An Overview of a Framework for Designing Robot Autonomy Stacks in Simulation

Thomas Hansen[1], Shouvik Chatterjee[1], Sriram Ashokkumar[1], Ishaan Mahajan[1], Harry Zhang[1], Stefan Caldararu[1], Abhiraj Dashora[1], Aaron Young[2], He Shen[3], Luning Fang[1], Dan Negrut[1]

[1] Department of Mechanical Engineering
University of Wisconsin-Madison
Madison, WI, 53705, USA
{thansen8,schatterjee9,ashokkumar2,
imahajan,hzhang699,scaldararu,dashora,
lfang9,negrut}@wisc.edu

[2] Department of Mechanical Engineering
Massachusetts Institute of Technology
Boston, MA, 02139, USA
aryoung@mit.edu

[3] Department of Mechanical Engineering,
The California State University - Los Angeles,
Los Angeles, California, United States
he.shen@calstate.edu

EXTENDED ABSTRACT

1 Introduction

Introducing new algorithms for use in autonomous vehicles can be difficult to test and may be facilitated by preliminary testing in simulation. Using physics-based simulators, in our case Chrono [1], one can recreate testing environments that in practice are cost prohibitive, unsafe, or impossible to recreate in a controlled manner. These environments can then be used to develop new algorithms and extract detailed system state information [2]. Unfortunately, these insights do not always lead to decisions that work well in reality due to the simulation-to-reality gap [3]. To address this, we developed an open-source platform for autonomy research. We placed an emphasis on the ability to operate these robots/AVs in off-road conditions and to add or remove sensors while seamlessly transitioning between simulation and reality. Our autonomy research testbed is tightly integrated with Chrono whose strengths are validated vehicle dynamics and terramechanics models.

2 System Overview

The proposed platform is a software environment development tool that contains two main components - an Autonomy Toolkit (ATK) and Autonomy Research Testbed (ART) which includes a physical vehicle, called V-1, and a digital version, called dtV-1 [4]. The hardware and software components of ART share the autonomy stack, which assembles all algorithms that enable the intelligence of the robot in both simulation and reality. The hardware platform uses a 1/6th scale vehicle shown in Fig. 2. Its suspension and steering mechanism closely mirror those of a full-sized vehicle, providing a payload capacity for the same class of sensors that full-sized vehicles in off-road conditions use. The digital twin is created using Chrono, which provides high fidelity simulation of vehicle dynamics, vehicle-terrain interactions, sensors, and multi-vehicle simulation. The software is built inside ATK using Docker to provide an OS agnostic platform. Setting up an autonomy stack is made easier by providing ROS bridges and interfaces to other system components to enable researchers to quickly compare and evaluate algorithms associated with the autonomy stack.

3 Autonomy Toolkit

ATK is a Python package that provides a containerized framework for developing, testing, and deploying autonomy algorithms in simulation and reality. ATK is built on top of Docker Compose, which allows its functionality to be agnostic of the platforms that it is deployed on. Further, other hardware and control stacks may implement their own containers and customize the default configurations, enabling containers operating outside the original scope of the toolkit to still work within the framework.

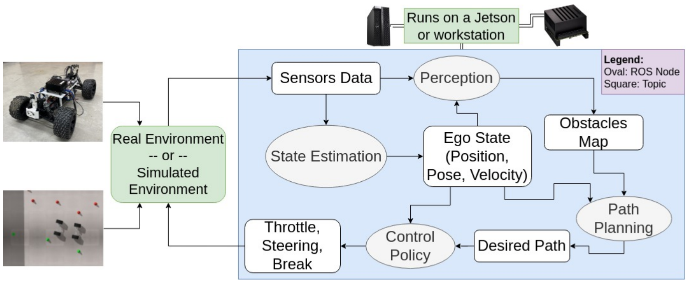

Figure 1: Overall design of the software and hardware platform.

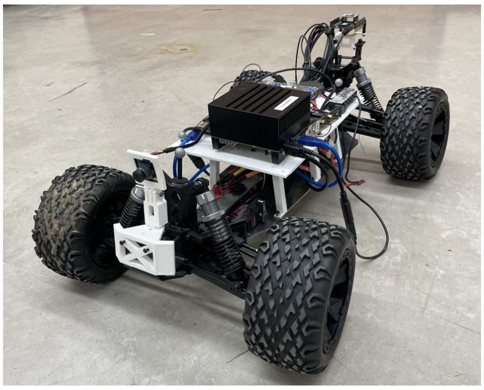

Figure 2: V-1 vehicle equiped with camera and MTi-7 sensor, LiDAR mount not pictured.

ATK is separated into “services” that can be combined to produce a container network that supports complex interactions across a variety of projects. The primary services we use are `dev`, `chrono`, and `vnc`. The `dev` container runs the autonomy stack, and is always built and run regardless of whether we are running in real life or simulation. The `chrono` container builds and runs the Chrono simulation environment, while the `vnc` container displays sensor information and ports it back to the user through a VNC viewer. Our multi container setup allows us to run `dev` on any hardware that is Docker and NVIDIA CUDA enabled. Specifically, the autonomy stack is run on Jetson hardware in the loop, while the `chrono` container runs on a more powerful workstation. This abstracts the “sensor and vehicle”, allowing the *same* autonomy stack running on the *same* embedded device (Jetson, in our case) operate either with an actual robot and its sensors, or with a Chrono simulation of the vehicle and its sensors.

4 Autonomy Research Testbed

The vehicle platform is modified from a 1/6th scale car. With a 47 cm wheelbase and a 34 cm track width, the vehicle is large enough to carry commonly used sensors for autonomy research. The base vehicle includes a double wishbone independent suspension at the front and rear and uses a 25 kg-cm servo for Pitman arm steering. The vehicle is powered with a 1300 KV brushless motor and is equipped with an onboard computer (Jetson Xavier AGX or NX), an IMU, GPS, and a USB camera. The base plate and all mounting components are designed for manufacturing through laser cutting or 3D printing. The final setup is shown in Fig. 2 with the electronics mounted above the motor and ESC on the base RC car. Additionally there is an optional VLP-16 lidar mount to lift the lidar clear of the rest of the vehicle. The camera is mounted to the front bumper while the IMU/GPS sensor is mounted in the rear. In addition to electronics and sensors, motion capture tracking targets can be mounted to the car using a scattering of holes across the vehicle.

To use simulation for designing and testing autonomous algorithms, a bridge to Chrono was developed to allow direct integration within the containerized system by passing JSON messages between ROS2 and Chrono. Generic publishers and subscribers, a feature of ROS2, are leveraged to allow for user defined message types and topic names to be used. Figure 1 shows how each of the major ROS nodes interact with each other, and how the `chrono` ROS bridge allows us to isolate the autonomy stack from knowing whether it’s in a simulation or a real environment. In Chrono we use dtV-1, or digital ART vehicle, which uses the TMeasy tire model, and has been calibrated through testing to match the vehicles throttle and steering input using Bayesian inference [5].

5 Conclusion

This brief outlines an open-source Autonomy Research Testbed (ART) whose purpose is twofold: conduct research in autonomy for wheeled vehicles in on/off-road conditions; and investigate the sim-to-real gap in robotics – understand what causes it, and how it can be controlled. Looking ahead, we will equip V-1 with additional sensors to test their use both in autonomy algorithms and to help close the sim to real gap in Chrono. Utilizing the available motion capture system and its millimeter position tracking, a richer family of sensors will allow us to investigate better sensor models, as well as perception, state estimation, planning, and controls algorithms, and understand how inaccuracy in different components of the autonomy stack propagate downstream and cause the sim-to-real gap. With this increased understanding of the senors we will be able to take V-1 into off-road conditions to further build our understanding of the vehicle.

Finally, ATK can be used to facilitate autonomy algorithm development by allowing researchers to configure a custom set of containers to host their development, deployment, simulation, and visualization needs. For instance, this platform is assisting UW-Madison students in their SAE-sponsored AutoDrive Challenge II, and is currently being used in our lab to improve the Chrono sensor package.

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] K. Liu and D. Negrut, “The role of physics-based simulators in robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 35–58, 2021. [Online]. Available: https://doi.org/10.1146/annurev-control-072220-093055

[3] N. Jakobi, P. Husbands, and I. Harvey, “Noise and the reality gap: The use of simulation in evolutionary robotics,” in *European Conference on Artificial Life*. Springer, 1995, pp. 704–720.

[4] 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.

[5] H. Unjhawala, R. Zhang, W. Hu, J. Wu, R. Serban, and D. Negrut, “Bayesian calibration models and scripts,” https://github.com/uwsbel/public-metadata/tree/master/2022/calibrationBayesianExamples, 2022, Simulation-Based Engineering Lab, University of Wisconsin-Madison.