Intelligent Roundabout Insertion using Deep Reinforcement Learning

Alessandro Paolo Capasso[1], Giulio Bacchiani[1] and Daniele Molinari[2]

[1]*VisLab - University of Parma, Parma, Italy*
[2]*VisLab, Parma, Italy*

Abstract: An important topic in the autonomous driving research is the development of maneuver planning systems. Vehicles have to interact and negotiate with each other so that optimal choices, in terms of time and safety, are taken. For this purpose, we present a maneuver planning module able to negotiate the entering in busy roundabouts. The proposed module is based on a neural network trained to predict when and how entering the roundabout throughout the whole duration of the maneuver. Our model is trained with a novel implementation of *A3C*, which we will call *Delayed A3C* (D-A3C), in a synthetic environment where vehicles move in a realistic manner with interaction capabilities. In addition, the system is trained such that agents feature a unique tunable behavior, emulating real world scenarios where drivers have their own driving styles. Similarly, the maneuver can be performed using different aggressiveness levels, which is particularly useful to manage busy scenarios where conservative rule-based policies would result in undefined waits.

1 INTRODUCTION

The study and development of autonomous vehicles have seen an increasing interest in recent years, becoming hot topics in both academia and industry. One of the main reasearch areas in this field is related to control systems, in particular planning and decision-making problems. The basic approaches for scheduling high-level maneuver execution modules are based on the concepts of time-to-collision (van der Horst and Hogema, 1994) and headway control (Hatipoglu et al., 1996). In order to add interpretation capabilities to the system, several approaches model the driving decision-making problem as a Partially Observable Markov Decision Process (POMDP, (Spaan, 2012)), as in (Liu et al., 2015) for urban scenarios and in (Song et al., 2016) for intersection handling. A further extension is proposed in (Bandyopadhyay et al., 2012) where a *Mixed Observability Markov Decision Process* (MOMDP) (Ong et al., 2010) is used to model uncertainties in agents intentions. However, since vehicles are assumed to behave in a deterministic way, the aforementioned approaches handle many situations with excessive prudence and would not be able to enter in a busy roundabout.

For this reason, the trend of using Deep Learning techniques (Goodfellow et al., 2016) for modeling such complex behaviors is growing; in particular *Deep Reinforcement Learning* (DRL) (François-Lavet et al., 2018) algorithms have proved to be efficient even in high-dimensional state spaces and have already been extended to the autonomous driving field, as in (Isele et al., 2018) for intersection handling and in (Hoel et al., 2018) for lane changes. However, these works show a major limitation, which is the lack of communication capabilities among vehicles. In fact, those models are trained on synthetic environments in which vehicles movements are based on hard coded rules. A solution to this problem is proposed in (Shalev-Shwartz et al., 2016), where vehicles inside the simulator were trained through *Imitation Learning* (Codevilla et al., 2017); however, this approach is expensive since it requires a huge amount of training data.

In our proposed work, this limitation has been overcome training the model in an environment populated by vehicles whose behavior has been learned in a multi-agent fashion as in (Bacchiani et al., 2019). In this way, drivers are able to implicitly communicate through actions and feature a unique, programmable, style of driving, enhancing the realism of the simulation. In order to train agents efficiently in this scenario, a different version of *A3C* (Mnih et al., 2016) has been implemented in which the asynchronous

agents policies are updated with a lower rate enhancing agents exploration; for this reason it has been called *Delayed A3C* (D-A3C). Desired actions are chosen based on a sequence of images representing what the agent perceives around it. Moreover, our solution permits to set the level of aggressiveness of the artificial driver executing the maneuver; this is essential in those situations in which an excessively cautious behavior or rule-based policies would lead to undefined waits, as in case of insertion in a highly busy roundabout.

As in (Behrisch et al., 2011) and in (Bansal et al., 2018), it has been adopted a simplified synthetic representation of the environment which is easily reproducible by both simulated and real data, so that the system trained offline can be easily validated on a real car equipped with perception systems. Furthermore, this representation greatly helps in reducing the sample complexity of the problem respect to simulators featuring a realistic graphic such as CARLA (Dosovitskiy et al., 2017) or GTA-based platforms (Richter et al., 2016). Our test-bed scenario is the synthetic reconstruction of a real roundabout built with the *Cairo* graphic library (Packard and Worth, 2019), shown in Figure 1.

However, since obstacles detected by the perception systems are not always accurate, our system has been evaluated also in the case of random noise added to the position, size and pose of the vehicles, as well as on the trajectory followed by the agents.

The model trained on the single scenario of Figure 1b has been tested on a different type of roundabout, shown in Figure 5, in order to evaluate the generalization capabilities of the system.

Finally, tests on real data have been performed using logs recorded with a vehicle equipped with proper sensors.

2 BACKGROUND

2.1 Reinforcement Learning

Reinforcement Learning (Sutton and Barto, 2018) deals with the interaction between an agent and its environment. The actor tries to learn from attempts and errors, receiving a reward signal and observing the state of the environment at every time step. The reward is typically a scalar value and it is related to the sequence of actions taken until that moment. The goal of an agent acting inside an environment at time t, is to learn a policy which maximizes the so called *expected return*, which is a function of future rewards; this is generally defined as

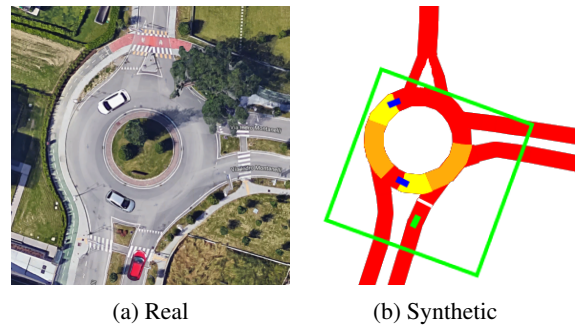
(a) Real   (b) Synthetic

Figure 1: Top view of a roundabout (a) and its synthetic representation (b). The green square in (b) highlights the portion of the surrounding perceived by the green vehicle, which is the artificial representation of the red car in (a).

$R_t = \sum_t^T r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t} r_{T-t}$, where T is the terminal time step and γ is a discount factor, used to reduce the importance of future rewards respect to the short-term ones.

2.2 Multi-agent A3C

One of the principal difficulties of DRL comes from the strict correlation between consecutive states; initially, the problem was solved by picking up indipendent states from a stored replay buffer (Mnih et al., 2015), but this proved to be inefficient in multi-agent scenarios (Gupta et al., 2017).

A different approach is taken in A3C (Mnih et al., 2016), where several copies of the agent take actions in parallel, so that each one experiences states of the environment which are independent from those of the others, enhancing the stability of the learning process. Agents send their updates and amend their local copy of the network every *n*-step frames.

Multi-agent A3C (Bacchiani et al., 2019) follows the same principle, but allows some of the agents to share the same instance of the environment, inducing them to learn how to interact in order to commonly achieve their goals. Thus, an implicit agent-to-agent negotiation can gradually arise, since actions taken from an agent will affect the state of others and vice-versa.

2.3 A2C

A2C (Wu et al., 2017) is the synchronous variant of A3C in which agents compute and send their updates at fixed time intervals. This solution is more time-efficient because it permits the computation of updates of all agents in a single pass exploiting GPU computing. However, since all agents hold the same policy, the probability of converging on a local mini-

mum of the loss function may increase, altough it has not yet proven empirically (Wu et al., 2017).

3 DECISION-MAKING MODULE

3.1 D-A3C Implementation

Our module is trained by a reinforcement learning algorithm which we will refer to as *Delayed A3C* (D-A3C), where the goal of the agent, called *active*, is a safe insertion in a roundabout populated by other vehicles, the *passives*, already trained in a multi-agent fashion in order to be endowed with interaction capabilities (Bacchiani et al., 2019). The actions performed by the *active* for entering the roundabout are controlled by the output of a neural network whose architecture is similar to the one adopted in (Bacchiani et al., 2019).

Our implementation differs from the original A3C in the way the asynchronous learners update the global neural network collecting all the actors' contributions. Indeed, our learners exchange the computed updates with the global network only at the end of their episode, keeping the same policy for the whole episode execution, while classic A3C does it at fixed and shorter time intervals. This reduces the synchronization burden of the algorithm, since the number of parameter exchanges diminishes. Moreover, in Section 4.1 we demonstrate that D-A3C leads to better results than classical A3C in the analyzed task. We did not carry further tests for evaluating the performances of the two algorithms in other tasks, since it is not the scope of this work; however, this comparison could be the subject of future studies.

(a) Navigable space   (b) Path
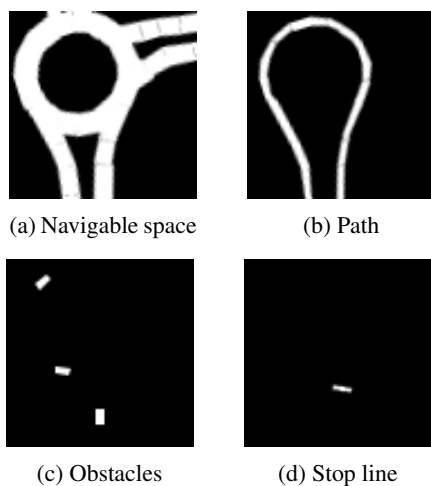
(c) Obstacles   (d) Stop line

Figure 2: Semantic layers of the input space of the agent.

The environment in which an agent is learning can be different from that of another agent, permitting to train the policy in a range of different scenarios simultaneously. Indeed, in our experiments we teach the agents how to enter in a three-entry-roundabout from all the entries simultaneously; nonetheless, in order to achieve a sufficient amount of agents for the learning process to be stable, we let multiple copies of the agents to learn from every entry in indipendent copies of the roundabout.

Multi-environment architectures should increase the model generalization capabilities: this is tested in Section 4.4 where the performance of our system is evaluated on an unseen roundabout.

3.2 Input Space

The input space of the system is composed by two different types of streams: a visual and a non-visual sensory channel. The visual input is a sequence of four images having size 84x84x4, that is a mapping of the 50x50 meters of the vehicle's sourrounding. These images represent 4 semantic layers consisting in the navigable space (Figure 2a) in which the agent can drive, the path (Figure 2b) that the agent should follow, the obstacles (Figure 2c) around the agent including itself and the stop line (Figure 2d) that is the position where the agent should stop if the entry cannot be made safely.

On the other hand, the non-visual sensory channel is composed by 4 entities: the first one is the *agent speed*, that is the absolute value of the current speed of the agent; the second one is the *target speed*, that represents the maximum speed that the actor should reach and maintain if there is no traffic and enough visibility; the third one is the *aggressiveness*, namely the degree of impetus in the maneuver execution and the last one represents the last action performed by the agent.

3.3 Output

The output of the maneuver planning system is a prediction over the following states:

- **Permitted:** the agent perceives the entry area of the roundabout as free and entering would not create any dangerous situation. This state sets the acceleration a of the *active* vehicle to a fixed comfort value a_{max} unless the *target speed* is reached.

- **Not Permitted:** the agent predicts the entry area of the roundabout as busy and entering would be dangerous. This state produces a deceleration computed as $\min(d_{max}, d_{stop\_line})$, where d_{max} is the maximum deceleration permitted following

comfort constraints, and d_{stop_line} is the deceleration needed to arrest the *active* vehicle at the stop line. If the agent has already overcome the stop line, this state causes a brake of a d_{max} value.

- **Caution:** the roundabout is perceived as not completely free by the *active* agent and the vehicle should approach it with prudence, either to improve the view for a safe entering or to observe if an oncoming *passive* vehicle is willing to let it enter the roundabout; the maximum speed permitted to the agent is $\frac{1}{2}$ *target speed* and *a* can assume one of the following values:

$$a = \begin{cases} \frac{a_{max}}{2}, & \textbf{if } agent\ speed < \frac{target\ speed}{2} \\ \frac{d_{max}}{2}, & \textbf{if } agent\ speed > \frac{target\ speed}{2} + h \\ 0, & otherwise \end{cases}$$

(1)

where *h* is a costant set to 0.5.

3.4 Reward

The reward r_t is composed by the following terms:

$$r_t = r_{danger} + r_{terminal} + r_{indecision} + r_{speed} \quad (2)$$

r_{danger} is a penalization given to the *active* agent when it performs dangerous maneuvers and it is defined as:

$$r_{danger} = -w_{d_s} \cdot \alpha \cdot d_s - w_{c_f} \cdot \alpha \cdot c_f \quad (3)$$

in which,

- d_s is a binary variable which is set to 1 when the *active* vehicle violates the safety distance from the *passive* one in front; this distance is equal to the space traveled from the *active* agent in one second, as shown from the yellow region in Figure 1b. When the safety distance is maintained the value of d_s is 0;

- c_f is a binary variable and it is set to 1 when the *active* agent cuts in front of a *passive* vehicle already inserted in the roundabout; this region is equal to three times the distance traveled from the *passive* vehicle in one second. This is shown from the orange region in Figure 1b. If the learning actor does not break this rule the value of c_f is 0.

- α depends on the aggressiveness level of the *active* agent and it is defined as $\alpha = (1 - aggressiveness)$. During the training phase, *aggressiveness* assumes a random value from 0 to 1 kept fixed for the whole episode. Higher values of *aggressiveness* should encourage the actor to increase the impatience; consequently, dangerous actions will be less penalized. In the test phase we fix

the *aggressiveness* value in order to perform comparisons among agents with different values of this parameter, as shown in Section 4.3.

- w_{d_s} and w_{c_f} are weights set to 0.002 and 0.005 respectively.

$r_{terminal}$ depends on how the episode ends. In order to avoid an excessively conservative behavior of the *active* agent, it is imposed a maximum available time for the actor to reach its target. The possible values $r_{terminal}$ can assume are:

- +1: if the *active* agent ends the episode safely, reaching its goal;

- $-\beta - \gamma \cdot \alpha$: if the *active* actor does not finish the episode because of a crash with another agent. β is a costant set to 0.2, while γ is the weight of α set to 1.8. Hence, when a crash occurs, we modulate $r_{terminal}$ based on the *aggressiveness*, for the same reason explained for r_{danger}.

- -1: if the time available to finish the episode expires.

$r_{indecision}$ is a negative reward in order to provide a realistic and smooth behavior to the agent, avoiding frequent changes of conflicting actions. It depends on the last two states of the system: we penalize the agent when the state passes from *Permitted* to one of the others. Calling $L1$ and $L2$ the last and the second to last outputs respectively, we can resume this reward with the following equation:

$$r_{indecision} = \begin{cases} -0.05, & \textbf{if} \quad L2 = Permitted \\ & \quad \text{and } L1 = Caution \\ -0.15, & \textbf{if} \quad L2 = Permitted \\ & \quad \text{and } L1 = Not\ Permitted \\ 0, & otherwise. \end{cases}$$

(4)

r_{speed} is a positive reward which encourages the *active* vehicle to increase the speed. It is defined as:

$$r_{speed} = \psi \cdot \frac{current\ speed}{target\ speed} \quad (5)$$

in which ψ is a constant set to 0.0045 and the *target speed* at the denominator acts as a normalizing factor. This reward shaping is essential to ensure that the agents learn the basic rules of the road like the right of way and the safety distance.

The following link (https://drive.google.com/open?id=1iGc820O_qeBSrWHbwhPTiTyby_HupM-_) shows how the *active* agent performs the entering in the roundabout.

4 EXPERIMENTS

4.1 Algorithms Comparison

We compared the A3C and A2C algorithms with our D-A3C implementation in order to test if our implementation improves the learning performances. The curves of Figure 3 show that A3C needs more episodes than our method for learning successfully the task. Instead, A2C converges on a suboptimal solution, consisting on always outputting the *Permitted* state, letting the agent entering the roundabout independently on the occupancy of the road. The *aggressiveness* used during the training phase is chosen randomly ([0, 1]) and kept fixed during each episode both for *passive* and for *active* vehicles, while the maximum number of *passive* vehicles populating the roundabout (Figure 1b) simultaneously is set to 8.
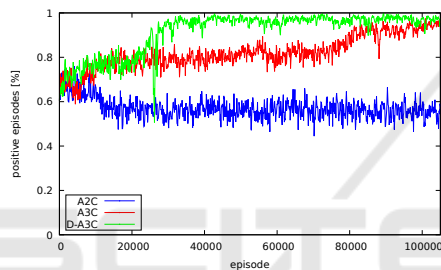

Figure 3: Moving average of the positive episodes ratio using D-A3C (green), A3C (red) and A2C (blue).

4.2 Comparison with a Rule-based Approach

The metrics used to evaluate the performances are *Reaches*, *Crashes* and *Time-overs* corresponding to the percentages of episodes ended successfully, with a crash and due to the depletion of the available time respectively. Every test is composed by three experiments (each one composed by 3000 episodes) using three different traffic conditions: *low*, *medium* and *high* which correspond to a maximum number of *passive* agents populating the roundabout to 4, 6 and 8 respectively. The results in Table 1 represent the average percentages of the three experiments. We compared the results obtained by D-A3C model on the

training roundabout (Figure 1b) with those achieved by a simple rule-based approach. In particular, we set four tresholds (25, 20, 15 and 10 meters) corresponding to the minimum distances required from a *passive* vehicle to the *active* one for starting the entering maneuver. Even if the percentages of crashes are rather low, the results in Table 1 show that a rule-based approach could lead to long waits since its lack of negotiation and interaction capabilities brings the agent to perform the entry only when the roundabout is completely free.

4.3 Aggressiveness Tests

As explained in Section 3.2, the aim of the *aggressiveness* input is to give the possibility of modulating the agent behavior depending on the traffic conditions. This is achieved by shaping the rewards accordingly to this input during the training phase as explained in Section 3.4, and exposing the agent to different traffic conditions. In order to prove the efficacy of the *aggressiveness* tuning, we tested the *D-A3C* model on a busy roundabout varying the aggressiveness level from low to high, highlighting the full spectrum of behaviors. We calculated the *average speed* of the *active* vehicle and the ratio of the episodes which ended successfully. As can be noted from Figure 4, the aggressiveness input acts a crucial role in determining the output of the module: higher values of this input rise the impatience of the *active* vehicle which tends to increase the risks taken. This produces an incre-
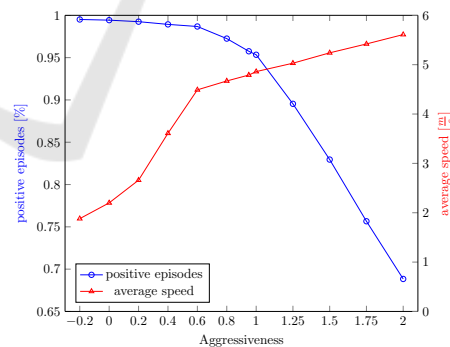

Figure 4: Values of average speed and positive episodes ratio depending on the aggressiveness level of the active agent.

Table 1: Comparison between D-A3C model and rule-based approach.

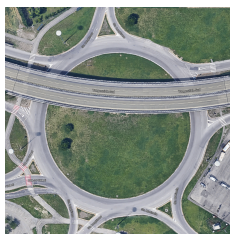|	Rule-based				D-A3C
	25m	20m	15m	10m	
Reaches %	0.456	0.732	0.831	0.783	0.989
Crashes %	0.0	0.002	0.012	0.100	0.011
Time-overs %	0.544	0.266	0.157	0.117	0.0

ment of crashes with a consequent decrease of the *positive episodes* ratio, but also an increase of the *average speed* value. In real-world tests, higher values of *aggressiveness* can be useful in deadlock situations (for example high traffic condition), flanking the module with safety systems in order to avoid collisions. Moreover, it is interesting to notice that the behavior of the system is coherent also for those *aggressiveness* values outside the range used during the training phase ([0,1]).
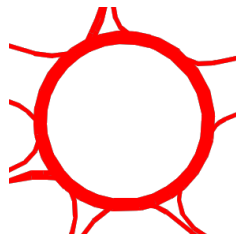
4.4 Performances on Unknown Roundabouts

We tested the system on a different type of roundabout from the one used in the training phase; the new roundabout, shown in Figure 5, features a different shape and number of entries. We compared the results achieved by the model with two different baselines: the first one obtained fixing the output of the module to the *Permitted* state independently on the occupancy of the road and the second one obtained with random actions. Due to a larger area involved in this test, *low*, *medium* and *high* traffic conditions correspond to a maximum number of passive agents inside the roundabout to 10, 15 and 20 respectively. The results in Table 2 show that the system features some generalization capabilities; however, considering the results achieved by *D-A3C* model on the training environment (Table 1), we can observe that the diversity of the training set environments has to be increased in order to improve the performances of the system in unseen roundabouts.

4.5 Perception-noise Injection

We introduced two types of noise in order to reduce the gap between synthetic and real data. We added gaussian noise in the position, size and pose estimation of *passive* agents to simulate those errors of the systems on-board the real vehicle. Then, we perturbed the path of *active* agents with Cubic Bézier curves computed by the De Casteljau algorithm (Boehm and Müller, 1999), in order to avoid following the same route as happens in the real world. This noise is also useful to make the system more robust to localization errors that may occur during tests on a self-driving vehicle. As shown in Figure 6, for each episode we randomly chose the initial and the final points, called P_i and P_f respectively; the only constraints are that *a*) P_i ranges between the first point of the original path (the green line in Figure 6) and the stop line; *b*) P_f ranges between the stop line and the last point of the path. Finally, we calculated two anchors P_1 and P_2 choosing two random points along the path and perturbating their coordinates (x, y) with gaussian noise.

Starting from the *D-A3C* model, we used *Curriculum Learning* (Bengio et al., 2009) to train the system in the noisy environment obtaining a new model which we call *Noised D-A3C*. We evaluated the two models in the noised environment, performing tests as in Section 4.2; the results in Table 3 show that the *Noised D-A3C* model becomes more robust to localization and perception errors. However, further tests on how to achieve better generalization on real data will be performed in future works.

Table 2: Results on the unknown roundabout.

	D-A3C	Random	Permitted
Reaches %	0.910	0.684	0.676
Crashes %	0.085	0.270	0.324
Time-overs %	0.003	0.046	0.0

Table 3: Results on the noised environment.

	D-A3C	Noised D-A3C
Reaches %	0.899	0.967
Crashes %	0.043	0.021
Time-overs %	0.058	0.012

(a) Real   (b) Synthetic

Figure 5: Top view of the real (a) and its synthetic representation (b) of the roundabout which was not seen by the agent during training.
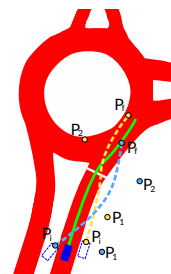

Figure 6: Example of Bézier curves: the green line (solid) represents the original path, while the light blue and the yellow lines (dotted) represent two possible Bézier curves.
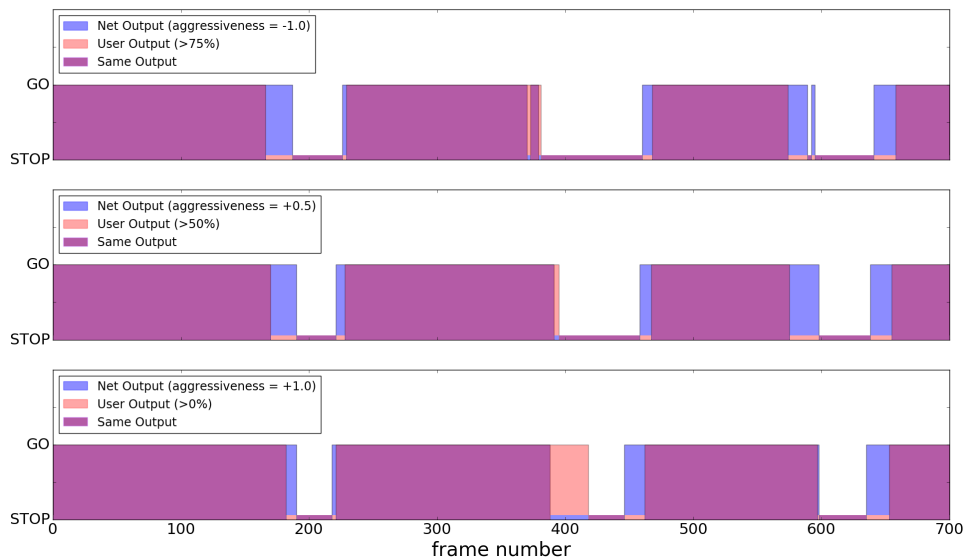
Figure 7: Comparison between the output of D-A3C module and those of the users for the first 700 frames (out of 2000). The blue and the red areas correspond to the decisions of our system and the users choices respectively, while the violet ones represent the frames in which the users and the net perform the same actions.

4.6 Test on Real Data

We evaluated our module with real data recording both around 2000 perception frames and video streams of the roundabout of Figure 1a with a car equipped with a stereo camera and a GPS. We projected the recorded traffic into our synthetic environment and performed a test with three different aggressiveness levels of the net ($-1.0, +0.5, +1.0$). These values represent different driving styles, from the most cautious to the most impetuous; since in Section 4.3 we noticed that the behavior of the system is coherent also for values outside the range used during the training, we chose -1.0 to accentuate the cautious behavior.

The frames have been recorded with the car stopped at the stop line and therefore also the agent in our synthetic environment took its decisions from the same point. However, in this way it is possible to evaluate the single-shot insertion but not the full capabilities of the system. In order to compare these results with human decisions, we developed a simple interface in which users, watching the real recorded sequences, have to choose when to enter in the roundabout and when to stay stop. However, since the output of our system is a prediction of three possible states (*Permitted*, *Not Permitted* and *Caution*), we modeled the *Caution* state as *Not Permitted* to make a correct comparison between users and *D-A3C* model actions.

We stored the decisions of 10 users and we set up a counter for each frame, representing the number of users that would perform the entry in the roundabout

at that time, such that its value ranges from 0 to 10. We created three different artificial user profiles based on these counter values: the first one in which the entry is performed if at least the 75% of the users would enter in the roundabout, the second one with this ratio equal to 50% and the last one in which at least one user ($>0\%$) would enter in the roundabout. These percentages represent three different human driving styles such that we could compare them with the different aggressiveness levels of the net explained previously. Figure 7 illustrates the comparison between the actions of our module (blue) and those of the users (red) in each of the three profiles. Moreover, Table 4 shows the average match percentages over the three video sequences between the first, the second and the third user profile with the results obtained setting the aggressiveness level of our system to -1.0 (*Comparison #1*), $+0.5$ (*Comparison #2*) and $+1.0$ (*Comparison #3*) respectively. Since the match percentages between different single user decisions range from 80% to 95%, we can observe that the results achieved in Table 4 represent a good match between our module output and human decisions.

Table 4: Average match percentages between user profiles and *D-A3C* model actions over the three video sequences.

Comparison #1%	81.288
Comparison #2%	84.389
Comparison #3%	84.515

5 CONCLUSION

In this paper we presented a decision-making module able to control autonomous vehicles during roundabout insertions. The system was trained inside a synthetic representation of a real roundabout with a novel implementation of A3C which we called *Delayed A3C*; this representation was chosen so that it can be easily reproduced with both simulated and real data. The developed module permits to execute the maneuver interpreting the intention of the other drivers and implicitly negotiating with them, since their simulated behavior was trained in a cooperative multi-agent fashion.

We proved that D-A3C is able to achieve better learning performances compared to A3C and A2C by increasing the exploration in the agent policies; moreover, we demonstrated that negotiation and interaction capabilities are essential in this scenario since a rule-based approach leads to superfluous waits.

It also emerged that the decision-making module features light generalization capabilities both for unseen scenarios and for real data, tested by introducing noise in the obstacles perception and in the trajectory of agents. However, these capabilities should be enforced in future works for making the system usable both in real-world and unseen scenarios.

Finally, we tested our module on real video sequences to compare the output of our module with the actions of 10 users and we observed that our system has a good match with human decisions.

REFERENCES

Bacchiani, G., Molinari, D., and Patander, M. (2019). Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning. *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '19, Montreal, QC, Canada, 2019*.

Bandyopadhyay, T. et al. (2012). Intention-aware motion planning. *WAFR*.

Bansal, M., Krizhevsky, A., and Ogale, A. S. (2018). Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. abs/1812.03079.

Behrisch, M., Bieker-Walz, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo – simulation of urban mobility: An overview. *Proceedings of SIMUL*, 2011.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. *Journal of the American Podiatry Association*, 60:6.

Boehm, W. and Müller, A. (1999). On de casteljau's algorithm. *Computer Aided Geometric Design*.

Codevilla, F. et al. (2017). End-to-end driving via conditional imitation learning.

Dosovitskiy, A. et al. (2017). Carla: An open urban driving simulator. *CoRL*.

François-Lavet, V. et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gupta, J., Egorov, M., and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. pages 66–83.

Hatipoglu, C., Ozguner, U., and Sommerville, M. (1996). Longitudinal headway control of autonomous vehicles. *Proceeding of the 1996 IEEE International Conference on Control Applications*, pages 721–726.

Hoel, C., Wolff, K., and Laine, L. (2018). Automated speed and lane change decision making using deep reinforcement learning. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*.

Isele, D. et al. (2018). Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2034–2039.

Liu, W., Kim, S., Pendleton, S., and Ang, M. H. (2015). Situation-aware decision making for autonomous driving on urban road using online POMDP. *2015 IEEE Intelligent Vehicles Symposium (IV)*.

Mnih et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–33.

Mnih, V. et al. (2016). Asynchronous methods for deep reinforcement learning. *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*.

Ong, S. C. W., Png, S. W., Hsu, D., and Lee, W. S. (2010). Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8):1053–1068.

Packard, K. and Worth, C. (2003-2019). Cairo graphics library.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. abs/1608.02192.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi agent, reinforcement learning for autonomous driving. abs/1610.03295.

Song, W., Xiong, G., and Chen, H. (2016). Intention-aware autonomous driving decision-making in an uncontrolled intersection. *Mathematical Problems in Engineering*, 2016:1–15.

Spaan, M. T. J. (2012). Partially observable markov decision processes. *Reinforcement Learning: State-of-the-Art*, pages 387–414.

Sutton, R. S. and Barto, A. G. (2018). *Introduction to Reinforcement Learning*. MIT Press.

van der Horst, A. and Hogema, J. H. (1994). Time-to-collision and collision avoidance systems. *ICTCT Workshop Salzburg*.

Wu, Y. et al. (2017). Openai baselines: ACKTR & A2C.