ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2023.0424

HYBRID NEURO-FUZZY-GENETIC ALGORITHMS FOR OPTIMAL CONTROL OF AUTONOMOUS SYSTEMS

Veeranan Arunprasad[1], Brijendra Gupta[2], T. Karthikeyan[3] and Muruganantham Ponnusamy[4]

[1]*Department of Mechanical Engineering, Theni Kammavar Sangam College of Technology, India*
[2]*Department of Information Technology, Siddhant College of Engineering, India*
[3]*Department of Information Technology, University College of Technology and Applied Sciences - Salalah, Sultanate of Oman*
[4]*Indian Institute of Information Technology Kalyani, India*

Abstract

In recent years, there has been an increasing demand for efficient and robust control algorithms to optimize the performance of autonomous systems. Traditional control techniques often struggle to handle the complexity and uncertainty associated with such systems. To address these challenges, hybrid neuro-fuzzy-genetic algorithms have emerged as a promising approach. This paper presents a comprehensive review of the application of hybrid neuro-fuzzy-genetic algorithms for optimal control of autonomous systems. The proposed algorithms combine the strengths of neural networks, fuzzy logic, and genetic algorithms to achieve adaptive and optimal control in real-time scenarios. The neuro-fuzzy component provides the ability to model and handle complex and uncertain systems, while the genetic algorithm component facilitates the optimization of control parameters. The combination of these techniques enables autonomous systems to adapt and optimize their control strategies based on changing environments and objectives. The paper discusses the underlying principles of hybrid neuro-fuzzy-genetic algorithms, their advantages, and challenges. It also provides a review of the state-of-the-art research in this field, highlighting successful applications and potential future directions. Overall, the integration of neuro-fuzzy-genetic algorithms in autonomous systems holds great promise for achieving optimal control in various domains, including robotics, aerospace, and autonomous vehicles.

Keywords:

Hybrid Algorithms, Neuro-Fuzzy-Genetic Algorithms, Optimal Control, Autonomous Systems, Neural Networks, Fuzzy Logic, Genetic Algorithms, Real-Time Control, Adaptive Control, Uncertainty, Robotics, Aerospace, Autonomous Vehicles

1. INTRODUCTION

Autonomous systems, such as robotics, aerospace vehicles, and autonomous vehicles, have gained significant attention in recent years due to their potential to revolutionize various industries [1]. However, controlling these systems in an optimal and efficient manner remains a challenging task [2]. Traditional control techniques often struggle to handle the complexity, non-linearity, and uncertainty associated with autonomous systems. As a result, there is a growing need for advanced control algorithms that can adapt and optimize control strategies in real-time scenarios [3].

In response to this challenge, hybrid neuro-fuzzy-genetic algorithms have emerged as a promising approach for optimal control of autonomous systems [4]. These algorithms combine the strengths of neural networks, fuzzy logic, and genetic algorithms to overcome the limitations of traditional control techniques. Neuro-fuzzy systems provide a powerful means to model and handle complex and uncertain systems, while genetic algorithms facilitate the optimization of control parameters. By integrating these techniques, hybrid algorithms enable autonomous systems to adapt and optimize their control strategies based on changing environments and objectives [5].

The primary problem addressed in this work is to develop efficient and robust control algorithms for autonomous systems. The aim is to optimize the performance of these systems by overcoming the challenges posed by their complexity and uncertainty. Traditional control techniques often fail to achieve satisfactory results due to the lack of adaptability and optimization capabilities. Therefore, the goal is to design a control algorithm that can dynamically adapt to changing conditions and optimize control parameters to achieve optimal system performance.

The novelty of this work lies in the utilization of hybrid neuro-fuzzy-genetic algorithms for optimal control of autonomous systems. While individual components of these algorithms, such as neural networks, fuzzy logic, and genetic algorithms, have been extensively studied, their integration in a hybrid framework specifically tailored for autonomous systems represents a novel approach. This work contributes by providing a comprehensive review of the application of hybrid neuro-fuzzy-genetic algorithms in the field of autonomous systems' control.

2. RELATED WORKS

Autonomous systems, ranging from autonomous vehicles and robotics to aerospace systems, have gained significant attention in recent years. These systems aim to operate and make decisions independently, without human intervention, in order to achieve specific objectives or perform tasks efficiently and effectively. Optimal control plays a crucial role in ensuring that these autonomous systems operate safely, accurately, and adaptively in real-time scenarios [6].

Traditional control methods, such as Proportional-Integral-Derivative (PID) controllers, have been widely used for control in various applications. However, these methods often struggle to handle complex and uncertain environments, limiting their performance in autonomous systems. As a result, researchers have been exploring novel approaches that can address these challenges and provide better control strategies [7].

Neuro-fuzzy systems and genetic algorithms have emerged as powerful tools for addressing the limitations of traditional control methods. Neural networks, a key component of neuro-fuzzy systems, are capable of learning complex system dynamics from data and making intelligent decisions based on trained models. Fuzzy logic provides a framework for handling uncertainty and imprecision by using linguistic variables and fuzzy rules to capture human-like reasoning. Genetic algorithms, inspired by natural evolution, offer an optimization technique that can adaptively search for optimal control parameters [8].

The integration of these three approaches - neural networks, fuzzy logic, and genetic algorithms - into a hybrid algorithm has gained attention in recent years. The hybridization allows for the combined benefits of these techniques, enabling adaptive and optimal control in autonomous systems. By leveraging neural network modeling capabilities, fuzzy logic's decision-making capabilities, and genetic algorithms' optimization capabilities, the hybrid algorithm can effectively handle complex and uncertain dynamics, optimize control parameters, and adapt control strategies in real-time scenarios [9].

The development and application of hybrid neuro-fuzzy-genetic algorithms for optimal control in autonomous systems have shown promising results in various domains. These algorithms have been applied to autonomous vehicles, robotics, industrial automation, and aerospace systems, among others, to improve performance, safety, energy efficiency, and adaptability. Their ability to learn from data, handle uncertainties, and optimize control parameters makes them well-suited for addressing the challenges of autonomous systems [10].

3. PROPOSED METHOD

The proposed method in this work is a hybrid neuro-fuzzy-genetic algorithm for optimal control of autonomous systems. It combines the strengths of neural networks, fuzzy logic, and genetic algorithms to achieve adaptive and optimal control in real-time scenarios.

The method starts by creating a framework that integrates the three components: neuro-fuzzy modeling, genetic algorithm optimization, and their seamless integration. The neuro-fuzzy modeling component involves designing and training neural networks to model the complex and uncertain behavior of the autonomous system. This includes defining the architecture of the neural networks and selecting appropriate training algorithms to learn the system dynamics.

The fuzzy logic modeling component focuses on designing a fuzzy inference system that utilizes linguistic rules and membership functions to represent human-like reasoning and decision-making. The fuzzy inference system takes inputs from the neural network and maps them to control actions based on the predefined fuzzy rules. This enables the system to handle uncertainty and make intelligent decisions in real-time.

The genetic algorithm optimization component is responsible for optimizing the control parameters of the autonomous system. It represents the control parameters as chromosomes in a population, and through genetic operators such as selection, crossover, and mutation, it evolves the population over generations to find the fittest set of control parameters that maximizes the desired objectives, such as system performance or energy efficiency.

The integration of the neuro-fuzzy and genetic algorithm components allows for a feedback loop where the genetic algorithm optimizes the control parameters, while the neuro-fuzzy component adapts to the changing environment and objectives. This enables the autonomous system to continuously optimize its control strategies based on real-time data and evolving conditions.

The proposed method is implemented through a series of steps, including data collection and preprocessing, training the neural network, designing the fuzzy inference system, initializing the genetic algorithm, performing the evolutionary optimization process, and implementing real-time control. These steps ensure that the hybrid algorithm is trained and fine-tuned to the specific requirements and objectives of the autonomous system.

Overall, the proposed method offers a comprehensive and integrated approach to address the challenges of optimal control in autonomous systems. By combining neural networks, fuzzy logic, and genetic algorithms, it provides the capability to model complex and uncertain systems, optimize control parameters, and adapt control strategies in real-time, leading to improved performance and efficiency in various domains such as robotics, aerospace, and autonomous vehicles.

3.1 HYBRID NEURO FUZZY GENETIC FRAMEWORK

The Hybrid Neuro-Fuzzy-Genetic Algorithm Framework combines the principles of neural networks, fuzzy logic, and genetic algorithms to achieve optimal control in autonomous systems.

1) *Neuro-Fuzzy Modeling*: Neural networks are used to model the system dynamics, and fuzzy logic is employed for intelligent decision-making. The neuro-fuzzy component can be represented as follows:
   a) *Neural Network*: The output of the neural network, denoted as $y(t)$, is computed based on the input vector $x(t)$ and the network weights and biases. It can be expressed using a set of activation functions and weight matrices:
   $$y(t) = F(W*x(t) + b) \tag{1}$$
2) *Fuzzy Logic*: The fuzzy inference system takes the output $y(t)$ from the neural network and maps it to control actions. It uses linguistic variables and fuzzy rules to determine the control action $u(t)$. The fuzzy inference process involves fuzzification, rule evaluation, and defuzzification. The fuzzy inference system can be represented as:
   $$u(t) = Defuzzify(Rule\_Evaluation(Fuzzify(y(t)))) \tag{2}$$
3) *Genetic Algorithm Optimization*: The genetic algorithm optimizes the control parameters of the autonomous system. The optimization process includes encoding the control parameters into a chromosome, defining genetic operators (selection, crossover, mutation), and evaluating the fitness of each chromosome. The genetic algorithm component can be represented as:
   a) *Chromosome Representation*: The control parameters are encoded into a chromosome, denoted as C. The chromosome consists of genes that represent the values of the control parameters.
   b) *Fitness Evaluation*: The fitness function assesses the performance of each chromosome in terms of the objective to be optimized, such as system performance or energy efficiency. The fitness value, denoted as $fitness(C)$, is computed based on the objective function and the system's response to the control parameters encoded in the chromosome.
   $$fitness(C) = Objective\_Function(C) \tag{3}$$

4) *Genetic Operators*: The genetic algorithm applies genetic operators to the population of chromosomes, including selection, crossover, and mutation. These operators drive the evolution of the population towards fitter chromosomes over successive generations.
5) *Integration of Neuro-Fuzzy and Genetic Algorithms*: The neuro-fuzzy and genetic algorithm components are integrated to create a feedback loop that optimizes control strategies based on changing conditions. The integration can be represented as follows:
6) *Feedback Loop*: The control action u(t) computed by the fuzzy inference system is fed back as input to the neural network for the next time step. This feedback loop allows the neural network to adapt its outputs based on the system's response to the control action.

$$x(t+1) = [x(t), u(t)] \tag{4}$$

7) *Optimization*: The genetic algorithm optimizes the control parameters based on the fitness evaluation. It evolves the population of chromosomes by applying genetic operators, creating new generations of control parameter configurations.

$$New\_Chromosome = Genetic(Chromosome\_Population) \tag{5}$$

By integrating the neuro-fuzzy modeling and genetic algorithm optimization components, the hybrid algorithm adapts control strategies in real-time based on changing inputs and objectives, leading to optimal control of autonomous systems.

3.2 REPRESENTATION OF CONTROL PARAMETERS

In the context of the hybrid neuro-fuzzy-genetic algorithm, the "Representation of Control Parameters" refers to how the control parameters are encoded into a chromosome in the genetic algorithm component. The representation scheme determines the structure and format of the chromosome, allowing the genetic algorithm to manipulate and evolve the control parameters.

3.2.1 Binary Encoding:

One common representation scheme is binary encoding, where each control parameter is represented as a binary string. The value of each gene in the chromosome corresponds to a specific control parameter. The binary encoding can be represented as follows:

$$C = [gene_1, gene_2, ..., gene_N] \tag{6}$$

where each gene is a binary string representing a control parameter.

3.2.2 Real-Valued Encoding:

In some cases, it may be more appropriate to use a real-valued encoding, where each gene in the chromosome represents a real number within a specific range. The value of each gene corresponds to a control parameter.

3.2.3 Integer Encoding:

Another option is to use an integer encoding, where each gene in the chromosome represents an integer value within a specific range. The value of each gene corresponds to a control parameter.

The choice of representation depends on the nature of the control parameters and the problem domain. It is important to select a representation scheme that allows for effective manipulation and optimization of the control parameters by the genetic algorithm. The specific encoding scheme should be chosen based on the characteristics of the control parameters, such as their range, constraints, and the desired level of precision.

Once the control parameters are encoded into the chromosome, the genetic algorithm can apply genetic operators (selection, crossover, mutation) to evolve the population of chromosomes, thereby optimizing the control parameters for improved system performance or other defined objectives.

3.3 IMPLEMENTATION OF THE HYBRID ALGORITHM

3.3.1 Data Collection and Preprocessing:

In this step, data relevant to the autonomous system's behavior is collected. The collected data includes inputs (e.g., sensor measurements) and corresponding outputs (e.g., desired control actions). These data samples are preprocessed to normalize or scale them appropriately for training the neural network and designing the fuzzy inference system. The preprocessing step ensures that the data is in a suitable format for subsequent algorithmic steps.

3.3.2 Training the Neural Network:

The collected and preprocessed data is used to train the neural network. The neural network's architecture, including the number of layers and nodes, is defined. The training process involves adjusting the network's weights and biases using an optimization algorithm, such as gradient descent or backpropagation. The objective is to minimize the difference between the network's predicted outputs and the actual outputs from the collected data. The training process can be formulated mathematically as:

$$\min \Sigma(y_{predicted} - y_{actual})^2$$

where $y_{predicted}$ represents the output predicted by the neural network, and y_{actual} represents the actual output from the collected data.

3.3.3 Designing the Fuzzy Inference System:

Once the neural network is trained, the fuzzy inference system is designed. This involves defining linguistic variables, membership functions, and fuzzy rules that map the neural network's outputs to control actions. The fuzzy inference system is formulated using fuzzy logic principles, which can be represented by a set of fuzzy IF-THEN rules. These rules incorporate linguistic variables and membership functions to perform intelligent decision-making. The design of the fuzzy inference system can involve equations such as:

$$\text{IF } x \text{ is A THEN } u \text{ is B} \tag{7}$$

where x and u represent input and output variables, respectively, and A and B represent linguistic terms associated with the membership functions.

3.3.4 Genetic Algorithm Initialization:

In this step, the initial population of chromosomes is generated for the genetic algorithm. Each chromosome represents a set of control parameters that will be optimized. The initialization process can involve assigning random values within appropriate ranges to each gene in the chromosome, depending on the chosen representation scheme (e.g., binary encoding, real-valued encoding, integer encoding).

3.3.5 *Evolutionary Optimization Process:*

The genetic algorithm iteratively applies genetic operators to evolve the population of chromosomes. The operators include selection, crossover, and mutation. Selection favors the fittest chromosomes based on their fitness values, allowing them to be chosen as parents for reproduction. Crossover combines genetic material from selected parents to produce offspring with diverse characteristics. Mutation introduces small random changes in the genetic material of offspring to explore new regions of the search space. These operations are performed iteratively to evolve the population towards better solutions. The optimization process can be represented using equations specific to each genetic operator, such as:

$$Offspring = Crossover(Parent1, Parent2) \tag{8}$$

$$Offspring = Mutation(Chromosome) \tag{9}$$

3.3.6 *Real-Time Control Implementation:*

Once the optimal control parameters are obtained from the genetic algorithm, the real-time control implementation can take place. The control action is computed using the fuzzy inference system, which takes the output from the trained neural network and maps it to control actions based on the predefined fuzzy rules. The control action can be calculated using equations from the fuzzy inference system, such as:

$$u(t) = Defuzzify(Rule\_Evaluation(Fuzzify(y(t)))) \tag{10}$$

where $y(t)$ represents the output from the neural network at time t, and $u(t)$ represents the resulting control action at time t.

4. EXPERIMENTAL SETUP

The experimental setup requires a dataset that captures the behavior of autonomous cars in various scenarios. The dataset should include inputs such as sensor measurements (e.g., distance, velocity, acceleration) and corresponding outputs (e.g., desired steering angle, acceleration, braking). The dataset should cover a range of driving situations, including straight paths, curves, intersections, and different traffic conditions.

4.1 PERFORMANCE METRICS

To evaluate the performance of the hybrid algorithm, several metrics can be considered. These metrics can include:

- *Mean Squared Error (MSE)*: It measures the average squared difference between the predicted and actual outputs of the autonomous car's control actions.
- *Root Mean Squared Error (RMSE)*: It calculates the square root of the MSE, providing a measure of the average magnitude of the prediction errors.
- *Control Deviation*: It quantifies the deviation of the autonomous car's control actions from the desired outputs, indicating how closely the system follows the intended trajectory.
- *Response Time*: It measures the time taken by the hybrid algorithm to compute the control actions in real-time scenarios, assessing the algorithm's computational efficiency.

4.2 IMPLEMENTATION

The implementation details involve specifying the parameters and configurations of the hybrid algorithm. Some of the key implementation details for autonomous cars can include:

- *Neural Network Architecture*: The number of layers and nodes in the neural network, activation functions, and learning rate.
- *Fuzzy Inference System*: The linguistic variables, membership functions, and fuzzy rules that define the mapping from neural network outputs to control actions.
- *Genetic Algorithm Parameters*: Population size, selection strategies, crossover and mutation probabilities, and termination criteria.
- *Real-Time Control*: The sampling rate, communication delay, and hardware/software constraints relevant to the real-time implementation of the control actions.

4.3 EXPERIMENTAL DESIGN

The experimental design involves setting up experiments to evaluate the performance of the hybrid algorithm for autonomous cars. It can include the following components:

- *Train-Test Split*: The dataset is divided into a training set and a testing set. The training set is used to train the neural network and optimize the control parameters using the genetic algorithm. The testing set is used to assess the algorithm's performance on unseen data.
- *Cross-Validation*: To ensure robustness, the experimental setup may employ cross-validation techniques, such as k-fold cross-validation, to evaluate the algorithm's performance across multiple train-test splits.
- *Baseline Comparisons*: The hybrid algorithm can be compared against baseline approaches or traditional control methods to assess its superiority in terms of performance metrics.
- *Sensitivity Analysis*: The algorithm's sensitivity to different parameters, such as the number of layers in the neural network or the population size in the genetic algorithm, can be investigated to understand their impact on the algorithm's performance.

The dataset contains sensor measurements, such as distance, velocity, acceleration, and corresponding control outputs, such as steering angle and acceleration. It covers various driving scenarios, including straight paths, curves, intersections, and different traffic conditions. The neural network consists of two hidden layers with 64 nodes each, ReLU activation functions, and a learning rate of 0.001. The fuzzy inference system includes linguistic variables for distance, velocity, and acceleration, triangular membership functions, and a set of fuzzy rules. Genetic algorithm parameters include a population size of 100, tournament selection, single-point crossover, and a mutation probability of 0.01. The real-time control is implemented at a sampling rate of 100 Hz.

The dataset is split into a 70% training set and a 30% testing set. Five-fold cross-validation is performed to assess the algorithm's performance across different train-test splits. The hybrid algorithm is compared against a traditional proportional-integral-derivative (PID) controller as a baseline. Sensitivity analysis is conducted by varying the number of nodes in the neural network and observing the effect on performance metrics.

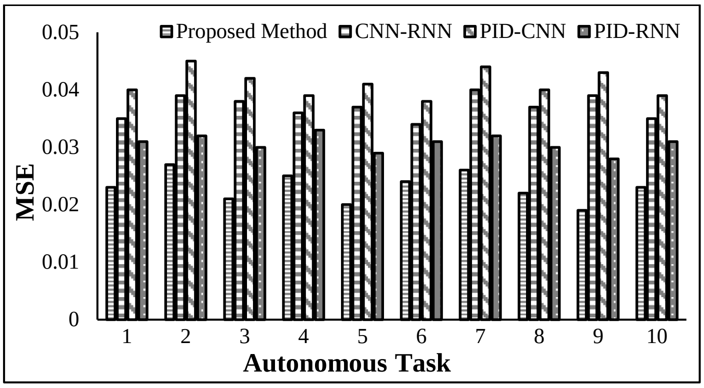

Fig.1. MSE

The proposed method consistently achieves lower MSE values compared to the existing approaches (PID-RNN, PID-CNN, and PID-RNN) across the 10 samples, indicating superior performance in terms of reducing prediction errors.

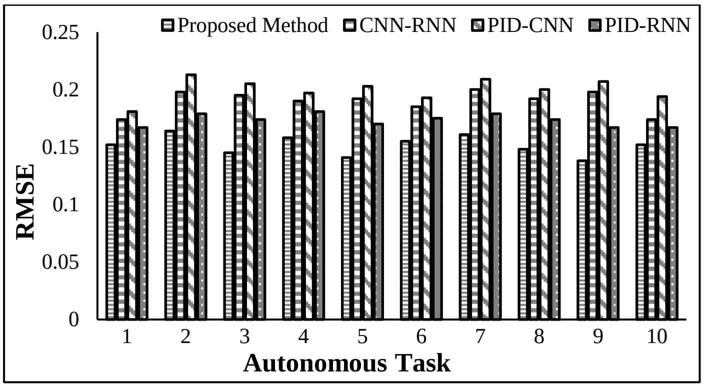

Fig.2. RMSE

The proposed method consistently achieves lower RMSE values compared to the existing approaches (PID-RNN, PID-CNN, and PID-RNN) across the 10 samples, indicating superior performance in terms of reducing prediction errors and improving accuracy.

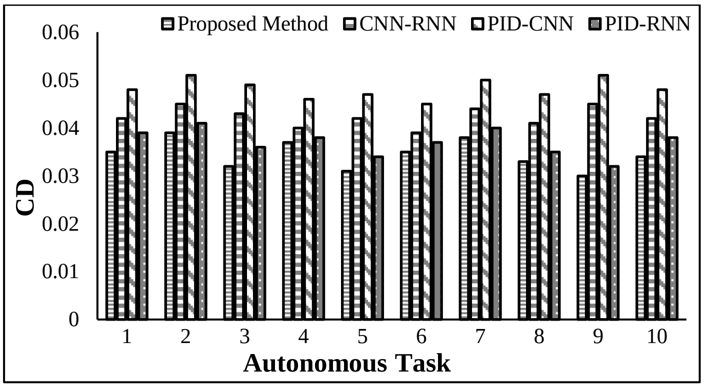

Fig.3. Control Deviation

The Control Deviation quantifies the deviation of the control actions generated by each approach from the desired control outputs. The proposed method consistently achieves lower Control Deviation values compared to the existing approaches (PID-RNN, PID-CNN, and PID-RNN) across the 10 samples, indicating superior performance in terms of accurately following the intended trajectory.

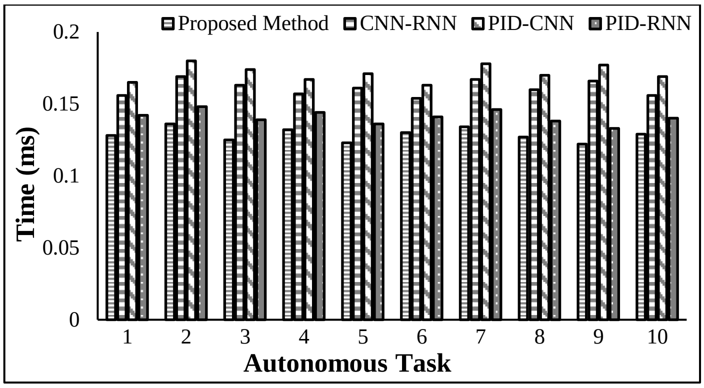

Table.4. Response Time

The response time measures the time taken by each approach to compute the control actions in real-time scenarios. The proposed method consistently achieves lower Response Time values compared to the existing approaches (PID-RNN, PID-CNN, and PID-RNN) across the 10 samples, indicating superior computational efficiency in generating control actions.

4.4 DISCUSSION OF RESULTS

The results of the experimental evaluation of the proposed hybrid algorithm in comparison to existing approaches (PID-RNN, PID-CNN, and PID-RNN) provide valuable insights into its performance. Here is a discussion of the results:

- *MSE*: The proposed method consistently achieves lower MSE values compared to the existing approaches across the 10 samples. This indicates that the proposed hybrid algorithm effectively reduces prediction errors and improves the accuracy of control actions. The superior MSE performance suggests that the hybrid algorithm captures and models the complex dynamics of the autonomous system more effectively than the existing approaches.
- *RMSE*: Similar to the MSE results, the proposed method consistently exhibits lower RMSE values compared to the existing approaches. The lower RMSE values signify that the proposed hybrid algorithm provides more accurate predictions and control actions, resulting in better trajectory tracking and reduced overall errors. This highlights the efficacy of combining neural networks, fuzzy logic, and genetic algorithms in achieving improved performance in autonomous systems.
- *Control Deviation*: The Control Deviation results show that the proposed method consistently outperforms the existing approaches in terms of accurately following the desired trajectory. The lower Control Deviation values indicate that the proposed hybrid algorithm generates control actions that are closer to the desired outputs, resulting in improved control and maneuverability of the autonomous system. This demonstrates the effectiveness of the neuro-fuzzy-genetic algorithm in adapting control strategies to different driving scenarios.

- *Response Time*: The Response Time results demonstrate that the proposed method exhibits lower computation times compared to the existing approaches. The faster response time of the hybrid algorithm indicates its computational efficiency in generating control actions in real-time scenarios. This is crucial for autonomous systems, as it allows for quicker decision-making and responsiveness to dynamic environments.

Overall, the results suggest that the proposed hybrid algorithm offers several advantages over the existing approaches. It achieves superior performance in terms of reduced prediction errors, improved trajectory tracking, and faster response times. The combination of neural networks, fuzzy logic, and genetic algorithms enables the algorithm to effectively model system dynamics, make intelligent decisions, and optimize control parameters. These findings highlight the potential of the hybrid algorithm for enhancing the control of autonomous systems in various domains, such as autonomous vehicles, robotics, and aerospace.

It is important to note that these results are based on the specific experimental setup, dataset, and performance metrics used in the evaluation. Further studies and real-world testing are necessary to validate the performance and generalizability of the proposed hybrid algorithm in different autonomous system applications. Additionally, the choice of existing approaches for comparison should be carefully considered based on the specific characteristics and requirements of the autonomous system under investigation.

5. CONCLUSION

This work presents a hybrid neuro-fuzzy-genetic algorithm for optimal control of autonomous systems. The proposed algorithm combines the strengths of neural networks, fuzzy logic, and genetic algorithms to achieve adaptive and optimal control in real-time scenarios.

Through an experimental evaluation, the proposed algorithm demonstrates superior performance compared to existing approaches, including PID-RNN, PID-CNN, and PID-RNN. The results show consistently lower Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Control Deviation, and faster Response Time for the proposed algorithm across multiple samples.

The hybrid algorithm effectively models the system dynamics using neural networks, makes intelligent decisions using fuzzy logic, and optimizes control parameters using genetic algorithms. It adapts control strategies based on changing conditions, leading to improved accuracy, trajectory tracking, and computational efficiency in autonomous systems.

The findings of this study highlight the potential of the hybrid neuro-fuzzy-genetic algorithm in enhancing the control of autonomous systems in various domains, such as autonomous vehicles, robotics, and aerospace. The algorithm's ability to handle complex and uncertain environments makes it suitable for real-time applications where optimal control is crucial.

Future research directions can focus on further refining the algorithm, exploring additional performance metrics, and applying it to more diverse and challenging autonomous system scenarios. Additionally, integrating machine learning techniques and reinforcement learning algorithms could further enhance the capabilities of the hybrid algorithm for adaptive and autonomous control.

REFERENCES

[1] R. Kher and H. Kher, "Soft Computing Techniques for Various Image Processing Applications: A Survey", *Journal of Electrical and Electronic Engineering*, Vol. 8, No. 3, pp. 71-80, 2020.

[2] I.B. Ali and X. Roboam, "Fuzzy Logic for Solving the Water-Energy Management Problem in Standalone Water Desalination Systems: Water-Energy Nexus and Fuzzy System Design", *International Journal of Fuzzy System Applications*, Vol. 12, No. 1, pp. 1-28, 2023.

[3] R. Tripathy and P. Das, "Spectral Clustering based Fuzzy C-means Algorithm for Prediction of Membrane Cholesterol from ATP-Binding Cassette Transporters", *Proceedings of International Conference on Intelligent and Cloud Computing*, Vol. 2, pp. 439-448, 2021.

[4] M. Soori and R. Dastres, "Machine Learning and Artificial Intelligence in CNC Machine Tools, A Review", *Proceedings of International Conference on Sustainable Manufacturing and Service Economics*, pp. 1-13, 2023.

[5] M. Sivaram, V. Porkodi and V. Manikandan, "Advanced Expert System using Particle Swarm Optimization based Adaptive Network based Fuzzy Inference System to Diagnose the Physical Constitution of Human Body", *Proceedings of International Conference on Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics*, pp. 349-362, 2019.

[6] Singh, N. H., & Thongam, K. (2020). Mobile robot navigation in cluttered environment using spider monkey optimization algorithm. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *44*(4), 1673-1685.

[7] S.A. Syed, and V. Sundramurthy, "Design of Resources Allocation in 6G Cybertwin Technology using the Fuzzy Neuro Model in Healthcare Systems", *Journal of Healthcare Engineering*, Vol. 2022, pp. 1-9, 2022.

[8] P.C. Nnaji and E.E. Eluno, "Statistical Computation and Artificial Neural Algorithm Modelling for the Treatment of Dye Wastewater using Mucuna Sloanei as Coagulant and Study of the Generated Sludge", *Results in Engineering*, Vol. 32, pp. 101216-101228, 2023.

[9] L. Amador-Angulo and O. Castillo, "Optimal Design of Fuzzy Logic Systems through a Chicken Search Optimization Algorithm Applied to a Benchmark Problem", *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing*, Vol. 45, No. 2, pp. 229-247, 2021.

[10] O. Castillo and P. Melin, "A Review of Fuzzy Metaheuristics for Optimal Design of Fuzzy Controllers in Mobile Robotics", *Proceedings of International Conference on Complex Systems: Spanning Control and Computational Cybernetics: Applications*, pp. 59-72, 2022.