Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections

Guillen-Perez Antonio and Cano Maria-Dolores, *Senior Member, IEEE*

***Abstract*—In recent years, the growing development of Connected Autonomous Vehicles (CAV), Intelligent Transport Systems (ITS), and 5G communication networks have led to the advent of Autonomous Intersection Management (AIM) systems. AIMs present a new paradigm for CAV control in future cities, taking control of CAVs in scenarios where cooperation is necessary and allowing safe and efficient traffic flows, eliminating traffic signals. So far, the development of AIM algorithms has been based on basic control algorithms, without the ability to adapt or keep learning new situations. To solve this, in this paper we present a new advanced AIM approach based on end-to-end Multi-Agent Deep Reinforcement Learning (MADRL) and trained using Curriculum through *Self-Play*, called advanced Reinforced AIM (*adv.*RAIM). *adv.*RAIM enables the control of CAVs at intersections in a collaborative way, autonomously learning complex real-life traffic dynamics. In addition, *adv*.RAIM provides a new way to build smarter AIMs capable of proactively controlling CAVs in other highly complex scenarios. Results show remarkable improvements when compared to traffic light control techniques (reducing travel time by 59% or reducing time lost due to congestion by 95%), as well as outperforming other recently proposed AIMs (reducing waiting time by 56%), highlighting the advantages of using MADRL.**

***Index Terms*—Autonomous intersection management, connected autonomous vehicles, deep reinforcement learning, intelligent transport systems, intersection traffic management, multi-agent deep reinforcement learning.**

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.

The authors are with the Information Technologies and Communication Department, Universidad Politécnica de Cartagena, 30203 Murcia, Spain (e-mail: antonio.guillen@edu.upct.es; mdolores.cano@upct.es).

Digital Object Identifier 10.1109/TVT.2022.3169907

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

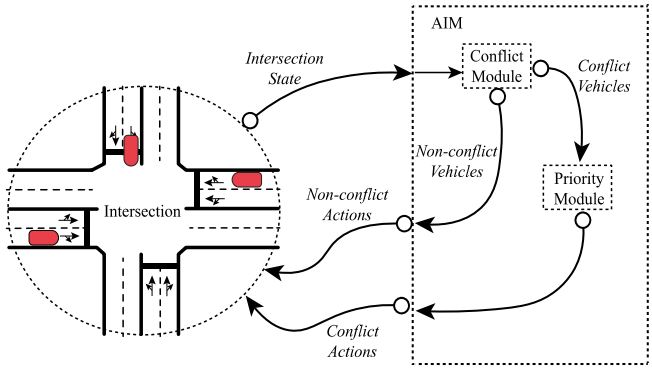

Fig. 1. AIM basic operation. AIM includes a *Conflict module* and a *Priority module* to control AV [10].

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.

A. Conflict Module

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.

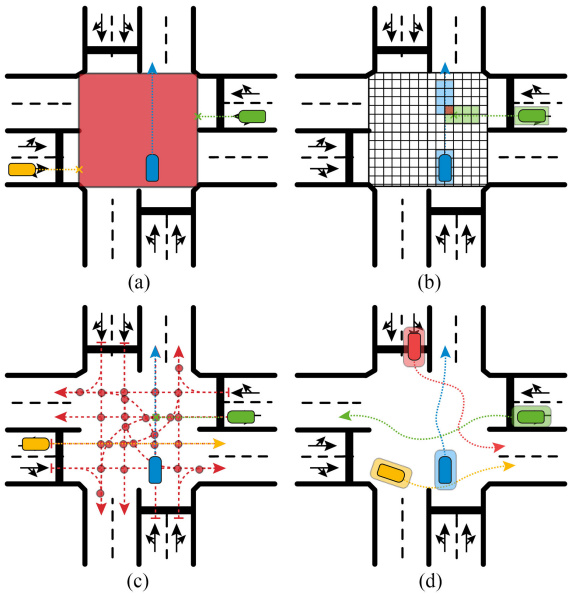

Fig. 2. Approaches developed for the conflict module of AIM.

B. Priority Module

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 development 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 incorporation 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 examples, 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 vehicles, 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.

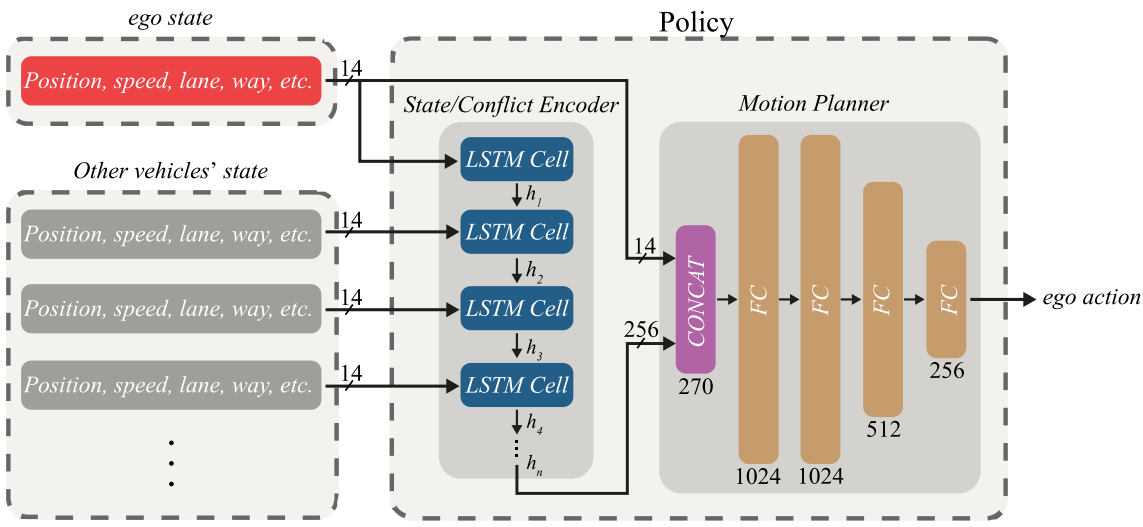

Fig. 3. New advanced RAIM (*adv.*RAIM) network. The action to be performed by the *ego* vehicle is calculated in the Policy. The output is the normalized speed that the *ego* vehicle must follow in the next timestep. Note that there is only one *LSTM cell* that is iteratively fed with the features of 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 I
INPUT FEATURES AND MEANING

Input Variable	Meaning	Range
relative_pos_x	Relative position to the center of the intersection on the x-axis.	[0, 100]
relative_pos_y	Relative position to the center of the intersection on the y-axis.	[0, 100]
Speed	Vehicle speed.	[0, 13.9]
Angle	Vehicle orientation angle.	[0, 360]
Lane	Lane of approach (left lane, center lane, right lane).	One-hot
Way	Way the vehicle will follow (right turn, straight path, left turn).	One-hot
Queue	Intersection branch through which the vehicle is approaching (North, South, East, West).	One-hot

TABLE II
SUMMARY OF THE SIMULATION SETUP AND RAIM PARAMETERS

Simulator	Simulation step	0.25 segs
	Flow step	100 veh/hour
	Min flow	200 veh/hour
	Train duration	5 mins/simulation
	Test duration	840 mins/simulation
	Scenario	4 branches, 3 lanes/way, and all ways.
	Control distance	100m
RAIM	Batch size	128
	Gamma	0.99
	Tau	4×10^{-3}
	Learning rate actor	1×10^{-5}
	Learning rate critics	1×10^{-4}
	Learn Batches	200
	Weigh decay	1×10^{-6}
	Optimizer	Adam
	Policy noise	0.2
	Policy noise clip	0.3
	Optimizer epochs	3
	TD3 update actor every	2
	PER Buffer Size	$2^{20} \approx 1\times10^{6}$

TABLE III
TESTING SCENARIO 1 (FIXED 200 VEH/H/LANE) RESULTS

	Algorithm	Travel Time (s)	Waiting Time (s)	Time loss (s)
Traffic Lights	*FT10*	*25.85 ± 6.25*	*8.31 ± 0.94*	*12.49 ± 1.16*
	FT15	*29.43 ± 6.46*	*11.28 ± 1.15*	*15.12 ± 1.88*
	FT20	*36.85 ± 6.11*	*15.71 ± 1.67*	*21.91 ± 2.41*
	FT30	*38.17 ± 5.39*	*18.11 ± 1.89*	*25.89 ± 2.70*
	iREDVD [8]	*22.99 ± 5.87*	*2.09 ± 1.03*	*4.62 ± 1.36*
Qian et al. [9]		*18.48 ± 5.04*	*0.81 ± 0.39*	*1.51 ± 0.38*
RAIM [10]		*18.64 ± 4.83*	*0.57 ± 0.30*	*0.93 ± 0.36*
adv.RAIM		***17.67 ± 4.94***	***0.38 ± 0.21***	***0.69 ± 0.16***

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

The state variables of each vehicle used as input parameters to *adv.*RAIM are shown in Table I. After coding the input parameters, each vehicle provides *14* features. The parameters with a continuous range (such as position, speed, and angle) are normalized within the range of [-1, 1], and the parameters with discrete values (i.e., lane, way, queue) are encoded with one-hot encoding.

Algorithm 1: adv.RAIM.

Input: State of all vehicles
Output: Speed of each vehicle in the following timestep

```
# Reset the environment.
for timestep t ∈ {0, ..., Max_episodie} do:
    # Obtain the vehicles currently being simulated.
    vehicles = sim.get_current_vehicles()
    # For each vehicle, the desired speed for the next
    timestep is obtained.
    # This loop is repeated for each vehicle in the
    intersection.
    # Clear input_params.
    for vehicle ego_veh ∈ vehicles do:
        # Obtain the params of ego_veh (position, speed,
        lane, etc.) and append them to input_params.
        vehicles_added = 0
        # Obtain the params of the other vehicles.
        for vehicle veh ∈ vehicles do:
            # Obtain the params of veh and add to
            input_params.
        end for
        # Obtain new speed of ego_veh using
        input_params and actor net of TD3 and set
        new_speed.
        new_speed = actor_net(input_params)
        sim.set_veh_speed(ego_veh, new_speed)
    end for
end for
```

Therefore, the observation space for the entire control policy, Fig. 3 (Policy), is formed by the 14 characteristics of the *ego*-vehicle plus the $n \times 14$ characteristics of the other vehicles, where n is the number of other vehicles with which the *ego*-vehicle must cooperate to ensure the safety and smooth flow of traffic.

The control timestep was set to 250 ms, considering that it is a sufficient time interval for efficient control without overloading the processing and allowing real-time control. This means that every 250 ms *adv*.RAIM updates the speed that all vehicles at the intersection must follow to ensure safety and maximum flow, considering the updated status of all vehicles. The pseudocode of RAIM is shown in Algorithm 1 and the new *adv.*RAIM network can be seen in Fig. 3. To optimize the controller, we use Twin Delayed Deep Deterministic Policy Gradients (TD3) [39]. TD3 is an improvement of the Deep Deterministic Policy Gradient (DDPG) algorithm, based on Actor/Critic control approach. It has become one of the most popular algorithms for continuous control problems within robotics and autonomous driving fields. In this approach, there are two types of neural networks, one that aims to predict the action to be taken (in this case, the normalized speed of *ego* vehicle) based on the state (*Actor*, learn the policy) and another that seeks to predict the expected reward of performing that action (*Critic*, learn the *action-value function*, *Q*-function).

TABLE IV
TESTING SCENARIO 1 (FIXED 200 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*	*0.68 ± 0.11*	*30.20 ± 5.66*	*4.15 ± 0.76*	*0.80 ± 0.31*	*82.05 ± 14.55*	*9.92 ± 1.52*	*12.39 ± 0.90*
	FT15	*0.72 ± 0.13*	*31.91 ± 5.07*	*4.38 ± 0.89*	*0.88 ± 0.30*	*95.61 ± 16.03*	*10.36 ± 1.59*	*13.43 ± 1.26*
	FT20	*0.79 ± 0.08*	*32.26 ± 4.50*	*4.45 ± 0.74*	*0.95 ± 0.27*	*96.89 ± 15.53*	*10.71 ± 1.52*	*13.50 ± 0.91*
	FT30	*0.83 ± 0.12*	*33.40 ± 5.78*	*4.64 ± 0.67*	*0.98 ± 0.31*	*103.46 ± 15.11*	*11.19 ± 1.49*	*14.53 ± 0.82*
	iREDVD [8]	*0.45 ± 0.08*	*28.21 ± 6.09*	*3.28 ± 0.60*	*0.77 ± 0.27*	*66.37 ± 13.57*	*8.18 ± 1.44*	*8.78 ± 1.03*
Qian et al. [9]		*0.30 ± 0.07*	*21.14 ± 5.01*	*2.15 ± 0.51*	*0.69 ± 0.23*	*37.64 ± 9.11*	*8.07 ± 0.97*	*7.19 ± 0.63*
RAIM [10]		*0.32 ± 0.08*	*21.03 ± 4.06*	*2.11 ± 0.33*	*0.65 ± 0.14*	*36.53 ± 13.27*	*7.93 ± 1.21*	*7.18 ± 0.92*
adv.RAIM		***0.31 ± 0.07***	***20.99 ± 3.61***	***2.12 ± 0.28***	***0.62 ± 0.15***	***35.85 ± 7.06***	***7.81 ± 1.01***	***7.16 ± 0.45***

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

TABLE V
TESTING SCENARIO 2 (FIXED 600 VEH/H/LANE) RESULTS

	Algorithm	Travel Time (s)	Waiting Time (s)	Time loss (s)
Traffic Lights	*FT10*	*43.65 ± 4.48*	*18.57 ± 2.95*	*26.27 ± 5.93*
	FT15	*32.81 ± 4.89*	*10.56 ± 2.25*	*15.36 ± 3.69*
	FT20	*33.62 ± 4.17*	*11.43 ± 2.34*	*16.23 ± 4.76*
	FT30	*36.34 ± 4.78*	*14.14 ± 2.99*	*18.78 ± 5.34*
	iREDVD [8]	*25.95 ± 3.67*	*6.10 ± 1.98*	*11.21 ± 2.66*
Qian et al. [9]		*19.52 ± 2.04*	*1.56 ± 0.51*	*2.93 ± 0.74*
RAIM [10]		*18.88 ± 1.65*	*1.14 ± 0.58*	*1.86 ± 0.68*
adv.RAIM		***17.94 ± 1.81***	***0.90 ± 0.45***	***1.31 ± 0.28***

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

As a reward signal, each vehicle (*agent*) received at each timestep:

- -100 (strong negative reward) if there was a collision.
- $+100$ (strong positive reward) if the intersection was crossed.
- $-timestep$ (weak negative reward) to encourage crossing as fast as possible.

Two main techniques are followed to achieve a more stable and fast training:

i) *Prioritized Experience Replay:* (PER) [40]. In DRL, a *replay buffer* is added to store past experiences and provide more stability during the training process. These past experiences are known as transition tuples, usually denoted as (s_t, a_t, r_t, s_{t+1}) with states, actions, rewards, and next states. Using this *replay buffer*, it is possible to remember the past experiences, not forget them, and thus approximate in a faster way Q-values of the DNN. However, not all actions provide the same information, and there are experiences where the *critic* can learn more. The experiences that PER considers the most "*learnable*" are those where the error committed between the predicted Q-value ($Q(s,a)$) and the actual Q-value ($Q^*(s,a)$) is high. This error is known as the Temporal Difference (*TD*) error and measures the uncertainty that the DNN possesses, being low when the error is low and high when it is high. Therefore, the higher this error is, the more likely it is to select an experience from the *replay buffe*r in the optimization process.

ii) *Learning by curriculum* [41]*:* Curriculum learning consists of training an intelligent agent in tasks with increasing complexity. First, the agent is trained in simple tasks, and once the agent is capable of completing them, the complexity of the tasks to be performed is gradually increased [41]. Compared to training without it, the adoption of the curriculum is expected to accelerate the speed of convergence and may improve the final performance of the model. Designing an efficient and effective curriculum is not an easy task. A bad curriculum can even make learning more difficult. In this work, curriculum-based learning through *Self-Play* is used. With this approach, the number of simulated vehicles was increased when a stable solution was reached after several simulations, i.e., when after a certain time, the system could not improve the solution found. Then, more vehicles were added to the simulation so that the controller could learn to work in a real multi-agent environment with multiple simultaneous vehicles.

V. SIMULATION SETUP

We employed the microscopic traffic simulation tool SUMO [42]. With this simulator, each vehicle is explicitly modeled. Pytorch 1.5.0 and Python 3.7 were also used to develop *adv.*RAIM. A 16-core processor was used together with an Nvidia 2080TI. The overall training process required 14 days (336 hours) due to the increasing complexity of the scenario and the convergence time of the DRL algorithm. Once adv.RAIM is trained, the size of the actor architecture is 20.3 MB, which any current GPU with more than 4 GB of memory can handle. In addition, adv.RAIM has an inference time of 1.89 ± 0.08 ms (mean ± est. dev. of 10,000 runs), so, during the 250 ms control period (the selected control time interval), a single adv.RAIM instance could control up to 106 vehicles in real-time. To control more vehicles simultaneously, several instances could be run in parallel, multiplying the control capacity.

The setup was divided into two parts: *i)* a training scenario, where the optimization of *adv.*RAIM was performed; and *ii)* four testing scenarios, three with a fixed flow rate and one closer to a "real-world scenario" with a variable flow rate. In all the testing scenarios, *adv.*RAIM was compared with 1) traditional traffic light control (Fixed Time, FT) with different cycle lengths (40, 60, 80, and 120 seconds, FT10, FT15, FT20, and FT30), 2) an advanced traffic-light control mechanism based on queue theory

TABLE VI
TESTING SCENARIO 2 (FIXED 600 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*	0.96 ± 0.20	48.35 ± 11.31	5.89 ± 1.50	2.00 ± 0.61	144.09 ± 33.09	19.85 ± 3.38	19.57 ± 2.74
	FT15	0.63 ± 0.19	45.73 ± 10.10	4.76 ± 1.73	1.34 ± 0.53	113.76 ± 32.01	16.68 ± 3.15	19.26 ± 2.52
	FT20	0.77 ± 0.16	42.51 ± 9.15	4.85 ± 1.40	1.48 ± 0.55	113.70 ± 31.08	17.39 ± 3.02	18.94 ± 1.77
	FT30	0.83 ± 0.23	44.78 ± 11.53	5.23 ± 1.27	1.71 ± 0.58	126.87 ± 30.15	18.30 ± 2.94	19.07 ± 1.64
	iREDVD [8]	0.63 ± 0.13	42.43 ± 12.15	4.54 ± 1.20	1.33 ± 0.54	92.68 ± 27.14	16.33 ± 2.79	17.48 ± 2.03
	Qian et al. [9]	0.61 ± 0.06	42.27 ± 9.94	4.23 ± 0.96	1.33 ± 0.42	75.28 ± 18.19	16.13 ± 1.89	14.31 ± 1.19
	RAIM [10]	0.62 ± 0.08	42.07 ± 8.10	4.22 ± 0.61	1.27 ± 0.31	73.03 ± 26.52	15.87 ± 2.85	14.34 ± 1.82
	adv.RAIM	$\mathbf{0.61 \pm 0.07}$	$\mathbf{41.93 \pm 7.22}$	$\mathbf{4.23 \pm 0.56}$	$\mathbf{1.25 \pm 0.27}$	$\mathbf{71.68 \pm 8.07}$	$\mathbf{15.62 \pm 2.44}$	$\mathbf{14.24 \pm 0.91}$

No collisions were recorded. [*mean* $\pm$ *std*. of 10 simulations].

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

	Algorithm	Travel Time (s)	Waiting Time (s)	Time loss (s)
Traffic Lights	*FT10*	107.35 ± 8.96	55.14 ± 5.89	72.58 ± 11.90
	FT15	95.66 ± 9.78	48.15 ± 4.53	60.75 ± 7.34
	FT20	87.30 ± 8.31	41.87 ± 4.72	55.46 ± 9.53
	FT30	82.70 ± 9.55	32.27 ± 5.99	47.52 ± 10.70
	iREDVD [8]	66.90 ± 7.38	25.18 ± 4.04	38.37 ± 5.32
	Qian et al. [9]	69.03 ± 4.11	17.10 ± 1.01	22.88 ± 1.52
	RAIM [10]	67.74 ± 3.30	16.27 ± 1.13	20.73 ± 1.36
	adv.RAIM	$\mathbf{65.90 \pm 3.65}$	$\mathbf{15.37 \pm 0.84}$	$\mathbf{19.62 \pm 0.57}$

No collisions were recorded. [*mean* $\pm$ *std*. of 10 simulations].

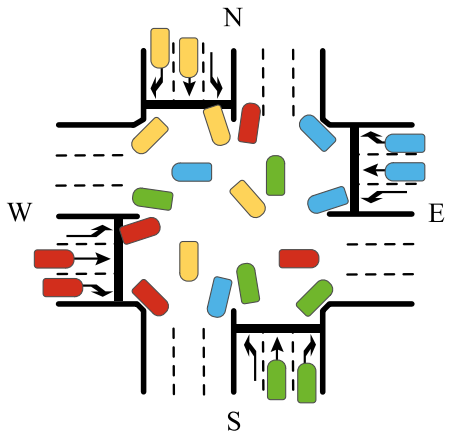

Fig. 4. Simulated intersection with 4 approaches and 3 lanes/approach, where the movements go straight, turn right, and turn left are allowed.

called *iREDVD* [8], and 3) an AIM previously proposed by Qian *et al.* [9].

A. Training Scenario

The training scenario consisted of an intersection with *4* branches of *3* lanes per direction, where left turns, right turns, and straight ahead were allowed. In our opinion, this is an important feature, as it offers a great variety of routes and possible collision zones, and very few AIMs allow it due to its high control complexity. A representation of the simulated intersection can be seen in Fig. 4. In each simulation, the *random seed* used was changed, to obtain a wider set of experiences. In this way, *adv.*RAIM was able to learn to deal with a variety of different input states and obtain a stable model.

Each simulation represented a 5-minute time-lapse, where the flow of vehicles increased (+150 veh/h) when a series of conditions were met (the variance of the reward in the last *150* simulations was less than 0.005 $\times$ the total simulated flow), following the curriculum methodology through *Self-Play* previously explained. This allowed us to train both the State/Conflict Encoder module and the Motion Planner simply and progressively, allowing the TD3 algorithm to find a stable control policy that allows it to instruct the speed of each autonomous vehicle at a busy intersection.

The metrics analyzed in the training scenario were global reward, number of collisions, and time loss (due to congestion). Time loss is due to having to drive slower than desired (below the desired speed), and includes waiting time.

The system took control of the vehicles when they were less than *100* m from the center of the intersection. A summary of the parameters used in the simulation can be seen in Table II. Therefore, the main objective of *adv.*RAIM was to maximize the total individual reward of each vehicle and to do so it strived to eliminate collisions (since it is a large penalty), as well as to reduce the travel time of the vehicles (since it will penalize each time step that the vehicles are within the intersection).

B. Testing Scenarios

The test scenarios provide a clear comparison of the performance offered by *adv.*RAIM and other traffic control algorithms in a more realistic scenario with conditions similar to reality. The simulated scenario consisted of an intersection with 4 approaches, and 3 lanes per approach as in the training scenario. Each simulation was run 10 times and represented a time span of 14 hours. To simulate various conditions, the simulated flow was separated into flow with vertical origin, and flow with horizontal origin. These flows could go in a straight line, turn right, or turn left. Four different scenarios were simulated, 3 with a fixed flow and one with a variable flow, shown in Fig. 5. The fixed flow scenarios were: one with low flow (200 veh/h/lane), one with medium flow (600 veh/h/lane), and one with high flow (1200 veh/h/lane). In the fourth variable flow test scenario, Fig. 5 shows that there were multiple different situations, with low (200 veh/h/lane), medium (600 veh/h/lane), and high (1200 veh/h/lane) flows, asymmetric and symmetric flows, and slow

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].

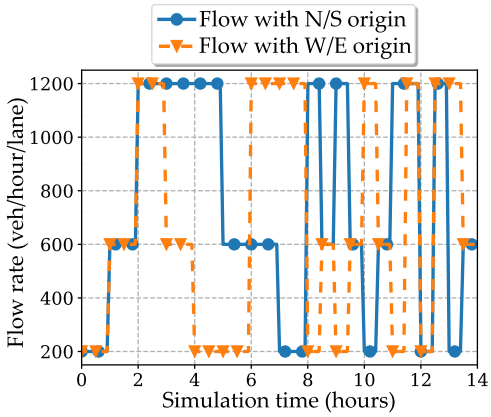

Fig. 5. Vehicle flow rate per lane used in the testing scenario.

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 scenarios.

Several metrics were studied. Due to the optimization technique, 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.

VI. 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 increasingly congested. This causes vehicles (on average) to drive progressively slower, but optimally to maximize the average reward received by each vehicle.

TABLE X
TESTING SCENARIO 4 (VARIABLE FLOW RATE) 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*	*1.58 ± 0.28*	*80.56 ± 18.84*	*9.78 ± 2.47*	*3.32 ± 0.98*	*240.13 ± 55.12*	*33.03 ± 5.58*	*32.57 ± 4.55*
	FT15	*1.04 ± 0.31*	*76.21 ± 16.81*	*7.88 ± 2.88*	*2.22 ± 0.88*	*180.56 ± 53.34*	*27.77 ± 5.21*	*32.08 ± 4.15*
	FT20	*1.27 ± 0.24*	*70.81 ± 14.99*	*8.08 ± 2.32*	*2.46 ± 0.91*	*189.49 ± 51.66*	*28.95 ± 5.01*	*31.54 ± 2.94*
	FT30	*1.38 ± 0.33*	*74.60 ± 19.22*	*8.69 ± 2.12*	*2.82 ± 0.92*	*211.44 ± 50.22*	*30.48 ± 4.88*	*31.74 ± 2.71*
	iREDVD [8]	*1.01 ± 0.21*	*70.70 ± 20.21*	*7.54 ± 1.99*	*2.21 ± 0.89*	*154.43 ± 45.22*	*27.21 ± 4.64*	*29.11 ± 3.33*
	Qian et al. [9]	*0.99 ± 0.09*	*70.44 ± 16.55*	*7.04 ± 1.58*	*2.18 ± 0.66*	*125.44 ± 30.31*	*26.84 ± 3.12*	*23.84 ± 1.98*
	RAIM [10]	*1.00 ± 0.10*	*70.08 ± 13.47*	*7.02 ± 0.97*	*2.11 ± 0.47*	*121.71 ± 44.19*	*26.42 ± 4.71*	*23.89 ± 3.01*
	adv.RAIM	***0.99 ± 0.08***	***69.84 ± 12.01***	***7.01 ± 0.91***	***2.07 ± 0.43***	***119.42 ± 13.42***	***25.99 ± 4.01***	***23.71 ± 1.48***

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

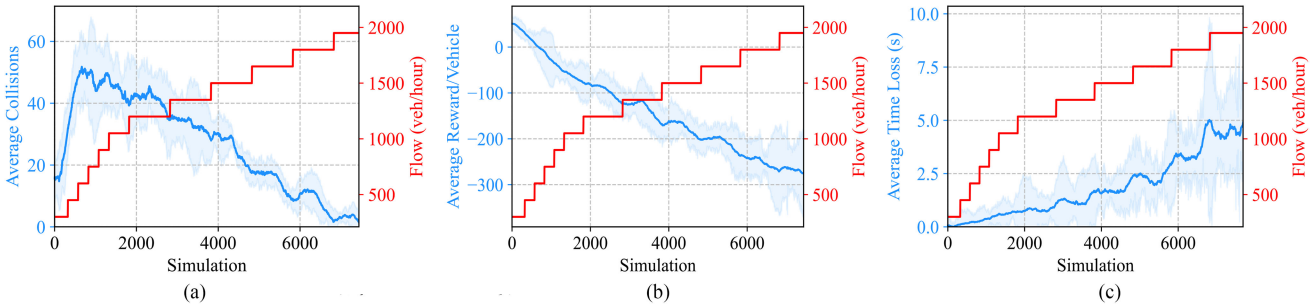

Fig. 6. Training results. We plot the smoothed mean with an exponential moving average and 90% confidence interval across 3 seeds. The red vertical axis on the right and the red curve show the flow of vehicles (veh/hour). The blue vertical axis on the left and the blue curve show each of the metrics studied: a) Average number of collisions; b) Average reward per vehicle; c) Average Time Loss.

Finally, if we look at the average time loss (Fig. 6c), it follows a trend inversely proportional to that of the reward received by each vehicle. As the number of simulated vehicles increases, the average time loss per vehicle also increases. In this metric, it is observed that as the flow increases, there are very abrupt changes in the trend, and after a few simulations, they stabilize. All the metrics analyzed remain stable within the confidence intervals, which confirms the good performance of the algorithms used.

The three figures show the convergence offered by TD3 and PER when the flow of vehicles increases. For all metrics, when the flow increases, some values break the trend they have (e.g., the number of collisions increases), but after a few simulations, the system can handle more vehicles and learns to deal with them to further optimize each metric.

B. Testing Scenarios

The results obtained in the fixed scenarios are shown in Tables III–VIII.

As can be seen, the advantage of not having traffic lights for the control of autonomous vehicles increases when traffic density is lower, particularly in the low traffic scenario (200 veh/h/lane, Tables III and IV), where the improvement in metrics is more significant in all cases.

For the medium flow scenario, the performance of the AIM approaches still outperforms that offered by the traffic lights-based approaches (Tables V and VI). Lastly, if we look at the high flow scenario (Tables VII and VIII), all algorithms present similar performance.

This behavior is because when the vehicular flow rate is low, the use of traffic light-based control algorithms to control an intersection is very inefficient since the use of the intersection is considerably reduced. Alternatively, if vehicles are allowed to circulate freely, taking into account safety regulations, the throughput they experience is superior, eliminating waiting times when there are almost no vehicles. However, this is not the case for high traffic flow rate scenarios, where traffic lights show good performance. In this case, the AIM algorithms behave like traffic light-based control algorithms.

Finally, Table IX includes the results obtained by the different algorithms in the fourth test scenario. This scenario is the closest to reality since it presents a wide variety of flows and conditions. If we focus on traditional traffic light control (FT and *iREDVD*), for all metrics under study, *adv.*RAIM shows its superiority. Using *adv.*RAIM, travel time is reduced by up to 59%. In turn, time loss is decreased by up to 95%. Another key result is achieved indirectly. The emissions of polluting gases (CO, CO_2, HC, PMx, and NOx) and consumption (fuel and electricity) are reduced, showing significant reductions of up to 37%, 13%, 28%, 37%, 50%, 21%, and 27%, respectively.

If we look at the additional results tables (Tables IV, VI, VIII, and X), we can see that *adv.*RAIM offers an improvement over the other methods. This can be explained because *adv.*RAIM is optimized to find a control policy that ensures that vehicles cross an intersection as fast as possible while guaranteeing safety. That is, vehicles spend as little time as possible within the intersection. Thus, since vehicles rarely stop at intersections (reducing waiting time and wasted time), vehicles do not have to

brake and then accelerate, but rather find the appropriate speed for each vehicle with which each vehicle crosses the intersection in the shortest possible time without collision. By not having to accelerate, other metrics such as fuel/energy consumption and pollutant emissions are reduced since it is at these times that higher consumption is realized.

On the other hand, if we compare the results obtained by *adv.*RAIM with the AIM algorithm proposed by Qian *et al.* in [9], we can observe similar results. However, if we focus on the operation of [9], we can see that this algorithm must have in advance the planning of all the vehicles. Then, employing FFS, it can organize the vehicles in a suboptimal way. In the event that there is a new vehicle or if the vehicles cannot adjust their speeds to those imposed by [9], the system must recalculate all vehicles' routes. Consequently, this prevents it from easily adapting to changing environmental conditions, which is very common at intersections where accidents, emergency vehicles, pedestrians, etc., may occur. With *adv.*RAIM, this does not happen because it can be trained to take into account these incidents and offer optimal solutions, considering all vehicles in each time interval and obtaining for each one the optimal speed that guarantees the greatest expected reward.

VII. 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, "A communications-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.

[7] B. Baker *et al.*, "Emergent tool use from multi-agent autocurricula," 2019, *arxiv.org/abs/1909.07528*.

[8] A. Guillen-Perez and M. Cano, "Intelligent IoT systems for traffic management: A practical application," *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 273–285, Feb. 2021.

[9] X. Qian, F. Altché, J. Grégoire, and A. Fortelle, "Autonomous intersection management systems: Criteria, implementation and evaluation," *IET Intell. Transp. Syst.*, vol. 11, no. 3, pp. 182–189, Apr. 2017.

[10] A. Guillen-Perez and M.-D. Cano, "RAIM: Reinforced autonomous intersection management - AIM based on MADRL," in *Proc. Workshop Challenges Real-World RL*, 2020, pp. 1–12.

[11] Y. Guo, J. Ma, C. Xiong, X. Li, F. Zhou, and W. Hao, "Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach," *Transp. Res. Part C: Emerg. Technol.*, vol. 98, pp. 54–72, Jan. 2019.

[12] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proc. Third Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2004, vol. 2, pp. 530–537.

[13] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst.*, 2013, pp. 529–534.

[14] M. Bashiri, H. Jafarzadeh, and C. H. Fleming, "PAIM: Platoon-based autonomous intersection management," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 374–380.

[15] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1908–1916, May 2019.

[16] P. Dai, K. Liu, Q. Zhuge, E. H. M. Sha, V. C. S. Lee, and S. H. Son, "Quality-of-Experience-Oriented autonomous intersection control in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1956–1967, Jul. 2016.

[17] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," *Transp. Res. Part C: Emerg. Technol.*, vol. 103, pp. 246–260, Jun. 2019.

[18] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4563–4571, Nov. 2020.

[19] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated intersection control," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2259, no. 1, pp. 223–232, Jan. 2011.

[20] M. A. S. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1136–1147, Jun. 2015.
[21] M. W. Levin, H. Fritz, and S. D. Boyles, "On optimizing reservation-based intersection controls," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 505–515, Mar. 2017.
[22] A. Mirheli, L. Hajibabai, and A. Hajbabaie, "Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment," *Transp. Res. Part C: Emerg. Technol.*, vol. 92, pp. 412–425, Jul. 2018.
[23] Z. He, L. Zheng, L. Lu, and W. Guan, "Erasing lane changes from roads: A design of future road intersections," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 173–184, Jun. 2018.
[24] B. Li, Y. Zhang, Y. Zhang, N. Jia, and Y. Ge, "Near-Optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1432–1437.
[25] B. Li and Y. Zhang, "Fault-Tolerant cooperative motion planning of connected and automated vehicles at a signal-free and lane-free intersection," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 60–67, 2018.
[26] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transp. Res. Part C: Emerg. Technol.*, vol. 100, pp. 161–176, Mar. 2019.
[27] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.
[28] X. Zhao, J. Wang, Y. Chen, and G. Yin, "Multi-objective cooperative scheduling of CAVs at non-signalized intersection," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 3314–3319.
[29] M. W. Levin and D. Rey, "Conflict-point formulation of intersection control for autonomous vehicles," *Transp. Res. Part C: Emerg. Technol.*, vol. 85, pp. 528–547, Dec. 2017.
[30] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *IEEE Trans. Veh. Technol.*, vol. 55, no. 6, pp. 1712–1724, Nov. 2006.
[31] J. Wu, A. Abbas-Turki, A. Correia, and A. El Moudni, "Discrete intersection signal control," in *Proc. IEEE Int. Conf. Service Operations Logistics Inform.*, 2007, pp. 1–6.
[32] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transp. Res. Part C: Emerg. Technol.*, vol. 55, pp. 363–378, Jun. 2015.
[33] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *Proc. Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2009, pp. 617–624.
[34] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: An ant colony system for autonomous intersection management," *Appl. Intell.*, vol. 37, no. 2, pp. 207–222, Sep. 2012.
[35] K. Dresner and P. Stone, "Sharing the road: Autonomous vehicles meet human drivers," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 1263–1268.
[36] P. Stone and Kurt Dresner, "Human-usable and emergency vehicle–aware control policies for autonomous intersection management," in *Proc. 4th Int. Workshop Agents Traffic Transp.*, 2016, pp. 17–25.
[37] M. W. Levin, S. D. Boyles, and R. Patel, "Paradoxes of reservation-based intersection controls in traffic networks," *Transp. Res. Part A: Policy Pract.*, vol. 90, pp. 14–25, Aug. 2016.
[38] B. Bakker, "Reinforcement learning memory," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 1475–1482.
[39] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 2587–2601.
[40] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. 4th Int. Conf. Learn. Representations*, 2016, pp. 1–21.
[41] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1–8.
[42] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package SUMO," in *Proc. RoboCup Infrastructure Simul. Competition*, 2006, pp. 1–5.

Antonio Guillen-Perez received the B.Sc. degree in telecommunication engineering and the M.Sc. degree in information and communication technologies in 2016 and 2018, respectively, from the Universidad Politécnica de Cartagena, Cartagena, Spain, where he is currently working toward the Ph.D. Degree with the Department of Information and Communication Technologies. His main research interests include intelligent transport systems, artificial intelligence, deep learning, multiagent systems, multiagent deep reinforcement learning, autonomous cars, IoT and smart cities. He has authored or coauthored several papers in national and international conferences and journals. He has collaborated as a Reviewer for international conferences and is a Member of many IEEE technical committees.

Maria-Dolores Cano (Senior Member, IEEE) received the Telecommunications Engineering degree from Universidad Politécnica de Valencia, Valencia, Spain, in 2000, and the Ph.D. degree from the Universidad Politécnica de Cartagena (UPCT), Cartagena, Spain, in 2004. In 2000, she joined UPCT, where she is currently an Associate Professor with the Department of Technologies and Communications. She has authored or coauthored numerous research works in international journals and conferences, in the areas of quality of service, quality of user experience, intelligent transportation systems, IoT, and cybersecurity. Dr. Cano was awarded a Fulbright grant as a Postdoctoral Researcher in 2006 at Columbia University, New York, NY, USA. She was the recipient of the Best Paper Award at the 10th IEEE International Symposium on Computers and Communications in 2005 and the Exemplary Reviewer Recognition by IEEE Communication Letters in 2010, among other recognitions.