International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Enhancing Autonomous Vehicles Using Deep Reinforcement Learning

M. Rithika, K.Indu Reddy, B.Charan Reddy , P. Jayanth Reddy, T. Vamshi Krishna, Prof.Suchitra Pattabirama

School of Enginnering, Department of AI&ML, Malla Reddy University, Hyderabad – 500043, Indias

***Abstract*— Our project focuses on enhancing collision avoidance between autonomous vehicles using deep reinforcement learning (DRL), and it also aims to improve the collision avoidance between autonomous vehicles using deep reinforcement learning. By leveraging advanced DRL algorithms and ensuring robust, realtime decision-making, the project seeks to create a safer and more efficient autonomous driving system. The approach integrates various state representations, actions, and reward functions to optimize driving behaviors. The system's state representation includes vehicle positions, velocities, distances to obstacles, and relative locations of other vehicles. Actions encompass accelerating, decelerating, and turning maneuvers. The reward function is designed to promote safe driving behaviors, such as maintaining speed and lane discipline, while penalizing collisions and erratic movements. The project also incorporates Multi-Agent Reinforcement Learning (MARL) to enable vehicle coordination, where each vehicle learns to maximize its own reward while considering the actions of others. Communication protocols between vehicles enhance decision-making and collision avoidance here vehicles share information about their states and intended actions, improving overall coordination and reducing the likelihood of collisions.Safety and robustness are ensured by integrating safe reinforcement learning techniques and making the policy resilient to environmental uncertainties. Real-world challenges addressed include transferring learned policies from simulation to real-world applications, ensuring scalability across different traffic conditions, and compliance with traffic regulations. The effectiveness of the proposed solution is demonstrated through case studies and validation in various driving scenarios. This project aims to advance the development of safer and more efficient autonomous driving systems.**

I. INTRODUCTION

TRAFFIC control is changing rapidly, as Connected Autonomous Vehicles (CAVs) are bringing new opportunities to control and manage vehicular, people, and goods flow in and around our cities. The new Intelligent Transportation Systems (ITS) are challenged to provide new ways to control CAVs to reduce congestion, pollution, or accidents [1]. Therefore, improving and introducing new control strategies is imperative for efficient traffic management decisions. In recent years, numerous approaches have been developed to implement CAVs control algorithms, however, this task is really complex and requires knowledge of the state of all actors involved in the traffic system (vehicles, pedestrians, priority vehicles, etc.).

Autonomous Intersection Management (AIM) systems are designed to efficiently manage CAVs at urban intersections, eliminating collisions, and optimizing overall traffic flow [2]. AIMs regulate the flow of vehicles through intersections by acting on their state (speed, acceleration, braking, steering, etc.). This control is usually based on simple rules and the intersection's current state, without considering other vehicle-specific parameters, environmental conditions, upcoming events, etc. [3], [4].

Deep Reinforcement Learning (DRL) successfully connects Reinforcement Learning (RL) algorithms with the strengths of Deep Neural Networks (DNN), accelerating these RL algorithms' training processes and performance. As a consequence of this success [5], DRL is being introduced in many areas. In Multi-Agent (MA) environments, multiple *agents* execute *actions* and can affect the *states* of other *agents*. Traditional MA-RL algorithms have recently been successfully extended with DNNs for MA DRL learning, giving rise to Multi-Agent DRL (MADRL). The reason lies in the availability of high computational power and the efficiency of distributed algorithms, leading to unexpected impressive results such as those obtained by DeepMind [6] and OpenAI [7].

Due to the advantages that MADRL can offer for cleverly finding a cooperative control policy, we decided to explore this path. In this work, we detail a new AIM system based on MADRL, called advanced Reinforced AIM (*adv.*RAIM), and its performance is extensively evaluated in a variety of realistic and complex scenarios. The proposed *adv.*RAIM is trained by DRL and uses end-to-end MADRL, along with other advanced methods such as Curriculum through *Self-Play* learning and Prioritized Experience Replay (PER), to learn and model the complex dynamics of the environment in the control of CAVs at urban intersections. The final goal of *adv.*RAIM is to periodically act on the *speed* of all CAVs collectively at intersections to reduce lost time, by eliminating collisions and traffic lights. To the authors' knowledge, this paper addresses the use of end-to-end MADRL in the field of AIM for the first time. Simulation results show that the performance of *adv.*RAIM is remarkably superior to other traditional traffic light control algorithms (like Fixed Time (FT) or *iREDVD* [8]). Furthermore, when compared to other recently proposed AIMs [9], *adv.*RAIM can reduce waiting time by 88%, and time loss by 55%, among

Manuscript received May 24, 2021; revised January 25, 2022 and April 18, 2022; accepted April 19, 2022. Date of publication April 25, 2022; date of current version July 18, 2022. This work was supported in part by MCIN/AEI/10.13039/501100011033 under Grant PID2020-116329GB-C22, and in part by the Fundación Séneca, Región de Murcia, Spain under Grant 20740/FPI/18. The review of this article was coordinated by Dr. Bilal Akin.

other metrics. This demonstrates the multiple advantages of MADRL to develop increasingly intelligent AIMs, which can provide advanced control policies and achieve smarter CAVs. Moreover, they can greatly surpass in control complexity the currently proposed AIMs, where they usually only allow straight or right turns, single-lane intersections, or very low vehicular flows.

A tentative version of this work was previously presented in [10], which served as a basis for the development of RAIM and to demonstrate that RL-based AIM could offer advantages over traditional control techniques. The present work adds significant aspects to the initial version.

- First, RAIM has been enhanced with a recurrent module (LSTM) to eliminate the problem of variation in the shape of the variable observations as a function of the number of vehicles. In addition, thanks to the nature of LSTMs, we can capture the long-term spatial and temporal dynamics of traffic conditions in the network. With this recurrent module, the speed calculation for each vehicle considers all other vehicles at the intersection.
- Secondly, the complexity of the simulation scenario has been considerably increased from a maximum flow of 450 veh/h/lane to 1200 veh/h/lane and from 2 lanes to 3 lanes per direction, which increases exponentially the complexity and training time, but allows maximizing the advantages offered by RL over traditional and other AIM techniques. In addition, each simulated vehicle had different characteristics within a random range of acceleration, shape, fuel consumption, etc., which offered additional complexity in learning how to model the simulated environment.
- Finally, the comparison of results is extended to more recently published algorithms, such as an intelligent traffic light control system (*iREDVD* [8]) and an already proposed AIM [9], and we confirm that our model continues to outperform existing approaches using different evaluation metrics. Furthermore, considerable new analysis and intuitive explanations are added to the training curves and testing results.

The rest of this paper is organized as follows. Section II provides an overview of the operating principles of AIM. Section III shows the state of the art of AIM. Section IV describes the system proposed in this paper. The simulator and parameters used are shown in Section V. Section VI includes the performance results obtained both in the training process and in a test scenario. Finally, the conclusions are summarized in Section VII.

II. Autonomous Intersection Management

Intersections are responsible for regulating the right-of-way of vehicles to control traffic flow, reduce accidents, and improve travel time, which is usually done with traffic lights, or traffic signals, in urban areas. With the arrival of CAVs, it requires a new way of controlling vehicles as a whole [11], more efficient and sophisticated than traditional techniques, allowing inefficient traffic lights to be eliminated.

AIM emerges as a new approach to building intelligent systems that can deal with the complex dynamics of real-life

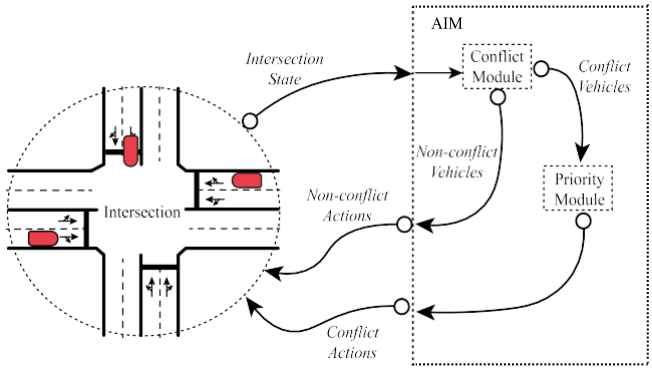

Fig. 1. AIM basic operation. AIM includes a *Conflict module* and a *Priority module* to control AV [10].

and control CAVs' state (speed, acceleration, steering, etc.) at intersections to provide the highest security level, increasing flow while increasing flow and decreasing time loss [12]. Traditionally, these AIMs are based on two modules, one dealing with conflict prediction and the other with the resolution of expected conflicts.

LITERATURE SURVEY

This module is responsible for deciding whether, or not, there will be conflicts between two vehicles when approaching or crossing the intersection. It follows a series of rules so that it can predict the routes that vehicles will take within the intersection along space-time and check if there are conflicts. That is, when two or more vehicles coincide temporally and spatially, this component identifies a conflict. The basic operation of AIM with the conflict module and priority module can be seen in Fig. 1.

This module can follow several approaches to conflict identification: *i) intersection-based* [12]–[14], *ii) tile-based* [15]–[18], *iii) conflict point-based* [9], [19]–[22], and *iv) vehicle-based* [23]–[26]. A representation of each approach can be seen in Fig. 2.

The first proposed approach laid the foundation for AIM [12]. This approach (*intersection-based*) does not allow more than one vehicle to be inside the intersection at the same time, regardless of the route the vehicles take. This option, while very simple, has multiple obvious disadvantages.

A more elaborated approach is the *tile-based* [18], which creates a mesh within the intersection, and vehicles cannot coincide in the same mesh cell simultaneously along their trajectory.

The *conflict point-based* [9] only takes into account the spots where the trajectories of the vehicles within the intersection overlap. This dramatically reduces the complexity of optimization tasks, but due to the variable geometry of the vehicles, unexpected accidents may occur, a situation that can never occur. Finally, the *vehicle-based* [26] approach offers vehicles total freedom of movement within the intersection. Here, vehicles are free to choose the route they take to reach their exit lane. The latter option is undoubtedly the one that offers the most freedom, but it requires enormous computing power since it becomes a multidimensional and multiagent problem of vast complexity.

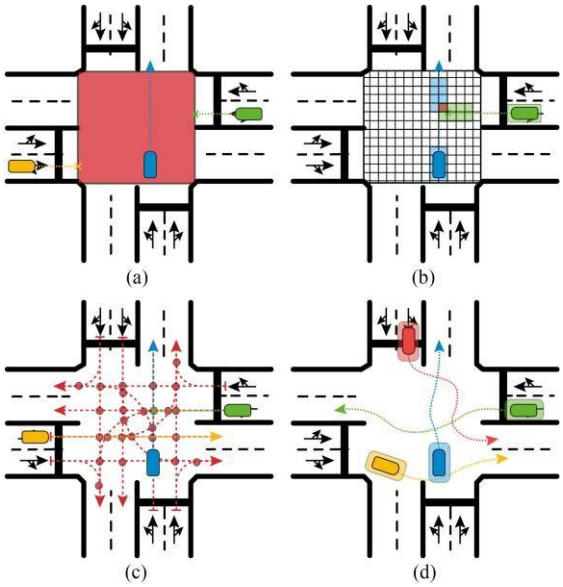

Fig. 2. Approaches developed for the conflict module of AIM.

PROBLEM STATEMENT

When conflicts are encountered, the priority module resolves them by acting on the vehicles' state (e.g., speed, acceleration, route, etc.) and managing the vehicles' right-of-way. This module is responsible for ensuring that the travel time of the vehicles is reduced most fairly, ensuring that no vehicle is stuck infinitely. Seeking to assign priorities to vehicles when crossing, this module can give the right-of-way of vehicles in several manners: *i)* based on the order of arrival at the intersection, with First-Come First-Served (FCFS) [12], [19], [27], [28]; *ii)* assigning priorities based on vehicle/intersection status, such as Fast First Service (FFS) [9] where vehicles arriving at the intersection fastest are given the highest priority, or Long Queue First (LQF) [17] where those vehicles with the longest entry queue have the highest priority; *iii)* using some heuristics like Dynamic Programming (DP) or Linear Mixed Integer Programming (MILP) where given a series of equations and conditions is used to solving them [15], [22], [26], [29]–[32]; however, this method requires a huge computational load every time a solution is required, and when sudden changes occur, a solution has to be obtained again from scratch, which increases the complexity to solve the problem in an almost exponential way and the complexity is not acceptable for real-time systems; *iv)* by auctions [13], [33] with higher priority being given to those vehicles with the highest bids, creating a market economy with the currency used for auctioning and generating problems of equality; *v)* or through artificial intelligence mechanisms such as genetic algorithms [34] or RL [17].

III. State of the Art

Having seen the principle of operation of the different AIMs, in this section we will look at the proposed works, as well as their benefits, drawbacks, and performance. The work presented by Stone *et al.* [12] was based on right-of-way reservations, following a policy based on FCFS, and began the develop- ment of these systems, which demonstrated that, in certain situations, the control protocol they proposed outperformed the traditional traffic light control protocols. Further, multiple variants of this work were presented that allowed the incorpo- ration of non-autonomous vehicles (FCFS-light) [2], [35], as well as emergency vehicles such as ambulances or police cars (FCFS-EMERG) [36]. The advantages offered by FCFS were a reduction in travel time of up to 80% compared to traffic lights and stop signals.

Another interesting work on AIM was proposed by [13], where it presented an auction-based reserve approach. These auctions were used to determine the order in which vehicles pass through, that is, within the priority module. The vehicles that bid the most were passed first. The results shown in four urban cities showed superior performance in three of the four simulated road networks when compared to traditional mechanisms, as well as when compared to FCFS. However, this mechanism presents several serious problems. The main problem is that the intrinsic problem of any auction mechanism is vehicle starvation, in the sense that the auction strategy may prevent others from winning, with the risk that they will experience indefinitely long waiting times, as well as generate a market economy of the currency used, inflation, discrimination, etc.

Using DRL, an AIM was presented in [17] where DRL is used in the priority module to create a *Q*-table with all possible combinations of vehicles per entrance and the best car to pass. This work offers improvements of more than 30% compared to FCFS and LQF, but however, very extensive training is required. Aside from creating all possible combinations of vehicles in the entry, you must find the best vehicle for each case, something that, for real situations, can take an enormous amount of training time. Nevertheless, the advantage of RL is that when such a policy has been found, the inference is extremely fast. However, in the work proposed by Levin *et al.* [37] it was shown that AIM has much room for improvement, since, in realistic ex- amples, conventional traffic systems were able to outperform the reservation-based systems proposed to date. To test this, FCFS was compared with a traditional traffic light system. In situations where vehicle flow is low, FCFS provided better performance, but when traffic is high (> 800 veh/h) traffic light control provided better performance. In addition, when traffic is asymmetric, in bursts, or there is a main avenue and streets connecting to it, the performance of FCFS was worse than that of traffic light control.

It's evident that having more control over autonomous ve- hicles, both individually and collectively, gives these systems a huge advantage over traditional control techniques in terms of increasing vehicle flow, avoiding accidents, and shortening vehicle travel times. However, the functioning of these modules can have serious disadvantages compared to traditional traffic light control techniques as they are based on simple control techniques. These disadvantages have been previously detailed in [37], where it is demonstrated that in the case of unstudied situations, the systems' behavior becomes unstable and obtains unexpected results. Furthermore, these techniques are incapable of considering past events or anticipating future ones.

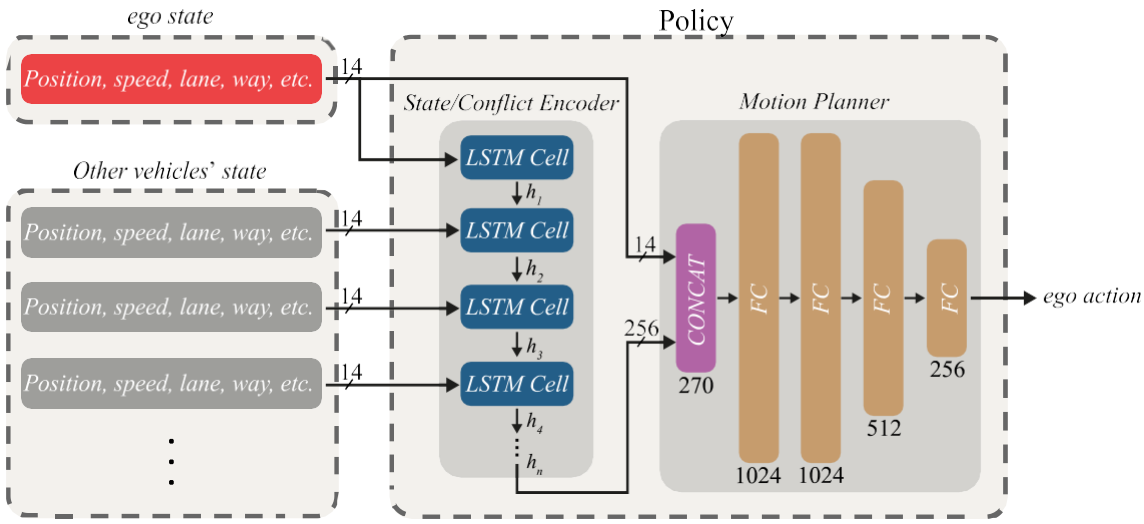

Fig. 3. New advanced RAIM (*adv*.RAIM) network. The action to be performed by the *ego* vehicle is calculated in the Policy. The output is the normalized speed that the *ego* vehicle must follow in the next timestep. Note that there is only one *LSTM cell* that is iteratively fed with the features of each vehicle (14), starting with the *ego* vehicle's state, and continuing with other vehicles' state. The State/Conflict Encoder output (h_x) was set to 256 hidden parameters.

IV. Advanced Reinforced AIM – adv.RAIM

Considering the enormous potential offered by AIM and the challenges that MADRL can address, in this work we proposed advanced Reinforced AIM (*adv*.RAIM). This new approach brings together the properties of the MADRL field with those of AIM. *adv*.RAIM can offer a new approach within AIM, opening an original path for the development of other advanced AIM solutions.

Our prior work [10] showed that RAIM offers a great advantage over the previously proposed AIM in simple scenarios. In addition, RAIM was able to adapt to the different conditions that may arise as well as, once trained, being able to infer a result extremely quickly. Furthermore, the preliminary findings suggested that RAIM could outperform other traffic control systems in more realistic scenarios than those shown in the previous.However, the main problem of RAIM was that it could only take into account 32 vehicles at a time, using a zero-filling approach when faced with fewer vehicles and ignoring them when there were more than 32 vehicles. To solve this problem, the proposal we made is to use a recurrent network (Long-Short Term Memory, LSTM) in which the features of each vehicle are fed into the input and encoding of the conflicts between the vehicle to be controlled and the other vehicles is obtained at the output. This module is called *State/Conflict encoder* and can be seen in Fig. 3, where an LSTM cell is used to which all the vehicle states are recurrently input, and an encoded value of the conflicts is learned during the training process.

The LSTM cell has the advantage of being able to learn long-term dependencies [38], i.e., between different vehicles depending on their state, since the feedback mechanism allows it to remember previous states of the vehicles. In addition, the output of the LSTM is a fixed-dimensional vector, eliminating the problem that RAIM had. An output size of 256 variables was used to allow encoding as much information as possible without restricting the information learned. This is the first time we have used an LSTM cell in a RAIM approach, and it is also a novel approach to conflict-based controller design.

As for the order in which the module is fed, the state variables of the ego vehicle are fed first, followed by those of the other vehicles, in the order of their distance from the center of the intersection. This allows learning the state variables in the local neighborhood when a conflict occurs (fed by the reward signal). The motivation for learning in this way is that it is easier to feed the data in a way that considers the different states in which there would be a conflict, or in which there would be a large impact on the RL information about a given vehicle state, thus increasing the reward for learning to encode conflicts.

After the *State/Conflict encoder* module, *adv*.RAIM presents a set of fully connected layers, which compose the Motion Planner module, see Fig. 3. This module decides the normalized speed to be carried by each CAV at the next timestep based on its features and the output of the state/conflict encoder to avoid collisions and optimize the traffic flow. This module was composed of 4 layers of fully connected neurons with ReLU activation functions and the number of neurons in each layer is shown in Fig. 3. *adv*.RAIM is termed as an *ego-centric* multi-agent system, having to deal with all the CAVs at the intersection simultaneously, but controlling each CAV individually. That is, *adv*.RAIM considers the current state of the vehicle (*ego*) and the other vehicles to obtain the normalized speed of the *ego* vehicle at the next time step.The action space is the normalized speed between 0 and 1 that the *ego*-vehicle must follow in the following time interval (labeled *ego*-action in Fig. 3). The speed was denormalized considering a maximum road speed of 13.9 m/s (= 50 Km/h). Each CAV has internal constraints of maximum accelerations and decelerations given by the simulation tool that it will be employed (typical values of 2.6 and 4.5 m/s^2), so each vehicle performs the indicated actions considering these speed change constraints.

TABLE VIII
TESTING SCENARIO 3 (FIXED 1200 VEH/H/LANE) ADDITIONAL RESULTS

	Algorithm	CO emiss. (g)	CO_2 emiss. (g)	HC emiss. (mg)	PMx emiss. (mg)	NOx emiss. (mg)	Fuel cons. (ml)	Elect. cons. (W)
Traffic Lights	*FT10*	5.90 ± 0.36	106.71 ± 22.62	21.74 ± 3.01	14.03 ± 1.21	298.17 ± 66.16	49.65 ± 6.71	48.11 ± 5.49
	FT15	5.28 ± 0.42	101.51 ± 20.21	19.46 ± 3.50	12.69 ± 1.10	237.48 ± 64.03	43.37 ± 6.30	46.53 ± 5.01
	FT20	5.56 ± 0.31	95.03 ± 18.04	19.73 ± 2.80	13.00 ± 1.13	237.41 ± 62.05	44.78 ± 6.06	45.86 ± 3.58
	FT30	5.71 ± 0.45	99.56 ± 23.08	20.44 ± 2.55	13.41 ± 1.12	263.75 ± 30.30	46.59 ± 5.88	46.14 ± 3.03
	iREDVD [8]	5.26 ± 0.26	94.87 ± 24.27	19.09 ± 2.43	12.70 ± 1.08	195.32 ± 54.27	42.69 ± 5.59	42.97 ± 4.05
Qian et al. [9]		5.22 ± 0.14	94.56 ± 19.88	18.50 ± 1.93	12.62 ± 0.83	190.56 ± 36.42	42.26 ± 3.75	40.13 ± 2.38
RAIM [10]		5.23 ± 0.17	94.12 ± 16.18	18.48 ± 1.18	12.57 ± 0.62	186.10 ± 53.08	41.75 ± 5.69	39.68 ± 3.67
adv.RAIM		**5.21 ± 0.15**	**93.81 ± 14.41**	**18.44 ± 1.01**	**12.54 ± 0.57**	**183.30 ± 16.16**	**41.20 ± 4.87**	**39.46 ± 1.81**

No collisions were recorded. [*mean* ± *std.* of 10 simulations].

TABLE IX
TESTING SCENARIO 4 (VARIABLE FLOW RATE) RESULTS

	Algorithm	Travel Time (s)	Waiting Time (s)	Time loss (s)
Traffic Lights	*FT10*	72.75 ± 7.44	30.91 ± 4.91	43.77 ± 9.88
	FT15	54.67 ± 8.11	17.59 ± 3.74	25.58 ± 6.12
	FT20	56.04 ± 6.91	19.02 ± 3.89	27.05 ± 7.91
	FT30	60.56 ± 7.92	23.56 ± 4.98	31.27 ± 8.89
	iREDVD [8]	43.23 ± 6.11	10.14 ± 3.33	18.64 ± 4.41
Qian et al. [9]		32.49 ± 3.39	2.55 ± 0.83	4.87 ± 1.22
RAIM [10]		31.44 ± 2.71	1.86 ± 0.94	3.08 ± 1.12
adv.RAIM		**29.88 ± 3.01**	**1.14 ± 0.71**	**2.16 ± 0.46**

No collisions were recorded. [*mean* ± *std.* of 10 simulations].

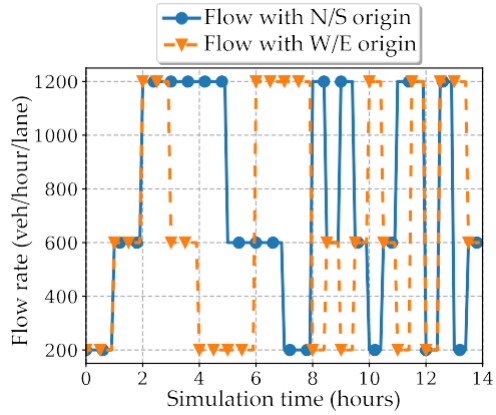

Fig. 5. Vehicle flow rate per lane used in the testing scenario.

and fast flow variations. This allowed us to test *adv.*RAIM in a large number of conditions as close to reality as possible, as well as to see the evolution of performance in different isolated scenariosSeveral metrics were studied. Due to the optimization tech- nique, the metrics directly optimized and studied were travel time, waiting time, and time loss due to congestion. Waiting time refers to the time in which the vehicle speed was less than or equal to 0.1 m/s. This time can be due to a variety of factors, including congestion. Although the travel time and time loss due to congestion metrics are directly related, we left both to show in perspective the time loss due to congestion in relation to the total travel time. Indirectly, pollution and consumption metrics (CO, CO_2, HC, PMx, NOx, and fuel and electricity) were analyzed to show the environmental benefits that these systems can offer, in addition to the reduction in travel time and congestion. The traditional traffic light algorithms used for comparison were: fixed time algorithm *(FT)* and *iREDVD* algorithm [8].

The FT algorithm sets a fixed passing priority time for each of the branches of an intersection and only allows vehicles from one branch to pass at a time. Several passing priority times were tested, 10, 15, 20, and 30 seconds (total cycle lengths of 40, 60, 80, and 120 seconds). They were named FT10, FT15, FT20, and FT30. *iREDVD* is an adaptive algorithm based on queuing theory and traffic lights. RAIM was also compared with the AIM approach developed by Qian *et al.* [9]. The vehicle distribution used was: 35% of diesel cars, 35% of gasoline cars, and 30% of electric cars with zero emissions.

RESULTS

The following section highlights the results obtained in the test scenario, along with a detailed comparative analysis of the test scenario results.

A. Training Scenario

Fig. 6 shows the results obtained in the training scenario in the studied metrics (number of collisions, reward, and time loss) versus the simulated vehicle flow throughout all simulations. One of the main quick observations is the stability of the system. This is especially noticeable at the peak of the first simulations in the average number of collisions metric (Fig. 6a) and is mitigated by the automatic *Self-Play* curriculum and RL nature. There are some outliers in the metrics as the flow increases. However, they eventually converge to stable values, demonstrating that TD3 and PER allow training to converge with increasing complexity. In addition, it is worth noting that the number of collisions shows a downward trend from the peak in the initial 750 simulations approximately, due in part, to the large negative reward when a collision occurs. Finally, the number of collisions can be seen to trend to 0 and presents a very low value from simulation 7000 onwards.The average reward per vehicle metric (Fig. 6b) also shows a negative trend, but acceptable stability within the confidence intervals. This negative trend is because the number of simulated vehicles increases over time, making the intersection increas- ingly congested. This causes vehicles (on average) to drive progressively slower, but optimally to maximize the average reward received by each vehicle.

CONCLUSION AND FUTURE WORK

The fields of robotics, CAVs, and ITS are advancing rapidly by virtue of MADRL, which provides a flexible and efficient way to solve complex and extreme optimization problems in these areas. This paper presents and evaluates *adv*.RAIM, a new and inspiring approach in AIM based on MADRL. *adv*.RAIM periodically controls the speed of CAVs passing through an intersection in a cooperative and decentralized manner, ensuring safety and maximum fluidity. *adv*.RAIM presents an architecture with an LSTM capable of capturing the long-term spatial and temporal dynamics of traffic conditions in the network. This allows it to better understand and encode possible collisions in space/time between different CAVs passing through an intersection and thus act proactively. In addition, apart from the LSTM module, it presents a module composed of deep neural networks in charge of crossing the collision information encoded by the LSTM module and the state of the CAV to be controlled, obtaining the speed at which the CAV should circulate during the following time interval. The control process is performed sequentially and periodically for all CAVs.

The results show that *adv*.RAIM is able to overcome some important disadvantages of traditional AIMs (performance loss when the vehicular flow is heavy), controlling challenging scenarios and achieving robust results through the coexistence of RL techniques such as TD3, PER, and *Self-Play* curriculum-based training techniques.

Quantitatively, the results show an improvement in several metrics, such as a reduction in travel time by 59%, or a reduction in time loss by 95% in the most complex scenario. The intensive training and the capability of operating proactively can explain the good outcomes obtained. Moreover, thanks to the nature of the optimization, *adv*.RAIM is able to obtain a control policy capable of indirectly optimizing other very important metrics such as fuel/energy consumption or pollutant gas emissions, due to the smaller number of accelerations/decelerations of the CAVs. Furthermore, the modularity of *adv*.RAIM could be an advantage to explore its use in other scenarios such as highways or sub-urban areas.

As future work, we will address some improvements such as incorporating a Transformer-based attention mechanism to identify conflicts, the crossing order of vehicles, or the exchange of information between intersections to increase collective intelligence.

REFERENCES

[1] S. Djahel,R. Doolan, G.M.Muntean,and J.Murphy,"Acommunications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Commun. Surv. Tut.*, vol. 17, no. 1, pp. 125–151, Jan.–Mar. 2015.

[2] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.

[3] N. Aloufi and A. Chatterjee, "Autonomous vehicle scheduling at intersections based on production line technique," in *Proc. IEEE Veh. Technol. Conf.*, 2018, pp. 1–5.

[4] S. Mariani, G. Cabri, and F. Zambonelli, "Coordination of autonomous vehicles: Taxonomy and survey," *ACM Comput. Surv.*, vol. 54, pp. 1–33, 2020.

[5] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[6] O. Vinyals *et al.*, "Grandmaster level in starcraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.