Multi-Vehicle Mixed Reality Reinforcement Learning for Autonomous Multi-Lane Driving

Extended Abstract

Rupert Mitchell, Jenny Fletcher, Jacopo Panerati, and Amanda Prorok
Department of Computer Science and Technology, University of Cambridge
Cambridge, United Kingdom
{rmjm3, jlf60, jp872, asp45}@cam.ac.uk

ABSTRACT

Autonomous driving promises to transform road transport. Multi-vehicle and multi-lane scenarios, however, present unique challenges due to constrained navigation and unpredictable vehicle interactions. Learning-based methods—such as deep reinforcement learning—are emerging as a promising approach to automatically design intelligent driving policies that can cope with these challenges. Yet, the process of *safely learning* multi-vehicle driving behaviours is hard: while collisions—and their near-avoidance—are essential to the learning process, directly executing immature policies on autonomous vehicles raises considerable safety concerns. In this article, we present a safe and efficient framework that enables the learning of driving policies for autonomous vehicles operating in a shared workspace, where the absence of collisions cannot be guaranteed. Key to our learning procedure is a sim2real approach that uses real-world online policy adaptation in a *mixed reality setup*, where other vehicles and static obstacles exist in the virtual domain. This allows us to perform safe learning by simulating (and learning from) collisions between the learning agent(s) and other objects in virtual reality. Our results demonstrate that, after only a few runs in mixed reality, collisions are significantly reduced.

KEYWORDS

Multi-robot systems; Machine learning for robotics; Reinforcement learning; Autonomous vehicles; Reality gap; Sim2real

ACM Reference Format:
Rupert Mitchell, Jenny Fletcher, Jacopo Panerati, and Amanda Prorok. 2020. Multi-Vehicle Mixed Reality Reinforcement Learning for Autonomous Multi-Lane Driving. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 RELATED WORK

The idea of exploiting mixed (and augmented) reality for robotics applications was originally introduced as a tool to facilitate development and prototyping. Early work experiments with virtual humanoids amongst real obstacles [7]. Chen et al. [1] use augmented reality to obtain a coherent display of visual feedback during interactions between a real robot and virtual objects. More recently, mixed reality has gained importance in shared human-robot environments [8]. The introduction of mixed reality to support reinforcement learning has barely been considered. In [5], Mohammadi et al. present an approach for online continuous deep reinforcement learning for a reach-to-grasp task. Although targets exist in the physical world, the learning procedure is carried out in simulation, before being transferred to the actual robot.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

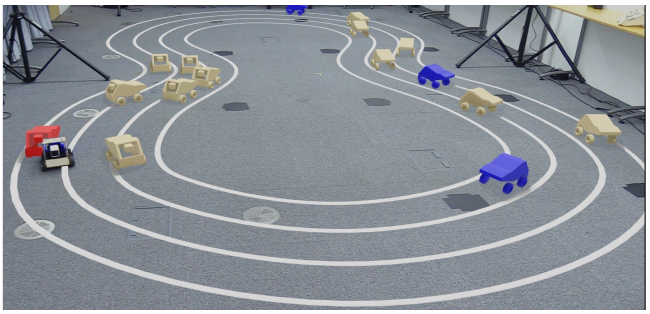

Figure 1: Mixed reality multi-vehicle multi-lane traffic circuit including one real DeepRacer robot and 16 virtual ones.

2 MULTI-VEHICLE SCENARIO

We consider the problem of high-level decision making in a multi-vehicle, multi-lane system—in particular, we are interested in lane changing manoeuvres. We introduce randomised static obstacles to perturb the traffic and to force such manoeuvres. In formalising this problem, we delegate (i) trajectory following and (ii) velocity regulation to low-level controllers and focus our learning efforts on high-level policies responsible for (i) changing lanes and (ii) selecting target velocities. We adopt the Amazon DeepRacer as our autonomous vehicle platform and deploy it in a 3-lane track together with 16 IDM/MOBIL [3] virtual cars (see Figure 1).

3 LEARNING FRAMEWORK

We formalise this problem as a reinforcement learning one in which an agent (the DeepRacer) receives noise-free but local observations. The observation space contains information about the position and desired velocity of the agent itself and (up to) six nearby vehicles. The action space is discrete: at every decision step, an agent chooses whether to (i) change lanes left, right, or not at all; as well as to (ii) accelerate, decelerate, or maintain its current velocity.

The reward function used to train the agent is presented in (1)—where c_0, c_1, and c_2 are hyper-parameters weighting velocity an proximity terms.

$$R(v_d, d_l, d_a) = -c_0|v_d| - max(0, c_1 L - d_l, c_2 \lambda - d_a) \qquad (1)$$

R contains (i) a penalty term for the deviation from the desired velocity, v_d; and (ii) a proximity penalty (with respect to other vehicles) calculated as the maximum of two terms. The first one considers the distance to the nearest vehicle in the current lane in either direction, d_l, and scales with the vehicle's length, L. The second term considers the distance to the nearest vehicle in any

lane, d_a, and scales with lane separation, λ—its purpose is to deter collision with vehicles in the process of changing lanes.

In our framework, the observations of each nearby vehicle are processed by a sequence of linear, ReLU activated layers before being max-pooled and concatenated with the observations of the agent's own state. These concatenated observations are then used as inputs for the actor and both critic networks, each composed of multiple ReLU linear layers. The actor network is followed by two additional soft-max layers, one for each of the two high-level actions (accelerating and lane-changing).

We update our network's weights using an adaptation of Asynchronous Advantage Actor Critic [4]. When updating the actor we use the PPO-Clip loss function [6] with an entropy term, and we use the smallest magnitude value function evaluation from the two critics [2].

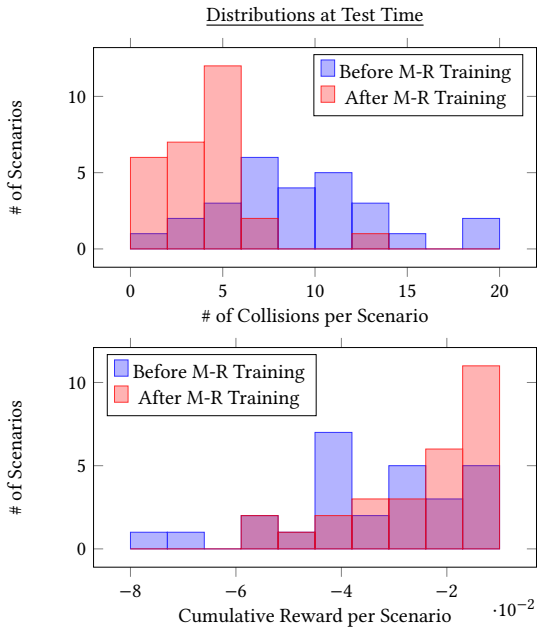

Figure 2: Empirical distributions at test time of *(i)* the number of collisions per scenario (top plot, left is best) and *(ii)* the total collected reward per scenario (bottom plot, right is best) before (blue) and after (red) training in mixed reality.

4 MIXED REALITY SETUP

The physics of the virtual vehicles are calculated by a C++ simulation. This environment is used to implement a fully virtual pre-training phase. Then, in mixed reality, the same simulator injects virtual information into the observations available to the DeepRacer robot. In the real world, the pose of the DeepRacer robot is tracked by six OptiTrack Prime 17W cameras. OptiTrack constantly updates the C++ simulation with this pose.

Learning in mixed reality is performed in an online fashion, with a small number of experience trajectories being collected across multiple initialisations of the environment between each optimisation step. Beyond the added safety of virtual collisions, our mixed reality framework also enables intuitive visualisation[1] by combining the C++ simulation and motion-tracking data.

[1]Video: https://www.youtube.com/watch?v=LlnaxZHWQOs

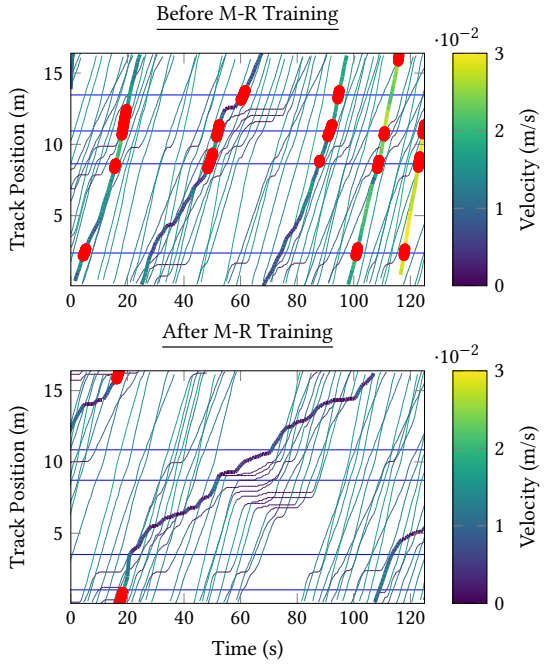

Figure 3: Plots of track positions (y axis) against time (x axis) of 4 static obstacles (horizontal lines), 12 virtual vehicles, and one real-life DeepRacer (thicker line). The colourmap captures velocity of each car. The red dots are collisions incurred by the DeepRacer. The top and bottom plots compare driving behaviours before and after mixed reality training.

5 EXPERIMENTAL RESULTS

We ran experiments in a 3-lane track with 16 virtual vehicles (12 running IDM/MOBIL, 4 acting as static obstacles) and one real, learning DeepRacer. After pre-training in a purely virtual environment, we measured performance before and after training in mixed reality.[2] Figure 2 shows that training in mixed reality caused a substantial reduction in mean collisions, as well as their variance. A qualitative portrait of the improved behaviour learned through mixed reality is given by Figure 3, which shows a substantial reduction in collisions (at the cost of a lower driving speed). The increase in average reward shown in Figure 2 demonstrates that the agent's increased caution is warranted by the trade-offs in the reward structure. This increase in optimal caution is likely due to the more unpredictable vehicle dynamics in the real world, when compared to simulation. Our mixed reality framework is first-of-its-kind, and we hope it will help bridge the reality gap that still stymies progress in reinforcement learning for robotics at large.

ACKNOWLEDGEMENTS

This work was supported by the Engineering and Physical Sciences Research Council (grant EP/S015493/1). Their support is gratefully acknowledged. The DeepRacer robots used in this work were a gift to Amanda Prorok from AWS. This article solely reflects the opinions and conclusions of its authors and not AWS or any other Amazon entity.

[2]Full paper: https://arxiv.org/abs/1911.11699

REFERENCES

[1] Ian Yen-Hung Chen, Bruce MacDonald, and Burkhard Wunsche. 2009. Mixed reality simulation for mobile robots. In *2009 IEEE International Conference on Robotics and Automation*. IEEE, 232–237.

[2] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 1587–1596. http://proceedings.mlr.press/v80/fujimoto18a.html

[3] Nicholas Hyldmar, Yijun He, and Amanda Prorok. 2019. A Fleet of Miniature Cars for Experiments in Cooperative Driving. *IEEE International Conference Robotics and Automation (ICRA)* (2019). https://doi.org/10.17863/CAM.37116

[4] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. *arXiv preprint arXiv:1602.01783* (2016).

[5] Hadi Beik Mohammadi, Mohammad Ali Zamani, Matthias Kerzel, and Stefan Wermter. 2019. Mixed-Reality Deep Reinforcement Learning for a Reach-to-grasp Task. In *International Conference on Artificial Neural Networks*. Springer, 611–623.

[6] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[7] Michael Stilman, Philipp Michel, Joel Chestnutt, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. 2005. Augmented reality for robot development and experimentation. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-55* 2, 3 (2005).

[8] Tom Williams, Daniel Szafir, Tathagata Chakraborti, and Heni Ben Amor. 2018. Virtual, augmented, and mixed reality for human-robot interaction. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 403–404.