Measuring the Influence of Environmental Factors on Obstacle Detection and Avoidance with an Autonomous Ground Vehicle

Daniel W. Carruth*
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Christopher Goodin
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Lalitha Dabbiru
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Nicklaus Scherrer
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Marc N. Moore
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Christopher H. Hudson
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Lucas D. Cagle
Center for Adv. Veh. Systems
Mississippi State University
Starkville, Mississippi 39759

Paramsothy Jayakumar
US Army DEVCOM
Ground Vehicle Systems Center
Warren, Michigan, USA

Abstract

This paper presents the results of a comprehensive study of obstacle detection and avoidance (ODOA) by an autonomous ground vehicle (AGV) in off-road and adverse environmental conditions. This study included both real and simulated testing of an AGV operating in challenging conditions such as rain, dust, and deformable terrain. A novel approach for analyzing the environmental impact on each subsystem (perception, planning, control) of the vehicle was implemented in simulation and used to evaluate multiple options for planning and perception algorithms. This work is the most complete and systematic test campaign of its kind to be conducted on a publicly available autonomy stack and will facilitate the development of test strategies for AGV in future work. The primary contributions of this work are the development of a free and open source autonomous software stack for off-road AGV, a method for quantitative assessment of AGV systems, and incorporation of combined simulated and physical testing into a comprehensive test approach. This work demonstrates how simulation can be used to measure aspects of AGV performance that are impossible to measure in physical tests, giving additional insight into the functioning of the autonomy stack.

1 Introduction

While self-driving or autonomous vehicles have become more capable and have been studied extensively in recent years [Peterson and Glancy, 2018], there have been few attempts to systematically quantify the effect

*dwc2@cavs.msstate.edu

of environmentally induced errors from sources such as dust, rain, and soft soil on the performance of these vehicles. While the struggle of these vehicles with these environmental conditions has been well documented, qualitatively [Stock, 2018], there has been relatively little quantitative evaluation of these effects.

Past work in quantitative error measurement tends to focus either directly on the sensor data or on the overall system-level performance. For example, there have been several studies that quantify the influence of phenomena like rain or snow on lidar sensor performance [Rasshofer et al., 2011]. Similarly, there have been some laboratory studies on the effect of dust on lidar [Goodin et al., 2013]. System-level analyses have focused on high-level metrics like average speed or distance traveled [Durst et al., 2017].

While there is value in sensor-level or system-level error studies, an understanding of how error propagates through the chain of autonomous subsystems is also needed. How do errors in lidar sensors operating in rainy conditions affect the point cloud, the navigation maps derived from the point cloud, and the path planned through those maps? Is there a level of rain for which the resulting path and map are not significantly affected, and at what rain rates would errors start to manifest in the planned path? Answering these questions is critical as self-driving vehicles transition to the consumer market.

The major difficulty in systematically measuring the relationship between environmentally induced sensor errors and system-level performance is the difficulty in controlling the environmental error sources. Physical tests involving these factors are both logistically difficult and prone to uncertainty in the input variables. It is impractical and expensive to perform repeatable, controlled experiments in conditions like dust, rain, or soft soil. In contrast, physics-based simulation provides a way to systematically study the effect of these phenomena on autonomous vehicle performance [Goodin et al., 2017]. In order to address these limitations, this study develops a method, using simulation, for studying error propagation through the subsystems of an AGV. These tests are validated through comparison to physical testing of the real vehicle on both hard and soft soil.

In the following sections, a review of related work in this area is presented (Section 2), followed by a summary of the method and approach of this multi-year study in Section 3. Next, a detailed description of the vehicle platform, sensors, and autonomy studied in this work (Section 4) is presented. The Mississippi State University Autonomous Vehicle Simulator (MAVS), which was a critical enabler for this work, is presented in Section 5. The test scenarios and metrics studied in this work are presented in Section 6, with the results of the experiments shown and discussed in Section 7. Some final conclusions are presented in Section 8.

2 Background and Related Work

There has been considerable research in recent years on the testing of AGV [Huang et al., 2016]. Testing is important both for determining the safety of commercial self-driving vehicles [Kalra and Paddock, 2016] and for evaluating the performance of military vehicles relative to specifications [Durst and Gray, 2014, Bostelman et al., 2016]. Early attempts at establishing consistent testing methodologies were driven by the DARPA Grand Challenges [Koon and Whittaker, 2006], with further development going towards documenting the state of the environment on test results [Sun et al., 2014]. Recent work by the European New Car Assessment Program (NCAP) has led to standardization of specific, reproducible test scenarios [Op den Camp et al., 2017]. Additionally, the United Nations Regulation 157 governs testing of automated lane keeping systems (ALKS) [Mattas et al., 2022]. These standards, in turn, promote the possibility of autonomy developers “building to the test”. To alleviate this possibility, significant work has been done to automatically create scenarios for detecting failure modes and edge effects [Mullins et al., 2017, Mullins et al., 2018].

Most of the work on testing discussed above is the context of a scenario - an AGV being tested performing a certain navigation or safety task to evaluate a certain requirement. This is inherently a test of the entire vehicle system. In fact, Li *et al.* [Li et al., 2016] identify two main classifications of tests of autonomous systems - *scenario*-based testing and *functionality*-based testing. Scenario-based testing puts a vehicle in a real situation and measures its performance, whereas functionality testing considers the components of AGV systems individually. To quote [Li et al., 2016]

> The most important benefit of functionality-based test is that we could quantitatively evaluate a part of driving intelligence within some specially designed tests. This benefit cannot be easily obtained in... scenario-based tests.

In this work, we will refer to Li's scenario-based testing as "system-level" tests and functionality tests as "subsystem-level" tests.

Li *et al* [Li et al., 2016] identify the three main subsystems of AGV as *perception*, *decision*, and *action*. Earlier references refer to these stages as *world modeling*, *planning*, and *action* [Nakhaeinia et al., 2011]. In the context of obstacle detection and avoidance in off-road scenarios (see, for example [Manduchi et al., 2005]), we will identify these subsystems with the three which will be studied in this work, namely, *perception*, *planning*, and AGV *control*. Further background on these subsystems will be given in Sections 2.1 and 2.2.

There may be many purposes for AGV testing. Consumer protection agencies like the NCAP may be primarily interested in measuring failure rates or other performance scores. In contrast, AGV developers may be testing to gain a deeper understanding about the AGV in question. In this case, Falco and Gilpin propose several important questions for evaluating an AGV system failure[Falco and Gilpin, 2021]. These include *what* did not perform as expected, *where* did the failure occur on the system, and *how* did the failure transpire?

When considering the current state of AGV testing presented above, it is concluded that a comprehensive test strategy should include a method for functionally testing the perception, planning, and control of an AGV, detecting system level failures, and determining the *what*, *where*, and *how* of each failure. In recent years, it has become increasingly clear that any such comprehensive test strategy must include simulation [Schöner, 2018].

As pointed out by a recent article in *Nature*, the testing of autonomous vehicles face three primary challenges [Feng et al., 2021]:

1. "The driving agent in (autonomous vehicles) is commonly developed based on statistics or artificial intelligence (AI) algorithms"
2. "The driving environment is usually complex and stochastic"
3. "Events of interest (e.g., accidents) for the driving intelligence test rarely happen"

These three considerations make simulation a necessity for comprehensive testing of autonomous vehicles. Concerning challenge #3, an oft-cited RAND study pointed out that, at expected failure rates, a vehicle may need to be driven hundreds of billions of miles to accurately assess reliability [Kalra and Paddock, 2016]. While [Feng et al., 2021] introduce methods for overcoming challenge #3 by focusing on "adversarial" testing, the research presented in this work focuses primarily on challenge #2 - the complexity and randomness of the driving environment. Because of the complex and stochastic nature of the real environment, it can often be difficult to determine exactly what feature of the environment caused a failure - in other words, even careful testing may not reveal the *where*, *how* and *why* of a failure mode because the "ground truth" environmental information may not be known.

In this work, this limitation on ground-truth availability is overcome by using a high-fidelity, physics-based simulation tool, the MAVS, to reproduce the complexity and randomness of the real world while also offering perfect knowledge of environmental ground truth, allowing the *where*, *how* and *why* of AGV testing to be studied more comprehensively than in previous work.

In order to conduct a study of AGV error propagation that is generalizable to many vehicles, a flexible autonomous architecture which allowed us to test multiple algorithms for the planning and perception subsystems was designed and implemented. Some background on the history and current state-of-the-art in these areas is given in the following subsections in order to give context for the results of this study. In addition, because simulation is a critical component of this work, a review of background work in simulation of AGV is presented in Section 2.3.

2.1 Perception

The *perception* subsystem takes raw sensor input and processes it to develop a world model for vehicle navigation tasks. Sensor input could be proprioceptive or exteroceptive. "Proprioceptive sensors make measurements of the internal state of the vehicle... Exteroceptive sensors make measurements of the external state surrounding the vehicle." [Adams et al., 2007] Proprioceptive sensors include accelerometers and wheel encoders, while exteroceptive include lidar and cameras. Although proprioceptive algorithms have been used in off-road mobility for quite some time (see [Welch and Connolly, 2006, Borja et al., 2009, Stavens and Thrun, 2012, Durst and Goodin, 2012], and, more recently [Gregory et al., 2021]), the scope of this study was limited to exteroceptive algorithms.

Many early exteroceptive perception algorithms for off-road navigation focused on extracting information from geometric properties of lidar point clouds [Matthies et al., 1996, Matthies and Rankin, 2003, Matthies et al., 2005, Manduchi et al., 2005, Kelly et al., 2006], although significant early work on stereo camera processing [Narayanan et al., 1998] eventually became viable for off-road navigation projects [Rieder et al., 2002]. Since this early work, the field of perception in off-road navigation has been greatly influenced by artificial intelligence and machine learning (AI/ML) algorithms. Several recent survey articles summarize the current state-of-the-art in this field for both on-road [Grigorescu et al., 2020, Ma et al., 2020] and off-road [Hu et al., 2020] applications.

In this work, well-known and well-studied perception algorithms were favored over the most cutting-edge algorithms currently available. This is in line with the goal of this project to study general aspects of autonomy, rather than the intricacies of any specific algorithm. The perception algorithms chosen for this work are discussed in more detail in Section 4.2.1.

2.2 Planning and Control

The *planning* subsystem determines the desired action of the vehicle. In the context of autonomous navigation, planning can occur on many levels, ranging from general intent (*ie* arrive at certain coordinates at a certain time) to very specific navigation tasks (*ie* steer left around an obstacle). Planning tasks can usually be delimited by their temporal and spatial scale relative to the vehicle. In autonomous navigation by AGV, planners are generally divided into *global* and *local* planners [Wang et al., 2002]. Global planners work on a task scale which may extend beyond the range of the AGVs exteroceptive sensors, while local planners work in a near-field, reactive range and perform tasks like obstacle avoidance. In this work, local planners will be evaluated, while the global plan will be considered an input to the system.

Local path planning, also known as motion planning, for mobile robots is an extensive sub-field. One of the most commonly used and well known planning algorithms is A* [Hart et al., 1968]. Furthermore, in the last decade, a review of motion planning algorithms [Oroko and Nyakoe, 2012] listed the potential field method [Barraquand et al., 1992], the vector field histogram [Borenstein et al., 1991], and the bug method [Lumelsky and Stepanov, 1987] as the state-of-the-art methods at that time. In a more recent review article, potential field, vector field histogram, and bug were again listed, along with the roadmap method, the cell decomposition method, and newer techniques that rely on machine learning; techniques such as genetic algorithms, fuzzy logic, and artificial neural networks [Campbell et al., 2020].

In comparison to the planning subsystem, the control subsystem takes the output of the planner and calculates actionable commands for the hardware actuators of the vehicle. In the case of an AGV, the control subsystem may take a local path from the planner as input and output throttle, steering, and braking settings to the hardware interfaces of the actuators of those components. In fact, steering, braking, and throttle may each have independent control algorithms. A recent review identified pure-pursuit [Coulter, 1992], the Stanley algorithm [Amer et al., 2018], and PID [Al-Mayyahi et al., 2015] as popular choices for AGV controllers [Yao et al., 2020].

In addition to these planning and control algorithms, many AGV systems use model predictive control (MPC) [Wurts et al., 2020], which is a combination of planning and control into a single algorithm that relies on a model of the vehicle motion to plan an optimal trajectory. Other recent algorithms plan a path under vehicle constraints by selecting from a finite number of paths deviating from a road centerline [Hu et al., 2018]. In this work, a variety of different planners in conjunction with the controller from [Hu et al., 2018] are studied. Similar to the algorithms chosen for perception, well-known and well-studied path planning algorithms are favored over cutting edge algorithms, in line with the goals of this study. The controllers used in this work are discussed further in Section 4.2.3.

2.3 Simulation

Recent overviews of simulators for AGV [Carruth, 2018, Feng et al., 2021] listed several of the most commonly currently used robotic simulators. These include the NVIDIA Drive Constellation [NVIDIA, 2022], CARLA [Dosovitskiy et al., 2017], AirSim [Shah et al., 2018], CarCraft[Madrigal, 2017], and AADS[Li et al., 2019]. These simulators fall into two main categories. Those in the first category, which includes CARLA and AirSim, use a game engine (Unreal Engine 4) to produce synthetic lidar and camera data for testing the simulated vehicle. The second category, which includes Drive, CarCraft, and AADS, do not produce fully synthetic sensor signals [Fadaie, 2019] and instead use a data-driven approach to generating simulated scenarios. This is similar to the approach used by many recent studies [Browning et al., 2012, Li et al., 2017, Jha et al., 2018, Jha et al., 2019, Michelmore et al., 2018].

The advantage of the fully synthetic approach is the ability to test any scenario that can be created with the game engine. The advantage of the data-driven approach is better agreement to real data for the empirical scenarios for which data is available. However, when it comes to off-road simulation, all of the simulators listed above have important deficiencies. These deficiencies relate to the way the sensors and vehicle interact with the simulated terrain, as discussed in [Letherwood and Jayakumar, 2021].

Off-road environments may have areas of soft-soil, even soil that cannot support navigation by a wheeled vehicle [Stevens et al., 2013]. Simulators like AirSim and CARLA cannot accurately reproduce the tire-soil interaction, the effect of which has been well-documented [Mason et al., 2018].

In addition, the lidar simulations supported by game engines like Unreal Engine [Karis and Games, 2013] do not support physics phenomena that are important for calculating lidar ranges like laser beam divergence. While the error introduced by this approximation may be negligible for solid surfaces typically found in urban, on-road environments, beam divergence effects may become quite pronounced when scanning extended objects like vegetation [Macedo et al., 2001, Larson and Trivedi, 2011, Goodin et al., 2018].

In this work, these limitations are overcome by using a physics-based simulator which does not rely on game engine technology, the MAVS. MAVS uses the same fully synthetic approach as simulators like AirSim and CARLA but without the limitations of on-road game engines. The MAVS is used to accurately simulate lidar interaction with vegetation [Goodin et al., 2018], rain [Goodin et al., 2019], and dust. Additionally, realistic soft-soil tire-terrain interaction is simulated in MAVS using algorithms from the DROVE database

[Vahedifard et al., 2017].

3 Method

One goal of this work was to use simulation to measure how system-level performance metrics correlate with subsystem-level metrics for planning and perception algorithms. By defining both system-level and subsystem-level metrics and using the simulator to measure perfect ground truth, the propagation of error through the system can be measured quantitatively. In addition, by comparing multiple perception and planning algorithms, it was possible to distinguish between effects that were general to the autonomous system and those that are specific to certain sensors and algorithms.

In order to perform scenario-based testing, an obstacle detection and avoidance (ODOA) test was selected as the baseline capability being tested in this work. ODOA is a critical capability for any autonomous or semi-autonomous system [Oroko and Nyakoe, 2012]. Since the focus of this work was on studying the generalized autonomous system, straight line ODOA was chosen for its simplicity and applicability to nearly any autonomous system.

The study presented in this article proceeded in three phases. In phase 1, the initial autonomous architecture was designed, built, and tested in simulation with the MAVS. The algorithms and simulation were integrated using ROS [Quigley et al., 2009]. Three different perception algorithms were studied in simulation, with the planning and control algorithms held constant. Appropriate performance metrics for the system-level performance and the subsystem-level performance were also developed in phase 1. Performance metrics are discussed in further detail in Section 6.1.2.

Phase 2 of the project focused on path planning. The perception and control were held constant, and three different planning algorithms were studied. The metrics from phase 1 were also used in phase 2. Like phase 1, phase 2 was completed entirely in simulation.

Finally, in phase 3, the autonomy stack that was developed in simulation in phases 1-2 was implemented on a real robotic vehicle. Using the same test scenarios and metrics that were developed in the previous phases, the autonomous robot was tested in both real-world experiments and simulated experiments in MAVS. The autonomy stack was refined during the real-world experimentation, which was conducted on both hard and soft soil. Phase 3 provided insight into the influence of soft-soil on AGV performance and served as a validation effort for the simulated experiments conducted in phases 1 and 2.

4 Autonomous Ground Vehicle

This section describes the hardware and software used on the both the real and simulated vehicle, which was an MRZR-D4 equipped with a real-time kinematic (RTK) sensor [Skoglund et al., 2016] for odometry measurements and a 3D lidar sensor.

4.1 Vehicle and Sensors

The vehicle used in these experiments was the Polaris MRZR-D4 with rear-mounted diesel engine. ODOA requires sensors that can generate information regarding the spatial location of obstacles in the environment. While there are a variety of sensors that can produce this type of spatial information, in this work the scope was narrowed to focus on two primary sensing types – stereo cameras and lidar. These sensors were chosen because, in contrast to range-based sensors like automotive radar, both can produce a 3D point cloud as output. This allowed the rest of the autonomous architecture to remain essentially unchanged when the sensor type was changed. While the lidar was tested on both the real and simulated vehicle, the stereo camera was only tested in simulation.

The lidar was a 64-beam Ouster OS1 set to single strongest return mode. In the phase 1 and 2 simulations, the lidar was mounted near the center of the roof of the vehicle in a standard vertical orientation, resulting in a sensor mount height of about 1.8 meters above the ground. In the phase 3 experiments and simulations, the lidar was mounted on the front pushbar of the vehicle at a height of about 0.75 meters. The rotation rate was set to 10 Hz and the point cloud was published to a ROS PointCloud2 message after each rotation.

The simulated stereo camera consisted of two machine vision cameras mounted on the roof of the vehicle, laterally colinear with the lidar sensor. The stereo baseline (horizontal distance between cameras) was 1 meter. This long baseline was necessary to ensure a reasonable accuracy in the range measurement at distances greater than 10 meters. Each camera had a resolution of 640x480 pixels, imaging plane dimensions 2.4x1.8 mm, and a focal length of 6 mm.

In order to generate a point cloud from stereo images, an intermediate step was required in which the disparity, or pixel-distance between corresponding features in the two images. The ROS "stereo_image_proc" package was used to calculate the disparity between the two images. This package requires the minimum and maximum distances for the disparity calculation to be set. In this work, the minimum distance was set to 2 meters, while the maximum distance was set to 45 meters.

4.2 Autonomy Software Stack

The software stack developed for this project is known as the NATURE (Navigating All Terrains Using Robotic Exploration) stack. The primary purpose of developing the NATURE stack was to create a navigation stack that was 1) purpose-built for off-road, 2) modular enough to swap out different elements, allowing comparisons between subsystems, 3) optimized for ackermann-steered vehicles, and 4) free and open source. To this last point, the NATURE stack is available on github at `https://github.com/CGoodin/nature-stack`.

The notional AGV software architecture for this project is shown in Figure 1. The vehicle autonomous software was conceptually divided into three parts: sensing/perception, path planning, and vehicle control. The architecture was implemented in ROS [Quigley et al., 2009]. In the autonomy system being studied, a 3D point cloud was generated by the lidar sensor or a stereo camera system. Perception data from one of the three algorithms was generated at a rate of 10 Hz and registered using the odometry published by the simulated RTK sensor.

A cost map was generated by one of three sensing-perception combinations. A path through the cost-map was calculated using one of three path planning algorithms, as shown in Figure 1. Throttle and steering commands were calculated from the path using the pure pursuit algorithm [Coulter, 1992]. Each of these is discussed in more detail in the following subsections.

4.2.1 Perception

The two sensor packages were paired with two different perception algorithms to form three unique combinations of sensing-perception pipelines. All three algorithms would create an occupancy grid from a point cloud. At each time step, the point cloud was registered to world coordinates using the current odometry.

The first perception algorithm was purely geometric. The algorithm compared each incoming point cloud (either from the lidar or stereo camera) to an existing 2D grid. Each grid cell stored the previously measured highest and lowest point in that cell. The cell was updated if the incoming scan had a point in the cell that was higher or lower than current values. The slope of the cell was found by taking the difference between

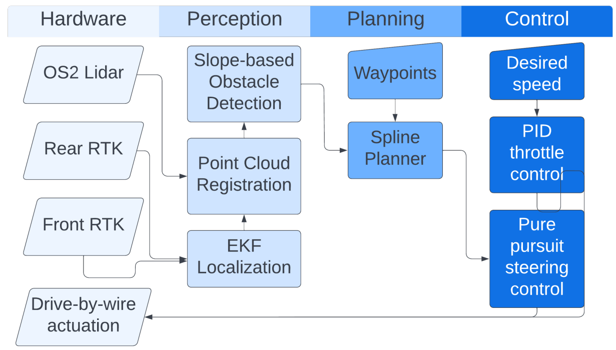

Figure 1: Autonomous architecture tested in this work.

the highest and lowest point and dividing by the cell width. If a cell had a slope greater than a certain threshold, it would be marked as an obstacle; otherwise the cell was unoccupied. In this work, the cell size was 0.5x0.5 meters, and the slope threshold was 1.0, meaning that obstacles greater than 0.5 meters tall would be recognized by the algorithm.

The second perception algorithm was adapted from SqueezeSeg, a Deep-learning Neural Network (DNN) that was trained to recognize obstacles from thousands of labeled point clouds - automatically generated and labeled with MAVS. This algorithm was capable of pointwise labeling the entire point cloud. The labeled points were placed in a grid structure similar to the grid used by the slope algorithm. Any cell that contained points labeled as "tree" or "obstacle" was marked as occupied. Because the DNN was trained on point cloud data only from the lidar sensor, it could not be used in conjunction with the stereo camera. The lidar segmentation algorithm is described in more detail in Dabbiru *et al* [Dabbiru et al., 2020].

4.2.2 Path Planner

Several path planning algorithms and their influence on system and subsystem performance of the autonomous architectures were considered. There are two main classes of path planning algorithms – those that use a grid-based world model and those that use a feature-based world model. There are also some path planning algorithms that use a combination of grid- and feature-based.

The A* algorithm [Hart et al., 1968] was implemented and used in the first year as the placeholder local path planner [Zeng and Church, 2009]. A spline-based planner was implemented based on [Hu et al., 2018, Hudson et al., 2018]. Finally, a vector field histogram (VFH) algorithm [Borenstein et al., 1991] was implemented as the third planner. All planners were implemented as ROS nodes with occupancy grids as obstacle inputs. While A* can work directly on the occupancy grid, the grid was converted to discrete obstacles for the spline-planner and VFH planners. The grid was updated at 10 Hz with new point cloud data. After each update, the path was re-planned with the current grid and the updated vehicle position.

4.2.3 Controller

The vehicle control subsystem used the pure pursuit algorithm [Coulter, 1992] and a PID controller [Islam et al., 2021] to convert the proposed path into throttle and steering commands, respectively. In the first step of the pure pursuit algorithm, a local goal point is found on the path at a look-ahead distance along the path from the current vehicle position. The look-ahead distance is proportional to the vehicle speed. Next, the steering angle necessary to reach the goal point is calculated from the vehicle wheelbase using a bicycle model of the vehicle.

Table 1: Vehicle parameters used in the MAVS

Parameter	Value
Tire section width (m)	0.2286
Tire diameter (m)	0.6604
Tire section height (m)	0.1651
Clearance height (m)	0.33
Tire deflection (m)	0.0313
# tires/axle	2
Sprung mass (kg)	1140.6
CG to front axle (m)	1.57
CG to rear axle (m)	1.15

5 Simulator - MAVS

The AGV simulator used in this work was the MAVS [Hudson et al., 2020, Goodin et al., 2018]. The MAVS provides a software library for physics-based simulation of lidar, camera, GPS, and other sensors. In this work, the MAVS library was integrated with ROS such that the simulated sensor data was published to standard ROS topics such as “PointCloud2”, “Image”, and “Odometry”.

The environment in MAVS is modeled as triangular meshes. In addition to RGB reflectance, each material definition in MAVS includes a reflectance spectrum that covers the UV to IR ranges, including the 905 nanometer reflectance that is relevant for the lidar simulated in this work. In addition, MAVS simulates the effect of rain using the model presented in [Goodin et al., 2019]. Dust is simulated in MAVS using a particle system model [Chen et al., 1999] with optical properties derived from laboratory measurements of lidar-dust interaction [Goodin et al., 2013].

Two different vehicle dynamics models were used in this project. In phase 1, the Chrono multibody dynamics engine [Tasora et al., 2015] was used to simulate the dynamics of the vehicle. Chrono was cosimulated with the MAVS sensor and terrain models.

In phases 2 and 3, the MAVS vehicle dynamics simulation was used. MAVS uses the Reactphysics 3D multi-body dynamics library and a lumped parameter vehicle suspension model along with a terrain-enveloping radial spring model [Davis, 1975], giving more realistic performance over rough terrain and more accurate results for tire load versus deflection. The parameters used in the MAVS vehicle dynamics model in the simulations presented in this work are shown in Table 1.

The powertrain was simulated in MAVS as an electric drive motor, whereas real MRZR-D4 vehicle has a diesel motor. In order to match the experimental results which will be presented later, the engine model in MAVS was modified by adjusting the requested throttle, τ_{in}, according to the equation

$$\tau = -6.01\tau_{in}^2 + 8.29\tau_{in} - 2.13 \tag{1}$$

The output value was clamped to a minimum value of zero. This reproduced the considerable “looseness” observed in the real vehicle whereby the throttle could be slightly engaged without causing the vehicle to move.

Camera and lidar are simulated in MAVS using the Embree ray-tracing kernel [Wald et al., 2014, Woop et al., 2017]. Camera output is simulated using path tracing [Jensen, 1995]. Camera radial distortions are simulated using a radial distortion model [Zhang, 1999]. Lidar simulations in MAVS use oversampled diverging beam pulses to account for effects like incident angle, reflectance properties, and mixed pixel effects [Goodin et al., 2018].

6 Test Scenarios and Metrics

As mentioned above, this project proceeded in three phases. In phase 1, the MAVS simulator was used to study how the performance of the perception subsystem correlated to system-level performance. In phase 2, the MAVS simulator was used to study how the performance of the path planning subsystem correlated to system-level performance. Finally, in phase 3, physical testing was used to both validate simulation results and to evaluate the performance of the real system in soft soil.

Although each phase featured obstacle avoidance in a straight-line driving scenario, there were minor differences between the test setup in each phase as the both the autonomy stack and the project capabilities changed over the duration of the three year study. In the following subsections, the unique aspects of each phase are presented. In addition, the metrics used in each phase are presented and discussed.

6.1 Phase 1: Perception

In phase 1, the test vehicle navigated a 90-meter-long test lane. At the center of the lane, 45 meters from the starting position, was a jersey barrier, 1-meter tall x 2 meters wide. The vehicle had to avoid the jersey barrier, return to the test lane, and reach the goal point that was 45 meters beyond the jersey barrier. The 90-meter test lane was preceded by a 100-meter lane in which the vehicle accelerated to reach the test speed of 10 m/s.

In phase 1, the primary focus was studying how extrinsic, environmentally induced error propagated through the subsystems of the AGV, as well as studying the effect of intrinsic sensor error on the perception subsystem and the propagation of this error through the other subsystems.

6.1.1 Phase 1 Error Sources

Experiments were conducted for the five error types shown in Table 2. Additionally, simulations with no injected errors were run for comparison. In total, 3000 simulations were run, totaling about 48 hours of simulated experiments. An example of output from simulations in clear, rainy, and dusty conditions are shown in Figure 2.

Figure 2: Environmental conditions tested in phase 1

Macfarlane and Stroila [Macfarlane and Stroila, 2016] identified three main classes of uncertainties in au-

Table 2: Types of error and values considered in the simulated experiments of phase 1

Variable	Range	Increment
Lidar Range Error	5 – 25 mm	5 mm
Camera Quality	0.75 – 1.75	0.20
RTK Pose Error	200-1400 mm	200 mm
Rain	4 – 28 mm/hour	4 mm/hour
Dust	1-8 units	1 unit

tonomous navigation: sensors, maps, and situations. Uncertainties in sensors include sensor noise and inaccuracy as well as environmentally induced errors caused by rain, dust, fog, or other phenomena. Map uncertainties include errors in object detection and localization. Finally, situational uncertainties pertain to predicting the future state of the dynamic environment. Since the environment in these experiments is static, situational uncertainties were not measured. Instead, the focus was on sensor and map errors.

Lidar Range Error Lidar sensors are subject to noise errors in measurement accuracy. A survey of reported lidar specifications shows that typical RMS range errors are between 1-5 cm [Halterman and Bruch, 2010, Glennie et al., 2016, Mittet et al., 2016]. In this work, error was added as Gaussian noise to the raw sensor signals to approximate range error for the lidar.

Stereo Camera Quality Factor The expected range error, Δz, in a stereo camera system can be estimated by

$$\Delta z = \frac{\Delta d z^2}{f b} \tag{2}$$

where z is the actual range, f is the focal length of the camera, b is the baseline between the cameras, and Δd is the size of a pixel. This means that the accuracy of a stereo system can be increased by reducing the size of the pixels. For a fixed imaging plane size, this equates to increasing the resolution. Conversely, the range error will increase if the pixel size is increased and the resolution is decreased. To assess the impact of this effect, a stereo quality factor, QF, was defined to scale the resolution of the image. The default resolution of the image when QF=1 was 640x480 pixels. For QF=0.5, the resolution was 320x240, for QF=2.0 the resolution was 1280x960, and so on. By adjusting this quality factor, the range error in the measured point cloud could also be adjusted.

Rain The influence of rain on lidar has been well documented [Rasshofer et al., 2011]. The primary effect of rain on lidar sensors is to reduce the range of the sensor and increase the range error. In the simulations reported in this work, the rain rate was varied from 0-28 mm/h, with 28 mm/h representing an unusually heavy rain.

Dust While it is well known that dust can obscure lidar targets, there has been little work on quantifying this effect for automotive lidar sensors. It has been shown that the optical depth of the dust cloud correlates well with the probability of the dust obscuring the lidar target [Goodin et al., 2013]. The primary error mode is for the lidar to return from the dust cloud itself, rather than the surfaces in the environment. Since the optical properties of the dust cloud are more important for lidar interaction than the mass properties, dust particles were added in nine increments from no dust to a totally opaque (optically thick) dust cloud. The dust cloud was added to the scene directly in front of the obstacle. This mimics the situation where dust could obscure a target like a vehicle or pedestrian.

RTK Noise RTK sensors are subject to noise errors in measurement accuracy. A survey of RTK lateral position errors indicates that error typically ranges from 0.5 to 3.0 meters [Mahmoud and Trilaksono, 2018]. Therefore, as with the lidar range noise, Gaussian noise is added to the raw lateral position to approximate error for the RTK.

6.1.2 Phase 1 Metrics

Information flows through the subsystems from perception to mapping to planning to control, and it is possible for error initiating in the sensing subsystem to manifest itself uniquely in the other subsystems. Therefore, metrics are proposed for each of these subsystems.

Time to complete The total amount of time from the time the vehicle starts moving until it reaches the goal point at the end of the course was recorded.

End-state If the vehicle reaches the goal point within 90 seconds, the vehicle successfully completes the trial and "Completion" is true, otherwise it is false. If the vehicle collided with the obstacle, it fails to complete the trial and "Collision" was true, and false otherwise. If the vehicle rolled over (flipped), the vehicle failed to complete the trial and "Rolled Over" was true, and false otherwise.

Point cloud error This quantifies the impact of environmental factors like rain and dust on the accuracy of the point cloud. Each time a scan is published in the simulation, the point cloud is compared to the "true" point cloud in the absence of rain or dust. The average point-wise error is computed by comparing the resulting clouds point by point.

Odometry error The average difference between the vehicle's actual position and its position as measured by the odometry at each time step.

Grid error Quantifies the propagation of error from the perception subsystem to the mapping subsystem. The "true" grid with perfect ground truth is compared to the grid created from the sensor data with errors. Error is quantified as the average difference in measured slope in each cell.

Path error Quantifies the propagation of error from the perception subsystem to the planning subsystem. The "ideal" path follows the center-line of the test lane. The path error metric quantifies the average deviation of the vehicle from this ideal path during the experiment.

6.2 Phase 2: Path-planning

In the second year, the metrics were the same as phase 1, but the test scenario was modified in two significant ways. First, the obstacle was changed to a box which varied in width from 0.1 m to 0.9 m. Second, barriers were placed along the sides of the test lane to fully bound the vehicle within a corridor. The corridor (Figure 3) consists of a 90 m lane with the obstacle placed at 45 m. Barriers were placed on the sides of a 10 m lane to constrain the space available for the vehicle to maneuver. The vehicle was required to navigate through a gap 4-5 m wide.

Multiple path planning algorithms were implemented, with the final three that were selected for testing including A*, vector field histogram, and the spline planner, as discussed in Section 4.2.2.

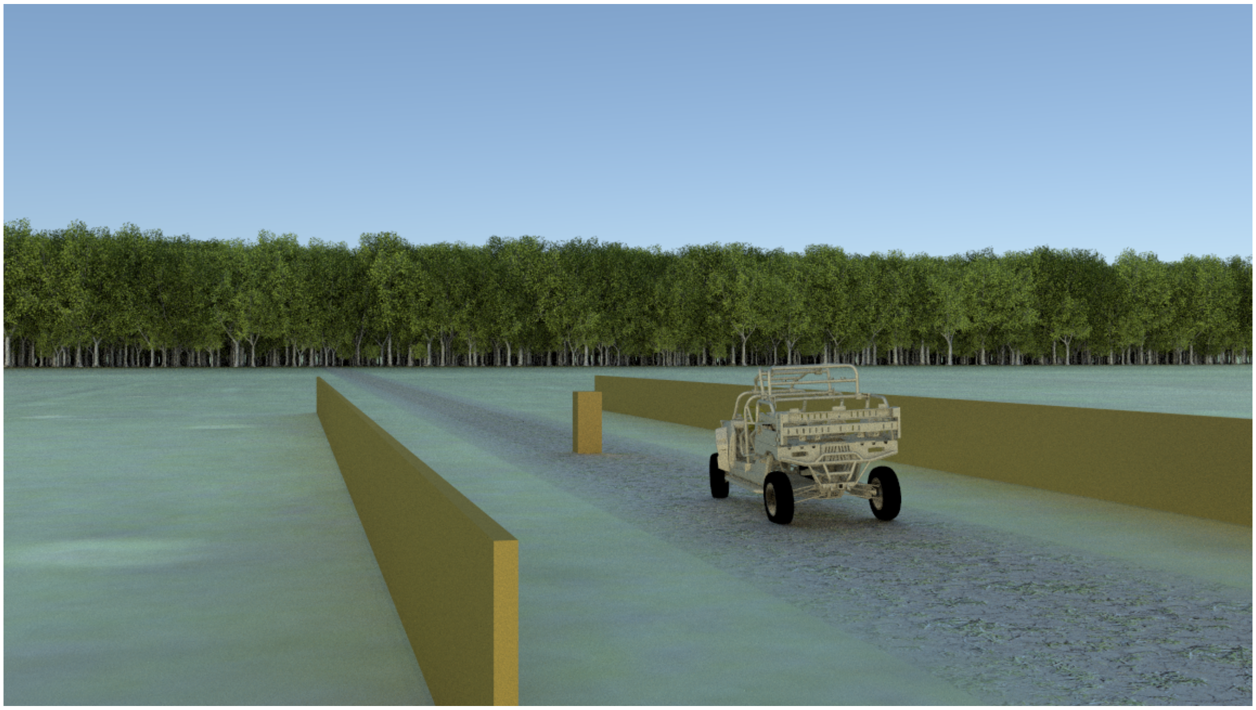

Figure 3: Phase 2 scenario with box obstacle and barriers on the side

6.2.1 Phase 2 Environmental Conditions

Soil Conditions Tests were run in two basic soil conditions: firm and soft. For the soft condition, the RCI varied from 28-38. In the firm condition, the surface approximated a paved asphalt surface. In the soft condition, the surface approximated a clay surface. In all conditions, the test lane was flat.

Objects The scene included a single box-shaped obstacle (1m tall x $0.1 - 0.9$m wide) placed in the center of the test lane. The scene also included two lines of 1 meter tall barriers placed on either side of the test lane creating a solid boundary.

In addition to variations in the environment, the desired speed of the vehicle also varied from 3 to 13 m/s. Trials were completed for a full factorial combination of inputs: three algorithms, six speeds, two surface types, and 9 obstacle sizes. Trials were repeated 100 times with small variations in initial position and orientation for a total of 32,400 simulations.

6.3 Phase 3: Validation and Soft-soil Testing

The final phase of this study had three main purposes. The first was to implement the NATURE stack, which had been developed exclusively in simulation in phases 1 and 2, on a real vehicle and verify that the stack functioned as expected. The second was to perform a limited validation of the simulated results from phases 1 and 2. The validation was limited because it was not possible to reproduce the rain and dust tests in real world experimentation. The third goal was to conduct soft-soil testing of the autonomy stack and compare the results to simulation.

While the scenario and metrics were essentially the same as phase 2, the scenario layout was modified slightly to accommodate physical testing. The test lanes were 150 meters long, including a 50 meter long lane that was used for the vehicle to reach the desired test speed. From the 110 to 140 meter length of the test lane, barriers were constructed from black silt fencing, which was approximately 0.91 meters tall. The barriers were placed 10 meters apart. A typical experiment is shown in Figure 4.

Figure 4: An example phase 3 field experiment

A cardboard box was placed directly in the desired path of the vehicle, at the 95 meter mark of the test lane, so that the vehicle had to successfully detect and avoid the box while avoiding collisions with the the surrounding barriers. The box was rectangular, 0.45 meters on the sides and 0.91 meters tall.

The hard-surface test area consisted of a tightly packed gravel road with grass on the sides. For the soft-soil testing, the simulations of phase 2 featured very soft soil (28-38 CI). However, the physical area that was available for testing had a much wider range of soil strengths over the length of the course (40-160 CI) but had an average strength of 140 CI in the top 6 inch layer, with a standard deviation of $\pm$66 CI. Additionally, the soft soil test area was a rice field that had been recently tilled to a depth of one foot, meaning the top layer of soil was quite loose and uncompacted prior to testing.

7 Results

In this section, the results from all three phases are presented. Phase 1 and 2 results are presented briefly (readers may consult previous publications for more details [Carruth et al., 2020]), while the results of phase 3 are presented in more detail.

7.1 Phase 1: Perception Results

The primary measure of system-level performance for the AGV is whether it successfully reached its objective and avoided a collision with the barrier and did not rollover. All tests were performed with a goal speed of 10 m/s. While a detailed description of the results can be found in [Carruth et al., 2020], a brief overview of the results of phase 1 are presented here for context.

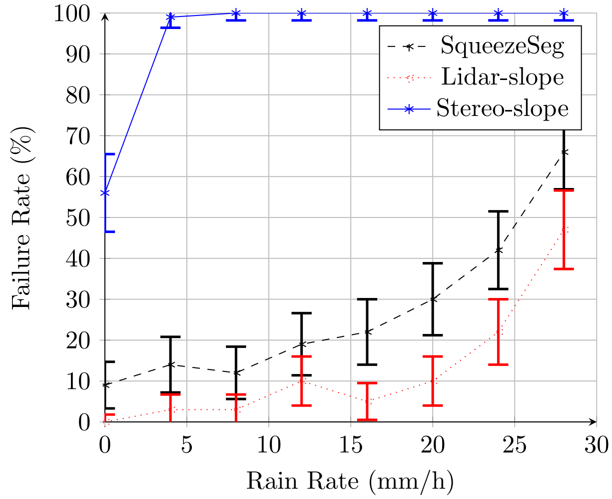

Figure 5: Phase 1 failure rates versus rain rate three perception algorithms

Three perception algorithms were tested. Each was tested in "ideal" conditions, as well as in varying amounts of rain or dust obscuring the obstacle. While the overall failure rate was fairly low (6.5% of trials), there were significant differences observed between the different perception algorithms. Figure 5 shows an example of this for rain. It is clear that while all systems performed worse as the rain rate increased, the stereo-camera perception algorithm was the most drastically affected. In addition, the lidar-based slope algorithm performed better than the machine-learning based algorithm.

In general, the failure rate was less than 1% in the "ideal" trials and overall was better for the lidar-based systems. The primary deficiency of the stereo system was significant range error in the point cloud with increasing range to target, leading to high observed failure rates for stereo system.

Concerning the relationship of subsystem performance to system-level performance, a small, limited relationship between point cloud error and occupancy grid error was observed. In contrast to the weak relationship between point cloud error and occupancy grid error, error in the odometry had a clear correlation with the observed occupancy grid error. However, the overall conclusion of phase 1 was that the subsystem-level metrics were, overall, not strong predictors of system-level performance.

7.2 Phase 2: Path Planning Results

In phase 2, the simulation scenarios were updated to incorporate the surface types, obstacles, desired speed conditions, and the modified autonomy stack, as discussed in Section 6.2. The slope-based lidar perception algorithm was used, along with three planners - the spline planner, A*, and vector field-histogram.

While all tests were performed at 10 m/s in phase 1, for phase 2 it was hypothesized that the vehicle would perform more effectively at lower speeds. Therefore, six speeds were tested for each path planning algorithm - 3, 5, 7, 9, 11, and 13 m/s. Additionally, obstacles of width 0.1-0.9 were tested, in 0.1 meter increments. Finally, two different simulated surface types were tested - dry pavement and soft soil with a cone index between 28-38 pounds/in^2. All combinations of test parameters (speed, obstacle size, surface type) were tested with 100 trials for each combination for a total of 32,400 simulated experiments. The initial start position of the vehicle was varied by ± 1 meter and the orientation varied by $\pm 0.1^o$ to introduce non-determinism into the simulations.

The effective speed - the distance of the course divided by the total time to completion was measured for each experiment, as well as the failure rate. The objective (desired) speed did influence overall mission success, but there was no significant difference in effective speed by planner or by obstacle. There was however, a significant difference in the planners in failure rates. A* and VFH were far more likely to hit the barrier at the lower speeds. This seems counter-intuitive but, particularly for A*, the algorithm had an issue at low speeds with the obstacle being in the center of the road. Because of the symmetry of the configuration, A* and VFH would wait until the last possible moment to commit to a left or right path, sometimes resulting in a collision with the obstacle. In contrast, the spline planner formulation imposed a penalty for the path flipping from left to right, so it tended to pick a path and stick to it earlier in the test. This effect is clear in Figure 6, which shows the failure rate versus speed for the three planners for the 0.9 meter obstacle.

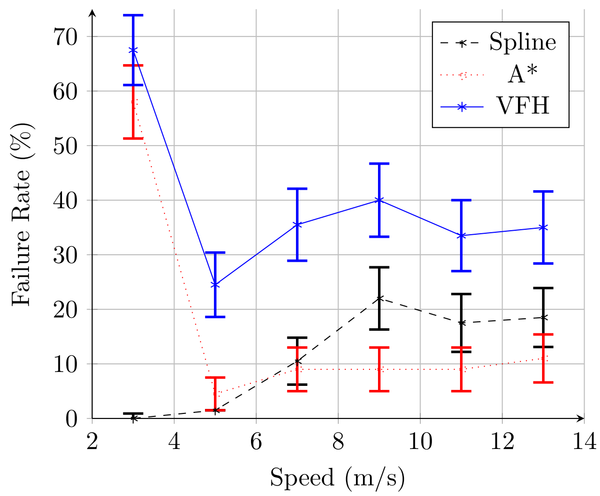

Figure 6: Phase 2 failure rates versus speed for the 0.9 meter obstacle

Each test used the same perception algorithm – the lidar-based slope algorithm that fed into the occupancy grid. Despite using the same perception algorithm, the obstacle size had different effects on the three planning algorithms. In the spline planner, the smaller or larger the obstacle was, the more likely a collision. For the A* algorithm, performance was lower for obstacles >0.6m but there was a particular issue with 0.7 m obstacles. These results are shown in Figure 7 with 95% confidence intervals.

For VFH, performance was reduced for obstacles larger than 0.6 meters. Two potential factors may cause this observed trend. First, the algorithms take different approaches to planning the path. It is possible that the path that they plan is too close to obstacles of certain sizes. Second, despite using the same perception algorithm, the different paths take them on different routes with different perspectives on the environment that may lead to different errors in the occupancy grid.

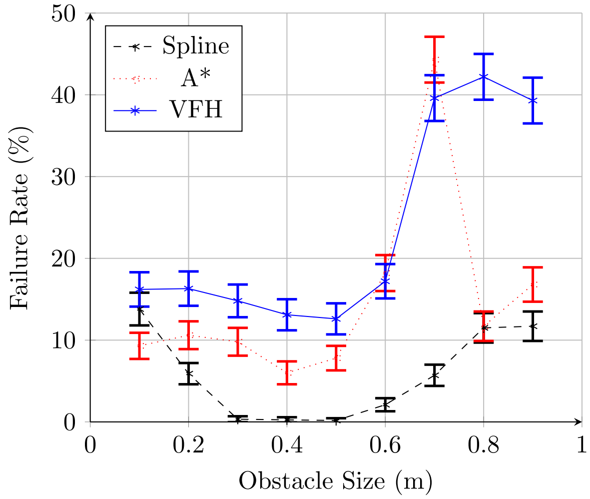

Figure 7: Phase 2 cumulative failure rates versus obstacle size for all speeds.

Finally, no effect of soil strength was observed on failure rate or effective speed. Calculations showed that the expected VCI_1 of the vehicle was ≈15 RCI, so the simulated soft soil with RCI=28-38 should be soft enough to impede the vehicle but not immobilize it. However, the speed controller may compensate for extra resistance by simply applying more throttle. In this case, additional fuel consumption would be observed, even if the speed and failure rate did not change. Although the throttle was not recorded in the simulated experiments, this hypothesis was tested in the physical experiments and simulations of phase 3 by recording the throttle in those experiments.

7.3 Phase 3: Physical Testing and Soft Soil Results

In phase 3, the autonomy stack that was developed in phases 1 and 2 was implemented on a real vehicle, the MRZR-D4, shown in Figure 4. The goal of phase 3 was to run obstacle avoidance tests at 5, 7, 9, and 11 m/s on both a hard surface (gravel road) and on soft soil. In order to compile meaningful statistics, 50 runs for each speed and soil condition were planned, for a total of 400 planned physical experiments. Additionally, the physical experiments were recreated in simulation for the purpose of validation.

Three primary system-level measurements were considered for comparison between the real vehicle and simulated one. These were 1) the effective vehicle speed, 2) the trajectory of the vehicle around the obstacle, and 3) the failure rate of the vehicle. In the following sections, the results of the both the real and simulated experiments for the hard and soft surfaces are presented.

7.3.1 Hard-surface testing

The hard surface testing was conducted on a long, straight gravel road, as shown in Figure 4. The figure also shows the steep ditches on either side of the road. In the experiments, the operator used the emergency stop (e-stop) to end the experiment if the vehicle left the road on a trajectory toward the ditches. This failure mode was recorded as “out of bounds”. Of note is that the localization module of the physical vehicle relied on two offset GPS units on opposite corners of the vehicle to derive a heading. The start-up procedure of these units at the beginning of each experiment occasionally resulted in incorrect heading calculations, and this seemed to be the primary contributor to “out of bounds” failures.

The operator would also e-stop the vehicle if it collided with the obstacle. This failure mode was recorded as “collision”. In some cases, the safety operator engaged the e-stop prior to an imminent collision. This failure mode was still recorded as a collision. While it was possible for the vehicle to collide with the physical boundaries, all the observed collisions were with the obstacle in the center of the road.

A final failure mode, which was only observed once in the 200 hard-surface trials, was a “time-out”. In this case, the vehicle simply came to a stop and would not drive any further. In total, there was one failure observed at 5 m/s (collision), four failures at 7 m/s (3 out of bounds, one time-out), two failures at 9 m/s (one collision, one out of bounds), and at 5 failures at 11 m/s (two collsions, one out of bounds).

Figures 8-11 show the real and simulated trajectories for the on road experiments and simulations, separated by speed for the 5, 7, 9, and 11 m/s tests, respectively. In these figures, the black lines are measured trajectories from the physical experiments, the cyan lines are measured trajectories from the physical experiments, the blue lines represent the barriers, the red dot represents the obstacle, and the green ellipse represents the goal region. The red line is the average experimental trajectory, while the magenta line is the average simulated trajectory. Note that the scale of the x and y axes are quite different in these figures.

Figure 8 shows the results for the 5 m/s tests. The first obvious feature is the strong preference for the vehicle to go left, with only two of the fifty experimental runs avoiding the obstacle to the right. The most likely cause of this effect is a slight misalignment of the two GPS sensors used to calculate the heading. With this assumption implemented in the MAVS, a similar trend was observed in the simulations, with only two of fifty simulations going right. Additionally, the overall character of the real and simulated trajectories is similar, with both the real and simulated average trajectory having a “knee” near the obstacle before bending back to the center line. However, the simulated vehicle did not veer as far left, on average, as the real vehicle, a trend which was observed at all speeds.

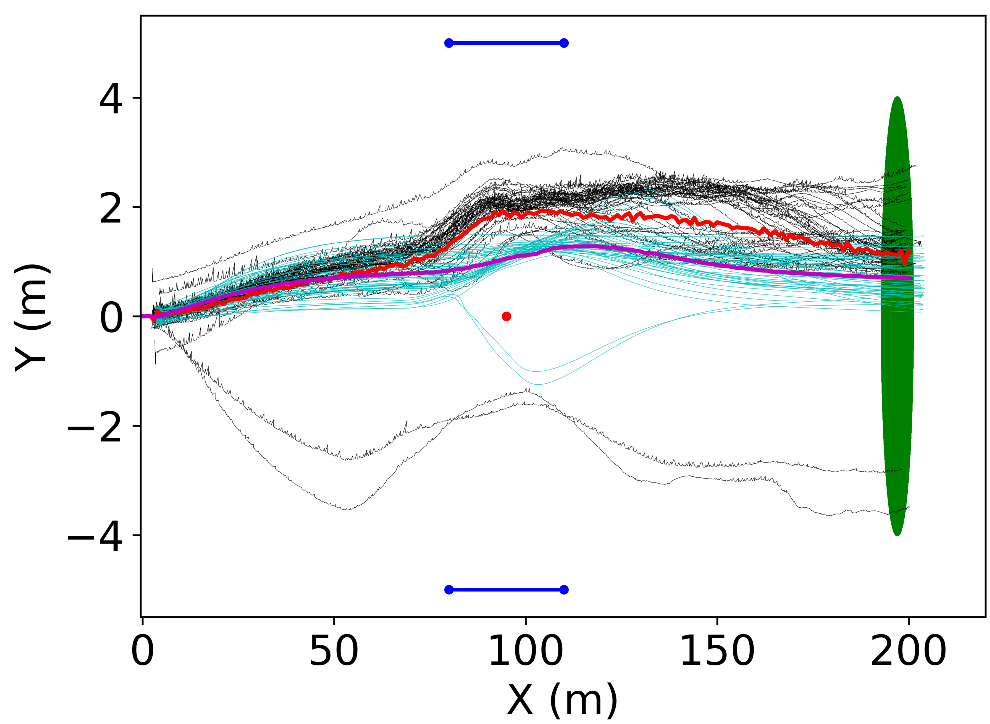

Figure 8: Real and simulated hard surface trajectories, 5 m/s

Figure 9 shows the trajectory plots for the 7 m/s tests. Several failed experiments stand out in this test, with the vehicle trajectories leaving the course. This trend was not reproduced in the simulated experiments. The failed experiments were due to infrequent errors in the localization system, which due to their unpredictable nature, were difficult to reproduce in the simulation. Otherwise, the same trend regarding average trajectory is true for 7 m/s as for 5 m/s - namely, the trajectories have the same basic shape, but the experimental runs tended to veer further left than the simulated ones.

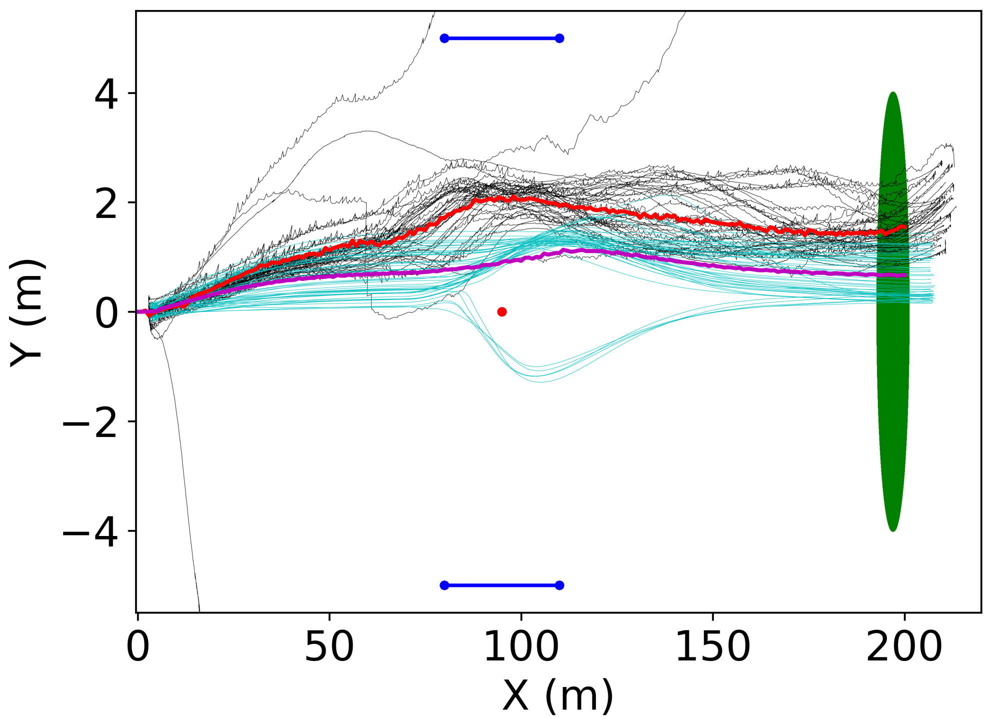

Figure 9: Real and simulated hard surface trajectories, 7 m/s

Similar patterns are observed for the 9 m/s and 11 m/s tests shown in Figures 10 and 11, although there are two anomalous cases in the simulated data for 9 m/s. In both these cases, the vehicle entered a reactionary over-steering condition. This condition did not result in a failure, but was not observed in physical tests. The difference is likely due to the fact that the simulated vehicle has the capability to actuate the steering somewhat faster than the real vehicle. Potential over-steers were therefore damped out in the real test vehicle, but could manifest in the simulation, as shown in Figure 10.

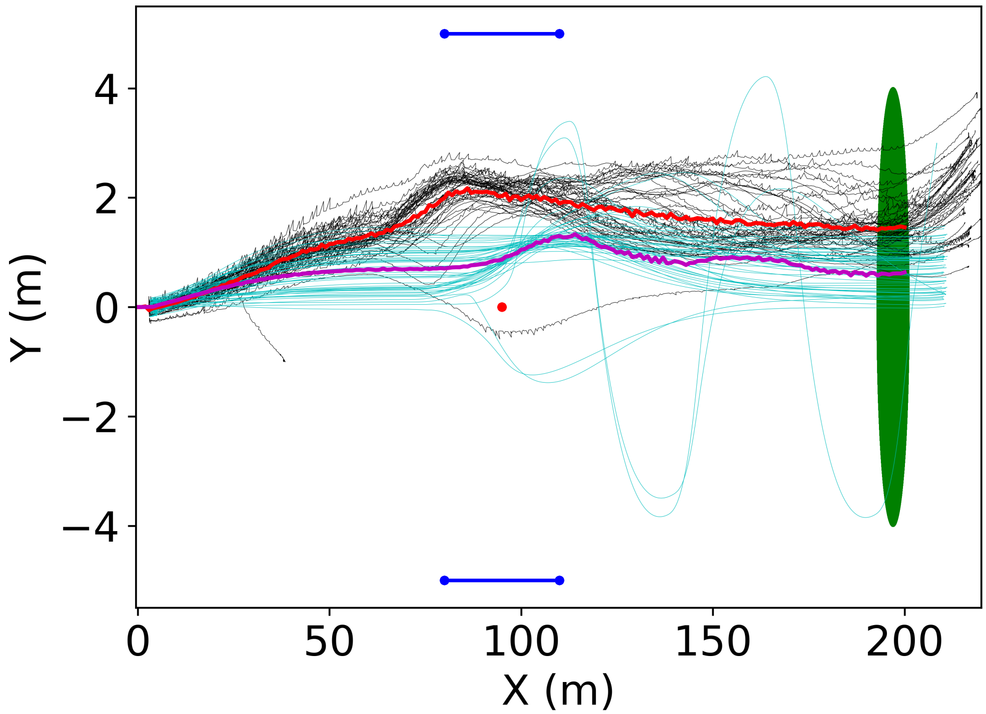

Figure 10: Real and simulated hard surface trajectories, 9 m/s

Both the 9 m/s and 11 m/s tests also show experiments where the vehicle left the course. Again, this was due to localization errors which occurred on start-up of the localization hardware and were difficult to diagnose or reproduce.

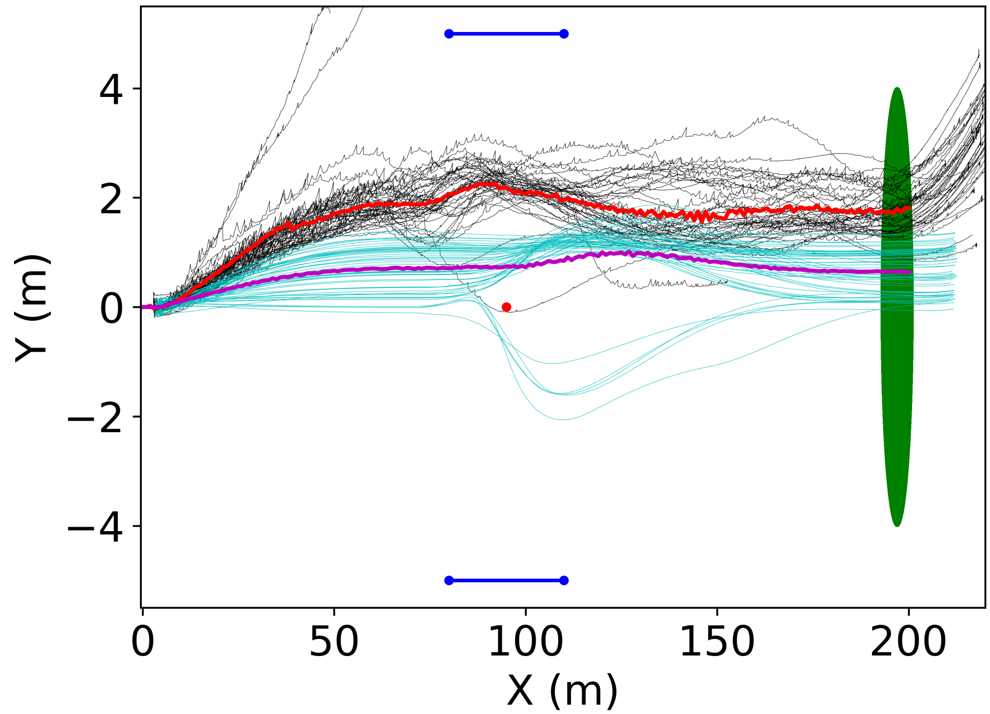

Figure 11: Real and simulated hard surface trajectories, 11 m/s

Figure 12 shows the results of the average speed versus the desired set speed for all experiments. The dashed black line shows the measured average speed over all successful hard-surface experimental runs, while the solid magenta line shows the average effective speed over all successful hard-surface simulations. It is clear that the predicted speeds from the MAVS match the experimentation quite well. The green and blue lines in Figure 12 are for the soft soil measurements and simulations, which will be discussed in the next section.

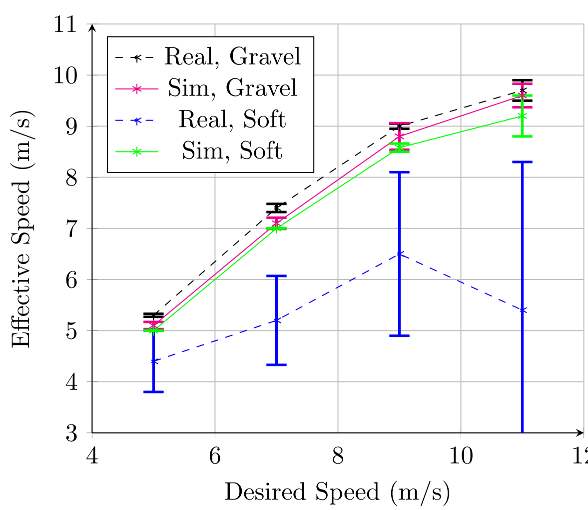

Figure 12: Average effective speed for real and simulated tests, versus desired speed

The final system-level measurement was failure rate. As mentioned above, the majority of failures in the physical experiments were due to errors in the localization subsystem, whereas no failures were observed in the simulated testing.

Figure 13 shows the measured failure rates for the real and simulated tests versus speed for the hard-surface testing. Error bars represent 90% confidence intervals, calculated using binomial statistics for 50 trials. From this figure it is clear that the simulations likely under-predicted the failure rate, but this can only be stated with at least 90% confidence for the 11 m/s trials. This uncertainty highlights the need for larger numbers of tests when evaluating relatively low failure rate events, as the 90% confidence intervals are still fairly large in our case with 50 trials at each speed.

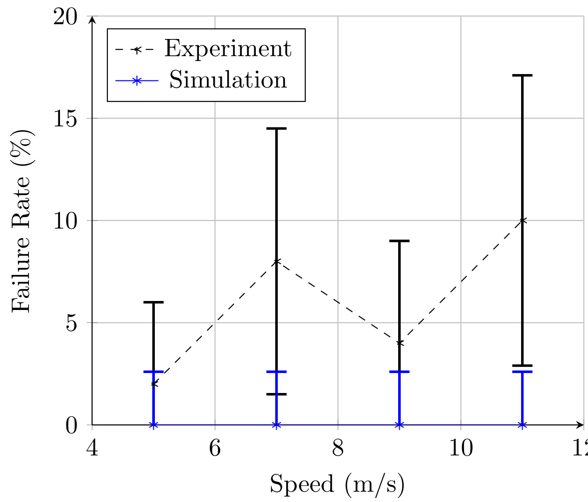

Figure 13: Hard surface failure rates with 90% confidence intervals, real and simulated

7.4 Soft-soil testing

The soft soil test area is shown in Figure 14. The course was shorter than the hard-surface test area (150 meters versus 200 meters) because of limited space in the soft soil area. The soil conditions during the time of testing are described in Section 6.3.

Figure 14: Deep ruts in the soft soil test area

Figure 14 shows the soft-soil test area after an experiment. Noting that there had recently been significant rain at the time of testing, the presence of deep ruts can be seen in the figure. The observed rut depth was 3-6 inches. The presence of the ruts had a significant impact on the results of the soft soil testing because they were so deep that the vehicle could not easily steer out of them. This resulted in the vehicle following the same path over and over, which can easily be seen by the tighter grouping of trajectories in the soft soil tests in Figure 15.

Overall, the character of the trajectories from the real and simulated vehicles (Figure 15) was similar, although there was more spread in the final state of the simulated vehicles. This is most likely due to the rutting constraining the steering of the real vehicle, as mentioned above. However, even the simulated trajectories in the soft soil have a tighter grouping than the simulated trajectories on the hard surface, indicating that the MAVS simulated lateral traction model effectively differentiates between hard and soft soil.

No failures were observed in any of the soft soil simulations or experiments.

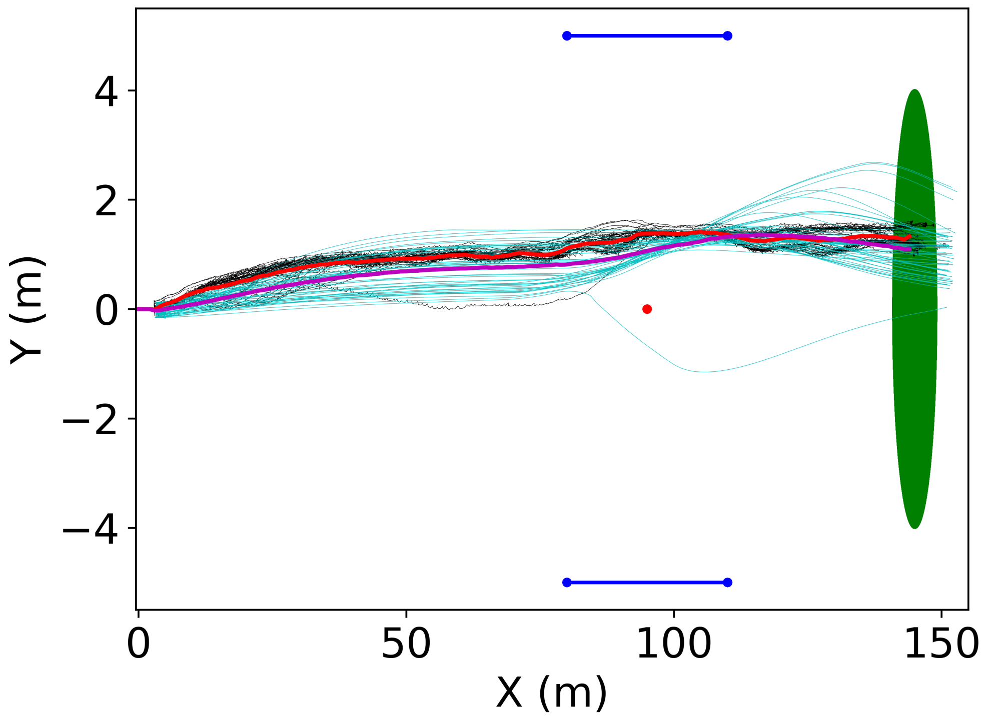

Figure 15: Soft soil trajectories, 5 m/s

While all fifty soft-soil experiments were completed for the 5 m/s speed setting, only two experiments were completed at 7 m/s, and only one each at 9 and 11 m/s. As the season transitioned to dry summer conditions at the test site in Starkville, Mississippi, the test area quickly dried out and the soft soil condition was lost, making it impossible to complete the full compliment of tests. However, as shown in Figure 12, even with only a few data points for the higher speeds, a significant reduction in actual speed was observed for the soft soil testing. This was primarily due to the controller being unable to compensate for the extra resistance of the soft soil.

Figure 12 shows the measured average speeds in soft soil (dashed blue line) and the simulated speeds (solid green line) with one standard deviation error bars. The experiment at 11 m/s gave very inconsistent results and the vehicle oscillated between extreme acceleration and coming to a complete stop, resulting in high uncertainty in the average speed. Even so, it is clear that the simulation overestimates the achievable speeds in the soft soil. This may attributable to several factors or limitations in the simulation model.

First, the presence of the deep ruts during experimentation was not accounted for in the simulation. This is the most likely contributing factor to the deviation of experiment and simulation, and one that can be remedied in future work. Second, the real test vehicle accumulated a significant amount of mud on the front, sides, and undercarriage of the vehicle - estimated to be up to 45 kg (100 pounds), potentially significantly altering the dynamics of the vehicle. Finally, the soft soil model used in MAVS is based on the equations from the DROVE database [Williams et al., 2019], which are for *in situ*, undisturbed fine-grained s oil. Obviously, this condition was violated in the physical testing, which took place in a recently-plowed field. This condition likely significantly altered the tire-soil interaction, resulting in simulated predictions that did not match the experimental conditions.

Another interesting difference between the soft and hard surface experiments was the effect on throttle and steering effort of the vehicle. Figure 16 shows the overall steering effort for the vehicle versus the overall throttle effort. These numbers represent the cumulative throttle and steering effort of the vehicle over the entire test, divided by the duration of the test in seconds. The units are arbitrary - the throttle and steering scales are unique to the drive-by-wire kit on the test vehicle. Nevertheless, the units are the same between the two sets of experiments, facilitating comparison between the two.

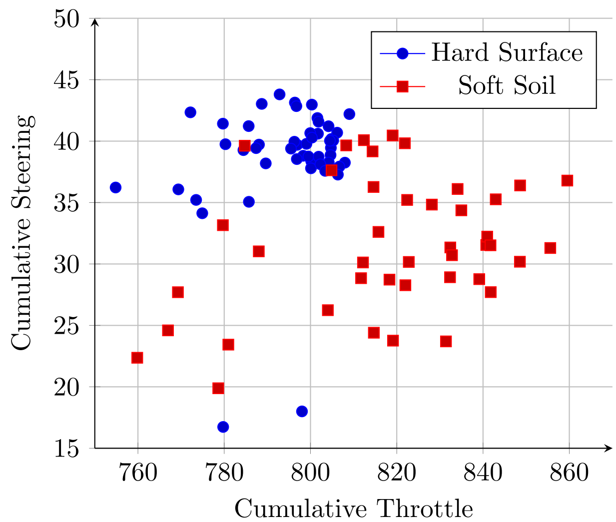

Figure 16: Steering effort vs throttle effort (arbitrary units) for soft and hard surface physical tests

As can clearly be seen in Figure 16, the overall throttle effort was higher in the soft soil, as expected. This means that more throttle was required in an attempt to reach the desired speed, even though the vehicle was unable to reach the desired speed in the soft soil in most cases. However, the steering effort is actually less in the soft soil condition. This lower steering effort is most likely due to the previously mentioned influence of the deep ruts during the testing. The vehicle had a tendency to drive in the same ruts over and over, and was unable to turn out of those ruts. In some sense, the ruts acted as a guide for the wheels, forcing them down the same path over and over while requiring less steering effort to achieve the desired lateral motion from the vehicle controller.

8 Summary and Conclusions

This work presented the results of a multi-year study on the impact of environmental factors on obstacle detection and avoidance by an off-road AGV. During the project, a free and open source autonomy stack for off-road navigation was developed. Using this stack, a testing method for the quantitative assessment of error propagation through the AGV subsystems was demonstrated in simulation. Finally, this testing method was validated through the use of field experiments with the M RZR-D4. This study is the most thorough and comprehensive of its kind to be made with a publicly available AGV software stack.

The first two years consisted entirely of simulated experiments. Year 1 focused on the performance of the perception subsystem in the presence of rain and dust, while year 2 focused on the performance of the planning subsystem as the obstacle size and surface condition were changed. The primary conclusions of the first two years were 1) lidar-based perception performed better in rain than stereo-camera based, 2) soft soil may result in increased fuel usage, and, most importantly, 3) subsystem-level error did not correlate well with system level failures.

In the third and final year the autonomy stack developed in years 1 and 2 was implemented on a real vehicle and physical testing was conducted for ODOA on both hard surfaces and soft soil. Simulations were conducted that matched the conditions of the physical tests. The were several important observations in the final phase. Experimentally, there was a slight increase in failure rate with increasing speed. Soft-soil caused increased fuel use and lower speeds, but not more failures.

The MAVS simulation matched the real tests well in some respects but not in others. Simulated trajectories matched the real trajectories fairly well, especially in soft soil. Simulated speeds matched very well for the hard surface, but not for soft soil, where MAVS over-predicted the speeds. The failure rates matched within 90% confidence intervals. One of the primary conclusions for measuring failure rates was that even 50 trials is not enough to precisely measure failure rates of <5%.

The results of this multi-year study suggest several potential avenues of future research. Potential upgrades to MAVS include simulating the presence of persistent rutting in soft soil and developing a simulated odometry model that accounts for the infrequent but severe failures in the localization system. Additionally, upgrades to the autonomy stack, which is available at `https://github.com/CGoodin/nature-stack`, may include more sophisticated throttle control which can detect wheel slip and adjust throttle settings in soft soil, as well as a more robust localization system that avoids catastrophic failures.

In summary, this article documented the results of a multi-year effort to measure the influence of adverse environmental conditions on AGV performance in ODOA scenarios, developing combined simulation and physical testing methods that demonstrated the value of this testing approach for future research on the performance of AGV.

Acknowledgments

The authors wish to acknowledge the financial support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-14-2-0001 U.S.

Army Tank Automotive Research, Development and Engineering Center (TARDEC) Warren, MI. DISTRIBUTION A. Approved for public release; Distribution unlimited. OPSEC# (Pending, NOT yet approved for release)

References

[Adams et al., 2007] Adams, M., Wijesoma, W. S., and Shacklock, A. (2007). Autonomous navigation: Achievements in complex environments. *IEEE Instrumentation & Measurement Magazine*, 10(3):15–21.

[Al-Mayyahi et al., 2015] Al-Mayyahi, A., Wang, W., and Birch, P. (2015). Path tracking of autonomous ground vehicle based on fractional order pid controller optimized by pso. In *2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 109–114. IEEE.

[Amer et al., 2018] Amer, N. H., Hudha, K., Zamzuri, H., Aparow, V. R., Abidin, A. F. Z., Abd Kadir, Z., and Murrad, M. (2018). Adaptive modified stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle. *Robotics and Autonomous Systems*, 105:94–111.

[Barraquand et al., 1992] Barraquand, J., Langlois, B., and Latombe, J.-C. (1992). Numerical potential field techniques for robot path planning. *IEEE transactions on systems, man, and cybernetics*, 22(2):224–241.

[Borenstein et al., 1991] Borenstein, J., Koren, Y., et al. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE transactions on robotics and automation*, 7(3):278–288.

[Borja et al., 2009] Borja, C. A., Tur, J. M. M., and Gordillo, J. L. (2009). State your position. *IEEE Robotics & Automation Magazine*, 16(2):82–90.

[Bostelman et al., 2016] Bostelman, R., Messina, E., et al. (2016). Towards development of an automated guided vehicle intelligence level performance standard. *Autonomous Industrial Vehicles: From the Laboratory to the Factory Floor*, pages 1–22.

[Browning et al., 2012] Browning, B., Deschaud, J.-E., Prasser, D., and Rander, P. (2012). 3d mapping for high-fidelity unmanned ground vehicle lidar simulation. *The International Journal of Robotics Research*, 31(12):1349–1376.

[Campbell et al., 2020] Campbell, S., O'Mahony, N., Carvalho, A., Krpalkova, L., Riordan, D., and Walsh, J. (2020). Path planning techniques for mobile robots a review. In *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pages 12–16. IEEE.

[Carruth et al., 2020] Carruth, D., Goodin, C., Dabbiru, L., Scherer, N., and Jayakumar, P. (2020). Predicting error propagation in autonomous ground vehicle subsystems. In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium*, pages 11–13.

[Carruth, 2018] Carruth, D. W. (2018). Simulation for training and testing intelligent systems. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 101–106. IEEE.

[Chen et al., 1999] Chen, J. X., Fu, X., and Wegman, J. (1999). Real-time simulation of dust behavior generated by a fast traveling vehicle. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 9(2):81–104.

[Coulter, 1992] Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST.

[Dabbiru et al., 2020] Dabbiru, L., Goodin, C., Scherrer, N., and Carruth, D. (2020). Lidar data segmentation in off-road environment using convolutional neural networks (cnn). *SAE International Journal of Advances and Current Practices in Mobility*, 2(2020-01-0696):3288–3292.

[Davis, 1975] Davis, D. C. (1975). A radial-spring terrain-enveloping tire model. *Vehicle System Dynamics*, 4(1):55–69.

[Dosovitskiy et al., 2017] Dosovitskiy, A., Ros, G., Codevilla, F., López, A., and Koltun, V. (2017). Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*.

[Durst and Goodin, 2012] Durst, P. J. and Goodin, C. (2012). High fidelity modelling and simulation of inertial sensors commonly used by autonomous mobile robots. *World Journal of Modelling and Simulation*, 8(3):172–184.

[Durst et al., 2017] Durst, P. J., Goodin, C. T., Anderson, D. T., and Bethel, C. L. (2017). A reference autonomous mobility model. In *2017 Winter Simulation Conference (WSC)*, pages 4026–4037. IEEE.

[Durst and Gray, 2014] Durst, P. J. and Gray, W. (2014). Levels of autonomy and autonomous system performance assessment for intelligent unmanned systems. Technical report, ENGINEER RESEARCH AND DEVELOPMENT CENTER VICKSBURG MS GEOTECHNICAL AND ....

[Fadaie, 2019] Fadaie, J. (2019). The state of modeling, simulation, and data utilization within industry: An autonomous vehicles perspective. *arXiv preprint arXiv:1910.06075*.

[Falco and Gilpin, 2021] Falco, G. and Gilpin, L. H. (2021). A stress testing framework for autonomous system verification and validation (v&v). In *2021 IEEE International Conference on Autonomous Systems (ICAS)*, pages 1–5. IEEE.

[Feng et al., 2021] Feng, S., Yan, X., Sun, H., Feng, Y., and Liu, H. X. (2021). Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):1–14.

[Glennie et al., 2016] Glennie, C., Kusari, A., and Facchin, A. (2016). Calibration and stability analysis of the vlp-16 laser scanner. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 9.

[Goodin et al., 2017] Goodin, C., Carrillo, J. T., McInnis, D. P., Cummins, C. L., Durst, P. J., Gates, B. Q., and Newell, B. S. (2017). Unmanned ground vehicle simulation with the virtual autonomous navigation environment. In *Military Technologies (ICMT), 2017 International Conference on*, pages 160–165. IEEE.

[Goodin et al., 2019] Goodin, C., Carruth, D., Doude, M., and Hudson, C. (2019). Predicting the influence of rain on lidar in adas. *Electronics*, 8(1):89.

[Goodin et al., 2018] Goodin, C., Doude, M., Hudson, C., and Carruth, D. (2018). Enabling off-road autonomous navigation-simulation of lidar in dense vegetation. *Electronics*, 7(9):154.

[Goodin et al., 2013] Goodin, C., Durst, P. J., Prevost, Z. T., and Compton, P. J. (2013). A probabilistic model for simulating the effect of airborne dust on ground-based lidar. In *Active and Passive Signatures IV*, volume 8734, page 87340D. International Society for Optics and Photonics.

[Gregory et al., 2021] Gregory, J. M., Warnell, G., Fink, J., and Gupta, S. K. (2021). Improving trajectory tracking accuracy for faster and safer autonomous navigation of ground vehicles in off-road settings. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 204–209. IEEE.

[Grigorescu et al., 2020] Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386.

[Halterman and Bruch, 2010] Halterman, R. and Bruch, M. (2010). Velodyne hdl-64e lidar for unmanned surface vehicle obstacle detection. In *Unmanned Systems Technology XII*, volume 7692, page 76920D. International Society for Optics and Photonics.

[Hart et al., 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

[Hu et al., 2020] Hu, J.-w., Zheng, B.-y., Wang, C., Zhao, C.-h., Hou, X.-l., Pan, Q., and Xu, Z. (2020). A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments. *Frontiers of Information Technology & Electronic Engineering*, 21(5):675–692.

[Hu et al., 2018] Hu, X., Chen, L., Tang, B., Cao, D., and He, H. (2018). Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mechanical Systems and Signal Processing*, 100:482–500.

[Huang et al., 2016] Huang, W., Wang, K., Lv, Y., and Zhu, F. (2016). Autonomous vehicles testing methods review. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 163–168. IEEE.

[Hudson et al., 2020] Hudson, C., Goodin, C., Miller, Z., Wheeler, W., and Carruth, D. (2020). Mississippi state university autonomous vehicle simulation library. In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium*, pages 11–13.

[Hudson et al., 2018] Hudson, C. R., Goodin, C., Doude, M., and Carruth, D. W. (2018). Analysis of dual lidar placement for off-road autonomy using mavs. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 137–142. IEEE.

[Islam et al., 2021] Islam, F., Nabi, M., Farhad, M. M., Peranich, P., Ball, J. E., and Goodin, C. (2021). Evaluating performance of extended kalman filter based adaptive cruise control using pid controller. In *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2021*, volume 11748, page 1174807. International Society for Optics and Photonics.

[Jensen, 1995] Jensen, H. W. (1995). Importance driven path tracing using the photon map. In *Eurographics Workshop on Rendering Techniques*, pages 326–335. Springer.

[Jha et al., 2018] Jha, S., Banerjee, S. S., Cyriac, J., Kalbarczyk, Z. T., and Iyer, R. K. (2018). Avfi: Fault injection for autonomous vehicles. In *2018 48th annual ieee/ifip international conference on dependable systems and networks workshops (dsn-w)*, pages 55–56. IEEE.

[Jha et al., 2019] Jha, S., Tsai, T., Hari, S., Sullivan, M., Kalbarczyk, Z., Keckler, S. W., and Iyer, R. K. (2019). Kayotee: A fault injection-based system to assess the safety and reliability of autonomous vehicles to faults and errors. *arXiv preprint arXiv:1907.01024*.

[Kalra and Paddock, 2016] Kalra, N. and Paddock, S. M. (2016). How many miles of driving would it take to demonstrate autonomous vehicle reliability. *RAND Corporation, Santa Monica, CA, Tech. Rep*, pages 1129–1134.

[Karis and Games, 2013] Karis, B. and Games, E. (2013). Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1.

[Kelly et al., 2006] Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., Happold, M., Herman, H., Mandelbaum, R., Pilarski, T., et al. (2006). Toward reliable off road autonomous vehicles operating in challenging environments. *The International Journal of Robotics Research*, 25(5-6):449–483.

[Koon and Whittaker, 2006] Koon, P. and Whittaker, W. (2006). Evaluation of autonomous ground vehicle skills. *Tech. report CMU-RI-TR-06-13*.

[Larson and Trivedi, 2011] Larson, J. and Trivedi, M. (2011). Lidar based off-road negative obstacle detection and analysis. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 192–197. IEEE.

[Letherwood and Jayakumar, 2021] Letherwood, M. and Jayakumar, P. (2021). Mobility assessment methods and tools for autonomous military ground systems. Technical Report TM-AVT-ET-194, North Atlantic Treaty Organization.

[Li et al., 2017] Li, G., Hari, S. K. S., Sullivan, M., Tsai, T., Pattabiraman, K., Emer, J., and Keckler, S. W. (2017). Understanding error propagation in deep learning neural network (dnn) accelerators and applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12.

[Li et al., 2016] Li, L., Huang, W.-L., Liu, Y., Zheng, N.-N., and Wang, F.-Y. (2016). Intelligence testing for autonomous vehicles: A new approach. *IEEE Transactions on Intelligent Vehicles*, 1(2):158–166.

[Li et al., 2019] Li, W., Pan, C., Zhang, R., Ren, J., Ma, Y., Fang, J., Yan, F., Geng, Q., Huang, X., Gong, H., et al. (2019). Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*.

[Lumelsky and Stepanov, 1987] Lumelsky, V. J. and Stepanov, A. A. (1987). Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2(1):403–430.

[Ma et al., 2020] Ma, Y., Wang, Z., Yang, H., and Yang, L. (2020). Artificial intelligence applications in the development of autonomous vehicles: a survey. *IEEE/CAA Journal of Automatica Sinica*, 7(2):315–329.

[Macedo et al., 2001] Macedo, J., Manduchi, R., and Matthies, L. (2001). Ladar-based discrimination of grass from obstacles for autonomous navigation. In *Experimental Robotics VII*, pages 111–120. Springer.

[Macfarlane and Stroila, 2016] Macfarlane, J. and Stroila, M. (2016). Addressing the uncertainties in autonomous driving. *SIGSPATIAL Special*, 8(2):35–40.

[Madrigal, 2017] Madrigal, A. C. (August 23, 2017). Inside waymo's secret world for training self-driving cars. *The Atlantic*.

[Mahmoud and Trilaksono, 2018] Mahmoud, T. and Trilaksono, B. R. (2018). Integrated ins/gps navigation system. *International Journal on Electrical Engineering and Informatics*, 10(3):491–512.

[Manduchi et al., 2005] Manduchi, R., Castano, A., Talukder, A., and Matthies, L. (2005). Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18(1):81–102.

[Mason et al., 2018] Mason, G. L., Williams, J. M., Vahedifard, F., and Priddy, J. D. (2018). A unified equation for predicting traction for wheels on sand over a range of braked, towed, and powered operations. *Journal of Terramechanics*, 79:33–40.

[Mattas et al., 2022] Mattas, K., Albano, G., Donà, R., Galassi, M. C., Suarez-Bertoa, R., Vass, S., and Ciuffo, B. (2022). Driver models for the definition of safety requirements of automated vehicles in international regulations. application to motorway driving conditions. *Accident Analysis & Prevention*, 174:106743.

[Matthies et al., 2005] Matthies, L., Bergh, C., Castano, A., Macedo, J., and Manduchi, R. (2005). Obstacle detection in foliage with ladar and radar. In *Robotics Research. The Eleventh International Symposium*, pages 291–300. Springer.

[Matthies et al., 1996] Matthies, L., Kelly, A., Litwin, T., and Tharp, G. (1996). Obstacle detection for unmanned ground vehicles: A progress report. In *Robotics Research*, pages 475–486. Springer.

[Matthies and Rankin, 2003] Matthies, L. and Rankin, A. (2003). Negative obstacle detection by thermal signature. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 1, pages 906–913. IEEE.

[Michelmore et al., 2018] Michelmore, R., Kwiatkowska, M., and Gal, Y. (2018). Evaluating uncertainty quantification in end-to-end autonomous driving control. *arXiv preprint arXiv:1811.06817*.

[Mittet et al., 2016] Mittet, M.-A., Nouira, H., Roynard, X., Goulette, F., and Deschaud, J.-E. (2016). Experimental assessment of the quanergy m8 lidar sensor. In *ISPRS 2016 congress*.

[Mullins et al., 2017] Mullins, G. E., Stankiewicz, P. G., and Gupta, S. K. (2017). Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1443–1450. IEEE.

[Mullins et al., 2018] Mullins, G. E., Stankiewicz, P. G., Hawthorne, R. C., and Gupta, S. K. (2018). Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles. *Journal of Systems and Software*, 137:197–215.

[Nakhaeinia et al., 2011] Nakhaeinia, D., Tang, S. H., Noor, S. M., and Motlagh, O. (2011). A review of control architectures for autonomous navigation of mobile robots. *International Journal of Physical Sciences*, 6(2):169–174.

[Narayanan et al., 1998] Narayanan, P., Rander, P. W., and Kanade, T. (1998). Constructing virtual worlds using dense stereo. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 3–10. IEEE.

[NVIDIA, 2022] NVIDIA (2022). Simulation for self-driving cars.

[Op den Camp et al., 2017] Op den Camp, O., van Montfort, S., Uittenbogaard, J., and Welten, J. (2017). Cyclist target and test setup for evaluation of cyclist-autonomous emergency braking. *International Journal of Automotive Technology*, 18(6):1085–1097.

[Oroko and Nyakoe, 2012] Oroko, J. A. and Nyakoe, G. (2012). Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: a review. *Proceedings of Sustainable Research and Innovation Conference*.

[Peterson and Glancy, 2018] Peterson, R. W. and Glancy, D. (2018). *The Future of Mobility and Shifting Risk*. PhD thesis, Santa Clara University.

[Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

[Rasshofer et al., 2011] Rasshofer, R. H., Spies, M., and Spies, H. (2011). Influences of weather phenomena on automotive laser radar systems. *Advances in Radio Science*, 9(B. 2):49–60.

[Rieder et al., 2002] Rieder, A., Southall, B., Salgian, G., Mandelbaum, R., Herman, H., Rander, P., and Stentz, T. (2002). Stereo perception on an off-road vehicle. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 1, pages 221–226. IEEE.

[Schöner, 2018] Schöner, H.-P. (2018). Simulation in development and testing of autonomous vehicles. In *18th Stuttgart international symposium, Stuttgart*, pages 03–14.

[Shah et al., 2018] Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer.

[Skoglund et al., 2016] Skoglund, M., Petig, T., Vedder, B., Eriksson, H., and Schiller, E. M. (2016). Static and dynamic performance evaluation of low-cost rtk gps receivers. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 16–19. IEEE.

[Stavens and Thrun, 2012] Stavens, D. and Thrun, S. (2012). A self-supervised terrain roughness estimator for off-road autonomous driving. *arXiv preprint arXiv:1206.6872*.

[Stevens et al., 2013] Stevens, M. T., Towne, B. W., Mason, G. L., Priddy, J. D., Osorio, J. E., and Barela, C. A. (2013). Procedures for one-pass vehicle cone index (vci1) determination for acquisition support. Technical report, ENGINEER RESEARCH AND DEVELOPMENT CENTER VICKSBURG MS GEOTECHNICAL AND . . . .

[Stock, 2018] Stock, K. (2018). Self-driving cars can handle neither rain nor sleet nor snow. *Bloomberg Businessweek*, 17.

[Sun et al., 2014] Sun, Y., Xiong, G., Song, W., Gong, J., and Chen, H. (2014). Test and evaluation of autonomous ground vehicles. *Advances in Mechanical Engineering*, 6:681326.

[Tasora et al., 2015] Tasora, A., Serban, R., Mazhar, H., Pazouki, A., Melanz, D., Fleischmann, J., Taylor, M., Sugiyama, H., and Negrut, D. (2015). Chrono: An open source multi-physics dynamics engine. In *International Conference on High Performance Computing in Science and Engineering*, pages 19–49. Springer.

[Vahedifard et al., 2017] Vahedifard, F., Williams, J. M., Mason, G. L., Howard, I. L., and Priddy, J. D. (2017). Development of a multi-year database to assess off-road mobility algorithms in fine-grained soils. *International journal of vehicle performance*, 3(1):3–18.

[Wald et al., 2014] Wald, I., Woop, S., Benthin, C., Johnson, G. S., and Ernst, M. (2014). Embree: a kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 33(4):143.

[Wang et al., 2002] Wang, L. C., Yong, L. S., and Ang, M. H. (2002). Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment. In *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, pages 821–826. IEEE.

[Welch and Connolly, 2006] Welch, B. W. and Connolly, J. W. (2006). Autonomous navigation error propagation assessment for lunar surface mobility applications. Technical report, NASA.

[Williams et al., 2019] Williams, J. M., Vahedifard, F., Howard, I. L., Borazjani, A., Mason, G. L., and Priddy, J. D. (2019). Mobility guidance for tracked vehicles on fine-grained soil from historical full-scale test data in drove 2.0. *Journal of Terramechanics*, 84:1–12.

[Woop et al., 2017] Woop, S., Benthin, C., and Áfra, A. T. (2017). Embree ray tracing kernels. *Available on: https://embree. github. io/data/embree-siggraph-2015-final. pdf.*

[Wurts et al., 2020] Wurts, J., Stein, J. L., and Ersal, T. (2020). Collision imminent steering at high speed using nonlinear model predictive control. *IEEE Transactions on Vehicular Technology*, 69(8):8278–8289.

[Yao et al., 2020] Yao, Q., Tian, Y., Wang, Q., and Wang, S. (2020). Control strategies on path tracking for autonomous vehicle: State of the art and future challenges. *IEEE Access*, 8:161211–161222.

[Zeng and Church, 2009] Zeng, W. and Church, R. L. (2009). Finding shortest paths on real road networks: the case for a. *International journal of geographical information science*, 23(4):531–543.

[Zhang, 1999] Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the seventh ieee international conference on computer vision*, volume 1, pages 666–673. Ieee.