Autonomous Vehicle Twin Hardware in the Loop Simulator

John Mikulskis

Abstract—We plan to create a Hardware in the Loop (HIL) simulation testbed for autonomous vehicle research to help achieve conditional automation (level 3 autonomy) with open-source self driving car software such as OpenPilot. Level 3 autonomy implies that a human is required to drive the vehicle, but the advanced driver-assistance system (ADAS) can take control of the steering, braking, and throttle when the environment demands it. We aim to do this by simulating a vehicle in software that communicates with a scaled down remote controlled (RC) car to receive real world sensor data, which will include a direct feedback loop to and from the software simulation.

1 Need for this Project

Autonomous vehicles are on the frontline of some of the most cutting-edge technology today, and many car owners and manufacturers want to take advantage of this new technology as soon as possible. The Society of Automotive Engineers (SAE) defines 6 levels of driving automation ranging from 0 (fully manual) to 5 (fully autonomous). These levels have been adopted by the U.S. Department of Transportation [1]. Different companies have their own proprietary solutions for achieving higher levels of vehicle autonomy, but open-source alternatives are starting to emerge as viable options for both research and even general use. For instance, open-source driver assistance systems, such as openpilot which was just released in 2016, are becoming advanced enough to lead to more stable solutions for autonomous vehicle control. Highly competent open-source car simulators such as CARLA, which was just released in 2017 at the Conference for Robot Learning (CoRL), help assist with this research and modeling, and are great tools for both researchers and amateurs breaking into the field. Although the CARLA simulator runs on software that uses physics modeling from advanced tools such as Nvidia's PhysX physics engine, having a real-world model to provide feedback for the simulation can lead to more accurate and realistic data for how the AI will perform on a car in the physical world.

CARLA runs as a layer for Epic Games' Unreal Engine 4 Game Engine (UE4), which provides advanced developer interfacing with actors in the simulation, including but not limited to adding sensors to the simulated car, and interfacing directly with parameters that the phsyics engine uses. Taking advantage of the vast features available in UE4 with CARLA's Client Server architecture will allow for both accurate and immediate feedback to the hardware and software component of the simulation. This has a direct use case for researchers in the autonomous vehicle field, and companies which require real testing of their artificial intelligence algorithms that may not have the means or safety measures for testing on real full-scale vehicles.

2 Problem Statement And deliverables

2.1 Problem Statement

As of today, cutting edge open-source simulation solutions have access to advanced physics and game engines, but testing the architecture of autonomous vehicles with this modeling is limited to readings from the software which oftentimes does not accurately capture many of the features of feedback due to real world physics. Our proposed solution would break into a new area of simulation, with direct feedback from both the software running with advanced game and physics engines, as well as sensors on a real scaled down version of an electric vehicle.

We believe that this new simulation architecture will help fill the need for accurately simulating and testing autonomous vehicle algorithms that need to take place in an environment that is as close to the real world as possible, with real world physics.

2.2 Deliverables

We plan to design and build a complete simulation architecture to control both the simulated and real car model that includes modern top-of-the line physical controls for a human actor using a Logitech G29 racing controller. This controller will interface directly with CARLA's simulation, providing human input to the software vehicle for throttle, brake, turning, and other basic controls. We will develop an API to communciate with the RC car, which will mirror the events enacted onto the CARLA vehicle (ego vehicle), as well as provide feedback to the CARLA server for corrections based on how the movements of the simulated car are compatible with the RC car in the real world.

The RC car will need to simulate a moving vehicle while staying in place, which we plan on modeling by using a conveyer belt that speeds up or slows down according to the speed of the ego vehicle in the simulation. The RC car will have 4 motors, each controlling a separate wheel, similar to the design of the car that our partner company uses. Control applied to the ego vehicle, such as the braking or turning of the wheels, will also be reflected in the RC car using these motors. We will control the vehicle in this way by using a microcontroller such as an arduino, which will run a script that communicates with the carla server via TCP, effectively making the RC car act as its own separate client. The human input from the Logitech G29 will interface with a client script on our simulation PC that communicates with the carla server via TCP as well.

Once we have the hardware set up correctly for these communication channels, we want to accurately receive data from the RC car in a format that is both reproducable and clear, such that we can viably use this data for both analysis and testing of how a real autonomous vehicle would perform. We plan to use our new simulation architecture to help achieve and show level 3 (conditional) autonomy with openpilot.

3 VISUALIZATION

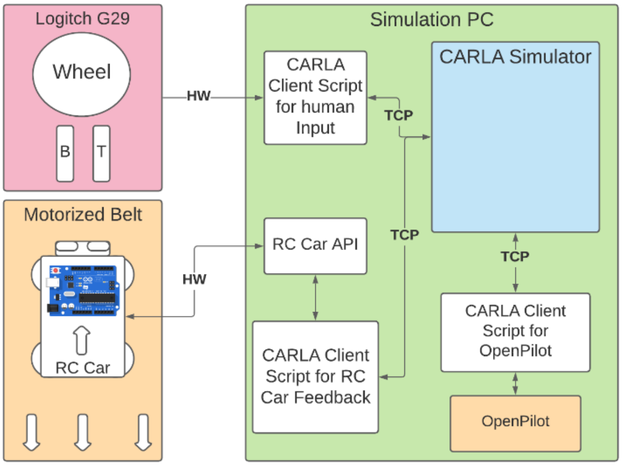

Fig. 1. This image shows the general architecture of the HIL simulator. The simulator receives control inputs from the Logitech G29 racing controller, RC car, and OpenPilot. All 3 have their own separate clients with connect to the CARLA Simulator via TCP.

4 COMPETING TECHNOLOGIES

There exist several different simulation technolgies that aim to create realistic simulations of the real world for a vehicle in software. Some of these simulators are extremely advanced, with the capability to set a variety of different physics parameters. Creating a HIL system for autonomous vehicle simulation with simulators such as CARLA is also not a novel idea in the research world, and different system architectures have been built and designed by various researchers.

4.1 Carmaker

Carmaker is a real-time capable simulator that can accurately model a wide variety of vehicle types along with their handling characteristics, the road and the surrounding environment, driver behavior, and the traffic situation in the virtual world [2]. Unlike CARLA, Carmaker is proprietary software owned by IPG Automotive, and it comes at a high price. This makes the software not as accesible to researchers and amateur users, and it is marketed toward an active customer base of automotive companies that can afford the software.

IPG Automotive also advertises their own HIL systems, which can connect to different Electronic Control Units (ECUs) to test how an ADAS system would perform in real time.

4.2 dSpace

Another highly advanced proprietary autnomous vehicle simulator that is available on the market today is dSpace. DSpace, like Carmaker, is less accesible to most users due to its high price, but it has advanced simulation features and HIL solutions avialable [3]. Although their software is proprietary, they have an extensive library of publically available documentation available to the general public, and they put a large emphesis on innovation and close customer cooperation.

4.3 Existing Research

Existing research on HIL simulation testbeds exists, and some of the newer autonomous vehicle HIL simulation soltuions even use CARLA. A former master's student at the University of Western Australia, Craig Brogle, wrote their thesis *Software Architecture and Hardware-in-the-loop Simulation for an Autonomous Formula SAE Vehicle* [4] which uses CARLA along with a Logitech G29 steering wheel and RC car to simulate a car in the software alongside its counterpart in the real world. This is the most similar parallel to what we are trying to achieve with our CARLA HIL simulator, since they also implemented a twin vehicle setup between a real-world RC car and the simulator in a research setting. The vehicle that Brogle uses drives freely in the real-world with its own camera on it, mirroring events in the simulator, while we plan to have a vehicle that is stationary and controlled due to a map environment in the simulation, not in the real world. Nevertheless, this paper presents an impressive HIL simulator setup, and it dives deeper into some of the viable use cases for simulating autonomous vehicles and the interesting features that CARLA has.

5 ENGINEERING REQUIREMENTS

The requirements for this autonomous vehicle research project include:

1. The simulator will have both a hardware and software component that should work in parallel with each other
2. The reactions of the CARLA ego vehicle in the software simulator should be accurately reflected to the RC car with a delay of no more than 1ms
3. The RC car should have sensors for useful data to grab metrics such as the rotation of the wheels, the orientation of the car, and the tilt of the car.
4. All client scripts should be able to freely send commands to the server via TCP.
5. All data collected from the RC car should be formatted in a way that is both reproducable, clear, and efficient.
6. The motorized belt should run at a speed that is accurately mapped to the speed of the car in the simulator, meaning that increasing the speed of the ego vehicle by some factor in the simulator should similarly increase the speed of the belt by the same factor.
7. All 4 wheels on the RC car must have their own

motor assigned to them that is programmable through the arduino

8. The hardware input from the human must be given higher priority than input from Openpilot, like how a real conditionally autonomous vehicle would operate.
9. Code must be well tested, with a test coverage of at least 60%
10. Use cases and examples must be documented so that they are easily reproducable such that the project can be handed off to future researchers with ease.
11. Our simulation when combined with openpilot should fulfill the Society of Automotive Engineers' level 3 (conditional) autonomous vehicle requirements with environmental detection capabilities and human override.

Acknowledgment

Special thanks to Prof. Rich West and Prof. Osama for their guidance and accomodations with the project.

References

[1] The 6 levels of vehicle autonomy explained. Synopsys Automotive. (n.d.). Retrieved October 22, 2021, from https://www.synopsys.com/automotive/autonomous-driving-levels.html.

[2] GmbH, IPG Automotive. “Carmaker.” CarMaker | IPG Automotive, IPG Automotive GmbH, 27 July 2021, https://ipg-automotive.com/products-services/simulation-software/carmaker/.

[3] “dSpace.” DSPACE, https://www.dspace.com/.

[4] Brogle, Craig, et al. "Hardware-in-the-loop autonomous driving simulation without real-time constraints." IEEE Transactions on Intelligent Vehicles 4.3 (2019): 375-384.