Using Simulation to Design Path Following and Obstacle Avoidance Policies for Autonomous Robots

Harry Zhang, Stefan Caldararu, Alexis Ruiz, Huzaifa Unjhawala, Nevindu Batagoda, Ishaan Mahajan, Sriram Ashokkumar, Jason Zhou, Aaron Young and Dan Negrut

May 16, 2024

Using Simulation to Design Path Following and Obstacle Avoidance Policies for Autonomous Robots

Harry Zhang, Stefan Caldararu, Alexis Ruiz, Huzaifa Mustafa Unjhawala, Nevindu Batagoda, Ishaan Mahajan, Sriram Ashokkumar, Jason Zhou, Aaron Young, Dan Negrut

Department of Mechanical Engineering
University of Wisconsin-Madison
1513 University Avenue, 53706, Madison, USA
[hzhang699,scaldararu,aruiz26,unjhawala,batagoda,imahajan,ashokkumar2,zzhou292,negrut]@wisc.edu

1 Introduction

The design of an autonomous control algorithm is a challenging task as it traditionally requires extensive real-world testing, which is time consuming and expensive. Simulation is a valuable tool for autonomy design, e.g., assisting with parameter tuning, algorithms testing, in a time and cost-effective manner. Moreover, within the scope of Machine Learning (ML), simulation is attractive due to its ability to generate training data. Herein, we demonstrate the use of a simulation engine [1] and the Autonomy Research Testbed (ART) [2] platform to facilitate the autonomous policy development process for an obstacle avoidance ML control policy. This work builds off previous contributions that demonstrated the transferability of various multi-speed path following control policies [3, 4]. This study demonstrates the additional capability of obstacle avoidance via Machine Learning (ML). The ML has been trained using data collected while a human driver drove in a simulator.

2 Method

The ART platform is designed to facilitate autonomy policy development. It is based on the open-source simulation engine Chrono [1], whose high-fidelity vehicle dynamics [5] and physically based sensor [6] simulators anchor the autonomy design process. ART works with ROS2 and leverages the recently implemented Chrono::ROS module.

Control Policy Design 1: The first control policy is an end-to-end learning-based approach that uses imitation learning. The policy is trained to map the sensor data directly to control inputs in order to perform both path following and obstacle avoidance. The training pipeline is shown in Fig. 1. After the training process, the policy is validated in simulation and then transferred to the real vehicle for testing.

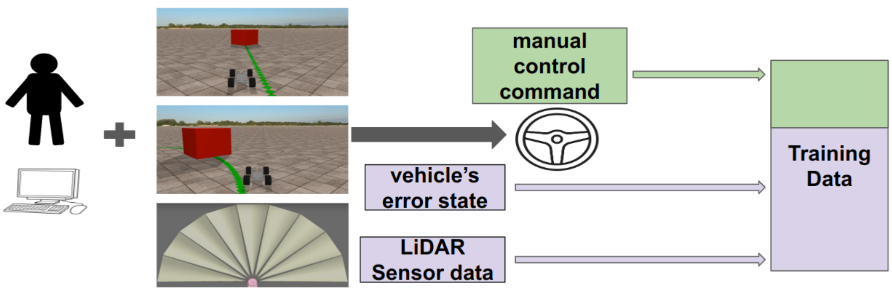
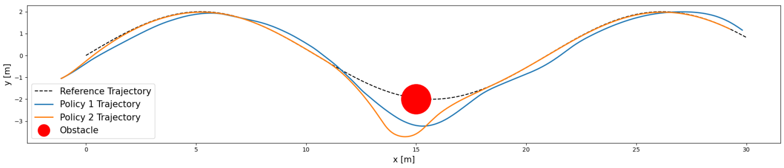
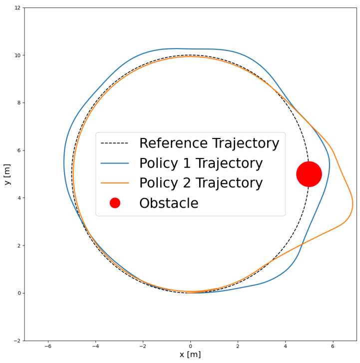


Figure 1: Training Pipeline for Control Policy Design 1. At each time step, LiDAR sensor data, the error state, and the control input from the human driver are recorded. Then, a NN is trained to map the LiDAR sensor data and error state to the control input from the human driver.

Control Policy Design 2: The second control policy designed combines a path following controller and an obstacle avoidance controller. The path follower controller is based on the work done in [3], which relies on "error states" feedback and a Neural Network (NN). The obstacle avoidance controller utilizes the LiDAR sensor and a defined value function to generate control inputs to avoid obstacles. The value function is designed so that the further away from the obstacle the robot is, the less impact the value function has on the control inputs; and vice-versa. A threshold distance value indicates the "switch" between the path following and the obstacle avoidance controllers. The parameters in the value function as well as threshold values are tuned within the framework provided by the ART platform.



(a) Sinusoidal Reference Trajectory



(b) Circular Reference Trajectory

3 Experiment and Analysis

After designing two control policies as described in Sec. 2, we conducted validation in simulation to verify the quality of the policies. As shown in Fig. 2a and Fig. 2b, the vehicle follows the reference trajectory and avoids obstacles well for both policies. Each control policy has its own advantages and drawbacks. For the first policy: although the path following performance is inferior, the obstacle avoidance controller is more aggressive and able to avoid obstacles in a more efficient manner. For the second policy: the tracking controller produces good performance in terms of accuracy, but the obstacle avoidance controller tends to be conservative and take a large detour around obstacles.

4 Conclusion and Future Work

This work demonstrated the possibility of designing a path following and obstacle avoidance NN control policy trained on a human driver using simulation and the ART platform. The policies were trained, fine-tuned, and validated in simulation. Both policies do well in path tracking and obstacles avoidance. Ongoing work is focused on deploying these policies onto actual scale vehicles. In our presentation, we will report on the process of transferring the policies to several real vehicles and reports on the sim-to-real gap.

References

[1] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science*, T. Kozubek, Ed. Springer International Publishing, 2016, pp. 19–49.

[2] A. Elmquist, A. Young, I. Mahajan, K. Fahey, A. Dashora, S. Ashokkumar, S. Caldararu, V. Freire, X. Xu, R. Serban, and D. Negrut, "A software toolkit and hardware platform for investigating and comparing robot autonomy algorithms in simulation and reality," *arXiv preprint arXiv:2206.06537*, 2022.

[3] H. Zhang, S. Caldararu, S. Ashokkumar, I. Mahajan, A. Young, A. Ruiz, H. Unjhawala, L. Bakke, and D. Negrut, "Zero-shot policy transferability for the control of a scale autonomous vehicle," *arXiv preprint arXiv:2309.09870*, 2023.

[4] H. Zhang, S. Caldararu, I. Mahajan, S. Chatterjee, T. Hansen, A. Dashora, S. Ashokkumar, L. Fang, X. Xu, S. He, and D. Negrut, "Using simulation to design an MPC policy for field navigation using GPS sensing," in *ECCOMAS Multibody Dyanmics Conference*, 2023.

[5] R. Serban, M. Taylor, D. Negrut, and A. Tasora, "Chrono::Vehicle template-based ground vehicle modeling and simulation," *Intl. J. Veh. Performance*, vol. 5, no. 1, pp. 18–39, 2019.

[6] A. Elmquist, R. Serban, and D. Negrut, "A sensor simulation framework for training and testing robots and autonomous vehicles," *Journal of Autonomous Vehicles and Systems*, vol. 1, no. 2, p. 021001, 2021.