The CAT Vehicle Testbed: A Simulator with Hardware in the Loop for Autonomous Vehicle Applications

Presenter: Rahul Kumar Bhadani, PhD Student
Authors: Rahul Kumar Bhadani, Jonathan Sprinkle, Matthew Bunting
The University of Arizona, Tucson, USA

http://csl.arizona.edu
Agenda

● Motivation
  ○ Hardware in the loop simulation in CPS

● Testbed Architecture
  ○ Virtual Environment
  ○ Physical Platform

● Modeling and Implementation
  ○ System Safety
  ○ Working with data
  ○ Demo with the Testbed

● Research Applications
  ○ 22-vehicles experiment
  ○ Applications on Domain Specific Modeling Language
  ○ REU Research

● Discussions and Future work
Hardware in the loop simulation (HILS) in CPS
Virtual Environment

Simulated World

- Uses Gazebo 2.2.3
- ODE Physics Engine
- Ability to manipulate behavior of simulated world
- Supports SDFormat for robot description
- Simulation can be performed in slower or faster than real time.
- Rich libraries to interface with ROS (the Robot Operating System)
System Abstraction:
Input \( X: f(v, \theta) \)
Output \( Y: f(x, v, \theta) \)
Significance of Vehicle Model in Simulation

- Runtime solvers approximate motion based on constraint satisfaction problems, which can be computationally expensive if the vehicle model's individual components are unlikely to approximate physical performance.

- Kinematic robotic simulation typically utilizes joint-based control, rather than velocity based (or based on transmission/accelerator angles and settings) like a physical platform.

- The dynamics of individual vehicle parts is such that physically unrealistic behavior may emerge, meaning that physical approximations of linear and angular acceleration should be imposed on individual joints, to prevent unlikely behaviors.
Virtual Environment

Vehicle Model

- Ackermann Steering Model for steering
Virtual Environment

**Simulated Sensors**

- Laser Range finder
- Side cameras
- Velodyne Lidar
The CAT Vehicle stands for the Cognitive and Autonomous Test Vehicle

- Modified Ford Hybrid Escape vehicle
- Emergency Stop
- Underlying protocol JAUS
- Developed JAUS-ROS Bridge to interface with Low Level Controller.
Physical Platform

The Perception Unit

Velodyne Lidar

Rangefinder

Pointgrey Side cameras

Bumblebee Stereocamera
HIL simulation mitigates the risk of failure or unintended action of controllers under test by extensive testing in the virtual environment with synthetic as well as real data and a combination of simulated and real sensors.

Design and Testing in software-in-the-loop simulation followed by hardware-in-the-loop simulation ensures that controller design not only meet the design requirement but it also remain safe to implement.

Another layer of safety package called as obstaclestopper is added for collision avoidance which uses rangefinder data to track minimum distance.

E-Stop in the physical vehicle in case of immediate emergency and software fails.
Data: velocity, brake, throttle, distance information, 3D data from velodyne, GPS Coordinates

Played back in realtime

Helpful in regression testing and debugging.

MATLAB Robotics System Toolbox to offline analysis
Demo with Testbed

- Download the testbed and compile them
  - git clone https://github.com/sprinkjm/catvehicle.git
  - git clone https://github.com/sprinkjm/obstaclestopper.git

- Simulation in Gazebo
- ROS Visualization
- Multi car simulation
- Modeling with Robotic System toolbox in Simulink
- Using code-generation feature to generate stand alone ROS node.
- How ROSBag file helps?
Research Applications

22-Vehicles Experiment

**Objective:** Testing hypothesis that sparse number of autonomous vehicles on the road can reduce congestions

**Outcome:** Dampening of congestions in terms of velocity standard deviation by 49.5% for one of the experiment.
Objective: Enabling non-expert programming for safety-critical applications such as autonomous vehicles

Outcome: 4th/5th graders were able to provide a path using DSML developed for the CAT Vehicle to follow.
CAT Vehicle Challenge

**Objective:** Producing most accurate visual of environment using least number of sensors on the CAT Vehicle for simulation purposes.

**Outcome**
Objective: This research experience for undergraduates (REU) is engaged in the myriad of applications that are related to autonomous ground vehicles.

Outcome: Several papers, improved CAT Vehicle testbed, Research experience for undergraduates
Outcomes


Discussion

- A Catvehicle Testbed provides an open-source, experimentally validated and scalable testbed with HIL support for autonomous driving applications that uses ROS.

- This work provides an overview of a multi-vehicle simulator that provides a virtual environment capable of testing a research application requiring vehicle to vehicle interaction from the inception of design to realization.

- We talked about a research paradigm that enables distributed teams to implement and validate a proof of concept before accessing the physical platform.

- Hardware-in-the-loop simulation increases development time and makes solution safer by increase test coverage.
Acknowledgement

- National Science Foundation under award numbers 1253334, 1262960, 1419419, 1446435, 1446690, 1446702, 1446715 1521617.

- Additional support from the Air Force Office of Scientific Research is provided under award 1262960.

Many thanks to contributors:
- Jonathan Sprinkle (sprinkjm@email.arizona.edu)
- Rahul Bhadani (rahulkumarbhadani@email.arizona.edu)
- Sam Taylor
- Kennon McKeever (kennondmckeever@email.arizona.edu)
- Alex Warren
- Swati Munjal (smunjal@email.arizona.edu)
- Ashley Kang (askang@email.arizona.edu)
- Matt Bunting (mosfet@email.arizona.edu)
- Sean Whitsitt

Github repo: https://github.com/sprinkjm/catvehicle
Lab page: http://csl.arizona.edu/
My webpage: http://math.arizona.edu/~rahulbhadani
Questions