Autonomous Intersection Management System

Martin Mamdouh
Department of Mechanical Engineering
German International University
Cairo, Egypt

Youssef Yasser
Department of Mechanical Engineering
German International University
Cairo, Egypt

Mariam Sherif
Department of Mechanical Engineering
German International University
Cairo, Egypt

Hazem Ashraf
Department of Mechanical Engineering
German International University
Cairo, Egypt

Mai Khaled
Department of Mechanical Engineering
German International University
Cairo, Egypt

Ibrahim Yousri
Department of Mechanical Engineering
German International University
Cairo, Egypt

***Abstract*—Modern transportation systems experience significant traffic delays and accidents as a result of intersections. Autonomous Intersection Management (AIM) is a system that uses intelligent algorithms and control technologies to manage traffic flow at intersections without human intervention. AIM optimizes the timing and order for each vehicle to cross the intersection, reducing waiting time and the risk of accidents. AIM can adapt to changing traffic conditions in real time and prioritize emergency vehicles. AIM has been extensively tested in simulation environments and has shown promising results, including reduced travel time, improved fuel efficiency, and enhanced safety. However, further testing and refinement are necessary before AIM can be widely implemented in real-world settings. First, a dynamic model for the vehicles was developed. Then, a low-level control was used to control the vehicle's physical components, such as the engine, steering, and brakes. Then, a high-level control was designed which is responsible for decision-making processes that guide a vehicle's behavior. It's the heart of the AIM system, and it must be designed to optimize the flow of traffic while considering safety, efficiency, and fairness. Finally, software was developed to integrate the model and two layers of control to regulate the autonomous intersection management system.**

***Index Terms*—Autonomous intersection, Manage traffic flow, Dynamic model, reinforcement learning, low-level control**

I. Introduction

A. Background

In recent years, traffic congestion has become a significant problem in urban areas due to the increasing number of vehicles on the roads. Inefficient traffic management at intersections exacerbates delays, fuel consumption, and environmental pollution. To tackle these challenges, autonomous intersection management systems (AIMS) have emerged as a promising solution. AIMS utilizes advanced technologies and communication systems to enable vehicles to coordinate and make real-time decisions, aiming to improve traffic efficiency, reduce congestion, and enhance safety.

This research project addresses the need for effective traffic management solutions by developing an AIMS. Traditional traffic signal systems often lead to suboptimal outcomes, resulting in idling vehicles and unnecessary delays. By leveraging autonomous vehicles and advanced control algorithms, the project aims to create an intelligent and adaptive traffic management system.

The project aligns with the trends in transportation and urban planning, considering the rise of autonomous vehicles and the push for smart cities. Developing innovative methods to manage traffic and reduce congestion has become a priority. By exploring the implementation of an AIMS, the project contributes to the knowledge in the field and provides insights into the benefits and challenges of such systems in real-world scenarios.

Furthermore, the project leverages emerging technologies and methodologies, particularly reinforcement learning. By incorporating reinforcement learning algorithms into the AIMS's high-level controller, the project explores the potential of this approach to optimize traffic flow and prevent collisions.

In summary, this research project focuses on developing an autonomous intersection management system to address the challenges associated with traditional traffic signal systems. By utilizing advanced technologies and control algorithms, the project aims to optimize traffic flow, reduce congestion, and enhance safety. It aligns with transportation and urban planning trends and offers an opportunity to explore the potential of emerging technologies like reinforcement learning in traffic management.

B. Problem Statement

According to the National Highway Traffic Safety Administration (NHTSA), in 2019, there were 36,096 fatalities and 2.74 million injuries in motor vehicle crashes in the US. Intersection crashes accounted for 22.5% of all fatalities and 40.9% of all injuries. According to the US Environmental Protection Agency (EPA), transportation is the largest source of greenhouse gas emissions in the US, accounting for 28% of total emissions. Cars and trucks are responsible for the majority of these emissions. Congested traffic and inefficient intersection designs can exacerbate these emissions. In Europe, the annual cost of congestion was estimated to be €170 billion in 2019, according to the European Commission's

Identify applicable funding agency here. If none, delete this.

Urban Mobility Package. Traditional traffic control systems like traffic lights operate on fixed schedules and cannot adapt to changing traffic patterns, resulting in wasted time and fuel for drivers [5]. As the number of autonomous vehicles on the roads increases, traditional traffic management methods, such as traffic lights, will become inefficient and unsuitable for handling the complexity of intersections [6].

C. Aims and Objectives

This study aims at developing and evaluating an AIMS algorithm in a simulated environment. The algorithm will be tested on various traffic scenarios to assess its efficiency and safety. The study will also examine the feasibility of implementing AIMS in real-world urban intersections, although actual implementation is outside the scope of this study. Additionally, the study will focus on the practical outcomes of AIMS implementation, such as traffic efficiency, emissions reduction, and safety enhancements, and will discuss the technical details of the AIMS algorithm and its implementation.

- To develop an AIMS that can effectively manage intersection traffic flow, reduce congestion, and enhance safety.
- To evaluate the performance of the AIMS in a simulated environment at different scenarios and compare it with traditional intersection management systems.
- To assess the feasibility of implementing AIMS in real-world urban intersections and identify potential challenges and solutions.
- To identify the potential practical outcomes of AIMS implementation, such as improved traffic efficiency, reduced emissions, and enhanced safety for drivers, pedestrians, and cyclists.

II. Literature Review

The literature review chapter demonstrates a thorough knowledge of the AIMs and provides arguments and historical background to support the study focus. This review helps to contextualize the current project and establish its contribution to the field. The review will cover different topics such RL algorithms used in autonomous vehicles and their applications in AIM systems. Additionally, the review will analyze the effectiveness of existing AIM systems and their potential applications. Overall, this literature review aims to provide a comprehensive understanding of RL algorithms and model in AIM systems to select the most suitable model and algorithm for the AIM system.

A. Historical Background

The development of AIMS has been a gradual process that started in the early 2000s. One of the earliest and most influential research works in this area was the work done by Professor Alain Kornhauser of Princeton University. He developed a concept called ”Smart Driving System” in 1996, which was primarily aimed at reducing congestion and improving mobility through a combination of technologies such as GPS and short-range communication devices.

Similarly, in 2003, researchers at the California PATH (Partners for Advanced Transportation Technology) program had begun exploring the use of intelligent transportation systems (ITS) to control traffic flow at intersections. They proposed a system that uses advanced wireless communication technologies and vehicle-to-infrastructure (V2I) communication to manage traffic flow at intersections.

Over the years, many researchers and organizations have contributed to the development of AIMs. For instance, in 2007, researchers from the University of Texas at Austin developed a system called ”Smart Intersection” that uses a combination of sensors, cameras, and communication technologies to optimize the flow of traffic through intersections. With this system, cars approaching an intersection send a signal to a communication system located at the intersection. The communication system then calculates the optimal speed for each vehicle to pass through the intersection without stopping, thus reducing congestion and improving safety.

In recent years, several experiments have been conducted to test the viability of AIMS. For example, in 2017, researchers from the University of Texas at Austin conducted a pilot study of their Smart Intersection system in the city of Austin. The study demonstrated that the system was effective in reducing congestion and improving traffic flow at intersections.

B. Reinforcement Learning Models

Sources [1] - [2] surveyed and critically evaluated the model-free and model-based RL methodologies. First, for model-free RL methodology, some of its advantages are simplicity, robustness, and flexibility. However, some of its disadvantages are sample inefficiency, exploration and exploitation trade-off and resultant instability. Second, for model-based RL methodology, it is more adaptable and has better sample efficiency. However, model approximation and computational complexity are disadvantages.

Source [3] evaluated Flow-based RL. This technique avoids the need for explicit value estimation or policy optimization, as the policy is directly modeled through the flow. However, it is less flexible and requires the policy to be represented through the flow, which can result in an information bottleneck that limits the performance of the reinforcement learning agent.

C. Reinforcement Learning in Autonomous Vehicles

Sources [7] and [8] present Deep Reinforcement Learning applied in autonomous vehicles. AVs using DRL can share the data with other vehicles which is essential in AIM. On the other hand, Like all AI systems, DRL models are susceptible to adversarial attacks, which can cause them to fail or behave unpredictably.

Sources [9] and [10] investigated Q-learning. AVs can use Q-learning to learn how to navigate through a complex intersection. However, it has some limitations. For example, it is difficult to apply with large and complex environments as the number of Q-values required grows exponentially. Also, it is not suitable for ethical decision-making. For example, Q-learning focuses explicitly on reward optimization and may ignore some ethical considerations, such as human safety.

D. Summary and Implications

To summarize, The literature review provides a comprehensive analysis of various topics related to reinforcement learning. It covers different RL models, including model-free, model-based, and flow-based models. It also analyzes multi-agent reinforcement learning and discusses the application of reinforcement learning in autonomous vehicles.

The review highlights that RL is a powerful tool for optimizing the decision-making process in AIM systems. The different models of RL provide distinct advantages, depending on the specific requirements of the system. For example, Model-free RL allows the system to learn through trial and error. In contrast, model-based reinforcement learning provides a more explicit representation of the environment. On the other hand, Flow-based models use temporal difference learning to improve the system's performance.

Moreover, the review also emphasizes the importance of multi-agent RL in AIM systems. The collaborative decision-making process of multi-agent systems can provide better results than single-agent models, especially in complex systems with multiple agents. Furthermore, the literature review highlights the application of RL in AVs. The use of RL can improve the decision-making process of AVs, leading to safer driving conditions and reducing the likelihood of accidents.

One potential gap in the previous papers is the lack of research on the effectiveness and adaptability of AIMs under different traffic conditions and scenarios. Although, current simulation-based studies offer valuable insights into the potential performance of AIMS, they often assume ideal conditions, and may not be representative of real-world traffic situations.

However, by using RL, an AIM system can learn to adapt to different traffic patterns and scenarios and can be tested under more realistic conditions. This type of research can provide valuable information about the practicality and effectiveness of AIMs and can inform the future development and implementation of these systems in real-world transportation networks.

As a result of this review, Twin Delayed deep deterministic policy gradient (TD3) algorithm is used in the AIM system. It is a model-free based model that is widely used RL. This algorithm is based on deep neural networks and can handle continuous action spaces. The TD3 algorithm is suitable for AIM systems as it does not require a model of the environment and can learn directly from experiences.

III. Methodology, Design, and Analysis

The project aims to design a simulation system to control cars at an intersection point. Each car has its own dynamic model. The cars will communicate their positions and velocities via a distributed communication system. Each car will have a high-level controller, which is a reinforcement learning algorithm to avoid collisions. The low-level controller will be a PID controller to regulate the velocity of the car. The system will be implemented using MATLAB and Simulink and tested to meet the requirements.

The powertrain block represents the power source for the vehicle, which is an internal combustion engine. The powertrain block outputs torque and rotational speed, which are then transmitted to the gearbox block.

A. Dynamic Model

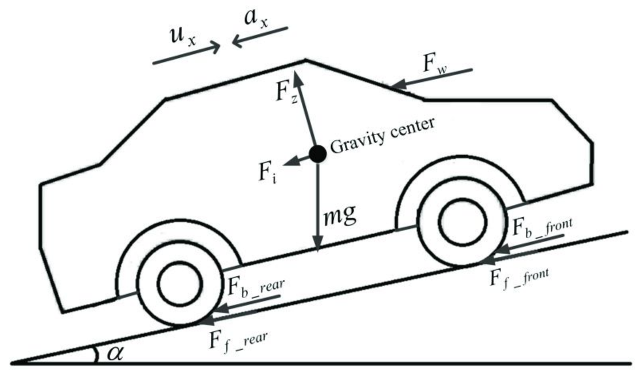

Fig. 1.
Longitudinal forces acting on a vehicle on an inclined road

The gearbox block represents the gearbox of the vehicle, which has multiple gears to allow the engine to operate efficiently at different speeds. The gearbox block takes in the torque and rotational speed from the powertrain block and outputs a new torque and speed based on the gear selected.

The engine block represents the engine of the vehicle and is modeled as a dynamic system that outputs torque and rotational speed based on the input from the powertrain block.

The final drive block represents the differential of the vehicle and is used to distribute the torque between the left and right wheels. The final drive block takes in the torque and speed from the wheel block and outputs the torque and speed at the left and right wheels.

The wheel block represents the vehicle's wheels and tires and is modeled as a rotating mass. The wheel block takes in the torque and rotational speed from the gearbox block and outputs the speed and torque at the wheel.

The brakes block represents the braking system of the vehicle and can be used to simulate the behavior of the braking system under two different conditions. In the first Condition, when the car is in motion the brake block decelerates the car till the car reaches the desired velocity. The second condition is when the car is not in motion the block's output will always be zero.

Connecting these blocks in a Simulink model to simulate the behavior of the vehicle's drivetrain system under different driving conditions and to analyze the system's performance. This model is used to optimize the design of the vehicle's drivetrain system and to evaluate the impact of changes to the system on performance, efficiency and velocity. The final equation representing the whole dynamic model system:

$$a = \frac{T_t - T_{\text{brakes}} - \text{reff} \cdot (F_{\text{aero}} + R_x)}{M}$$

a represents acceleration, T_t represents the applied torque, T_{brakes} represents the braking torque, reff represents the effective radius, F_{aero} represents the aerodynamic force, R_x represents the resistance force, and M represents moments.

B. Controllers

1) Low-Level Controller (PID): The PID (Proportional-Integral-Derivative) simulation block is a component commonly used in control engineering. Its purpose is to simulate the behavior of a closed-loop control system, where the output of the PID block is used as an input to a dynamic model, which generates a velocity output that is fed back to the PID block as one of its inputs.

The other input to the PID block is a desired velocity signal from a high-level controller. The PID block compares this setpoint signal to the actual velocity signal from the dynamic model and generates a control signal that adjusts the dynamic model's output velocity to match the setpoint.

The PID block uses three parameters to calculate the control signal: proportional gain, integral gain, and derivative gain. These gains determine how much weight is given to the current error (the difference between the setpoint and the actual velocity), the past error, and the future error (the rate of change of the error), respectively. The combination of these three gains determines the overall response of the system to changes in the setpoint in the dynamic model.

2) High-level controller TD3: The high-level control is achieved using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. TD3 is a reinforcement learning algorithm that learns a policy for controlling the car based on the observed states and rewards.

The intersection scenario involves three cars: one approaching from the east, which is controlled using TD3, and two cars approaching from the north and south, respectively, moving at random velocities ranging from 5 to 15 m/s. Each car starts from a random position within the intersection. The distance between the cars is calculated using the Euclidean distance equation.

To facilitate control, the system employs nine observations: the velocity and position of each car, the distance between the car from the east and the car from the north, the distance between the car from the east and the car from the south, and a flag indicating whether the car is close to the intersection.

The action space consists of a single action, which represents the desired velocity of the car and ranges between 0 and 15 m/s.

The reward function combines various factors to provide a measure of the agent's performance. It penalizes collisions and early stopping, encourages reaching the destination, and considers the car's velocity and distance to the desired position within the intersection. The reward function is designed as follows:

The episode is considered finished based on either a collision or successful arrival at the destination.

The integration of the TD3 algorithm and the defined reward function enables the high-level control of the car from the east, allowing it to navigate the intersection optimally while considering its observations and maximizing the cumulative rewards.

Throughout the software implementation phase, the TD3 algorithm is trained and fine-tuned using appropriate techniques to improve its control capabilities and ensure efficient intersection management.

The reward function is designed as follows:

Collision penalty = -4000 if the car collided with any of the other two cars.

Arrival reward = 1000 if the car reaches the finish line.

Velocity penalty = $-35 \times \left(1 - \frac{v}{20}\right)$ Penalty increases as velocity decreases.

Distance reward = $50 \times \left(1 - \frac{\text{distance}}{164}\right)$ Reward increases as the distance to the finish line decreases.

Stopping early penalty = -1000 if the car stops at the beginning of the episode.

Reward = Velocity penalty + Distance reward + Collision penalty + Arrival reward + Stopping early penalty

IV. Results

We will evaluate the performance of the RL Controller based on the action graphs and velocity graphs in two different scenarios. First, we look at the scenario where the car has to decelerate to avoid colliding with the other two vehicles. We can assess the success of the RL Controller in producing the desired slowing down behavior by looking at the action and velocity graphs throughout this scenario. Second, we evaluate the RL Controller's performance when the car is told to accelerate to its top speed in order to avoid colliding with the other two vehicles. The action and velocity graphs will provide insights on the RL Controller's ability in keeping a safe distance and avoiding collisions when moving at high speeds. In addition, we constructed a fuzzy logic controller as a comparison by which to measure the effectiveness of the RL Controller. We may compare the two control strategies in terms of how well they can manage deceleration and acceleration jobs while avoiding collisions by examining the data from the fuzzy logic controller. This comparison will give important insights into the benefits and drawbacks of each control mechanism, assisting in the selection of the best strategy for the particular situation.

A. Case 1

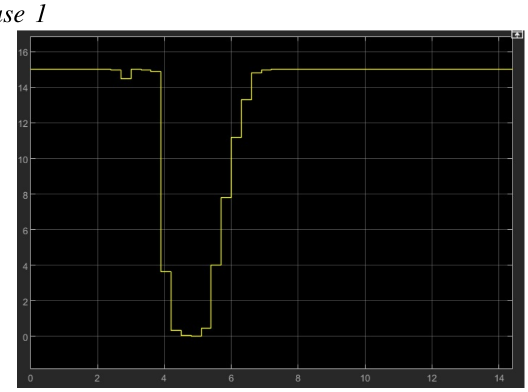

Fig. 2. Action for Case 1

The first graph represents the action from the RL Controller. The graph starts with a horizontal line indicating a constant velocity of 15 m/s. This means that car is intended to maintain a steady speed for a certain period. Following the constant velocity phase, the graph shows a decreasing slope, indicating a decrease in velocity till it reaches 0 m/s. Finally, the graph shows an increasing slope, indicating an increase in velocity until it reaches 15 m/s again and then remains constant.

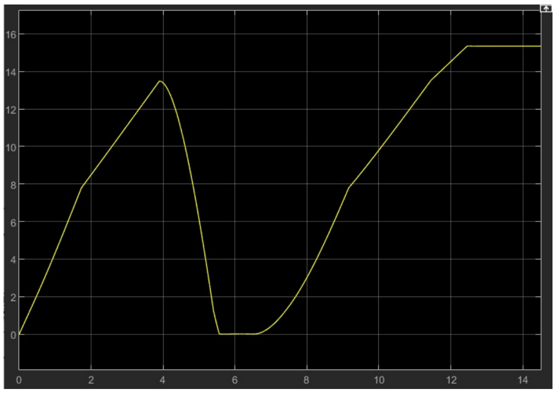

Fig. 3. Velocity for Case 1

This graph represents the actual velocity profile of the car, which is the result of simulating the desired action given in figure 1. The graph starts with an increase in velocity which starts from 0 m/s unlike figure 1, this suggests that the car experiences a continuous increase in velocity until it reaches 13.75 m/s over time instead of starting with a constant velocity. Then the graph shows a decreasing slope, indicating a decrease in velocity until it reaches 0 m/s again. After the decrease in velocity, the graph shows a final increasing slope, indicating that the car's velocity is increasing again. Finally, the graph levels off into a horizontal line, indicating that the car has reached a constant velocity at 15 m/s.

B. Case 2

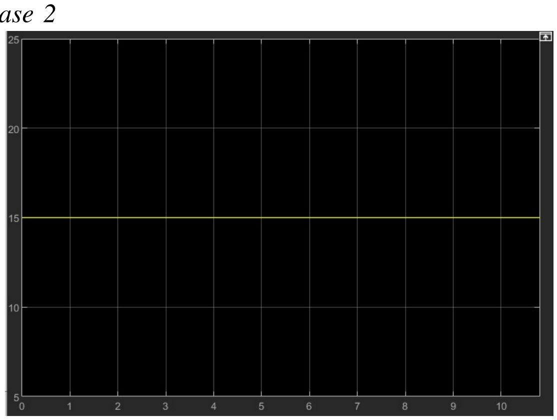

Fig. 4. Action for Case 2

If the car needs to move at its maximum speed to avoid intersecting any other cars, the behavior of the action parameter will depend on the current state of the environment. The Reinforcement Learning algorithm will evaluate the car's current state, including its speed, distance from other cars and obstacles, and the road conditions. If the car is currently not intersecting any other cars and the road ahead is clear, the RL algorithm may decide to maintain the car's current speed without any changes to the action parameter. This means that the car will continue moving at its maximum speed until there is an obstacle or another car in its path.

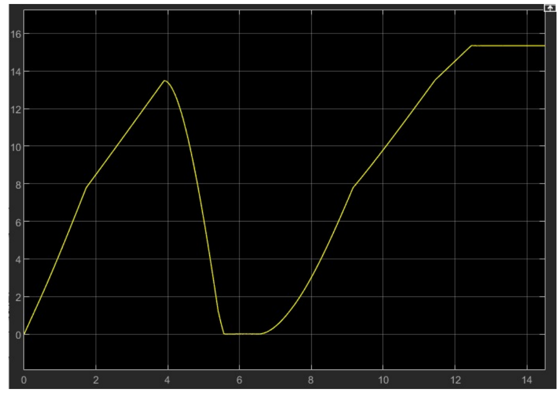

Fig. 5. Velocity for Case 2

Initially, when the car starts moving, the RL algorithm sets a minimum speed for the car to maintain momentum and move efficiently. As the car approaches a clear path, RL gradually increases the speed parameter, ensuring that it stays within the safe limits of the road and the traffic. As the car continues to move forward, the RL algorithm continually manages the speed parameter to ensure that the car maintains a safe and efficient speed while avoiding any collisions along the way. As the car approaches other vehicles or obstacles, RL may adjust the speed parameter and gradually decelerate the car to avoid any potential collisions. Once the car has successfully avoided any potential collisions, the RL algorithm once again increases the speed parameter, ensuring that the car reaches its maximum speed. This process is gradual, to avoid overshooting the target speed and to maintain control while still ensuring maximum efficiency.

C. Comparison with Fuzzy Logic Controller

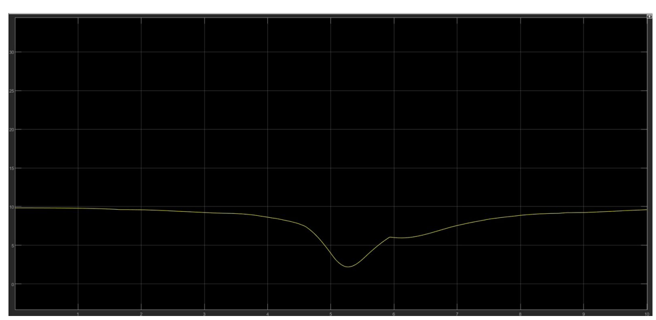

Fig. 6. Velocity for Fuzzy Controlled Car

As shown in Figure 6, the results of the velocity graph analysis show that the slope of the graph for the Fuzzy

Logic Controller is substantially smoother than that of the RL Controller. The Fuzzy Logic Controller, on the other hand, is seen to work more slowly in terms of reaching the desired velocity changes. The RL Controller, on the other hand, provides more precise control while adjusting the car's speed.

Due to the lack of training period, it is remarkable that the Fuzzy Logic Controller advances more quickly than the RL Controller. The Fuzzy Logic Controller may be put into use more rapidly because training is not necessary. Despite the time invested in training, the RL Controller manages to govern the behaviour of the car with more accuracy. Although the fuzzy logic controller is an option, it's vital to remember that it performs less accurately and responds more slowly than the RL Controller. Although time-consuming, the RL Controller's training process allows it to pick up on and adjust to the unique dynamics of the system, leading to more precise control. The trade-offs between training time, accuracy, and speed are highlighted by the comparison between the RL Controller and the Fuzzy Logic Controller. While the Fuzzy Logic Controller offers speedier implementation at the expense of accuracy and speed, the RL Controller delivers accurate and responsive control but necessitates training time.

Our analysis of the RL Controller's performance in comparison to that of the Fuzzy Logic Controller led us to the conclusion that the RL Controller is superior in terms of getting to the intended location more quickly. The RL Controller shows improved effectiveness and efficiency in guiding the car to its intended destination despite the training time required. The RL Controller's capacity to learn from and adjust to the dynamics of the system enables it to optimize its actions and make better-informed decisions, which leads to more rapid movement towards the intended location. As a result, according to our assessment, the RL Controller surpasses the Fuzzy Logic Controller in terms of efficiency and speed.

V. Conclusion

The project successfully addressed the problem statement related to reinforcement learning (RL) in autonomous vehicles (AVs) and achieved its objectives. The conclusion highlights the significance of RL in enhancing AVs' decision-making capabilities and navigating complex environments effectively. The project's methodology, design, and analysis were presented in Chapter 3, including the development of a dynamic model, low-level and high-level controllers, and research design. Ethical considerations and limitations were also discussed to ensure responsible research. Chapter 4 focused on the software implementation of the RL-based control system, adhering to relevant IEEE standards. The evaluation of the system's performance was presented in Chapter 5, demonstrating the effectiveness of RL in improving decision-making and navigation in various scenarios. Future work is proposed to further enhance the field of RL in AVs, including exploring advanced RL techniques, conducting real-world testing, addressing decision-making in uncertain environments, incorporating human-centric approaches, and promoting interoperability and standardization. In summary, the AIMS project successfully contributed to the understanding and application of RL in AVs. Moving forward, future work in this field could focus on exploring novel RL algorithms and architectures, integrating transfer learning techniques, extending the system to handle multiagent scenarios, and adapting the control system for real-world deployment. Attention should be given to ensuring robustness, safety, user experience, and human-machine interaction in AVs. Overall, our AIMS project successfully contributed to the objectives set in the introduction by providing insights into RL models and algorithms for AVs and designing an effective RL-based control system. The future work outlined aims to further improve the system's capabilities and address emerging challenges in the field of autonomous vehicle control

Acknowledgment

We would like to express our sincere gratitude to Dr. Khaled Tolba and Eng. Mostafa Ghaith for their invaluable contributions and unwavering support throughout my research. Dr. Tolba's exceptional mentorship, profound knowledge, and critical feedback have shaped the direction of this study, while Eng. Ghaith's technical expertise and assistance have enhanced its quality. Their constant availability, provision of resources, and belief in our capabilities have been instrumental in the successful completion of this research project. We are deeply grateful for the opportunity to work with such esteemed professionals and mentors, and their guidance has profoundly influenced my academic and professional growth.

References

[1] A. Dolatabadi, H. Abdeltawab and Y. A. -R. I. Mohamed, "A Novel Model-Free Deep Reinforcement Learning Framework for Energy Management of a PV Integrated Energy Hub," in IEEE Transactions on Power Systems, 2022, doi: 10.1109/TPWRS.2022.321293

[2] M. Doody, M.M.H. Van Swieten, and S.G. Manohar, "Model-based learning retrospectively updates model-free values," *Scientific Reports*, vol. 12, p. 2358, 2022, doi:10.1038/s41598-022-05567.

[3] D. Samarasinghe, M. Barlow and E. Lakshika, "Flow-Based Reinforcement Learning," in IEEE Access, vol. 10, pp. 102247-102265, 2022, doi: 10.1109/ACCESS.2022.320926

[4] M. Ye, C. Tianqing, and F. Wenhui, "A single-task and multi-decision evolutionary game model based on multi-agent reinforcement learning," in *Journal of Systems Engineering and Electronics*, vol. 32, no. 3, pp. 642-657, June 2021, doi: 10.23919/JSEE.2021.00005.

[5] A. Guillen-Perez and M.-D. Cano, "Learning From Oracle Demonstrations—A New Approach to Develop Autonomous Intersection Management Control Algorithms Based on Multiagent Deep Reinforcement Learning," in *IEEE Access*, vol. 10, pp. 53601-53613, 2022, doi: 10.1109/ACCESS.2022.3175493.

[6] G.-P. Antonio and C. Maria-Dolores, "Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections," in *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7033-7043, July 2022, doi: 10.1109/TVT.2022.3169907.

[7] S. Aradi, "Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740-759, Feb. 2022, doi: 10.1109/TITS.2020.3024655.

[8] M. Kaloev and G. Krastev, "Experiments Focused on Exploration in Deep Reinforcement Learning," in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey, 2021, pp. 351-355, doi: 10.1109/ISMSIT52890.2021.9604690.

[9] M. Masmoudi, H. Friji, H. Ghazzai, and Y. Massoud, ”A Reinforcement Learning Framework for Video Frame-Based Autonomous Car-Following,” in *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 111-127, 2021, doi: 10.1109/OJITS.2021.3083201.

[10] Z. Shi, W. Ma, S. Yin, H. Zhang, and X. Zhao, ”Interactive Reinforcement Learning Strategy,” in *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, Atlanta, GA, USA, 2021, pp. 507-512, doi: 10.1109/SWC50871.2021.00075.