2024 IEEE Intelligent Vehicles Symposium (IV)
June 2-5, 2024. Jeju Island, Korea

2024 IEEE Intelligent Vehicle Symposium (IV) | 979-8-3503-4881-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/IV55156.2024.10588560

Decision Making for Autonomous Driving Stack: Shortening the Gap from Simulation to Real-World Implementations

Rodrigo Gutiérrez-Moreno[1], Rafael Barea[1], Elena López-Guillén[1], Felipe Arango[1],
Pedro Revenga[1] and Luis M. Bergasa[1]

Abstract— **This paper introduces a novel methodology for implementing a practical Decision Making module within an Autonomous Driving Stack, focusing on merge scenarios in urban environments. Our approach leverages Deep Reinforcement Learning and Curriculum Learning, structured into three stages: initial training in a lightweight simulator (SUMO), refinement in a high-fidelity simulation (CARLA) through a Digital Twin, and final validation in real-world scenarios with Parallel Execution. We propose a Partially Observable Markov Decision Process framework and employ the Trust Region Policy Optimization algorithm to train our agent. Our method significantly narrows the gap between simulated training and real-world application, offering a cost-effective and flexible solution for Autonomous Driving development. The paper details the experimental setup and outcomes in each stage, demonstrating the effectiveness of the proposed methodology.**

I. Introduction

The integration of Autonomous Driving (AD) within urban environments requires the development of an intelligent Decision Making (DM) system, which processes environmental information and executes safe actions. In recent years, Reinforcement Learning (RL) has emerged as a promising approach for dealing with the complexities and uncertainties inherent in these environments. However, it is crucial to acknowledge that the RL training process can be costly and unsafe when applied directly to real vehicles. Therefore, safety considerations must extend not only to the algorithmic level but also encompass the expenses associated with sensors and potential vehicle damage during training [1], [2]. To address this challenge, many approaches have shifted their focus towards an initial experimental phase that relies on a high-fidelity simulation. Such simulations can encompass critical scenarios, allowing for the detection of risky situations and the development of the DM system before transitioning to real-world testing [1], [3]. In the context of AD, a notable disparity often exists between simulation and reality, a phenomenon commonly referred to as the Reality Gap (RG). Various solutions have been proposed to bridge this RG, primarily categorized into three groups: 1) Sim2real, involving knowledge transfer from simulation to the real world. The core idea is to train AD systems in simulation and then apply them in real-world vehicles. 2) Digital twins (DT), utilizing virtual representations of the real environment allow real vehicles to learn knowledge of their DT by synchronizing data from both the real and simulated worlds in an offline way. 3) Parallel Intelligent technology (PI), which combines the advantages of both Sim2real and DT. In this last group the learned knowledge is applied to the real vehicle through parallel execution with real-time interaction between the real and virtual worlds and online feedback [4], [5].

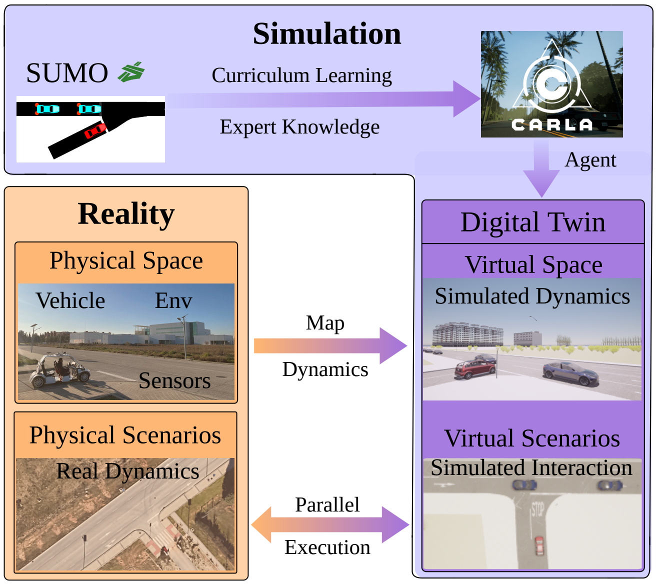


Fig. 1. Methodology for DM design. 1) DRL agent training in SUMO. 2) Second training of the model in CARLA using a DT of our real environment and vehicle. The AD stack with the prior knowledge of the DRL is evaluated in the virtual scenario. 3) Validation of the AD stack in a real scenario with virtual perception through a parallel execution, where the real vehicle and its DT are executed synchronously.

In this study, we introduce a novel methodology for the practical implementation of a DM module for our AD stack, specifically focusing on merge intersection scenarios and following a Curriculum Learning (CL) strategy. Our approach leverages Deep Reinforcement Learning (DRL) and is structured in three key stages. Firstly, the DRL agent is trained in a lightweight simulator, such as SUMO [6], to establish a prior behaviour model. Secondly, this model is transferred to a hyperrealistic simulation environment in CARLA [7] for a second training stage using a DT of our real environment and vehicle. This process includes capturing

*This work has been supported by the Spanish PID2021-126623OB-I00 project, funded by MICIN/AEI and FEDER, TED2021-130131A-I00, PDC2022-133470-I00 projects from MICIN/AEI and the European Union NextGenerationEU/PRTR, PLEC2023-010343 project (INARTRANS 4.0) from MCIN/AEI/10.13039/501100011033, and ELLIS Unit Madrid funded by Autonomous Community of Madrid.

[1] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, F. Arango, P. Revenga and L.M. Bergasa are with the Electronics Departament, University of Alcalá (UAH), Spain.{rodrigo.gutierrez, rafael.barea, elena.lopezg, juanfelipe.arango, pedro.revenga, luism.bergasa}@uah.es

the dynamics of the vehicle and its sensors, and replicating the road layout of our university Campus, culminating in a virtual representation of both the vehicle and the Campus. This DT serves as a virtual testing setup for designed scenarios, allowing for the evaluation of our AD stack in simulation in a safety way. The third stage is the validation of our AD stack in a real scenario with virtual perception through a parallel execution, where simulated and real-world experiments are conducted synchronously in real-time. Interaction with adversarial vehicles is simulated, while the framework is evaluated in our real vehicle. This innovative approach narrows the gap between simulated training and real-world application allowing great flexibility in the design of use cases at a low cost. The comprehensive methodology is depicted in Fig. 1.

A. Related Works

Autonomous Vehicles (AVs) DM has garnered significant research attention last years, with a range of approaches aimed at achieving safety and efficiency. Traditionally, classical approaches were used, but in this paper, we focus on learning-based methods, which have gained prominence in recent times [8].

We can distinguish different approaches. Statistical learning-based methods enable AVs to acquire human-like DM skills through extensive training data [9]. Deep learning-based methods are prominent for end-to-end approaches, utilizing raw sensor data for low-level control [10], [11]. RL-based methods aim to maximize returns through trial-and-error strategies. In this realm, Deep Reinforcement Learning (DRL) has emerged as a leading approach. Traditional methods to state representation typically revolve around lower-dimensional features, such as proximity to obstacles, lane positioning, and vehicular velocities [12]. These models exhibit robust behavior in complex situations, demonstrating strong resilience and adaptability. Other approaches include the adoption of higher-dimensional, such as Bird-Eye-View imagery [13], image augmentation techniques [14], and the use of occupancy grids [15]. Moreover, some approaches propose the use of high-level actions, such as ”stop,” ”drive slow,” or ”drive fast” [16], and decisions like ”take way” or ”give way” [17]. Others, focus on lane changing with commands like ”change left,” ”idle,” and ”change right” [18]. All these approaches have been applied to specific scenarios, but these only provide partial results applicable to larger applications.

On the other hand, some works present complete AD architectures based on DRL. The first architecture introduced in this context employs a Scene-Rep Transformer to enhance RL DM capabilities [19]. The actions proposed in this work are the longitudinal velocity of the ego vehicle and a lane change signal, executed by the SUMO simulator. Other works place a greater emphasis on realistic implementation within an AD architecture. The authors of [20] propose an attention-based driving policy for managing unprotected intersections, employing DRL. A hybrid approach is proposed by the authors of [21], presenting a DM and control framework that capitalizes on the strengths of both rule-based and learning-based techniques, while mitigating their disadvantages.

Most research on DRL-based methods has focused on simulated experiments. However, when working with real vehicles, safety and cost play a pivotal role. In this way, the transition from the simulated world to the real world is very important. To overcome this reality gap, as we mentioned before, three different approaches are considered.

Regarding Sim2Real techniques, Curriculum learning (CL) is a training strategy that trains a machine learning model from easier data to harder data, which imitates the meaningful learning order in human curricula [22]. In [23], an automatic curriculum generation method is proposed, and [24] obtains a better-overtaking performance using a tree-stages CL methodology. Transfer Learning (TL) is a technique in which knowledge learned from a task is re-used to boost performance on a related task. [25] proves that transfer learning using simulated accident data leads to better generalization to more diverse scenarios. In [26] a transfer learning for semantic segmentation of off-road driving environments using a pre-trained segmentation network is performed.

Another approach is the use of DT, which is a digital representation of a physical entity, which can simulate the entire life cycle of the operating system and synchronize the mapping with the physical twin [27]. A transfer learning strategy to efficiently train a DRL policy in simulation and deploy it in a real-time vehicle application is shown in [28]. In [29], a DT environment model that can predict the transition dynamics of the physical driving scene is proposed to improve the data efficiency of RL, which often requires a large amount of agent-environment interactions during the training process.

Recently, researcher have employed Parallel Intelligence (PI) approaches to facilitate the transfer of knowledge from simulated environments to the real world. These approaches combine the benefits of Sim2real and DT in modeling complex systems, addressing challenges that individual methods encounter in handling the RG problem. Liu et al.[30] enhance the safety and reliability of intelligent vehicles by integrating virtual vehicles with diverse roles in complex physical scenarios. Wang et al. [31] introduce the foundational idea of parallel testing, utilizing a cyclic updating method to tackle the RG problem and validate the performance of autonomous driving.

B. Contribution

Reviewing existing literature, we identified a significant opportunity in developing a DRL-based system tailored for real-world implementation. While numerous studies have focused on simulation-based solutions, there is a clear need for approaches that transition successfully into real applications using the different strategies identified in the state-of-the-art in an ad-hoc methodology. In light of this, we present our contributions as follows:

- Development of a hybrid DRL framework, which uses classic and learning-based techniques to create a practical DM for our AD stack.
- Methodology for DM design that reduces the gap between simulation and real-world application, consisting of a CL strategy for the DRL agent training, a DT of our real vehicle and university Campus, and a parallel execution that synchronizes the real world and the simulation.
- Validation of our AD stack, which includes the designed DM, in a real merge scenario with the physical vehicle and virtual perception. This novel strategy enables comprehensive testing and refinement of our framework using virtual adversarial vehicles, significantly reducing the cost and risks associated with real-work experiments.
- Comparison of various state-of-the-art DRL algorithms within the merge scenario in simulation.

II. Hybrid Decision Module for our Autonomous Driving Stack

The architecture of our AD stack, detailed in our previous work [32], is structured into four distinct levels, as illustrated in Figure 2.

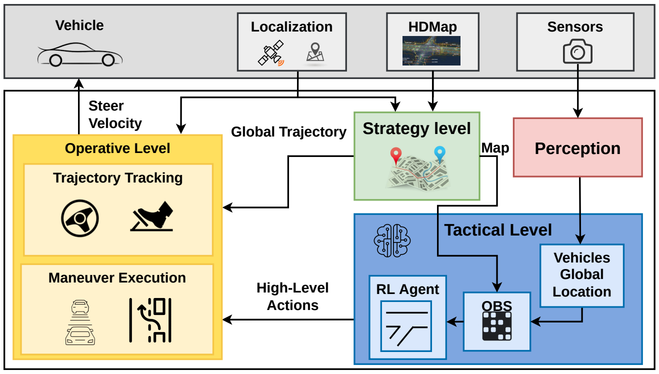


Fig. 2. The proposed hybrid DM architecture. The strategy level defines a tactical trajectory with the map information and the ego vehicle location. The tactical level executes high-level actions in correlation with the perception. The operative level combines the trajectory and the actions, calculating the driving commands.

The perception level is responsible for processing sensor data. However, this aspect is out of the focus of this work. In our setup, the perception layer directly extracts ground truth information from the simulator, bypassing the need for real-world sensor data processing. The strategy level [33] of our system is split into two main parts: the global planner and the scenario planner. The global planner develops the overall route for the vehicle, while the scenario planner creates a tactical path that highlights specific driving situations on the map. Next is the tactical level, where key decisions are made. It takes in data like the locations of different driving scenarios, HD map information, and the vehicle's current position. Based on this information, it makes high-level decisions about the vehicle's immediate actions, which are then implemented at the operative level. The operative level uses two main control systems: a Linear Quadratic Regulator [34] for following the planned route and a Model Predictive Control [35] system for carrying out manoeuvres. These systems work together to ensure the vehicle smoothly follows the strategic plan while responding to real-time situations.

III. Methodology for Decision Making Design

Our methodology for developing a DM module progresses through three phases. The first phase involves training basic behaviours and actions within a lightweight simulator, establishing the DM layer. The second phase includes the creation of a DT of our real vehicle and university Campus. Then transferring the pre-trained knowledge into a hyper realistic simulator, including vehicle dynamics. With the DT and the prior DM system a second training stage is carried on. Finally, the third phase tests and refines these techniques in a real-world setting, using real vehicle dynamics and actuators, but with simulated adversarial vehicles to ensure controlled, yet realistic, testing conditions.

A. Deep Reinforcement Learning for Merge Scenario

Our study focuses on merge intersections. While these might seem simple in the first place, their complexity lies in the high traffic density and the intentions of other vehicles. In these urban merge scenarios, the behaviour of other vehicles varies; some may be cooperative and yield, while others may not. Therefore, the agent's challenge is to accurately predict the actions of these vehicles to navigate the intersection effectively.

1) POMDP Formulation: A Partially observable Markov decision process (POMDP) is defined as a tuple (S, A, Ω, T, O, R), where S is a set of states, A is a set of actions, Ω is a set of observations, T is a transition function, O is an observation function, and R is a reward function. The agent receives an observation $o \in \Omega$, rather than observing the true state s' directly. The agent's internal knowledge of the state is represented by the belief $b(s)$, which is the probability of being in a state s. The optimal policy $\pi^*(b)$ maps beliefs to actions.

We define the state of a vehicle by its distance to the intersection, its speed, and its cooperation level. There are two categories of adversarial vehicles in this scenario depending on their cooperation level: the first type consistently yields and the second type always proceeds without yielding. For observation, the focus is on the velocity and distance to the merge point of the vehicles, forming an observation vector that informs the agent about the two closest vehicles. The action space for navigating intersections is simplified to two high-level actions: 'stop' and 'drive'. Lastly, the reward function aims to encourage rapid and safe navigation through intersections. It includes rewards for maintaining high velocities and crossing intersections, and penalties for collisions. A formal definition is provided in Table I.

2) DRL agent: Building upon our defined POMDP framework, we integrate a DRL agent for learning an optimal policy. While various algorithms exist, the results presented in Section III-B suggest that the TRPO [36] algorithm excels

TABLE I. POMDP Formulation for AD DM.

Set	Description	Vector
State	Distances, Velocities and Cooperation	$s_i = (d_i, v_i, i_i))$
Observation	Distances and Velocities	$\Omega = (d_e, v_e, d_1, v_1, d_2, v_2)$
Action	Drive and Stop	$a = (stop, drive)$
Reward	Success, Collision and velocity	$r = k_v * v_{ego} + kc * c + kv * v$

in robustly addressing these types of scenarios. In section IV-A we conduct a study comparing different algorithms in the proposed scenario.

TRPO is designed for steady policy performance improvement, making significant yet controlled policy updates. This is achieved by optimizing a surrogate objective function within a trust region, constrained by:

$$\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot|s), \pi_\theta(\cdot|s)] \leq \delta \tag{1}$$

where δ is a small positive value defining the trust region's size. We propose a DRL approach in which we update the policy using an actor-critic architecture. The implementation is done using the SB3 libraries. The loss function of the algorithm has the following form:

$$L(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right] \tag{2}$$

where $\pi_\theta(a_t|s_t)$ is the action probability under new policy parameters θ, and $\hat{A}_t$ is the advantage function at time t.

B. Training in SUMO

We define a merge scenario in SUMO simulator. The traffic density is characterized by an adversarial vehicle appearing every 1-3 seconds. The ego vehicle always enters the intersection in a specific lane. The adversarial vehicles are divided into two types: some may be cooperative and yield, while others may not. The speed of these adversarial vehicles is maintained at 8 m/s (29 km/h). A graphical representation of this scenario is shown in Figure 3, where the TRPO model is trained for 1M episodes.

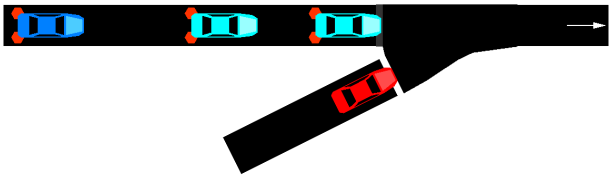
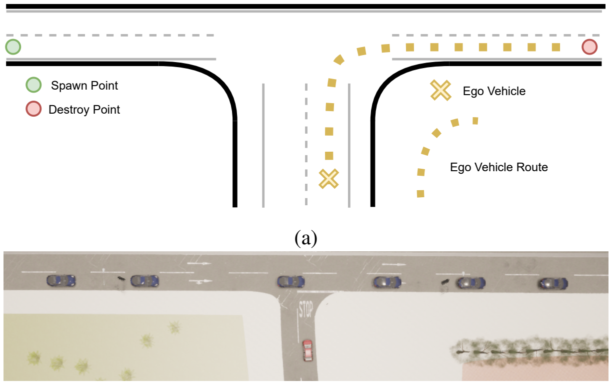


Fig. 3. A bird-eye-view of the simulation framework in SUMO. The ego vehicle (red) is entering the intersection. The color intensity of the adversarial vehicles (blue) is related to their intention. The clear yields and the dark does not yield.

C. Digital Twins in CARLA

In this phase, we integrate the DM module with the remaining modules of our AD stack, involving two critical steps. Initially, we define the characteristics of our real vehicle and we take measures of the scenario. With this information, we replicate a realistic configuration in CARLA. This is followed by a secondary training phase, allowing the agent to adapt to the new dynamics and control signals taken the SUMO model as a prior. Training from scratch in this hyperrealistic simulator could be time-intensive and has risk of non-convergence, as discussed in our previous work [37].

We define a merge scenario within the DT of our Campus. Adversarial vehicles are generated on a lane perpendicular to the ego vehicle's lane. These vehicles are subsequently destroyed when they reach the end of the scenario. The objective in this scenario is to navigate to the endpoint without collisions. A depiction of the scenario is presented in Figure 4.



(a)

(b)

Fig. 4. Representation of the merge intersection in CARLA. (**a**) Visual representation of the traffic flow, featuring spawn points (green), destroy points (red), and the initial location of the ego vehicle (yellow). The trajectory to be followed by the ego vehicle is also depicted. (**b**) Bird-eye-view of the scenario that shows the ego vehicle approaching the intersection and the adversarial vehicles crossing it.

The ego vehicle is defined with specific parameters to have the same response as the real vehicle under a certain control command. The most relevant aspects that we define are *mass, toque curve, drag coefficient, tyre friction and max steer angle*. Besides, we add a retardation factor to simulate delays or damping in the vehicle's response, which helps in mimicking real-world behaviour.

D. Parallel Execution (Virtual/Real)

To bridge the gap between simulation and real-world applications, we have developed an agent capable of translating the vehicle's movements from the real environment into the simulation. This approach enables us to apply decisions derived from the simulated environment directly to a physical vehicle, thereby facilitating a seamless transition from virtual to real-world testing. The real vehicle is mirrored in the simulator, and the simulation data feeds the decision system. This is achieved through two principal agents. The interface connecting these two agents with the simulation is depicted in Fig. 5.

- **Real Agent:** This agent processes input from a GNSS system to create a localization pose within the Campus map. The localization data is then fed into the operative level, which generates control commands. The Drive-By-Wire (DBW) module is responsible for translating

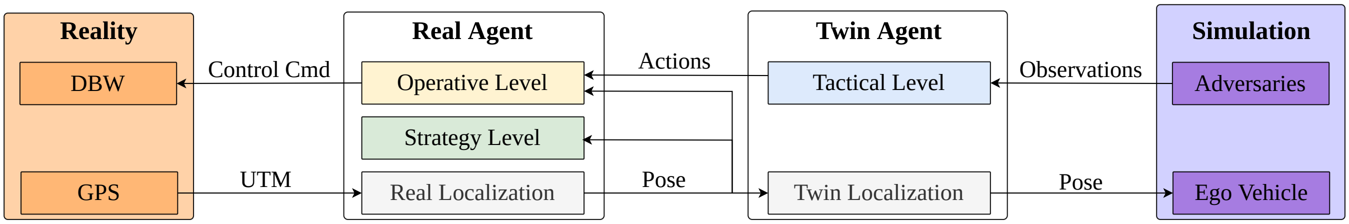


Fig. 5. The Real Agent processes GPS input and generates control signals, the DBW module controls the real vehicle, while the Twin Agent synchronizes the simulation. Simultaneously, the DM (Tactical level) coordinates actions with simulated observations for comprehensive control and coordination within the system.

these commands into electric signals for the real vehicle. The modules comprising the real-world platform, which include the localization module and the DBW system, are thoroughly described in our prior work [38].

- **Twin Agent:** The Twin Agent receives the vehicle's location data, provided by the real-world localization layer, and adjusts the simulated vehicle accordingly. Meanwhile, the DM layer processes the observations corresponding to the adversarial vehicles and generates the corresponding actions, which are sent back to the Real Agent.

IV. Experiments

In this section, we present the outcomes of the experiments conducted for each stage, as detailed in the preceding section.

A. Training and Testing in SUMO

We conducted a comparison between various state-of-the-art DRL algorithms in the scenario introduced in Section III-B. This evaluation facilitated the selection of an optimal algorithm for integration into the AD stack. In Figure 6, the progression of the training mean reward is depicted.

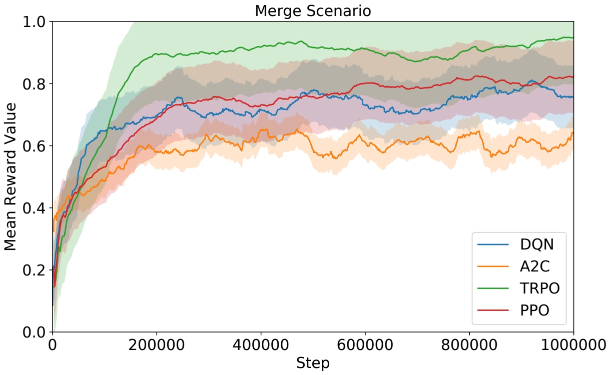


Fig. 6. Evolution of the mean rewards during the training process for the DRL agents in the merge scenario: DQN(blue), A2C(orange), TRPO(green) and PPO(red).

The TRPO emerges as the top performer, with the PPO and DQN agents demonstrating performances that are also competitive. In contrast, the A2C agent exhibits a notably poorer result. The agents are evaluated over 1000 episodes, with the outcomes detailed in Table II. The TRPO agent achieved the highest performance, showing a 92.9% success rate, which also marks it as the safest among the agents. Consequently, this agent was selected for integration into the hybrid DM system.

TABLE II. A comparison of the four DRL agents in the merge scenario. The success rate [%] and the average time (sec) are presented.

Metric	DQN	A2C	TRPO	PPO
Success Rate [%]	82.10	82.80	**92.90**	86.70
Average Time (sec)	6.81	**6.61**	7.84	7.18

B. Training in CARLA

Some experiment was conducted in the CARLA simulation to evaluate the performance and training efficiency of the DM system. Three different use cases were executed: training exclusively in SUMO, fine-tuning the prior SUMO model with CARLA data, and training the model from scratch in CARLA. The outcomes of the experiments are depicted in Table III.

TABLE III. A comparison in terms of performance and training time of the proposed training approaches.

Metric	Prior model	Fine-Tuning	From Scratch
Success Rate [%]	75.60	91.80	-
Average Time (sec)	21.53	19.98	-
Episodes Convergence	1M	1M + 10K	1M
Training Time (h)	5	21.5	1650

As anticipated, the prior model in SUMO reduces its performance when tested in CARLA, primarily due to the dynamic environment present in this simulator. It is worth highlighting that the fine-tuning process proved to be highly effective, achieving similar success rates as those observed in previous sections, specifically a 91.80% success rate. Additionally, the average processing times for both the prior and fine-tuned models were similar.

On the other hand, the curriculum learning strategy, employing both pre-training in SUMO and fine-tuning in CARLA, enables us to achieve model convergence 75 times faster compared to training the model from scratch. In this case, training time was estimated and given its long duration, the performance parameters were not calculated for this case.

C. Evaluation of the Parallel Execution

In these experiments, we do not assess the approach in terms of success rate, as such evaluation would be overly time-consuming for each episode in a real-world implementation. Our focus is on identifying the discrepancies between

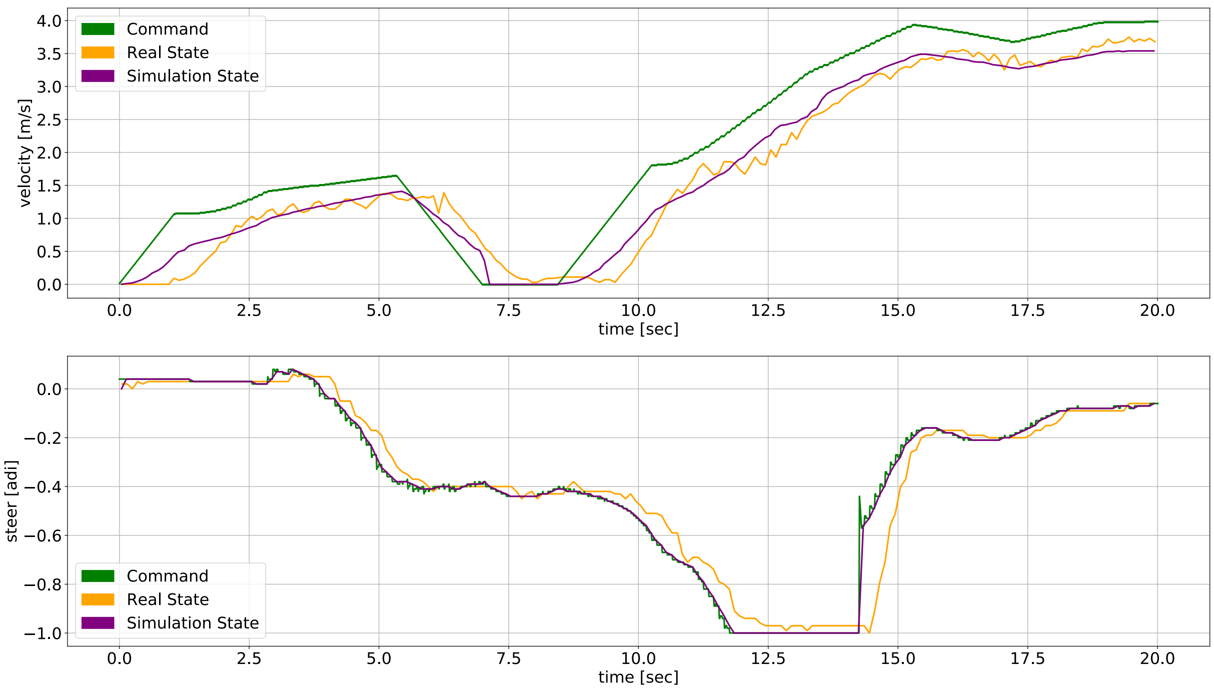
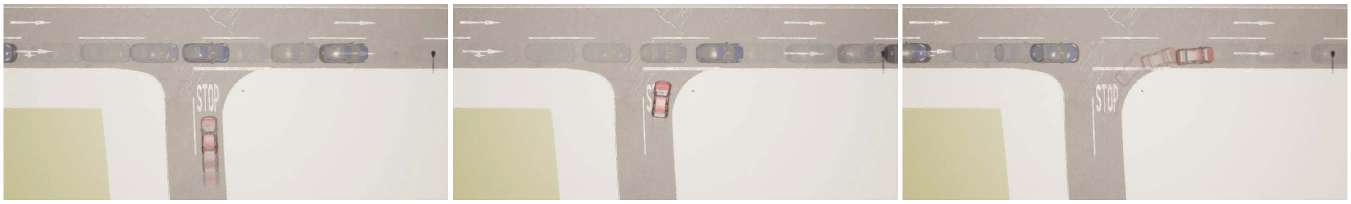


Fig. 7. Representation of the control signals during a parallel execution (virtual and real) within the merge scenario. Both, the real and the simulated vehicle are fed with the same command signal. The linear velocity is presented in the top chart and the steer is presented in the bottom one. The command (green), real (yellow) and simulate (purple) signals are represented.



(a) (b) (c)

Fig. 8. Bird's-eye view of a simulated episode in CARLA describing three scenes from the same episode at five-second intervals. The most prominent vehicles in each scene represent the latest positions, while the positions from earlier moments are gradually faded. Regarding the graphics presented in Fig 7: **(a)** Initially, at the 5-second mark, the vehicle receives a stop signal. **(b)** This is followed by a command to resume movement at the 8-second mark. **(c)** The experiment's final phase, occurring between the 11th and 14th seconds, showcases the vehicle turning.

real and simulated signals. To this end, we execute identical scenarios using the DT only in CARLA and the parallel execution with the Real and Twin agents. The control signals generated in each experiment are depicted in Figure 7.

Observations reveal that both the simulated and real vehicles are adept at adhering to their respective commands. Notably, both sets of signals exhibit a delayed response, a characteristic intentionally simulated in CARLA to mirror the behaviour observed in the real vehicle.

Finally, we present a bird's-eye view representation of our approach, which illustrates the scenario where the ego vehicle approaches a merge, stops, and then continues when it identifies a gap. This approach is showcased in Fig. 8.

V. Conclusions and Future Works

In this study, we presented an innovative approach for developing a DM module as part of an AD stack. Our method involved a three-phase development process: initial training in a lightweight simulator, employing a DT method in a highly realistic simulator to fine-tuning the model, and finally testing the complete AD stack in real-world scenarios with virtual observations through parallel execution. This strategy was specifically applied to urban merging situations. The study's key achievements include the successful utilization of the TRPO algorithm in the SUMO simulator and the remarkable efficiency benefits extracted from a two-stage training approach. The DT method effectively served as a transitional step, safely bridging the gap between simulated and real environments. Additionally, the parallel execution proved instrumental in validating our AD architecture, underscoring its viability and potential in advancing AD development.

Future research will focus on expanding this methodology to various driving scenarios, integrating advanced DRL algorithms, and enhancing the DT model. A key focus will be on the introduction and integration of various sensors, leading to a final comprehensive experiment with a fully equipped real-world setup. This will be conducted to validate the complete system in a variety of realistic driving conditions, and not only test the system's effectiveness but also its adaptability and reliability in real-world scenarios.

REFERENCES

[1] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.

[2] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep learning sensor fusion for autonomous vehicle perception and localization: A review," *Sensors*, vol. 20, no. 15, 2020.

[3] X. Hu, S. Li, T. Huang, B. Tang, R. Huai, and L. Chen, "How simulation helps autonomous driving:a survey of sim2real, digital twins, and parallel intelligence," *CoRR*, vol. abs/2305.01263, 2023.

[4] K. Kang, S. Belkhale, G. Kahn, P. Abbeel, and S. Levine, "Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight," in *2019 International Conference on Robotics and Automation (ICRA)*, p. 6008–6014, IEEE Press, 2019.

[5] Y. Liu, B. Sun, Y. Tian, X. Wang, Y. Zhu, R. Huai, and Y. Shen, "Software-defined active lidars for autonomous driving: A parallel intelligence-based adaptive model," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 8, pp. 4047–4056, 2023.

[6] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo - simulation of urban mobility: An overview," in *in SIMUL 2011, The Third International Conference on Advances in System Simulation*, pp. 63–68, 2011.

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning* (S. Levine, V. Vanhoucke, and K. Goldberg, eds.), vol. 78 of *Proceedings of Machine Learning Research*, pp. 1–16, PMLR, 13–15 Nov 2017.

[8] P. Wang, S. Gao, L. Li, S. Cheng, and H. xia Zhao, "Research on driving behavior decision making system of autonomous driving vehicle based on benefit evaluation model," *Archives of Transport*, 2020.

[9] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1590–1595, 2017.

[10] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J. Allen, V. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," *CoRR*, vol. abs/1807.00412, 2018.

[11] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2722–2730, 2015.

[12] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2156–2162, 2018.

[13] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. V. Gool, "End-to-end urban driving by imitating a reinforcement learning coach," 2021.

[14] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," 2021.

[15] M. Moghadam and G. H. Elkaim, "A hierarchical architecture for sequential decision-making in autonomous driving using deep reinforcement learning," 2019.

[16] D. Kamran, C. F. Lopez, M. Lauer, and C. Stiller, "Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning," 2020.

[17] T. Tram, I. Batkovic, M. Ali, and J. Sjöberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3263–3268, 2019.

[18] A. Alizadeh, M. Moghadam, Y. Bicer, N. K. Ure, M. U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment," *CoRR*, vol. abs/1909.11538, 2019.

[19] H. Liu, Z. Huang, X. Mo, and C. Lv, "Augmenting reinforcement learning with transformer-based scene representation learning for decision-making of autonomous driving," 2023.

[20] H. Seong, C. Jung, S. Lee, and D. H. Shim, "Learning to drive at unsignalized intersections using attention-based deep reinforcement learning," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 559–566, 2021.

[21] A. Aksjonov and V. Kyrki, "A safety-critical decision making and control framework combining machine learning and rule-based algorithms," 2022.

[22] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, vol. 44, pp. 4555–4576, sep 2022.

[23] Z. Qiao, K. Muelling, J. M. Dolan, P. Palanisamy, and P. Mudalige, "Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1233–1238, 2018.

[24] Y. Song, H. Lin, E. Kaufmann, P. A. Duerr, and D. Scaramuzza, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9403–9409, 2021.

[25] S. Akhauri, L. Zheng, and M. C. Lin, "Enhanced transfer learning for autonomous driving with systematic accident simulation," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5986–5993, 2020.

[26] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, "Semantic segmentation with transfer learning for off-road autonomous driving," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[27] A. Niaz, M. U. Shoukat, Y. Jia, S. Khan, F. Niaz, and M. U. Raza, "Autonomous driving test method based on digital twin: A survey," in *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, pp. 1–7, 2021.

[28] K. Voogd, J. P. Allamaa, J. Alonso-Mora, and T. Duy Son, "Reinforcement learning from simulation to real world autonomous driving using digital twin," *IFAC-PapersOnLine*, vol. 56, pp. 1510–1515, 01 2023.

[29] J. Wu, Z. Huang, P. Hang, C. Huang, N. De Boer, and C. Lv, "Digital twin-enabled reinforcement learning for end-to-end autonomous driving," in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pp. 62–65, 2021.

[30] T. Liu, X. Wang, Y. Xing, Y. Gao, B. Tian, and L. Chen, "Research on digital quadruplets in cyber-physical-social space-based parallel driving," *Chinese Journal of Intelligent Science and Technology*, vol. 1, no. 1, pp. 40–51, 2019.

[31] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in cpss: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 577–587, 2017.

[32] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, F. Arango, N. Abdeselam, and L. M. Bergasa, "Hybrid decision making for autonomous driving in complex urban scenarios," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[33] A. Diaz-Diaz, M. Ocaña, A. Llamazares, C. Gómez-Huélamo, P. Revenga, and L. M. Bergasa, "Hd maps: Exploiting opendrive potential for path planning and map monitoring," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[34] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluce, and L. M. Bergasa, "Reinforcement learning-based autonomous driving at intersections in carla simulator," *Sensors*, vol. 22, no. 21, 2022.

[35] N. Abdeselam, R. Gutiérrez-Moreno, E. López-Guillén, R. Barea, S. Montiel-Marín, and L. M. Bergasa, "Hybrid mpc and spline-based controller for lane change maneuvers in autonomous vehicles," in *2023 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2023.

[36] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1889–1897, PMLR, 07–09 Jul 2015.

[37] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluce, and L. M. Bergasa, "Reinforcement learning-based autonomous driving at intersections in carla simulator," *Sensors*, vol. 22, no. 21, 2022.

[38] J. F. Arango, L. M. Bergasa, P. A. Revenga, R. Barea, E. López-Guillén, C. Gómez-Huélamo, J. Araluce, and R. Gutiérrez, "Drive-by-wire development process based on ros for an autonomous electric vehicle," *Sensors*, vol. 20, no. 21, 2020.