Simulation-Based Testing of Autonomous Ground Vehicles

Christopher Goodin, Ph.D.
Daniel W. Carruth , Ph.D.
Lalitha Dabbiru , Ph.D.
Mississippi State University

Christopher H. Hudson
Lucas D. Cagle
Nicklaus Scherrer
Marc N. Moore
Mississippi State University

Paramsothy Jayakumar, Ph.D.
DEVCOM Ground Vehicle Systems Center

How do we study autonomy?

Functionality:

Sensing	Modelling	Planning	Action
Sensor noise Weather effects Sensor placement	Perception Classification Localization	Motion planning Path planning Safety monitor	Vehicle control Actuation Mobility

Sensing → Modelling → Planning → Action

Scenario:

Cross-country navigation
Pedestrian detection & avoidance
4-way intersection

1. Nakhaeinia, Danial, et al. "A review of control architectures for autonomous navigation of mobile robots." *International Journal of Physical Sciences* 6.2 (2011): 169-174.
2. Li, Li, et al. "Intelligence testing for autonomous vehicles: A new approach." *IEEE Transactions on Intelligent Vehicles* 1.2 (2016): 158-166.

Studying Failures

- When a failure happens, we would like to know[1]
  - **Where?**
  - **How**?
  - **What**?
- But there are challenges[2]
  - Built on statistics or AI algorithms
  - Environment is complex and random
  - Failures are (hopefully!) rare

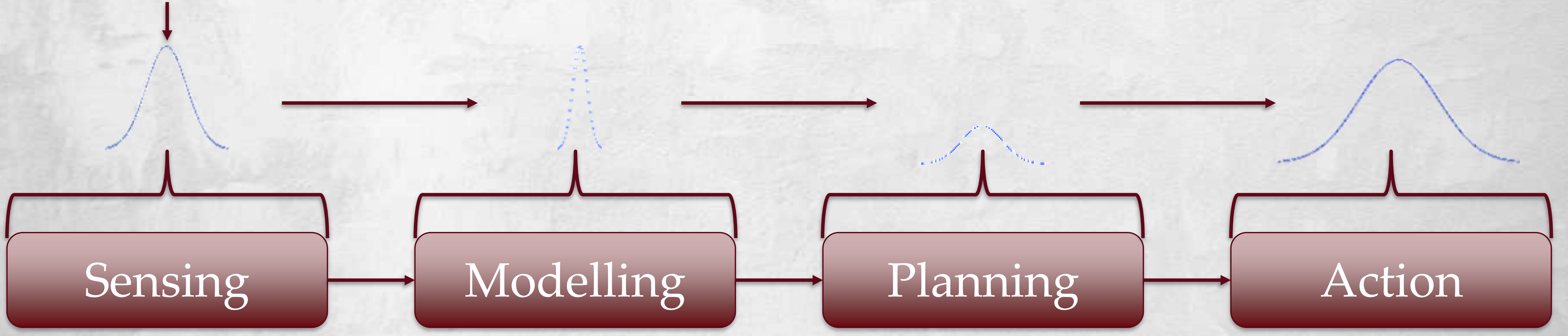
Sensing → Modelling → Planning → Action

1. Falco, Gregory, and Leilani H. Gilpin. "A stress testing framework for autonomous system verification and validation (v&v)." *2021 IEEE International Conference on Autonomous Systems (ICAS)*. IEEE, 2021.
2. Feng, Shuo, et al. "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment." *Nature communications* 12.1 (2021): 1-14.

Error Propagation in Autonomous Systems

- How do *external* errors propagate through the *internal* subsystems of the AGV?
- How resilient is each subsystem to errors? What errors cause failures?
- To answer these questions, we need:
  1. Ground truth information about the environment and vehicle state
  2. A way to control the environmental conditions
  3. Metrics for measuring performance/error of each subsystem

Error sources:
- Rain
- Dust
- Sensor noise



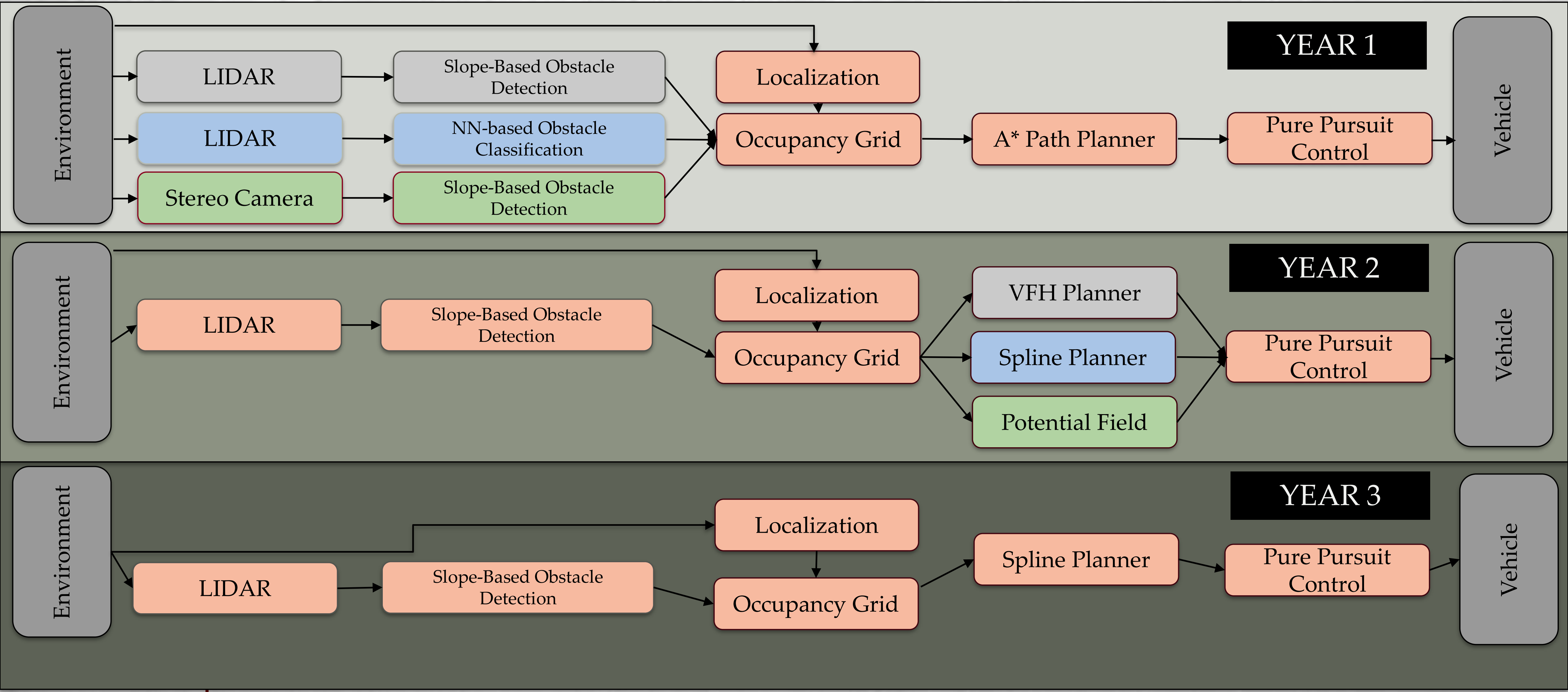
Simulation Based Testing

- Use simulation to mitigate each challenge associated with testing AGV
  - Simulation allows us to measure subsystem level metrics more accurately
  - Simulation offers perfect ground truth
  - Thousands of times more simulated hours than real
- Must constrain with real tests!

Research Overview

- **Goal**: Measure how environmentally induced errors propagate through the subsystems of an Autonomous Ground Vehicle (AGV) and manifest themselves in system-level performance
- **Method**: Use simulation to access ground truth and control environmental conditions
  - Y1: Perception
  - Y2: Planning
  - Y3: Physical Testing / Validation

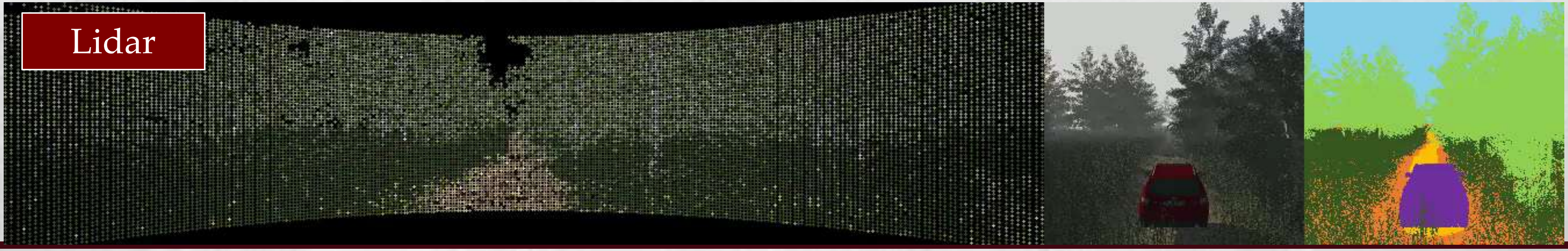
Autonomy Tested



The MSU Autonomous Vehicle Simulator (MAVS)

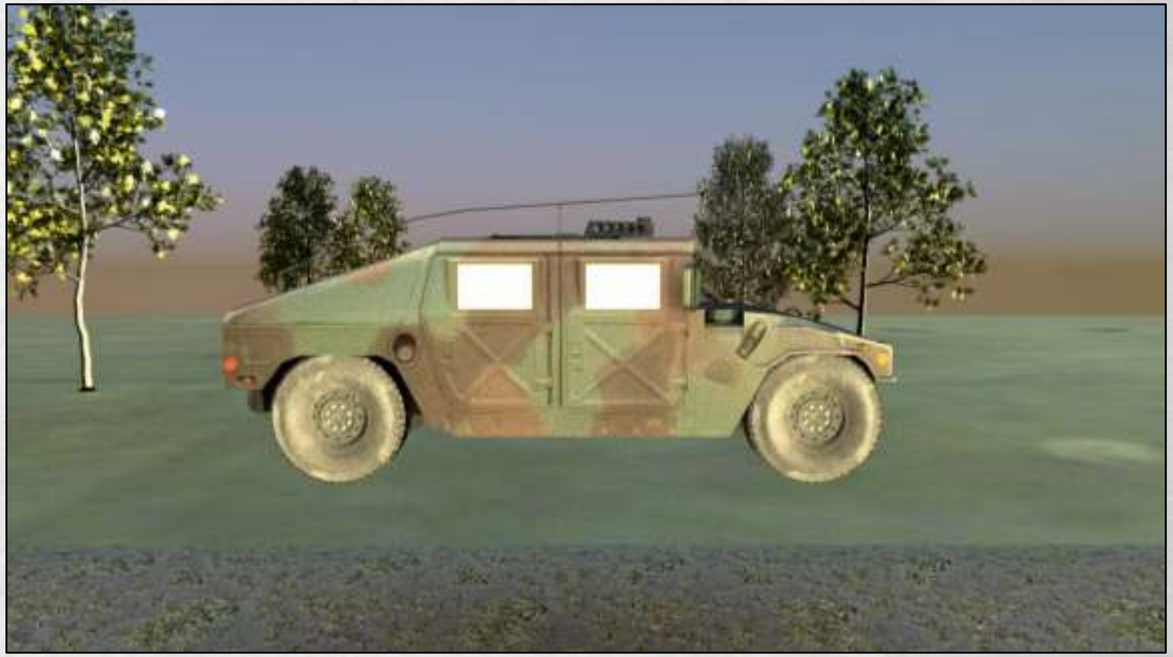
MAVS is:

- A C++ software library for simulating autonomous vehicles in realistic digital terrain.
- A real-time simulator for evaluating the performance of autonomous perception and navigation software.
- A user-friendly Python API for composing custom simulations.
- A physics-based sensor simulator for LIDAR, GPS, cameras, and other sensors.
- A lumped-parameter vehicle-dynamics simulator.
- ROS compatible

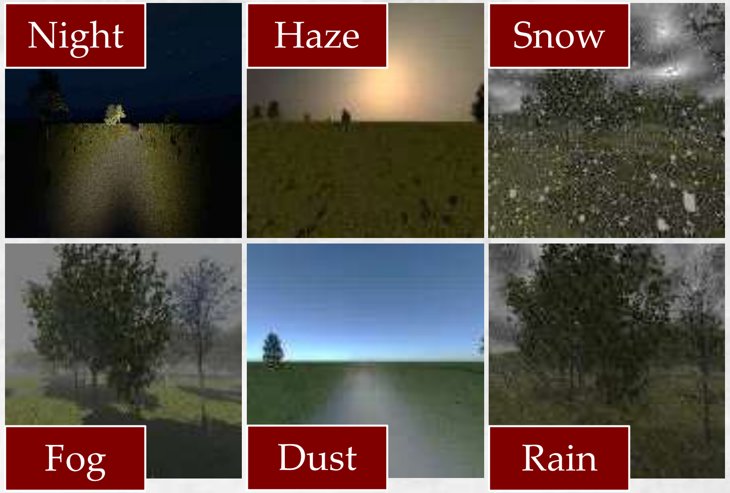


MAVS Vehicle Physics

- MAVS has a variety of vehicle dynamics options, including the default implemented in RP3D and links to Chrono::Engine
- Both physics engines were used at different points in this project



MAVS Sensor-Environment Interaction




electronics

MDPI

Article

Predicting the Influence of Rain on LIDAR in ADAS

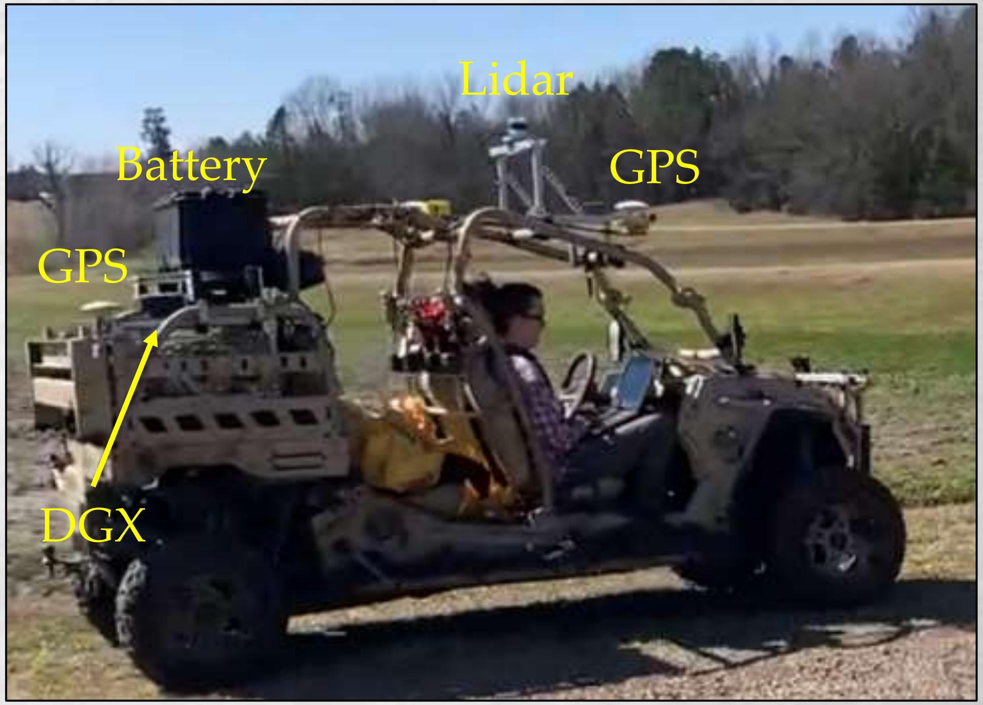
Abstract:

Keywords:

1. Introduction

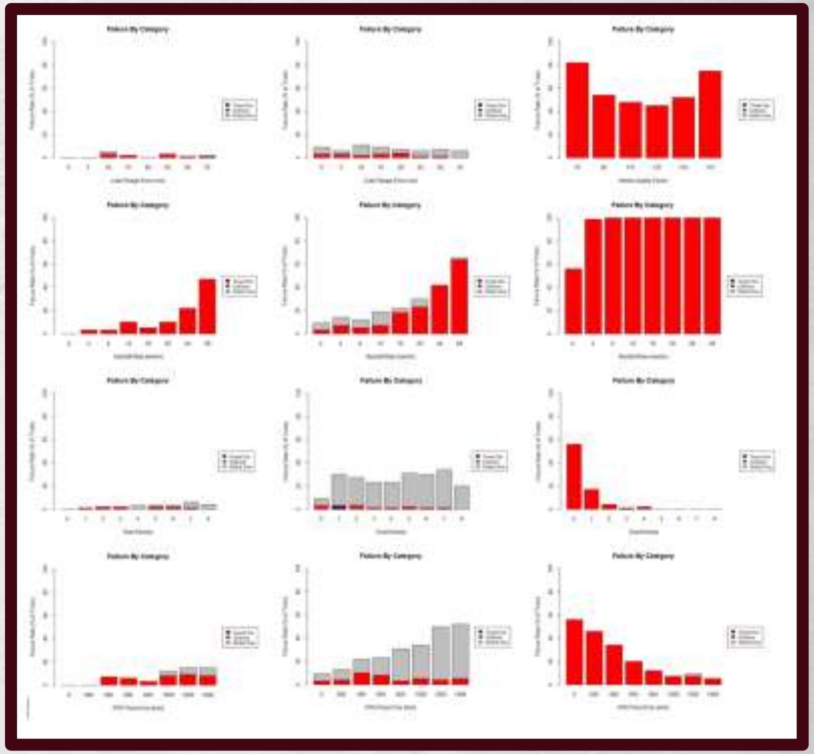
Test Vehicle

- Test vehicle is an MRZR-D4
  - Operated in Low-gear
- Sensors
  - Ouster OS-2 in 64 beam config
  - 2 X Swiftnav Piksi with RTK correction for heading and location
  - 3-axis IMU for pitch & roll
  - NVIDIA DGX interfacing to ROS



Year 1

- Perception: Lidar-slope, lidar-segmentation, stereo camera slope
- Independent Variables : Rain rate, Dust rate, Lidar range error, RTK localization error
- Performance Metrics
  - System level: Failure rate and type, speed
  - Subsystem level: Lidar range error, Occupancy grid error, Localization error, Path planner error
- Results
  - Stereo camera performs very poorly in rain
  - Failure rates increase with rain
  - System is robust to localization and lidar range errors
  - **<u>Subsystem metrics do not correlate well with system level metrics</u>**



Year 2

- Planners: VFH, Potential field, and Spline planner
- Independent Variables: Soil strength, Obstacle size, AGV speed
- Dependent variables:
  - System-level: Failure rate and type, vehicle speed, fuel use
  - Subsystem level: Occupancy grid error, Localization error, Path planner error
- Results:
  - Spline planner was the most robust
  - Softer soil caused increased fuel usage, but did not affect other system level variables
  - Still no correlation between subsystem-level metrics and system-level performance

Year 3

Year 3

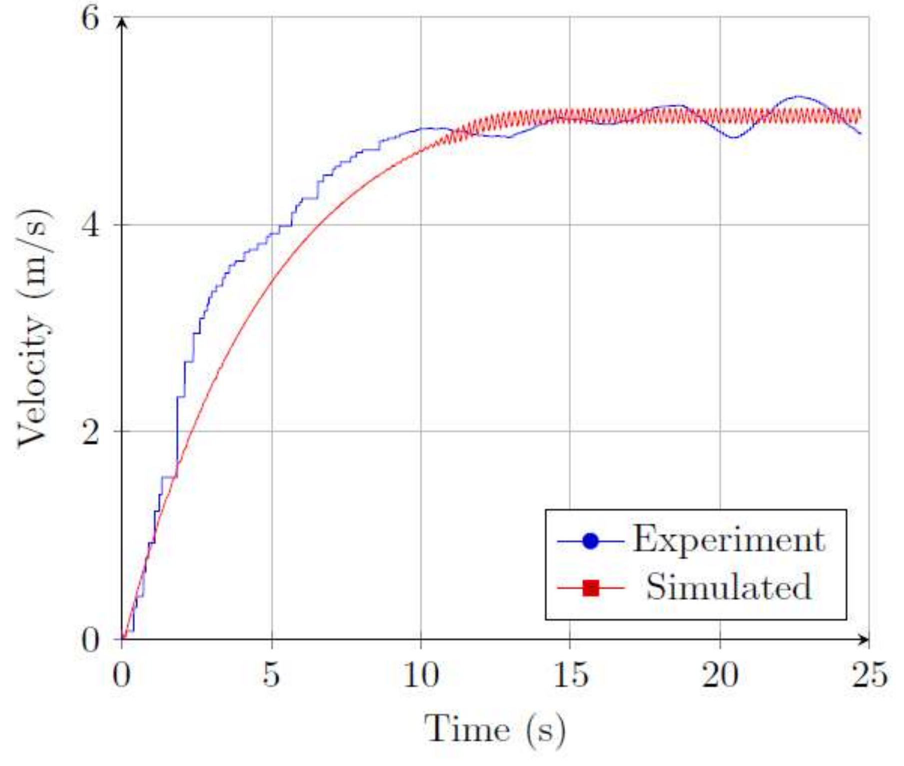
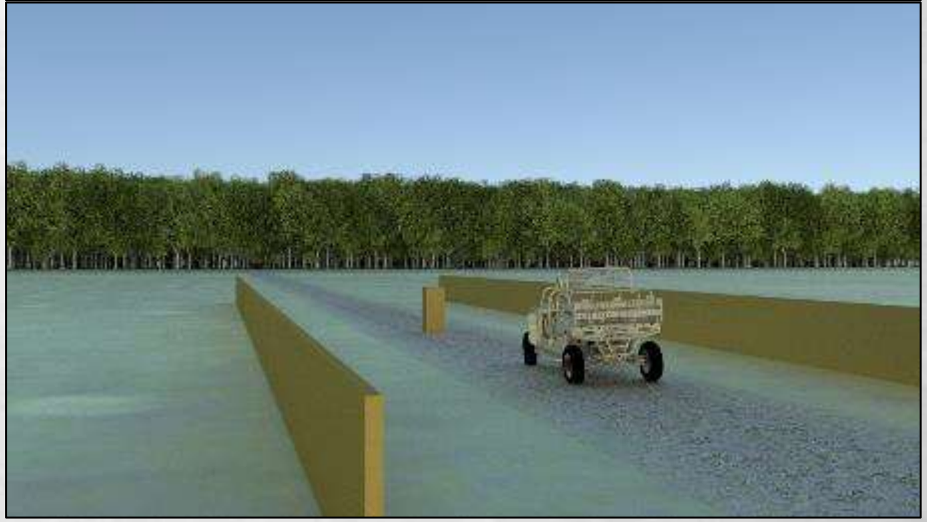


Figure 6. Real vs simulated vehicle speed



Summary / Conclusions

- We are finishing the third year of a 3-year project to study autonomy
- Primary conclusions from first 2 year:
  - Subsystem-level metrics do not correlate well with system-level results
  - Measured affect of rain, dust, sensor error, and soft soil on performance metrics
- Ongoing for year 3 – measuring ODOA with real vehicle for validation of simulation results