Experimental Testbed for Autonomous Vehicle Development

Vincent Onggoputra[a], Jerry Zhang[a]
Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089, USA

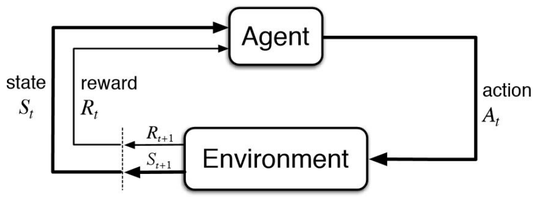
USC Viterbi School of Engineering

Abstract

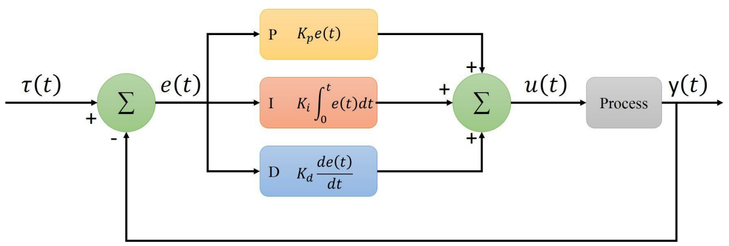
- Addressing the **challenge of autonomous driving** as a critical issue in transportation technology.
- **DesCyPhy Lab** focuses on developing innovative methodologies to design, verify, and optimize cyber-physical systems, particularly through our **Experimental Testbed for Autonomous Vehicle Development**
- Aiming to develop a testbed focused on supporting self-driving tasks and advancing **RL with logical constraints**
- Utilizing the established **F1-tenth platform** to build and implement custom control algorithms as a proof of concept.

Background

- **ROS (Robot Operating System):** A suite of software tools for building and managing robotic applications.
  - This allows for modularity in the project, where individual components (nodes) that perform specific tasks like camera control, motor management, and sensor data handling
- **RL (Reinforcement Learning):** A machine learning paradigm where an agent learns to make decisions by optimizing actions based on rewards received from the environment.



- **PID (Proportional, Integral, Derivative Control):** It is a feedback mechanism that helps maintain a desired setpoint by adjusting control inputs.
  - Utilized in the motor controllers to achieve the desired speed in a smooth manner
  - Utilized in wall-following algorithm to maintain a uniform distance from the wall



Methods

- **Drive Straight & PID Wall Following:** Implemented two custom controllers as a proof of concept
- **Sim-2-Real Fine-Tuning:** Adjusted algorithms for consistency between simulation and physical car behaviour
- **Steering Correction:** Resolved sporadic oscillation by dynamically adjusting steering angle
- **PID Parameter Adjustment:** Fine-tuned to enhance the wall following algorithms smoothness on the actual vehicle
- **Motor Controller Tuning:** Used VESC software for refined acceleration control
- **Camera SDK:** Set up for depth perception camera calibration and live video streaming

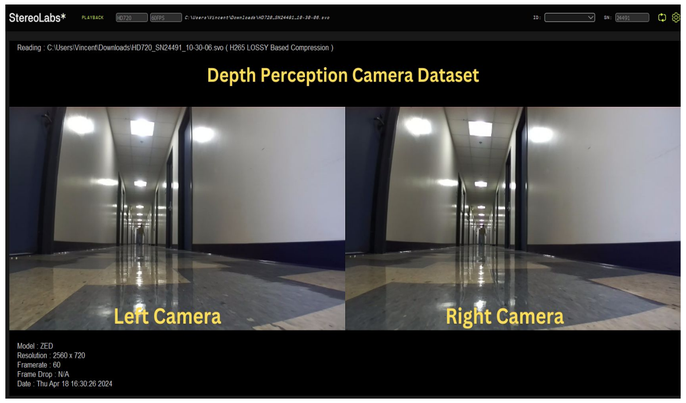


Figure 1. Successfully streamed video footage from the car's depth perception camera. This will be used to train computer vision models and reinforcement learning algorithms for autonomous driving.

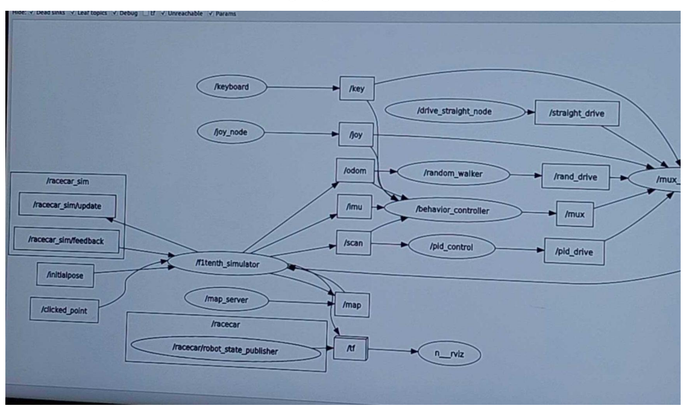


Figure 2. ROS rqt_graph for the RViz simulation, displaying both the custom drive_straight_node and pid_control being successfully connected to the proper topics.

Conclusion

- **Self-Driving Vehicle Testbed Project Overview:**
  - Aims to bridge the gap between theoretical research and practical application in autonomous systems.
- **Development of Testbeds:**
  - Machine learning (RL) with logical constraints is one promising approach in developing autonomous driving, but it has not been explored much in physical hardware
  - The F1-Tenth car was built to enable dynamic testing and analysis of these algorithms
- **Dataset and Training:**
  - Focus on creating datasets inspired by real-world environments - currently training on USC hallways
  - Post-dataset establishment this summer, the focus will shift to pushing forward reinforcement learning model with logical constraints for autonomous driving
- **Project Goals and Impact:**
  - Enhances the knowledge base and expertise in the autonomous systems field.
  - Aims to advance the safety, efficiency, and dependability of self-driving technology in real-world applications.

Figure 3. F1-Tenth Car (Hardware Based Testbed) - Equipped with LiDAR sensor and Depth Perception Camera.

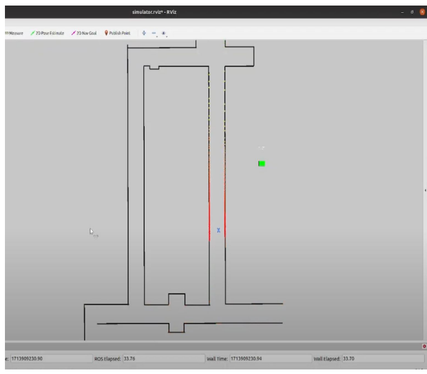


Figure 4. RViZ Simulator (Software Based Testbed) - Used to test self-driving algorithms, such as drive_straight and wall following PID algorithms.

References

Abbas, Houssam, et al. "F1/10 Reference Manual." *F1TENTH*, Aug. 2018, f1tenth.org/build/BuildV2.pdf

Luong, Kim, and Billy Hongrui Zheng. "F1tenth_simulator." Github, 5 Jan. 2022, https://github.com/f1tenth/f1tenth_simulator. Accessed 24 Apr. 2024.