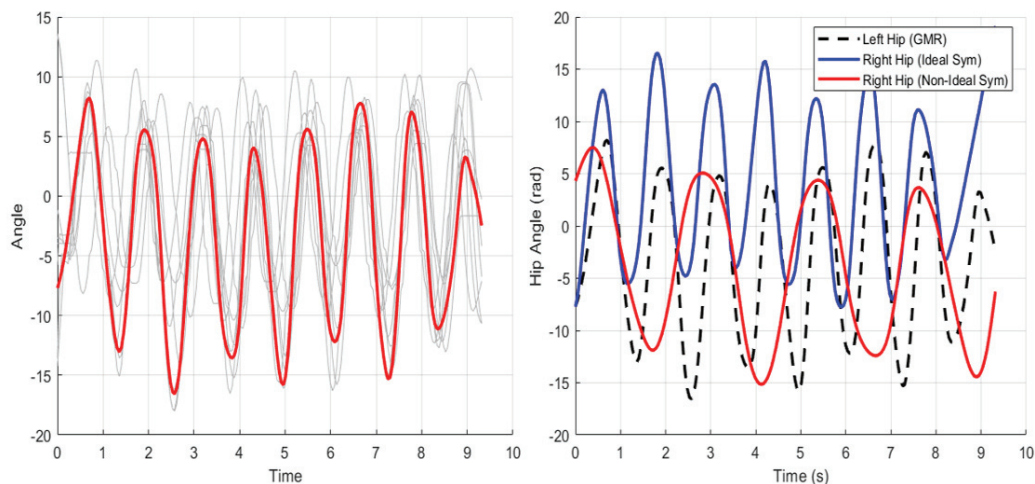


Figure 8. Hip joint: Right-hip motion estimated using both symmetry models.

2.8. PID Control Performance

The reconstructed joint trajectories—generated through DTW alignment, GMM-GMR modeling, and adaptive symmetry transformation—were used as reference inputs for joint-level control using a classical Proportional–Integral–Derivative (PID) controller. The system was evaluated under ideal linear conditions, with no additional spring compliance or gravitational loads, simulating a best-case actuation scenario.

Figure 9 presents representative tracking results for the left and right hip and knee joints. Reference trajectories are shown as dashed lines, while solid lines indicate actual joint responses. The performance of the tracking was evaluated for three types of motions: walking, running, and climbing stairs. The following observations summarize the controller’s behavior. Accurate tracking with minimal overshoot or delay. The controller handled the periodic, low-speed motion with smooth transitions and negligible steady-state error.

Overall, the results show that classical PID control performs adequately when applied to an exoskeleton model governed by approximately linear dynamics. Under these conditions, the static gain structure of PID is sufficient to achieve stable tracking with low steady-state error [8,9,19]. Its simplicity, fast response, and ease of implementation have made it a longstanding baseline in robotic control [7,13,20]. However, this performance assumes ideal actuation and neglects realistic factors such as joint elasticity, load disturbances, and unmodeled dynamics [3,21,32]. As these nonlinearities are introduced in the following sections, PID’s limitations become increasingly evident, motivating the transition to more adaptive control strategies.

2.9. Effect of Spring and Gravity Compensation on PID Performance

To simulate more realistic actuation dynamics, virtual joint stiffness and gravity effects were introduced into the exoskeleton model. Joint compliance was modeled using linear torsional springs, and segment-wise gravitational torque was incorporated based on limb orientation and mass [3,21]. The original PID controller, tuned under idealized dynamics, was retained without modification to assess its robustness under these altered conditions.

Figure 10 illustrates the impact of these added nonlinearities on joint tracking for a walking cycle. While reference trajectories remain unchanged, the simulated joint responses deviate significantly, especially in the knee joints. Amplitude attenuation, high-frequency oscillations, and phase lag are apparent, indicating deteriorated tracking performance.

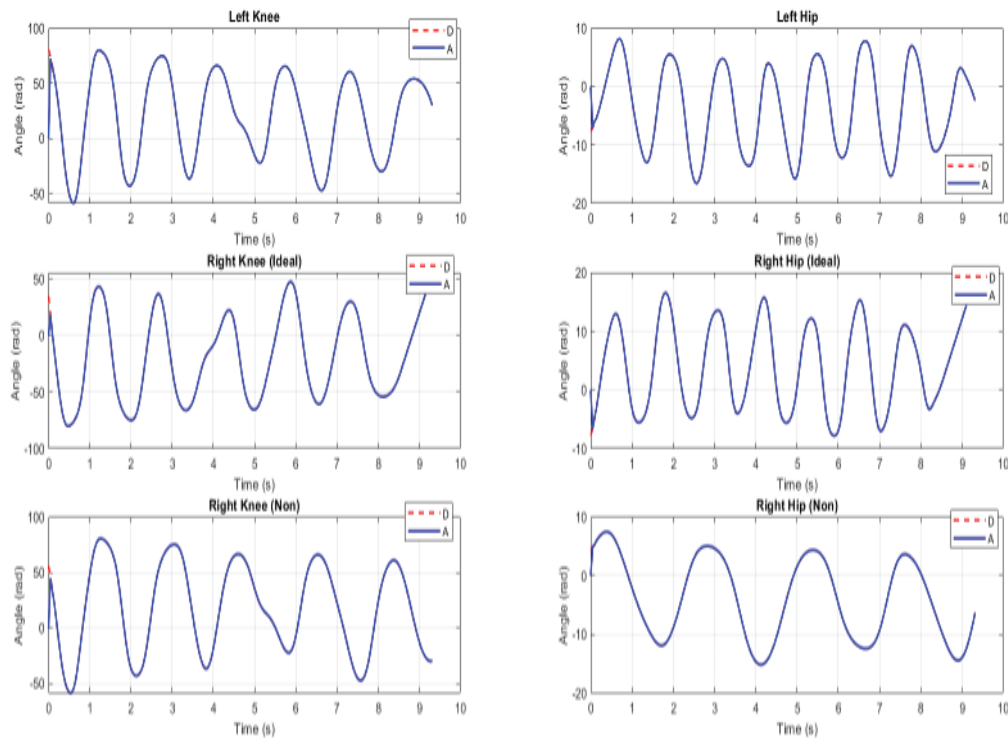


Figure 9. Joint angle tracking performance using classical PID controller under linear system dynamics. Reference trajectories (dashed), joint responses (solid).

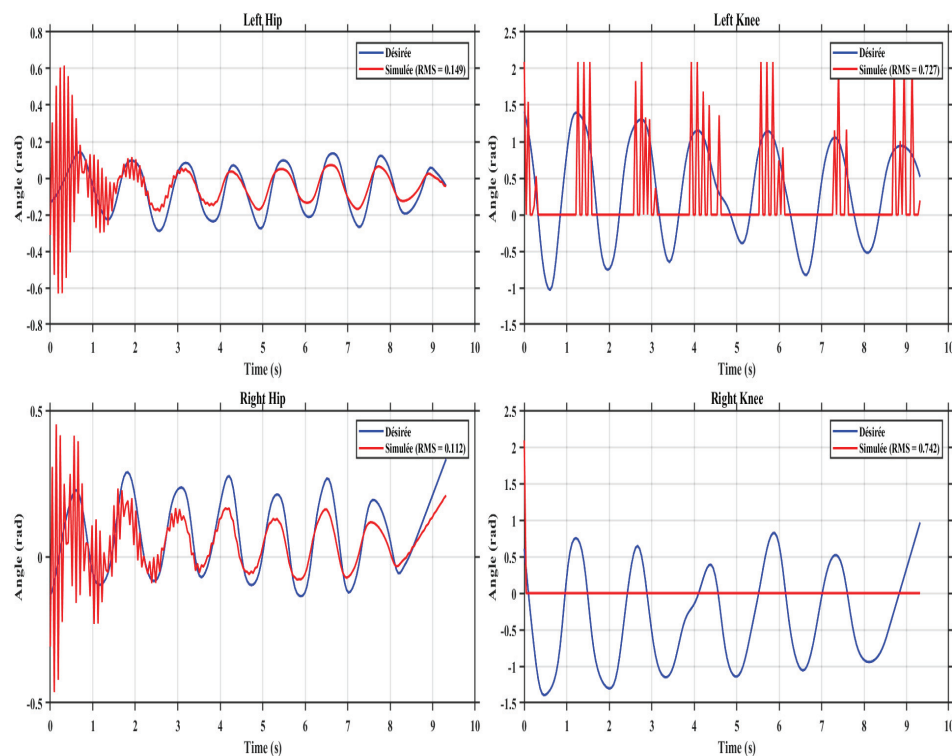


Figure 10. PID tracking performance under spring and gravity dynamics for walking. Desired trajectories (blue); simulated joint responses (red).

The following observations highlight specific failure modes during walking:

- **Left and Right Hip:** Spring-induced oscillations degrade early-cycle stability. Phase lag appears post toe-off, with the system underreacting to flexion demands during swing.

- **Left Knee:** High overshoot and instability are observed, likely due to resonance between the spring dynamics and PID gains. The controller fails to suppress abrupt transitions near peak flexion.
- **Right Knee:** Motion becomes highly discontinuous, indicating loss of closed-loop stability. Gravity-induced drift and compliance mismatch lead to poor synchronization with the contralateral leg.

These results emphasize that classical PID controllers, while simple and effective under linear assumptions, are inadequate for managing the time-varying dynamics introduced by compliance and gravitational forces. Their static gain structure cannot compensate for joint-dependent variability or disturbance-induced coupling [9,19,21]. In wearable robotic systems where elastic actuators and uneven load conditions are common, more adaptive strategies such as fuzzy-PID control are required to ensure stable and reliable motion tracking [7,20,23,32].

2.10. Adaptive Fuzzy PID Controller

To overcome the limitations of classical PID under nonlinear joint dynamics, an adaptive fuzzy PID controller was developed. This controller extends a standard PID structure by integrating a Mamdani-type fuzzy inference system (FIS), enabling dynamic tuning of K_p , K_i , and K_d gains based on real-time tracking behavior [7,20,23,32,33].

As shown in Figure 11, the FIS receives two inputs: the tracking error $e(t)$ and its derivative $\dot{e}(t)$, both normalized to the range $[-3, 3]$. Each input is described by three triangular membership functions (MFs): *Negative*, *Zero*, and *Positive*. The FIS outputs three incremental adjustments: ΔK_p , ΔK_i , and ΔK_d , each mapped over the range $[-1, 1]$ using five triangular MFs (from strong decrease to strong increase).

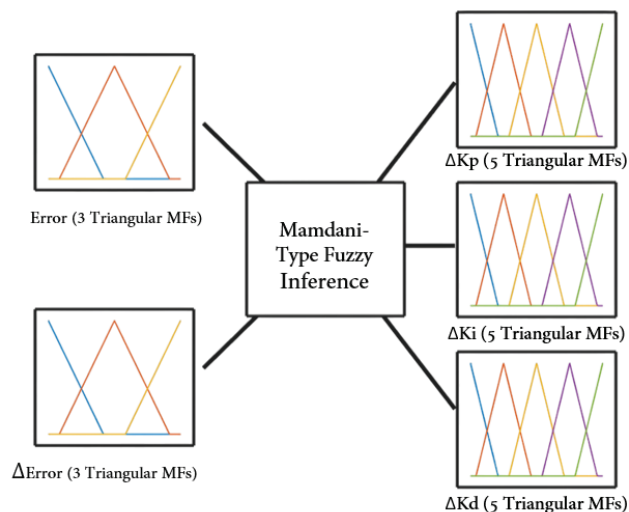


Figure 11. Fuzzy PID architecture. Error and error rate inputs are mapped to triangular MFs and processed through a Mamdani inference engine to tune ΔK_p , ΔK_i , and ΔK_d in real time.

The rule base contains nine core fuzzy rules designed to reflect intuitive control logic. For example:

- **Rule 1:** If Error is Negative and DeltaError is Negative, then ΔK_p is Strong Increase, ΔK_i is Neutral, and ΔK_d is Increase. This boosts responsiveness while maintaining damping.
- **Rule 5:** If both Error and DeltaError are Zero, then no gain changes occur—preserving current control behavior in steady state.

- **Rule 9:** If Error is Positive and DeltaError is Positive, then ΔK_p is Strong Decrease, ΔK_i is Slight Decrease, and ΔK_d is Increase. This reduces overshoot and improves damping.

The fuzzy system uses a min–max composition, Mamdani inference method, and centroid defuzzification. All gain updates are applied incrementally to baseline PID values, enabling adaptive behavior throughout the gait cycle [23,32].

The fuzzy PID controller was evaluated using walking trials under nonlinear joint conditions (spring compliance and gravity torque). Figures 12 and 13 illustrate tracking results for ideal and non-ideal symmetry assumptions.

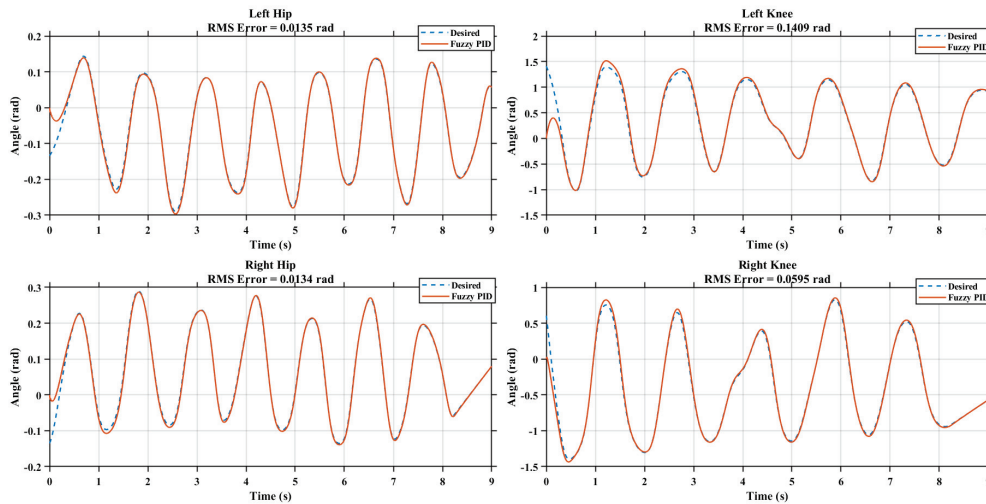


Figure 12. Tracking performance using fuzzy PID under nonlinear dynamics with ideal symmetry. All joints follow reference trajectories closely with minimal steady-state error and smooth transitions.

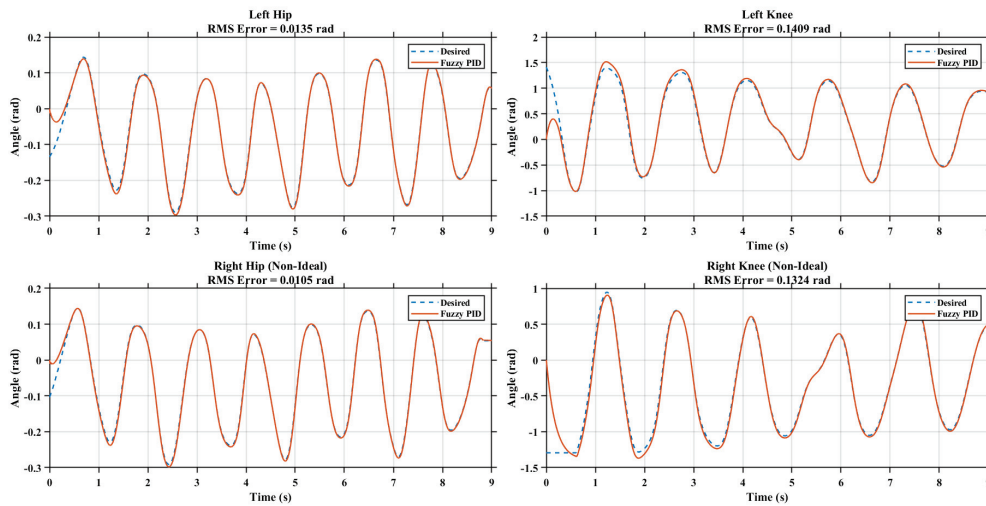


Figure 13. Tracking performance using fuzzy PID under nonlinear dynamics with non-ideal symmetry. The controller maintains stable and smooth output despite asymmetry in reference trajectories.

Observations (Ideal Symmetry):

- The controller enables precise tracking across all joints.
- Spring-induced oscillations are effectively attenuated.
- The system maintains phase alignment and amplitude fidelity.

Observations (Non-Ideal Symmetry):

- Slight increases in phase lag are observable, especially at the right knee.
- The controller compensates well for asymmetries, avoiding large deviations.

- Joint transitions remain smooth and natural despite variability.

Overall, this adaptive control strategy improves stability and tracking fidelity in the presence of real-world nonlinearities. It enables real-time adaptation to changing joint dynamics, which is particularly advantageous for wearable robotic systems [7,20,23,32,33].

While classical PID controllers are simple and widely used, they lack adaptability under time-varying dynamics. Alternative approaches such as Gain Scheduling, Model Predictive Control (MPC), and Adaptive Neural Controllers have been explored for exoskeletons. Gain scheduling requires predefined operating points and extensive tuning; MPC offers optimal control but is computationally intensive and less practical for embedded real-time systems. Neural controllers can adapt to nonlinearities but require large datasets and may lack interpretability. Fuzzy PID, by contrast, combines rule-based adaptability with low computational overhead, enabling real-time tuning of gains based on error dynamics. Its interpretability, fast response, and robustness make it a superior choice for wearable robotics where safety and transparency are essential [7,20,23,32].

2.11. Validation and Quantitative Evaluation

To rigorously assess the performance of the proposed control framework, walking experiments were conducted comparing a classical PID controller and the proposed adaptive fuzzy PID controller. The root mean square (RMS) error between actual and reference joint trajectories was used to quantify tracking performance across the left and right hip and knee joints. For comparison, the PID controller was evaluated both in an ideal (simulation-based) scenario and under real-world limitations [7,20,32].

Observations: Table 4 highlights the joint-level tracking accuracy across controllers. In simulation (ideal conditions), the PID controller achieves low RMS errors. However, when deployed under real-world limitations, its performance degrades significantly. For instance, the right knee error rises from 0.042 to 0.742 rad—a more than 17-fold increase.

In contrast, the adaptive fuzzy PID controller maintains consistently low errors in all joints, even under uncertainty. Relative to the PID under limitations, the RMS error was reduced:

- **Left hip:** 90.6% (from 0.149 to 0.014 rad)
- **Right hip:** 88.4% (from 0.112 to 0.013 rad)
- **Left knee:** 80.6% (from 0.727 to 0.141 rad)
- **Right knee:** 91.9% (from 0.742 to 0.060 rad)

Table 4. RMS Tracking Errors (in radians) for Each Joint Across Controllers.

Joint	PID (Ideal)	PID Limitations	Adaptive Fuzzy PID
Left Hip	0.019	0.149	0.014
Right Hip	0.012	0.112	0.013
Left Knee	0.127	0.727	0.141
Right Knee	0.042	0.742	0.060

These results show that while PID may perform adequately in linear, controlled conditions, it lacks the robustness needed to handle variability. The adaptive fuzzy PID, by contrast, responds to joint-specific deviations in real time, offering reliable and accurate performance across all lower-limb joints [7,32].

Table 5 compares the overall control effort and motion smoothness for both controllers. The adaptive fuzzy PID reduced the control effort by 38.8%, indicating less torque demand to achieve accurate tracking. This reduction directly contributes to energy efficiency, reduced actuator strain, and longer operational endurance—all critical in wearable robotic systems.

Table 5. Control Effort and Motion Smoothness Comparison.

Metric	PID Controller	Adaptive Fuzzy PID Controller
Control Effort [$\int \tau - \dot{\theta} $]	23.7	14.5
Jerk Index [$\int \ddot{\theta}^2 dt$]	91.6	53.8

Additionally, the jerk index was reduced by 41.2%, indicating that the adaptive fuzzy controller produces smoother joint trajectories with fewer abrupt changes in acceleration. This smoother motion not only improves user comfort but also reflects better dynamic stability of the controlled limb. These improvements are also supported by the integration of a virtual spring component in the control architecture. This spring introduces passive compliance that emulates biological elasticity, allowing the system to absorb minor deviations and offload some of the control burden from the actuators. The result is more natural, energy-efficient motion with improved stability [23,33].

3. Discussion

This study presented a simulation-based framework that integrates sparse sensing and symmetry modeling for adaptive gait trajectory reconstruction and exoskeleton control. While the proposed approach demonstrated promising results in terms of tracking fidelity, robustness, and control efficiency, several ongoing developments aim to further bridge the gap between simulation and real-world deployment. A key direction is the hardware implementation and real-world validation of the proposed methods. A physical lower-limb exoskeleton system has already been constructed, integrating the same sparse IMU, flex, and pressure sensor setup described in this paper. Initial bench testing has been completed, and work is currently underway to deploy the adaptive fuzzy PID controller and energy-tank-based impedance modulation strategies on the embedded platform. These efforts will culminate in human-in-the-loop experiments that evaluate control responsiveness, trajectory accuracy, and safety under walking, stair climbing, and running conditions.

Another extension involves enhancing the sensor fusion architecture to improve estimation of the center of pressure (COP) and center of gravity (COG). Specifically, pressure and flex sensors will be leveraged to extract foot-ground contact and arch curvature information, providing richer feedback for real-time stability monitoring and energy-aware control, particularly during locomotion transitions and when traversing uneven terrain. Moreover, the current fuzzy controller operates with a predefined rule base, which, while effective, may lack adaptability for highly personalized gait dynamics. To address this limitation, future versions may adopt online learning techniques such as neuro-fuzzy adaptation or reinforcement learning [32,34]. These techniques would enable subject-specific tuning based on historical gait data and dynamic sensor patterns, thereby enhancing personalization and long-term performance.

To further improve motion planning, model predictive control (MPC) is under consideration as a complementary strategy. MPC can offer anticipatory control capabilities, particularly beneficial in scenarios involving multi-step planning or rapidly changing environments [30]. Its predictive nature could complement the adaptive fuzzy controller by providing higher-level trajectory modulation.

Finally, while the current system is limited to sagittal-plane motion, an important direction is the expansion toward 3D gait modeling and lateral control. This would involve the integration of lateral joint dynamics and quaternion-based orientation estimation for full-body motion reconstruction [35]. Such an extension would greatly enhance the system's applicability to real-world ambulatory scenarios, including complex maneuvers such as turning, side-stepping, or adapting to inclined surfaces. In summary, while the current

work establishes a strong foundation for adaptive gait tracking and control using minimal sensors, ongoing and future developments are expected to significantly advance its practical utility in wearable robotic systems.

4. Conclusions

This paper presented an integrated framework that combines gait trajectory reconstruction using unilateral IMU data with adaptive fuzzy PID control for lower-limb exoskeletons. Through a structured pipeline involving signal filtering, dynamic time warping, statistical modeling, and symmetry-based extrapolation, we reconstructed full-joint trajectories for walking, running, and stair climbing using data from a single leg. We evaluated classical PID control and found that it performs well under linear, idealized conditions. However, its effectiveness diminishes when nonlinearities such as joint stiffness and gravitational loading are introduced. To address these challenges, we incorporated a virtual spring-damper model and implemented an adaptive fuzzy PID controller capable of adjusting gains in real time based on error dynamics.

The results demonstrated that the adaptive fuzzy control strategy significantly improves tracking accuracy, motion smoothness, and energy efficiency under more realistic joint dynamics. The overall control architecture remains lightweight, modular, and scalable, making it well-suited for deployment in wearable robotic systems aimed at rehabilitation or mobility support. This work lays a foundation for future extensions involving predictive control, online adaptation, and real-world implementation in diverse locomotion environments.

Author Contributions: Conceptualization, H.S. and S.A.; Methodology, L.B. and A.I.; Validation, L.B. and H.L.; Formal analysis, L.B. and H.L.; Investigation, L.B. and A.I.; Data curation, A.I.; Writing—original draft, L.B., A.I. and H.L.; Writing—review and editing, W.Q. and H.S.; Supervision, W.Q., H.S. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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 conflicts of interest.

References

1. Chacon-Murguia, M.I.; Sandoval-Rodriguez, R.; Arias-Enriquez, O. Human gait feature extraction including a kinematic analysis toward robotic power assistance. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 68. [CrossRef]
2. Viteckova, S.; Kutilek, P.; Jirina, M. Wearable lower limb robotics: A review. *Biocybern. Biomed. Eng.* **2013**, *33*, 96–105. [CrossRef]
3. Tejima, N. Rehabilitation robotics: A review. *Adv. Robot.* **2001**, *14*, 551–564. [CrossRef]
4. Hu, B.; Rouse, E.; Hargrove, L. Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals. *Front. Robot. AI* **2018**, *5*, 14. [CrossRef]
5. Jleilat, S.; Ammounah, A.; Abdulmalek, G.; Nouveliere, L.; Su, H.; Alfayad, S. Distributed real-time control architecture for electrohydraulic humanoid robots. *Robot. Intell. Autom.* **2024**, *44*, 607–620. [CrossRef]
6. Sharma, R.; Gaur, P.; Bhatt, S.; Joshi, D. Optimal fuzzy logic-based control strategy for lower limb rehabilitation exoskeleton. *Appl. Soft Comput.* **2021**, *105*, 107226. [CrossRef]
7. Zhong, B.; Cao, J.; Guo, K.; McDaid, A.; Peng, Y.; Miao, Q.; Xie, S.; Zhang, M. Fuzzy logic compliance adaptation for an assist-as-needed controller on the Gait Rehabilitation Exoskeleton (GAREX). *Robot. Auton. Syst.* **2020**, *133*, 103642. [CrossRef]
8. Toan, N.V.; Khoi, P.B. Fuzzy-based-admittance controller for safe natural human–robot interaction. *Adv. Robot.* **2019**, *33*, 815–823. [CrossRef]

9. Zawawi, M.Z.F.B.M.; Elamvazuthi, I.; Aziz, A.B.A.; Daud, S.A. Comparison of PID and fuzzy logic controller for DC servo motor in the development of lower extremity exoskeleton for rehabilitation. In Proceedings of the 2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA), Kuala Lumpur, Malaysia, 19–21 September 2017; pp. 1–6.
10. Xu, Y.; Su, H.; Ma, G.; Liu, X. A novel dual-modal emotion recognition algorithm with fusing hybrid features of audio signal and speech context. *Complex Intell. Syst.* **2023**, *9*, 951–963. [CrossRef]
11. Gaspar, M.; Welke, B.; Seehaus, F.; Hurschler, C.; Schwarze, M. Dynamic Time Warping compared to established methods for validation of musculoskeletal models. *J. Biomech.* **2017**, *55*, 156–161. [CrossRef]
12. Avola, D.; Cinque, L.; De Marsico, M.; Fagioli, A.; Foresti, G.L.; Mancini, M.; Mecca, A. Signal enhancement and efficient DTW-based comparison for wearable gait recognition. *Comput. Secur.* **2024**, *137*, 103643. [CrossRef]
13. Huang, L.; Zheng, J.; Hu, H. A gait phase detection method in complex environment based on DTW-mean templates. *IEEE Sens. J.* **2021**, *21*, 15114–15123. [CrossRef]
14. Wang, X.; Kyrarini, M.; Ristić-Durrant, D.; Spranger, M.; Gräser, A. Monitoring of gait performance using dynamic time warping on IMU-sensor data. In Proceedings of the 2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Benevento, Italy, 15–18 May 2016; pp. 1–6.
15. Ti, B.; Gao, Y.; Li, Q.; Zhao, J. Dynamic movement primitives for movement generation using GMM-GMR analytical method. In Proceedings of the IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 14–17 March 2019.
16. Wiest, J.; Höffken, M.; Kreßel, U.; Dietmayer, K. Probabilistic trajectory prediction with Gaussian mixture models. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 141–146.
17. Maeda, G.J.; Neumann, G.; Ewerton, M.; Lioutikov, R.; Kroemer, O.; Peters, J. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Auton. Robot.* **2017**, *41*, 593–612. [CrossRef]
18. Li, Q.; Cicirelli, F.; Vinci, A.; Guerrieri, A.; Qi, W.; Fortino, G. Quadruped Robots: Bridging Mechanical Design, Control, and Applications. *Robotics* **2025**, *14*, 57. [CrossRef]
19. Andersson, R.; Cronhjort, M.; Chilo, J. Wireless PID-Based Control for a Single-Legged Rehabilitation Exoskeleton. *Machines* **2024**, *12*, 745. [CrossRef]
20. Belkadi, A.; Oulhadj, H.; Touati, Y.; Khan, S.A.; Daachi, B. On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach. *Appl. Soft Comput.* **2017**, *60*, 87–100. [CrossRef]
21. Yamazaki, T.; Sakaino, S.; Tsuji, T. Estimation and kinetic modeling of human arm using wearable robot arm. *Electr. Eng. Jpn.* **2017**, *199*, 57–67. [CrossRef]
22. Su, H.; Ovr, S.E.; Xu, Z.; Alfayad, S. Exploring the potential of fuzzy sets in cyborg enhancement: A comprehensive review. *IEEE Trans. Fuzzy Syst.* **2024**, *33*, 810–827. [CrossRef]
23. Ferdaus, M.M.; Pratama, M.; Anavatti, S.G.; Garratt, M.A.; Pan, Y. Generic evolving self-organizing neuro-fuzzy control of bio-inspired unmanned aerial vehicles. *IEEE Trans. Fuzzy Syst.* **2019**, *28*, 1542–1556. [CrossRef]
24. Fusca, M.; Negrini, F.; Perego, P.; Magoni, L.; Molteni, F.; Andreoni, G. Validation of a wearable IMU system for gait analysis: Protocol and application to a new system. *Appl. Sci.* **2018**, *8*, 1167. [CrossRef]
25. Chen, J.; Wu, Y.; Qiao, H. Memory, attention, and muscle synergies based reinforcement and transfer learning for musculoskeletal robots under imperfect observation. *IEEE/ASME Trans. Mechatron.* **2024**, *30*, 1853–1864. [CrossRef]
26. Chen, J.; Wu, Y.; Yao, C.; Huang, X. Robust motion learning for musculoskeletal robots based on a recurrent neural network and muscle synergies. *IEEE Trans. Autom. Sci. Eng.* **2024**, *22*, 2405–2420. [CrossRef]
27. Su, H.; Li, Z.; Li, G.; Yang, C. EMG-Based neural network control of an upper-limb power-assist exoskeleton robot. In *International Symposium on Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 204–211.
28. Amiri, M.S.; Ramli, R. Fuzzy Adaptive Controller of a Wearable Assistive Upper Limb Exoskeleton Using a Disturbance Observer. *IEEE Trans. Hum.-Mach. Syst.* **2025**, *55*, 197–206. [CrossRef]
29. Hsu, H.H.; Lee, W.K.; Lee, P.L. Designing an EEG Signal-Driven Dual-Path Fuzzy Neural Network-Controlled Pneumatic Exoskeleton for Upper Limb Rehabilitation. *Int. J. Fuzzy Syst.* **2025**, *1*–19. [CrossRef]
30. Incremona, G.P.; Cucuzzella, M.; Magni, L.; Ferrara, A. MPC with sliding mode control for the energy management system of microgrids. *Ifac-Papersonline* **2017**, *50*, 7397–7402. [CrossRef]
31. Mondal, A.; Ganguli, S. *Controller Design for Industrial Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2025.
32. Sun, Z.; Li, F.; Duan, X.; Jin, L.; Lian, Y.; Liu, S.; Liu, K. A novel adaptive iterative learning control approach and human-in-the-loop control pattern for lower limb rehabilitation robot in disturbances environment. *Auton. Robot.* **2021**, *45*, 595–610. [CrossRef]
33. Li, Z.; Li, X.; Li, Q.; Su, H.; Kan, Z.; He, W. Human-in-the-loop control of soft exosuits using impedance learning on different terrains. *IEEE Trans. Robot.* **2022**, *38*, 2979–2993. [CrossRef]

34. Aly, A.A.; The Vu, M.; El-Sousy, F.F.; Alotaibi, A.; Mousa, G.; Le, D.N.; Mobayen, S. Fuzzy-based fixed-time nonsingular tracker of exoskeleton robots for disabilities using sliding mode state observer. *Mathematics* **2022**, *10*, 3147. [CrossRef]
35. Kim, H.; Lee, D.H. Stable whole-body control of exoskeletons using 3D motion reconstruction. *Robot. Auton. Syst.* **2019**, *119*, 21–33.

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 Fuzzy Logic-Based eHealth Mobile App for Activity Detection and Behavioral Analysis in Remote Monitoring of Elderly People: A Pilot Study

Abdussalam Salama ^{1,*}, Reza Saatchi ², Maryam Bagheri ¹, Karim Shebani ³, Yasir Javed ¹, Raksha Balaraman ¹ and Kavya Adhikari ¹

¹ School of Computing and Digital Technologies, Sheffield Hallam University, Sheffield S1 1WB, UK; maryam.bagheri@shu.ac.uk (M.B.); y.javed@shu.ac.uk (Y.J.); balaramanraksha@gmail.com (R.B.); adhikarikavya1995@gmail.com (K.A.)

² School of Engineering and Built Environment, Sheffield Hallam University, Sheffield S1 1WB, UK; r.saatchi@shu.ac.uk

³ Rotherham NHS Foundation Trust, Rotherham S60 2UD, UK; kshebani@nhs.net

* Correspondence: abdussalam.salama@shu.ac.uk; Tel.: +44-114-225-6088

Abstract

The challenges and increasing number of elderly individuals requiring remote monitoring at home highlight the need for technological innovations. This study devised an eHealth mobile application designed to detect abnormal movement behavior and alert caregivers when a lack of movement is detected for an abnormal period. By utilizing the built-in accelerometer of a conventional mobile phone, an application was developed to accurately record movement patterns and identify active and idle states. Fuzzy logic, an artificial intelligence (AI)-inspired paradigm particularly effective for real-time reasoning under uncertainty, was integrated to analyze activity data and generate timely alerts, ensuring rapid response in emergencies. The approach reduced development costs while leveraging the widespread familiarity with mobile phones, facilitating easy adoption. The approach involved collecting real-time accelerometry data, analyzing movement patterns using fuzzy logic-based inferencing, and implementing a rule-based decision system to classify user activity and detect inactivity. This pilot study primarily validated the devised fuzzy logic method and the functional prototype of the mobile application, demonstrating its potential to leverage universal smartphone accelerometers for accessible remote monitoring. Using fuzzy logic, temporal and behavioral symmetry in movement patterns were adapted to detect asymmetric anomalies, e.g., abnormal inactivity or falls. The study is particularly relevant considering lonely individuals found deceased in their homes long after dying. By providing real-time monitoring and proactive alerts, this eHealth solution offers a scalable, cost-effective approach to improving elderly care, enhancing safety, and reducing the risk of unnoticed deaths through fuzzy logic.

Keywords: eHealth; elderly care; remote monitoring; activity detection; behavioral analysis; symmetric movements; fuzzy logic; accelerometer

1. Introduction

The well-being of the growing elderly population is becoming an increasingly important social and governmental concern [1]. A significant proportion of frail and lonely older adults prefer to stay in their own homes rather than moving to residential care

establishments, as relocation could be both financially burdensome and psychologically distressing [2]. Aging can cause various complications, particularly for those living alone. Studies indicate a rising trend of older adults found deceased in their homes in England and Wales, unnoticed long after passing away. These so-called “undefined deaths” have been steadily increasing since 1980, linked to social isolation, austerity measures, and societal fragmentation [3]. As reported by a study, “many people would be shocked that someone can lie dead at home for days, weeks, or even longer without anyone raising an alarm” [4]. This growing crisis highlights a critical need for technological developments that can detect abnormal inactivity and provide timely alerts to caregivers.

Advancements in mobile technology offer potential solutions to this issue. Mobile phones are widely used across populations and contain built-in sensors, such as accelerometers, which, when combined with appropriate software, can monitor movement and detect anomalies in daily activity patterns [5,6]. Despite these opportunities, existing Internet of Things (IoT) deployments for healthcare, including innovative smart home environments and sensor networks specifically aimed at elderly monitoring, often present limitations [7,8]. These systems, while providing continuous data collection and alert capabilities, frequently face hurdles related to installation complexity, power management, and user acceptance, particularly for solutions requiring dedicated infrastructure beyond a conventional mobile phone [9]. While recent advances in intelligent monitoring have leveraged powerful AI-driven models such as hybrid deep learning architectures (e.g., convolutional neural networks and long short-term memory neural networks) to improve anomaly detection [10–12], there remains a significant research gap for a highly accessible, unobtrusive, and computationally efficient system suitable for real-time monitoring directly on conventional smartphones.

This study addresses this gap by introducing a novel remote monitoring system designed to enhance the safety and well-being of elderly individuals living independently. A key innovation of this system is its integration of fuzzy logic, an artificial intelligence (AI)-inspired approach, to analyze movement data and accurately detect abnormal patterns, such as prolonged inactivity. Fuzzy logic has been widely applied in various domains, including wireless computer networks, for identifying irregular patterns [13,14]. Our approach uniquely harnesses fuzzy logic due to its interpretability, low computational footprint, and suitability for real-time processing directly on standard mobile devices. This makes our system exceptionally cost-effective, non-wearable, and user-friendly, operating unobtrusively in the background without requiring undue technological expertise from users. Proactive remote monitoring not only enhances the safety and well-being of elderly individuals but also provides peace of mind for caregivers and family members, ensuring timely intervention in critical situations.

The primary objective of this pilot study is to develop and validate a functional prototype of a smartphone-based remote monitoring application that utilizes a novel fuzzy logic methodology to classify and detect abnormal movements, specifically prolonged inactivity. This system is intended to automatically dispatch timely alerts to designated caregivers or family members, thereby providing a simple, cheap, and easy solution to prevent unnoticed medical emergencies and directly address the concerning issue of elderly individuals dying unnoticed at home for extended periods.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of related studies on elderly monitoring and anomaly detection. Section 3 details the methodology of our proposed fuzzy logic-based system. Section 4 presents and discusses the experimental results. Section 5 includes a discussion of our findings. Section 6 outlines the study’s limitations and suggests future directions, while Section 7 concludes the paper.

2. Literature Reviews

2.1. Remote Patient Monitoring

Remote patient monitoring (RPM) is a modern healthcare tool that enables healthcare providers and caregivers to track a patient's health from a distance. Utilizing devices like smartphones, wearable sensors, or other technologies, RPM collects vital health data, such as heart rate, blood pressure, or activity levels, and transmits it to healthcare providers in real time. This capability aids in the early detection of potential problems, allowing for timely intervention before conditions worsen. For patients, particularly those with chronic conditions or recovering from illness, RPM offers a convenient way to maintain a connection with their care team, reducing the need for frequent hospital visits. It can be an effective means to improve health outcomes and make healthcare more accessible [15]. Technological advancements have increasingly enabled patients to engage in normal daily activities at home while still being monitored through modern communication methods [3,16].

RPM systems facilitate timely emergency responses, contributing to reduced untimely deaths and healthcare costs through various communication technologies [17]. Remote monitoring has proven useful in predicting diseases like diabetes and hypertension [18], and in overseeing physiological signs [19], pulmonary disease [20], chronic illnesses [21], and human activities [22]. These systems target diverse subgroups, including individuals diagnosed with chronic illnesses, those with mobility issues or other disabilities, post-surgery patients, neonates, and elderly patients [15,16].

However, the main challenges with existing RPM approaches include variability in the accuracy of data collected by devices, which can lead to potential misdiagnosis or missed alerts. Patients, especially older adults or those with limited technological skills, may struggle to use RPM devices effectively. Furthermore, RPM can increase the workload for healthcare providers, potentially leading to burnout or inefficiencies. While RPM shows promise, long-term data on its comprehensive effectiveness in improving health outcomes or significantly reducing healthcare costs remains limited. RPM devices may also face technical challenges such as restricted battery life, connectivity issues, or inconsistent performance in diverse environments. Prior studies have additionally demonstrated that the accuracy of these methods can decline significantly when applied to both institutionalized and independent elderly individuals [17]. To address these shortcomings, research trends in RPM are increasingly focusing on solutions that leverage readily available technology for specific needs like movement detection and analysis in home environments. By exploring approaches that utilize built-in sensors of conventional mobile phones and integrate advanced logic for activity analysis, studies aim to provide user-friendly, cost-effective, and non-intrusive solutions tailored to the elderly population, thereby enhancing the effectiveness and accessibility of remote patient monitoring.

2.2. Mobile Apps for Remote Monitoring

Traditional monitoring systems often rely on body-attached sensors, which can limit patient mobility and affect daily activities. This discomfort may distort physiological readings, especially among elderly patients, potentially misrepresenting their actual health conditions [17]. To overcome these limitations, the use of mobile devices equipped with a variety of built-in sensors—such as GPS, vision, audio, light, temperature, direction, and acceleration—has created significant opportunities for data mining applications in health monitoring [23–25]. Accurate body motion tracking with sensors typically requires strategic placement, ideally near the body's center of mass (e.g., breastbone, lower back, or waist) to maximize data accuracy [22]. Despite the potential of RPM, current devices often present challenges, including prohibitive costs that limit accessibility. Furthermore, their designs may not adequately address the diverse needs, experiences, and limita-

tions of individual users, hindering effective adoption and long-term use. Smartphones, equipped with accelerometers and gyroscopes, offer a readily available option, and their common placement in pockets allows for effective motion tracking. However, further research is needed to establish their comprehensive accuracy and reliability for diverse monitoring applications.

The development of mobile applications for RPM running on tablets and smartphones eliminates the requirement for expensive, dedicated hardware [26]. Such applications also enable patients to report symptoms and notify medical teams promptly if conditions worsen, facilitating timely healthcare decisions and reducing the need for in-person visits and associated delays [26]. Several frameworks exist for mobile application development. Researchers have developed RPM applications using Java under the ADT (Android Development Toolkit by Google) [27]. Despite its advantages, Java has limitations, such as a lack of inherent cross-platform integration [28]. This limitation can be addressed by using cross-platform frameworks like Xamarin or React-Native, maintained by Microsoft and Facebook, respectively [29]. For example, the Xamarin framework enables developers to create high-performance, native-like mobile applications for Android and iOS from a single C# codebase, significantly reducing development time and effort. It also offers seamless integration with the .NET ecosystem and access to native APIs, ensuring robust and feature-rich applications [29].

2.3. Accelerometry Operation

Research in activity monitoring increasingly focuses on non-invasive sensors, which overcome the limitations of invasive counterparts, such as subject discomfort [30]. Researchers, such as [25], have investigated the utilization of multiple sensors, including microphones, accelerometers, GPS, and cameras, embedded in commercial smartphones for recognizing activities and enhancing mobile social networking applications. Among these, the accelerometer sensor, integrated as part of a mobile device, is particularly appealing due to its unobtrusive nature, its ability to gather data without extra equipment, and its potential for precise interpretation [24].

Human activity detection using a triaxial accelerometer typically involves three stages: pre-processing, feature extraction, and classification [31]. Feature extraction commonly involves deriving acceleration peaks, variance, fundamental frequencies' magnitudes, intensity, and position relations [24,31]. Studies in this area often utilize magnitude accelerometry features to predict human movement or its absence, classifying it as normal or abnormal.

The ubiquity of smartphones presents a compelling opportunity for RPM, particularly given the advancements in accelerometer technology integrated within these devices. Leveraging existing hardware like smartphones reduces the burden on both patients and healthcare systems by eliminating the need for specialized, often costly, equipment [32]. The increasing integration of mobile devices, connected objects, and sensors into healthcare applications highlights the viability of smartphone-based monitoring for diverse patient populations [33]. This technological availability plays a crucial role in enabling the scalability and feasibility of remote monitoring programs [33].

Modern smartphone accelerometers, although not classified as medical-grade devices, can offer effective operation for numerous remote monitoring applications, especially those focused on gross motor activity, falls, and certain movement disorders [34]. Numerous studies have directly compared smartphone accelerometry data with data obtained from research-grade wearable sensors or gold-standard motion capture systems. These studies often assess metrics such as fall detection accuracy, step count accuracy, and activity recognition accuracy. For instance, research has shown that smartphone accelerometers can

achieve high sensitivity and specificity in fall detection algorithms [34]. Similarly, studies have explored the use of these accelerometers for monitoring physical activity levels, gait patterns, and even subtle tremors in patients with conditions like Parkinson's disease [35]. While it is essential to acknowledge the limitations of smartphone accelerometers in detecting very fine movements or subtle changes, the level of accuracy they provide is often sufficient for many remote monitoring applications. A balanced discussion of these limitations strengthens the argument for their use by demonstrating a thorough understanding of the technology, including ongoing efforts to improve reliability and reduce issues such as false fall alerts through advanced techniques like deep learning [36].

The real-time or near real-time data acquisition capabilities of smartphone accelerometers enable rapid detection of critical events, such as falls, and can trigger timely interventions. Smartphone processors are powerful enough to handle accelerometry data and execute algorithms in real-time, facilitating immediate feedback and alerts. Furthermore, the inherent connectivity of smartphones allows for seamless data transmission to healthcare providers or caregivers. For example, systems can be designed to automatically detect a fall and send an alert to designated contacts [37]. Real-time activity data can also be used to provide personalized feedback and reminders to encourage adherence to exercise programs.

2.4. Fuzzy Logic in Activity Detection

As discussed by researchers [31], the classification stage is crucial in detecting human activity. Researchers [23] employed the k-nearest algorithm (KNN) to detect five human activities like phone detached, idle, walking, running, and jumping. The KNN classifier, although straightforward and widely employed, can achieve performance comparable to that of more sophisticated classifiers [37]. Researchers such as [37,38] employed convolutional neural networks (CNNs) for predicting six human activities, including jogging, walking, going upstairs, going downstairs, sitting, and standing. Deep learning models like CNNs have demonstrated notable accuracy potential but necessitate more training data and computational resources.

To address limitations observed in existing RPM approaches, fuzzy logic has emerged as a promising method for interpreting accelerometry data. Fuzzy logic offers advantages in modeling complex systems with imprecise data by allowing variables to have degrees of truth, which enhances its suitability for real-world applications. Fuzzy logic can perform well with smaller datasets because it relies on fuzzy rules and membership functions rather than extensive training datasets. It can generalize effectively even with limited data, making it suitable for applications where data collection is challenging or expensive. Additionally, fuzzy logic systems are interpretable, allowing experts to understand and refine the rules and membership functions to improve system performance [39]. Collected data can undergo analysis using fuzzy logic to identify human activities. In cases where anomalies are detected, alerts would be promptly sent to the emergency contact people added in the mobile application. This approach leverages fuzzy logic's capability to interpret and respond to complex, uncertain data patterns, ensuring timely intervention and enhanced monitoring effectiveness [40].

In summary, while many studies highlight the growing field of Remote Patient Monitoring (RPM), offering broad insights into its potential, they often do so without delving into specific, cost-effective solutions for individual behavioral monitoring using readily available technology. For instance, reviews like Shaik et al. [41] broadly cover how AI and technologies such as IoT wearables can transform RPM. Holtz et al. [42] focus on the promise of RPM from a strategic, policy, and infrastructure standpoint, discussing its role in addressing healthcare inequities. Similarly, systematic reviews by Tan et al. [43] examine

the impacts of various RPM interventions, including smartphone apps and wearables, on patient safety, adherence, and cost-related outcomes during care transitions. While these papers provide valuable overviews and impact analyses, they do not focus on the specific, practical needs our study addresses. For example, systems like the one in Zhang et al. [44] do use smartphones, but they often rely on external wearable sensors for physiological data, which adds cost and complexity. Another advanced approach by Chinnaperumal et al. [45] integrates complex AI with 5G networks for secure medical data acquisition, focusing on high-tech infrastructure. In sharp contrast to these approaches, our study does not use any separate IoT devices or external wearables. Instead, we uniquely leverage only the built-in accelerometer of a standard smartphone, making our system highly affordable and accessible. Our research further distinguishes itself by applying fuzzy logic—an AI method well-suited for real-time decision-making under uncertainty—to precisely detect abnormal inactivity (lack of movement). This specific focus directly addresses a critical, often overlooked, safety concern: preventing unnoticed deaths of lonely individuals, thereby offering a practical, user-friendly, and life-saving application within the broader RPM landscape.

3. Methods

This study employed quantitative and experimental methodologies to develop and validate a smartphone-based remote patient monitoring system. Our primary goal was to create a user-friendly and cost-effective eHealth solution capable of detecting abnormal inactivity and potential falls in elderly individuals, triggering timely alerts to caregivers. This section details the system’s architecture, mobile application development, data acquisition, the core fuzzy logic-based anomaly detection algorithm, and the notification system.

3.1. System Overview and Architecture

Our proposed system operates on a conventional mobile phone, leveraging its built-in accelerometer to continuously monitor user activity. Figure 1 illustrates the overarching workflow. The system collects movement data in real-time, processes it using a fuzzy logic module, and then dispatches immediate alerts to designated caregivers if abnormal activity (prolonged inactivity or a fall) is detected. If activity is deemed normal, the system remains unobtrusive and takes no action.

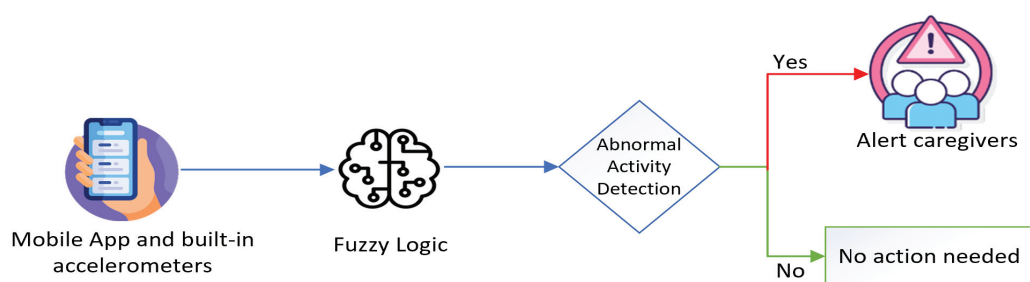


Figure 1. Workflow of the proposed system.

The mobile application was developed using the ASP.NET Xamarin framework (version 5.0.0, Microsoft Corporation, Redmond, WA, USA) with C# as the primary programming language, and deployed on Samsung Galaxy 5G smartphones (Samsung, Suwon, Republic of Korea), with testing specifically conducted on models such as the Galaxy A53 5G, A73 5G, and M54 5G running Android version 12.0.0. It integrates a Python-based fuzzy logic engine (Python 3.10.x, Python Software Foundation, Beaverton, OR, USA) for intelligent decision-making, utilizing scikit-fuzzy (version 0.4.1), a common Python fuzzy logic library. Accelerometer data, collected by the Xamarin.Essentials.Accelerometer API (version 1.7.0, Microsoft Corporation, Redmond, WA, USA), are processed to extract the

Signal Magnitude Vector (SMV) and timestamp. These data points are stored locally on the phone's internal memory using SQLite (version 3.39.x, open-source). The stored data then serves as inputs to the Python fuzzy logic engine, which analyzes movement patterns based on predefined static rules to classify user activity and detect anomalies.

3.2. Mobile Application Development and Features

The mobile application was developed with user-friendliness and continuous, unobtrusive monitoring in mind, particularly for elderly users. We used the ASP.Net Xamarin framework with C# as the primary programming language, allowing for cross-platform development targeting Android devices. Testing was conducted on various Samsung 5G smartphones (e.g., Galaxy A53 5G, A73 5G, and M54 5G) running Android version 12.0.0, all equipped with built-in accelerometers and gyroscopes.

Designed for minimal user interference, the app operates seamlessly in the background, continuously collecting accelerometry data and performing activity analysis without requiring constant interaction. Upon installation, users grant necessary permissions and receive essential information about the app's functionality. The application's interface, exemplified by Figure 2, prominently displays participant details (name, age, and gender) at the top, along with real-time motion activity graphs for the current day. To facilitate long-term health management and trend analysis, it also generates historical motion activity graphs for one week, one month, and three months. In these activity graphs, green lines illustrate a detected moving state (activity level plotted on the y-axis against time on the x-axis), while red lines signify an idle (not moving) state. This aggregated activity information is subsequently fed into the fuzzy logic module for classification and alert decision-making. Figure 3 presents the interface for managing caregiver contact details, including name, phone number, email address, and relationship to the monitored individual.

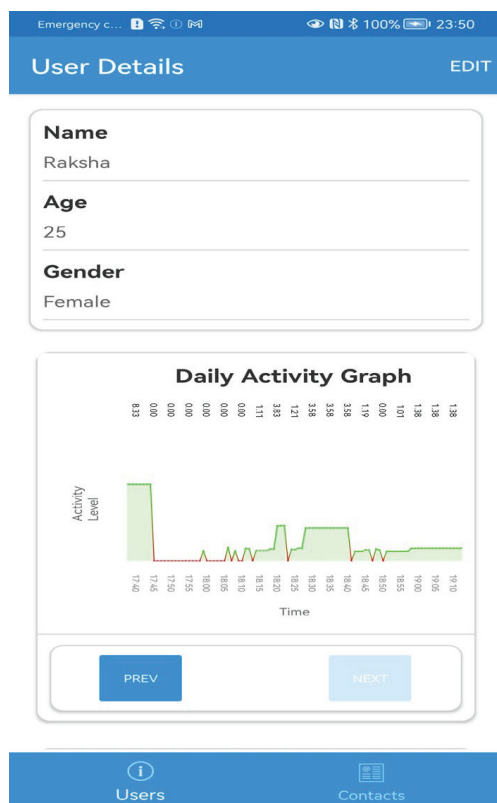


Figure 2. Activity and participant details.

Emergency c... 100% 17:16

← Contact Details EDIT

Name
Emma

Relationship
Daughter

Email
emmagreen@gmail.com

Phone Number
7843091685

Users Contacts

Figure 3. View/edit caregiver contacts.

Further enhancing user control and safety, Figure 4 shows all configured caregiver (emergency) contacts, providing a comprehensive overview of designated recipients for alerts. Additionally, the application includes a prominent emergency button (shown in red in Figure 5), enabling elderly users to manually send immediate alerts to caregivers in critical situations.

Emergency c... 100% 17:17

Emergency Contacts ADD

Sankar Caretaker	DELETE
Rachel Son	DELETE
Joey Daughter	DELETE
Emma Daughter	DELETE

Users Contacts

Figure 4. Caregiver (emergency) contacts.

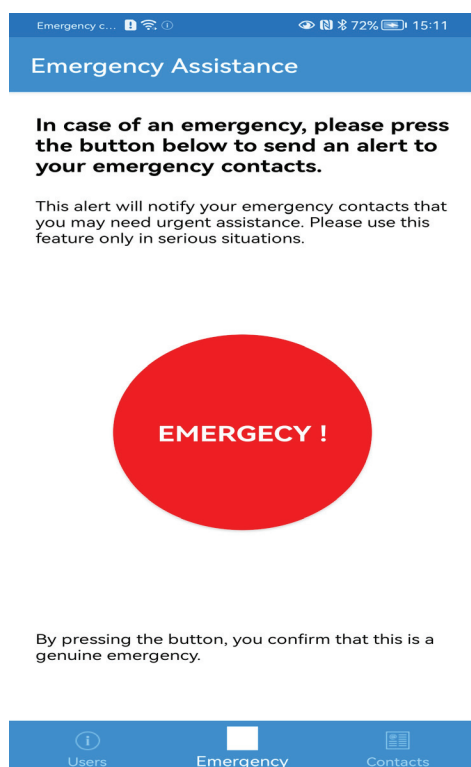


Figure 5. Emergency button.

3.3. Accelerometer Data Acquisition and Local Storage

Accelerometer data, essential for monitoring user activity, are collected using Xamarin.Essentials.Accelerometer (version 1.7.0, Microsoft Corporation, Redmond, WA, USA), which provides processed x , y , and z values along with corresponding timestamps. These raw accelerometer data are pre-processed to extract the Signal Magnitude Vector (SMV), calculated using

$$SMV = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

For real-time monitoring, accelerometer data are collected at one-minute intervals (every 60 s). For each one-minute interval, only the highest Signal Magnitude Vector (SMV) recorded within that minute was retained. This single, highest SMV value per minute then served as the primary input feature for the fuzzy logic classification system.

A key advantage of this design is that all data are stored locally on the smartphone using SQLite, a lightweight, open-source, serverless database engine. This approach eliminates reliance on cloud storage, which may be susceptible to Internet connectivity issues, and significantly enhances data privacy and security by keeping sensitive information confined to the user's device, reducing the risk of unauthorized access.

The SQLite database stores the three-dimensional acceleration data, the calculated SMV , and the corresponding timestamps. This method optimizes storage demands while preserving meaningful information for analysis. The database is designed to hold up to three months of historical acceleration data, enabling long-term activity tracking and pattern analysis valuable for identifying behavioral trends or sudden changes. To maintain optimal performance and storage capacity, data older than three months are automatically deleted. This design balances data privacy, local storage constraints, and real-time processing efficiency, ensuring the application remains responsive on mobile devices.

3.4. Fuzzy Logic-Based Activity and Anomaly Detection

The developed mobile application, integrated into conventional smartphones, continuously monitors user activity using its built-in accelerometer to collect movement data. This data is then processed via a fuzzy logic module to detect abnormal activity, such as prolonged inactivity or falls, enabling informed alert decisions. The system's overall workflow is presented and generally explained in Section 3.1 and visualized in Figure 1.

This section details the fuzzy logic system, which forms the core AI for classifying activity levels and detecting anomalies. The system operates on a static set of fuzzy rules and membership functions, optimized for real-time performance on resource-constrained mobile devices. The system utilizes two primary fuzzy inputs: the Signal Magnitude Vector (SMV), derived from accelerometer data as described in Section 3.3, and the time of day input.

Before the main fuzzy logic processing, an initial classification algorithm processes raw accelerometer data to identify immediate abnormal movements (like falls) and categorize general activity states (moving or still). Figure 6 presents the flowchart for this developed classification algorithm (where μ = Mean).

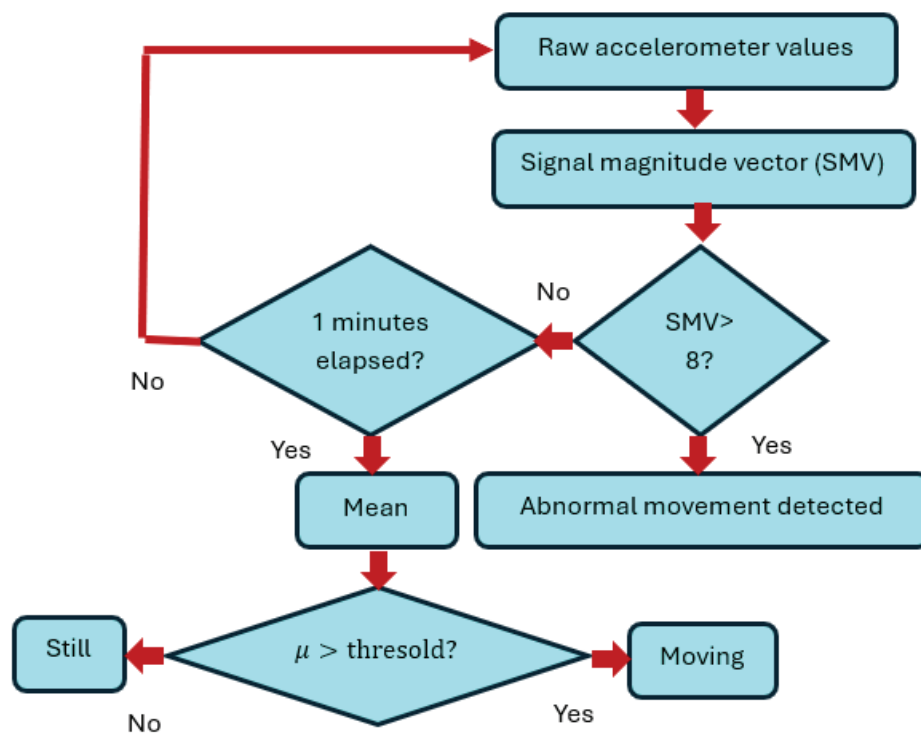


Figure 6. Flowchart of developed classification algorithm (μ = Mean).

The process of detecting activity and potential anomalies starts by continuously gathering raw movement data from the smartphone's built-in accelerometer. This raw data is immediately used to calculate the Signal Magnitude Vector (SMV), which is a single number representing the overall strength or intensity of the user's movement at any given moment. The system first performs a quick check: If this SMV value instantly jumps above 8, it is considered a sudden, strong impact, like a fall, and an "Abnormal movement detected" alert is triggered right away for a quick response. If the SMV does not jump above 8, meaning no immediate fall is detected, the system then begins to monitor activity over time. It continuously collects SMV data and watches if one minute has passed. If not, it keeps gathering more data until a full minute is complete. Once a minute has passed, the system calculates the average (or mean, represented by μ) of all the SMV values collected during that minute. This average SMV provides a smoother picture of the user's activity

level during that time. Finally, this average SMV (μ) is compared to a set crisp threshold to define the user's preliminary activity state. If the average movement (μ) is above this crisp threshold, the user is classified as "Moving", indicating they are active. If the average movement is at or below the crisp threshold, the user is classified as "Still", indicating a period of inactivity. This initial classification of "Moving", "Still", or "Abnormal movement detected" provides the basic information that feeds into the more advanced fuzzy logic system. In the fuzzy logic module (as detailed in upcoming sections describing fuzzy sets and rules), these states and SMV values are further processed to determine nuanced activity levels and trigger alerts based on complex conditions, such as prolonged inactivity.

3.4.1. Fuzzification

Fuzzification is the process of converting crisp, numerical input values (like SMV or time of day) into degrees of membership for predefined fuzzy sets. For this system, standard Triangular and Trapezoidal membership functions were employed, as commonly found in Python fuzzy logic libraries like scikit-fuzzy. The parameters defining the precise shape and range of these membership functions were initially established using an "Intuition method" and subsequently refined through manual adjustment based on empirical observations and optimization of detection performance.

Activity Level Fuzzification (SMV Input)

The initial SMV (Signal Magnitude Vector) input undergoes fuzzification, transforming its precise numerical values into degrees of membership across three overlapping fuzzy sets: Low, Medium, and High activity levels. This process allows for a more natural representation of activity and enables smoother transitions between states compared to rigid thresholds. These fuzzy sets and their defining parameters are empirically chosen based on best practices in activity monitoring to accurately reflect varying human movement intensities. Table 1 details the type of membership function used for each set and provides their specific parameters, along with a "Corresponding General SMV Range" for approximate clarity.

Table 1. Illustrative parameters for activity level membership functions.

Fuzzy Set	Membership Function Type	Defining Parameters (SMV Values)	Corresponding General SMV Range
Low	Trapezoidal	[0, 0, 0.5, 1.5]	$SMV \approx 0$ to 1.5
Medium	Triangular	[1, 4.5, 8.5]	$SMV \approx 1.5$ to 8.5
High	Trapezoidal	[8, 9, 20, 20]	$SMV \approx > 8$

For example, the "Low" fuzzy set uses a Trapezoidal function with parameters [0, 0, 0.5, 1.5], meaning SMV values from 0 to 0.5 have full "Low" membership, gradually decreasing to zero by 1.5. The "Medium" fuzzy set, a Triangular function with parameters [1, 4.5, 8.5], primarily covers typical daily activity, with its full membership peaking at an SMV of 4.5. Lastly, the "High" fuzzy set, also a Trapezoidal function with parameters [8, 9, 20, 20], is specifically designed for high-intensity movements indicative of a fall, gaining full membership from an SMV of 9 upwards. These overlapping fuzzy definitions allow the system to make more nuanced and robust decisions.

Time of Day Fuzzification

The "Time of Day" input is fuzzified into three distinct fuzzy sets: Night, Day, and Evening. This process allows the system to apply different logic based on the typical behavioral patterns expected during various periods of a 24 h cycle. These sets utilize static

membership functions defined by specific time boundaries, transforming a precise hour into a degree of belonging to “Night”, “Day”, or “Evening”. Table 2 provides illustrative examples of the parameters for these membership functions.

Table 2. Illustrative parameters for time of day membership functions.

Fuzzy Set	Membership Function Type	Example Parameters (Hours 0–24)
Day	Trapezoidal	[22, 23, 5, 6]
Evening	Triangular	[7, 13, 19]
Night	Trapezoidal	[17, 19, 22, 24]

These membership functions define how each hour of the day contributes to the “Day”, “Evening”, or “Night” categories, with overlaps allowing for smooth transitions between periods. For example, the “Day” fuzzy set uses a Trapezoidal membership function with parameters [22, 23, 5, 6]. This means its membership begins to increase from 22:00 (10 p.m.), reaches full (100%) membership between 23:00 (11 p.m.) and 5:00 (5 a.m.), and then gradually decreases to zero by 6:00 (6 a.m.).

The “Evening” fuzzy set uses a Triangular membership function with parameters [7, 13, 19]. For this set, membership starts at 7:00 (7 a.m.), reaches its peak (100%) at 13:00 (1 p.m.), and then drops to zero by 19:00 (7 p.m.).

Lastly, the “Night” fuzzy set, also a Trapezoidal function, has parameters [17, 19, 22, 24]. Here, membership begins to increase from 17:00 (5 p.m.), reaches full membership between 19:00 (7 p.m.) and 22:00 (10 p.m.), and then gradually decreases to zero by midnight (24:00). These specific fuzzy definitions enable the system to adapt its behavior and alert strategies to different periods of the day as defined by these sets.

3.4.2. Fuzzy Inference System (FIS)

The fuzzified inputs are fed into the Fuzzy Inference System. This system employs a Mamdani-type inference engine, which is widely used for its interpretability. It processes the inputs using a static set of nine predefined “IF-THEN” fuzzy rules. These rules, detailed in Table 3, combine the fuzzified activity level (Low, Medium, High) and time of day (Night, Day, Evening) to infer the appropriate output, such as “Normal Activity”, “Inactivity Detected”, or “Fall Detected”.

Table 3. Fuzzy rules used by the tool.

Rule	Activity Level	Time of Day	Activity Output	Action
1	Low	Day	Abnormal	Start 4 h timer
2	Medium	Day	Normal	None
3	Low	Evening	Abnormal	Start 4 h timer
4	Medium	Evening	Normal	None
5	Low	Night	Normal	None
6	Medium	Night	Possibly Normal	None
7	High	Day	High Abnormal	Start 3 s timer
8	High	Evening	High Abnormal	Start 3 s timer
9	High	Night	High Abnormal	Start 3 s timer

The “Action” column in Table 3 defines the system’s immediate response based on the inferred activity output. For instance, the “Start 4-h timer” action is triggered when the system detects a “Low” activity level during “Day” or “Evening” hours (Rules 1 and 3). Instead of immediately triggering an alert, the system initiates a 4 h inactivity timer. This means the system will wait for four continuous hours of low activity; if movement does

not resume within this 4 h period, it indicates prolonged inactivity, and an alert will then be dispatched to caregivers. This timer helps to avoid false alarms for short periods of rest, ensuring alerts are only sent for genuinely concerning extended inactivity.

Conversely, the “Start 3 s timer” action is specifically triggered when a “High” activity level is detected, regardless of the time of day (Rules 7, 8, and 9). A high activity level is indicative of a sudden, forceful movement, strongly suggesting a fall. Upon detecting such high activity, the system immediately starts a 3 s monitoring timer. During these 3 s, the system re-confirms the fall event; if the high activity is sustained or validated within this very short window (e.g., no immediate recovery movement), an emergency alert is rapidly sent to caregivers. This brief timer ensures quick verification while prioritizing swift notification for critical events like falls.

3.4.3. Defuzzification

Defuzzification is the final stage of the fuzzy logic process, as illustrated in the overall architecture shown in Figure 7. Its primary function is to convert the aggregated fuzzy output from the Fuzzy Inference System (FIS), which represents the inferred state (e.g., degree of “Inactivity Detected”), into a single, crisp, and actionable decision that the system can execute. For this specific system, the Centroid (Center of Area) method was employed for defuzzification. This method calculates the center of gravity of the combined output fuzzy set, effectively yielding a precise numerical value that then directly triggers the appropriate actionable response.

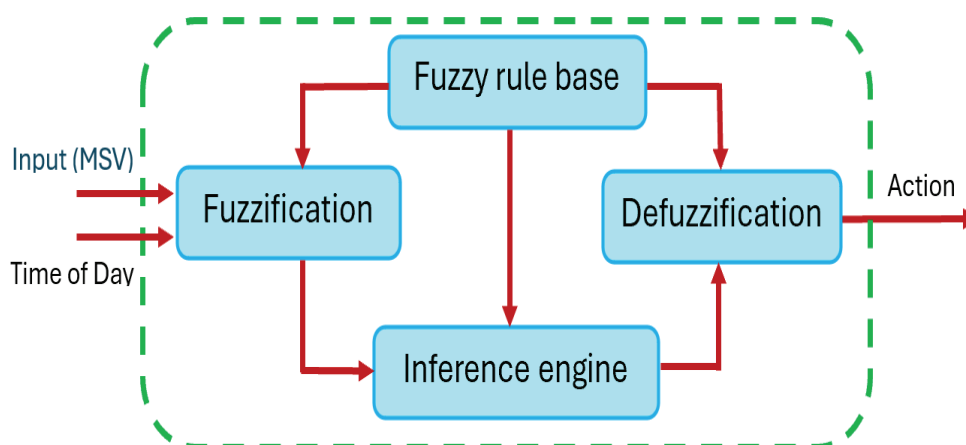


Figure 7. Architecture of the fuzzy logic system.

This crisp numerical output then dictates the system’s subsequent actions, as defined by the fuzzy rules detailed in Table 3. For instance, if the fuzzy logic system identifies a “Low” activity level during “Day” or “Evening” hours (corresponding to Rules 1 and 3 in Table 3), the defuzzified output triggers a 4 h inactivity timer. This timer ensures that an alert is only sent to caregivers if the user remains continuously inactive for this full four-hour period, providing a contextual buffer to avoid false alarms during normal, extended rest. Conversely, if a “High” activity level is detected, indicating a potential fall (corresponding to Rules 7, 8, and 9 in Table 3), the defuzzified output instantly activates a 3 s monitoring timer. This brief timer allows for immediate verification of the high-intensity event; if normal activity does not resume within these three seconds, caregivers are promptly notified of a potential emergency, prioritizing rapid response for critical situations.

3.5. Alert Notification System

When abnormal activity is detected based on the fuzzy logic output and timer thresholds, the system automatically dispatches alerts to designated emergency contacts. These alerts are sent via email using SMTP (Simple Mail Transfer Protocol) to all contacts specified by the individual during the application's setup. Additionally, an SMS notification is dispatched to the primary carer's phone number. Figure 8 provides an example of an email alert dispatched by the system upon detecting an abnormality.

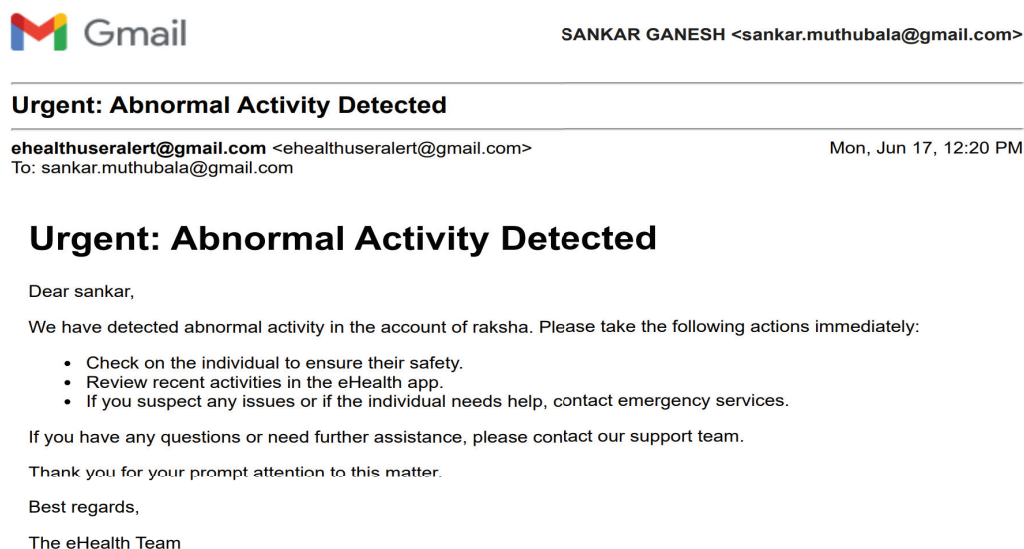


Figure 8. Email alert sent to emergency message when abnormality is detected.

4. Results

This section presents the experimental findings and performance validation of the developed smartphone-based remote monitoring system. The system was rigorously tested with six volunteer participants, comprising university staff and MSc students, who installed and used the application on their mobile phones over a continuous two-month period. These tests were designed to evaluate the system's effectiveness in detecting both abnormal events (such as falls) and prolonged periods of inactivity, triggering timely alerts when appropriate. In total, 123 fall scenarios and 138 idle scenarios were tested across the six participants, as detailed below in Table 4.

Table 4. Number of idle and fall scenarios tested for each user.

User	Number of Fall Scenarios Tested	Number of Idle Scenarios Tested
1	20	28
2	22	26
3	28	20
4	20	18
5	15	22
6	18	24
Total scenarios	123	138

4.1. Evaluation Metrics and Overall Performance

The performance of the eHealth mobile application was primarily evaluated using a confusion matrix, which provided a statistical summary of the model's classification results, highlighting errors and their categories. This matrix is often referred to as the error matrix. Figure 9 illustrates the confusion matrix for the abnormal detection (ABD) system, defining its key components:

		Perfect Classes	
		Positive	Negative
Actual Classes	Positive	True Positive (<i>TP</i>)	False Negative (<i>FN</i>)
	Negative	False Positive (<i>FP</i>)	True Negative (<i>TN</i>)

Figure 9. Confusion matrix.

- True Positive (*TP*): The number of abnormal instances correctly classified as abnormal.
- False Positive (*FP*) (Type I Error): The number of normal instances incorrectly classified as abnormal.
- True Negative (*TN*): The number of normal instances correctly classified as normal.
- False Negative (*FN*) (Type II Error): The number of abnormal instances incorrectly classified as normal.

Accuracy represents the percentage of correct predictions across all samples and is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

The overall performance for both fall and idle detection is summarized in Table 5.

Table 5. Accuracy calculation for fall and idle detection.

Detection Type	True Positives (<i>TP</i>)	False Positives (<i>FP</i>)	True Negatives (<i>TN</i>)	False Negative (<i>FN</i>)	Detection Type
Fall	92	10	15	6	87.70%
Idle	104	7	22	5	91.3%

4.2. Activity Monitoring and Behavioral Trends

The application effectively captured and visualized users' daily, weekly, and monthly physical activity patterns throughout the two-month study. Figure 10 illustrates a representative report of weekly and monthly active hours for User 1, demonstrating the system's capability to track long-term behavioral trends over periods greater than 24 h.

In the "Weekly Active Hours" graph (the top visualization in Figure 10), each vertical bar represents a day of the week. The numerical value displayed at the top of each bar indicates the total active hours for that specific day. Within each bar, the bright green sections consistently represent periods where movement was detected, signifying that the user was actively using or carrying the phone. The lighter, transparent (or uncolored) sections within the bars indicate periods of detected inactivity. Users can navigate through different weeks' reports using the "PREV" and "NEXT" buttons.

The "Monthly Active Hours" graph (the bottom visualization in Figure 10) displays the active hours per day over a period of approximately one month, presented as a line graph. The area shaded in green underneath the line also signifies periods of detected movement, allowing caregivers and users to visualize overarching activity trends and patterns across several weeks or a full month. This comprehensive visual data significantly aids in monitoring activity trends over time, supporting the early detection of significant changes in behavior or health status that might indicate a change in health.

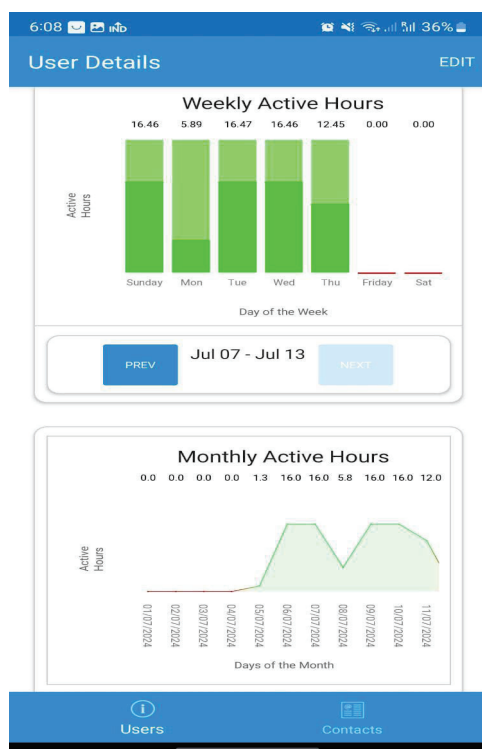


Figure 10. Weekly and monthly report.

4.3. Inactivity Detection Performance

The system's ability to identify prolonged periods of user inactivity was a primary focus of its evaluation. Through quantitative analysis of the 138 idle scenarios tested, the system achieved a high classification performance. As detailed in Table 5, the idle detection demonstrated 104 True Positives, 7 False Positives, 22 True Negatives, and 5 False Negatives, resulting in an overall accuracy of 91.3%.

The integrated fuzzy logic effectively managed alert triggering based on contextual rules. For instance, in accordance with Rule 1 from Table 3, the system consistently triggered an alert when continuous low activity (inactivity) was detected for four consecutive hours during daytime (07:00–18:00). Conversely, adhering to Rule 5 (Table 3), no alerts were dispatched for similar four-hour inactivity periods occurring during nighttime (after 22:00), accurately recognizing typical sleep patterns. The system further demonstrated robustness by automatically resetting the inactivity timer if movement resumed within the four-hour threshold, preventing unnecessary alerts. While the system performed well in classifying expected scenarios, false positives did occur, predominantly when participants' normal behaviors deviated from the predefined fuzzy rules. For instance, in one observed case, a participant who took a longer-than-usual mid-day rest (exceeding the 4 h inactivity threshold during daytime) triggered an alert despite being entirely normal. This highlights the inherent challenge of distinguishing genuine anomalies from individual behavioral variations with static rules.

4.4. Fall Detection Performance

Fall detection, a critical aspect of the system's functionality, was rigorously tested through 123 simulated emergency scenarios. In these controlled tests, the system demonstrated strong performance. As presented in Table 5, the fall detection achieved 92 True Positives, 10 False Positives, 15 True Negatives, and 6 False Negatives, resulting in an overall accuracy of 87.7%. In all simulated fall events that were correctly detected (falling

under the 92 True Positives), the system successfully triggered an emergency alert within three seconds of a detected high-intensity movement.

Figure 11 provides a visual representation of the system’s daily activity monitoring, including an example of a detected fall. At the top of the interface, essential participant details are displayed, providing immediate context for the activity data. The main “Daily Activity Graph” visualizes the “Activity Level” on the Y-axis against “Time” on the X-axis, presented in a 24 h format (e.g., from 20:00 to 24:00). In this graph, continuous green lines represent periods of normal, sustained activity, while sharp red peaks distinctively indicate sudden, forceful changes in movement characteristic of a detected fall, as seen around 23:40 in the example. These sharp red peaks are direct visual evidence of the high-intensity movements that trigger the fall detection mechanism. The corresponding fuzzy rules—Rule 7 (daytime), Rule 8 (evening), and Rule 9 (nighttime) from Table 3—were instrumental in ensuring that the system immediately bypassed the four-hour inactivity threshold for these critical events, directly triggering a 3 s monitoring period for fall confirmation and subsequent alerts.

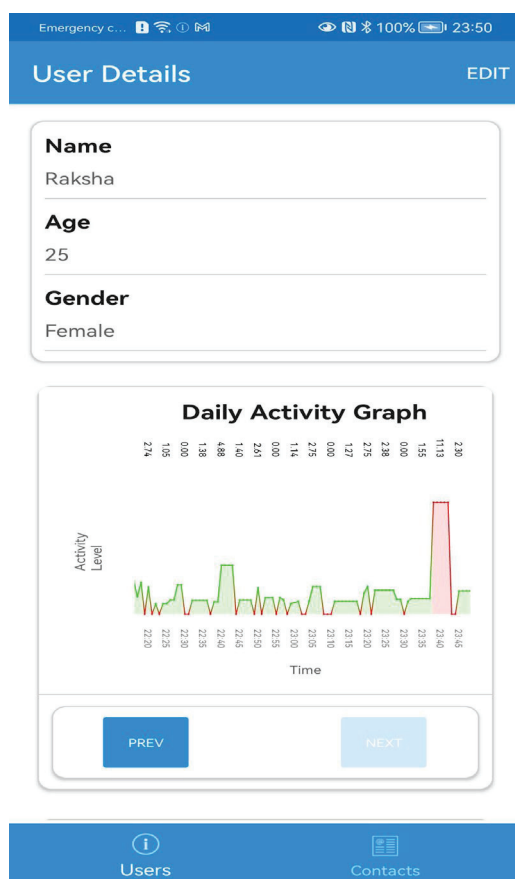


Figure 11. Daily activity.

4.5. Alert Notification Efficacy and Robustness

Upon the detection of a confirmed anomaly (either prolonged inactivity or a fall), the system consistently dispatched timely alerts to designated emergency contacts. These notifications were sent via email (using SMTP) to all registered contacts and via SMS to the primary carer’s phone number. Figure 12 provides a representative example of an email notification triggered by the system upon detecting an abnormality, demonstrating the seamless and automated nature of the alert mechanism.



SANKAR GANESH <sankar.muthubala@gmail.com>

Urgent: Abnormal Activity Detected

ehealthuseralert@gmail.com <ehealthuseralert@gmail.com>
To: sankar.muthubala@gmail.com

Mon, Jun 17, 4:41 AM

Urgent: Abnormal Activity Detected

Dear sankar,

We have detected abnormal activity in the account of raksha. Please take the following actions immediately:

- Check on the individual to ensure their safety.
- Review recent activities in the eHealth app.
- If you suspect any issues or if the individual needs help, contact emergency services.

If you have any questions or need further assistance, please contact our support team.

Thank you for your prompt attention to this matter.

Best regards,

The eHealth Team

Figure 12. Email notification to caregiver triggered at midnight.

Furthermore, the system's robustness was confirmed in various normal usage scenarios. Participants intentionally remained idle for short durations before resuming movement within the set threshold, and the system correctly identified these as non-alert scenarios. This ensured that unnecessary alerts were not triggered during routine, short periods of stillness, thereby enhancing user experience and minimizing false positives. The quantitative results, as detailed in Section 4.1 and Table 5, serve as a strong validation of the proposed fuzzy logic methodology and the prototype application's capability to identify abnormal motion patterns using built-in smartphone accelerometers. These figures, derived from controlled and simulated scenarios with six participants over a two-month period, demonstrate the system's technical feasibility and real-time detection potential within the scope of this pilot study.

5. Discussions

This section interprets the findings of the eHealth mobile application, considering its implications for remote elderly monitoring. The system represents a significant advancement by leveraging readily available mobile phone sensors and fuzzy logic, offering a continuous and non-invasive health monitoring solution. Unlike often expensive and uncomfortable wearable IoT devices, our system's reliance on built-in mobile accelerometers provides a cost-effective and highly accessible approach for tracking daily activity. We opted for fuzzy logic over more complex machine learning models due to its proven efficiency and capability for offline data processing, which enables real-time decision-making without constant Internet connectivity.

One key challenge observed during testing was the occurrence of false alerts. As detailed in the Results Section, these arose when individual activity and sleep behaviors deviated from the static, predefined fuzzy logic rules. For instance, a participant taking an unscheduled, longer-than-usual mid-day rest could trigger an alert despite being entirely normal. This highlights the inherent difficulty in distinguishing genuine anomalies from typical individual behavioral variations with fixed rules, pointing towards the need for more adaptable systems.

The pilot study's results, demonstrating an 87.7% accuracy for fall detection and 91.3% for idle detection, validate the technical feasibility of our fuzzy logic-based methodology.

These figures confirm the prototype application's capability to accurately identify abnormal motion patterns using conventional smartphone accelerometers. The findings highlight the potential of remote patient monitoring (RPM) systems in enhancing elderly care by enabling real-time prolonged inactivity detection and rapid emergency responses, offering a more accessible and scalable alternative to traditional wearable devices.

6. Limitations and Future Directions

A primary limitation of the current study is the small experimental sample size, involving six volunteer participants over a two-month period. This pilot study was primarily designed to validate the feasibility of the fuzzy logic-based methodology and the functional prototype of the eHealth mobile application. While these initial results are promising for confirming the method's working principle and the app's operational capabilities, a comprehensive assessment of its long-term efficacy in enhancing health outcomes or ensuring sustained user adaptation necessitates further work.

Despite the advantages of smartphone-based remote monitoring, we acknowledge inherent technical constraints such as battery life, reliance on network connectivity for alerts, and potential sensor performance variability in diverse environments. Our design deliberately minimizes battery drain by utilizing one-minute interval data sampling and performing all core activity detection and fuzzy logic processing locally on the device. This approach reduces continuous power demands and reliance on constant Internet connectivity.

Future research necessitates expanding the cohort significantly by recruiting a larger and more diverse group of elderly participants to enhance the generalizability of findings and gather more extensive long-term data on various activity patterns. Alongside this, comprehensive usability studies with diverse elderly individuals are crucial to further assess the app's ease of use and refine its interface, complemented by the development of accessible user guides. Future work will also investigate optimized power management techniques and strategies for handling prolonged connectivity outages, such as localized on-device alerts for the user, and explore cloud storage for long-term data retention.

Expanding on the adaptive fuzzy logic, future developments will focus on integrating AI-driven learning to dynamically adjust fuzzy rules based on individual behavioral patterns over a one-week period, thereby significantly reducing false alerts and improving accuracy and personalization. Furthermore, a critical next step involves conducting comprehensive comparative experiments with state-of-the-art deep learning-based activity detection methods, such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. This will allow for a robust evaluation of our fuzzy logic approach against these advanced techniques in terms of accuracy, robustness to diverse user behaviors, and computational complexity, especially considering the resource constraints and real-time processing requirements of mobile devices. The integration of more advanced fuzzy logic techniques, such as an adaptive neuro-Fuzzy Inference System, can further improve the tool's operation. Future enhancements will also include activity trend graphs (daily, weekly, and monthly) to provide caregivers and healthcare professionals with deeper insights into an individual's activity patterns, supporting personalized and preventive care. Finally, exploring integration with iOS devices and expanded activity detection features will enhance accessibility and effectiveness.

7. Conclusions

This pilot study successfully demonstrated the technical feasibility and preliminary effectiveness of a secure eHealth mobile application for remote elderly monitoring. By utilizing triaxial accelerometers from conventional smartphones, it provides a non-invasive,

cost-effective, and user-friendly solution for continuous health tracking. The system, leveraging fuzzy logic, efficiently detected activity patterns and identified prolonged inactivity or potential falls, ensuring timely alerts to caregivers or emergency contacts. The promising results from this initial study, conducted with six participants over two months, validate the core methodology and the functional prototype's capability to classify activities and trigger alerts, thereby laying a foundational step for future extensive evaluations. This mobile-based approach offers a highly accessible and scalable alternative to traditional wearable IoT devices, potentially reducing healthcare costs and the burden on caregivers while offering enhanced safety, improved healthcare outcomes, and peace of mind for families.

Author Contributions: A.S. managed the overall project and took charge of its administration. R.S. focused on the methodology, design, and implementation, particularly the fuzzy logic aspect, while M.B. contributed to designing the security features of the system. K.A. and R.B. were responsible for coding and implementing the technical components of the project. Y.J. led the literature review, providing critical research to support the work. K.S. gathered user feedback and worked on optimizing the system based on those insights. A.S., R.S., K.S. and M.B. worked together to validate and review the manuscript, and all authors were involved in writing, editing, and finalizing the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by an internal Department Funded Research Project grant from Sheffield Hallam University (internal project reference N160). The Article Processing Charge (APC) was funded by the authors.

Institutional Review Board Statement: Ethical approval for this study was obtained from Sheffield Hallam University and reviewed by the University's Research Ethics Committee. The ethics application code: ER76195762 was approved on 25 January 2025. All procedures were conducted in accordance with institutional guidelines and regulations.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The implementation code, datasets, and additional materials related to this study are available as open-source and can be accessed for replication or further research. All supporting resources will be shared upon request or through a designated public repository.

Conflicts of Interest: Author Karim Shebani was employed by The Rotherham NHS Foundation Trust. 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. Ahmed, S.; Irfan, S.; Kiran, N.; Masood, N.; Anjum, N.; Ramzan, N. Remote Health Monitoring Systems for Elderly People: A Survey. *Sensors* **2023**, *23*, 7095. [CrossRef] [PubMed]
2. Kok, L.; Berden, C.; Sadiraj, K. Costs and Benefits of Home Care for the Elderly versus Residential Care: A Comparison Using Propensity Scores. *Eur. J. Health Econ.* **2015**, *16*, 119–131. [CrossRef] [PubMed]
3. Philip, N.Y.; Rodrigues, J.J.; Wang, H.; Fong, S.J.; Chen, J. Internet of Things for In-Home Health Monitoring Systems: Current Advances, Challenges and Future Directions. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 300–310. [CrossRef]
4. Taiwo, O.; Ezugwu, A.E. Smart Healthcare Support for Remote Patient Monitoring during COVID-19 Quarantine. *Inform. Med. Unlocked* **2020**, *20*, 100428. [CrossRef]
5. Soni, V.; Yadav, H.; Semwal, V.B.; Roy, B.; Choubey, D.K.; Mallick, D.K. A Novel Smartphone-Based Human Activity Recognition Using Deep Learning in Health Care. In *Machine Learning, Image Processing, Network Security and Data Sciences*; Springer: Singapore, 2023; pp. 493–503. [CrossRef]
6. Rocha, M.; Nepomuceno, A.C.; Chi, H.R.; Radwan, A.; André, P.; Alberto, N.; Antunes, P.F.d.C.; Domingues, M.d.F. Indoor Localization Using Fiber Bragg Grating-Based Accelerometers for Smart Healthcare. *IEEE Trans. Consum. Electron.* **2023**, *70*, 68–77. [CrossRef]

7. Ali, A.; Montanaro, T.; Sergi, I.; Carrisi, S.; Galli, D.; Distante, C.; Patrono, L. An Innovative IoT and Edge Intelligence Framework for Monitoring Elderly People Using Anomaly Detection on Data from Non-Wearable Sensors. *Sensors* **2025**, *25*, 1735. [CrossRef]
8. Khan, W.; Ahmad, R.; Rizwan, S.; Abid, Z. IoT-Based Elderly Health Monitoring System Using Firebase Cloud Computing. *Wirel. Commun. Mob. Comput.* **2025**, *8*, e70498. [CrossRef]
9. Muñoz Esquivel, K.; Gillespie, J.; Kelly, D.; Condell, J.; Davies, R.; McHugh, C.; Duffy, W.; Nevala, E.; Alamäki, A.; Jalovaara, J.; et al. Factors Influencing Continued Wearable Device Use in Older Adult Populations: Quantitative Study. *JMIR Aging* **2023**, *6*, e36807. [CrossRef]
10. Peng, Z.; Chen, S.; Lin, J.; Chen, W. A method for real-time detection of vessel abnormal behavior based on CNN-LSTM. *Expert Syst. Appl.* **2025**, *288*, 128303. [CrossRef]
11. Zhao, S.; Zhu, J.; Lu, J.; Song, R.; Ma, S. Lightweight Human Behavior Recognition Method for Visual Communication AGV Based on CNN-LSTM. *Int. J. Crowd Sci.* **2025**, *9*, 133–138. [CrossRef]
12. Al-Otaibi, Z.A.; Mohammad, T.K.; Al-Khalidi, Q.H. Enhancing Healthcare Monitoring: A Deep Learning Approach to Human Activity Recognition using Wearable Sensors. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 12933–12938. [CrossRef]
13. Salama, A.; Saatchi, R.; Burke, D. Fuzzy Logic and Regression Approaches for Adaptive Sampling of Multimedia Traffic in Wireless Computer Networks. *Technologies* **2018**, *6*, 24. [CrossRef]
14. Salama, A.; Saatchi, R. Evaluation of Wirelessly Transmitted Video Quality Using a Modular Fuzzy Logic System. *Technologies* **2019**, *7*, 67. [CrossRef]
15. Manani, S.M.; Baretta, M.; Giuliani, A.; Virzì, G.M.; Martino, F.; Crepaldi, C.; Ronco, C. Remote Monitoring in Peritoneal Dialysis: Benefits on Clinical Outcomes and on Quality of Life. *J. Nephrol.* **2020**, *33*, 1301–1308. [CrossRef]
16. Malasinghe, L.P.; Ramzan, N.; Dahal, K. Remote Patient Monitoring: A Comprehensive Study. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 57–76. [CrossRef]
17. Boikanyo, K.; Zungeru, A.M.; Sigweni, B.; Yahya, A.; Lebekwe, C. Remote Patient Monitoring Systems: Applications, Architecture, and Challenges. *Sci. Afr.* **2023**, *20*, e01638. [CrossRef]
18. Chatrati, S.P.; Hossain, G.; Goyal, A.; Bhan, A.; Bhattacharya, S.; Gaurav, D.; Tiwari, S.M. Smart Home Health Monitoring System for Predicting Type 2 Diabetes and Hypertension. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 862–870. [CrossRef]
19. Al Bassam, N.; Hussain, S.A.; Al Qaraghuli, A.; Khan, J.; Sumesh, E.P.; Lavanya, V. IoT Based Wearable Device to Monitor the Signs of Quarantined Remote Patients of COVID-19. *Inform. Med. Unlocked* **2021**, *24*, 100588. [CrossRef]
20. Van Lieshout, F.; Yang, R.; Stamenova, V.; Agarwal, P.; Palma, D.C.; Sidhu, A.; Engel, K.; Erwood, A.; Bhatia, R.S.; Bhattacharyya, O.; et al. Evaluating the Implementation of a Remote-Monitoring Program for Chronic Obstructive Pulmonary Disease: Qualitative Methods from a Service Design Perspective. *J. Med. Internet Res.* **2020**, *22*, e18148. [CrossRef]
21. Shaik, T.; Tao, X.; Higgins, N.; Li, L.; Gururajan, R.; Zhou, X.; Acharya, U.R. Remote Patient Monitoring Using Artificial Intelligence: Current State, Applications, and Challenges. *WIREs Data Min. Knowl. Discov.* **2023**, *13*, e1485. [CrossRef]
22. Palanisamy, P.; Padmanabhan, A.; Ramasamy, A.; Subramaniam, S. Remote Patient Activity Monitoring System by Integrating IoT Sensors and Artificial Intelligence Techniques. *Sensors* **2023**, *23*, 5869. [CrossRef] [PubMed]
23. Hung, W.-C.; Shen, F.; Wu, Y.-L.; Hor, M.-K.; Tang, C.-Y. Activity Recognition with Sensors on Mobile Devices. In Proceedings of the 2014 International Conference on Machine Learning and Cybernetics, Lanzhou, China, 13–16 July 2014; pp. 449–454. [CrossRef]
24. Wang, X.; Kim, H. Detecting User Activities with the Accelerometer on Android Smartphones. *J. Multimed. Inf. Syst.* **2015**, *2*, 207. [CrossRef]
25. Cornacchia, M.; Ozcan, K.; Zheng, Y.; Velipasalar, S. A Survey on Activity Detection and Classification Using Wearable Sensors. *IEEE Sens. J.* **2017**, *17*, 386–403. [CrossRef]
26. George, L.A.; Cross, R.K. Remote Monitoring and Telemedicine in IBD: Are We There Yet? *Curr. Gastroenterol. Rep.* **2020**, *22*, 12. [CrossRef]
27. Bessmeltsev, V.P.; Katasonov, D.N.; Mazurok, B.S.; Makeev, I.V.; Sluev, V.A.; Morozov, V.V.; Shevela, A.I. Mobile System for Automated Remote Monitoring of Cardiac Activity. *Biomed. Eng.* **2015**, *49*, 7–11. [CrossRef]
28. Abolfazli, S.; Sanaei, Z.; Gani, A.; Xia, F.; Yang, L.T. Rich Mobile Applications: Genesis, Taxonomy, and Open Issues. *J. Netw. Comput. Appl.* **2014**, *40*, 345–362. [CrossRef]
29. Kousar, N.; Malik, M.S.; Sarwar, A.; Mohy-ud-din, B.; Shahid, A. Software Engineering: Challenges and Their Solution in Mobile App Development. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 267–272. [CrossRef]
30. Villena Gonzales, W.; Mobashsher, A.T.; Abbosh, A. The Progress of Glucose Monitoring—A Review of Invasive to Minimally and Non-Invasive Techniques, Devices and Sensors. *Sensors* **2019**, *19*, 800. [CrossRef]
31. Sucerquia, A.; López, J.D.; Vargas-Bonilla, J.F. Real-Life/Real-Time Elderly Fall Detection with a Triaxial Accelerometer. *Sensors* **2018**, *18*, 1101. [CrossRef]
32. Jafleh, E.A.; Alnaqbi, F.A.; Almaeeni, H.A.; Faqeeh, S.; Alzaabi, M.A.; Al Zaman, K. The Role of Wearable Devices in Chronic Disease Monitoring and Patient Care: A Comprehensive Review. *Cureus* **2024**, *16*, e68921. [CrossRef]

33. Mondragón-González, S.L.; Burguière, E.; N'diaye, K. Mobile Devices, Connected Objects, and Sensors. In *Machine Learning for Brain Disorders*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 355–388. [CrossRef]
34. Stampfler, T.; Elgendi, M.; Fletcher, R.R.; Menon, C. Fall Detection Using Accelerometer-Based Smartphones: Where Do We Go from Here? *Front. Public Health* **2022**, *10*, 996021. [CrossRef] [PubMed]
35. Toffersson, A.; Lindén, M. A Systematic Review of Wearable Sensors for Monitoring Physical Activity. *Sensors* **2022**, *22*, 573. [CrossRef] [PubMed]
36. Mishra, R. Reducing False Fall Alerts in Fall Detection Using Deep Learning Models. Ph.D. Thesis, University of Missouri, Columbia, MO, USA, 2024. [CrossRef]
37. Ignatov, A. Real-Time Human Activity Recognition from Accelerometer Data Using Convolutional Neural Networks. *Appl. Soft Comput.* **2018**, *62*, 915–922. [CrossRef]
38. Jordao, A.; Torres, L.A.B.; Schwartz, W.R. Novel Approaches to Human Activity Recognition Based on Accelerometer Data. *Signal Image Video Process.* **2018**, *12*, 1387–1394. [CrossRef]
39. Peckol, J.K. *Introduction to Fuzzy Logic*; Wiley: Hoboken, NJ, USA, 2021.
40. Panja, S.; Chattopadhyay, A.K.; Nag, A.; Singh, J.P. Fuzzy-Logic-Based IoMT Framework for COVID-19 Patient Monitoring. *Comput. Ind. Eng.* **2023**, *176*, 108941. [CrossRef]
41. Ullah, F.; Khan, F.; Abdullah; Khan, M.I.; Ashraf, M.U.; Bibi, S.; Nadeem, A. Artificial Intelligence: Current State, Applications, and Challenges. *Comput. Intell. Neurosci.* **2022**, *2022*, 8415710. [CrossRef]
42. Holtz, B.E.; Urban, F.A.; Oesterle, J.; Blake, R.; Henry, A. The Promise of Remote Patient Monitoring. *Telemed. E-Health* **2024**, *30*, 31–38. [CrossRef]
43. Tan, S.Y.; Sumner, J.; Wang, Y.; Yip, A.W. A Systematic Review of the Impacts of Remote Patient Monitoring (RPM) Interventions on Safety, Adherence, Quality-of-Life and Cost-Related Outcomes. *NPJ Digit. Med.* **2024**, *7*, 192. [CrossRef]
44. Zhang, Y.; Liu, H.; Su, X.; Jiang, P.; Wei, D. Remote Mobile Health Monitoring System Based on Smart Phone and Browser/Server Structure. *J. Healthc. Eng.* **2015**, *6*, 717–738. [CrossRef]
45. Chinnaperumal, S.; Periyasamy, M.; AliAlhussan, A.A.; Kannan, S.; Khafaga, D.S.; Raju, S.K.; Eid, M.M.; El-kenawy, E.M. Secure and Intelligent 5G-Enabled Remote Patient Monitoring Using ANN and Choquet Integral Fuzzy VIKOR. *Sci. Rep.* **2024**, *28*, 1947–1965. [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

Takagi–Sugeno–Kang Fuzzy Inference Tracking Controller for UAV Bicopter

José R. Rivera-Ruiz ¹, José R. García-Martínez ^{1,2,*}, Trinidad Martínez-Sánchez ^{1,2}, Edson E. Cruz-Miguel ^{1,2}, Luis D. Ramírez-González ^{1,2}, Omar A. Barra-Vázquez ^{1,2} and Ákos Odry ^{3,4}

¹ Laboratorio de Control y Robótica, Facultad de Ingeniería en Electrónica y Comunicaciones, Universidad Veracruzana, Poza Rica 93390, Mexico; zs20008704@estudiantes.uv.mx (J.R.R.-R.); trmartinez@uv.mx (T.M.-S.); edsoncruz@uv.mx (E.E.C.-M.); dramirez@uv.mx (L.D.R.-G.); omabarra@uv.mx (O.A.B.-V.)

² Análisis de Sistemas y Tecnologías Emergentes, Facultad de Ingeniería en Electrónica y Comunicaciones, Universidad Veracruzana, Poza Rica 93390, Mexico

³ Department of Mechatronics and Automation, Faculty of Engineering, University of Szeged, 6725 Szeged, Hungary; odrya@uniduna.hu

⁴ Institute of Informatics, University of Dunaújváros, 2400 Dunaújváros, Hungary

* Correspondence: romangarcia@uv.mx

Abstract: The UAV bicopter is a double-propeller system whose main objective is to stabilize a rod at a given angle by precisely controlling the rotation speed of each propeller. This mechanism generates asymmetric thrust forces that induce a torque on the bar, thus allowing its pitch angle to be modified. Since its dynamics involve complex interactions between the thrust generated by the rotors, aerodynamic effects, and the pendulum behavior of the system, the bicopter is classified as a highly nonlinear system sensitive to external disturbances. To address this complexity, the implementation of a fuzzy Takagi–Sugeno–Kang (TSK) controller is proposed. This controller decomposes the nonlinear dynamics into multiple local linear models associated with a specific operating condition, such as different pitch angles and rotor speeds. The control strategy provides accurate trajectory tracking and effectively handles disturbances and varying conditions, making this approach a practical solution for both dynamic and uncertain environments. This strategy ensures precise trajectory tracking and demonstrates robust performance compared to other control methods, such as PID and LQR, which often struggle with disturbances and system nonlinearities. The TSK controller has proven its effectiveness in experimental trajectory tracking tests, achieving root mean square errors (RMSEs) of 0.2049, 0.3269, 0.3899, 0.3335, and 0.2494, which evaluate the average error in degrees of the system concerning the target position, for tracking trajectories of -10 to 10 , -12 to 12 , -15 to 15 , -17 to 17 , and -20 to 20 degrees, respectively.

Keywords: fuzzy control; UAV; bicopter; tracking controller; Takagi–Sugeno–Kang controller

1. Introduction

The Unmanned Aerial Vehicle (UAV) bicopter is a double-propeller system whose main purpose is to stabilize a rigid bar at a certain angle, precisely controlling the rotation speed of each of its propellers [1]. Its operating principle is based on the generation of asymmetric thrust forces: varying the relative speed between the two rotors produces an imbalance in thrust that induces a torque on the bar, allowing its angular inclination to be actively modified and controlled. This action is key to maintaining the system's stability under various operating conditions. However, despite the apparent simplicity of its mechanical structure, the UAV bicopter's dynamic behavior is considerably complex. This

is because the interaction between thrust forces, aerodynamic effects, and the pendulum-type oscillatory behavior of the bar generates strongly coupled and nonlinear dynamics [2]. In this context, even minor external disturbances or changes in the payload can trigger significant responses, making the system highly sensitive and unstable if an appropriate control strategy is not applied. Furthermore, the dependence of thrust on rotor speed is nonlinear, and the effects of airflow generated by one propeller can influence the behavior of the other, further increasing the complexity of the system [3].

TSK fuzzy systems represent an advanced extension of classical fuzzy logic, designed to model complex nonlinear systems by combining local rules based on linear models [4]. Unlike Mamdani-type fuzzy systems, in TSK models, the conclusions of the fuzzy rules are not linguistic values but mathematical functions that depend on the input variables [5]. Each rule is activated based on the degree of membership of the inputs in certain fuzzy sets, and the overall output of the system is obtained as a weighted combination of the individual outputs of all activated rules [6]. This structure allows for a more compact and efficient representation of the dynamics of highly nonlinear systems [7,8]. In the field of dynamical systems, TSK models are widely used to approximate the dynamics of physical, mechanical, or electrical processes that exhibit complex, nonlinear, and context-dependent behavior, such as UAVs, robotic manipulators, industrial process control systems, and biomechanical systems [9–13]. Their main advantage lies in the ability to decompose global dynamics into multiple local linear models that can be efficiently designed and analyzed with classical linear control tools [14]. At the same time, fuzzy logic ensures smooth transitions between them. This allows for the design of adaptive and robust controllers capable of maintaining system performance even under uncertainty, external disturbances, or changes in operating conditions. The main contributions of this work are summarized as follows:

- We design and implement a TSK fuzzy control methodology for a UAV bicopter system based on interpolating local linear models using triangular and trapezoidal membership functions. This methodology effectively approximates the nonlinear behavior inherent to the system dynamics.
- Proposal of a flexible control curve generation strategy, where the designer can manually adjust the fuzzy control curves to achieve accurate reference trajectory tracking under different operating conditions, providing adaptability at the design stage rather than in real-time operation.
- Experimental validation demonstrating enhanced structural robustness compared to existing state-of-the-art methodologies, confirmed through consistent low RMSE and IAE values and rapid settling times, even under external disturbances and uncertainties.

Finally, this paper is structured into several key sections. Section 2 presents related work in control applied to the bicopter, the dynamic model of the system in UAV bicopter system dynamics, the theoretical foundations of the method in Takagi–Sugeno Fuzzy Inference, and the proposed controller structure for the TSK Fuzzy Controller, as well as the description of tracking tasks in Parabolic Trajectory. Section 3 describes the design and implementation of the TSK fuzzy controller, addressing the general structure, linguistic variables, linear output functions, and the rule base. Section 4 presents the system performance in five tracking scenarios with different start and end points. Section 5 analyzes the results compared to previous methodologies. Finally, Section 6 summarizes the findings of the study, followed by the bibliography.

2. Background

2.1. Related Works

The mechanical system of the bicopter has been increasingly analyzed in recent years due to its interesting nonlinear dynamics and the complexities of controlling the system. Using controllers based on fuzzy logic has been widely applied in recent research because of their ability to improve the performance of highly complex and nonlinear systems. Furthermore, the study of trajectory tracking has recently been a highly researched methodology because it is one of the complex challenges in UAVs, which requires advanced control techniques capable of handling uncertainties and nonlinearities in the system. Previous studies have shown that PID (Proportional–Integral–Derivative) controllers are one of the methods used. Derived from classical control theory, it is one of the most widely applied methodologies in control systems due to its effectiveness in stabilizing linear systems [15]. Ref. [16] presents a case study where the authors implement these classical controllers to stabilize a bicopter, a variant of a UAV. The results showed the PID controller as an effective methodology for keeping the system balanced, demonstrating through experimental tests an appropriate response when the system is subjected to load tests on any of its axes. Recent research has applied optimization algorithms to improve PID controllers' performance. The authors of [17] proposed a multi-criteria genetic algorithm (GA) to optimize PI (Proportional–Integral) controller gains for a balancing bicopter. This work underscores the importance of multi-objective optimization in UAV control. In recent years, intelligent controllers have gained prominence to ensure system stability [18]. Cukdar et al. [19] proposed an Adaptive Fuzzy 2-DOF PID controller for a BLDCM-driven two-rotor UAV, demonstrating superior performance under variable operating conditions compared to classical and 2-DOF PID controllers. The study utilized Particle Swarm Optimization (PSO) to tune PID parameters. Recent advances in bicopter control have explored integrating intelligent control methods with robust nonlinear strategies to improve system performance under uncertainties and disturbances. The authors of [20] present a Direct Fuzzy Logic Controller (DFLC) combined with a Fast Terminal Sliding Mode Control (FTSMC) to address the attitude stabilization problem of a bicopter system. The DFLC provides a rule-based nonlinear control law without requiring an explicit dynamic model, while the FTSMC enhances robustness and ensures finite-time convergence. This study is particularly relevant as it exemplifies the synergy between model-free fuzzy control and model-based robust control. In [21], a Mamdani-type fuzzy controller was proposed to stabilize the roll motion of a UAV Bicopter. Considering the nonlinear, unstable, and underactuated nature of the system, the FLC was implemented on a Teensy 3.6 along with an MPU6050 IMU and servomotors for rotor tilting. Different rule base sizes (3×3 , 5×5 , 7×7 , and 9×9) were evaluated, finding that a larger number of rules (9×9) provided the best performance. Bench tests demonstrated quick recovery from disturbances and reference changes, thus validating the effectiveness of the fuzzy approach in attitude control. The study in [22] presents the control of a three-degree-of-freedom tilt-rotor bicopter test bench. The system combines a tilt-rotor bicopter with an articulated arm to enable safe and repeatable testing of control strategies for vehicles with thrust-vectoring capabilities. The authors address the challenge of decoupling attitude and position control in multi-rotor systems by employing a dual-mode control strategy, which integrates a Linear Quadratic Regulator (LQR) for stability and a Model Predictive Controller (MPC) for reference tracking and input constraint enforcement. Experimental results validate the controller's robustness against model uncertainties and disturbances, demonstrating its effectiveness in position tracking while minimizing attitude variations. Additionally, the case study presented in [23] employs a methodology that integrates fuzzy logic, PID, and Active Disturbance Rejection Control (ADRC) applied to a UAV bicopter. Simulations have demonstrated

the effectiveness of this strategy, showing improvements in response speed, disturbance rejection, and overall system robustness. This approach provides a promising direction for tackling the complexities of controlling bicopter UAVs in practical applications.

On the other hand, control proposals from modern control theory, such as LQR controllers, have been analyzed. Prakosa et al. [24] experimentally evaluated the Linear Quadratic Regulator (LQR) method to optimize PID gains for bicopter take-off position control. Their study highlighted the critical role of cost matrix weighting in achieving asymptotic stability for roll angles. In addition, LQR controllers have also been implemented alongside optimization algorithms. This case study [25] uses a Genetic Algorithm (GA) to find the optimal Q and R parameters for the Linear Quadratic control method. This approach is applied to a bicopter to stabilize the system through roll angle control.

Other control methodologies have been applied in the bicopter system. The work by Arrofiq et al. [26] presents a rigorous method for modeling a balancing bicopter's roll dynamics using multi-level periodic perturbation signals. Their approach combines experimental system identification with cross-distribution validation, achieving close alignment between simulation and real-world performance.

In recent years, various approaches such as traditional control, controllers optimized through metaheuristic algorithms, and classical fuzzy controllers have been applied to nonlinear systems with high uncertainty, such as UAV bicopters. However, these methods present certain limitations. PID controllers exhibit reduced performance when facing nonlinearities and external disturbances. Optimization algorithm-based methods are effective for parameter tuning but require high computational cost and do not always guarantee real-time adaptability. Traditional DFLCs, while robust, tend to rely on many rules to achieve acceptable performance, making their implementation in embedded systems challenging due to the high computational demand. In contrast, implementing a TSK fuzzy controller represents a promising alternative to traditional methodologies, due to its ability to handle system nonlinearities while representing outputs through linear functions, significantly reducing computational cost compared to classical fuzzy control methods. This feature is particularly relevant for embedded systems with limited resources.

Table 1 presents several control methods from the most relevant research related to the bicopter system. Additionally, this case study is a continuation of the work presented in [27], where an adaptive fuzzy PID controller, self-tuned using the Hunting Search optimization algorithm, was previously applied to a classical single-rotor aero pendulum system. Building on that work, the present study introduces two significant advancements: integrating a second rotor to enhance control authority and the design of a TSK fuzzy controller that significantly reduces computational complexity. While both approaches are based on fuzzy logic, the TSK fuzzy controller represents a clear improvement over the adaptive fuzzy PID framework. Unlike the prior method, which was limited by the linearity of the PID structure, the TSK controller employs rule-based local models, enabling a more accurate approximation of nonlinear dynamics. This results in smoother control actions, greater robustness to disturbances, faster convergence, and superior trajectory tracking under dynamic and uncertain conditions, demonstrating substantial conceptual and practical advancement.

Table 1. Several methodologies are applied to the UAV bicopter system.

Methodology	Controller	Remarks
[16]	PID	A PID controller is implemented for the balancing control of a bicopter vehicle. For the experimental results, different weight loads are applied at various prototype points to validate the controller.
[17]	Optimized PI	They optimize the PI controller using a Genetic Algorithm to find the best gains for the control method and implement it in the proposed system. Two approaches are employed to optimize the controller gains: one incorporates the Integral of Absolute Error (IAE), while the other does not.
[19]	Adaptive Fuzzy 2-DOF PID	An Adaptive Fuzzy (AF) 2-DOF PID controller is employed to balance a two-rotor UAV. The fuzzy controller is compared with a classical PID controller to demonstrate and validate the method proposed by the authors.
[20]	DFLC combined with FTSMC	They use a direct Takagi–Sugeno fuzzy logic controller adapted to a Fast Terminal Sliding Mode Control (FTSMC) to reduce the nonlinearities and uncertainties present in the system.

2.2. System Dynamics of the UAV Bicopter

The UAV bicopter system consists of a rigid bar driven by two motorized propellers at its ends. Each propeller generates vertical thrust to achieve stabilization, and an inertial measurement unit (IMU) is employed to measure the angular position of the bar relative to the vertical axis. Both motors are positioned at opposite bar ends and rotate in opposite directions to generate lift and counteract torque effects. The mechanical structure of the proposed system, illustrated in Figure 1, comprises a suspended bar connected to two brushless DC (BLDC) motors equipped with propellers. These motors are driven by 30-A electronic speed controllers (ESCs), which regulate the rotational speed of the BLDC motors. The system uses an MPU6050 inertial measurement unit to provide real-time angular position measurements, enabling precise control and stabilization of the bicopter platform.

The UAV bicopter’s movement analysis focuses solely on the rotational subsystem. The free-body diagram is presented in Figure 2, which illustrates the forces and moments affecting the system proposed. Table 2 shows the parameters used in the mathematical model of the UAV bicopter, including its physical properties, system constants, and key state variables for its dynamic analysis.

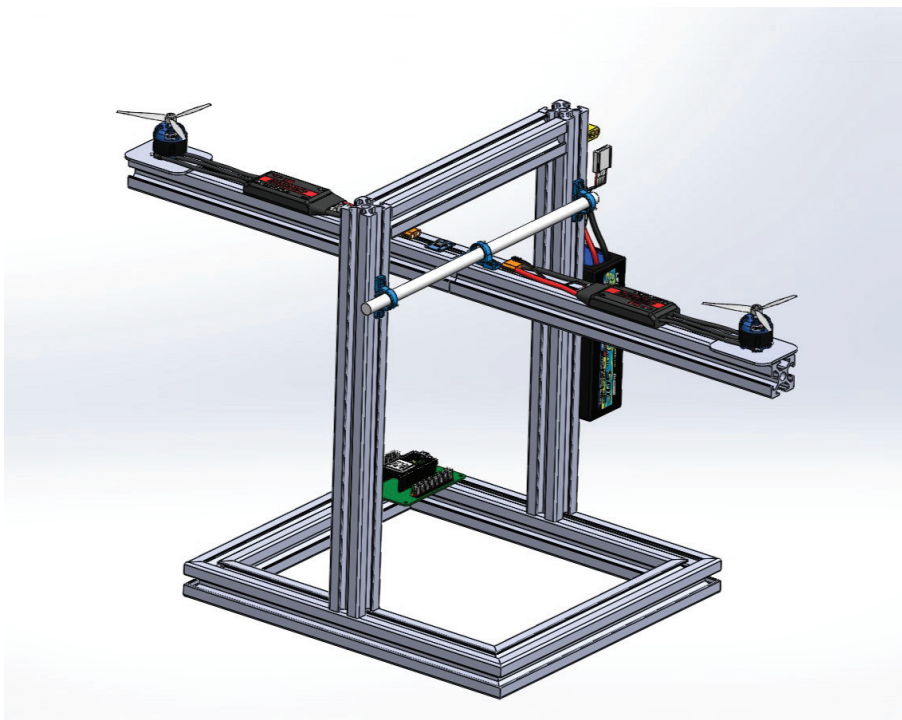


Figure 1. Three-dimensional design of the UAV bicopter system.

Table 2. Mechanical parameters of the UAV bicopter.

Parameter	Symbol	Value	Units
Propulsive force of rotor 1	P_1	-	N
Propulsive force of rotor 2	P_2	-	N
Fulcrum to rotor 1 distance	l_1	0.325	m
Fulcrum to rotor 2 distance	l_2	0.325	m
Bar length	L	0.65	m
Rotor 1 angular velocity	ω_1	-	rad/s
Rotor 2 angular velocity	ω_2	-	rad/s
UAV bicopter angle	θ	-	Degrees

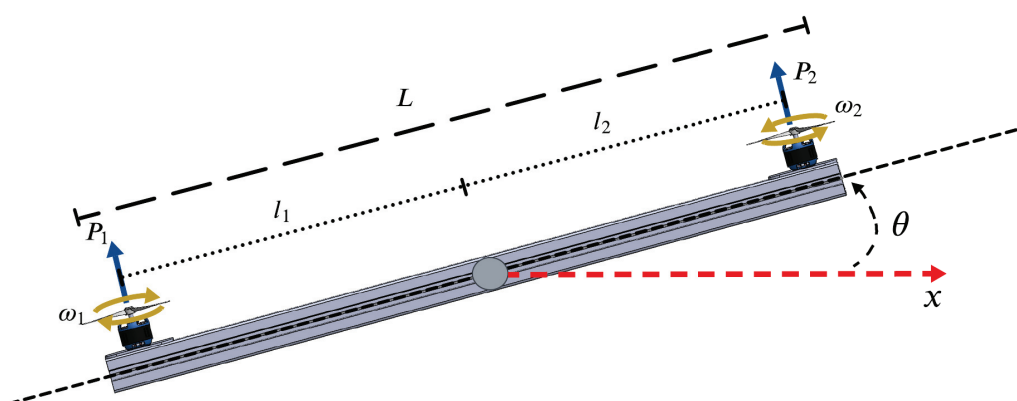


Figure 2. Free-body diagram of the UAV bicopter.

The system includes a pivot bar at its center, with two rotors at the ends generating propulsive forces, denoted as P_1 and P_2 . The system dynamics are derived using Newton–Euler’s rotational equation, which is centered at the pivot point. The governing dynamics of the system are presented in Equation (1).

$$J\ddot{\theta}(t) + b\dot{\theta}(t) + k\theta(t) = \tau(t) \quad (1)$$

where

- J is the moment of inertia of the bar for the pivot.
- b is the viscous friction coefficient at the pivot.
- k is a restoring coefficient.
- τ is the net torque generated by the rotors.

The torque generated by the motors can be determined as in Equation (2):

$$\tau(t) = l_2 P_2 - l_1 P_1 \quad (2)$$

Substituting the total torque in Equation (1), we have

$$J\ddot{\theta}(t) + b\dot{\theta}(t) + k\theta(t) = P_2(t) \cdot l_2 - P_1(t) \cdot l_1 \quad (3)$$

Assuming that the thrust forces P_1 and P_2 generated by the engines are directly related to the square of their respective angular velocities, it follows that $P_i(t) = k_f \omega_i^2(t)$, where k_f represents the aerodynamic thrust constant, as given in Equation (4).

$$J\ddot{\theta}(t) + b\dot{\theta}(t) + k\theta(t) = k_f [\omega_2^2(t) \cdot l_2 - \omega_1^2(t) \cdot l_1] \quad (4)$$

where k_f represents the aerodynamic constant of thrust generation, which depends on the physical characteristics of the rotor and the fluid medium. A state space representation of the system is proposed to find a stability criterion for the system. For this case, the following states of the system are chosen: $x_1(t) = \theta(t)$ and $x_2(t) = \dot{\theta}(t)$. Also, $u(t)$ is defined as the control input that encapsulates the inherent nonlinearities of the system; it means $u(t) = k_f [\omega_2^2(t) \cdot l_2 - \omega_1^2(t) \cdot l_1]$. From a practical implementation perspective, these nonlinear behaviors are not explicitly modeled but are instead addressed through the fuzzy controller's inference mechanism, which dynamically adapts the control action based on the system's operating conditions. Thus, the system can be expressed as Equation (5).

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{J} & -\frac{b}{J} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} u(t) \quad (5)$$

The quadratic Lyapunov function expressed in Equation (6) is proposed, directly reflecting the energy stored in the modeled UAV bi-copter.

$$V(x) = \frac{1}{2} J x_2^2 + \frac{1}{2} k x_1^2 \quad (6)$$

It can be observed that $V(x)$ is positive definite because $V(x) > 0$ for all $x \neq 0$, given that both $J > 0$ and $K > 0$ are physical parameters and the square of a real number is non-negative. Additionally, it holds that $V(0) = 0$, which is a necessary condition for a Lyapunov function.

The time derivative of the Lyapunov function along the trajectories of the system provides crucial information about the stability behavior. Differentiating Equation (6) concerning time gives Equation (7).

$$\dot{V}(x) = \frac{\partial V}{\partial x_1} \dot{x}_1 + \frac{\partial V}{\partial x_2} \dot{x}_2 = k x_1 \dot{x}_1 + J x_2 \dot{x}_2. \quad (7)$$

By substituting the state equations from Equation (5) into Equation (7), the resulting expression given by Equation (8) is obtained.

$$\dot{V}(x) = kx_1x_2 + Jx_2\left(-\frac{k}{J}x_1 - \frac{b}{J}x_2 + \frac{1}{J}u(t)\right), \quad (8)$$

which expands to

$$\dot{V}(x) = kx_1x_2 - kx_1x_2 - bx_2^2 + x_2u(t).$$

Observing that the terms kx_1x_2 and $-kx_1x_2$ cancel out, the derivative simplifies to Equation (9).

$$\dot{V}(x) = -bx_2^2 + x_2u(t). \quad (9)$$

This expression is fundamental for establishing the system's stable conditions. The derivative $\dot{V}(x)$ consists of two terms:

- The term $-bx_2^2$ is strictly non-positive for all x_2 , since $b > 0$ and $x_2^2 \geq 0$.
- The term $x_2u(t)$ depends on the control input $u(t)$ and the system's angular velocity x_2 .

If the control input $u(t)$ is designed such that it does not dominate the natural dissipative effect introduced by the damping term $-bx_2^2$, the system's total energy, represented by $V(x)$, will decrease over time. This implies that the system will gradually dissipate energy, leading to convergence toward the equilibrium point.

2.3. Takagi–Sugeno Fuzzy Inference

In 1985, a new methodology for a fuzzy inference engine was proposed by Takagi and Sugeno [28]. It was introduced due to the limitations of the Mamdani inference engine, which presented high computational loads, causing limitations in fast dynamic systems. Takagi and Sugeno replaced the fuzzy outputs of the rules with mathematical functions [29,30], making the system more efficient. This allows implementations in nonlinear systems that are too complex to be solved by the Mamdani method [31].

All fuzzy sets are associated with linear membership functions, characterized by two parameters representing the maximum degree of 1 and the minimum degree of 0. The truth value of a proposition “ x is A and y is B ” is defined as follows: $|x \text{ is } A \text{ and } y \text{ is } B| = A(x) \wedge B(x)$.

Equation (10) shows the expression of fuzzy implication.

$$R : \text{If } f(x_1 \text{ is } A_1, \dots, x_i \text{ is } A_i), \text{ then } z_i = (a_ix + b_i, i = (1, 2, 3, \dots, n)). \quad (10)$$

where:

- z_i : Variable of the consequence whose value is inferred.
- $x_1 - x_i$: Variables of the premise that also appear in the consequence part.
- $A_1 - A_i$: Fuzzy sets with linear membership functions representing a fuzzy subspace where the implication R can be applied for reasoning.
- f : Logical function that connects the propositions in the premise.
- $a_ix + b_i$: Function that implies the value of z_i when $x_1 - x_i$ satisfy the premise.

The final output z_0 is inferred from n implications and is given as the average of all z_i with the weights as can be seen in Equation (11), assuming that $a_i = A_i(x)$.

$$z_0 = \frac{\sum_{i=1}^n a_i z_i}{\sum_{i=1}^n a_i} \quad (11)$$

2.4. Takagi–Sugeno–Kang Fuzzy Controller

In 1998, ref. [32] presented a Takagi–Sugeno–Kang (TSK) fuzzy controller based on a TS fuzzy model consisting of TS fuzzy rules whose consequents are state-space equations with

a constant term. In the TS fuzzy model, each rule corresponds to a specific operating region or condition of the system. These rules define linear input–output relationships within their regions, enabling a simplified representation of the local dynamics [33]. The overall system behavior is then obtained by combining these local linear models using fuzzy logic principles. This combination process ensures smooth transitions between operating regions, producing a cohesive and accurate representation of the system’s global behavior.

The structure of the TSK fuzzy controller consists of four stages: fuzzification, rule formulation, TS fuzzy inference, and defuzzification.

1. **Fuzzification:** This stage assigns each input a degree of membership to the defined fuzzy sets, enabling a qualitative representation of the system’s quantitative information. Each input variable is transformed into linguistic variables through fuzzy membership functions.
2. **Rules formulation:** The designer specifies the decision-making rules based on the relationship between the inputs and outputs.
3. **TS fuzzy inference:** In this stage, the TS fuzzy inference links the rules established in the previous stage with the outputs. This fuzzy inference relates the rules to the linear output functions proposed by the designer.
4. **Defuzzification:** Finally, in the defuzzification stage, a crisp output is obtained from the linear output functions using a weighted average, as seen in Equation (11).

2.5. Parabolic Trajectory

In control systems, a trajectory refers to the path a system follows through a sequence of movements within a specified time frame. One of the most widely used methods for generating such trajectories is the parabolic velocity profile [34], which provides smooth and predictable motion. This model accelerates the actuator at a constant rate until a midpoint is reached, followed by a symmetrical deceleration phase until the velocity returns to zero.

Equation (12) defines the resulting position over time, which describes the displacement at each instant. Figures 3 and 4 illustrate this parabolic trajectory’s corresponding position and velocity profiles.

$$\theta(t) = \theta_0 + \frac{2\dot{\theta}_f}{t_f}(t - t_0)^2 - \frac{4\dot{\theta}_f}{3t_f^2}(t - t_0)^3, \quad 0 < t \leq t_f \quad (12)$$

Equation (13) relates the velocity and the execution time of the motion.

$$t_f = \frac{3h}{2\dot{\theta}_f} \quad (13)$$

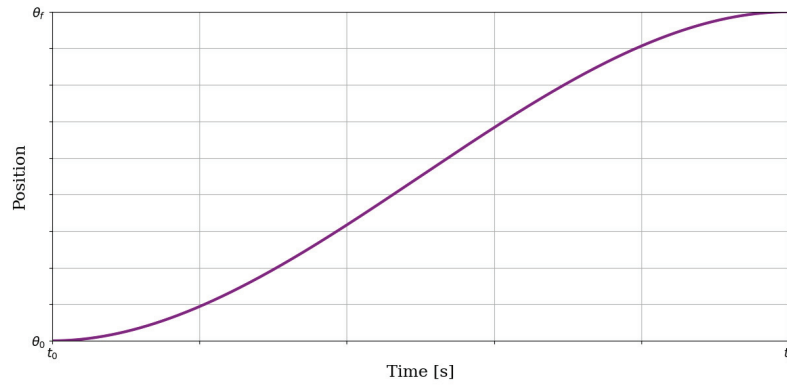


Figure 3. Position [Degrees] in a parabolic velocity profile.

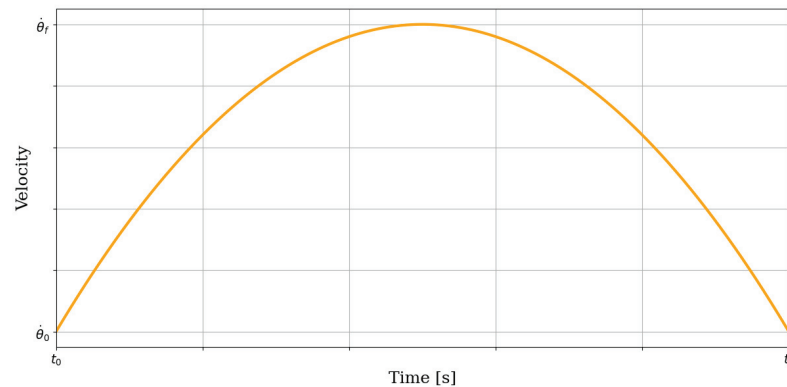


Figure 4. Velocity [Degrees/s] in a parabolic velocity profile.

3. Materials and Methods

The Takagi–Sugeno–Kang controller is the fuzzy controller presented in this research. This methodology is a rule-based control approach that combines fuzzy logic with linear models to handle complex, nonlinear systems. Figure 5 illustrates the methodology employed in the design of the TSK fuzzy controller. The first step is to generate the parabolic trajectory presented in Section 2.5, where the designer specifies the target final position and the execution time of the motion. The next step is the fuzzification process, in which linguistic variables are proposed based on the angular error exhibited by the system through membership functions. The third step is establishing the IF–THEN rules that relate the inputs to the outputs. Subsequently, the rules are evaluated through TS fuzzy inference. The next step is to perform defuzzification using Equation (11) in Section 2.3, where the output of the controller is computed as a weighted average of the rule outputs, with the weights determined by the firing strengths of the rules. The final stage corresponds to sending the control signal to both rotors of the UAV bicopter, denoted by $U(t)$, to correct the error presented by the system.

The error denoted by $e(t)$ is defined as the difference between the reference or target position $r(t)$ and the current position measured by the angular sensor denoted by θ , as shown in Equation (14).

$$e(t) = r(t) - \theta(t) \quad (14)$$

The fuzzification stage is based on the angular error using fifteen linguistic values. The linguistic values that describe it are shown with the following abbreviations: VLN (Very Large Negative), LN (Large Negative), MLN (Moderately Large Negative), MN (Medium Negative), MSN (Moderately Small Negative), VSN (Very Small Negative), Z (Zero), VSP (Very Small Positive), SP (Small Positive), MSP (Moderately Small Positive),

MP (Medium Positive), MLP (Moderately Large Positive), LP (Large Positive), and VLP (Very Large Positive).

The linguistic variable (error) is described by two trapezoidal and thirteen triangular linguistic values. Equations (15) and (16) present the triangular and trapezoidal membership functions, respectively.

$$\mu_{\text{triangular}}(x) = \begin{cases} \frac{x_i - \alpha_i}{m_i - \alpha_i} & \text{if } x \in [\alpha_i, m_i] \\ \frac{x_i - m_i}{m_i - \beta_i} & \text{if } x \in (m, \beta_i] \\ 0 & \text{other values.} \end{cases} \quad (15)$$

$$\mu_{\text{trapezoidal}}(x) = \begin{cases} \frac{x_i - \alpha_i}{\beta_i - \alpha_i} & \text{if } x \in [\alpha_i, \beta_i) \\ 1 & \text{if } x \in [\beta_i, \gamma_i] \\ \frac{\gamma_i - x_i}{\epsilon_i - \gamma_i} & \text{if } x \in (\gamma_i, \epsilon_i] \\ 0 & \text{other values} \end{cases} \quad (16)$$

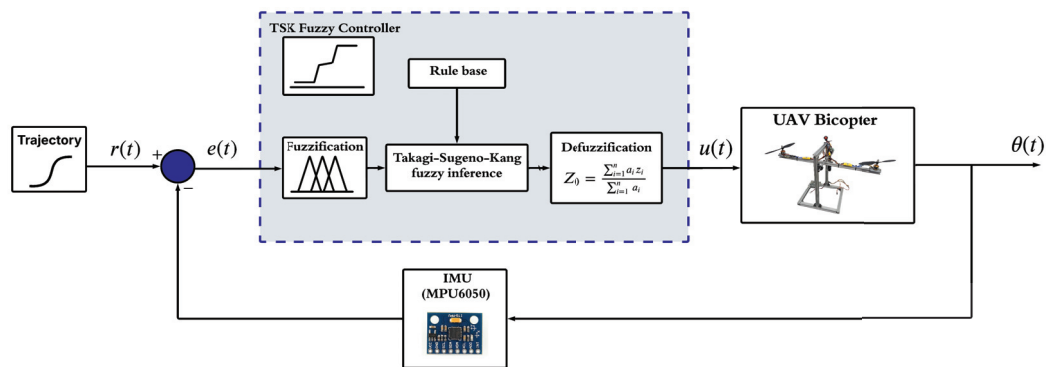


Figure 5. Takagi–Sugeno–Kang fuzzy tracking controller design.

For the triangular membership function, α_i , μ_i , and β_i represent the start, the central point, and the end, respectively. For the trapezoidal membership function, α_i , β_i , γ_i , and ϵ_i represent the start, the left central point, the right central point, and the end, respectively.

The distribution of the linguistic values is symmetrically arranged to ensure a similar system response when the error takes on either positive or negative values. Trapezoidal membership functions are proposed to provide a constant response when the error becomes significantly large. On the other hand, triangular membership functions are employed at the center of the distribution because the system requires a fast and precise response to correct the error due to nonlinearities.

Figure 6 illustrates the symmetrical distribution of the linguistic values used to represent the error.

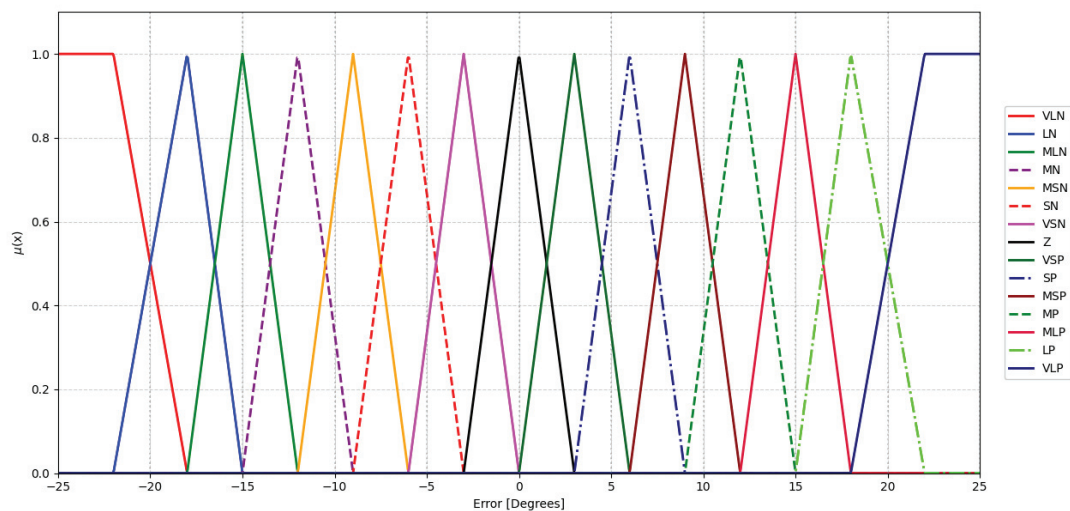


Figure 6. Error linguistic variables proposed.

Table 3 presents the linear output functions associated with the proposed TSK fuzzy tracking controller. The table is divided into two sections representing the output control functions U_1 and U_2 , which are applied to the two rotors (BLDC1 and BLDC2 motors) of the UAV bi-copter. For each linguistic rule z_i , a specific linear function of the form $a_i \cdot e(t) + b_i$ is designed to adjust the controller response. The outputs for U_1 start with positive coefficients a_i and progressively symmetrically shift toward negative values. In contrast, U_2 starts with negative coefficients and gradually shifts to positive values. This symmetric distribution ensures a balanced and coordinated actuation of both rotors. However, this symmetry only holds if the motors have the same characteristics. In practical applications, where slight differences between motors may exist, the symmetry can be complemented, allowing the use of a reduced set of 15 IF–THEN rules while maintaining effective control.

Table 3. Linear output functions of the TSK fuzzy tracking controller proposal.

U_1		U_2	
Variable	Linear Function	Variable	Linear Function
$z1_{BLDC1}$	$a_1 \cdot e(t) + b_1$	$z1_{BLDC2}$	$-a_{30} \cdot e(t) + b_{30}$
$z2_{BLDC1}$	$a_2 \cdot e(t) + b_2$	$z2_{BLDC2}$	$-a_{29} \cdot e(t) + b_{29}$
$z3_{BLDC1}$	$a_3 \cdot e(t) + b_3$	$z3_{BLDC2}$	$-a_{28} \cdot e(t) + b_{28}$
$z4_{BLDC1}$	$a_4 \cdot e(t) + b_4$	$z4_{BLDC2}$	$-a_{27} \cdot e(t) + b_{27}$
$z5_{BLDC1}$	$a_5 \cdot e(t) + b_5$	$z5_{BLDC2}$	$-a_{26} \cdot e(t) + b_{26}$
$z6_{BLDC1}$	$a_6 \cdot e(t) + b_6$	$z6_{BLDC2}$	$-a_{25} \cdot e(t) + b_{25}$
$z7_{BLDC1}$	$a_7 \cdot e(t) + b_7$	$z7_{BLDC2}$	$-a_{24} \cdot e(t) + b_{24}$
$z8_{BLDC1}$	$a_8 \cdot e(t) + b_8$	$z8_{BLDC2}$	$a_{23} \cdot e(t) + b_{23}$
$z9_{BLDC1}$	$-a_9 \cdot e(t) + b_9$	$z9_{BLDC2}$	$a_{22} \cdot e(t) + b_{22}$
$z10_{BLDC1}$	$-a_{10} \cdot e(t) + b_{10}$	$z10_{BLDC2}$	$a_{21} \cdot e(t) + b_{21}$
$z11_{BLDC1}$	$-a_{11} \cdot e(t) + b_{11}$	$z11_{BLDC2}$	$a_{20} \cdot e(t) + b_{20}$
$z12_{BLDC1}$	$-a_{12} \cdot e(t) + b_{12}$	$z12_{BLDC2}$	$a_{19} \cdot e(t) + b_{19}$
$z13_{BLDC1}$	$-a_{13} \cdot e(t) + b_{13}$	$z13_{BLDC2}$	$a_{18} \cdot e(t) + b_{18}$
$z14_{BLDC1}$	$-a_{14} \cdot e(t) + b_{14}$	$z14_{BLDC2}$	$a_{17} \cdot e(t) + b_{17}$
$z15_{BLDC1}$	$-a_{15} \cdot e(t) + b_{15}$	$z15_{BLDC2}$	$a_{16} \cdot e(t) + b_{16}$

Table 4 shows the fuzzy inference rules defined for the proposed TSK fuzzy tracking controller. Each rule follows the IF–THEN structure. Thirty fuzzy rules are proposed, with fifteen fuzzy rules for each output (U_1 , U_2).

Table 4. Fuzzy rules for the bicopter TSK fuzzy tracking controller.

U_1	U_2
Rules	Rules
If $e(t)$ is VLN, then U_1 is $z1_{BLDC1}$	If $e(t)$ is VLN, then U_2 is $z1_{BLDC2}$
If $e(t)$ is LN, then U_1 is $z2_{BLDC1}$	If $e(t)$ is LN, then U_2 is $z2_{BLDC2}$
If $e(t)$ is MLN, then U_1 is $z3_{BLDC1}$	If $e(t)$ is MLN, then U_2 is $z3_{BLDC2}$
If $e(t)$ is MN, then U_1 is $z4_{BLDC1}$	If $e(t)$ is MN, then U_2 is $z4_{BLDC2}$
If $e(t)$ is MSN, then U_1 is $z5_{BLDC1}$	If $e(t)$ is MSN, then U_2 is $z5_{BLDC2}$
If $e(t)$ is SN, then U_1 is $z6_{BLDC1}$	If $e(t)$ is SN, then U_2 is $z6_{BLDC2}$
If $e(t)$ is VSN, then U_1 is $z7_{BLDC1}$	If $e(t)$ is VSN, then U_2 is $z7_{BLDC2}$
If $e(t)$ is Z, then U_1 is $z8_{BLDC1}$	If $e(t)$ is Z, then U_2 is $z8_{BLDC2}$
If $e(t)$ is VSP, then U_1 is $z9_{BLDC1}$	If $e(t)$ is VSP, then U_2 is $z9_{BLDC2}$
If $e(t)$ is SP, then U_1 is $z10_{BLDC1}$	If $e(t)$ is SP, then U_2 is $z10_{BLDC2}$
If $e(t)$ is MSP, then U_1 is $z11_{BLDC1}$	If $e(t)$ is MSP, then U_2 is $z11_{BLDC2}$
If $e(t)$ is MP, then U_1 is $z12_{BLDC1}$	If $e(t)$ is MP, then U_2 is $z12_{BLDC2}$
If $e(t)$ is MLP, then U_1 is $z13_{BLDC1}$	If $e(t)$ is MLP, then U_2 is $z13_{BLDC2}$
If $e(t)$ is LP, then U_1 is $z14_{BLDC1}$	If $e(t)$ is LP, then U_2 is $z14_{BLDC2}$
If $e(t)$ is VLP, then U_1 is $z15_{BLDC1}$	If $e(t)$ is VLP, then U_2 is $z15_{BLDC2}$

The defuzzification stage is evaluated based on a weighted average, as shown in Equation (11). Therefore, Equations (17) and (18) determine the output values for both rotors.

$$U_1 = \frac{\sum_i^n a_i z_i}{\sum_i^n a_i} = \frac{VLN(z1_{BLDC1}) + LN(z2_{BLDC1}) + \dots + VLP(z15_{BLDC1})}{VLN + LN + \dots + VLP} \quad (17)$$

$$U_2 = \frac{\sum_i^n a_i z_i}{\sum_i^n a_i} = \frac{VLN(z1_{BLDC2}) + LN(z2_{BLDC2}) + \dots + VLP(z15_{BLDC2})}{VLN + LN + \dots + VLP} \quad (18)$$

To quantify the performance tracking controller, it is evaluated using the Root Mean Squared Error (RMSE) and the Integral Absolute Error (IAE). Equation (19) defines the calculation of the RMSE, which evaluates the average error between the system's behavior and the target position. On the other hand, the IAE computes the total accumulated error during the entire test duration. Its expression is defined in Equation (20).

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=0}^N (r(kT_s) - \theta(kT_s))^2} \quad (19)$$

$$IAE = \sum_{k=0}^N |r(kT_s) - \theta(kT_s)| \cdot T_s \quad (20)$$

where

- N is the total number of samples.
- $\theta(kT_s)$ is the position value at sample k .
- $r(kT_s)$ is the target position value at sample k .
- T_s is the sampling time.

4. Results

This section presents the final design of the TSK fuzzy tracking controller. The fuzzy tracking controller is implemented on an embedded system based on the ESP32 microcontroller. The MPU6050 sensor is responsible for acquiring the angular position data measured from the UAV bicopter. The embedded system is programmed in C++ using a sampling time of 0.02 s for the experimental phase. The execution time of the implemented TSK fuzzy controller has an average execution time of 387.8063 μ s per control

cycle. This execution time includes evaluating membership functions, the fuzzy inference process, and calculating the controller output. The selected sampling time is constrained by the physical limitations of the BLDC motors, which cannot adjust their rotational speed instantaneously due to the rotor's mechanical inertia. As a result, several milliseconds are required for the motor to respond effectively to changes in the PWM signal. Figure 7 illustrates the mechanical system of the UAV bicopter in experimentation. Algorithm 1 shows how the TSK fuzzy tracking controller works.

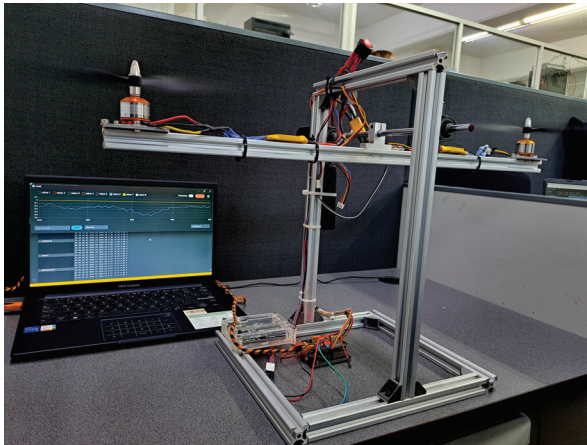


Figure 7. UAV bicopter stabilized at 0 degrees.

Algorithm 1: TSK fuzzy tracking controller algorithm.

```

Declare the inputs and outputs
Declare the constants and variables
Declare the linear function of the TSK controller
Set the final time and position of the trajectory  $t_f, r(t)$ 
Set the sampling time  $T_s$ 
Function LinguisticValues(error, anchoring points):
    Create the figures of the functions
    Calculate the membership degree
Function SetUp():
    Configure the BLDC frequency, PWM, and channel
    Configure the IMU sensor
    Create the interruption of the sampling time
while True do
    Calculate the angular position
    Filter the readings obtained from the IMU sensor
    Generate the parabolic trajectory in Figure 3
    Calculate error:  $e(t) = r(t) - \theta(t)$ 
    Evaluate the linguistic variable of error  $e(t)$  presented in Figure 6
    Evaluate the fuzzy rules
    Evaluate the defuzzification process using Equation (11)
    Send the output signal to the BLDC
end

```

Table 5 shows the distribution parameters of the triangular and trapezoidal error linguistic values. Figure 8 illustrates the visual representation of the membership functions used in the final controller proposal. The distribution of the functions closer to zero aims to achieve a fast system response within the error range of $[-1.5$ to $1.5]$ degrees, reducing the steady-state error.

Table 5. Parameters of the error linguistic variables.

Linguistic Value	α_i	m_i	β_i	γ_i	ϵ_i
VLN	−25	−	−25	−20	−15
LN	−18	−15	−12	−	−
MLN	−15	−12	−9	−	−
MN	−12	−9	−6	−	−
MSN	−8	−4	−1	−	−
SN	−1.5	−1	−0.5	−	−
VSN	−1	−0.5	0	−	−
Z	−0.5	0	0.5	−	−
VSP	0	0.5	1	−	−
SP	0.5	1	1.5	−	−
MSP	1	4	8	−	−
MP	6	9	12	−	−
MLP	9	12	15	−	−
LP	12	15	18	−	−
VLP	15	−	20	25	25

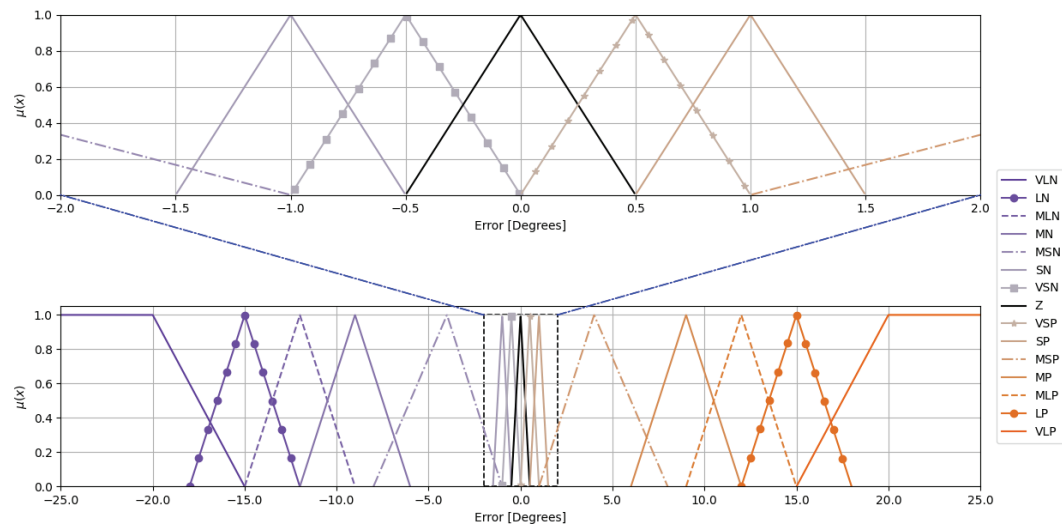
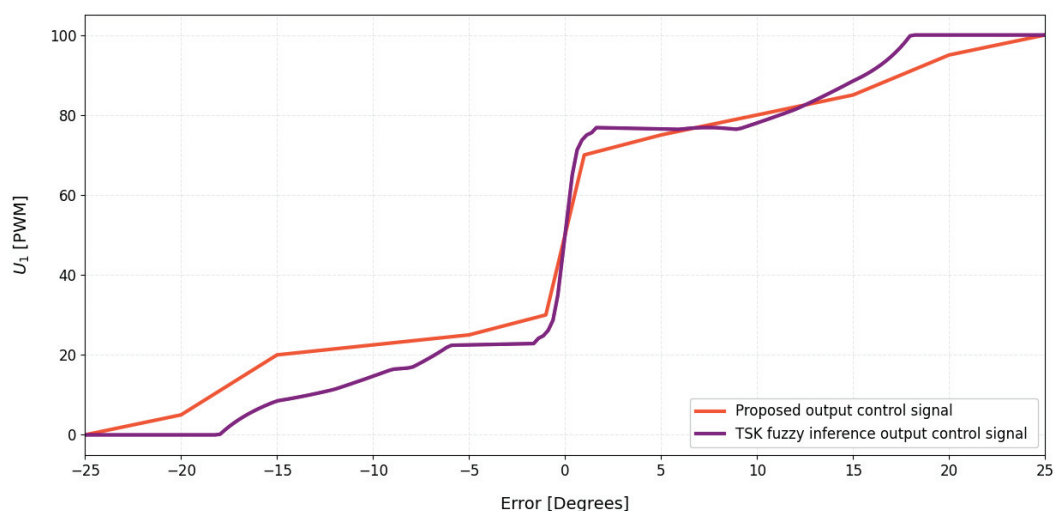
**Figure 8.** Error linguistic variables in the final controller proposal.

Table 6 presents the linear functions used in the final test. These functions are based on the response required by the system to overcome the inherent nonlinearities of the UAV bicopter. A quicker response is needed to stabilize the system when the error ranges from 0.5 to 1.5 degrees. Due to the nonlinearity, the UAV bicopter can enter steady-state error if a smoother response is observed within this range of values. The definition of the coefficients α_i and the constant terms b_i in the output linear functions of the TSK fuzzy controller follows a design process to ensure robust and precise system performance in response to variations in the reference error. The main objective in selecting these coefficients is to reduce the steady-state error, considering the system's inherent nonlinearities. Rapid and aggressive corrections are applied to achieve this in scenarios where immediate stabilization of the target position is required. In contrast, smoother corrections are used in situations far from the reference position to avoid oscillatory movements that could compromise the system's stability.

Table 6. Final linear output functions of the TSK fuzzy tracking controller.

U_1		U_2	
Variable	Linear Function	Variable	Linear Function
$z1_{BLDC1}$	$0 \cdot e(t) + 0$	$z1_{BLDC2}$	$0 \cdot e(t) + 100$
$z2_{BLDC1}$	$0.1 \cdot e(t) + 10$	$z2_{BLDC2}$	$-0.1 \cdot e(t) + 90$
$z3_{BLDC1}$	$0.3 \cdot e(t) + 15$	$z3_{BLDC2}$	$-0.5 \cdot e(t) + 85$
$z4_{BLDC1}$	$0.4 \cdot e(t) + 20$	$z4_{BLDC2}$	$-0.4 \cdot e(t) + 80$
$z5_{BLDC1}$	$0.1 \cdot e(t) + 23$	$z5_{BLDC2}$	$-0.3 \cdot e(t) + 77$
$z6_{BLDC1}$	$0.1 \cdot e(t) + 25$	$z6_{BLDC2}$	$-0.2 \cdot e(t) + 75$
$z7_{BLDC1}$	$0.05 \cdot e(t) + 30$	$z7_{BLDC2}$	$-0.2 \cdot e(t) + 70$
$z8_{BLDC1}$	$0 \cdot e(t) + 50$	$z8_{BLDC2}$	$0 \cdot e(t) + 50$
$z9_{BLDC1}$	$-0.05 \cdot e(t) + 70$	$z9_{BLDC2}$	$0.2 \cdot e(t) + 30$
$z10_{BLDC1}$	$-0.1 \cdot e(t) + 75$	$z10_{BLDC2}$	$0.2 \cdot e(t) + 25$
$z11_{BLDC1}$	$-0.1 \cdot e(t) + 77$	$z11_{BLDC2}$	$0.3 \cdot e(t) + 23$
$z12_{BLDC1}$	$-0.4 \cdot e(t) + 80$	$z12_{BLDC2}$	$0.4 \cdot e(t) + 20$
$z13_{BLDC1}$	$-0.3 \cdot e(t) + 85$	$z13_{BLDC2}$	$0.5 \cdot e(t) + 15$
$z14_{BLDC1}$	$-0.1 \cdot e(t) + 90$	$z14_{BLDC2}$	$0.1 \cdot e(t) + 10$
$z15_{BLDC1}$	$-0 \cdot e(t) + 100$	$z15_{BLDC2}$	$0 \cdot e(t) + 0$

Figures 9 and 10 show the control output surfaces obtained through the TSK fuzzy inference and the expected designed control surface. Although the expected control surface was designed using linear segments, the control signal processed through the TSK fuzzy inference system exhibits nonlinear behavior. This distortion arises from the nature of the membership functions employed, which influence the weighted combination of the local outputs. With two trapezoidal membership functions at the extremes, a slower response is obtained to prevent the system from aggressively correcting the error. In contrast, implementing the triangular functions for small angular errors generates a faster response to correct the system's output. The asymmetry in the output surface between the two rotors arises because the rotors do not have identical mechanical characteristics, causing them to break inertia in different ways. As a result, 15 fuzzy rules are proposed for each motor to account for their differing behaviors.

**Figure 9.** Proposed versus TSK fuzzy inference output control surface [Duty Cycle] of U_1 .

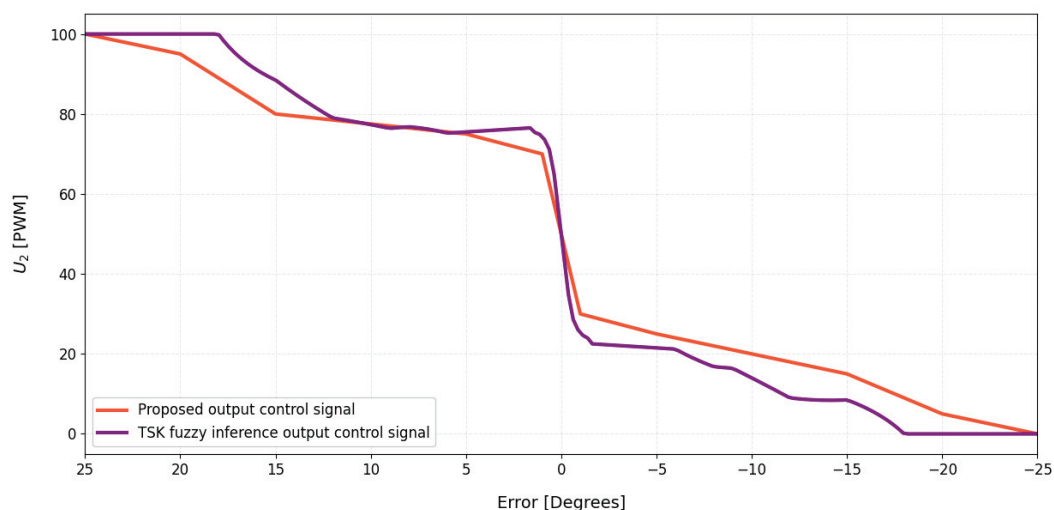


Figure 10. Proposed versus TSK fuzzy inference output control surface [Duty Cycle] of U_2 .

The first test was conducted to evaluate the performance of the TSK controller using a target position corresponding to a 10-degree step input, as illustrated in Figure 11. The system is stabilized within 1.5 s, achieving a minimal steady-state error of 0.25 degrees. Figure 12 presents the control signal applied in the step-test.

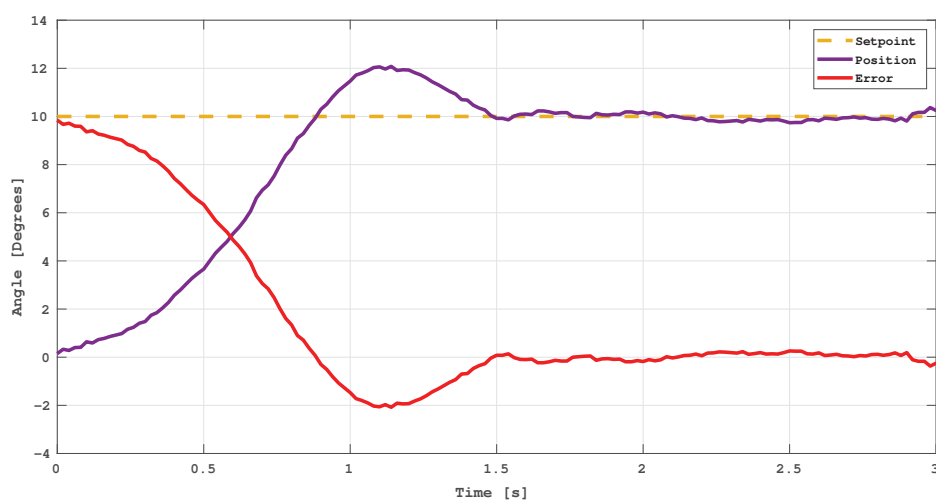


Figure 11. Ten-degree step response.

The experimental results are presented in five different trajectory tracking cases, with the main objective being to validate that the TSK controller successfully stabilizes the proposed system within the target time. For the five cases, a final time of 1.7 s is proposed. Figure 13 illustrates the angular position response of the system under the control of the TSK fuzzy controller. The reference trajectory is compared with the actual angular position obtained during the experiment. Figure 14 shows the angular error reported during several tests. Figure 15a,b show the control signals U_1 and U_2 , respectively, applied to the rotors during each trajectory tracking test. These signals reflect the controller's dynamic response to the desired reference inputs. The variation in the PWM levels for each rotor illustrates how the system adjusts the actuation to maintain stability and accurately follow the desired trajectory.

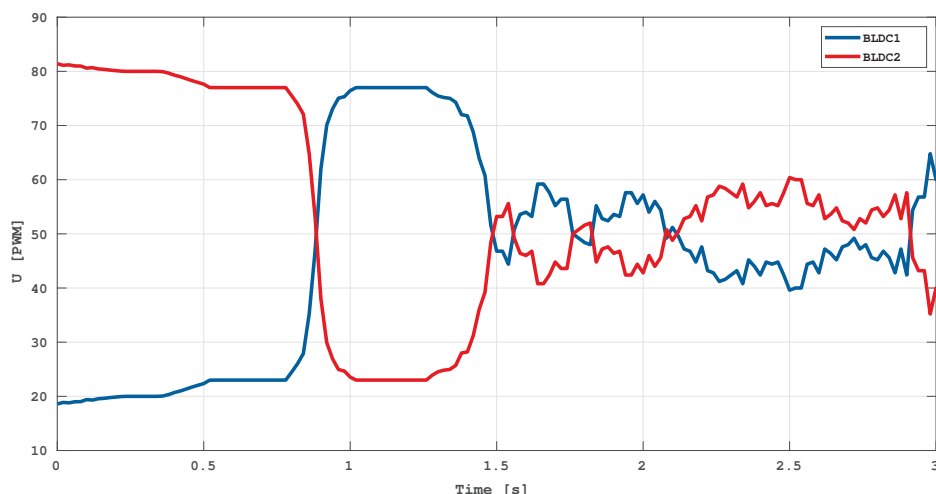


Figure 12. Control signals [Duty Cycle] applied to the system.

On the other hand, the first test starts at an initial position of -10 degrees and ends at the final position of 10 degrees. The maximum angular error observed during the experiment test is 0.6 degrees, while the steady-state error is 0.415. The system exhibits an RMSE value of 0.2049 and an IAE value of 0.9876. Additionally, the OS recorded during the test was 2.89915%. The second test from -12 to 12 reports an RMSE value of 0.3269 and an IAE of 2.7460, with a steady-state error of 0.35 degrees and an overshoot (OS) of 1.8333%, making it the trajectory case with the lowest values obtained. The third test starts from an initial position of -15 and ends at the final position of 15 degrees. The bicopter shows a steady-state error of 0.49 degrees, while the OS obtained during the test was 2.866%. This test presents an RMSE value of 0.3899 and IAE value of 2.5244. The four tests from -17 to 17 present an IAE value of 2.27615 and an RMSE value of 0.3335. The system shows a steady-state error of 0.39 degrees, while the overshoot obtained during the test was 2.294%. The fifth test begins at -20 degrees and reaches a final position of 20 degrees. The maximum angular error recorded was 0.86 degrees, while the steady-state error measured 0.43 throughout the experimental test. The system achieved an RMSE of 0.2494 and an IAE of 1.1799. Furthermore, the overshoot observed during the test was 1.1799%.

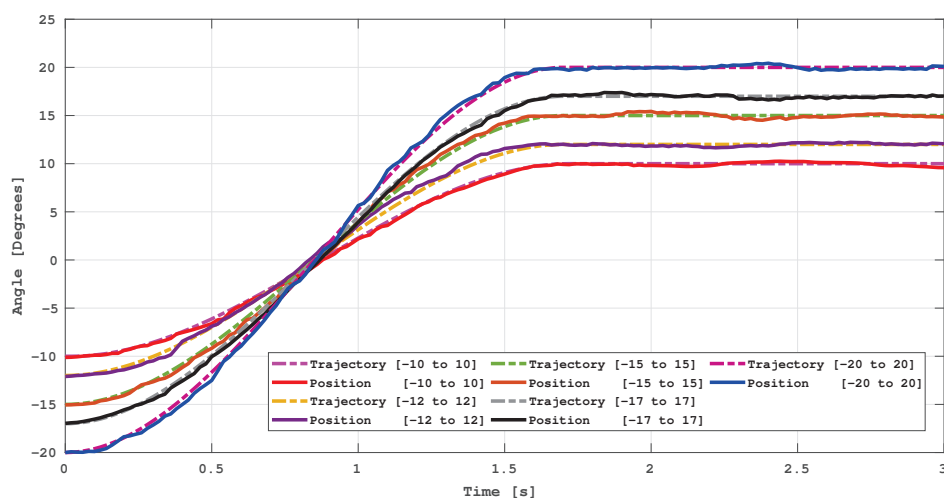


Figure 13. Bicopter response in trajectory tracking comparison.

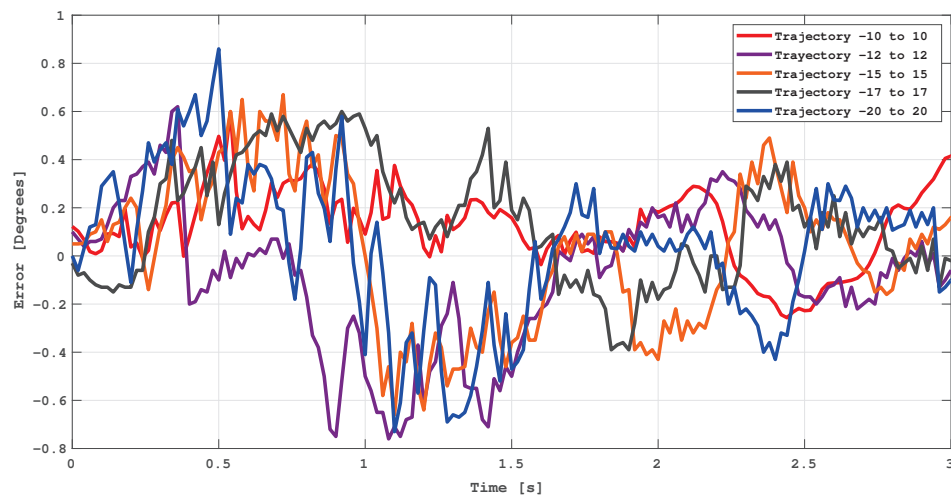
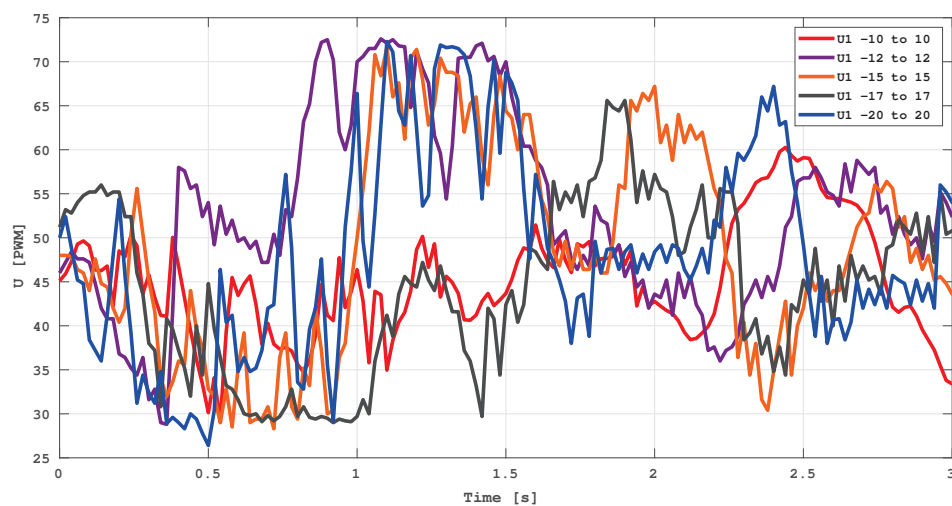
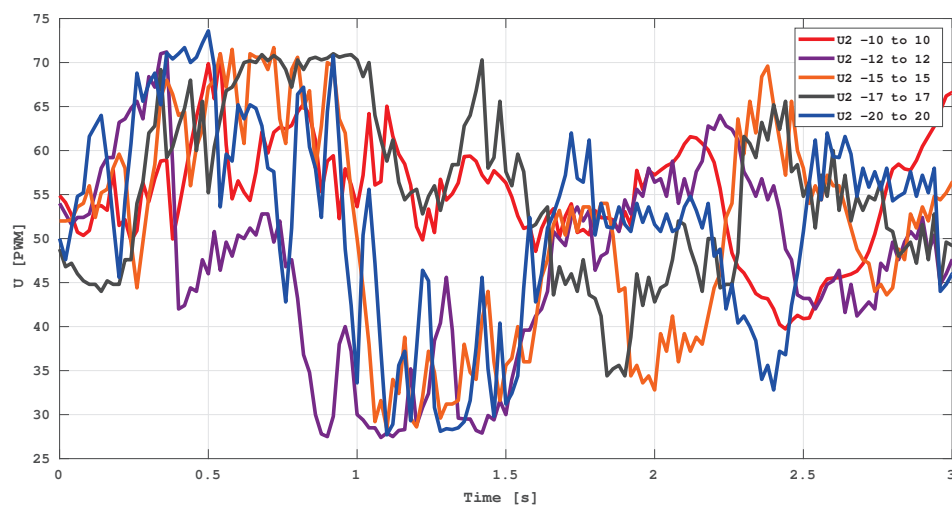


Figure 14. Error [Degrees] in several trajectories.



(a)



(b)

Figure 15. Control signal [Duty Cycle] of U_1 in several trajectories: (a) control signal of U_1 ; (b) control signal of U_2 .

Robustness Test

Figure 16 presents three scenarios of robustness tests applied to trajectory tracking within an angular range of -10 to 10 degrees. These tests aim to evaluate the capability of the proposed controller to maintain system performance under external disturbances. The methodology used to induce such disturbances involves periodically altering the target position. Every 8 s, the cycle duty of one rotor is increased to 90% while the other is decreased to 10% for a brief interval of 500 milliseconds. This abrupt and cyclical modification of the rotors' duty cycle is designed to temporarily destabilize the system, enabling the analysis of the controller's response to unexpected variations in the desired trajectory. The results of each test allow for the assessment of the controller's effectiveness in terms of recovery, stability, and accuracy under unanticipated external conditions. The system initially tracks the generated parabolic trajectory. Once the steady-state error is reached, the computationally generated perturbations are introduced. The response demonstrates the effectiveness of the controller under perturbations, as it returns to the target position after the disturbance. Figure 17 shows the error observed during the tests. On the other hand, Figure 18a,b present the control signals U_1 and U_2 , respectively, applied to rotors in three tests.

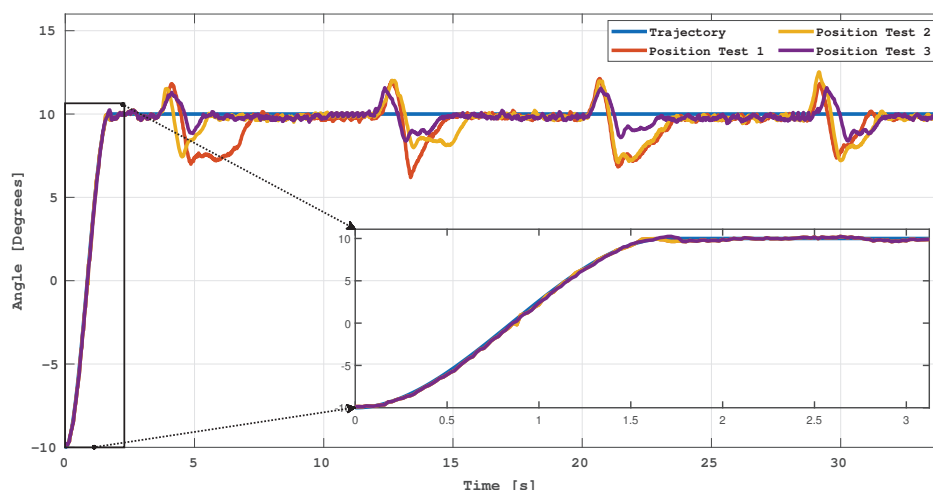


Figure 16. Trajectory tracking robustness tests: -10 to 10 degrees with computational perturbations.

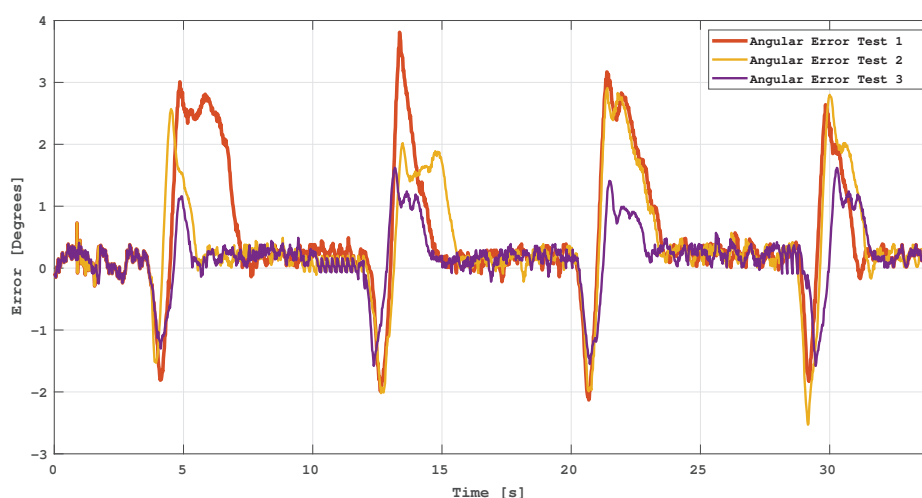


Figure 17. Error [Degrees] during trajectory tracking robustness tests.

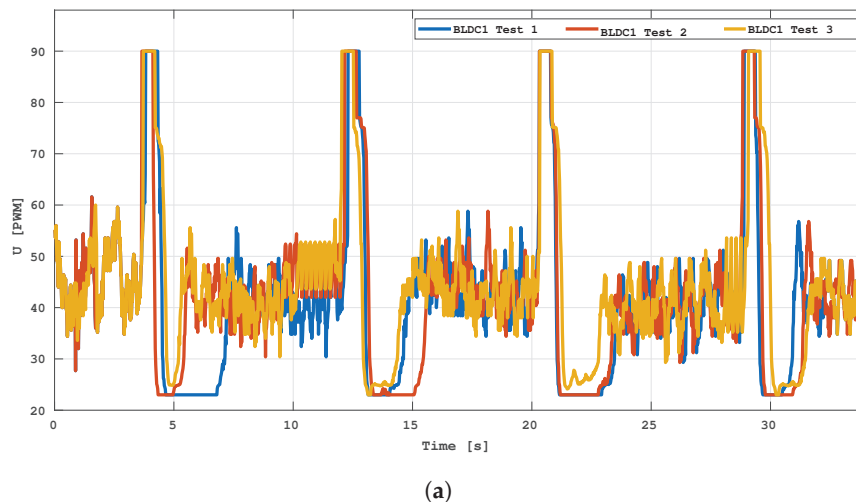
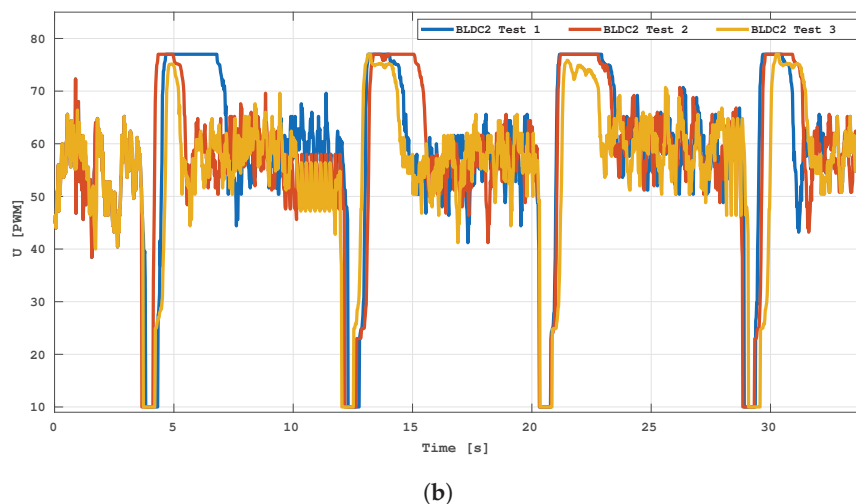
Figure 18. *Cont.*

Figure 18. Control signal [Duty Cycle] of U_1 during trajectory tracking robustness tests: (a) control signal of U_1 ; (b) control signal of U_2 .

5. Discussion

Figure 13 presents several trajectory tracking tests on the proposed control system. In all cases, the settling time was consistently 1.7 s. The first test yielded the lowest root mean square error (RMSE) of 0.2049 and the lowest integral of absolute error (IAE) of 0.9876, while the second test exhibited the smallest overshoot at 1.8333%. The fourth test achieved the lowest steady-state error, with a final deviation of only 0.39 degrees. Complementarily, Figure 14 compares the angular errors across all experiments: the fifth test recorded the highest angular error (0.86 degrees), whereas the fourth test again stood out with the lowest error (0.6 degrees). To validate the trajectory tracking strategy's robustness, 100 experimental trials were performed—50 within the range of -10 to 10 degrees and 50 within -20 to 20 degrees. The TSK fuzzy controller demonstrated consistent performance, achieving low average RMSE values and small standard deviations in both scenarios, reflecting high accuracy and stable system behavior. Although the IAE increased slightly in the broader range, it remained within acceptable margins, confirming the controller's capacity to manage larger reference deviations without significant cumulative error. The average duration of the trajectory tracking trials was approximately 9 s, as summarized in Table 7.

Table 7. Statistical analysis for trajectory tracking.

Metrics	−10 to 10		−20 to 20	
	RMSE	IAE	RMSE	IAE
Mean	0.2687	1.1087	0.2701	1.2933
Standard deviation (σ)	0.0500	0.0739	0.0173	0.0587

For the 10-degree step input test, the overall mean error was 0.2268 with a standard deviation of 0.055, while the mean of individual standard deviations was 0.1428 (± 0.0316), reflecting a consistent and precise system response. These results were obtained from 50 experimental trials, each with an average duration of 14 s, demonstrating the thoroughness of our experimental process. Table 8 presents a comparative analysis with other related methodologies implemented on similar UAV configurations. Four case studies are considered: the first employs an AF2-DOF PID controller for a two-rotor UAV, providing only simulation results; the second focuses on balancing control through optimization of the Q and R parameters in a bicopter, also limited to simulations; the third implements a classical PID controller for experimental bicopter control; and the fourth explores pitch regulation using a state feedback controller, again through simulation. Compared to these approaches, the proposed TSK fuzzy controller offers experimentally validated results and superior tracking precision under real-world conditions.

Table 8. Comparison of several works.

Work	Method	Implementation
[17]	LQ-GA	Simulation
[16]	PID	Experimental
[19]	Adaptative Fuzzy 2-DOF PID	Simulation
[35]	State Feedback	Simulation
Proposed	TSK Fuzzy Tracking	Experimental

Table 9 presents a comparative analysis of key performance metrics—RMSE, IAE, steady-state error (SE), overshoot (OS), and settling time (ST)—between the proposed methodology and several related works. In the study by [19], the controller achieved an SE of 0.05 and an ST of 0.0867 s, with no overshoot reported. While these results indicate a highly efficient response, they are limited to simulation-based validation. Similarly, ref. [17] applied a genetic algorithm for parameter optimization and obtained an OS of 12.8244% and an ST of 7.1450 s, also without steady-state error; however, this approach was likewise validated only through simulation. In contrast, ref. [16] reported experimental results on a bicopter, achieving an SE of 0.25 degrees, an OS of 6.68%, and a settling time of 2.62 s. Another simulation-based study by [35] employed a state feedback controller for pitch regulation, showing an OS of 4.86% and an ST of 1.13 s, with no steady-state error. The methodology proposed in this article differs by incorporating both simulated and experimentally validated results, reporting RMSE and IAE values of 0.2049 and 0.9876, respectively—performance indices not addressed in the studies above. Additionally, it achieved an SE of 0.415 degrees and an OS of 2.8915%, with a target settling time of 1.7 s. Although the proposed method exhibits a higher steady-state error than that reported in [16], it ranks third in settling time among all compared approaches. These results not only reflect the challenges imposed by the system’s inherent nonlinearities and experimental uncertainties but also demonstrate a competitive and practically viable control performance, providing reassurance and confidence in its application.

Table 9. Results comparison of several works: simulated and experimental.

Metrics	[19] Simulation	[17] Simulation	[16] Experimental	[35] Simulation	Proposed Experimental
RMSE	-	-	-	-	0.2049
IAE	-	-	-	-	0.9876
SSE (°)	0.05	0	0.25	0	0.415
OS (%)	0	12.8244	6.68	4.86	2.8915
ST (s)	0.0867	7.1450	2.62	1.13	1.7

6. Conclusions

This paper introduced the design and experimental implementation of a TSK fuzzy controller for trajectory tracking in a UAV bicopter, characterized by strong nonlinearities and dynamic complexity. The proposed strategy demonstrated high tracking accuracy, achieving low RMSE and minimal steady-state deviations across all test scenarios. Quantitative metrics such as RMSE and the integral of absolute error (IAE) validated the controller's precision and robustness. All experiments yielded consistent settling times of approximately 1.7 s. The most favorable trajectory exhibited an RMSE of 0.2049 and an IAE of 0.9876. Furthermore, the controller maintained effective performance under external computational disturbances. When a sudden perturbation was introduced to one of the rotors, the system quickly stabilized and returned to the desired trajectory, with a maximum angular deviation of only 3.81 degrees. These results highlight the controller's strong disturbance rejection capabilities and ability to ensure stable operation in real-world conditions. The practical implementation confirmed the viability of the proposed TSK fuzzy approach, reinforcing its relevance for applications requiring real-time, reliable control in uncertain and nonlinear environments. In summary, the results underscore the effectiveness of the TSK controller as a robust solution for precise trajectory tracking in UAV systems. Beyond the current scope, this case study opens several avenues for future research. One direction involves developing an adaptive version of the controller that can modify its fuzzy rules in real-time based on changing system dynamics. Another promising extension is the integration of neuro-fuzzy architectures, enabling learning from simulations and experimental data. Perhaps the most ambitious line of inquiry is scaling the approach to multicopter platforms—tricopters, quadcopters, or hexacopters—thus enabling three-dimensional trajectory tracking through coordinated control of altitude, pitch, and roll.

Author Contributions: Conceptualization, J.R.R.-R. and J.R.G.-M.; methodology, J.R.R.-R., J.R.G.-M. and E.E.C.-M.; software, J.R.R.-R., J.R.G.-M. and T.M.-S.; validation, J.R.R.-R., J.R.G.-M. and Á.O.; formal analysis, J.R.G.-M., Á.O., L.D.R.-G. and O.A.B.-V.; investigation, J.R.R.-R., J.R.G.-M. and T.M.-S.; resources, E.E.C.-M. and L.D.R.-G.; data curation, J.R.R.-R., J.R.G.-M., E.E.C.-M. and Á.O.; writing—original draft preparation, J.R.R.-R. and J.R.G.-M.; writing—review and editing, J.R.G.-M., E.E.C.-M. and Á.O.; visualization, J.R.G.-M. and E.E.C.-M.; supervision, J.R.G.-M. and Á.O.; project administration, J.R.G.-M.; funding acquisition, J.R.G.-M., T.M.-S., E.E.C.-M., L.D.R.-G. and O.A.B.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Acknowledgments: The first author, an undergraduate student, thanks the “Facultad de Ingeniería en Electrónica y Comunicaciones” at the “Universidad Veracruzana” for supporting the publication of this scientific article. We would also like to thank Grammarly Premium, a grammar-checking software, for its general review of the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
TSK	Takagi–Sugeno–Kang
PID	Proportional–Integrative–Derivative
GA	Genetic Algorithm
BLDCM	Brushless Direct Current Motor
PSO	Particle Swarm Optimization
DFLC	Direct Fuzzy Logic Controller
FTSMC	Fast Terminal Sliding Mode Control
LQR	Linear Quadratic Regulator
IMU	Inertial Measurement Unit
ESC	Electronic Speed Controller
VLN	Very Large Negative
LN	Large Negative
MLN	Moderately Large Negative
MN	Medium Negative
MSN	Moderately Small Negative
SN	Small Negative
VSN	Very Small Negative
Z	Zero
VSP	Very Small Positive
SP	Small Positive
MSP	Moderately Small Positive
MP	Medium Positive
MLP	Moderately Large Positive
LP	Large Positive
VLP	Very Large Positive
RMSE	Root Square Mean Error
IAE	Integral Absolute Error
OS	Overshoot
PWM	Pulse Width Modulation
SSE	Steady-State Error
ST	Settling Time

References

1. He, X.; Wang, Y. Design and trajectory tracking control of a new bi-copter UAV. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9191–9198. [CrossRef]
2. Vundela, V.R.; Muralidharan, V. Nonlinear attitude and altitude trajectory tracking control of a bi-copter UAV in geometric framework. *J. Robot.* **2024**, *2024*, 5552493. [CrossRef]
3. Beliautso, V.; Beliautso, A. Prop-plane—New convertible VTOL UAV as a combination of a longitudinal bi-copter and a flying wing with a tilt-rotor powertrain. *Aerosp. Sci. Technol.* **2024**, *155*, 109650. [CrossRef]
4. Zhang, Y.; Wang, G.; Zhou, T.; Huang, X.; Lam, S.; Sheng, J.; Choi, K.S.; Cai, J.; Ding, W. Takagi–Sugeno–Kang fuzzy system fusion: A survey at hierarchical, wide and stacked levels. *Inf. Fusion* **2024**, *101*, 101977. [CrossRef]
5. Ouifak, H.; Idri, A. On the performance and interpretability of Mamdani and Takagi–Sugeno–Kang based neuro-fuzzy systems for medical diagnosis. *Sci. Afr.* **2023**, *20*, E01610. [CrossRef]
6. Jovanović, R.; Zarić, V.; Bučevac, Z.; Bugarić, U. Discrete-Time System Conditional Optimization Based on Takagi–Sugeno Fuzzy Model Using the Full Transfer Function. *Appl. Sci.* **2022**, *12*, 7705. [CrossRef]
7. Subramani, S.; Vijayarangan, K.K.; Chenniappan, M. Improved African Buffalo Optimization-Based Takagi–Sugeno–Kang Fuzzy PI Controller for Speed Control in BLDC Motor. *Electr. Power Components Syst.* **2023**, *51*, 1948–1962. [CrossRef]

8. Aslam, S.; Chak, Y.-C.; Jaffery, M.H.; Varatharajoo, R.; Ansari, E.A. Model predictive control for Takagi–Sugeno fuzzy model-based Spacecraft combined energy and attitude control system. *Adv. Space Res.* **2023**, *71*, 4155–4172. [CrossRef]
9. Ozcan, O.F.; Kilic, H.; Ozguven, O.F. A unified robust hybrid optimized Takagi–Sugeno fuzzy control for hydrogen fuel cell-integrated microgrids. *Int. J. Hydrogen Energy* **2025**, *in press*. [CrossRef]
10. Nguyen, V.-T.; Duong, D.-N.; Pham, D.-H.; Ngo, V.-T.; Tuan, L.A. Optimal nonlinear PID TSK3DCMAC controller based on balancing composite motion optimization for ballbot with external forces. *ISA Trans.* **2025**, *158*, 654–673. [CrossRef]
11. Tripathi, R.P.; Singh, A.K.; Gangwar, P.; Verma, R.K. Sensorless speed control of DC motor using EKF estimator and TSK fuzzy logic controller. *Automatika* **2022**, *63*, 338–348. [CrossRef]
12. Zaman, M.Q.; Wu, H.-M. Genetic Algorithm Based Takagi–Sugeno Fuzzy Modelling of a Mecanum Wheeled Mobile Robot. *Int. J. Control. Autom. Syst.* **2025**, *23*, 907–919. [CrossRef]
13. Ramezani, M.S.; Ajami, S.; Ghafarirad, H. Path-tracking control of quadrotor using adaptive critic-based neurofuzzy controller. In Proceedings of the 2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 17–19 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 248–254.
14. Lou, Q.; Deng, Z.; Xiao, Z.; Choi, K.-S.; Wang, S. Multilabel Takagi–Sugeno–Kang fuzzy system. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 3410–3425. [CrossRef]
15. Bhar, A.; Mezni, Y.; Sayadi, M. Tuning PID control parameters: A combined experimental and simulation study. In Proceedings of the 2024 International Conference on Control, Automation and Diagnosis (ICCAD), Paris, France, 15–17 May 2024; Volume 1, pp. 1–6.
16. Jayadi, A.; Adhinata, F.D.; Sembiring, J.P.; Putri, N.U.; Pranita, E.; Setiawan, A. Implementation of PID Control as a One Axis Balancing Control System on the Bicopter Vehicle. In Proceedings of the 2023 International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE), Bandar Lampung, Indonesia, 25–26 October 2023; Volume 1, pp. 113–117.
17. Apriaskar, E.; Prastiyanto, D.; Manaf, A.A.; Amelia, I. Multi-criteria Genetic Algorithm Optimization Approach for Balancing Bicopter Control. *IOP Conf. Ser. Earth Environ. Sci.* **2023**, *1203*, 012027. [CrossRef]
18. Phelps, A.; Salazar, J.A.P.; Goel, A. Data-driven Fuzzy Control for Time-Optimal Aggressive Trajectory Following. *arXiv* **2025**, arXiv:2504.06500.
19. Cukdar, I.; Yigit, T.; Celik, H. Balance Control of Brushless Direct Current Motor Driven Two-Rotor UAV. *Appl. Sci.* **2024**, *14*, 4059. [CrossRef]
20. Jennan, N.; Mellouli, E. M.; Žuraulis, V. Direct Fuzzy Logic Controller Combined with Fast Terminal Sliding Mode for a Bicopter. In Proceedings of the 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Fez, Morocco, 16–17 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
21. Fahmizal, F.; Nugroho, H.A.; Cahyadi, A.I.; Ardiyanto, I. Fuzzy Logic Controller for Stabilizing the Rolling Movement of UAV Bicopter. *ICIC Express Lett.* **2023**, *17*, 967–978.
22. Marques, F.M.M.; Alves, G.R.O.; Silva, G.H.C.E.; de Assis, P. A.Q. Model-Based Predictive Controller Applied to a 3 DOF Tilt-Rotor Bi-copter Test Bench. *J. Control. Autom. Electr. Syst.* **2024**, *35*, 474–484. [CrossRef]
23. Li, W.; Liu, Y.; Yu, Y.; Chen, J. Optimized Cascade Control Strategy for Bi-copter UAV Based on Fuzzy-PID and ADRC. In *International Conference on Guidance, Navigation and Control*; Springer Nature: Singapore, 2024; pp. 411–420.
24. Prakosa, J. A.; Wang, H.; Kurniawan, E.; Agmal, S.; Kholili, M. J. Experimental Studies of Linear Quadratic Regulator (LQR) Cost Matrices Weighting to Control an Accurate Take-off Position of Bicopter Unmanned Aerial Vehicles (UAVs). *J. Mechatron. Electr. Power Veh. Technol.* **2022**, *13*, 101–112. [CrossRef]
25. Apriaskar, E.; Prastiyanto, D.; Utomo, A. B.; Manaf, A. A.; Amelia, I.; Ilham, D. A.; Photong, C. Genetic Algorithm-Enhanced Linear Quadratic Control for Balancing Bicopter System with Non-Zero Set Point. *J. Mechatronics Electr. Power, Veh. Technol.* **2023**, *14*, 105–113. [CrossRef]
26. Arrofiq, M.; Apriaskar, E.; Mayub, A. Rigorous Modelling Steps on Roll Movement of Balancing Bicopter using Multi-Level Periodic Perturbation Signals. In Proceedings of the 2019 6th International Conference on Instrumentation, Control, and Automation (ICA), Bandung, Indonesia, 31 July–2 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 52–57.
27. Rojas-Galván, R.; García-Martínez, J.R.; Cruz-Miguel, E.E.; Barra-Vázquez, O.A.; Olmedo-García, L.F.; Rodríguez-Reséndiz, J. Hunting Search Algorithm-Based Adaptive Fuzzy Tracking Controller for an Aero-Pendulum. *Technologies* **2024**, *12*, 63. [CrossRef]
28. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132. [CrossRef]
29. Yoneyama, J.; Nishikawa, M.; Katayama, H.; Ichikawa, A. Output stabilization of Takagi–Sugeno fuzzy systems. *Fuzzy Sets Syst.* **2000**, *111*, 253–266. [CrossRef]
30. Kiew-ong-art, M.; Chotikunnan, P.; Wongkamhang, A.; Chotikunnan, R.; Nirapai, A.; Imura, P.; Sangworasil, M.; Thongpance, N.; Srisiriwat, A. Comparative Study of Takagi-Sugeno-Kang and Madani Algorithms in Type-1 and Interval Type-2 Fuzzy Control for Self-Balancing Wheelchairs. *Int. J. Robot. Control Syst.* **2023**, *3*, 643–657. [CrossRef]
31. Rehman, A.; Singh, A.K.; Singh, A.; Dubey, E.P. Mamdani and Sugeno fuzzy logic approach of PID controller. *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)* **2022**, *10*. [CrossRef]

32. Kang, G.; Lee, W.; Sugeno, M. Design of TSK fuzzy controller based on TSK fuzzy model using pole placement. In Proceedings of the 1998 IEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 4–9 May 1998; Volume 1, pp. 246–251. [CrossRef]
33. Surono, S.; Wen, G. K.; Onn, C.W.; Dasril, Y.B.; Astuti, A.Y.; Periasamy, N. Effectiveness of Enhanced Takagi Sugeno Kang’s Fuzzy Inference Model. *J. Theor. Appl. Inf. Technol.* **2022**, *100*, 2383–2392.
34. Olmedo-García, L.F.; García-Martínez, J.R.; Cruz-Miguel, E.E.; Barra-Vázquez, O.A.; González-Lee, M.; Martínez-Sánchez, T. Real-Time Embedded System-Based Approach for Sensing Power Consumption on Motion Profiles. *Electronics* **2023**, *12*, 3853. [CrossRef]
35. Uddin, N.; Harno, H.G.; Manurung, A.; Nasucha, M. A Pitch Control System Design for Bicopter UAVs. In Proceedings of the 2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE), Semarang, Indonesia, 23–24 September 2021, Piscataway, NJ, USA, 2021; pp. 57–62.

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

Image-Tracking-Driven Symmetrical Steering Control with Long Short-Term Memory for Linear Charge-Coupled-Device-Based Two-Wheeled Self-Balancing Cart

Yi-Jen Mon

Department of Electronic Engineering, Ming-Chuan University, Guei-Shan District, Taoyuan City 333, Taiwan; yjm@mail.mcu.edu.tw

Abstract: This paper presents a control framework for the image tracking of two-wheeled self-balancing carts, with the objective of achieving precise tracking control. Exploiting the remarkable memory capacity of the Long Short-Term Memory (LSTM) neural network for sequence signals, the framework conducts image memory judgment and memorization, aiming to enhance control accuracy. After the training phase, comprehensive simulations and real-world experiments are carried out based on the established model to verify the effectiveness and practicality of the proposed control strategy. The system utilizes the TSL1401 linear array CCD lens to detect black tapes on the ground and identify and memorize surrounding images. Through the establishment of a continuous set of training sample points, the LSTM network is trained using Python and TensorFlow. This training process optimizes the network's weights and generates weight files, which can be readily converted into machine code for physical implementation. Initially, the effectiveness of the control law is verified through simulating the symmetrical steering control of the two-wheeled cart. The simulation results demonstrate the validity of the proposed design method and its superior performance. Finally, a physical two-wheeled self-balancing cart is developed to further validate the feasibility of the framework. Experimental results confirm that this method is highly effective, demonstrating robust image tracking capabilities and optimal tracking performance.

Keywords: two wheeled; image tracking; control; Python; Tensorflow; LSTM; fuzzy; neural networks; cart; symmetry; symmetrical steering

1. Introduction

In recent years, two-wheeled self-balancing carts have attracted significant attention due to their potential applications in various fields, such as intelligent transportation, logistics automation, and service robotics [1–4]. These systems have inherently unstable structures which are similar to inverted pendulums. Thus, sophisticated control algorithms are required to maintain balance and ensure accurate steering. Traditional control methods, including PID (Proportional–Integral–Derivative) control, fuzzy logic control, and some other complex algorithmic control approaches [5,6], have been widely adopted. However, these methods often face challenges when dealing with complex and dynamic environments. They rely on predefined rules and fixed parameters, which may not adapt well to changing conditions, especially in the context of symmetrical steering control.

With the rapid development of computer vision technology, image-based control strategies have emerged as a promising approach for self-balancing carts. By utilizing

cameras, these systems can acquire rich environmental information, enabling more intelligent and adaptive control. Linear CCD (Charge-Coupled Device) sensors, in particular, offer high-resolution linear imaging capabilities, which are advantageous for precise path tracking and obstacle detection [7–10]. Nevertheless, most existing image-based control methods for two-wheeled self-balancing carts focus on short-term motion prediction and lack the ability to handle long-term dependencies and temporal dynamics within image sequences.

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), have demonstrated outstanding performance in processing sequential data with long-term dependencies. LSTM networks can selectively remember and forget information over time, making them suitable for tasks that require an understanding of temporal context, such as speech recognition, time series prediction, and video analysis [11–15]. Although recently, models based on TimeGPT and other advanced architectures have shown better performance than LSTM-based models [16–18], the LSTM is still used in this paper for the following reasons: Firstly, the data used in this research has specific characteristics and constraints. LSTM models are well studied and have proven effective in handling sequential data with temporal dependencies, which aligns well with the nature of the linear CCD data collected from the two-wheeled vehicle. In contrast, TimeGPT-based models, although promising, require a large amount of data for training and fine-tuning, which was not available in this study due to the limited data collection resources. Secondly, the computational resources and time available for this study were also factors. LSTM-based models are relatively more computationally efficient compared to the complex architectures of TimeGPT-based models, allowing for the timely completion of the research. Finally, the novelty of this study lies in the innovative way of preprocessing CCD data for LSTM-based turning judgment. While TimeGPT-based models are an interesting area of research, they do not directly contribute to the main objectives of this study.

Some of the latest studies about two-wheeled cart control are detailed as follows [19–22]: In [19], by training the model with historical motion and environmental data, it is able to predict the future states of the mobile robot. The proposed approach is expected to improve the robot's adaptability and control performance. In [20], the focus of the research is on the coordination of the robot's wheels, its body posture, and its interaction with the environment. By formulating a comprehensive control algorithm, the authors aim to improve the robot's performance in terms of balance maintenance, maneuverability, and load-carrying capacity. In [21], based on their model, an optimal control strategy is proposed to maintain the robot's balance and achieve desired motions. The control approach likely aims to minimize certain cost functions related to energy consumption, balance error, and tracking performance. In [22], the feedback mechanism is integrated to enhance the controller's performance by continuously adjusting the control signals according to the actual system states. This combination aims to improve the system's response time, stability, and robustness against disturbances. The authors conduct extensive simulations and, possibly, real-world experiments to evaluate the performance of the proposed controller.

In this paper, we propose a novel symmetrical steering control method for linear CCD-based two-wheeled self-balancing carts that integrates image tracking and LSTM networks. The image tracking module extracts dynamic features from linear CCD images, while the LSTM network models the long-term temporal relationships in the tracking data to predict future motion states. By combining these two components, our approach aims to enhance the control performance of self-balancing carts in complex and dynamic environments, overcoming the limitations of traditional control methods and existing image-based approaches. The contributions of this study include the development of

an LSTM-based control framework for self-balancing carts and the integration of image tracking and LSTM for more accurate motion prediction.

During the implementation process, we first use the Python language [23] to train the LSTM neural network. After training is completed, we use the TensorFlow Lite 2.1 [24] package to convert the LSTM weights into C language and integrate them with the cart control code using the Keil IDE (Integrated Development Environment) uVision 5 [25]. Experimental results on two-wheeled self-balancing carts [26] validate the effectiveness and superiority of the proposed method. The design objectives of this paper are to prove the feasibility of the method through simulation and then implement the method using an actual two-wheeled self-balancing cart for symmetrical operation.

2. Designing Symmetrical Steering Motion of Cart

Similarly to forward motion, we can also establish a simplified symmetrical steering motion model, as shown in Figure 1 [26]. The symmetrical steering motion is induced by the unequal magnitudes of the reaction forces F_L and F_R exerted on the cart body by the left and right wheels in the horizontal direction. Based on the law of rotation of a rigid body with a fixed axis, we can obtain the fundamental equations that govern the rotational motion of the object. These equations form the foundation for the subsequent experimental verification and simulation analysis, which I will elaborate upon in the following paragraphs.

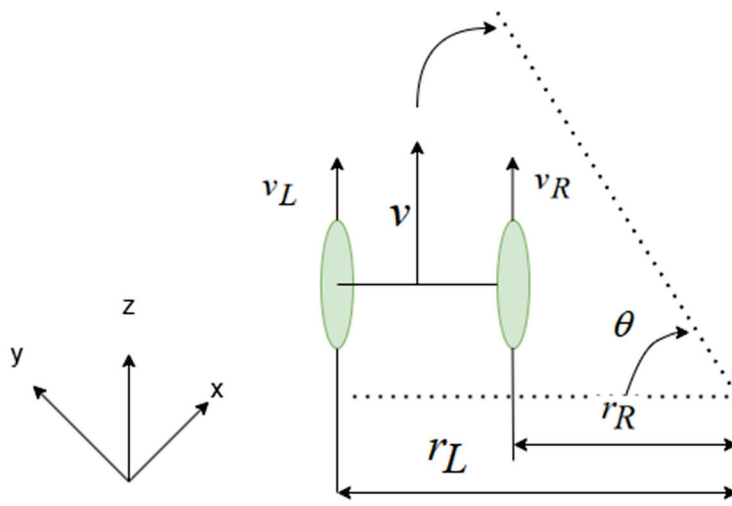


Figure 1. The symmetrical steering diagram of the two-wheeled self-balancing cart.

Similarly to the forward-motion model, we can also establish a simplified symmetrical steering motion model, as shown in Figure 1. The symmetrical steering motion arises from the unequal horizontal reaction forces F_L and F_R exerted on the cart body by the left and right wheels. By applying the rotational form of Newton's second law to a rigid body with a fixed axis, we obtain the following dynamic equation:

$$J_{\theta} \ddot{\theta} = \frac{d}{2} (F_L - F_R) \quad (1)$$

After considering the displacement relationship between the left and right wheels, the dynamic equation is established as follows: When the speeds of the left and right wheels are not equal, the cart body turns, as shown in Figure 1. From the geometric relationship, we can derive

$$\left(m + \frac{I}{r^2}\right) (\ddot{x}_L - \ddot{x}_R) = \frac{T_L - T_R}{r} - (F_L - F_R) \quad (2)$$

$$\dot{x}_L = \dot{\theta} r_L \quad (3)$$

$$\dot{x}_R = \dot{\theta} r_R \quad (4)$$

$$r_L = r_R + d \quad (5)$$

After subtracting Equation (2) from Equation (1), we obtain:

$$\dot{\theta} = \frac{\dot{x}_L - \dot{x}_R}{d} \quad (6)$$

$$\ddot{\theta} = \frac{\ddot{x}_L - \ddot{x}_R}{d} \quad (7)$$

Then, the dynamic model of symmetrical steering of the cart can be obtained as

$$\ddot{\theta} = \frac{1}{r(md + \frac{Id}{r^2} + \frac{2J_\theta}{d})} (T_L - T_R) \quad (8)$$

Since the magnitude of the motor output torque is difficult to control directly, it is converted into the acceleration of the two wheels according to the law of rigid body fixed-axis rotation

$$\dot{v}_{Ro} = \frac{r}{I} T_R \quad (9)$$

$$\dot{v}_{Lo} = \frac{r}{I} T_L \quad (10)$$

$$\ddot{\theta} = \frac{1}{I(md + \frac{Id}{r^2} + \frac{2J_\theta}{d})} (\dot{v}_{Lo} - \dot{v}_{Ro}) \quad (11)$$

$$\ddot{\theta} = fu + dis \quad (12)$$

where $f = \frac{1}{I(md + \frac{Id}{r^2} + \frac{2J_\theta}{d})}$, $u = (\dot{v}_{Lo} - \dot{v}_{Ro})$. The *dis* is defined as the disturbances and the uncertainties.

The parameters of model are defined as follows:

x_R : Horizontal displacement of right wheel (m);

x_L : Horizontal displacement of left wheel (m);

θ : Yaw angle of the cart (rad);

J_θ : The moment of inertia of the cart body when rotating around the z-axis ($\text{kg} \times \text{m}^2$);

m : Wheel mass (kg);

r : Radius of wheel (m);

r_R : Distance from right wheel to rotation center (m)

r_L : Distance from left wheel to rotation center (m)

v : Velocity of cart (m/s);

v_R : Velocity of right wheel (m/s);

v_L : Velocity of left wheel (m/s);

$\dot{v}_{Ro}$: The right wheel acceleration which is also the right control output (m/s^2);

$\dot{v}_{Lo}$: The left wheel acceleration which is also the left control output (m/s^2);

d : Distance between wheels (m);

T_R : The output torque of the right wheel motor ($\text{N} \cdot \text{m}$);

T_L : The output torque of the left wheel motor ($\text{N} \cdot \text{m}$);

F_R : The magnitude of the horizontal component of the force acting on the right wheel (N);

F_L : The magnitude of the horizontal component of the force acting on the left wheel (N);

I : Moment of inertia of the wheel ($\text{kg} \times \text{m}^2$).

To prove the stability of the designed system, we first define its error as follows:

$e = \theta_d - \theta$. Then, based on this error, we design an ideal controller as follows:

$$u_{ideal} = f^{-1}(\ddot{\theta}_d - \dot{dis} + k_1\dot{e} + k_2e) \quad (13)$$

Substituting this equation into Formula (7), we can derive the error dynamic equation as follows:

$$\ddot{e} + k_1\dot{e} + k_2e = 0 \quad (14)$$

According to Laplace transformation, as long as k_1 and k_2 are selected correctly so that the characteristic roots of Equation (5) are located in the left half-plane, the system is guaranteed to be stable, and the system error will converge, i.e.: $\lim_{t \rightarrow \infty} e(t) = 0$.

The ideal controller u_{ideal} involves excessive uncertainties, necessitating the design of a robust controller u_{LSTM} to address these challenges.

$$u_{ideal} = k_1\dot{e} + k_2e + u_{LSTM} \quad (15)$$

The control forces for the left and right wheels are determined by the symmetrical steering direction. For instance, when turning right, the left wheel must accelerate; when turning left, the right wheel must accelerate. Their relationship with the ideal control input u_{ideal} is as follows:

$$\dot{v}_{Lo} = u_{ideal} + \dot{v}_{Ro} \quad (16)$$

$$\dot{v}_{Ro} = \dot{v}_{Lo} - u_{ideal} \quad (17)$$

3. LSTM Neural Network Design and Simulation Results

First, the characteristics of the linear CCD in TSL 1401 [26] need to be introduced. Figure 2 illustrates the structure of the TSL1401, a linear image sensor celebrated for its high sensitivity and extensive dynamic range, rendering it well suited for light detection applications. It consists of 128 photodiodes arranged in a single row, allowing for highly precise light intensity detection. With low dark current and high responsivity, the sensor ensures reliable performance even in low-light conditions. Moreover, its simple serial interface simplifies its integration with microcontrollers and other digital systems.

Each photodiode in the array converts incident light into an electrical charge, with the amount of charge generated being proportional to the local light intensity. These charges are then sequentially transferred and processed. The sensor samples the charge from each photodiode, converts it into a voltage-proportional signal, and outputs the data serially. By analyzing the output values, users can determine the light distribution across the 128-element array, which finds application in various scenarios, including light-level monitoring, edge detection, and optical signal analysis.

This paper utilizes this CCD to track the black tape on the ground and control the direction of the cart, with the goal of enabling the cart to move along the black tape. Previously, infrared sensors were predominantly used due to their short-range detection and relatively low susceptibility to interference. Control laws such as PID and LQR were applied to ensure the cart moved stably along the tape. In this study, image-based tracking is adopted, presenting significant challenges. Images are vulnerable to environmental brightness, reflections, and the reflective background of the tape, all of which can impact tracking accuracy. Therefore, simulations are first conducted, and a neural-network-like LSTM is employed to memorize the cart's movement patterns. Similarly to human cognitive abilities, the system can still identify the position of the black tape in ambiguous environments.

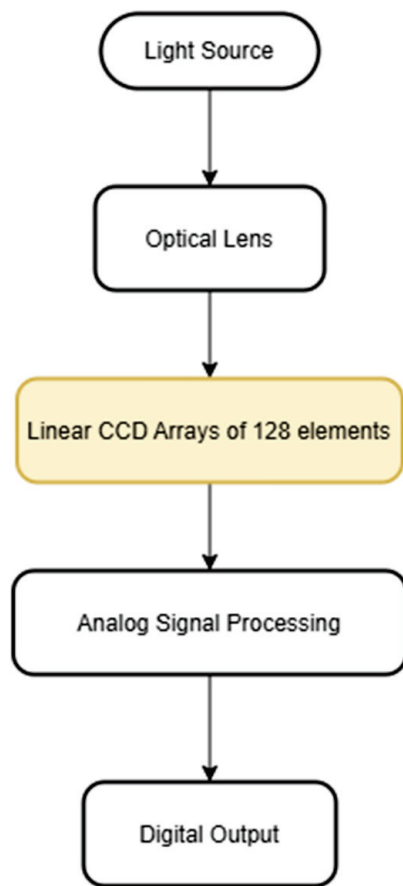


Figure 2. The diagram of the TSL 1401 structure.

We select 64 elements from the middle of the 128-element CCD as the target values. Specifically, the control objective is to maintain the lowest voltage value at the 64th position, as shown in Figure 3. To accomplish this, we create learning samples for an LSTM network to be trained on. LSTM networks are particularly adept at predicting time-series data, making them highly suitable for this task.

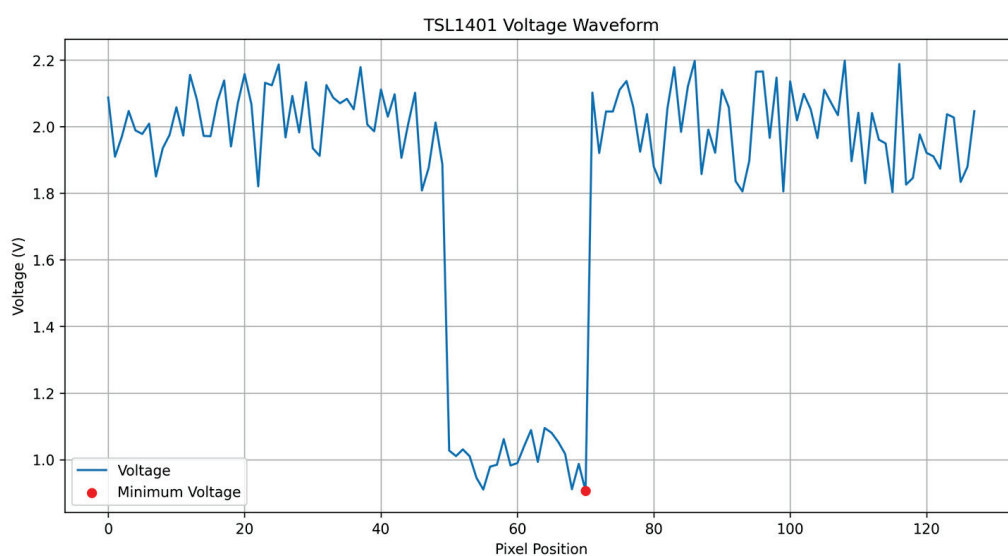


Figure 3. The correct waveform of TSL 1401.

Python is an ideal programming language for this task due to its simplicity, readability, and extensive libraries. Its high-level syntax allows for rapid development, enabling

researchers to quickly prototype and iterate on the LSTM model. With libraries like NumPy and Pandas, data preprocessing, manipulation, and analysis become straightforward, which are crucial for handling the linear CCD data used in this paper. TensorFlow, on the other hand, provides a powerful and flexible framework for building and training neural networks. It offers high-level APIs, such as Keras, that simplify the construction of complex models like LSTM, making it accessible even for those new to deep learning. TensorFlow also supports automatic differentiation, which streamlines the computation of gradients during the training process, significantly reducing the development time. Additionally, it has excellent support for distributed training, which can accelerate the training of large-scale LSTM models. TensorFlow's compatibility with various hardware platforms, including GPUs and TPUs, further boosts the training efficiency.

The neural network architecture consists of one LSTM layer followed by a densely connected layer, as illustrated in Figure 4, with its operational flow depicted in Figure 5. We implement this model using Python [23]. To mitigate overfitting and ensure reliable predictions, we carefully regulate the training batch size. The model is trained with a learning rate of 0.001 for 20 epochs, with a batch size of 32. After training, simulations are conducted. The model consists of an LSTM layer with 32 units followed by two Dense layers, and the activation function of the output layer is 'softmax'. Regarding the hyperparameters, the learning rate is initially set at 0.001 and tuned within the range of 0.0001–0.01. Through the 'adam' optimizer and a grid search combined with cross-validation, the optimal learning rate is determined to be 0.001. The final accuracy of the model is 0.9, and the loss is 0.25. The training process is depicted in Figure 6.

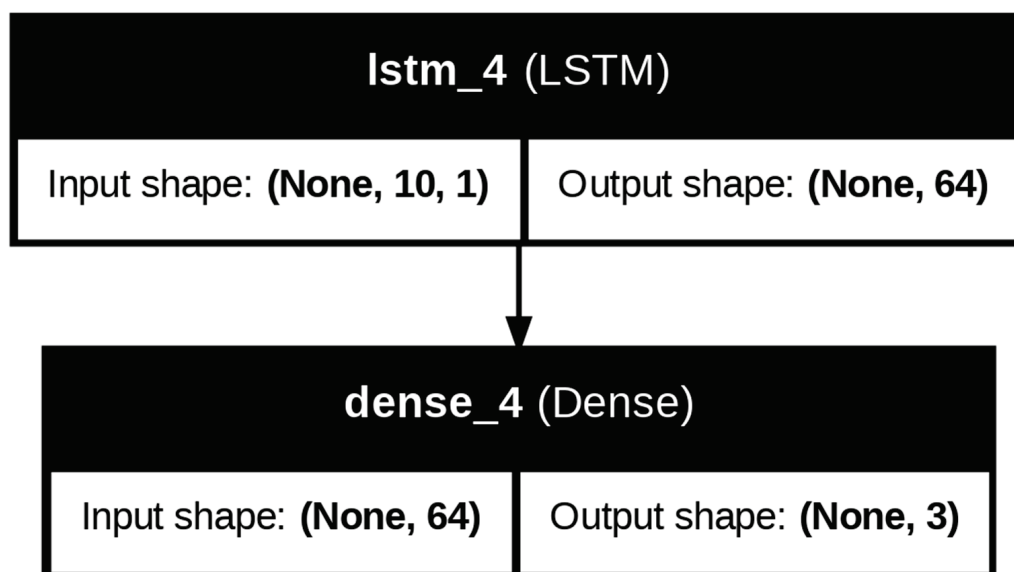


Figure 4. The LSTM neuron architecture.

We iteratively generate 128 training points within the range of 35 to 95, enabling the LSTM model to learn three types of movements: moving forward (output 0), turning right (output 10), and turning left (output −10), as shown in Figure 7. The results clearly show that when the input value is around 64, the model outputs 0, indicating forward movement; when the input is approximately 40, it outputs −10, signifying a left turn; and when the input is near 80, the output is 10, corresponding to a right turn.

In the absence of the LSTM component, the system loses its predictive ability. As demonstrated in Figure 8, just relying solely on a PD (proportional-differential) controller leads to subpar control performance. This method involves using the distance between the image point and the mid-point as a parameter, and then considering its differential term

for simple traditional control. In contrast, Figure 9 presents the simulation results after integrating the trained LSTM model and its learned weights. The results clearly indicate that the inclusion of LSTM enables the cart to follow the desired path accurately, while without it, the cart fails to execute proper turns and moves erratically. The reason for this is mainly due to the unique characteristics of the data and the problem at hand. The data source, e.g., linear CCD data for the two-wheeled cart used in this study contains complex, non-linear, and time-varying patterns that are difficult to model accurately using traditional control methods.

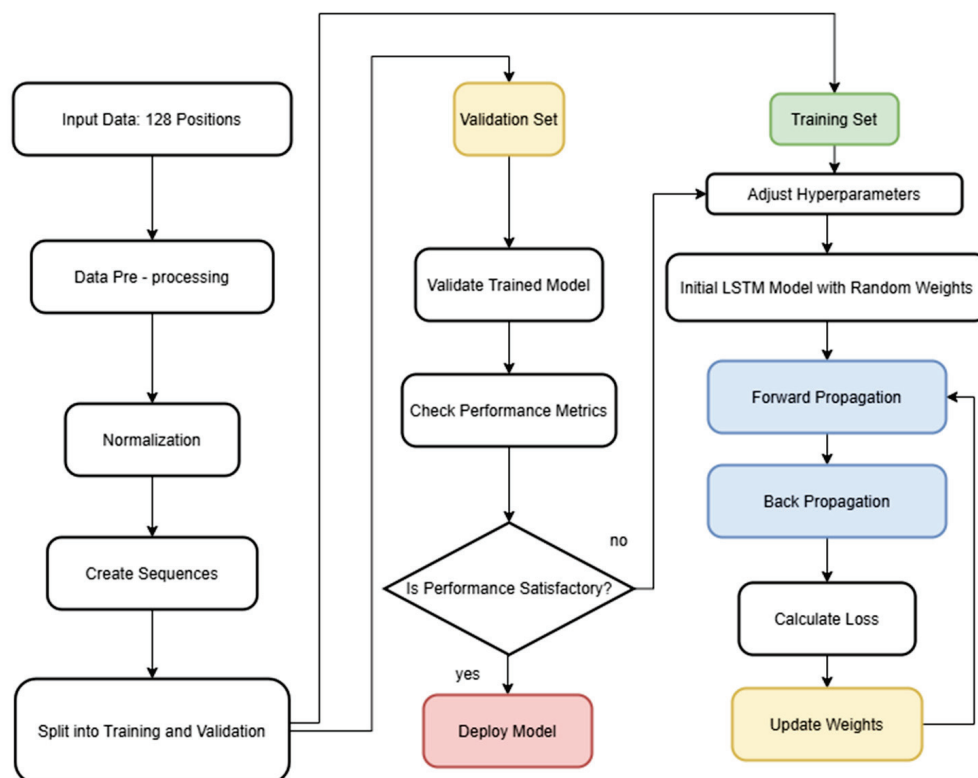


Figure 5. LSTM operation flow chart.

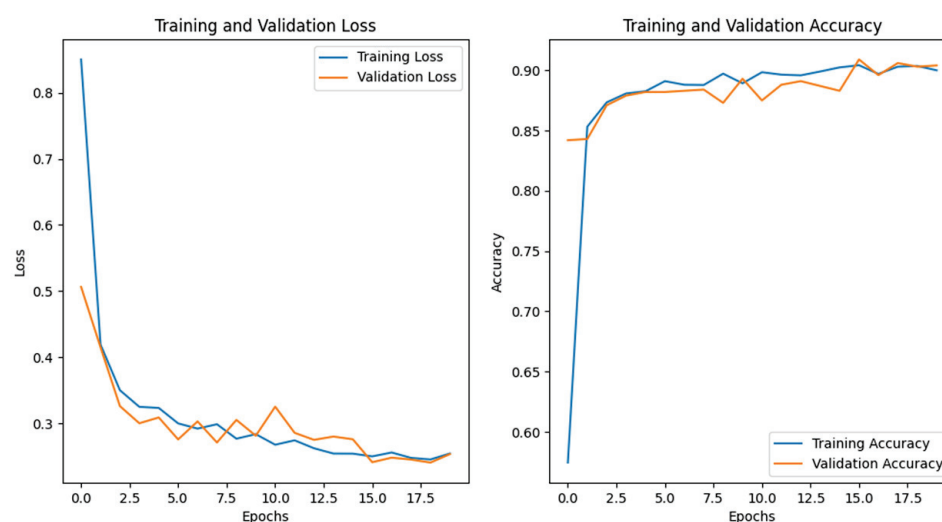


Figure 6. LSTM training process diagram.

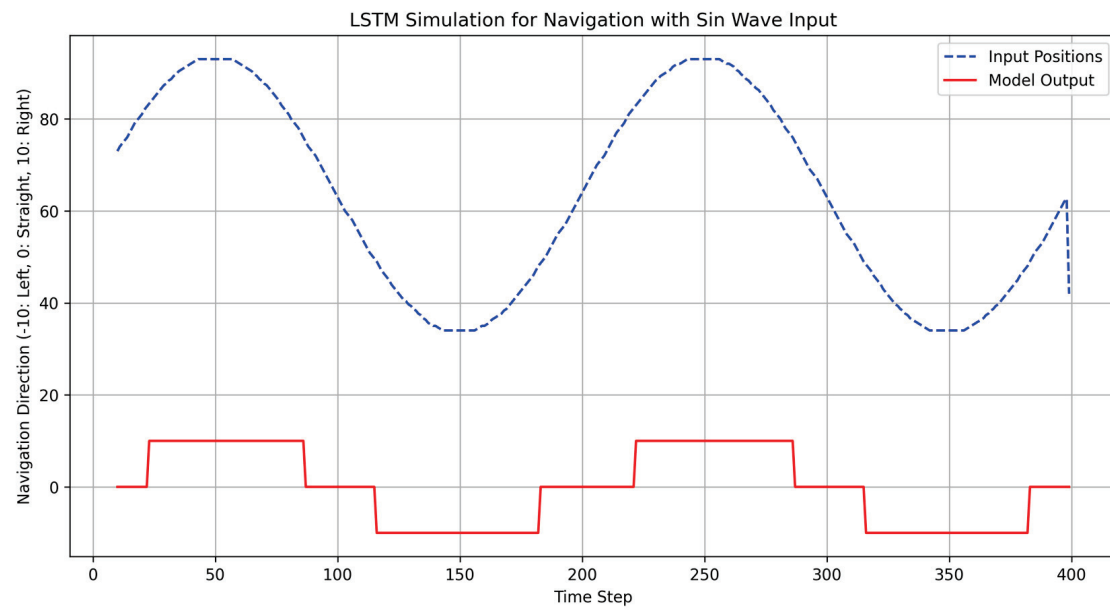


Figure 7. LSTM training results diagram.

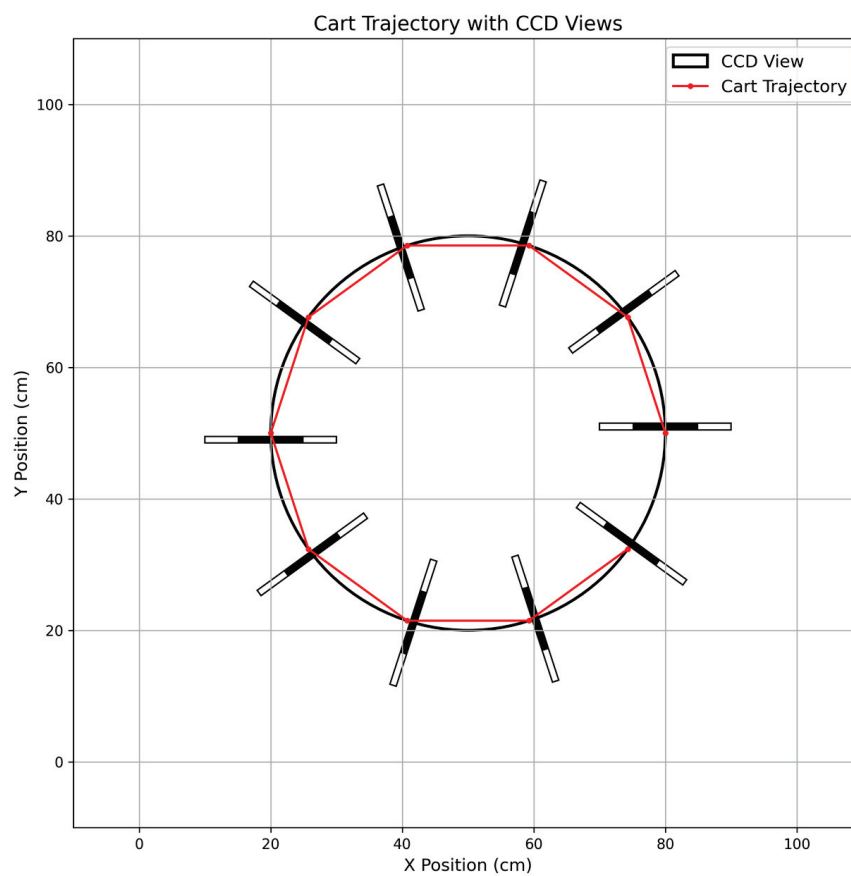


Figure 8. Cart trajectory diagram without LSTM but with PID control.

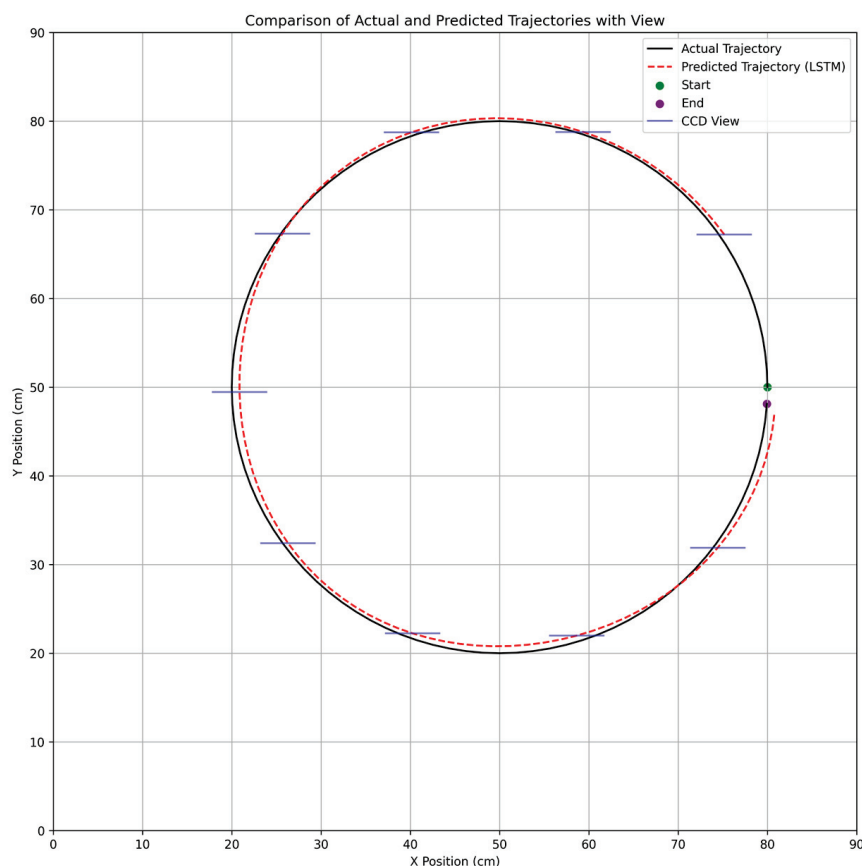


Figure 9. Cart trajectory diagram with LSTM.

4. Practical Implementation Results

The cart body used in the actual experimental implementation is shown in Figure 10. In real-world applications, especially when implementing the model on embedded systems or devices with limited computational resources, such as the two-wheeled self-balancing cart in this study, machine code offers significant advantages. Machine code is the lowest level of programming representation that can be directly executed by the hardware. By converting the trained weights into machine code, we can eliminate the overhead associated with higher level programming languages and frameworks. This results in faster execution speeds, reduced memory consumption, and improved real-time performance, which are crucial for the real-time control and operation of the self-balancing cart. Moreover, machine code enables better integration with the specific hardware architecture of the target device. It allows for fine-grained optimization tailored to the unique features and capabilities of the hardware, further enhancing the overall efficiency of the model.

When moving from simulation to real-world testing, several significant challenges emerge. Firstly, the simulation environment is often an idealized representation that fails to account for the complex and unpredictable real-world factors. For instance, in the case of the two-wheeled self-balancing cart, simulation may not accurately replicate variations in terrain, ambient noise, and interference from external electromagnetic fields. These real-world elements can significantly affect the performance of the control system based on the trained LSTM model. Secondly, hardware limitations pose a major hurdle. In real-world testing, the physical sensors and actuators of the self-balancing cart may have inherent inaccuracies, non-linearities, and signal delays that are not present in simulations. Calibrating these components to work in harmony with the model can be extremely challenging and time-consuming. Another critical factor is data deviation. The data collected during real-world testing may have different statistical characteristics compared to

the simulated data used for training. This data mismatch can lead to the poor generalization of the model, resulting in suboptimal performance.

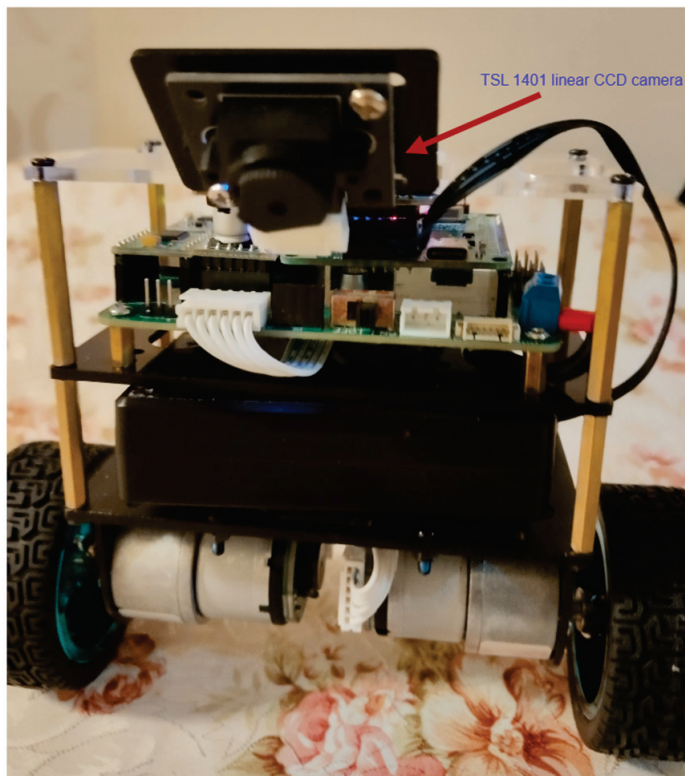


Figure 10. Cart body and CCD diagram.

Since we need to load the TensorFlow model data into the STM32 MCU [27], an installation step is necessary. We employ TensorFlow Lite [24], and its implementation steps are illustrated in Figure 11. The screenshots of several important software files during generation are shown in Figures 12 and 13. The software development environment, uVision 5 developed by Keil Corp. [25], is utilized to generate machine code, as depicted in Figure 14. Figures 11–13 illustrate a continuous process. This process primarily demonstrates how we utilize Python to convert an LSTM model into a .tflite header file. Subsequently, we transform this header file into a C-language module. This module can be integrated into our software project for joint compilation, generating machine code that is then flashed into the program memory of the hardware. The experimental results are presented in Figure 15. In practice, we observe that the image tracking performance of the two-wheeled self-balancing cart is excellent.

This approach still has the following limitations: One of the primary limitations is that this method is highly dependent on a critical factor, such as the quality and quantity of linear CCD data. When the data contains substantial noise or outliers, the performance of the method may deteriorate. Moreover, the proposed approach is validated under specific experimental conditions, for instance, in controlled indoor environments. Its effectiveness in more intricate real-world scenarios, including varying lighting conditions or high-speed motion, still needs to be further explored.

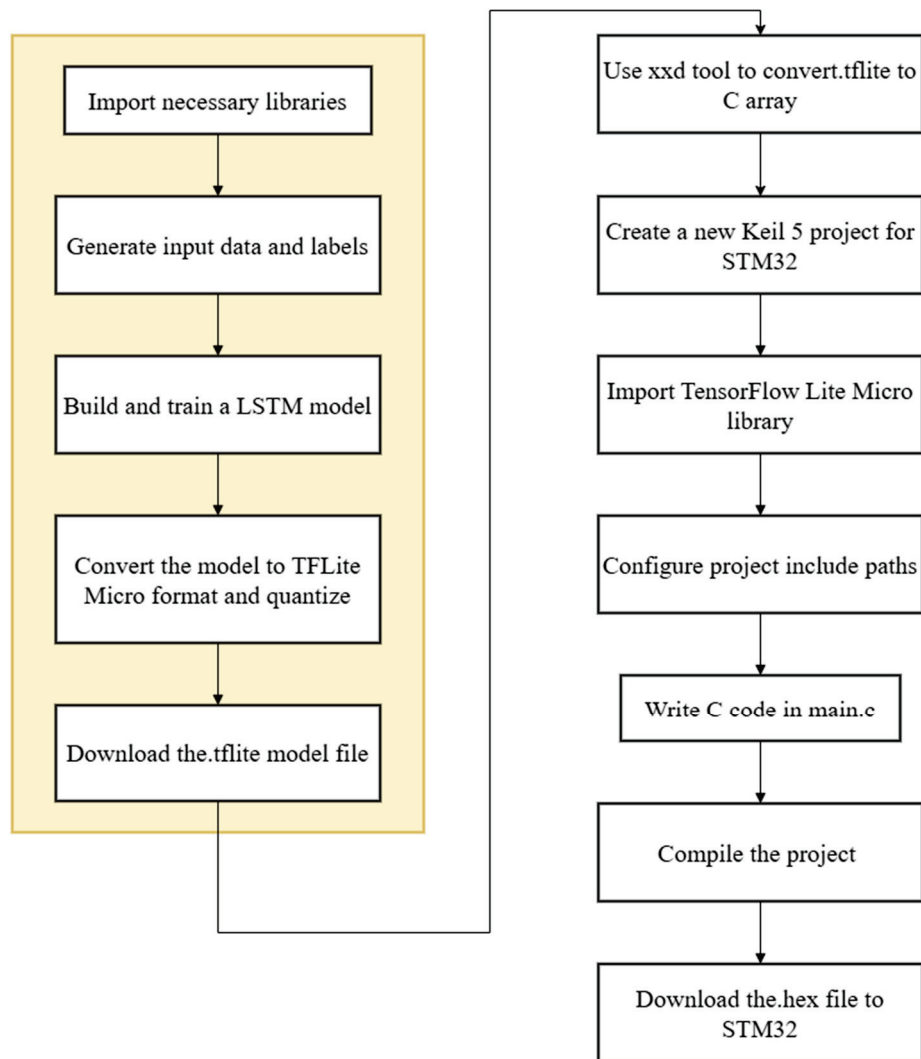


Figure 11. Installation steps diagram.

```

1 import subprocess
2
3 try:
4     # execution of transfer command
5     subprocess.run(['xxd', '-i', 'lstm_model.tflite'], stdout=open('lstm_model_data.h', 'w'), check=True)
6     print("generate lstm_model_data.h document successfully")
7 except subprocess.CalledProcessError as e:
8     print(f"execution error: {e}")
9 except FileNotFoundError:
10    print("no finding of lstm_model.tflite document, please check path")
  
```

generate lstm_model_data.h document successfully

Figure 12. Diagram of generation of head file.

```

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): TensorSpec(shape=(None, 10, 1), dtype=tf.float32, name='input_layer')
Output Type:
  TensorSpec(shape=(None, 3), dtype=tf.float32, name=None)
Captures:
  132357472245456: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472245072: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472252176: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472253328: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472254096: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472252368: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472244112: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472255248: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472249872: TensorSpec(shape=(), dtype=tf.resource, name=None)
  132357472255632: TensorSpec(shape=(), dtype=tf.resource, name=None)

```

The model of LSTM has been transferred to lstm_model.tflite successfully

Figure 13. Diagram of generation of tflite file.

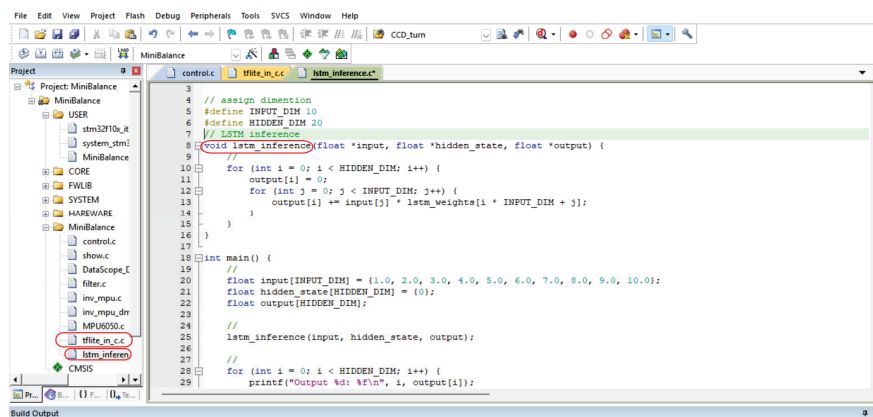


Figure 14. Diagram of generation of .hex file using Keil uVision 5.

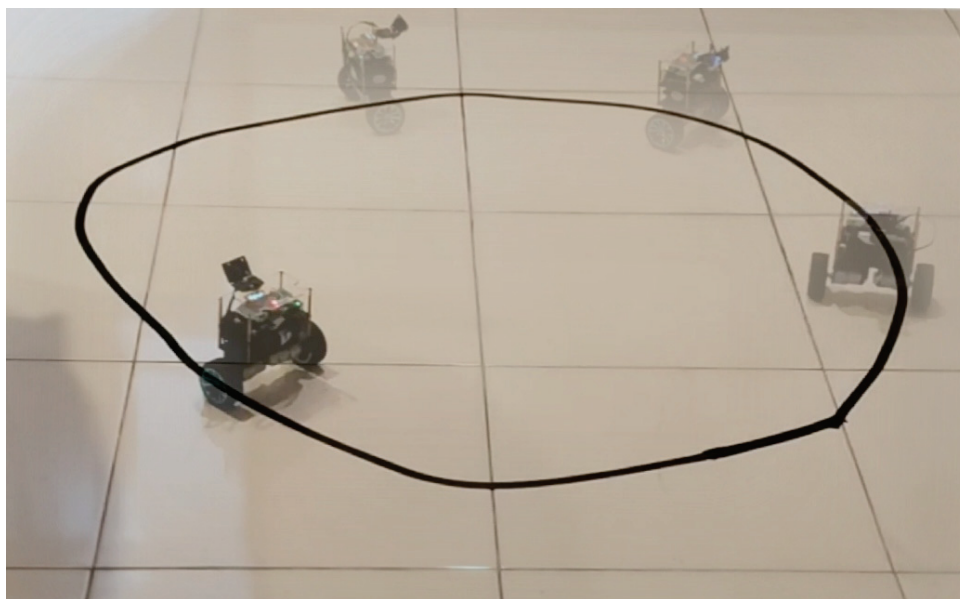


Figure 15. The overlay diagram of the experimental results.

5. Conclusions

This paper innovatively presents an application that fully harnesses the exceptional memory capabilities of Long Short-Term Memory (LSTM) neural networks for sequence signals, providing a new solution for the precise control of two-wheeled self-balancing carts. A linear array CCD lens is used to detect the black tape on the ground. After establishing a series of continuous training sample points, Python and TensorFlow are employed to train the LSTM network and optimize its weights. The efficacy of the training results is verified by simulating the symmetrical steering control of a two-wheeled cart. This process validates the proposed design method and showcases its remarkable performance. Experimental results demonstrate that this method is highly effective, characterized by robust image-tracking capabilities and optimal computational performance.

The proposed LSTM-based control strategy can be applied in various real-world scenarios. For example, in two-wheeled self-balancing carts used in industrial automation, logistics, or domestic services, it can be developed into an effective control strategy to enhance the safety, stability, and efficiency of all robots in these applications. This may include applications in other types of mobile robots, intelligent transportation systems, or wearable devices that require precise symmetrical motion control.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

Acknowledgments: The author is grateful to the editors and reviewers for their constructive comments, which have significantly improved this work.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Le, H.N.; Pham, M.K.; Pham, D.H.; Nguyen, T.V. Trajectory tracking and stabilization of two-wheeled balancing mobile robot with hierarchical and sliding mode control. *Int. J. Dynam. Control* **2025**, *13*, 10. [CrossRef]
2. Li, C.-H.G.; Zhou, L.-P.; Chao, Y.-H. Self-Balancing Two-Wheeled Robot Featuring Intelligent End-to-End Deep Visual-Steering. *IEEE ASME Trans. Mechatron.* **2021**, *26*, 2263–2273. [CrossRef]
3. Yan, C.; Li, X. Research on Stability Control System of Two-Wheel Heavy-Load Self-Balancing Vehicles in Complex Terrain. *Appl. Sci.* **2024**, *14*, 7682. [CrossRef]
4. Mon, Y.-J. Fuzzy PDC-Based LQR Sliding Neural Network Control for Two-Wheeled Self-Balancing Cart. *Electronics* **2025**, *14*, 1842. [CrossRef]
5. Lee, T.; Ju, D.; Lee, Y.S. Transition Control of a Double-Inverted Pendulum System Using Sim2Real Reinforcement Learning. *Machines* **2025**, *13*, 186. [CrossRef]
6. Mon, Y.-J. Tikhonov-Tuned Sliding Neural Network Decoupling Control for an Inverted Pendulum. *Electronics* **2023**, *12*, 4415. [CrossRef]
7. Sun, B.; Li, W.; Wang, Z.; Zhu, Y.; He, Q.; Guan, X.; Dai, G.; Yuan, D.; Li, A.; Cui, W.; et al. Recent Progress in Modeling and Control of Bio-Inspired Fish Robots. *J. Mar. Sci. Eng.* **2022**, *10*, 773. [CrossRef]
8. Chaudhary, A.; Bhushan, B. Self-Balancing and Position Control of a Balancer System Using a Pattern-Based Intelligent Optimization Method. *Int. J. Pattern Recognit. Artif. Intell.* **2023**, *37*, 2357011. [CrossRef]
9. Unluturk, A.; Aydogdu, O. Machine Learning Based Self-Balancing and Motion Control of the Underactuated Mobile Inverted Pendulum with Variable Load. *IEEE Access* **2022**, *10*, 104706–104718. [CrossRef]
10. Zhao, J.; Li, J.; Zhou, J. Research on Two-Round Self-Balancing Robot SLAM Based on the Gmapping Algorithm. *Sensors* **2023**, *23*, 2489. [CrossRef]
11. Liu, Q.; Zhou, F.; Hang, R.; Yuan, X. Bidirectional-Convolutional LSTM Based Spectral-Spatial Feature Learning for Hyper-spectral Image Classification. *Remote Sens.* **2017**, *9*, 1330. [CrossRef]
12. Zarzycki, K.; Ławryńczuk, M. LSTM and GRU Neural Networks as Models of Dynamical Processes Used in Predictive Control: A Comparison of Models Developed for Two Chemical Reactors. *Sensors* **2021**, *21*, 5625. [CrossRef] [PubMed]
13. Ycart, A.; Benetos, E. Learning and Evaluation Methodologies for Polyphonic Music Sequence Prediction with LSTMs. *IEEE ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1328–1341. [CrossRef]

14. Huo, Y.; Jin, M.; You, S. LSTM-Based Framework for the Synthesis of Original Soundtracks. *IEEE Access* **2024**, *12*, 33832–33842. [CrossRef]
15. Akilan, T.; Wu, Q.J.; Safaei, A.; Huo, J.; Yang, Y. A 3D CNN-LSTM-Based Image-to-Image Foreground Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 959–971. [CrossRef]
16. Vishwas, B.V.K.; Macharla, S.R. *Time Series Forecasting Using Generative AI: Leveraging AI for Precision Forecasting*; Apress: Berkeley, CA, USA, 2025. [CrossRef]
17. Liao, W.; Wang, S.; Yang, D.; Yang, Z.; Fang, J.; Rehtanz, C.; Porté-Agel, F. TimeGPT in Load Forecasting: A Large Time Series Model Perspective. *Appl. Energy* **2025**, *379*, 124973. [CrossRef]
18. Paroha, A.; Chotrani, A. A Comparative Analysis of TimeGPT and Time-LLM in Predicting ESP Maintenance Needs in the Oil and Gas Sector. *Int. J. Comput. Appl.* **2024**, *186*, 8887.
19. Zheng, L. Predictive Control of the Mobile Robot under the Deep Long-Short Term Memory Neural Network Model. *Comput. Intell. Neurosci.* **2022**, *2022*, 1835798. [CrossRef]
20. Zambella, G.; Caporale, D.; Grioli, G.; Pallottino, L.; Bicchi, A. Composite Whole-Body Control of Two-Wheeled Robots. *IEEE Trans. Robot.* **2025**, *41*, 2301–2321. [CrossRef]
21. Do, Q.T.; Tran, V.T.; Ngo, M.T.; Tran, M.Q.; Thiem, Q.L.; Nguyen, H.S.; Pham, B.K.; Phan, N.P.; Nguyen, D.H.; Nguyen, D.H.; et al. Modeling and Optimal Control for Two-Wheeled Self-Balancing Robot. *J. Fuzzy Syst. Control.* **2024**, *2*, 22–28. [CrossRef]
22. Daksh, M.S.; Mishra, P. A Feedback-Assisted Inverse Neural Network Controller for Cart-Mounted Inverted Pen-dulum. *Appl. Comput. Intell. Soft Comput.* **2025**, *2025*, 4873425.
23. Anaconda Inc. *Python Reference Manual*; Anaconda Inc.: Austin, TX, USA, 2025. Available online: <https://anaconda.com> (accessed on 2 May 2025).
24. Google Brain Teams. *Tensorflow Lite Reference Manual*; Google Brain Teams: Mountain View, CA, USA, 2025. Available online: <https://www.tensorflow.org/lite/microcontrollers?hl=zh-tw> (accessed on 2 May 2025).
25. Keil Electronics Co., Ltd. *uVision 5 Reference Manual*; Keil Electronics Co., Ltd.: Munich, Germany, 2025. Available online: <https://keil.com/> (accessed on 2 May 2025).
26. WheelTech Inc. *Two-Wheeled Self Balancing Cart Technical Development Manual*; WheelTech Inc.: Dongguan, China, 2025. Available online: <https://www.wheeltec.net> (accessed on 2 May 2025).
27. STMicroelectronics. *STM32F103xx Reference Manual*; STMicroelectronics: Geneva, Switzerland, 2025. Available online: https://www.st.com.cn/content/st_com/zh.html (accessed on 2 May 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

Fuzzy Double Yang Transform and Its Application to Fuzzy Parabolic Volterra Integro-Differential Equation

Atanaska Georgieva *, Slav I. Cholakov, Maria Vasileva and Yordanka Gudalova

Faculty of Mathematics and Informatics, University of Plovdiv, 4000 Plovdiv, Bulgaria;
cholakovs@uni-plovdiv.bg (S.I.C.); mariavasileva@uni-plovdiv.bg (M.V.); dgudalova@uni-plovdiv.bg (Y.G.)

* Correspondence: atanaska@uni-plovdiv.bg

Abstract: This article introduces a new fuzzy double integral transformation called fuzzy double Yang transformation. We review some of the main properties of the transformation and find the conditions for its existence. We prove the theorems for partial derivatives and fuzzy unitary convolution. All of the new results are applied to find an analytical solution to the fuzzy parabolic Volterra integro-differential equation (FPVIDE) with a suitably selected memory kernel. In addition, a numerical example is provided to illustrate how the proposed method might be helpful for solving FPVIDE utilizing symmetric triangular fuzzy numbers. Compared with other symmetric transforms, we conclude that our new approach is simpler and needs less calculations.

Keywords: fuzzy double Yang transform; fuzzy parabolic Volterra integro-differential equation; generalized Hukuhara partial differentiable

MSC: 35A22; 44A15; 44A35

1. Introduction

Partial differential equations are used for dynamic modeling of complex processes in various fields such as physics, chemistry, fluid and quantum mechanics, biology, and economics. They are predominantly applied to the so-called instantaneous phenomena, whose behavior depends on their momentary state. A large part of the processes require the model to account for their behavior over a previous time interval. As a result, it is necessary to use partial integro-differential equations, as they show the cumulative behavior of the process. Different types of partial differential equations are related to the various types of differential and integral operators. One of them is the parabolic Volterra integro-differential equation. It has important physical applications in modeling dynamical systems, where one can explore the effects of the “memory” of the system. Such systems are developed, for example, in the compression of viscoelastic media [1], nuclear reactor dynamics [2], expansion problems [3], reaction diffusion problems [4], and thermally conductive materials with functional memory [5].

Over the last few years, we have noticed an incredible interest in fuzzy mathematics due to the many applications in various fields, especially physics [6], artificial intelligence [7], medicine [8], engineering [9], and biology [10,11]. This trend explains the need for studying fuzzy differential equations. The solution to the first-order linear fuzzy differential equation was provided in [12], using different versions of the variation of constants formula. An generalized Euler approximation method was applied to solve a first-order linear fuzzy differential equation [13]. Osman et al. [14] investigated the fuzzy Adomian decomposition method and the fuzzy variational iteration method, and applied to solve fuzzy

heat-like and wave equations with variable coefficients, in the sense of gH-differentiability. In [15], the fuzzy Elzaki transform and the fuzzy Elzaki decomposition method were used to find solutions to fuzzy linear–nonlinear Schrodinger differential equations. In recent years, fuzzy double integral transformations have been used to find the exact solution to a linear fuzzy partial differential equation [16–18]. Analytical and numerical solutions to fuzzy integro-differential equations were obtained in [19–21].

Numerical solutions to the fuzzy partial Volterra integro-differential equation using the reproducing kernel Hilbert space method can be found in [22]. Recently, in order to find the exact solution to the linear fuzzy integro-differential equation, fuzzy integral transforms have been used. In [23], using the fuzzy Laplace transform, an analytical solution to FPVIDE under generalized partial Hukuhara differentiability was found. The fuzzy single and fuzzy double Sumudu transformation [24,25], as well as the fuzzy double Natural transformation [26], have been applied to the fuzzy partial Volterra integro-differential equation.

The Yang transform was introduced by Yang [27] and has been applied in the differential equation of the steady heat-transfer problem. Recently, Ullah et al. [28] proposed a fuzzy single Yang transform to find the solution to second-order fuzzy differential equations of an integer and fractional-order.

The main goal of the article is to extend the fuzzy single Yang transform to the fuzzy double Yang transform (FDYT). The fundamental properties and theorems of FDYT are presented and proven, and we compute the values of FDYT for some functions. New relations related to partial derivatives and the single convolution theorem are proven, which allows us to find the exact solution to an FPVIDE under generalized Hukuhara differentiability. More precisely, we look at the following nonhomogeneous FPVIDE with a symmetric memory kernel $k(t)$ in the infinite domain

$$g'_{t,gH}(x,t) \oplus \int_0^t k(t-s) \odot g(x,t) dx dt = \sigma \odot g''_{xx,gH}(x,t) \oplus f(x,t), \quad x \geq 0, t \geq 0, \quad (1)$$

where σ is any positive constant, $g(x,t)$ is the unknown fuzzy function, and $f(x,t)$ is a given fuzzy function. A simple formula for the solution to the Equation (1) is obtained and applied to solve a numerical example in order to display the efficiency of this new approach.

The remainder of this work is structured as follows:

In Section 2, we briefly present some basic concepts regarding fuzzy numbers and fuzzy calculus that will be used in the paper. Section 3 introduces the single fuzzy Yang transform (FYT), some fundamental concepts, and the basic properties of this transformation.

In Section 4, an FDYT for a fuzzy function is defined as well as some properties and theorems, and several relations regarding the existence, gH-partial derivatives, and single convolution are presented. An FPVIDE with the memory kernel is defined under generalized partial Hukuhara differentiability and a solution to this equation using the FDYT method is investigated in Section 5. Moreover, a numerical example is constructed to clarify the details and efficiency of the method in Section 6. Conclusions are given in Section 7.

2. Preliminaries

The following section consists of the necessary notations, definitions, and theorems which are useful in this research.

Let E^1 denote the set of fuzzy subsets of the real axis, i.e., $v : \mathbb{R} \rightarrow [0,1]$, which possesses the following properties:

- (i) ν is upper semi-continuous on $\mathbb{R}$ for all $\nu \in E^1$;
- (ii) ν is normal for all $\nu \in E^1$;
- (iii) ν is fuzzy convex for all $\nu \in E^1$;
- (iv) $cl\{\eta \in \mathbb{R} : \nu(\eta) > 0\}$ is compact, where cl denotes the closure of a subset.

Then, we say that E^1 is a space of fuzzy numbers. It is clear that any real number a can be interpreted as a fuzzy number $\tilde{a} = \chi(a)$; therefore, $\mathbb{R} \subset E^1$. The r -level set of the fuzzy number ν is denoted as

$$[\nu]^r = \begin{cases} \{\eta \in \mathbb{R} : \nu(\eta) \geq r\}, & 0 < r \leq 1, \\ cl\{\eta \in \mathbb{R} : \nu(\eta) > 0\}, & r = 0. \end{cases}$$

Then, from (i) to (iv), it follows that for each $0 \leq r \leq 1$, the r -level sets of fuzzy number ν are nonempty closed intervals of the form

$$[\nu]^r = [\underline{\nu}(r), \bar{\nu}(r)].$$

A triangular fuzzy number ν is defined as an ordered triple $\nu = (\nu_1, \nu_2, \nu_3)$, where $\nu_1 \leq \nu_2 \leq \nu_3$ has r -cuts

$$[\nu]^r = [\nu_1 + (\nu_2 - \nu_1)r, \nu_3 - (\nu_3 - \nu_2)r], \quad 0 \leq r \leq 1.$$

Let μ and ν be two fuzzy numbers and $k \in \mathbb{R}$. Then, the addition $\mu \oplus \nu \in E^1$ and the scalar multiplication $k \odot \mu \in E^1$ are defined as having level cuts

$$[\mu \oplus \nu]^r = [\mu]^r + [\nu]^r = \{\xi + \eta : \xi \in [\mu]^r, \eta \in [\nu]^r\}$$

$$[k \odot \mu]^r = k \cdot [\mu]^r = \{k\xi : \xi \in [\mu]^r\}, \quad [0]^r = \{0\} \text{ for all } 0 \leq r \leq 1.$$

Denote $\mathbb{R}_+ = [0, +\infty)$.

Definition 1 ([29]). The Hausdorff distance between fuzzy numbers is given by $D : E^1 \times E^1 \rightarrow \mathbb{R}_+$ as

$$D(\mu, \nu) = \sup_{0 \leq r \leq 1} \max\{|\underline{\mu}(r) - \underline{\nu}(r)|, |\bar{\mu}(r) - \bar{\nu}(r)|\},$$

where $[\mu]^r = [\underline{\mu}(r), \bar{\mu}(r)]$ and $[\nu]^r = [\underline{\nu}(r), \bar{\nu}(r)]$.

The metric space (E^1, D) is complete separable and locally compact, and the following properties of the metric D are well known:

- (i) $D(\lambda \oplus \nu, \mu \oplus \nu) = D(\lambda, \mu)$ for all $\lambda, \mu, \nu \in E^1$;
- (ii) $D(k \odot \mu, k \odot \nu) = |k|D(\mu, \nu)$ for all $\mu, \nu \in E^1$ and $k \in \mathbb{R}$;
- (iii) $D(\lambda \oplus \mu, \nu \oplus \kappa) \leq D(\lambda, \nu) + D(\mu, \kappa)$ for all $\lambda, \mu, \nu, \kappa \in E^1$.

Definition 2 ([29]). Let $\mu, \nu \in E^1$. If there exists a fuzzy number λ , such that $\mu = \nu \oplus \lambda$, then λ is called the Hukuhara difference (H-difference) of μ and ν , and it is denoted by $\mu \ominus_H \nu$.

The r -cuts of H-difference are

$$[\mu \ominus_H \nu]^r = [\underline{\mu}(r) - \underline{\nu}(r), \bar{\mu}(r) - \bar{\nu}(r)],$$

where $[\mu]^r = [\underline{\mu}(r), \bar{\mu}(r)]$ and $[\nu]^r = [\underline{\nu}(r), \bar{\nu}(r)]$.

Clearly, $\mu \ominus_H \mu = \{\tilde{0}\}$; if $\mu \ominus_H \nu$ exists, it is unique.

Definition 3 ([29]). Given $\mu, \nu \in E^1$, the generalized Hukuhara difference (gH-difference) is the fuzzy quantity $\lambda \in E^1$, if it exists, such that

$$\mu \ominus_{gH} \nu = \lambda \Leftrightarrow \begin{cases} (i) & \mu = \nu \oplus \lambda, \\ \text{or } (ii) & \nu = \mu \oplus (-1) \odot \lambda. \end{cases} \quad (2)$$

It is easy to show that (i) and (ii) are valid if and only if λ is a crisp number. In terms of r -cuts, we have

$$[\mu \ominus_{gH} \nu]^r = [\min\{\underline{\mu}(r) - \underline{\nu}(r), \bar{\mu}(r) - \bar{\nu}(r)\}, \max\{\underline{\mu}(r) - \underline{\nu}(r), \bar{\mu}(r) - \bar{\nu}(r)\}]$$

and if the H-difference exists, then $\mu \ominus_H \nu = \mu \ominus_{gH} \nu$. The conditions for the existence of $\mu \ominus_{gH} \nu = w \in E^1$ are given in [30].

Proposition 1 ([29]). Let $\mu, \nu \in E^1$, then

$$D(\mu \ominus_{gH} \nu, \tilde{0}) = D(\mu, \nu).$$

Proposition 2 ([30]). Let $\mu, \nu \in E^1$. If $\mu \ominus_{gH} \nu$ exists, it is unique and has the following properties

- (i) $\mu \ominus_{gH} \mu = \tilde{0}$;
- (ii) $(\mu \oplus \nu) \ominus_{gH} \nu = \mu$, $\mu \ominus_{gH} (\mu \ominus_H \nu) = \nu$;
- (iii) if $\mu \ominus_{gH} \nu$ exists, then $(-\nu) \ominus_{gH} (-\mu)$ also does and $\tilde{0} \ominus_{gH} (\mu \ominus_{gH} \nu) = (-\nu) \ominus_{gH} (-\mu)$;
- (iv) $\mu \ominus_{gH} \nu = \nu \ominus_{gH} \mu = \lambda$ if and only if $\lambda = -\lambda$; furthermore, $\lambda = \tilde{0}$ if and only if $\mu = \nu$;
- (v) If $\nu \ominus_{gH} \mu$ exists, then either $\mu \oplus (\nu \ominus_{gH} \mu) = \mu$ or $\nu \ominus_H (\nu \ominus_{gH} \mu) = \mu$, and if both equalities hold then $\nu \ominus_{gH} \mu$ is a crisp set.

2.1. The One-Variable Fuzzy Calculus

In this section, we present basic definitions and theorems for a fuzzy-valued function of one-variable that will be used throughout the paper.

A function $g : [c, d] \subset \mathbb{R} \rightarrow E^1$ is called a fuzzy-valued function. The r -level representation of this fuzzy function g is given by $g(t, r) = [\underline{g}(t, r), \bar{g}(t, r)]$, $t \in [c, d]$ for all $0 \leq r \leq 1$.

Definition 4 ([31]). We say that fuzzy-valued function $g : [c, d] \rightarrow E^1$ is continuous at $t_0 \in [c, d]$, if

$$\lim_{t \rightarrow t_0} D(g(t), g(t_0)) = 0$$

provided that limits exist.

The function g is fuzzy continuous on $[c, d]$ if g is continuous in each $t_0 \in [c, d]$.

Definition 5 ([30]). Let $t_0 \in (c, d)$ and k be such that $t_0 + k \in (c, d)$. Then, the generalized Hukuhara derivative (gH-derivative) of a function $g : (c, d) \rightarrow E^1$ at t_0 is called the fuzzy number $g'_{gH}(t_0)$, which is defined as

$$g'_{gH}(t_0) = \lim_{k \rightarrow 0} \frac{1}{k} [g(t_0 + k) \ominus_{gH} g(t_0)], \quad (3)$$

if the limit exists.

Theorem 1. Let $g : (c, d) \rightarrow E^1$ be gH-differentiable at $t_0 \in [c, d]$. Then, g is fuzzy continuous at t_0 .

Proof. Using properties of distance D , along with gH-differentiability of g and Proposition 1, we have

$$\begin{aligned}\lim_{t \rightarrow t_0} D(g(t), g(t_0)) &= \lim_{t \rightarrow t_0} D(g(t) \ominus_{gH} g(t_0), \tilde{0}) \\ &= \lim_{t \rightarrow t_0} D\left(\frac{g(t) \ominus_{gH} g(t_0)}{t - t_0} (t - t_0), \tilde{0}\right) \\ &= |t - t_0| \lim_{t \rightarrow t_0} D\left(\frac{g(t) \ominus_{gH} g(t_0)}{t - t_0}, \tilde{0}\right) = 0 \cdot D(g'_{gH}(t_0), \tilde{0}) = 0.\end{aligned}$$

□

The next theorem gives the expression of the fuzzy gH-derivative in terms of the derivatives of the endpoints of the level sets.

Theorem 2. Let $g : [c, d] \rightarrow E^1$ be a fuzzy-valued function with r -levels $g(t, r) = [\underline{g}(t, r), \bar{g}(t, r)]$ and the real-valued functions $\underline{g}(\cdot, r)$ and $\bar{g}(\cdot, r)$ be differentiable at t_0 for all $0 \leq r \leq 1$. Then, the function $g(t)$ is gH-differentiable at $t_0 \in (c, d)$, if and only if one of the following two cases holds:

- (i) $\underline{g}(t_0, r)$ is increasing, $\bar{g}(t_0, r)$ is decreasing, and $\underline{g}(t_0, r) \leq \bar{g}(t_0, r)$;
- (ii) $\underline{g}(t_0, r)$ is decreasing, $\bar{g}(t_0, r)$ is increasing, and $\bar{g}(t_0, r) \leq \underline{g}(t_0, r)$.

Moreover, we have

$$g'_{gH}(t_0, r) = [\min\{\underline{g}(t_0, r), \bar{g}(t_0, r)\}, \max\{\underline{g}(t_0, r), \bar{g}(t_0, r)\}]$$

for all $0 \leq r \leq 1$.

Proof. See Theorem 24 in [30]. □

If $\underline{g}(\cdot, r)$ and $\bar{g}(\cdot, r)$ are both differentiable, according to Theorem 2, for the definition of gH-differentiability, we distinguish two cases corresponding to (i) and (ii) of Equation (3).

Definition 6 ([30]). Let $g : [c, d] \rightarrow E^1$ and $t_0 \in (c, d)$, with $\underline{g}(\cdot, r)$ and $\bar{g}(\cdot, r)$ both be differentiable at t_0 . The fuzzy-valued function g is called:

1. (i)-gH-differentiable at t_0 if

$$g'_{gH}(t_0, r) = [\underline{g}'(t_0, r), \bar{g}'(t_0, r)] \text{ for all } r \in [0, 1]; \quad (4)$$

2. (ii)-gH-differentiable at t_0 if

$$g'_{gH}(t_0, r) = [\bar{g}'(t_0, r), \underline{g}'(t_0, r)] \text{ for all } r \in [0, 1]. \quad (5)$$

Theorem 3. Let $f, g : (c, d) \rightarrow E^1$ be gH-differentiable. Then, $f(t) \oplus g(t)$ is gH-differentiable and

$$(f \oplus g)'_{gH}(t) = f'_{gH}(t) \oplus g'_{gH}(t).$$

Proof. Suppose that f and g are both (i)-gH-differentiable. Then, for every $r \in [0, 1]$ we have,

$$f'_{gH}(t_0, r) = [\underline{f}'(t_0, r), \bar{f}'(t_0, r)]$$

and

$$g'_{gH}(t_0, r) = [\underline{g}'(t_0, r), \bar{g}'(t_0, r)].$$

Then

$$\begin{aligned} f'_{gH}(t_0, r) \oplus g'_{gH}(t_0, r) &= [\underline{f}'(t_0, r), \bar{f}'(t_0, r)] + [\underline{g}'(t_0, r), \bar{g}'(t_0, r)] \\ &= [\underline{f}'(t_0, r) + \underline{g}'(t_0, r), \bar{f}'(t_0, r) + \bar{g}'(t_0, r)] \\ &= (f \oplus g)'_{gH}(t_0, r). \end{aligned}$$

□

Theorem 4. Let $f : [c, d] \rightarrow E^1$ and $g : [c, d] \rightarrow \mathbb{R}$ be two differentiable functions. Then,

$$\int_c^d g(t) \odot f'_{gH}(t) dt = (g(d) \odot f(d)) \ominus_{gH} (g(c) \odot f(c)) \ominus_{gH} \int_c^d g'(t) \odot f(t) dt.$$

Proof. See [29]. □

Theorem 5. Let $g : \mathbb{R} \rightarrow E^1$ be a fuzzy-valued function with r -levels $g(t, r) = [\underline{g}(t, r), \bar{g}(t, r)]$. Suppose that the functions $\underline{g}(t, r)$ and $\bar{g}(t, r)$ are Riemann integrable on $\mathbb{R}$ for all $0 \leq r \leq 1$. Then, $g(t)$ is improper fuzzy Riemann-integrable in $\mathbb{R}$. Moreover, we have

$$\left[\int_{-\infty}^{\infty} g(t) dt \right]^r = \left[\int_{-\infty}^{\infty} \underline{g}(t, r) dt, \int_{-\infty}^{\infty} \bar{g}(t, r) dt \right]$$

for all $0 \leq r \leq 1$.

Proof. See [29]. □

2.2. The Two-Variable Fuzzy Calculus

Let $g : Q \subset \mathbb{R} \times \mathbb{R} \rightarrow E^1$ be a fuzzy-valued function of two variables with r -levels $g(x, t, r) = [\underline{g}(x, t, r), \bar{g}(x, t, r)]$ for all $(x, t) \in Q$ and $0 \leq r \leq 1$.

Definition 7 ([32]). For $(x_0, t_0) \in Q$, let the constants h and k be such that $(x_0 + h, t_0) \in Q$ and $(x_0, t_0 + k) \in Q$. Then, the first generalized Hukuhara partial derivatives (gH - p -derivative) of a fuzzy-valued function $g : Q \rightarrow E^1$ at $(x_0, t_0) \in Q$ with respect to x and t are called the fuzzy numbers $g'_x(x_0, t_0)$ and $g'_t(x_0, t_0)$, and are defined as

$$g'_{x,gH}(x_0, t_0) = \lim_{h \rightarrow 0} \frac{1}{h} [g(x_0 + h, t_0) \ominus_{gH} g(x_0, t_0)],$$

$$g'_{t,gH}(x_0, t_0) = \lim_{k \rightarrow 0} \frac{1}{k} [g(x_0, t_0 + k) \ominus_{gH} g(x_0, t_0)].$$

Definition 8 ([32]). Let $g : Q \rightarrow E^1$ be fuzzy-valued function, and $(x_0, t_0) \in Q$. Suppose that the functions $\underline{g}(x, t, r)$ and $\bar{g}(x, t, r)$ are partially differentiable in (x_0, t_0) with respect to variable t . Also, we say that the function $g(x, t)$ is:

1. (i)- p - gH -differentiable at (x_0, t_0) with respect to t if

$$g'_{t,gH}(x_0, t_0, r) = [\underline{g}'_t(x_0, t_0, r), \bar{g}'_t(x_0, t_0, r)] \text{ for all } r \in [0, 1], \quad (6)$$

2. (ii)- p - gH -differentiable at (x_0, y_0) with respect to variable x if

$$g'_{x,gH}(x_0, t_0, r) = [\underline{g}'_x(x_0, t_0, r), \bar{g}'_x(x_0, t_0, r)] \text{ for all } r \in [0, 1]. \quad (7)$$

Theorem 6. Let $g : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow E^1$ be a fuzzy-valued function. Assume that $\int_0^\infty g(x, t)dt$ is convergent for each $\mathbb{R}_+$ and $\int_0^\infty g(x, t)dx$, as function t is convergent on $\mathbb{R}_+$. Then

$$\int_0^\infty \int_0^\infty g(x, t)dt dx = \int_0^\infty \int_0^\infty g(x, t)dx dt.$$

Proof. See [33]. $\square$

3. Fuzzy Yang Transform

In this section, we present the definition and basic properties of FYT [28].

Definition 9. The FYT for a fuzzy function $g(t)$ is defined as

$$Y_t[g(t)] = G(\beta) = \int_0^\infty e^{-\frac{t}{\beta}} \odot g(t)dt, \quad (8)$$

provided that the improper fuzzy integral exists and where t and β are transform variables.

Definition 10. The inverse fuzzy Yang transform is given by

$$Y_\beta^{-1}[G(\beta)] = g(t) = \frac{1}{2\pi i} \int_{b-i\infty}^{b+i\infty} e^{\frac{t}{\beta}} \odot G(\beta)d\beta, \quad (9)$$

where the function $G(\beta)$ is analytic for all β , such that $\operatorname{Re}\beta > b$.

Definition 11. A fuzzy-valued function $g : \mathbb{R}_+ \rightarrow E^1$ is said to be of exponential order e^{dt} as $t \rightarrow \infty$, if there exists a positive constant L , such that for all $t > T$, we have

$$D(g(t), \tilde{0}) \leq Le^{dt}.$$

Theorem 7. If $g(t)$ is a continuous fuzzy function in every finite interval $0 \leq t \leq T$ and $g(t)$ is of exponential order e^{dt} .

Then, the FYT of $g(t)$ exists for all β , such that $\operatorname{Re}(\frac{1}{\beta}) > d$.

Proof. Using Definition 9, we obtain

$$D(Y_t[g(t)], \tilde{0}) = D(G(\beta), \tilde{0}) = D\left(\int_0^\infty e^{-\frac{t}{\beta}} \odot g(t)dt, \tilde{0}\right).$$

Using the property of the improper fuzzy integral, we obtain

$$\begin{aligned} D(G(\beta), \tilde{0}) &= D\left(\int_0^\infty e^{-\frac{t}{\beta}} \odot g(t)dt, \tilde{0}\right) \leq \int_0^\infty e^{-\frac{t}{\beta}} D(g(t), \tilde{0})dt \\ &\leq L \int_0^\infty e^{-(\frac{1}{\beta}-d)t} dt = \frac{L\alpha}{1-d\beta}. \end{aligned}$$

Thus, the improper fuzzy integral converges for all $\operatorname{Re}(\frac{1}{\beta}) > d$, and $Y_t[g(t)]$ exists. $\square$

The classical Yang transform is applied to some special functions in [27].

- (i) $Y_t[1] = \beta$;
- (ii) $Y_t[t^n] = \begin{cases} \beta^{n+1}n! & n = 1, 2, 3, \dots \\ \beta^{n+1}\Gamma(n+1) & n > 0 \end{cases}$
- (iii) $Y_t[e^{dt}] = \frac{\beta}{1-d\beta}$ for all $d \in \mathbb{R}$;
- (iv) $Y_t[\sin dt] = \frac{d\beta}{1+d^2\beta^2}$ for all $d \in \mathbb{R}$;
- (v) $Y_t[\cos dt] = \frac{\beta^2}{1+d^2\beta^2}$ for all $d \in \mathbb{R}$;
- (vi) $Y_t[\sinh dt] = \frac{d\beta^2}{1-d^2\beta^2}$ for all $d \in \mathbb{R}$;
- (vii) $Y_t[\cosh dt] = \frac{\beta}{1-d^2\beta^2}$ for all $d \in \mathbb{R}$.

We will give some of the basic properties of FYT.

Theorem 8 (Linearity). If $G_1(\beta) = Y_t[g_1(t)]$ and $G_2(\beta) = Y_t[g_2(t)]$. Then

$$Y_t[b_1 \odot g_1(t) \oplus b_2 \odot g_2(t)] = b_1 \odot Y_t[g_1(t)] \oplus b_2 \odot Y_t[g_2(t)],$$

where $b_1, b_2 \in \mathbb{R}$, such that $b_1, b_2 \geq 0$ or $b_1, b_2 \leq 0$.

Proof. Using Definition 9 and the property of the improper fuzzy integral, we obtain

$$\begin{aligned} Y_t[b_1 \odot g_1(t) \oplus b_2 \odot g_2(t)] &= \int_0^\infty e^{-\frac{t}{\beta}} (b_1 \odot g_1(t) \oplus b_2 \odot g_2(t)) dt \\ &= \int_0^\infty b_1 e^{-\frac{t}{\beta}} \odot g_1(t) dt \oplus \int_0^\infty b_2 e^{-\frac{t}{\beta}} \odot g_2(t) dt \\ &= b_1 \odot Y_t[g_1(t)] \oplus b_2 \odot Y_t[g_2(t)]. \end{aligned}$$

□

Remark 1. Using the Definition 10, we can show that Y_t^{-1} a linear transformation, i.e.,

$$Y_t^{-1}[b_1 \odot G_1(\beta) \oplus b_2 \odot G_2(\beta)] = b_1 \odot Y_t^{-1}[G_1(\beta)] \oplus b_2 \odot Y_t^{-1}[G_2(\beta)]$$

Theorem 9 (Change of Scale). If $G(\beta) = Y_t[g(t)]$, then for some constant b , it follows

$$Y_t[g(bt)] = \frac{1}{b} \odot G\left(\frac{1}{b}\beta\right).$$

Proof. Using Definition 9, we have

$$Y_t[g(bt)] = \int_0^\infty e^{-\frac{t}{\beta}} \odot g(bt) dt.$$

Put $bt = t_1$ and $dt = \frac{1}{b}dt_1$ in the above equation, we have

$$Y_t[g(bt)] = \int_0^\infty \frac{1}{b} e^{-\frac{t_1}{b\beta}} \odot g(t_1) d(t_1) = \frac{1}{b} \odot G\left(\frac{1}{b}\beta\right).$$

□

Theorem 10 (Duality). If $Y_t[g(t)] = G(\beta)$ is FYT and $L_t[g(t)] = F(\beta)$ is the fuzzy Laplace transform of $g(t)$, then

$$G(\beta) = F\left(\frac{1}{\beta}\right) \text{ and } F(\beta) = G\left(\frac{1}{\beta}\right)$$

Proof. Using Definition 9, we have

$$G(\beta) = Y_t[g(t)] = \int_0^\infty e^{-\frac{t}{\beta}} \odot g(t) dt = \int_0^\infty e^{-\frac{1}{\beta}t} \odot g(t) dt = F\left(\frac{1}{\beta}\right).$$

and

$$F(\beta) = \int_0^\infty e^{-\beta t} \odot g(t) dt = \int_0^\infty e^{-\frac{t}{\frac{1}{\beta}}} \odot g(t) dt = G\left(\frac{1}{\beta}\right).$$

□

Theorem 11. Let us consider

- (i) $g(t)$ is a continuous fuzzy function for all $t \geq 0$;
- (ii) $g(t)$ is of exponential order e^{dt} , i.e.,

$$D(g(t), \tilde{0}) \leq Le^{dt}, \quad t \in [0, T], \quad L > 0;$$

- (iii) $g'_{gH}(t)$ is continuous in every finite closed interval $0 \leq t \leq T$.

Then,

1. $Y_t[g'_{gH}(t)] = (-1) \odot g(0) \ominus_{gH} (-1) \frac{1}{\beta} \odot Y_t[g(t)];$
2. $Y_t[g''_{gH}(t)] = (-1) \odot g'_{gH}(0) \ominus_{gH} \left(\frac{1}{\beta} \odot g(0) \ominus_{gH} \frac{1}{\beta^2} \odot Y_t[g(t)] \right),$

for all $\operatorname{Re}(\frac{1}{\beta}) > d$.

Proof. We prove case 1. Using definition of an improper fuzzy integral and Theorem 4, we obtain

$$\begin{aligned} Y_t[g'_{gH}(t)] &= \int_0^\infty e^{-\frac{t}{\beta}} \odot g'_{gH}(t) dt = \lim_{\xi \rightarrow \infty} \int_0^\xi e^{-\frac{t}{\beta}} \odot g'_{gH}(t) dt \\ &= \lim_{\xi \rightarrow \infty} (e^{-\frac{\xi}{\beta}} \odot g(\xi)) \ominus_{gH} (-1) \frac{1}{\beta} \lim_{\xi \rightarrow \infty} \int_0^\xi e^{-\frac{t}{\beta}} \odot g(t) dt \\ &= \lim_{\xi \rightarrow \infty} (e^{-\frac{\xi}{\beta}} \odot g(\xi)) \ominus_{gH} g(0) \ominus_{gH} (-1) \frac{1}{\beta} \int_0^\infty e^{-\frac{t}{\beta}} \odot g(t) dt. \end{aligned}$$

From condition (ii), we obtain

$$\lim_{\xi \rightarrow \infty} D(e^{-\frac{\xi}{\beta}} \odot g(\xi), \tilde{0}) = \lim_{\xi \rightarrow \infty} e^{-\frac{\xi}{\beta}} D(g(\xi), \tilde{0}) \leq \lim_{\xi \rightarrow \infty} Le^{-(\frac{1}{\beta}-d)\xi} = 0. \quad (10)$$

Hence, using Proposition 1 and Equation (10), we have

$$Y_t[g'_{gH}(t)] = (-1) \odot g(0) \ominus_{gH} (-1) \frac{1}{\beta} \odot Y_t[g(t)]. \quad (11)$$

Similarly, from Equation (11) and Definition 9, we obtain

$$\begin{aligned} Y_t[g''_{gH}(t)] &= (-1) \odot g'_{gH}(0) \ominus_{gH} (-1) \frac{1}{\beta} \odot Y_t[g'_{gH}(t)] \\ &= (-1) \odot g'_{gH}(0) \ominus_{gH} (-1) \frac{1}{\beta} \odot \left((-1) \odot g(0) \ominus_{gH} (-1) \frac{1}{\beta} \odot Y_t[g(t)] \right) \\ &= (-1) \odot g'_{gH}(0) \ominus_{gH} \left(\frac{1}{\beta} \odot g(0) \ominus_{gH} \frac{1}{\beta^2} \odot Y_t[g(t)] \right). \end{aligned}$$

□

Corollary 1. Let $g(x, t)$ be a fuzzy function of two variables. Then, we have

- (i) $Y_t[g'_{t,gH}(x, t)] = (-1) \odot g(x, 0) \ominus_{gH} (-1)^{\frac{1}{\beta}} \odot Y_t[g(x, t)];$
- (ii) $Y_t[g''_{tt,gH}(x, t)] = (-1) \odot g'_{t,gH}(x, 0) \ominus_{gH} \left(\frac{1}{\beta} \odot g(x, 0) \ominus_{gH} \frac{1}{\beta^2} \odot Y_t[g(x, t)] \right).$

4. Fuzzy Double Yang Transform

In the following section, we introduce FDYT, that is, two fuzzy Yang transforms of order one. We give the fundamental properties and theorems related to the existence and fuzzy partial derivatives. Moreover, the fuzzy single convolution theorem is illustrated.

Definition 12. The FDYT of a fuzzy function $g : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow E^1$ is defined by

$$G(\alpha, \beta) = Y_2[g(x, t)] = Y_x[Y_t[g(x, t)]] = \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} \odot g(x, t) dt dx, \quad (12)$$

provided that the improper fuzzy double integral exists. Here, α and β are complex numbers.

Definition 13. The inverse FDYT is given by

$$Y_2^{-1}[G(\alpha, \beta)] = g(x, t) = \frac{1}{2\pi i} \int_{a-i\infty}^{a+i\infty} \int_{b-i\infty}^{b+i\infty} e^{\frac{x}{\alpha}} e^{\frac{t}{\beta}} \odot G(\alpha, \beta) d\alpha d\beta, \quad (13)$$

where the function $G(\alpha, \beta)$ is analytic for all α and β , such that $\operatorname{Re} \alpha > a$ and $\operatorname{Re} \beta > b$.

Definition 14. A fuzzy-valued function $g : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow E^1$ is said to be of exponential order e^{cx+dt} as $x \rightarrow \infty$ and $t \rightarrow \infty$, if there exists a positive constant L , such that for all $x > X$ and $t > T$, we have

$$D(g(x, t), \tilde{0}) \leq L e^{cx+dt}.$$

Theorem 12. Let $g(x, t)$ be a continuous fuzzy function in $(0, X) \times (0, T)$ and $g(x, t)$ be of exponential order e^{cx+dt} .

Then, the FDYT of the function $g(x, t)$ exists for all α and β such that $\operatorname{Re}(\frac{1}{\alpha}) > c$ and $\operatorname{Re}(\frac{1}{\beta}) > d$.

Proof. Using Definition 12 and the property of improper fuzzy double integral, we obtain

$$\begin{aligned} D(Y_2[g(x, t)], \tilde{0}) &= D(G(\alpha, \beta), \tilde{0}) = D\left(\int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} \odot g(x, t) dx dt, \tilde{0}\right) \\ &\leq \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} D(g(x, t), \tilde{0}) dx dt \\ &\leq L \int_0^\infty \int_0^\infty e^{-(\frac{1}{\alpha}-c)x} e^{-(\frac{1}{\beta}-d)t} dx dt = \frac{L\alpha\beta}{(1-c\alpha)(1-d\beta)}. \end{aligned}$$

Thus, the improper fuzzy double integral converges for all $\operatorname{Re}(\frac{1}{\alpha}) > c$ and $\operatorname{Re}(\frac{1}{\beta}) > d$, and $Y_2[g(x, t)]$ exist. $\square$

Double Yang transform of some important functions.

- (i) $Y_2[1] = \alpha\beta;$
- (ii) $Y_2[x^m t^n] = \begin{cases} \alpha^{m+1} \beta^{n+1} m! n! & m, n = 1, 2, 3, \dots \\ \alpha^{m+1} \beta^{n+1} \Gamma(m+1) \Gamma(n+1) & m > 0, n > 0 \end{cases}$
- (iii) $Y_2[e^{cx+dt}] = \frac{\alpha\beta}{(1-c\alpha)(1-d\beta)}$ for all $c, d \in \mathbb{R};$

- (iv) $Y_2[\sin(cx + dt)] = \frac{\alpha\beta}{(1+c^2\alpha^2)(1+d^2\beta^2)}$ for all $c, d \in \mathbb{R}$;
 (v) $Y_2[\cos(cx + dt)] = \frac{cd\alpha^2\beta^2}{(1+c^2\alpha^2)(1+d^2\beta^2)}$ for all $c, d \in \mathbb{R}$.

Now, we present some properties for FDYT.

Remark 2. According to Theorem 8, we can prove that if $g_1(x, t)$ and $g_2(x, t)$ are fuzzy functions, then

$$Y_2[\gamma_1 \odot g_1(x, t) \oplus \gamma_2 \odot g_2(x, t)] = \gamma_1 \odot Y_2[g_1(x, t)] \oplus \gamma_2 \odot Y_2[g_2(x, t)],$$

where $\gamma_1, \gamma_2 \in \mathbb{R}$, such that $\gamma_1, \gamma_2 \geq 0$ or $\gamma_1, \gamma_2 \leq 0$.

Theorem 13 (Shifting). Let c and d be any constants, and $g(x, t)$ be a continuous fuzzy function of two variables x and t . Then,

$$Y_2[e^{-(cx+dt)} \odot g(x, t)] = G\left(\frac{\alpha}{1+c\alpha}, \frac{\beta}{1+b\beta}\right). \quad (14)$$

Proof. Using Definition 12, we have

$$\begin{aligned} Y_2[e^{-(cx+dt)} \odot g(x, t)] &= \int_0^\infty \int_0^\infty e^{-(cx+dt)} e^{-\left(\frac{x}{\alpha} - \frac{t}{\beta}\right)} \odot g(x, t) dx dt \\ &= \int_0^\infty \int_0^\infty e^{-x\left(\frac{1}{\alpha} + c\right) - t\left(\frac{1}{\beta} + d\right)} \odot g(x, t) dx dt \\ &= \int_0^\infty \int_0^\infty e^{-x\left(\frac{1+c\alpha}{\alpha}\right) - t\left(\frac{1+b\beta}{\beta}\right)} \odot g(x, t) dx dt \\ &= G\left(\frac{\alpha}{1+c\alpha}, \frac{\beta}{1+b\beta}\right). \end{aligned}$$

□

Theorem 14 (Heaviside Function). Let $g(x, t)$ be a continuous fuzzy function and

$$H(x - \delta, t - \varepsilon) = \begin{cases} 1, & x > \delta, t > \varepsilon \\ 0, & x < \delta, t < \varepsilon, \end{cases}$$

where $H(x - \delta, t - \varepsilon)$ is the Heaviside function and $\delta, \varepsilon \in \mathbb{R}$. If $Y_2[g(x, t)] = G(\alpha, \beta)$, then

$$Y_2[H(x - \delta, t - \varepsilon) \odot g(x - \delta, t - \varepsilon)] = e^{-\frac{\delta}{\alpha} - \frac{\varepsilon}{\beta}} \odot G(\alpha, \beta).$$

Proof. Using Definition 12, we find

$$\begin{aligned} Y_2[H(x - \delta, t - \varepsilon) \odot g(x - \delta, t - \varepsilon)] &= \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} H(x - \delta, t - \varepsilon) \odot g(x - \delta, t - \varepsilon) dx dt \\ &= \int_\varepsilon^\infty \int_\delta^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} \odot g(x - \delta, t - \varepsilon) dx dt. \end{aligned}$$

We make a change of variable

$$v = x - \delta, \quad \mu = t - \varepsilon.$$

Then

$$x = v + \delta, \quad t = \mu + \varepsilon, \quad dv = dx, \quad d\mu = dt.$$

Hence

$$\begin{aligned} Y_2[H(x - \delta, t - \varepsilon) \odot g(x - \delta, t - \varepsilon)] &= \int_0^\infty \int_0^\infty e^{-\frac{v+\delta}{\alpha} - \frac{\mu+\varepsilon}{\beta}} \odot g(v, \mu) dv d\mu \\ &= e^{-\frac{\delta}{\alpha} - \frac{\varepsilon}{\beta}} \int_0^\infty \int_0^\infty e^{-\frac{v}{\alpha} - \frac{\mu}{\beta}} \odot g(v, \mu) dv d\mu = e^{-\frac{\delta}{\alpha} - \frac{\varepsilon}{\beta}} \odot G(\alpha, \beta). \end{aligned}$$

□

Definition 15 ([26]). If $k(t)$ and $g(x, t)$ are fuzzy Riemann integrable functions defined for all $x, t \geq 0$, then fuzzy convolution of $k(t)$ and $g(x, t)$, with respect to t , is given by

$$(k * g)(x, t) = \int_0^t k(t-s)g(x, s)ds$$

and the symbol $*$ denotes the fuzzy convolution with respect to t .

Theorem 15 (Convolution Theorem). Let $k : \mathbb{R}_+ \rightarrow E^1$ and $g : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow E^1$ be continuous fuzzy functions. Then, the FDYT of the convolution of these two functions is as

$$Y_2[(k * g)(x, t)] = Y_t[k(t)]Y_2[g(x, t)]. \quad (15)$$

Proof. Using the definition of fuzzy double Yang transform and convolution, we find

$$\begin{aligned} Y_2[(k * g)(x, t)] &= \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} \odot (k * g)(x, t) dx dt \\ &= \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{t}{\beta}} \odot \left(\int_0^t k(t-s)g(x, s)ds \right) dx dt. \end{aligned}$$

Let $\xi = t - s$, $d\xi = dt$. Then

$$\begin{aligned} Y_2[(k * g)(x, t)] &= \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{\xi+s}{\beta}} \odot \left(\int_0^\infty k(\xi)g(x, s)ds \right) dx d\xi \\ &= \int_0^\infty e^{-\frac{\xi}{\beta}} \odot k(\xi) d\xi \int_0^\infty \int_0^\infty e^{-\frac{x}{\alpha} - \frac{s}{\beta}} \odot g(x, s) dx ds \\ &= Y_t[k(t)]Y_2[g(x, t)]. \end{aligned}$$

□

Theorem 16. Let $g(x, t)$ be a continuous fuzzy function and $Y_2[g(x, t)] = G(\alpha, \beta)$, then

- (i) $Y_2[g'_{x,gH}(x, t)] = (-1) \odot Y_t[g(0, t)] \ominus_{gH} (-1) \frac{1}{\alpha} \odot G(\alpha, \beta);$
- (ii) $Y_2[g'_{t,gH}(x, t)] = (-1) \odot Y_x[g(x, 0)] \ominus_{gH} (-1) \frac{1}{\beta} \odot G(\alpha, \beta);$
- (iii) $Y_2[g''_{xx,gH}(x, t)] = (-1) \odot Y_t[g'_x(0, t)] \ominus_{gH} \left(\frac{1}{\alpha} \odot Y_t[g(0, t)] \ominus_{gH} \frac{1}{\alpha^2} \odot G(\alpha, \beta) \right);$
- (iv) $Y_2[g''_{tt,gH}(x, t)] = (-1) \odot Y_x[g'_t(x, 0)] \ominus_{gH} \left(\frac{1}{\beta} \odot Y_x[g(x, 0)] \ominus_{gH} \frac{1}{\beta^2} \odot G(\alpha, \beta) \right);$
- (v) $Y_2[g''_{xt,gH}(x, y)] = (-1) \odot Y_x[g'_{x,gH}(x, 0)] \ominus_{gH} \left(\frac{1}{\beta} \odot Y_t[g(0, t)] \ominus_{gH} \frac{1}{\alpha\beta} \odot G(\alpha, \beta) \right);$
- (vi) $S_2[g''_{tx,gH}(x, y)] = (-1) \odot Y_t[g'_{t,gH}(0, t)] \ominus_{gH} \left(\frac{1}{\alpha} \odot Y_x[g(x, 0)] \ominus_{gH} \frac{1}{\alpha\beta} \odot G(\alpha, \beta) \right).$

Proof. Using Theorem 11, we find

$$\begin{aligned} Y_2[g'_{x,gH}(x,t)] &= Y_t[Y_x[g'_{x,gH}(x,t)]] = Y_t\left[(-1) \odot g(0,t) \ominus_{gH} (-1)^{\frac{1}{\alpha}} \odot Y_x[g(x,t)]\right] \\ &= (-1) \odot Y_t[g(0,t)] \ominus_{gH} (-1)^{\frac{1}{\alpha}} \odot Y_t[Y_x[g(x,t)]] \\ &= (-1) \odot Y_t[g(0,t)] \ominus_{gH} (-1)^{\frac{1}{\alpha}} \odot G(\alpha, \beta). \end{aligned}$$

In the same manner, we can obtain the case (iii).

$$\begin{aligned} Y_2[g''_{xx,gH}(x,t)] &= Y_t[Y_x[g''_{xx,gH}(x,t)]] \\ &= Y_t\left[(-1) \odot g'_{x,gH}(0,t) \ominus_{gH} \left(\frac{1}{\alpha} \odot g(0,t) \ominus_{gH} \frac{1}{\alpha^2} \odot Y_x[g(x,t)]\right)\right] \\ &= (-1) \odot Y_t[g'_{x,gH}(0,t)] \ominus_{gH} \left(\frac{1}{\alpha} \odot Y_t[g(0,t)] \ominus_{gH} \frac{1}{\alpha^2} \odot Y_t[Y_x[g(x,t)]]\right) \\ &= (-1) \odot Y_t[g'_{x,gH}(0,t)] \ominus_{gH} \left(\frac{1}{\alpha} \odot Y_t[g(0,t)] \ominus_{gH} \frac{1}{\alpha^2} \odot G(\alpha, \beta)\right). \end{aligned}$$

The proof of case (v) is analogous to the proof of case (iii).

$$\begin{aligned} Y_2[g''_{xt,gH}(x,t)] &= Y_x[Y_t[g''_{xt,gH}(x,t)]] \\ &= Y_x\left[(-1) \odot g'_{x,gH}(x,0) \ominus_{gH} (-1)^{\frac{1}{\beta}} \odot Y_t[g'_{x,gH}(x,t)]\right] \\ &= (-1) \odot Y_x[g'_{x,gH}(x,0)] \ominus_{gH} (-1)^{\frac{1}{\beta}} \odot Y_t[Y_x[g'_{x,gH}(x,t)]] \\ &= (-1) \odot Y_x[g'_{x,gH}(x,0)] \ominus_{gH} (-1)^{\frac{1}{\beta}} \odot Y_t\left[(-1) \odot g(0,t) \ominus_{gH} (-1)^{\frac{1}{\alpha}} \odot Y_x[g(x,y)]\right] \\ &= (-1) \odot Y_x[g'_{x,gH}(x,0)] \ominus_{gH} \left(\frac{1}{\beta} \odot Y_t[g(0,t)] \ominus_{gH} \frac{1}{\alpha\beta} \odot S_y[S_x[f(x,y)]]\right) \\ &= (-1) \odot Y_x[g'_{x,gH}(x,0)] \ominus_{gH} \left(\frac{1}{\beta} \odot Y_t[g(0,t)] \ominus_{gH} \frac{1}{\alpha\beta} \odot G(\alpha, \beta)\right). \end{aligned}$$

□

5. Method of Fuzzy Double Yang Transform

To illustrate the use of FDYT, we solve FPVIDE with a memory kernel $k : \mathbb{R}_+ \rightarrow \mathbb{R}_+$. This equation is defined as

$$g'_{t,gH}(x,t) \oplus \int_0^t k(t-s) \odot g(x,t) dx dt = \sigma \odot g''_{xx,gH}(x,t) \oplus f(x,t), \quad x \geq 0, \quad t \geq 0, \quad (16)$$

where σ is any positive constant, $g(x,t)$ is the unknown fuzzy function, and $f(x,t)$ is a given fuzzy function. Furthermore, we assume that the fuzzy Yang transformation of the kernel $k(t)$ satisfies the condition

$$\alpha^2 + \alpha^2 \beta K(\beta) - \sigma \beta \neq 0,$$

where $K(\beta) = Y_t[k(t)]$.

Assume initial conditions of

$$g(x,0) = \psi_0(x) \quad (17)$$

and boundary conditions of

$$g(0, t) = \varphi_0(t), \quad g'_x(0, t) = \varphi_1(t). \quad (18)$$

We will provide an algorithm for solving Equations (16)–(18).

- (i) We apply the fuzzy double Yang transform to both sides of Equation (16);
- (ii) We apply Equation (15) of Theorem 15;
- (iii) We use the derivative properties of FDYT-case (i) and (iii) of Theorem 16;
- (iv) We apply FYT to the initial condition (17) and boundary conditions (18);
- (v) We use Proposition 1 and we obtain $G(\alpha, \beta) = Y_2[g(x, t)]$;
- (vi) We apply the inverse FDYT and we find to the analytical solution of Equations (16)–(18).

First, we apply FDYT to (16) as

$$Y_2[g'_{t,gH}(x, t)] \oplus Y_2 \left[\int_0^t k(t-s) \odot g(x, t) dx dt \right] = Y_2[\sigma \odot g''_{xx,gH}(x, t)] \oplus Y_2[f(x, t)].$$

Using the convolution theorem, we have

$$Y_2[g'_{t,gH}(x, t)] \oplus Y_t[k(t)] \odot Y_2[g(x, t)] = \sigma \odot Y_2[g''_{xx,gH}(x, t)] \oplus Y_2[f(x, t)].$$

The derivative properties of FDYT (Theorem 16) and the above equation yield

$$\begin{aligned} & (-1) \odot Y_x[g(x, 0)] \ominus_{gH} (-1)^{\frac{1}{\beta}} \odot G(\alpha, \beta) \oplus K(\beta) \odot G(\alpha, \beta) \\ & = \sigma \odot \left[(-1) \odot Y_t[g'_x(0, t)] \ominus_{gH} \left(\frac{1}{\alpha} \odot Y_t[g(0, t)] \ominus_{gH} \frac{1}{\alpha^2} \odot G(\alpha, \beta) \right) \right] \oplus F(\alpha, \beta). \end{aligned} \quad (19)$$

where

$$K(\beta) = Y_t[k(t)], \quad G(\alpha, \beta) = Y_2[g(x, t)], \quad F(\alpha, \beta) = Y_2[f(x, t)].$$

Next, apply the FYT to the initial and boundary conditions

$$\Psi_0(\alpha) = Y_x[\psi_0(x)], \quad \Phi_0(\beta) = Y_t[\varphi_0(t)], \quad \Phi_1(\beta) = Y_t[\varphi_1(t)].$$

and then put in (19) as

$$\begin{aligned} & (-1) \odot \Psi_0(\alpha) \ominus_{gH} (-1)^{\frac{1}{\beta}} \odot G(\alpha, \beta) \oplus K(\beta) \odot G(\alpha, \beta) \\ & = (-\sigma) \odot \Phi_1(\beta) \ominus_{gH} \left(\frac{\sigma}{\alpha} \odot \Phi_0(\beta) \ominus_{gH} \frac{\sigma}{\alpha^2} \odot G(\alpha, \beta) \right) \oplus F(\alpha, \beta). \end{aligned}$$

Using Proposition 1, we have

$$\begin{aligned} & \left(\frac{1}{\beta} + K(\beta) - \frac{\sigma}{\alpha^2} \right) \odot G(\alpha, \beta) \\ & = \Psi_0(\alpha) \oplus (-\sigma) \odot \Phi_1(\beta) \oplus \frac{(-\sigma)}{\alpha} \odot \Phi_0(\beta) \oplus F(\alpha, \beta). \end{aligned}$$

Hence

$$G(\alpha, \beta) = B \odot \left[\Psi_0(\alpha) \oplus (-\sigma) \odot \Phi_1(\beta) \oplus \frac{(-\sigma)}{\alpha} \odot \Phi_0(\beta) \oplus F(\alpha, \beta) \right], \quad (20)$$

where

$$B = \frac{\alpha^2 \beta}{\alpha^2 + \alpha^2 \beta K(\beta) - \sigma \beta}. \quad (21)$$

Finally, take the inverse FDYT of (20) as

$$g(x, t) = Y_2^{-1} \left[B \odot \left(\Psi_0(\alpha) \oplus (-\sigma) \odot \Phi_1(\beta) \oplus \frac{(-\sigma)}{\alpha} \odot \Phi_0(\beta) \oplus F(\alpha, \beta) \right) \right]. \quad (22)$$

6. Examples

Example 1. Consider the following FVIDE

$$g'_{t,gH}(x, t) \oplus \int_0^t k(t-s) \odot g(x, t) dx dt = g''_{xx,gH}(x, t) \oplus f(x, t), \quad x \geq 0, \quad t \geq 0 \quad (23)$$

with initial conditions

$$g(x, 0, r) = x \odot (1, 2, 3) \quad (24)$$

and boundary conditions

$$g(0, t, r) = 0 \odot (1, 2, 3), \quad g'_x(0, t, r) = e^{-t} \odot (1, 2, 3). \quad (25)$$

In this case

$$\sigma = 1, \quad k(t) = 2e^t, \quad f(x, t) = (xe^t - 2xe^{-t}) \odot (1, 2, 3).$$

Then, we have

$$\psi_0(x) = x \odot (1, 2, 3), \quad \varphi_0(t) = 0 \odot (1, 2, 3), \quad \varphi_1(t) = e^{-t} \odot (1, 2, 3).$$

Hence

$$Y_t[k(t)] = K(\beta) = \frac{2\beta}{1-\beta} \odot (1, 2, 3), \quad Y_x[\psi_0(x)] = \Psi_0(\alpha) = \alpha^2 \odot (1, 2, 3),$$

$$Y_t[\varphi_0(t)] = \Phi_0(\beta) = 0 \odot (1, 2, 3), \quad Y_t[\varphi_1(t)] = \Phi_1(\beta) = \frac{\beta}{1+\beta} \odot (1, 2, 3),$$

$$Y_2[f(x, t)] = F(\alpha, \beta) = \alpha^2 \beta \left(\frac{1}{1-\beta} + \frac{2}{1+\beta} \right) \odot (1, 2, 3).$$

By substituting the values of the fuzzy functions $K(\beta)$, $\Psi_0(\alpha)$, $\Phi_0(\beta)$ and $\Phi_1(\beta)$ in Equations (20) and (21), we obtain

$$G(\alpha, \beta) = B \odot \left[\alpha^2 \odot (1, 2, 3) \oplus \frac{(-\beta)}{1+\beta} \odot (1, 2, 3) \oplus \alpha^2 \beta \left(\frac{1}{1-\beta} + \frac{2}{1+\beta} \right) \odot (1, 2, 3) \right],$$

where

$$B = \frac{\alpha^2 \beta (1 - \beta)}{\alpha^2 (2\beta^2 - \beta + 1) - \beta (1 - \beta)}.$$

Hence

$$G(\alpha, \beta) = B \left[\alpha^2 - \frac{\beta}{1+\beta} + \frac{\alpha^2 \beta}{1-\beta} - \frac{2\alpha^2 \beta}{1+\beta} \right] \odot (1, 2, 3) = \frac{\alpha^2 \beta}{1+\beta} \odot (1, 2, 3).$$

Taking the inverse of FDYT, we find the solution to the Equations (23)–(25) is

$$g(x, t) = Y_2^{-1}[G(\alpha, \beta)] = Y_2^{-1} \left[\frac{\alpha^2 \beta}{1+\beta} \odot (1, 2, 3) \right] = xe^{-t} \odot (1, 2, 3).$$

7. Conclusions

In this research paper, we introduce a new fuzzy integral transformation called FDYT, which is defined with the help of the fuzzy unitary Yang transform. We find conditions for its existence and establish some of its basic properties. We prove theorems about partial derivatives and fuzzy unit convolution. Using these new results, we successfully obtained the exact solution to an FVIDE with a symmetric memory kernel. We construct a numerical example to verify the application of the new method.

Fuzzy integral transforms cannot solve nonlinear problems directly unless they are combined with iteration methods. As a result, we propose that this method is further expanded in future work, so that it can be applied to the solution of various nonlinear fuzzy partial differential and integro-differential equations related to physical and engineering problems.

Author Contributions: Conceptualization, A.G., S.I.C. and M.V.; methodology, A.G., S.I.C. and M.V.; validation, A.G., S.I.C. and M.V.; formal analysis, A.G., S.I.C., M.V. and Y.G.; writing—original draft preparation, A.G., S.I.C., M.V. and Y.G.; writing—review and editing, A.G., S.I.C., M.V. and Y.G.; funding acquisition, A.G., S.I.C., M.V. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study is financed by the European Union-Next Generation EU through the National Recovery and Resilience Plan of the Republic of Bulgaria, project DUECOS BG-RRP-2.004-0001-C01.

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. Habetler, G.J.; Schiffman, R.L. A finite difference method for analyzing the compression of poro-viscoelastic media. *Computing* **1970**, *6*, 342–348. [CrossRef]
2. Yanik, E.G.; Fairweather, G. Finite element methods for parabolic and hyperbolic partial integro-differential equations. *Nonlinear Anal. Theory Methods Appl.* **1988**, *12*, 785–809. [CrossRef]
3. Souplet, P. Blow-up in nonlocal reaction-diffusion equations. *SIAM J. Math. Anal.* **1998**, *29*, 1301–1334. [CrossRef]
4. Dehghan, M.; Shokri, A. A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. *Comput. Math. Appl.* **2007**, *54*, 136–146. [CrossRef]
5. Dehghan, M.; Shakeri, F. Solution of parabolic integro-differential equations arising in heat conduction in materials with memory via He's variational iteration technique. *Int. J. Numer. Methods Biomed. Eng.* **2010**, *26*, 705–715. [CrossRef]
6. Leite, J.; Ceconello, M.; Leite, J.; Bassanezi, R. C. On Fuzzy Solutions for Diffusion Equation. *J. Appl. Math.* **2015**, *2015*, 874931. [CrossRef]
7. Di Martino, F.; Perfilieva, I.; Sessa, S. A Summary of F-Transform Techniques in Data Analysis. *Electronics* **2021**, *10*, 1771. [CrossRef]
8. Kadham, S.; Alkiffai, A. Model Tumor Response to Cancer Treatment Using Fuzzy Partial SH-Transform: An Analytic Study. *Int. J. Math. Comput. Sci.* **2022**, *18*, 23–28.
9. Napole, C.; Derbeli, M.; Barambones, O. Experimental Analysis of a Fuzzy Scheme against a Robust Controller for a Proton Exchange Membrane Fuel Cell System. *Symmetry* **2022**, *14*, 139. [CrossRef]
10. Almeida, C.G.; Meyer, J.F.; Vasconcelos, H.L. Fuzzy parameter in a partial differential equation model for population dispersal of Leaf-Cutting Ants. *Nonlinear Anal. Real World Appl.* **2011**, *12*, 3397–3412.
11. Routaray, M.; Sahu, P.K.; Chalisehar, D.N. The Fuzzy Differential Transform Method for the Solution of the System of Fuzzy Integro-Differential Equations Arising in Biological Model. *Mathematics* **2023**, *11*, 3840. [CrossRef]
12. Bede, B.; Rudas, I.J.; Bencsik, A.L. First order linear fuzzy differential equations under generalized differentiability. *Inf. Sci.* **2007**, *177*, 1648–1662. [CrossRef]
13. Nieto, J.; Khastan, A.; Ivaz, K. Numerical solution of fuzzy differential equations under generalized differentiability. *Nonlinear Anal. Hybrid Syst.* **2009**, *3*, 7000–7007. [CrossRef]
14. Osman, M.; Gong, Z.; Mustafa, A. Comparison of fuzzy Adomian decomposition method with fuzzy VIM for solving fuzzy heat-like and wave-like equations with variable coefficients. *Adv. Differ. Equ.* **2020**, *2020*, 327. [CrossRef]

15. Osman, M.; Xia, Y.; Omer, O.A.; Hamoud, A. On the Fuzzy Solution of Linear-Nonlinear Partial Differential Equations. *Mathematics* **2022**, *10*, 2295. [CrossRef]
16. Stabestari, R.M.; Ezzati, R. The Fuzzy Double Laplace Transforms and their Properties with Applications to Fuzzy Wave Equation. *New Math. Natural Comp.* **2021**, *17*, 319–338. [CrossRef]
17. Kshirsagar, K.A.; Nikam, V.R.; Gaikwad, S.B.; Tarate, S.A. The double fuzzy Elzaki transform for solving fuzzy partial differential equations. *J. Chungcheong Math. Soc.* **2022**, *35*, 2.
18. Abdeljawad, T.; Younus, A.; Alqudah, M.A.; Atta, U. On Fuzzy Conformable Double Laplace Transform with Applications to Partial Differential Equations. *Comput. Model. Eng. Sci.* **2023**, *134*, 2163–2191.
19. Gumah, G.; Naser, M.; Al-Smadi, M.; Al-Omari, S. Application of reproducing kernel Hilbert space method for solving second order fuzzy Volterra integro-differential equations. *Adv. Differ. Equ.* **2018**, *475*, 1–15. [CrossRef]
20. Ullaha, Z.; Ahmada, S.; Ullaha, A.; Akgul, A. On solution of fuzzy Volterra integro-differential equations. *Arab. J. Basic Appl. Sci.* **2021**, *28*, 330–339.
21. Araour, M.; Mennouni, A. A New Procedures for Solving Two Classes of Fuzzy Singular Integro-Differential Equations: Airfoil Collocation Methods. *Int. J. Appl. Comput. Math.* **2022**, *8*, 35. [CrossRef]
22. Gumah, G. Reproducing kernel Hilbert space method to solve fuzzy partial Volterra integro-differential equations. *Filomat* **2024**, *38*, 8553–8564.
23. Qahremania, E.; Allahviranloo, T.; Abbasbandy, S.; Ahmady, N. A study on the fuzzy parabolic Volterra partial integro-differential equations. *J. Intell. Fuzzy Syst.* **2021**, *40*, 1639–1654. [CrossRef]
24. Georgieva, A.; Spasova, M. Solving partial fuzzy integro-differential equations using fuzzy Sumudu transform method. *AIP Conf. Proc.* **2021**, *2321*, 030010-1–030010-8.
25. Georgieva, A. Double Fuzzy Sumudu transform to solve partial Volterra fuzzy integro-differential equations. *Mathematics* **2020**, *8*, 692. [CrossRef]
26. Georgieva, A.; Spasova, M. Solution of partial fuzzy integro-differential equations by double natural transform. *AIP Conf. Proc.* **2022**, *2459*, 030012-1–030012-8.
27. Yang, X.J. A new integral transform method for solving steady heat-transfer problem. *Therm. Sci.* **2016**, *20*, 639–642. [CrossRef]
28. Ullah, A.; Ullah, A.; Ahmad, S.; Van Hoa, N. Fuzzy Yang transform for second order fuzzy differential equations of integer and fractional order. *Phys. Scr.* **2023**, *98*, 044003. [CrossRef]
29. Armand, A.; Allahviranloo, T.; Gouyandeh, Z. Some fundamental results on fuzzy calculus. *Iran. J. Fuzzy Syst.* **2018**, *15*, 27–46.
30. Bede, B.; Stefanini, L. Generalized differentiability of fuzzy-valued functions. *Fuzzy Sets Syst.* **2013**, *230*, 119–141. [CrossRef]
31. Anastassiou, G.A. Fuzzy Mathematics: Approximation Theory. In *Studies in Fuzziness and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2010.
32. Allahviranloo, T.; Gouyandeh, Z.; Ahmand, A.; Hasanoglu, A. On fuzzy solutions of heat equation based on generalized Hukuhara differentiability. *Fuzzy Sets Syst.* **2015**, *265*, 1–23. [CrossRef]
33. Gouyandeh, Z.; Allahviranloo, T.; Abbasbandy, S.; Armand, A. A fuzzy solution of heat equation under generalized Hukuhara differentiability by fuzzy Fourier transform. *Fuzzy Sets Syst.* **2017**, *309*, 81–97. [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

Kinematic Fuzzy Logic-Based Controller for Trajectory Tracking of Wheeled Mobile Robots in Virtual Environments

José G. Pérez-Juárez ¹, José R. García-Martínez ^{2,*}, Alejandro Medina Santiago ³, Edson E. Cruz-Miguel ², Luis F. Olmedo-García ¹, Omar A. Barra-Vázquez ¹ and Miguel A. Rojas-Hernández ¹

¹ Laboratorio de Control y Robótica, Facultad de Ingeniería en Electrónica y Comunicaciones, Universidad Veracruzana, Poza Rica 93390, Mexico; joseperez07@uv.mx (J.G.P.-J.); lolmedo@uv.mx (L.F.O.-G.); omabarra@uv.mx (O.A.B.-V.); mrojas@uv.mx (M.A.R.-H.)

² Análisis de Sistemas y Tecnologías Emergentes, Facultad de Ingeniería en Electrónica y Comunicaciones, Universidad Veracruzana, Poza Rica 93390, Mexico; edsoncruz@uv.mx

³ SECIHTI-National Institute for Astrophysics, Optics and Electronics, Computer Science Coordination, Puebla 72840, Mexico; amedina@inaoep.mx

* Correspondence: romangarcia@uv.mx

Abstract: Mobile robots represent one of the most relevant areas of study within robotics due to their potential for designing and developing new nonlinear control structures that can be implemented in simulations and applications in specific environments. In this work, a fuzzy steering controller with a symmetric distribution of fuzzy numbers is proposed and designed for implementation in the kinematic model of a non-holonomic mobile robot. The symmetry in the distribution of triangular fuzzy numbers contributes to a balanced response to disturbances and minimizes systematic errors in direction estimation. Additionally, it improves the system's adaptability to various reference paths, ensuring accurate tracking and optimized performance in robot navigation. Furthermore, this fuzzy logic-based controller emulates the behavior of a classic PID controller by offering a robust and flexible alternative to traditional methods. A virtual environment was also developed using the UNITY platform to evaluate the performance of the fuzzy controller. The results were evaluated by considering the average tracking error, maximum error, steady-state error, settling time, and total distance traveled, emphasizing the trajectory error. The circular trajectory showed high accuracy with an average error of 0.0089 m, while the cross trajectory presented 0.01814 m, reflecting slight deviations in the turns. The point-to-point trajectory registered a more significant error of 0.9531 m due to abrupt transitions, although with effective corrections in a steady state. The simulation results validate the robustness of the proposed fuzzy controller, providing quantitative insights into its precision and efficiency in a virtual environment, and demonstrating the effectiveness of the proposal.

Keywords: fuzzy controller; mobile robot; UNITY; kinematics control

1. Introduction

Mobile robots have gained remarkable popularity in recent years, extending their presence to several areas, such as industrial, professional, and educational settings [1]. In the latter, students actively participate in developing robots for school projects, contributing to the continuous growth of mobile robotics. This growth has generated the need to locate mobile robots at any point in space, which requires obtaining a kinematic model that allows one to determine in real time the robot coordinates and its orientation for a fixed reference frame [2].

Within the field of mobile robotics, there is a wide variety of models with different configurations, depending on their structural type or physical characteristics, resulting in various types of movement and, therefore, different kinematics in each case. Ackerman-type mobile robots are characterized because their movement is comparable to that of a car [3]. For this reason, its kinematic model allows for applications such as autonomous ground vehicles and exploration robots [4]; this same model, characterized by the inherent turning restrictions of its directional system, poses significant challenges, such as designing controllers that allow precise tracking of the trajectory.

The design of kinematic controllers is essential for ensuring that mobile robots follow trajectories accurately and efficiently [5]. Among the most common and prominent techniques is the use of proportional–integral–derivative (PID) controllers, whose simplicity and effectiveness have always been characterized. However, for complex scenarios, integrating fuzzy logic (FL) in the controller design offers a more robust strategy capable of adapting to variable conditions [6]. However, implementing these controllers often requires advanced navigation sensors that accurately measure the robot’s movement and orientation in real applications. These sensors are essential to address environmental challenges and ensure reliable performance.

Detailed simulations are crucial before implementing kinematic control systems in a physical environment. These allow for the evaluation of the system’s behavior under different conditions and the adjustment of parameters to optimize the design [7]. Today, multiple virtual environments offer advanced tools for designing and testing controllers, facilitating the development of more robust and secure solutions for mobile robots. Unity is a 3D development engine widely used in video game development. However, its applications are not limited to that; its great versatility has made it a key tool in numerous fields, such as education, simulation, and autonomous system design [8,9]. Unity is a flexible and powerful tool that allows one to model physical systems and replicate real-world scenarios with great fidelity [10].

This article offers two significant contributions to the state of the art:

- The design of a fuzzy kinematic controller for an Ackermann-type robotic structure offers an efficient and flexible solution for regulating robot motions, standing out for its ability to adaptively adjust gains in real-time. Unlike traditional methods, this controller does not require an exact mathematical model of the system, which simplifies its implementation and improves robustness against uncertainties and variations in the environment. The main innovation lies in the use of a fuzzy approach to determine PID-type gains, allowing for greater precision and stability in tracking specific trajectories. This dynamic adaptation capability makes the controller especially useful in highly variable scenarios where conventional methods could fail to maintain optimal system performance.
- A methodology for implementing the controller in a virtual environment developed in UNITY, which accurately simulates the robot’s behavior. In this environment, control data are received and applied to the virtual robot, allowing the robot to move along predefined trajectories. This implementation validates the controller’s performance and provides a clear and detailed view of the robot’s behavior in different scenarios, facilitating the evaluation of its performance and the continuous improvement of the system. Real-time visualization within the virtual environment allows the controller to be efficiently tuned and optimized before its implementation on physical hardware.

This paper is structured as follows: Section 2 briefly discusses related works to our proposed research. Section 3 presents the developed methodology, and is divided into three main parts: the derivation of the kinematic model for the Ackermann-type robot, the design of the fuzzy PID controller, and the development of the 3D simulation in Unity. Section 4

presents the simulation results, considering different trajectories. Section 5 discusses the performance of the kinematic fuzzy-logic-based controller for trajectory tracking. Finally, Section 6 provides the conclusions and references.

2. Related Works

Trajectory tracking is one of the critical challenges in mobile robotics as it requires advanced control techniques that are capable of handling uncertainty and nonlinearity [11]. To better understand the current landscape on this subject, we performed a bibliometric analysis that visualized development trends and emerging areas of interest within the literature, as shown in Figure 1. Our study revealed that traditional control methods, such as PID and adaptive control laws, are highly cited. However, recent research shows a growing interest in optimal control, reinforcement learning, fuzzy logic-based controllers (FLCs), and other artificial intelligence techniques. Classical PID controllers are used due to their simplicity and effectiveness in maintaining stability; however, their performance deteriorates in highly nonlinear systems, requiring parameter tuning and model linearization as observed in [12]. Shojaei et al. [13] applied an adaptive control law to a differential-type robot; in this, the kinematic model was obtained, and from it, the linearization of the system was carried out using the feedback technique to obtain a model in which a control system can be applied. Although this technique presents robust results and is applied to a physical model, it is necessary to have a linear model to implement or resort to linearization techniques to approximate a model. To address these limitations, optimal control methods such as time-varying linear quadratic control (TVLQ) have been proposed by Caran et al. [14]. Their optimal control-based algorithm is founded upon the direct numerical calculation of the time-varying elements of the control law matrix, avoiding formalism of the set of linear matrix inequalities (LMIs) and the approximation of the solution of the Riccati partial differential matrix equation. On the other hand, Veselov et al. [15] took into account the nonlinear properties of a mobile tracked robot and employed an analytical design of aggregated regulators (ADARs), a method derived from synergetic control theory (SCT) that allows the synthesis of control laws for complex nonlinear systems without using linearization procedures. Unlike model-based approaches, which require precise system identification, reinforcement learning methods have been explored. Xie et al. [16] solved the trajectory tracking problem using an optimized reward reinforcement learning (ORRL) algorithm based on the Q-learning framework. Similarly, Ha et al. [17] evaluated the performance of an intelligent controller built on a deep Q-network (DQN) algorithm. While these approaches demonstrate good adaptive ability in complex driving environments, they demand significant computational resources and mathematical formulation, which may limit their real-time applicability. In contrast, fuzzy-logic-based controllers offer a viable alternative in such situations, as they dynamically adjust gains based on system behavior, making them particularly well-suited for scenarios where uncertainties or variations may arise. Several researchers have evaluated the effectiveness of FLCs in mobile robotics. Khesrani et al. [18] employed FL to auto-tune the endogenous feedback controller parameters, demonstrating good tracking of the desired tasks despite uncertainties. Meanwhile, Thuong et al. [19] implemented a proportional-derivative FL controller for trajectory following, while Mai et al. [20] combined a backstepping and adaptive fuzzy PID approach for a nonholonomic mobile robot. These results demonstrated good performance in terms of small distance errors, fast responses, and accuracy. Recent works have further explored fuzzy controllers integrated with genetic algorithms [21] and neural networks [22] to enhance system performance. A comprehensive description of fuzzy hybridization with other artificial intelligence techniques, as well as current fuzzy logic approaches, is reviewed in [23,24]. In order to evaluate the performances of different

3. Materials and Methods

3.1. Mathematical Model

Within the group of four-wheeled mobile robots, the Ackerman-type configuration consists of two groups of two wheels: the group of rear drive wheels that are responsible for providing traction to the vehicle and the group of front wheels that are responsible for giving direction to the robot. In both groups, there is a single actuator in charge of this action, that is, traction and direction are combined. The movement of the Ackerman robot is characterized by how it addresses turns. Like other robot models, the robot can move in a straight line without a problem but cannot turn instantly since it requires a certain minimum radius to act. In addition, the drive wheels will experience some slippage during this action.

The representation of the Ackerman robot movement can be visualized in Figure 2; from there, it is possible to obtain the mathematical model that describes the kinematics of its movement; this kinematic model (KM) will be used to carry out the simulation. In Table 1, it is possible to observe the description of each of the elements.

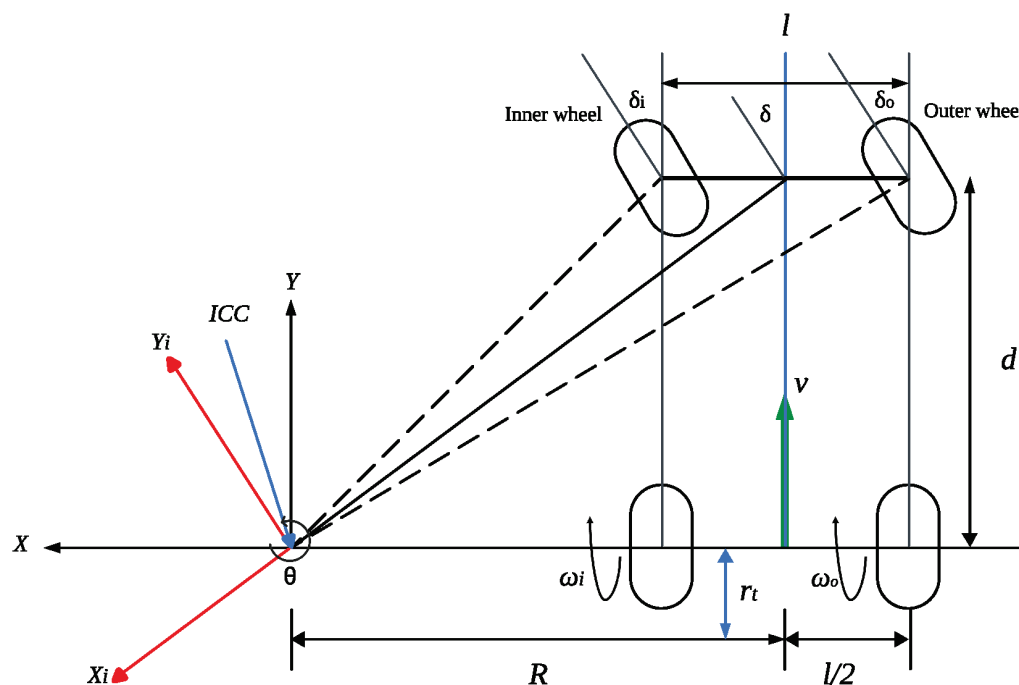


Figure 2. Kinematic model of the Ackerman-type robot.

Table 1. Elements of the Ackermann-type robot’s kinematic model.

Variable	Description
ICC	Instantaneous center of curvature
l	Separation distance between front wheels
d	Separation distance between side wheels
δ	Front wheels turning angle
δ_i	Inner wheel turning angle
δ_o	Outer wheel turning angle
v	Linear speed of the robot
R	Radius of gyration (radius of curvature of the path)
ω_i	Inner wheel angular velocity
ω_o	Outer wheel angular velocity
θ	Turning angle

Analyzing the Ackerman configuration displayed in Figure 2, we observe how the turn occurs in this configuration. This model is distinctive in that it cannot rotate around its own reference system. The Ackerman configuration, when turning, does so through a circumference with a central point known as the instantaneous center of curvature (ICC). The steering angles of the front wheels determine this rotation center. The ICC is located precisely at the intersection of the extended axes of all the wheels, and the robot will rotate around this point. The ICC is not a fixed point; it can change at any moment with the slightest adjustment in the steering angle of the front wheels. Consequently, the dimensions of the turning circle will also be affected. The ICC does not depend on the number of wheels but rather on the number of constraints. In the case of the Ackerman configuration, the ICC is determined by two kinematic constraints: 1. The front wheels produce one constraint. Despite having two wheels, they generate the same set of constraints. 2. The rear axle provides the second constraint, which is fixed and cannot rotate [33]. In short, when the robot moves its front wheels, its entire body will turn around an instantaneous circle of radius R . This angle can be estimated by Equation (1):

$$R = \frac{d}{\tan \delta} \quad (1)$$

this equation describes how the curvature of the turning radius of the Ackerman robot is inversely proportional to the tangent of the steering angle, δ , and the distance between the lateral axes, d . When R tends to infinity, the movement is straight. Additionally, it is possible to observe how the physical characteristics and movements of the robot affect its trajectory.

The Ackerman robot's mechanism allows movement so the rear wheels do not produce a steering angle. Based on this, it is possible to determine the orientations of each wheel, with each wheel producing a different angle concerning the ICC; the inner wheel must be oriented at an angle, δ_i , greater than the outer wheel, δ_o ; this allows the robot to turn around the midpoint between the rear wheel axes. The resulting angle at the midpoint corresponds to δ . The steering angles of the front wheels are determined by Equations (2) and (3).

$$\delta_o = \frac{\pi}{2} + \arctan \frac{R + \frac{l}{2}}{d} \quad (2)$$

$$\delta_i = \frac{\pi}{2} + \arctan \frac{R - \frac{l}{2}}{d} \quad (3)$$

Consequently, the inner wheel must move at a lower speed than the outer wheel, that is, $v_i < v_o$. Due to this configuration and the fact that only one actuator is responsible for the rotation of the rear wheels, both wheels rotate at the same angular speed, denoted as ω , around ICC. Therefore, using Equations (4) and (5), it is possible to calculate the linear velocities of each of the rear wheels:

$$v_o = \omega \left(R + \frac{l}{2} \right) \quad (4)$$

$$v_i = \omega \left(R - \frac{l}{2} \right) \quad (5)$$

To obtain the direct KM (DKM) of this configuration, it must be assumed that the steering angles of the front wheels are equal (as if there were only one steering wheel). This is because the steering angle around ICC is taken at the center of this axis, resulting in an average of both angles denoted by Equation (6):

$$\delta = \frac{\pi}{2} + \arctan \left(\frac{R}{d} \right) \quad (6)$$

then, the DKM for an inertial reference system is constituted by Equation (7).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \omega \end{bmatrix} \quad (7)$$

This model describes linear velocities for the values of $\dot{x}$ and $\dot{y}$, and angular velocities for the value of $\dot{\theta}$. In this model, it can be observed that the linear velocities $\dot{x}$ and $\dot{y}$ depend on the system's rotation angle θ , while the rotational velocity $\dot{\theta}$ strictly depends on the angular velocity of the rear wheels, ω , since they are identical. This velocity can be determined using Equation (8):

$$\omega = \frac{v}{R} \quad (8)$$

It is assumed that $\dot{\theta}$ can be expressed in terms of the linear speed (v) and the radius of ICC (R). Still, it is also possible to express it in terms of the linear speed (v), the turning angle of the front wheels (δ), and the distance between the lateral axles (d); therefore, $\dot{\theta}$ can be expressed as shown in Equation (9):

$$\dot{\theta} = \frac{v}{R} = \frac{v}{d} \tan \delta \quad (9)$$

This equation can be used for the simulation process since ω corresponds to a physical measurement. By substituting Equation (9) into Equation (7), the following DKM of the Ackerman robot is obtained, as presented in Equation (10):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \frac{v}{d} \tan \delta \end{bmatrix} \quad (10)$$

Equation (10) and the modification in Equation (9) are implemented for the simulation. Since these equations are expressed in the continuous domain, it is necessary to apply temporal discretization. To achieve this, we use the forward finite difference method. The formulas obtained from this process are presented in Equations (11)–(13):

$$x[k] = x[k-1] + \Delta t \cdot v \cos(\theta[k-1]) \quad (11)$$

$$y[k] = y[k-1] + \Delta t \cdot v \sin(\theta[k-1]) \quad (12)$$

$$\theta[k] = \theta[k-1] + \frac{v \cdot \Delta t}{d} \tan(\delta) \quad (13)$$

The discrete model presented in Equations (11) and (12) does not consider possible changes in the robot orientation during time intervals. Therefore, it can produce errors when curved trajectories are performed or when Δt is not small enough. Consequently, corrections to the basic model based on numerical integration were added to obtain more precise approximations. This adjustment in the model can be observed in Equations (14) and (15):

$$x[k] = x[k-1] + \Delta t \cdot v \cos\left(\theta[k-1] + \left(\frac{v \cdot \Delta t}{2d}\right)\right) \quad (14)$$

$$y[k] = y[k-1] + \Delta t \cdot v \sin\left(\theta[k-1] + \left(\frac{v \cdot \Delta t}{2d}\right)\right) \quad (15)$$

The term $\frac{v \cdot \Delta t}{2d}$ represents the midpoint of the angular change; this refinement significantly enhances the accuracy for non-linear trajectories, resulting in more precise approximations.

Equations (13)–(15) will be used later in the design of the controller and for 3D simulations within Unity.

3.2. Controller Proposal

The implemented controller corresponds to a PID control with adjustable gains using FL. PID controllers are widely recognized and used for their efficiency and simplicity. However, as mentioned before, their performance can be limited in systems with non-linear dynamics or high uncertainty, factors that represent significant challenges in their implementation [34]. It is precisely in this context that FL demonstrates its usefulness. A fuzzy PID controller can adjust its gains, adapting to changing environments dynamically. This provides greater flexibility to the system and significantly improves the controller's performance [35]. The mathematical model of the PID controller is represented by Equation (16).

$$\delta(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t) \quad (16)$$

where $e(t)$ is the error, and K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively. $\delta(t)$ is the orientation of the KM of the mobile robot. These are the values that FL will adjust. The control diagram of the system can be seen in Figure 3.

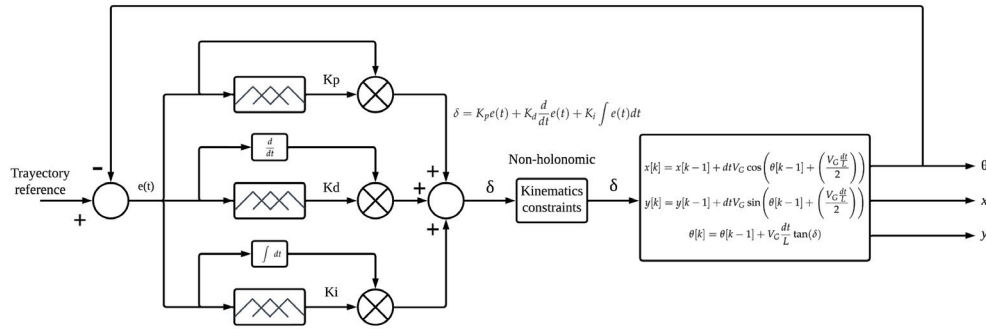


Figure 3. Control structure proposed.

According to the control scheme presented in Figure 3, it can be observed that the fuzzy PID control focuses on controlling the robot's steering angle $\delta(t)$. The input to the controller is a reference trajectory, from which the trajectory generated by the simulation is subtracted to determine the directional error that can be visualized in Equation (17):

$$e(t) = \theta_{des} - \theta(t) \quad (17)$$

Considering the trajectory data, $\theta(t)$ is determined at each trajectory moment and compared with the results obtained from the simulation. θ_{des} is the desired orientation. This comparison is made within the kinematic limitations of the model; in this study, $\delta(t)$ is restricted to a range from $\frac{\pi}{4}$ to $-\frac{\pi}{4}$ rad.

Subsequently, $\delta(t)$ is utilized in the discrete DKM through Equations (13)–(15), yielding the positions x and y , as well as the ICC angle $\theta(t)$. The obtained values of x and y are employed to generate the trajectory for simulation, after which, the process returns to the start for error comparison. On the other hand, to demonstrate the system's stability, the closed-loop transfer function of the orientation of the Ackermann structure must be found [36]. First, the steering angle Equation (6) is substituted into Equation (17), corresponding to the PID controller. This substitution yields Equation (17), describing the behavior of the controller in terms of the variables $\delta(t)$ and $\dot{\delta}(t)$:

$$\dot{\theta}(t) = \frac{v(t)}{d} \tan \left(K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \right) \quad (18)$$

This equation exhibits non-linear behavior due to the function $\tan(\cdot)$, which introduces complexities in its analysis and a direct solution using conventional analytical techniques. However, under the assumption of small deviations in the steering angle ($\delta(t) \approx 0$), applying a linearization by first-order approximation is possible, considering that $\tan(\delta(t)) \approx \delta(t)$. This simplification reduces the complexity of the model, allowing the system to be represented by linear equations that facilitate both the stability analysis and the design of controllers using classical linear control methods, as shown in Equation (19):

$$\dot{\theta}(t) \approx \frac{v(t)}{d} \left(K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \right) \quad (19)$$

If the velocity is assumed to be constant $v(t) = V$, Equation (20) can be obtained:

$$\dot{e}(t) + \frac{V}{d} K_p e(t) + \frac{V}{d} K_i \int e(t) dt + \frac{V}{d} K_d \frac{de(t)}{dt} = 0 \quad (20)$$

The Laplace transform with zero initial conditions is applied to Equation (20) to obtain Equation (21), which represents the behavior of the PID controller in the complex domain:

$$\left(s + \frac{V}{d} K_d s + \frac{V}{d} K_p + \frac{V}{d} \frac{K_i}{s} \right) E(s) = 0 \quad (21)$$

By applying the Laplace transform to Equation (17) and substituting the result into Equation (21), followed by rearranging the terms, Equation (22) is derived:

$$G_{cl}(s) = \frac{\theta(s)}{\theta_{ref}(s)} = \frac{\frac{V}{d} (K_d s^2 + K_p s + K_i)}{\left(1 + \frac{V}{d} K_d \right) s^2 + \frac{V}{d} K_p s + \frac{V}{d} K_i} \quad (22)$$

The poles are evaluated by solving the characteristic equation derived from the transfer function presented in Equation (22) to assess the stability of the system. This characteristic equation corresponds to a second-order system, whose general form allows the identification of the coefficients that determine the location of the poles in the complex plane. The poles of the system, which define its dynamic behavior, are expressed by Equation (23). The location of these poles is essential as it determines whether the system is stable, unstable, or oscillatory. For the system to be stable, both poles must have a negative real part, thus ensuring that the oscillations decrease over time and the system tends to a steady state:

$$s = \frac{-\frac{V}{d} K_p \pm \sqrt{\left(\frac{V^2}{d^2} K_p^2 \right) - 4 \frac{V}{d} K_i \left(1 + \frac{V}{d} K_d \right)}}{2 \left(1 + \frac{V}{d} K_d \right)} \quad (23)$$

If $K_p^2 - 4(1 + K_d)K_i > 0$, the poles of the system are real and negative whenever $K_p > 0$, which ensures that the system will be stable and free of oscillations. This condition is especially suitable for robot orientation control since a non-oscillatory response ensures more precise and smoother movements, avoiding unwanted overshoots that could affect the robot's trajectory. Based on this assumption, the fuzzy controller is designed to adjust the control parameters dynamically to maintain stability under different operating conditions. This ensures that the poles remain in the left half-plane and that the system responds stably and efficiently. For the steering angle of an Ackermann robot, the linguistic variable error, error derivative, and error integral are employed to replicate the structure of a classical PID adapted to nonlinear systems. The error indicates the angular deviation from the desired path, the error derivative anticipates abrupt changes in orientation to improve dynamic response, and the error integral corrects persistent deviations, ensuring accurate

tracking. Three fuzzy systems are designed to tune the controller gains, one for each control gain (K_p , K_i , K_d), using the error, error integral, and error derivative as linguistic variables, respectively. Linguistic values are described by the following descriptors: *negative-large* (NB), *negative-small* (NS), *zero* (ZE), *positive-small* (PS), and *positive-large* (PL), allowing continuous adaptation of the gains to optimize steering angle control while respecting the holonomic constraints of the system.

For the gain K_p , five linguistic values are defined in Table 2, distributed symmetrically to ensure an equitable adjustment in calculating the control gains. This symmetry is essential to avoid biases in the controller response, ensuring that the corrections are homogeneous when faced with deviations in both directions. Furthermore, it allows for a smooth transition between the different gain levels, improving the accuracy of the system. Triangular membership functions are used to represent each of the linguistic values. This type of function is selected due to its practicality in situations with a clearly defined central value or inflection point. Furthermore, they stand out for their mathematical simplicity and computational efficiency, significantly reducing the load during the inference process. This feature makes them an ideal choice for systems where performance is necessary. Thanks to their ease of implementation and low computational costs, triangular membership functions are particularly useful in simulation applications, where a balance between accuracy and efficiency is required. The triangular membership functions corresponding to these linguistic values, designed with a symmetrical distribution, are illustrated in Figure 4, highlighting their impact on the linearity of the adjustment.

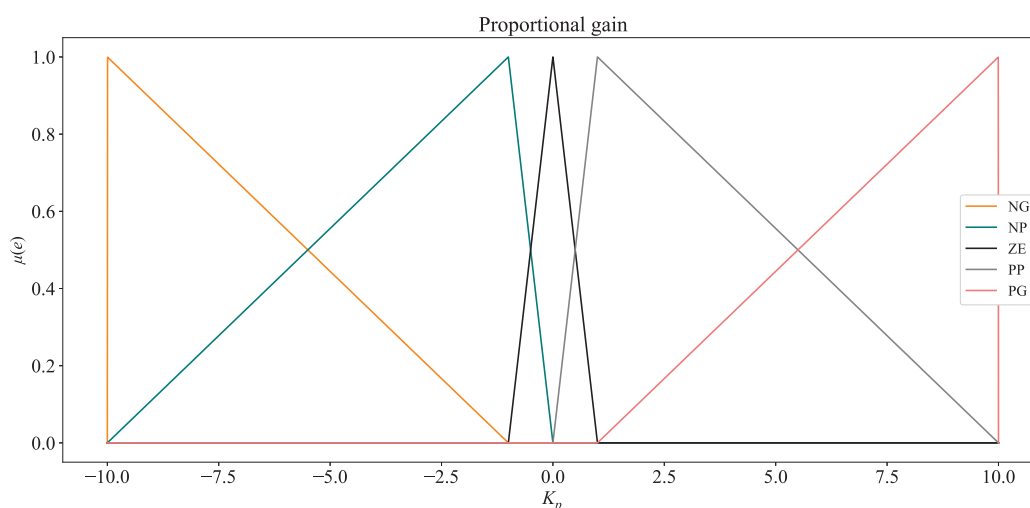


Figure 4. Error linguistic variable distribution.

Table 2. Error linguistic variable.

Linguistic Value	Positions
NL	$[-10, -10, -1]$
NS	$[-10, -1, 0]$
ZE	$[-1, 0, 1]$
PS	$[0, 1, 10]$
PL	$[1, 10, 10]$

For the output, singleton functions were chosen to reduce the computational costs; the descriptions of the output values can be seen in Table 3. Only positive values were selected for singletons because controller gains were restricted to positive values to ensure stability. Therefore, three linguistic values, big (B), medium (M), and small (S), were selected to compute the gain K_p .

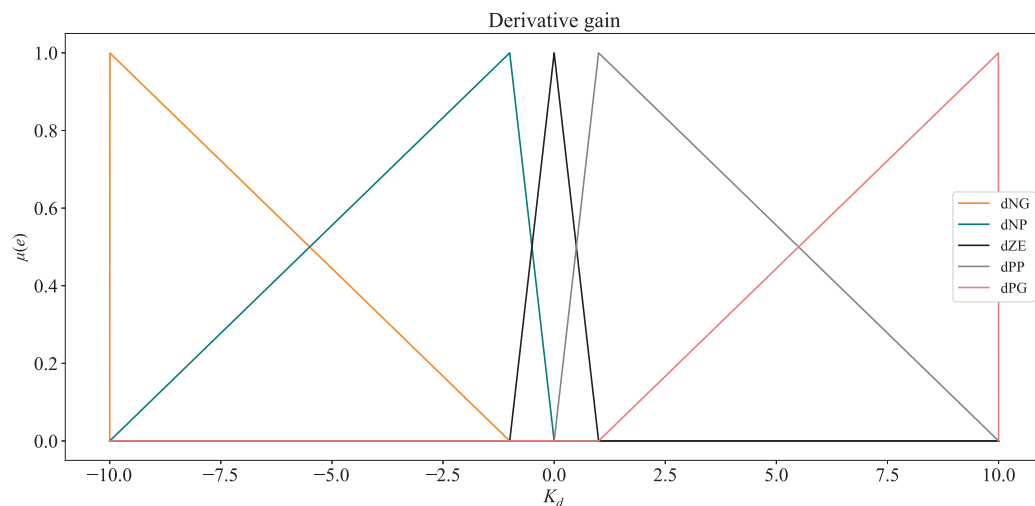
Table 3. Singleton distribution for K_p computing.

Linguistic Value	Positions P_i
B	10.0
M	4.0
S	2.5

A similar methodology determines the linguistic values for K_d and K_i . Table 4 provides a description of the linguistic values for K_d , and Figure 5 illustrates its membership functions.

Table 4. Error derivative linguistic variables.

Linguistic Value	Positions
dNL	$[-10, -10, -1]$
dNS	$[-10, -1, 0]$
dZE	$[-1, 0, 1]$
dPS	$[0, 1, 10]$
dPL	$[1, 10, 10]$

**Figure 5.** Error derivative linguistic variable distribution.

The values for the system output for the derivative gain, K_d , are shown in Table 5.

Table 5. Singleton distribution for K_d computing.

Linguistic Value	Positions P_i
dB	0.9
dM	0.6
dS	0.0

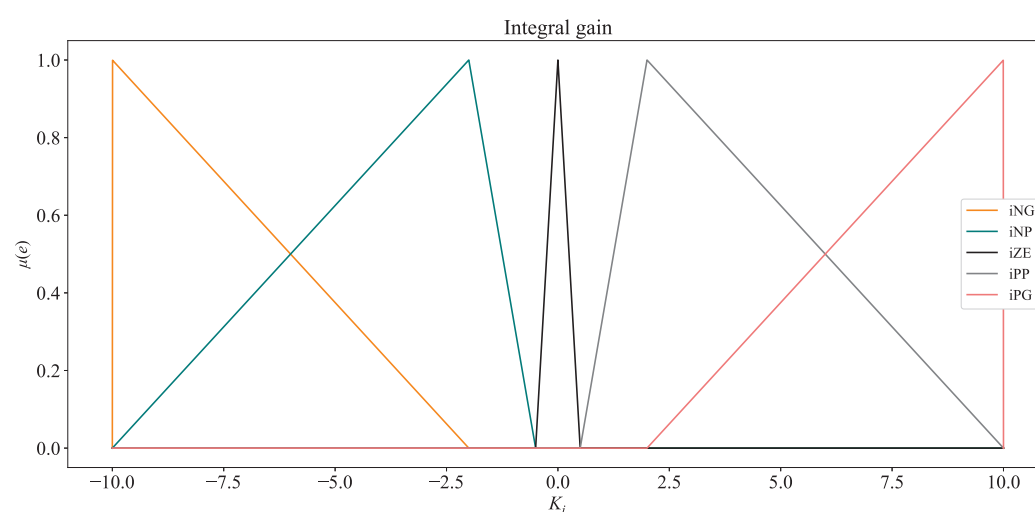
For the integral gain, K_i , the input linguistic values are displayed in Table 6, and the output values are displayed in Table 7. The membership functions are shown in Figure 6.

Table 6. Integral of error linguistic variable.

Linguistic Value	Positions
<i>iNL</i>	$[-10, -10, -2]$
<i>iNS</i>	$[-10, -2, -0.5]$
<i>iZE</i>	$[-0.5, 0, 0.5]$
<i>iPS</i>	$[0.5, 2, 10]$
<i>iPL</i>	$[2, 10, 10]$

Table 7. Singleton distribution for K_i computing.

Linguistic Value	Positions P_i
<i>iB</i>	8.0
<i>iM</i>	4.0
<i>iS</i>	0.0

**Figure 6.** Integral of error linguistic variable distribution.

A one-to-one relationship is established with the input values and five corresponding output values, allowing for direct fuzzification. Once this symmetric relationship is defined, the output of the fuzzy system is determined using the centroid method. This approach provides a smooth and gradual output, which is particularly useful in motor applications. The centroid defuzzification method is derived from Equation (24):

$$k_{gains} = \frac{\sum \mu(P_i) \cdot P_i}{\sum \mu(P_i)} \quad (24)$$

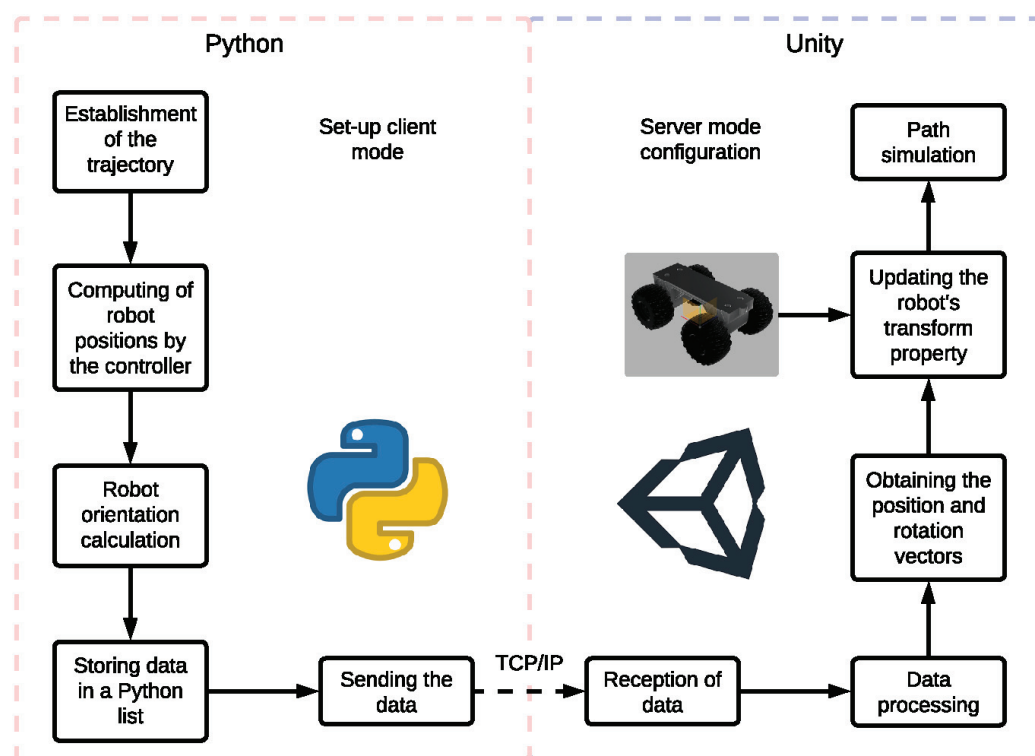
In this context, P_i denotes the position of the i -th singleton, while $\mu(P_i)$ represents a fuzzy number. The rules for the controller are outlined in Table 8. These rules allow the fuzzy PID controller to adapt to different system conditions. For large errors or rapid changes, the controller increases gains to apply stronger corrections, while under stable conditions, it reduces gains to minimize overfitting and maintain system stability. This is critical for Ackermann-type robots, where rotational constraints require precise, dynamic adjustments to the control response.

Table 8. Fuzzy rules for the mobile robot.

K_p	K_d	K_i
If $e(t)$ is NL , then K_p is B	If $de(t)$ is dNL , then K_d is dB	If $ie(t)$ is iNL , then K_i is iB
If $e(t)$ is NS , then K_p is M	If $de(t)$ is dNS , then K_d is dM	If $ie(t)$ is iNS , then K_i is iM
If $e(t)$ is ZE , then K_p is S	If $de(t)$ is dZE , then K_d is dS	If $ie(t)$ is iZE , then K_i is iS
If $e(t)$ is PS , then K_p is M	If $de(t)$ is dPS , then K_d is dM	If $ie(t)$ is iPS , then K_i is iM
If $e(t)$ is PL , then K_p is B	If $de(t)$ is dPL , then K_d is dB	If $ie(t)$ is iPL , then K_i is iB

3.3. 3D Simulation Environment Design

The controller simulation consists of two stages: first, a Python script calculates and sends the route coordinates, and second, a C# script in Unity receives these data for processing and simulation. Figure 7 illustrates the block diagram of the procedure. It is important to note that this simulation method is chosen to prevent overloading processes in Unity.

**Figure 7.** Virtual environment block diagram.

The fuzzy controller described in Section 3.2, programmed in Python 3.11.5, is implemented in the first stage. Various trajectories to be followed are proposed, and, using the controlled model, the positions of x , y , and the angle θ are obtained. As previously discussed in the kinematic model presented in Section 3.1, for non-holonomic robots, θ represents the instantaneous curvature angle generated by the robot relative to its ICC. Therefore, data are still required to perform the 3D simulation. In order to implement such data in the simulation in Unity, it is necessary to analyze the parameters from a three-dimensional point of view (x , y , z). Conventional KMs, such as the one reviewed in Section 3.1, are represented in a two-dimensional environment (x , y) and do not analyze the robot's orientation, an essential detail to be able to perform a 3D simulation. This problem can be seen reflected in Figure 8.

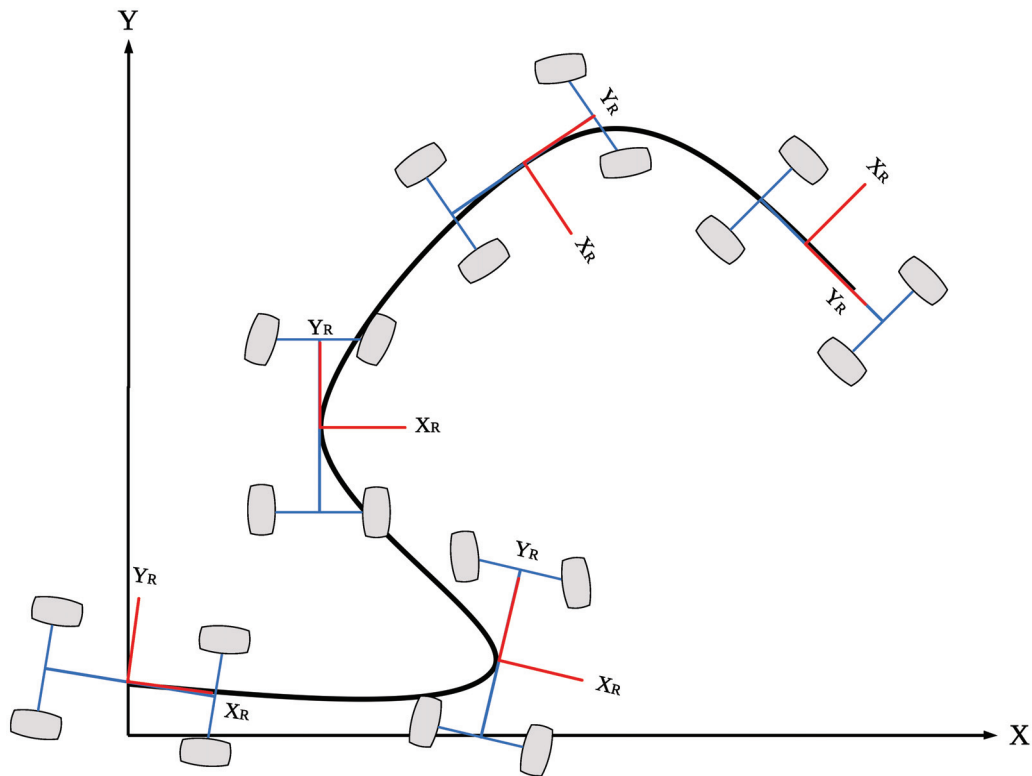


Figure 8. Determination of the yaw angle.

As illustrated in Figure 8, the steering mechanism does not consider how the object's rotation varies as it moves along the path. In 3D environments like Unity, each object has its own reference frame independent of the global frame. This local frame allows the object to not only move within the global frame (changing its x and y positions in a 2D view) but also to orient or rotate around its own axis, reflecting more complex 3D dynamics. Within Unity, these rotations correspond to Euler angles (*pitch*, *roll*, and *yaw*), which are angles also used in robotics to identify the inclinations of robots. *Yaw* is the angle responsible for determining orientation along the z -axis, representing the robot's self-rotation; this is the missing element needed to perform the 3D simulation.

From the analysis in Figure 8 and taking the orientation calculations used in SLAM (simultaneous localization and mapping) navigation systems as a reference, it can be observed that the angle *yaw* represents the rate of change of the position in the plane. Graphically, this angle corresponds to the slope of the path followed by the system. The angle *yaw* can be determined as the tangent of the angle formed between two consecutive points on the path, which allows it to be calculated using Equation (25). In it, *arctan2* is used to account for the signs of the subtractions:

$$\psi = \arctan 2 \left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right) \quad (25)$$

The Python script also calculates Equation (25). Once the method for obtaining all necessary values is defined, these values are computed and stored in a list that organizes the data in the following format $(x, 0, y, 0, \psi, 0)$. This structure is used due to the way Unity interprets positions. Subsequently, all the data stored in the list are sent sequentially by a thread through the TCP/IP protocol, establishing a local communication where the script acts as a client of the connection. The complete development of the program can be observed in Figure 9.

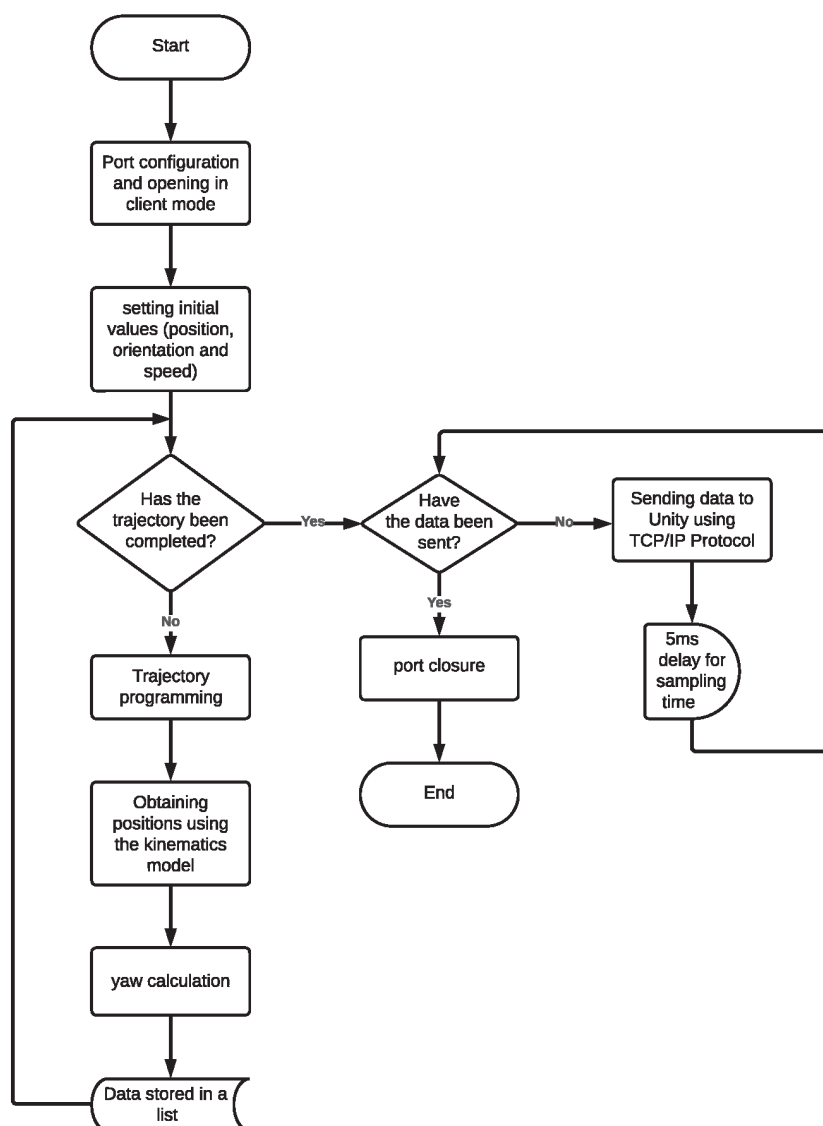


Figure 9. Flowchart of the first stage.

The second stage takes place in Unity, where the 3D simulation of the path taken by the robot is carried out. This process uses a script in C#, Unity's native language. This script receives the reception of data sent from Python via the TCP/IP protocol. Since the Python script is configured in client mode, Unity is the server that listens to all incoming connections.

Once the connection is established, the data are read sequentially. As they are received, the corresponding elements are interpreted and processed. The data arrive in the form of a text string (*string*) with the format $(x, 0, y, 0, \theta, 0)$. This is attributed to how Unity manages the positions and rotations of objects. In Unity, every object has two motion vectors: one for the position coordinates, represented as *(left/right, up/down, forward/backward)*, and one for the rotation values, expressed as *(pitch, yaw, roll)*. In this way, when receiving the data, it is only necessary to split them into two vectors: one for the position and one for the rotation. The splitting process begins by checking that a string containing precisely six comma-separated values is received. The six elements are extracted and stored in an array if this condition is met. A variable of the *Vector3* type is then created to assign the first three data points corresponding to the position. The remaining three data points, representing the rotation values, are stored in a variable of the *Quaternion* type. Before being stored, each datum is cast from string to float. Once each piece of data is processed, the robot's

position and orientation are updated in the *Update* method of the script. To trace the robot's path, within this method, each time the position is updated and it is detected that it is different from the previous one, a new *GameObject* is generated, leaving a mark on the position traveled by the robot. This process allows the complete path of the robot to be visualized during the simulation. Figure 10 shows the flow diagram corresponding to the described process.

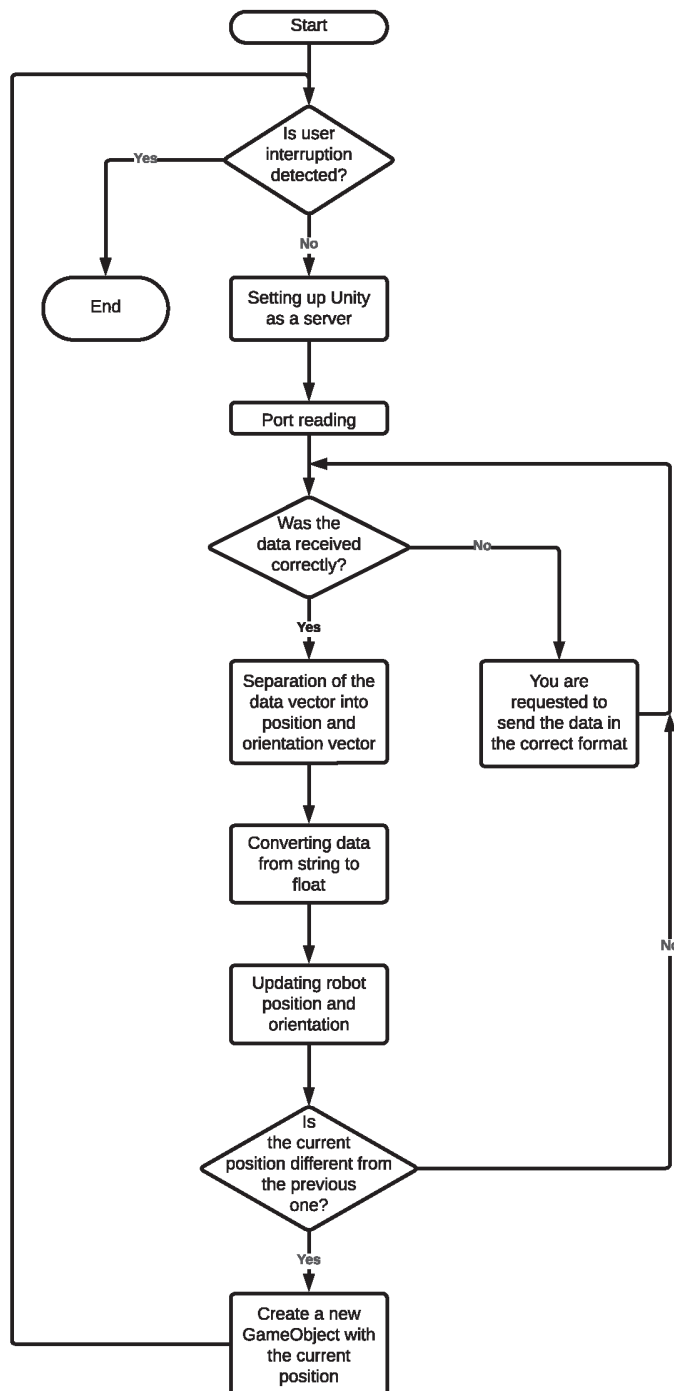


Figure 10. Flowchart of the second stage.

The simulation starts by initializing the scene in Unity, which includes opening the port and waiting for the data to be received. The Python script then calculates all the data corresponding to the robot's path. Once this calculation is completed, the data are sent

to Unity via the TCP/IP protocol. As the data are received, it is possible to visualize the robot's path in real time within the simulated environment.

To visualize the objects in the 3D environment, a scene is designed to allow the robot's path to be observed. This scene includes a base terrain, the *GameObject* representing the robot, and other fundamental elements provided by the Unity environment to complete the simulation. The design of the scene ensures a clear and understandable representation of the path. The layout of this environment can be seen in Figure 11.

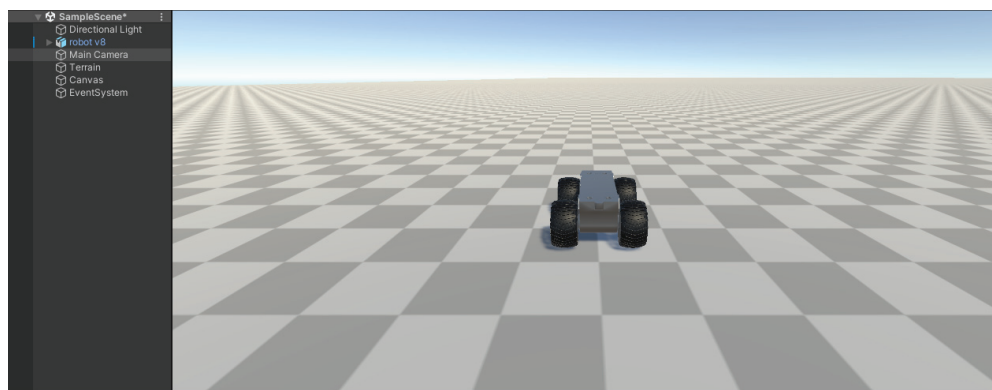


Figure 11. Virtual environment scene.

The 3D model of the robot was designed using Fusion 360 software, with dimensions of $174 \times 73 \times 126$ mm. This model was then scaled within Unity to fit the requirements of each test. The model was exported in *obj* format. It was important to ensure that it was oriented according to the Unity coordinate system, where the *y*-axis represents the height, as opposed to classic formats where this corresponds to the *z*-axis. Likewise, the reference system of the model had to be centered on the robot to ensure that the rotations were accurate and corresponded to those expected. Figure 12 shows the 3D design of the robot used in the simulation. Table 9 depicts the geometry of the 3D model of the robot.

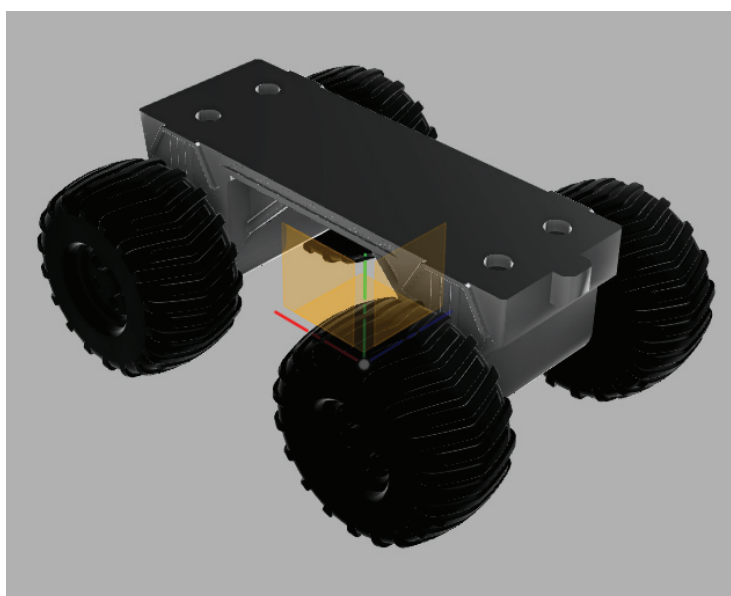


Figure 12. A 3D model of the robot in Fusion 360.

Table 9. Characteristics of the 3D model of the robot.

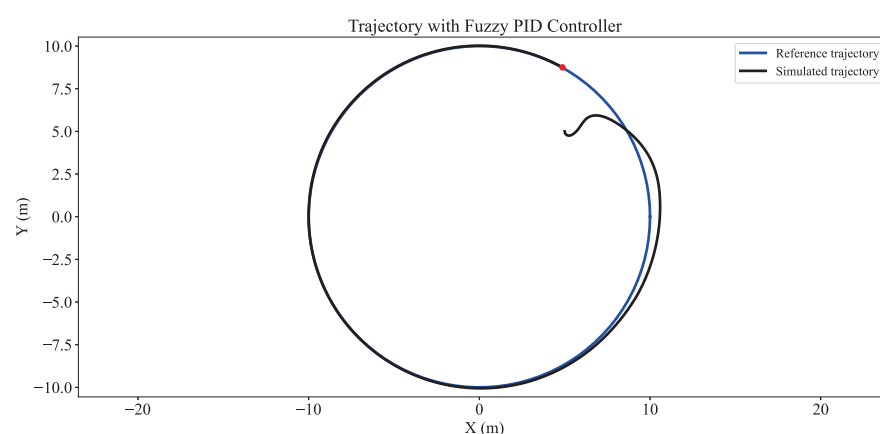
Description	Measures
Width	73 mm
Long	174 mm
Height	12.6 mm
Distance between the axles	107.6 mm
Wheel separation	91.75 mm
Wheel radius	33.34 mm

4. Results

Three different path-tracking scenarios are analyzed. In the first scenario, the following circular reference path is considered, which allows for the evaluation of the system's ability to maintain a smooth and continuous movement. In the second scenario, the tracking of discrete points distributed on a map is proposed, simulating trajectories composed of specific positions to be reached. Finally, in the third scenario, the following of crossed paths is addressed, which implies more abrupt changes in direction and greater complexity in control. In each of these cases, different initial configurations and variations in the dimensions of the robot are analyzed, with the aim of evaluating the performance of the system under different operating conditions. An ideal surface is always assumed for the simulation process, which implies that rotations in *pitch* and *roll* are not considered.

4.1. Circular Trajectory

For this test, a constant linear speed of 2 m/s and a spacing between axes of 0.25 m are used. The proposed path is circular with a radius of 10 m. Initially, the robot is located at the coordinates (5,5) as a reference point, oriented at an angle of $-\frac{\pi}{2}$ rad. From these initial parameters, the successive positions of the robot are determined during the path tracking. Figure 13 illustrates both the proposed reference path and the one followed by the robot, highlighting its performance in the simulation.

**Figure 13.** First test: simulate circular path in Python.

The controller identifies the initial position of the robot and, based on the proposed restrictions, begins the movement to correct its position continuously. Figure 14 shows how the error evolves as the robot moves. In this case, it can be seen that the error never reaches zero, which is consistent with the nature of the circular path. The angle δ must remain constant to maintain the turn. If the angle δ were equal to zero, it would imply that the robot would follow a straight-line path, which would be incompatible with the proposed circular movement.

Figure 15 illustrates the applied control signal, which shows its correlation with the error. The signal is consistently directed to maintain the desired trajectory. The data presented in the graph are confined to angles of $\frac{\pi}{4}$ and $-\frac{\pi}{4}$ rad due to the constraints established in the model.

Figure 16, shows the simulation of the path in the Unity environment. In these images, it can be observed how the positions generated along the route were marked, showing a high coincidence with the trajectory previously obtained in Python. This correspondence between both simulations validates the accuracy of the implemented model.

It is also noteworthy that the orientation of the robot in Unity is precisely aligned with the direction of the path, which reflects the coherent and realistic behavior of the control system. This detail allows us to appreciate not only the movement of the robot but also its ability to dynamically adapt to the expected path. The combination of positional accuracy and correct orientation demonstrates that the simulation process offers a faithful and visually realistic representation of the robot's performance in the virtual environment.

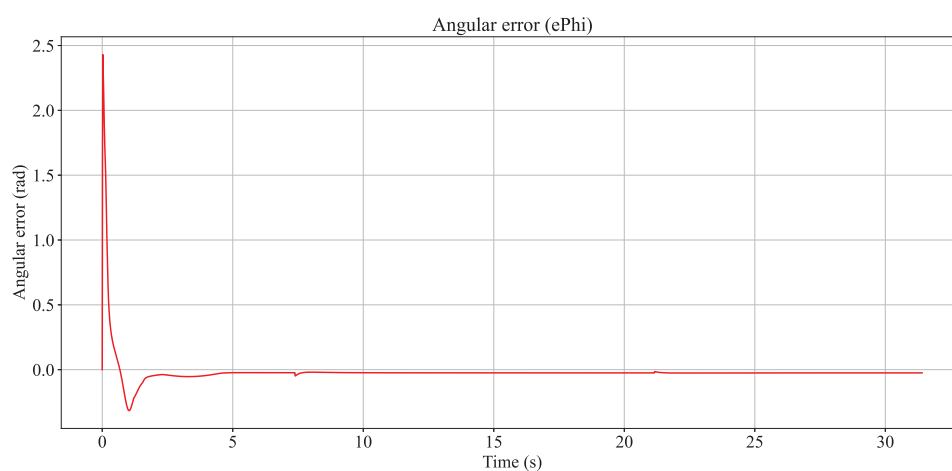


Figure 14. Angular error for circular path.

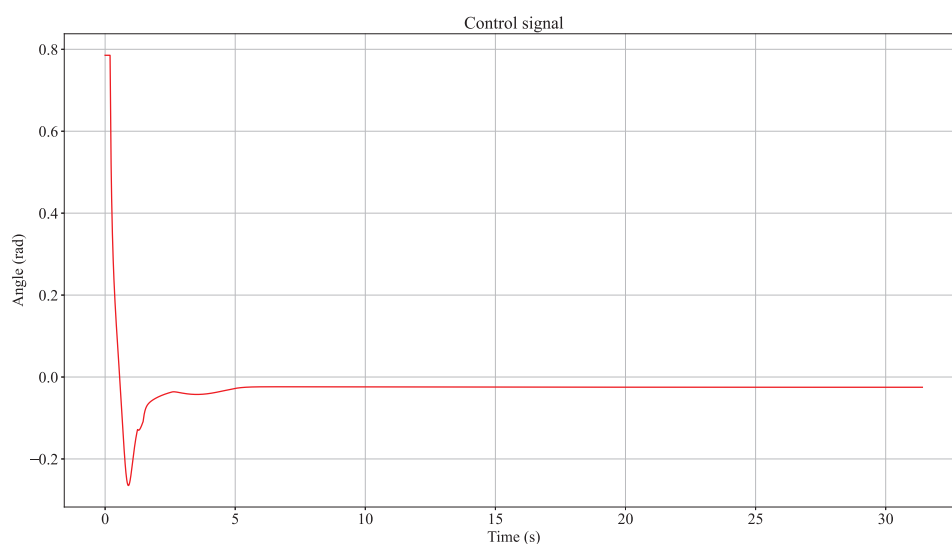


Figure 15. Circular path control signal

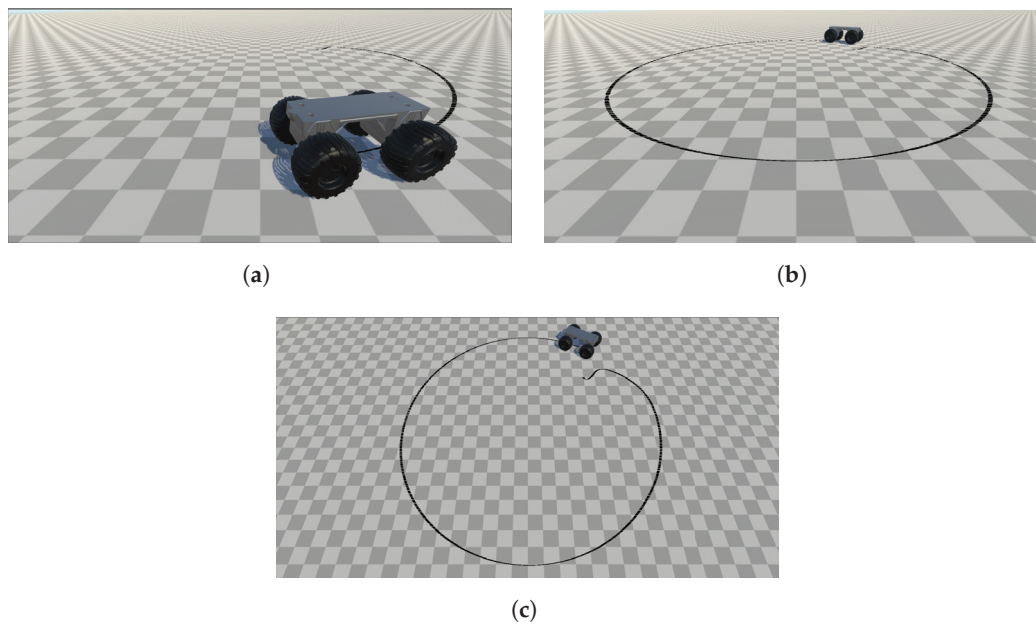


Figure 16. Simulation of the circular trajectory made in Unity. (a) View of the robot orientation for the path. (b) Front view of the circular path simulation. (c) Aerial view of the circular path simulation.

4.2. Point-to-Point Trajectory

In this second case, multiple points distributed along the map were defined, and straight lines were drawn from them to connect the points and form the desired path. This test used a constant linear speed of 2 m/s and a separation between axes of 0.5 m. The robot was initially positioned at the coordinates (2, 1) with an orientation of $\frac{3\pi}{5}$ rad.

The result obtained in this simulation is shown in Figure 17, where the tracking performed by the robot can be observed. This analysis allows it to evaluate the capacity of the system to maintain a precise and continuous movement between the points of the path, considering the geometric and dynamic restrictions established for the model.

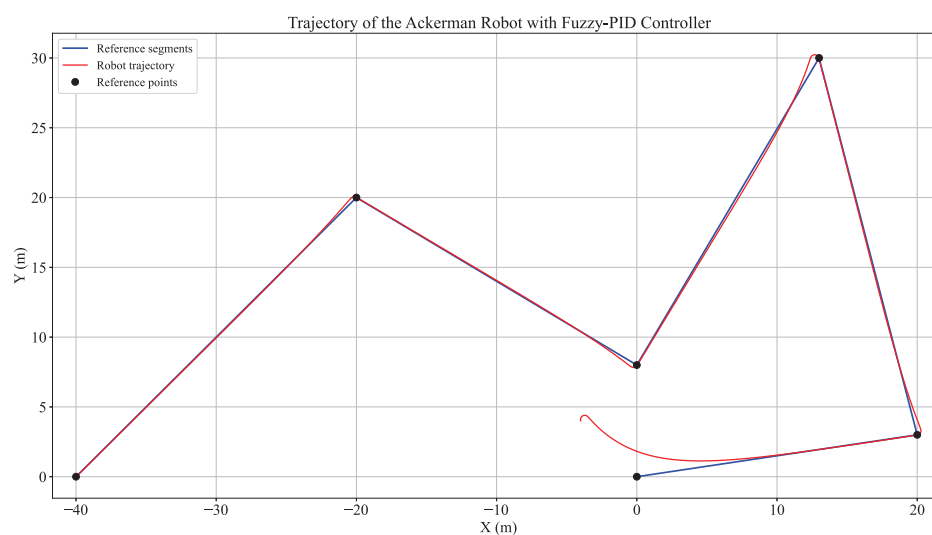


Figure 17. Second trajectory test: point-to-point case.

Figure 18 shows the error graph corresponding to the test performed in this section. The changes that occur when the robot reaches one of the objectives and generates a zero error can be observed. As mentioned above, this behavior indicates that the robot continues moving in a straight line after correcting its trajectory.

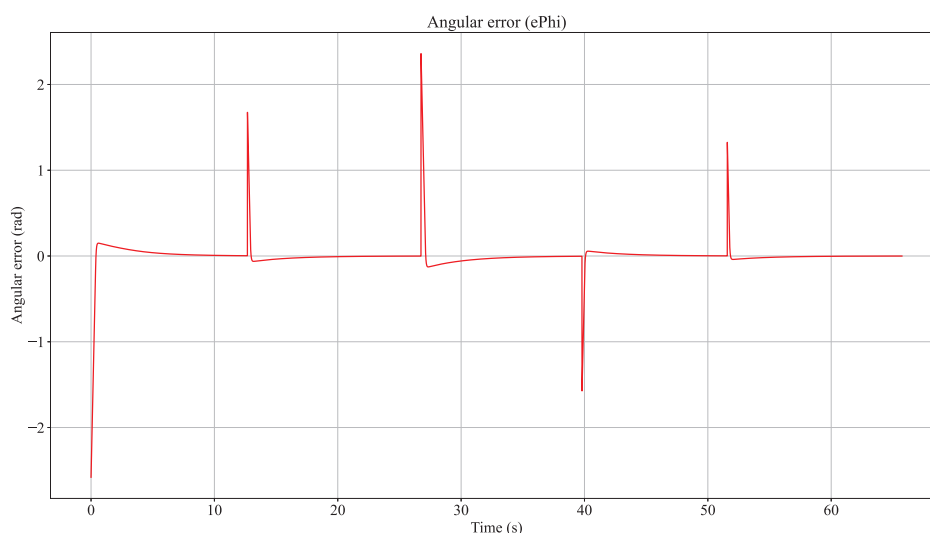


Figure 18. Angular error for point tracking path.

Figure 19 presents the control signal graph for the current case, highlighting the overshoots that occur when reaching specific points along the path. These overshoots are inherent consequences of the kinematic configuration of the Ackermann-type robot, requiring turning maneuvers at reference points before continuing its movement. Since the Ackermann robot has a geometry with a single steering axis, when reaching a turning point, an adjustment in the steering angle is necessary to compensate for the path error, which generates a small oscillation or overshoot in the control signal. In addition, a fast and efficient response to the control action is observed, suggesting that the system possesses a high capacity for making quick adjustments in the robot's direction. This responsiveness is critical for minimizing stabilization time after overshoots and ensuring that the robot returns to its desired path as accurately as possible.

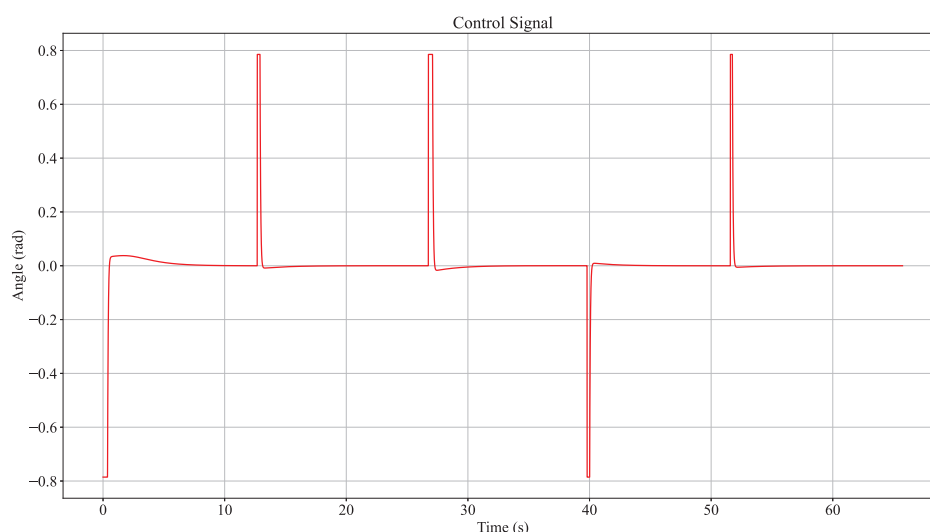


Figure 19. Control signal for point-to-point trajectory.

The simulation in Unity is displayed in Figure 20, where it can be seen how the robot's path was carried out.

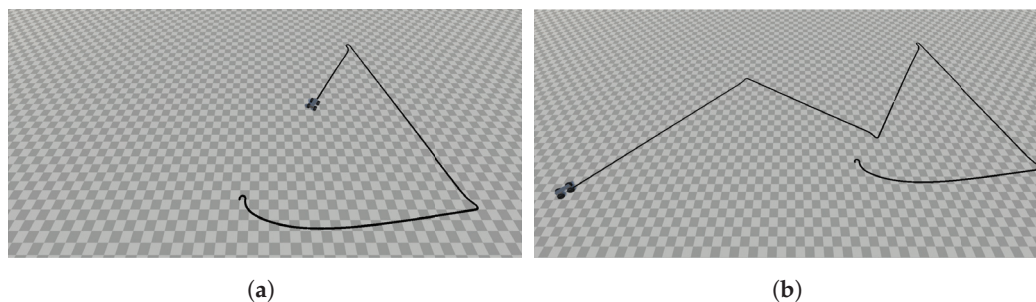


Figure 20. Point tracking path simulation done in Unity. (a) Partial path of the simulated robot in Unity. (b) Complete robot path in Unity.

4.3. Crossed Trajectory

Finally, a case where the trajectory crosses the same area is proposed; the test configurations are as follows: linear velocity of 3 m/s, a distance between the axes of 0.1 m, an initial position at coordinates (8,3.5), and an initial orientation of 0° rad. The results of this test can be observed in Figure 21, demonstrating how the robot quickly integrates into the trajectory and follows it almost perfectly.

Figure 22 shows the error graph, where it can be observed that the error is remarkably small. This behavior is mainly attributed to the geometry of the trajectory, which includes sections with linear segments. The controller effectively manages these straight sections, allowing the error to remain low at all times. The controller's ability to adjust the orientation and maintain precision during linear sections contributes to the stability of the system and minimizes the necessary corrections. On the other hand, in sections where the trajectory presents greater complexity, the error tends to increase slightly. However, it remains low, highlighting the controller's efficiency in maintaining performance under various geometric conditions.

Figure 23 presents the control signal generated for the third evaluated trajectory. Given the geometry of this route, it can be observed that the required control actions are significantly reduced, especially in the sections characterized by sharper curves. This is due to the inherent smoothness of the trajectory, which limits the need for constant adjustments in the robot's orientation. Additionally, the control signal shows high stability in most sections, indicating efficient performance of the control system by minimizing oscillations and abrupt corrections even in scenarios with geometric variations.

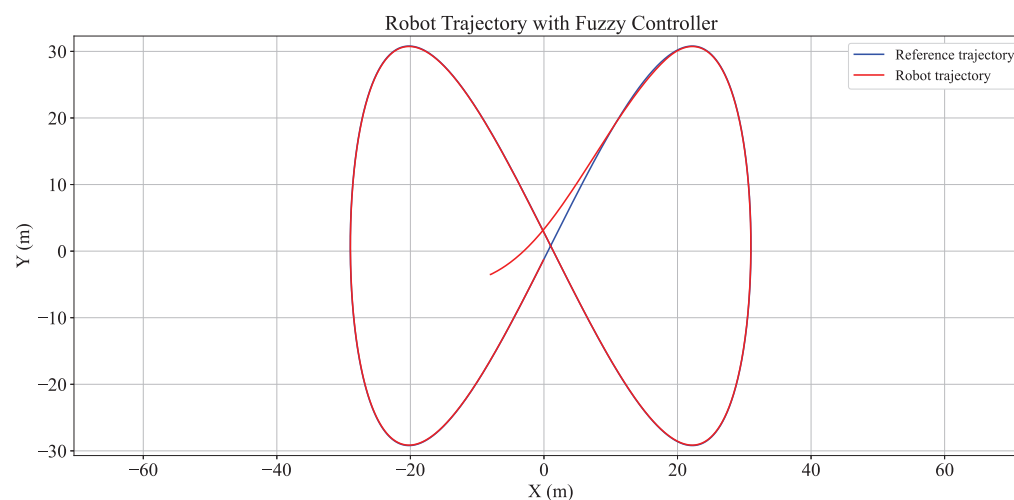


Figure 21. Third case of trajectories: movement always in change.

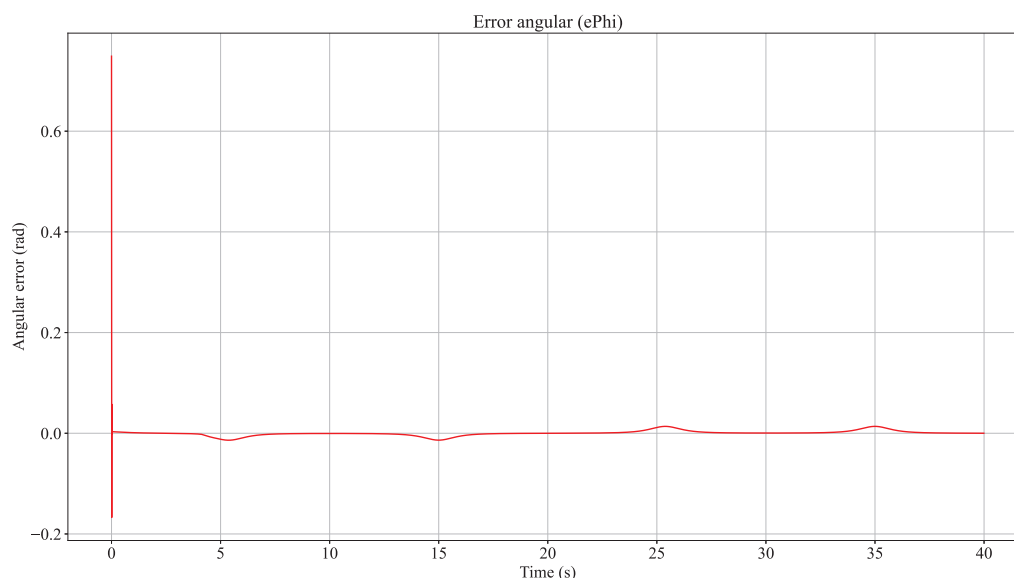


Figure 22. Angular error of rotation δ for the crossed trajectory.

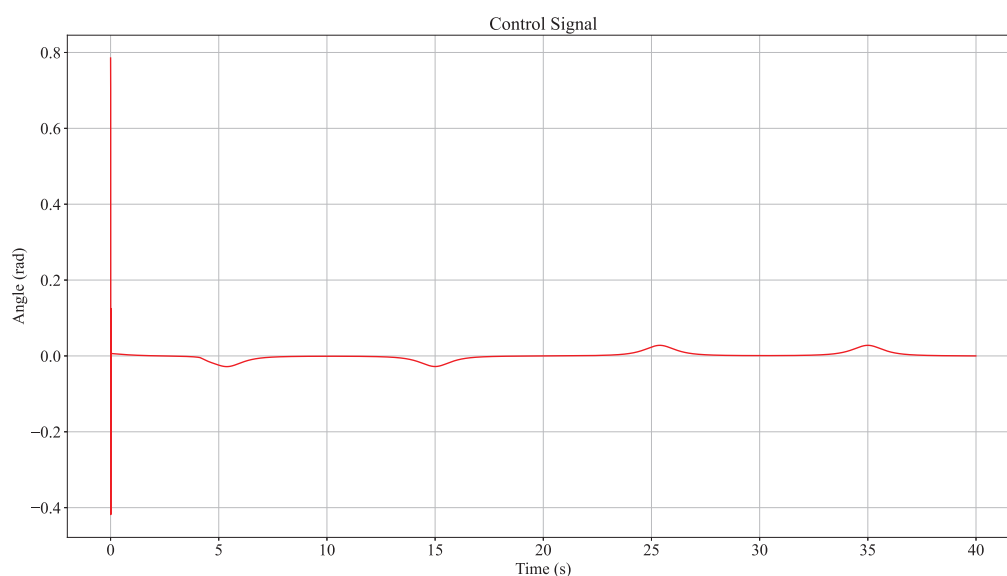


Figure 23. Control signal for the cross path.

Finally, Figure 24 shows the simulation of the last trajectory, which runs correctly.

The simulation was carried out on a laptop equipped with an Intel Core i3-1115G4 processor with a frequency of 3.00 GHz, 12 GB of RAM, and integrated Intel(R) UHD graphics. Different load levels were recorded on the system components during the simulation process. On average, the CPU operated at 75% of its capacity, while the RAM remained at 50% usage. On the other hand, the GPU was at 100% for most of the simulation, although in the final stage, its consumption decreased to approximately 80%.

In the second case, a significantly larger volume of data was generated for processing due to a higher sampling time, considerably impacting system performance. In this scenario, the processor constantly worked at 100% of its capacity, the RAM reached an average usage of 85%, and the GPU remained operating at 100% throughout the simulation time. This behavior shows the increasing computational demands associated with more intensive data processing and highlights the importance of optimizing resources for more complex scenarios.

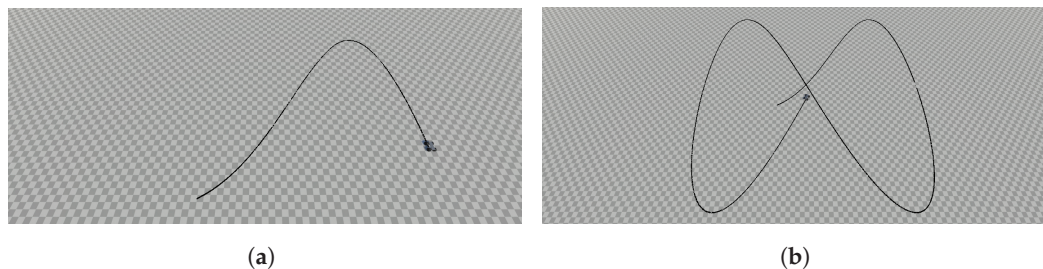


Figure 24. Simulation of the crossing trajectory implemented in Unity. **(a)** Partial robot path of the crossed trajectory. **(b)** Full robot path of the cross trajectory.

5. Discussion

Table 10 shows the trajectory tracking metrics of the mobile robot controlled using a fuzzy PID. The trajectories analyzed include a circular pattern, a point-to-point displacement, and a cross path. Each of these trajectories presents different control challenges, allowing the robustness and adaptability of the controller to be assessed. The metrics considered include the average tracking error, maximum error, steady-state error, settling time, average orientation error in rad, total time, and total distance traveled. These indicators provide a comprehensive view of the system's performance in various scenarios. Also, it is important to note that the average tracking error, the steady-state error, and the average orientation error in rad are measured for times greater than the settling time in order to validate the controller's performance. The maximum error tends to be large because the error exists from the robot's position to the reference at $t = 0$.

For the circular path, the controller demonstrates highly accurate performance. The average tracking error is 0.0089 m, while the maximum error reaches 3.7 m, which is attributed to the robot starting at a point outside the path. This initial deviation highlights the controller's ability to correct significant initial errors and quickly converge toward the desired reference. The steady-state error is only 0.0012 m, demonstrating the system's ability to stabilize near the desired reference once initial perturbations are mitigated. The settling time of 7.246 s reflects the period required for the robot to stabilize within acceptable error margins, demonstrating the controller's robustness under adverse initial conditions. The average orientation error is 0.015 rad, highlighting the effectiveness of the steering angle control in maintaining proper orientation throughout the path. Finally, the total distance traveled is 62.8 m, with a total time of 32.5 s—values that agree with the perimeter expected for a circular path of 10 m radius, thus validating the overall accuracy of the system.

On the point-to-point trajectory, the results show significantly different behavior. The average tracking error is considerably higher, at 0.9531 m, and the maximum error reaches 4.56 m. These deviations are mainly due to the robot's need to correct its initial position outside the trajectory, demonstrating the controller's ability to handle abrupt transitions and correct significant errors. Despite these initial deviations, the steady-state error is low, at 0.001543 m, indicating that the robot is able to arrive with high precision at the endpoint. The settling time is 5.784 s, reflecting how quickly the robot stabilizes its trajectory once the initial errors are corrected. The average orientation error is the largest among the analyzed trajectories, at 0.9743 rad, indicating that the steering angle control faces greater challenges in direct displacements. The total distance traveled is 75.4 m, and the total time taken is 67 s, reflecting the length of the straight path plus the necessary corrections.

The cross-track path exhibits intermediate performance in terms of accuracy and stability. The average tracking error is 0.01814 m, and the maximum error is 8.96 m, which are significant deviations attributable to the complexity of the path and the robot's initial position outside the reference. However, the controller shows robustness by reducing the steady-state error to 0.00232 m, evidencing its ability to correct complex deviations. The

settling time is 9.940 s, indicating the time needed to stabilize after overcoming the abrupt changes in direction that are characteristic of this path. The average orientation error is 0.021 rad, suggesting effective control of the steering angle despite the complexity of the path. The total distance traveled is 58.3 m, and the total time is 40 s, which reflects efficiency in navigating complex paths.

Overall, these findings highlight the effectiveness and reliability of the fuzzy PID controller in tracking various trajectories. Despite starting at points outside the trajectory, the controller is able to correct significant initial errors, reach the desired reference with high accuracy, and maintain stable tracking under diverse conditions. The variation in settling time and maximum errors highlights the system's ability to adapt to control challenges, validating its applicability in dynamic and complex scenarios.

Table 10. Trajectory tracking metrics.

Trajectory	Average Tracking Error (m)	Maximum Error (m)	Steady-State Error (m)	Settling Time (s)	Average δ Error (rad)	Total Time (s)	Total Distance Traveled (m)
<i>Circular</i>	0.0089	3.7	0.0012	7.246	0.015	32.5	62.8
<i>Point-to-point</i>	0.9531	5.784	0.001543	4.564	0.9743	67	75.4
<i>Cross</i>	0.01814	8.96	0.00232	9.940	0.021	40	58.3

A FL-based kinematic controller designed to guide a mobile robot in a virtual environment shares fundamental principles with control systems used in autonomous vehicles. When comparing this technique with others, such as the one presented in [26], where a comparison is made between a traditional PID and an H_∞ -enhanced PID controller based on robust adaptive learning, the work mainly focuses on accuracy when giving a curve using the given references. In the results of the traditional PID, a turn with a minimum radius of 3 m was obtained, but it was necessary to place more reference points to be able to re-enter the trajectory. When using the H_∞ improvement, they obtained precise trajectory control, resulting in a radius of 2.9 m when turning, which improved stability and accuracy. Despite these factors, these controllers present their details. For the traditional PID, the disadvantages were previously discussed: it requires a well-identified plant and performs inaccurately in the presence of non-linearities; in addition, when uncertainties arise, its performance is not better. As for the adaptive controller based on H_∞ , although it offers improvements in stability and precision, it entails a more exhaustive mathematical analysis, which increases the computational cost. This greater complexity can represent a limitation in applications where computational resources are restricted or computation time efficiency is critical for its implementation. The work presented in [37] proposed a control scheme based on adaptive dynamic programming combined with a critical neural network. The results demonstrated accurate responses of the controller, achieving reduced errors with RMSE values of 0.21 m in e_x , 0.07 m in e_y , and 0.01 rad in e_δ . However, the authors themselves recognized several disadvantages associated with this methodology. Adaptive dynamic programming is a technique that has still not been explored, which limits the availability of references and complementary studies in the literature. In addition, using a critical neural network significantly increases the computational cost, which represents a challenge for real-time applications. The network training process also requires considerable computational resources, which forces the use of high-performance equipment to carry out this task efficiently. As a final case, the work proposed by [38] presented path-tracking algorithms based on predictive control models, focusing on execution times and tracking accuracy. The three algorithms developed are conflict-based search (CBS), spatiotemporal hybrid A* conflict-based search (STH-CBS), and weighted spatiotemporal hybrid A*

conflict-based search (WSTH-CBS), achieving success rates of 85.71%, 100%, and 100%, respectively, in their best tests. Despite these promising results, an important limitation of this technique lies in the mathematical complexity inherent in predictive controls, which are susceptible to modeling errors. This can affect the robustness of the system in scenarios where the dynamics of the environment or the system itself are not completely defined or present uncertainties.

The ability of FL to handle uncertainty and imprecision, inherent characteristics of real environments, makes it a valuable tool for precise trajectory tracking. Furthermore, this technique is relatively easy for any designer to implement, as it allows for omitting complex mathematical analyses that are indispensable in other methodologies. Unlike learning-based approaches, it does not require training processes for its operation, representing a significant advantage in simplicity and development times. This feature also contributes to reducing computational costs, facilitating its implementation on devices with low processing capacity and making it a viable option for real-time applications or in environments with limited resources. This technology can be applied to various systems in urban mobility, from adaptive cruise control to fully autonomous driving. By enabling smoother and more adaptive decision-making, fuzzy controllers contribute to safer and more efficient driving, especially in complex and dynamic environments.

6. Conclusions

This paper presents a significant advancement in mobile robotics using a kinematic controller based on FL for path following. The fuzzy controller overcomes the limitations of traditional PID controllers, especially in dynamic and uncertain environments, offering greater adaptability and precision. Simulations carried out within the virtual environment developed in Unity demonstrated the effectiveness of the control system, recording an average error of 0.0089 m in circular trajectories, 0.01814 m in cross trajectories, and 0.9531 m in point-to-point trajectories, where the most significant error is associated with the system's difficulties in maintaining precision during sudden changes in direction. However, it manages to correct these deviations with a low steady-state error. A key contribution of this work is the controller's adaptability, which is crucial for autonomous mobile robots navigating dynamic and unpredictable environments. This controller not only emulates the responses of a classic PID but also improves the flexibility and performance of the system, consolidating itself as a robust alternative. In addition, using Unity facilitates real-time visualization and parameter adjustments, reducing physical testing risks and costs and speeding up robotic development. Practical implications include applications such as autonomous vehicles and robotic exploration, where precision and adaptability are crucial. In the future, incorporating multi-sensory feedback and hybrid strategies with machine learning could enhance their capabilities, improving interpretation and response to complex environments.

Author Contributions: Conceptualization, J.G.P.-J. and J.R.G.-M.; methodology, J.G.P.-J., A.M.S. and J.R.G.-M.; software, J.G.P.-J. and J.R.G.-M.; validation, A.M.S., O.A.B.-V. and E.E.C.-M.; formal analysis, J.R.G.-M. and A.M.S.; investigation, J.R.G.-M. and L.F.O.-G.; resources, E.E.C.-M., O.A.B.-V. and M.A.R.-H.; data curation, A.M.S., O.A.B.-V., J.R.G.-M., L.F.O.-G. and M.A.R.-H.; writing—original draft preparation, J.R.G.-M., A.M.S., J.G.P.-J. and L.F.O.-G.; writing—review and editing, O.A.B.-V., A.M.S., E.E.C.-M. and J.R.G.-M.; visualization, J.G.P.-J. and M.A.R.-H.; supervision, J.R.G.-M. and A.M.S.; project administration, J.R.G.-M. and A.M.S.; funding acquisition, J.R.G.-M., L.F.O.-G., A.M.S., E.E.C.-M., O.A.B.-V. and M.A.R.-H. All authors have read and agreed to the published version of the manuscript.

Funding: SECIHTI funded this research under the scholarship 2006985.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The first author thanks SECIHTI for the scholarship awarded for being a student in the Master’s Degree program in Engineering Sciences at the Faculty of Mechanical and Electrical Engineering, Universidad Veracruzana, Poza Rica—Tuxpan Region. Grammarly was used to enhance the paragraphs in this work.

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

Abbreviations

The following abbreviations are used in this manuscript:

PID	proportional–integral–derivative
FL	fuzzy logic
FLC	fuzzy-logic-based controller
TVLQ	time-varying linear quadratic
LMIs	linear matrix inequalities
ADAR	analytical design of aggregated regulators
SCT	synergetic control theory
ORRL	optimized reward reinforcement learning
DQN	deep Q-network
ROS	robot operating system
LSTM	long short-term memory
UGV	unmanned ground vehicle
BIM	building information modeling
VR	virtual reality
ICC	instantaneous center of curvature
KM	kinematic model
DKM	direct kinematic model
NB	negative-large
NS	negative-small
ZE	zero
PS	positive-small
PL	positive-large
B	big
M	medium
S	small
CBS	conflict-based search
STH-CBS	spatiotemporal hybrid A* conflict-based search
WSTH-CBS	weighted spatiotemporal hybrid A* conflict-based search

References

1. Atchade-Adelomou, P.; Alonso-Linaje, G.; Albo-Canals, J.; Casado-Fauli, D. qRobot: A Quantum Computing Approach in Mobile Robot Order Picking and Batching Problem Solver Optimization. *Algorithms* **2021**, *14*, 194. [CrossRef]
2. Amer, N.H.; Zamzuri, H.; Hudha, K.; Kadir, Z.A. Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges. *Intell. Robot. Syst.* **2017**, *86*, 225–254. [CrossRef]
3. Zhang, X.; Xie, B.; Yang, Y.; Liu, Y.; Jiang, P. Research on Running Performance Optimization of Four-Wheel-Driving Ackerman Chassis by the Combining Method of Quantitative Experiment with Dynamic Simulation. *Machines* **2024**, *12*, 198. [CrossRef]
4. Cao, Y.; Li, B.; Deng, Z.; Guo, X. Parking Trajectory Planning for Autonomous Vehicles Under Narrow Terminal Constraints. *Electronics* **2024**, *13*, 5041. [CrossRef]
5. Kang, Y.; Xue, B.; Zeng, R. Self-Adaptive Path Tracking Control for Mobile Robots under Slippage Conditions Based on an RBF Neural Network. *Algorithms* **2021**, *14*, 196. [CrossRef]
6. Ju, J.; Zhao, Y.; Zhang, C.; Liu, Y. Vibration Suppression of a Flexible-Joint Robot Based on Parameter Identification and Fuzzy PID Control. *Algorithms* **2018**, *11*, 189. [CrossRef]

7. Grosset, J.; Oukacha, O.; Fougères, A.-J.; Djoko-Kouam, M.; Bonnin, J.-M. Fuzzy Multi-Agent Simulation for Collective Energy Management of Autonomous Industrial Vehicle Fleets. *Algorithms* **2024**, *17*, 484. [CrossRef]
8. Hussein, A.; García, F.; Olaverri-Monreal, C. ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, 12–14 September 2018; pp. 1–6.
9. Eisenträger, K.; Haubner, J.; Brade, J.; Einhäuser, W.; Bendixen, A.; Winkler, S.; Klimant, P.; Jahn, G. Evaluating the Effects of Virtual Reality Environment Learning on Subsequent Robot Teleoperation in an Unfamiliar Building. *IEEE Trans. Vis. Comput. Graph.* **2023**, *29*, 2220–2229. [CrossRef] [PubMed]
10. Haces-Garcia, A.; Zhu, W. Building an Accessible and Flexible Multi-User Robotic Simulation Framework with Unity-MATLAB Bridge. *Computers* **2024**, *13*, 282. [CrossRef]
11. Pham, T.T.; Le, M.T.; Nguyen, C.N. Omnidirectional Mobile Robot Trajectory Tracking Control with Diversity of Inputs. *Int. J. Mech. Eng. Robot. Res.* **2021**, *10*, 639–644. [CrossRef]
12. Carmona, R.R.; Sung, H.G.; Kim, Y.S.; Vazquez, H.A. Stable PID Control for Mobile Robots. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; IEEE: New York, NY, USA, 2018; pp. 1891–1896.
13. Shojaei, K.; Shahri, A.M.; Tarakameh, A.; Tabibian, B. Adaptive Trajectory Tracking Control of a Differential Drive Wheeled Mobile Robot. *Robotica* **2011**, *29*, 391–402. [CrossRef]
14. Čaran, B.; Milić, V.; Švaco, M.; Jerbić, B. Optimal Control-Based Algorithm Design and Application for Trajectory Tracking of a Mobile Robot with Four Independently Steered and Four Independently Actuated Wheels. *Actuators* **2024**, *13*, 279. [CrossRef]
15. Veselov, G.; Sklyarov, A.; Chávez, J.V. Non-linear Control of a Tracked Robot. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; pp. 641–646.
16. Xie, H.; Ma, X.; Qin, Q.; Sun, X. Trajectory Tracking Control for Ackerman Vehicle Based on Improved Reward Function. In Proceedings of the 2024 43rd Chinese Control Conference (CCC), Kunming, China, 26–28 July 2024; pp. 519–524.
17. Ha, T.N.; Tu, T.N.; Dung, N.T.; Mien, T.L.; Thuy, C.T. Deep Q-Network (DQN) Approach for Automatic Vehicles Applied in the Intelligent Transportation System (ITS). In Proceedings of the 2023 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh, Vietnam, 27–28 July 2023; pp. 527–532.
18. Khesrani, S.; Hassam, A.; Boutalbi, O.; Boubezoula, M. Motion Planning and Control of Nonholonomic Mobile Robot Using Flatness and Fuzzy Logic Concepts. *Int. J. Dyn. Control* **2021**, *9*, 1660–1671. [CrossRef]
19. Thuong, T.T.; Ha, V.T.; Truc, L.N. Intelligent Control for Mobile Robots Based on Fuzzy Logic Controller. In Proceedings of the International Conference on Intelligent Systems & Networks, Hanoi, Vietnam, 18–19 March 2023; Springer Nature: Singapore, 2023; pp. 566–573.
20. Mai, T.A.; Dang, T.S.; Duong, D.T.; Le, V.C.; Banerjee, S. A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots. *J. Braz. Soc. Mech. Sci. Eng.* **2021**, *43*, 1–13. [CrossRef]
21. Tolossa, T.D.; Gunasekaran, M.; Halder, K.; Verma, H.K.; Parswal, S.S.; Jorwal, N.; Maria Joseph, F.O.; Hote, Y.V. Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters. *Robotica* **2024**, *42*, 2801–2824. [CrossRef]
22. Zhao, T.; Qin, P.; Zhong, Y. Trajectory Tracking Control Method for Omnidirectional Mobile Robot Based on Self-Organizing Fuzzy Neural Network and Preview Strategy. *Entropy* **2023**, *25*, 248. [CrossRef]
23. Tang, H.H.; Ahmad, N.S. Fuzzy logic approach for controlling uncertain and nonlinear systems: A comprehensive review of applications and advances. *Syst. Sci. Control Eng.* **2024**, *12*, 2394429. [CrossRef]
24. Hentout, A.; Maoudj, A.; Aouache, M. A review of the literature on fuzzy-logic approaches for collision-free path planning of manipulator robots. *Artif. Intell. Rev.* **2023**, *56*, 3369–3444. [CrossRef]
25. Bai, G.; Liu, L.; Meng, Y.; Luo, W.; Gu, Q.; Wang, J. Path Tracking of Wheeled Mobile Robots Based on Dynamic Prediction Model. *IEEE Access* **2019**, *7*, 39690–39701. [CrossRef]
26. Shamshiri, R.R.; Azimi, A.; Behjati, M.; Ghasemzadeh, A.; Dworak, V.; Weltzien, C.; Karydis, K.; Cheein, F.A.A. Online path tracking with an integrated H_∞ robust adaptive controller for a double-Ackermann steering robot for orchard waypoint navigation. In *International Journal of Intelligent Robotics and Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–21.
27. Molina-Leal, A.; Gómez-Espinosa, A.; Escobedo Cabello, J.A.; Cuan-Urquiza, E.; Cruz-Ramírez, S.R. Trajectory Planning for a Mobile Robot in a Dynamic Environment Using an LSTM Neural Network. *Appl. Sci.* **2021**, *11*, 10689. [CrossRef]
28. Zhang, Q.; Ma, W.; Zheng, Q.; Zhai, X.; Zhang, W.; Zhang, T.; Wang, S. Path Planning of Mobile Robot in Dynamic Obstacle Avoidance Environment Based on Deep Reinforcement Learning. *IEEE Access* **2024**, *12*, 189136–189152. [CrossRef]
29. Chen, Z.; Chen, K.; Song, C.; Zhang, X.; Cheng, J.C.P.; Li, D. Global Path Planning Based on BIM and Physics Engine for UGVs in Indoor Environments. *Autom. Constr.* **2022**, *139*, 104263. [CrossRef]
30. Liu, Y.; Novotny, G.; Smirnov, N.; Morales-Alvarez, W.; Olaverri-Monreal, C. Mobile Delivery Robots: Mixed Reality-Based Simulation Relying on ROS and Unity 3D. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19–21 October 2020; pp. 15–20.

31. Pantusin, F.J.; Carvajal, C.P.; Ortiz, J.S.; Andaluz, V.H. Virtual Teleoperation System for Mobile Manipulator Robots Focused on Object Transport and Manipulation. *Technologies* **2024**, *12*, 146. [CrossRef]
32. Galarza, B.R.; Ayala, P.; Manzano, S.; Garcia, M.V. Virtual Reality Teleoperation System for Mobile Robot Manipulation. *Robotics* **2023**, *12*, 163. [CrossRef]
33. Figueroa-Olmedo, J.R.; Montalvo-Lopez, W.M.; Bayas-Sampedro, M.M. *Cinemática y Dinámica de Robots Móviles con Ruedas*, 1st ed.; CILADI: Guayaquil, Ecuador, 2023; pp. 136–138.
34. Carlucho, I.; De Paula, M.; Acosta, G.G. An Adaptive Deep Reinforcement Learning Approach for MIMO PID Control of Mobile Robots. *ISA Trans.* **2020**, *102*, 280–294. [CrossRef]
35. Saidi, S.M.; Mellah, R.; Fekik, A.; Azar, A.T. Real-Time Fuzzy-PID for Mobile Robot Control and Vision-Based Obstacle Avoidance. *Int. J. Serv. Sci. Manag. Eng. Technol.* **2022**, *13*, 1–32. [CrossRef]
36. Klancar, G.; Zdesar, A.; Blazic, S.; Skrjanc, I. *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*; Butterworth-Heinemann: Oxford, UK, 2017.
37. Azimi, A.; Shamshiri, R.R.; Ghasemzadeh, A. Adaptive Dynamic Programming for Robust Path Tracking in an Agricultural Robot Using Critic Neural Networks. *Agric. Eng. EU* **2025**, *80*, 1.
38. Shen, Z.; Du, H.; Yu, L.; Zhu, W.; Zhu, M. A Path Planning and Tracking Control Algorithm for Multi-Autonomous Mobile Robot Systems Based on Spatiotemporal Conflicts and Nonholonomic Constraints. *Actuators* **2024**, *13*, 399. [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

Symmetry Editorial Office
E-mail: symmetry@mdpi.com
www.mdpi.com/journal/symmetry



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 978-3-7258-8946-4