doi:10.3311/BMEZalaZONE2022-014

Double Lane Change Path Planning Using Reinforcement Learning with Field Tests

Árpád Fehér[1], Szilárd Aradi[1a], Tamás Bécsi[1]

[1] *Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

[a] *Corresponding author: aradi.szilard@kjk.bme.hu*

Abstract

Performing dynamic double lane-change maneuvers can be a challenge for highly automated vehicles. The algorithm must meet safety requirements while keeping the vehicle stable and controllable. The problem of path planning is numerically complex and must be run at a high refresh rate. The article presents a new approach to avoiding obstacles for autonomous vehicles. To solve this problem, a geometric path generation is provided by a single-step continuous Reinforcement Learning (RL) agent. At the same time, a model-predictive controller (MPC) handles the lateral control to perform the dual lane-change maneuver. The task of the learning agent in this architecture is optimization. It is trained for different scenarios to provide geometric route planning parameters at the output of a neural network. During training, the goodness of the generated track is evaluated using an MPC controller. A hardware architecture was developed to test the local planner on a test track. The real-time operation of the planner has been proven. Its performance has also been compared to human drivers.

Keywords: *Local path planning, Model predictive control, Reinforcement learning, Vehicle dynamics*

1 Introduction

With the beginning of the 2010s, machine learning, in-depth learning and artificial intelligence have undergone rapid development. In addition to classical control planning and decision algorithms, they are advancing in solving control tasks, especially for various vehicle control tasks. It seems that combining artificial intelligence-based developments with control techniques can be an effective method for autonomous vehicle controls.

All vehicles need to drive the most optimal route when performing a critical maneuver, such as a double lane change. Several optimization criteria can be specified to minimize jerk and lateral acceleration. The moose test defined by ISO 3888-2 is a good tool for testing the stability of a vehicle in a dynamic limit situation.

The best path and trajectory planning algorithms for fully autonomous vehicle functions were covered in the following review publications [1], [2], [3]. A path is a sequence of waypoints that the vehicle must follow, referred to as a trajectory with supplied velocity information.

Several solvers can handle the entire constrained optimization problem, albeit it presents the issue of real-timeness [4]. One approach is to use deep learning to train a neural network for the solutions of many optimization results and use it as a practical solution or an initial guess for the solver [5], which can work for simple setups but is a difficult task to cover the entire state space in more complex scenarios.

Another method is using Reinforcement Learning (RL). The agent interacts with its environment based on trial-and-error and previous experiences and learns the best behavior using performance measurements called rewards [6]. These techniques often use end-to-end solutions, which means the agent responds to steering and acceleration demands. The trajectory planning is planted somewhere in the knowledge acquired via training. Many sensor models, such as grid-based [7], beam sensors [8], camera [9], or ground truth position information [10], can be utilized since the agent can cope with unstructured data. The other end of the RL-based research spectrum focuses on strategic decisions, defining high-level actions, and delegating task execution to an underlying controller, which is beyond the scope of this study. Only a few solutions exist in the literature where the path planning is done with RL [11], [12]. A survey on RL-based motion planning can be found in [13].

2 Methodology

Our current research aimed to create an experimental method to plan the optimal route for a double lane change in real-time, and some simplifications were introduced: constant asphalt-wheel friction coefficient, the vehicle has an ideal high-level sensor model, nonlinear single track vehicle model is used, and the maneuver is performed on a straight section of the road.

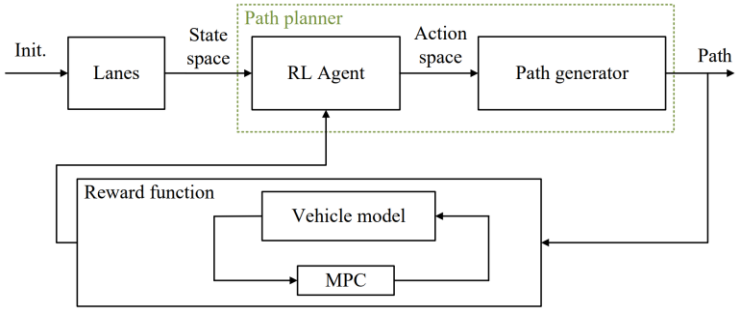

Fig. 1 *RL agent training architecture with classical control loop*

This task was solved differently from the classical optimalisation based approaches. The reinforcement learning was combined with geometric path planning. The state-space contains the lane's width, length, and position that the vehicle must traverse without touching the lane boundaries. Arc length and curvature parameters of polynomial and straight sections constitute the action-space. The generation of points x, y of the track is performed by a path generator based on the action-space. The path consists of two straight and three polynomial sections. The coefficients of the polynomials are calculated by determining their curvature function. The function of the path yaw angle and x, y points are determined by numerical integration along the arc length, which gives a good approximation.

The state space is 11, and the action space is 10 continuous values. The task is to generate the optimal route for a given condition or lane option.

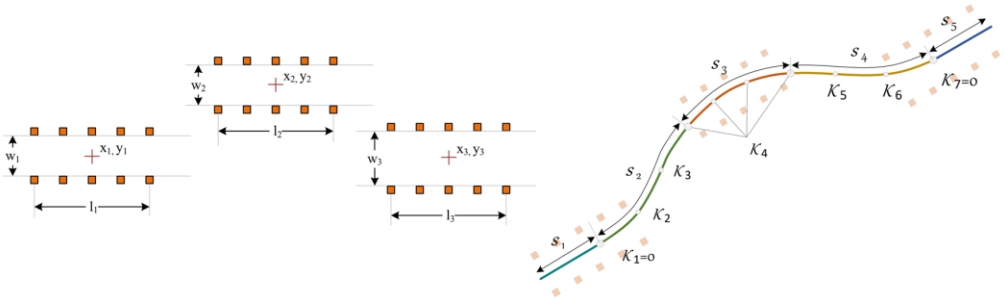

Fig. 2 *State and action-space representation of the training environment*

In reinforcement learning, an agent is taught, which requires a simulation environment. The training process needs many iterations. An iteration is called a step, and a series of steps is called an episode. One-step teaching was used wherein each step, the agent receives a different random but executable or near-executable state space and predicts an action space from which the path can be generated. The modeled vehicle is guided along the track by an MPC controller responsible for lateral control. Longitudinal control of the vehicle should only be performed until acceleration. The slip values, lateral acceleration, angle, and distance errors from the course are evaluated during the course. The value of the reward or the penalty can be determined if a cone has been touched or the track has been left. A TD3 (Twin-Delayed Deep Deterministic Policy Gradient) agent was implemented for training in a Python environment.

3 Hardware architecture for field test

The trained neural network of the agent consists of 3 hidden layers and a few hundred neurons, allowing fast and real-time prediction. The agent has been implemented on a Jetson AGX Xavier running Robot Operating System 2, which receives state space elements via CAN network and predicts action space.

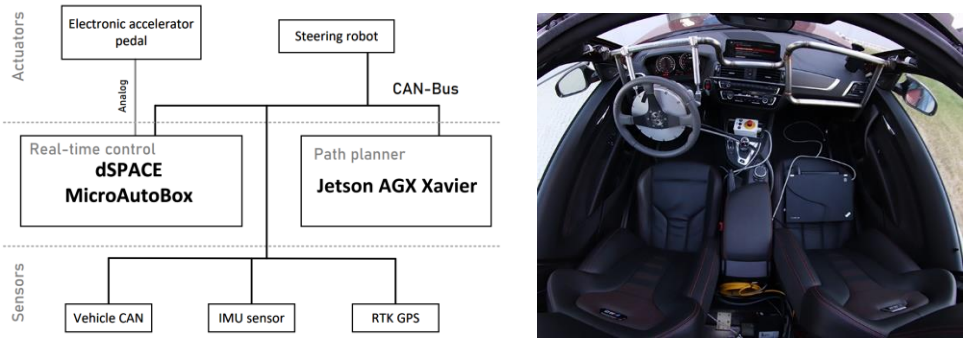

Fig. 3 *Hardware architecture and experimental setup*

A real-time dSPACE MicroAutoBox generates the track, and an MPC controller performs lateral control-based RTK GPS based on the action space. The real-time device also performs longitudinal control. A steering robot has been installed, and by removing the accelerator pedal, the engine control electronics receive the analog signal directly. The RTK GPS, IMU, and vehicle CAN data are logged for later evaluation.

A series-production BMW M2 car with a competition package was used for the tests, which after the modifications mentioned above, is suitable for throttle and steer-by-wire operation. The installation of the instruments is shown in the figure.

4 Results

After training the agent, the results were also validated with field tests. For real vehicle tests, a moose test as defined by ISO 3888-2 was performed at the ZalaZONE test track in Zalaegerszeg. The cones of the test were placed cm accurately with RTK GPS on the center of the 300 m diameter ultra-flat dynamic platform. While two supervising drivers were sitting in the vehicle, the system automatically accelerated to an initial speed until the ISO standard specified torque release point and then performed the double lane change.

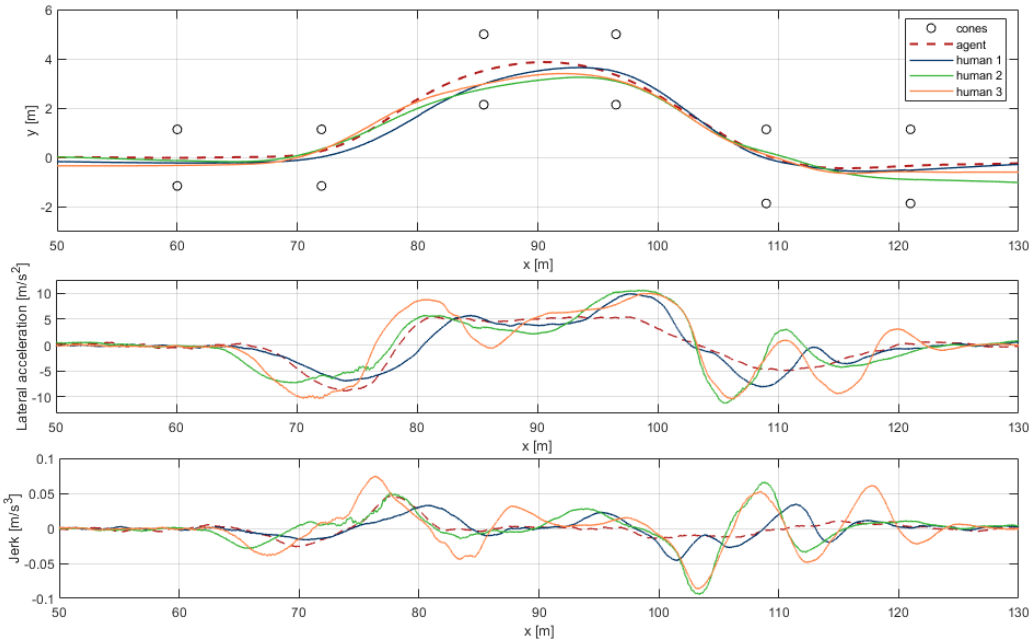

Fig. 4 *Traversed routes, lateral acceleration, and jerk of moose test experiments*

The agent successfully passed the test with 60 km/h maximum initial speed. The limitations of the lateral controller determine the maximum speed. In the case of a human driver, only 5 out of 14 ended without cone touch. There was already a cone touch when entering the first lane in many cases. Fig. 4 shows an attempt at nearly the same initial speed. The dashed line represents the agent's traversed path, side acceleration, and jerk. The blue successful and the green and orange represent unsuccessful attempts of three different human drivers. The figure shows that the agent performs the path with less lateral acceleration and jerk. A driver who succeeds often yanks

the steering wheel, while drivers who have performed with errors have made many corrections. The tests also showed that the first lane-change maneuver was the most difficult, as the vehicle's speed here was the highest, and the drivers are shown in the diagram also made a mistake here.

5 Conclusion

The article presents real-time geometric path planning by an agent which used reinforcement learning to perform a double lane maneuver. It is clear how effectively the machine learning approach can be combined with classical control algorithms since an MPC controller was used for lateral control during agent teaching and real-time tests. Drivers of average ability find it challenging to solve such a tricky maneuver, even at relatively low speeds. The following research topic will be the avoidance of moving objects.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

[1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," IEEE Transactions on intelligent vehicles, vol. 1, no. 1, pp. 33–55, 2016.

[2] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 5, pp. 1826–1848, 2019.

[3] D. Gonzalez, J. Perez, V. Milan'es, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1135–1145, 2015.

[4] Y. Lin, J. McPhee, and N. L. Azad, "Longitudinal dynamic versus kinematic models for car-following control using deep reinforcement learning," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019, pp. 1504–1510.

[5] F. Hegedüs, T. Bécsi, S. Aradi, and P. Gáspár, "Motion planning for highly automated road vehicles with a hybrid approach using nonlinear optimization and artificial neural networks," STROJNISKI VESTNIKJOURNAL OF MECHANICAL ENGINEERING, vol. 65, no. 3, pp. 148–160, 2019.

[6] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[7] A. Folkers, M. Rick, and C. Buskens, "Controlling an Autonomous Vehicle with Deep Reinforcement Learning," in 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 6 2019, pp. 2025–2031.

[8] J. Lee, T. Kim, and H. J. Kim, "Autonomous lane keeping based on approximate Q-learning," in 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, 6 2017, pp. 402–405.

[9] P. Wolf, C. Hubschneider, M. Weber, A. Bauer, J. Hartl, F. Durr, and J. M. Zollner, "Learning how to drive in a real world simulation with deep Q-Networks," in 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 6 2017, pp. 244–250.

[10] W. Xia, H. Li, and B. Li, "A Control Strategy of Autonomous Vehicles Based on Deep Reinforcement Learning," in 2016 9th International Symposium on Computational Intelligence and Design (ISCID). IEEE, 12 2016, pp. 198–201.

[11] M. P. Ronecker and Y. Zhu, "Deep Q-Network Based Decision Making for Autonomous Driving," in 2019 3rd International Conference on Robotics and Automation Sciences (ICRAS). IEEE, 6 2019, pp. 154–160.

[12] Á. Fehér, S. Aradi, T. Bécsi, P. Gáspár, and Z. Szalay, "Proving Ground Test of a DDPG-based Vehicle Trajectory Planner," in 2020 European Control Conference (ECC), 2020, pp. 332–337.

[13] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," IEEE Transactions on Intelligent Transportation Systems, pp. 1–20, 2020