

A COMPARISON OF VEHICLE HANDLING FIDELITY BETWEEN THE GAZEBO AND ANVEL SIMULATORS

Robert Brothers¹, David Bevly, PhD¹

¹Mechanical Engineering Department, Auburn University, Auburn, AL

ABSTRACT

This paper provides a comparison of the Gazebo and ANVEL simulators and analyzes the aspects of vehicle modeling fidelity that are critical to the design of unmanned ground vehicle (UGV) control and estimation algorithms. The robotic simulators Gazebo [1], from the Open Source Robotics Foundation (OSRF), and Autonomous Navigation Virtual Environment Laboratory (ANVEL) [2], from Quantum Signal, are two popular new tools that are being used extensively in academic, commercial, and military development of perception, navigation, and control algorithms for UGVs. Despite the similarities between Gazebo and ANVEL there has been no direct comparison between the two simulators with respect to their validity as vehicle dynamics simulators.

Citation: R. Brothers, D. Bevly, "A Comparison of Vehicle Handling Fidelity Between the Gazebo and ANVEL Simulators", In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, NDIA, Novi, MI, Aug. 13-15, 2019.

1. INTRODUCTION

Simulation environments have become a vital tool for developers of unmanned ground vehicle (UGV) algorithms. Applications of UGVs will revolutionize personnel and cargo transportation safety and efficiency. These applications require extensive testing which may not always be practical, or even feasible, on a real vehicle. Vehicle simulation testing offers an inexpensive way to accelerate UGV algorithm development and provide a way to test potentially dangerous situations at the vehicle's limits.

Over the past decade, a large number of robotic simulators have been developed to provide software-in-the-loop (SIL) and hardware-in-the-loop (HIL) interfaces to a realistic physics engine and video game quality graphics. The power of

HIL/SIL development is the ability to transfer the exact software or hardware component being developed directly to the target robot platform. The robotic simulators Gazebo [1], from the Open Source Robotics Foundation (OSRF), and Autonomous Navigation Virtual Environment Laboratory (ANVEL) [2], from Quantum Signal, are two popular new tools that are being used extensively in academic, commercial, and military development of perception, navigation, and control algorithms for UGVs.

2. BACKGROUND

In the field of robotics, simulator software changes rapidly. With so many options for simulators, no comparison can truly capture every choice that must be weighted when deciding on a

particular software package. Furthermore, simulator evaluations may become quickly outdated depending on development pace, user-community involvement, project cancellations, etc. However, many merit-worthy robotic simulator comparisons have been made. Craighead, et al. defined a set of simulator evaluation criteria that were mostly user experience focused and concluded that it is no longer necessary to build a simulator from scratch [3]. In 2009, Michal and Etkorn provided a comparison of Gazebo and Microsoft Robotics Development Studio (MRDS) [4], which evaluated the implementations of a common robot model, the fidelity of 2 common simulation tasks, and the difficulty of transferring the robot's simulation control software to the actual robot hardware [5]. Harris and Conrad also provided an extended survey of robotic simulators in 2011 that concluded all those available were not suitable for simulating the resource constraints that come with embedded development for small robotic platforms [6]. Their survey included Gazebo when it was still associated with the Player/Stage project [7], as well as USARSim [8], MRDS, and many others. A more recent survey [9] compared simulator features and performance benchmark tests between Gazebo, V-REP [10] and ARGoS [11].

Gazebo has made significant changes and improvements since many of the previously mentioned evaluations were published. It is now maintained by the OSRF and is widely used as the simulation tool-of-choice for users of the Robot Operating System (ROS) [12]. In 2015, Gazebo demonstrated its abilities to run near real-time simulations on cloud-based computing resources during the DARPA Virtual Robotics Challenge [13]. This competition simulated complex disaster scenarios to test humanoid robots as disaster response aides. The Gazebo team used ROS to manage the communications between multiple, distributed "field computers" running the robot's software and the simulation. Gazebo also claims to be the first low-cost robot simulator to accurately

model walking and grasping for a high degree-of-freedom (DOF) humanoid robot. By supporting multiple physics engines, Gazebo provides flexibility in the dynamic modeling of high DOF systems. The default physics engine for Gazebo is the Open Dynamics Engine (ODE) [14], which provides realistic multi-body physics and force-based joint controllers for any generic robot configuration.

Gazebo also gives developers the option to write a plugin that can modify almost any aspect of the simulation. For example, a Gazebo plugin was used to simulate fluid dynamics effects and custom actuator models for an unmanned underwater vehicle (UUV) [15]. Other researchers have used Gazebo plugins to model the flight dynamics of a quadrotor unmanned aerial vehicle (UAV) and accurately represent the sensor suite used on their real UAV [16]. The sensor plugins provided the same ROS interface used on the hardware and simulated complex error characteristics. These researchers successfully demonstrated SIL testing of their ROS-based simultaneous localization and mapping (SLAM) algorithm in Gazebo. The ease of ROS integration with Gazebo makes it an attractive choice for UGV simulation. ROS has been declared the *de facto* standard for robotic system research and development in academia, industry, and government organizations [17], as collaborative projects like ROS-Military have proven.

The creators of ANVEL provide a UGV focused simulator that combines realistic physical models and state-of-the-art video game graphics technology. Much like Gazebo, ANVEL uses ODE to simulate dynamics for UGV and robotic manipulator models but also incorporates advanced vehicle-terrain interaction (VTI) models to provide high fidelity traction forces on a variety of surface types (e.g. gravel, sand, concrete). ANVEL provides a number of standard sensors, such as inertial measurement units (IMU), GPS, and LiDAR, which are modeled with realistic noise, scale-factor, and bias effects [2]. Also similar to Gazebo, ANVEL provides a plugin interface that

allows users to interact with simulation elements directly. ROS users have taken advantage of this direct access to create an ANVEL-ROS bridge. A plugin is used to communicate with a set of ROS nodes that publish sensor streams using standard ROS messaging formats. Hudson et al. successfully ran an open-source SLAM algorithm with LiDAR point clouds published from the ANVEL-ROS bridge [18]. Other researchers have shown that custom wheel speed encoders, IMUs, and GPS sensors can be implemented as an ANVEL plugin and used in ROS-based SIL/HIL validation of navigation algorithms [19]. The current version of ANVEL (version 3.5) includes limited support for ROS in its distribution, as it has clearly been important to the user community.

Despite the similarities between Gazebo and ANVEL there has been no direct comparison between the two simulators with respect to their validity as vehicle dynamics simulators. This paper provides that comparison, using CarSim as the simulation standard. CarSim is a high-fidelity vehicle dynamics simulator used as the simulation standard by many OEMs and universities [20]. The work of Lo et al. is most similar to this work, as they provide a vehicle dynamics validation by comparing ANVEL, CarSim, and real vehicle data [21]. In this work, Auburn University's 2003 Infinity G35 test vehicle was modeled in ANVEL, CarSim, and Gazebo. Steady-state handling tests were conducted with the test vehicle and compared to CarSim to validate it as the simulation standard. Dynamic handling tests were then conducted in CarSim, ANVEL, and Gazebo for comparison.

The rest of the paper will be organized as follows: Section 3 presents vehicle models and explains their relevance in this analysis, Section 4 explains the test procedures conducted and the simulation model validation, Section 5 presents the simulation results and discussion, and finally, Section 6 gives a conclusion and discussion of future work.

3. VEHICLE MODELING

Modeling for a general dynamics engine is different from the writing of a purpose built simulation from a specific model. ODE provides primitive link and joint types which simulators, like Gazebo and ANVEL, use to allow users to build high DOF models to any level of complexity. Domain specific simulations often use optimizations and assumptions that a general dynamics simulator cannot. This section will first detail the model building process in the Gazebo and ANVEL simulators and then present some analytical modeling specific to the UGV domain that can help to understand the simulation results.

3.1. Simulator Model Building

Gazebo provides users the ability to create links in a model from primitive shapes, like boxes or cylinders, or more complex shapes from custom mesh files. Users must define the inertial properties for these links and can optionally define visual and collision properties. Model links can be chained together with any type of joint that ODE offers. These joints limit the model's degrees of freedom and provide a way to control and measure the state of the links that are attached. Although individual models are not restricted to a single kinematic chain, vehicle models typically are.

The kinematic chain for the Gazebo G35 model is given below in Figure 1. The link elements are all based on box and cylinder shapes and represented in the tree by boxes, and the joints are represented by the lines connecting the links.

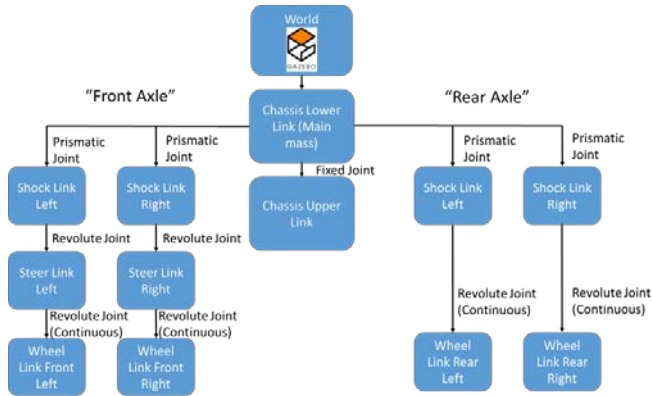


Figure 1: Gazebo G35 Kinematic Chain

The main chassis link is the “sprung mass” of the vehicle and sets the vehicle’s inertial properties. The sprung mass is connected to four independent suspension elements with prismatic joints. These prismatic joints have linear spring rates and linear damping that model the vehicle’s actual suspension. At the “front axle”, the suspension links are connected to a single revolute type joint that allows rotation about one axis. This joint is used to control the steering of the vehicle. The steering plugin converts steering commands from the angle at the steering wheel to the correct Ackermann angle that the tire should be controlled to. Each wheel element is modeled as a simple cylinder and is connected to either a steering joint or suspension link through a revolute joint that allows the wheel to roll about its spin axis. In a custom plugin, the vehicle’s drive train is modeled by converting throttle and brake signals to torque controls at the rear wheels’ revolute joints.

In ANVEL, modeling began by modifying the Sedan vehicle provided in its distribution. Starting with this existing model as a template made the process of creating a custom vehicle much simpler than in Gazebo. ANVEL provides abstractions for the vehicle control interfaces (steering, throttle, and braking) in the vehicle definition file. These interfaces allow the vehicle to be controlled through ANVEL’s API rather than having to write a custom plugin, although the latter can certainly be done and may be desirable for some users.

Abstractions are also provided for defining suspension and wheel elements in ANVEL. Users can define linear dynamic properties of these model elements as well as custom nonlinear lookup tables. The ANVEL G35 model uses only linear dynamic properties in an attempt to make a fair comparison with the Gazebo model. ANVEL’s modeling abstractions make it hard to determine the actual kinematic chain that is implemented with ODE link and joint elements, but the way that the wheels and suspension are defined suggest that the kinematic chain looks like the one given in Figure 2.

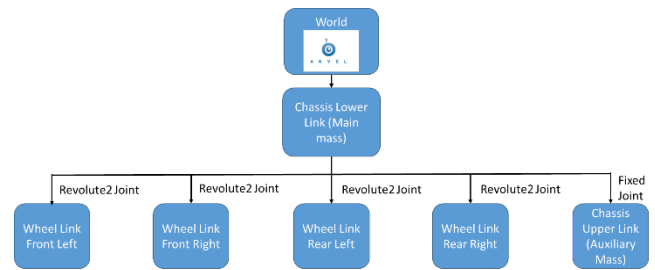


Figure 2: ANVEL G35 Kinematic Chain

Because the suspension properties are defined within the wheel element, the user could guess that the underlying implementation uses the ODE hinge-2 joint type. This joint provides two axes of rotation between links. One axis can be used for steering and the other for the wheel’s spin axis. ODE explicitly defines that this joint type has suspension-like spring and damping properties. The ODE user guide depiction of this joint is given below in Figure 3.

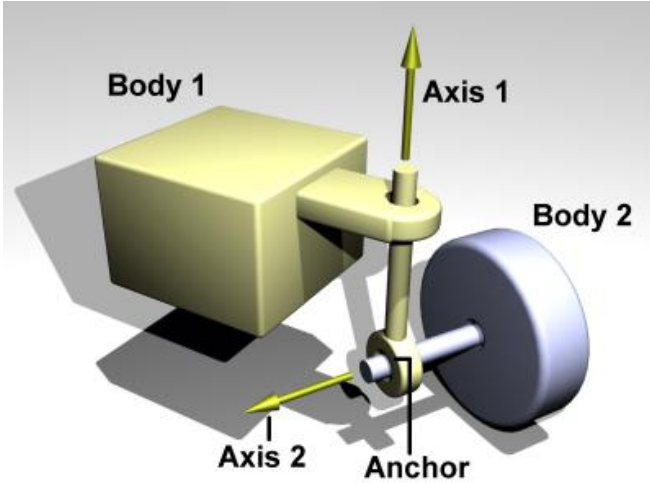


Figure 3: ODE Hinge 2 Joint [14]

While there are many differences in the model definition between Gazebo and ANVEL that are beyond the designer’s control, there are many common properties that were used in both Gazebo and ANVEL. The following table defines the vehicle properties that are common to both simulators.

Table 1: Simulated Vehicles' Common Properties

Property	Value
CG to front axle (a)	1.3680 m
CG to rear axle (b)	1.4820 m
Track width	1.5062 m
Sprung mass (m)	1348.2 Kg
Yaw inertia (I_{zz})	2698.9 $Kg \cdot m^2$
Front suspension stiffness	34,000 N/m
Rear suspension stiffness	46,000 N/m
Suspension damping	3286.8 $N \cdot s/m$

Some of these properties were not identified from the actual G35 test vehicle. For these properties, the reference CarSim vehicle, which was derived from the built in class-D sedan, were duplicated in the Gazebo and ANVEL models. The yaw inertia was also duplicated from the CarSim vehicle. This is a reasonable assumption given the physical

dimensions of all the models are equivalent. The spring and damper values for the suspension were duplicated from the CarSim vehicle as well to try and minimize unintended differences in each model’s dynamic performance due to weight transfer.

3.2. Analytical Vehicle Modeling

While ground vehicles are complex multi-body systems that can be represented with high DOF dynamic models, many vehicle control designers use simplified low order models that can still accurately represent the vehicle’s transient response. One such simplification is the planar bicycle model, shown below in Figure 4.

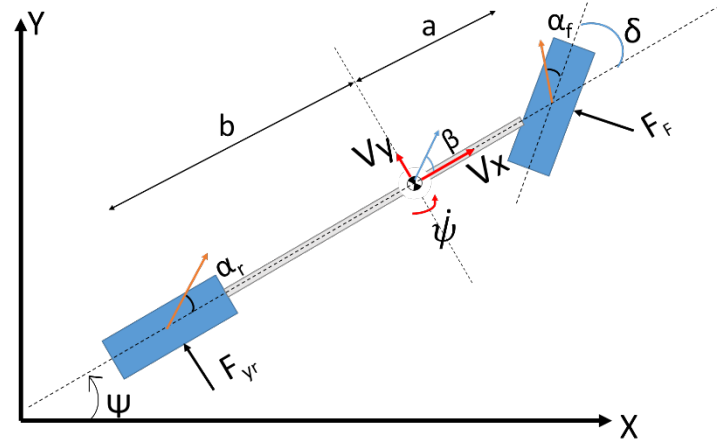


Figure 4: Bicycle Model

The following coordinate frames are used in Figure 4 and throughout the rest of this section: $\{x, y, z\}$ refers to the vehicle forward, left, and up axes, and $\{X, Y, Z\}$ refers to a local fixed frame, such as the East, North, Up convention. Although the bicycle model ignores pitch and roll motions, it gives important insight into the vehicle yaw and lateral velocity dynamic. The planar bicycle model equations of motion are given in Equations (1-2), where F_f and F_r are the front and rear lateral tire forces respectively, m is the vehicle mass, I_{zz} is the vehicle yaw inertia, δ is the vehicle steering angle at the tire, a is the distance from the vehicle center

of gravity (CG) to the front axle, and b is the distance from the vehicle CG to the rear axle.

$$m\dot{y} = F_f \cos(\delta) + F_r \quad (1)$$

$$I_{zz}\ddot{\psi} = aF_f \cos(\delta) - bF_r \quad (2)$$

The tire slip angles, α_f and α_r , relate the vehicle's lateral kinematics to the tire force generation. The front and rear slip angles are given in Equation (3) and Equation (4) respectively in terms of kinematic variables at the vehicle CG.

$$\alpha_f = \arctan\left(\frac{V_y + a\dot{\psi}}{V_x}\right) - \delta \quad (3)$$

$$\alpha_r = \arctan\left(\frac{V_y - b\dot{\psi}}{V_x}\right) \quad (4)$$

In the equations above, V_x and V_y are the vehicle forward and lateral velocities respectively, and $\dot{\psi}$ is the vehicle yaw-rate. Assuming these angles remain small, the following approximations given below in Equations (5-6) can be used.

$$\alpha_f \approx \left(\frac{V_y}{V_x}\right) + \frac{a\dot{\psi}}{V_x} - \delta \quad (5)$$

$$\alpha_r \approx \left(\frac{V_y}{V_x}\right) - \frac{b\dot{\psi}}{V_x} \quad (6)$$

The lateral force generated at each tire is related to the tire slip angle. This relationship is nonlinear in general and depends on many variables such as tire normal force, tire pressure, tire construction, etc. Complex nonlinear models, such as the Pacejka tire model, capture the effects of tire force saturation [22]. An example nonlinear tire curve is shown in Figure 5.

For a given normal force and small slip angles, the lateral force is linearly proportional to tire slip. The proportionality constant, C_{α} , is commonly referred to as the tire cornering stiffness. This linear

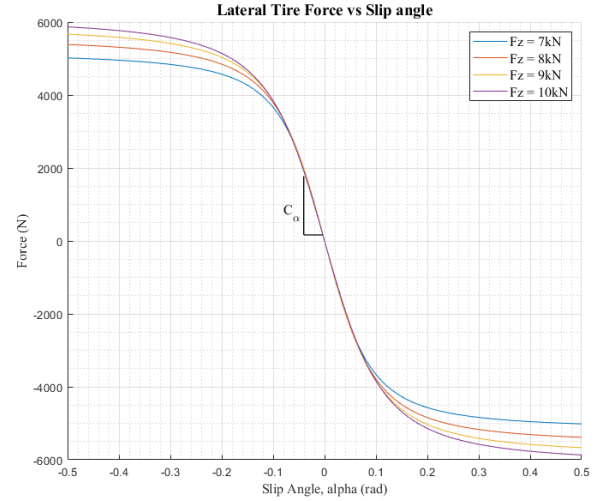


Figure 5: Nonlinear Tire Force Curve

relationship is shown in Equations (7-8) for the front and rear lateral tire force of the bicycle model.

$$F_f = -C_{\alpha f} \alpha_f \quad (7)$$

$$F_r = -C_{\alpha r} \alpha_r \quad (8)$$

Combining Equations (5-8) and substituting them into Equations (1-2), the lateral velocity and yaw-rate dynamic equations can be formulated with a single steering angle input.

$$\dot{V}_y = \frac{-(C_{\alpha f} + C_{\alpha r})}{mV_x} V_y + \frac{-(aC_{\alpha f} - bC_{\alpha r}) - mV_x^2}{mV_x} \dot{\psi} + \frac{C_{\alpha f}}{m} \delta \quad (9)$$

$$\ddot{\psi} = \frac{-(aC_{\alpha f} - bC_{\alpha r})}{I_{zz}V_x} V_y + \frac{-(a^2C_{\alpha f} + b^2C_{\alpha r})}{I_{zz}V_x} \dot{\psi} + \frac{aC_{\alpha f}}{I_{zz}} \delta \quad (10)$$

The model in Equations (9-10) is linear if a constant forward velocity, V_x , is assumed. However, if V_x is varying, the poles and zero position of this model will change. Above a certain speed, the model may become non-minimum phase, i.e. the model zero becomes positive. This non-minimum phase effect appears in the transients of the lateral velocity state, and will be important in evaluating the validity of the simulated vehicle models. Using Equations (9-10) the transfer function from steering input to

lateral velocity output is given below in Equation (11).

$$\frac{V_y(s)}{\delta(s)} = \frac{\frac{C_{\alpha f}}{mV_x}s + \frac{C_2 C_{\alpha f} - a C_1 C_{\alpha f} - a m C_{\alpha f} V_x^2}{I_{zz} m V_x^2}}{s^2 + \frac{I_{zz} C_0 + m C_2}{I_{zz} m V_x} s + \frac{C_2 C_0 - C_1^2 + m V_x^2 C_1}{I_{zz} m V_x^2}} \quad (11)$$

The simplification terms C_0 , C_1 , and C_2 are given in equations (12-14) respectively.

$$C_0 = C_{\alpha f} + C_{\alpha r} \quad (12)$$

$$C_1 = a C_{\alpha f} - b C_{\alpha r} \quad (13)$$

$$C_2 = a^2 C_{\alpha f} + b^2 C_{\alpha r} \quad (14)$$

Solving the numerator of Equation (11) for the vehicle's forward speed, V_x , results in the speed at which the model becomes non-minimum phase. This crossover speed is given by Equation (15).

$$V_x = \sqrt{\frac{b(a+b)C_{\alpha r}}{am}} \quad (15)$$

Note that the only constant in Equation (15) that is not explicitly set in the Gazebo and ANVEL models is the tire stiffness for the rear axle.

Vehicle handling is commonly characterized using steady-state steering response. Gillespie claims that the understeer gradient is the most common performance measure for open-loop handling [23]. The understeer gradient, K_{us} , is a constant that relates the amount of steering input required to hold a steady-state turn given the vehicle's lateral acceleration at the CG, a_y . This relationship is given in Equation (16).

$$\delta = \frac{L}{R} + K_{us} \cdot a_y \quad (16)$$

In the equation above, L is the vehicle wheel-base length and R is the turning radius. Equation (16) can be derived by substituting Equations (5-6) into Equations (1-2), assuming R is known and the

lateral acceleration in steady-state is given by Equation (17).

$$a_y = \frac{V^2}{R} \quad (17)$$

A detailed derivation can be found in [23].

The models given in Equations (9-10) and Equation (16) are common approximations used by vehicle control engineers to characterize and design steering control systems. The understeer gradient can be easily estimated with minimal instrumentation for a baseline model validation. The linear dynamic model can be used as an intuitive starting point for validating the transient behavior of a more complex model or real vehicle. The next section will detail the testing procedures used to analyze each simulator's vehicle dynamics performance, keeping these simple models in mind.

4. TESTING PROCEDURES

Two different testing procedures were performed as the basis for this comparison. First, the steady-state characteristics of the real test vehicle are compared to CarSim to check that the CarSim model's handling characteristics are close to the actual vehicle. Second, the CarSim model was used to generate reference steering inputs for testing the transient response of the Gazebo and ANVEL models. Using CarSim as the reference for the transient tests reduces sources of uncertainty for comparison purposes because the CarSim outputs are not subject to random noise or human error in conducting the tests.

In the previous work done by Lo et al., this same two step validation was done: CarSim was compared to real data from the target vehicle, and the ANVEL model's transient response was compared to CarSim with a double lane change maneuver [21]. This work introduces Gazebo to the mix (version 7.15) and uses a newer version of ANVEL (version 3.5).

4.1. Steady-state Handling Tests

Four common test procedures can be performed to estimate the understeer gradient for steady-state validation. Any two of the three independent variables (δ, V, R) in Equation (16) can be varied while the other is held constant. The constant radius test holds R constant along a circular track. The vehicle speed is slowly incremented such that the steady-state steering angle is developed at each step. This test was selected because of testing space constraints and simple instrumentation.

A 50ft radius circular track was set up at Auburn University’s NCAT test track, and three test runs were conducted. A 2003 Infinity G35 test vehicle, shown in Figure 6, was outfitted with a Septentrio PolaRx2 GPS receiver for accurate velocity measurements, and steering wheel angle data was collected over the vehicle’s CAN bus. A least squares estimation technique was used to find the best fit understeer gradient from all three datasets. The same test was conducted with the CarSim G35 model, and simulation truth lateral acceleration and steering angle were recorded. The simulation data is plotted over the test vehicle data and understeer gradient fit in Figure 7.



Figure 6: G35 Test Vehicle at NCAT Test Track

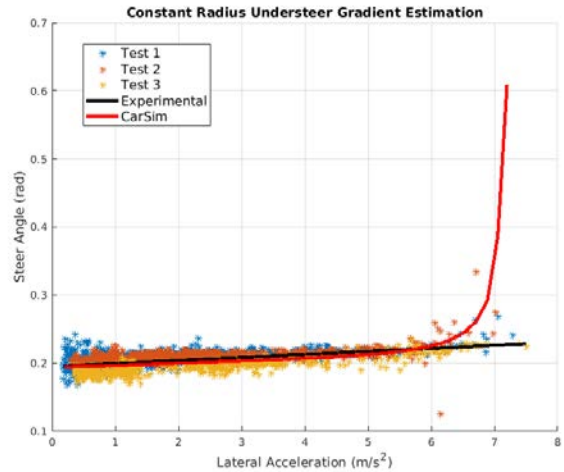


Figure 7: Understeer Gradient Estimation Experimental vs CarSim

The linear fit of understeer gradient matches closely with CarSim in the lateral acceleration range for normal driving. The CarSim simulation was driven to full oversteer, or until the vehicle completely lost traction, at $\sim 7 \text{ m/s}^2$. The linear fit does not reflect this traction limit, but the data appears to show a similar maximum lateral acceleration.

4.2. Transient Handling Tests

The double lane change test is often used to characterize a vehicle’s transient steering response. The double lane change simulates a real-world emergency driving maneuver that can produce high lateral acceleration, or even trigger a roll-over event for vehicles with a high CG. In this study, the lateral velocity and yaw-rate states will be used as the comparison metrics. As mentioned in the section above, the pole and zero positions of the dynamic model defined in Equations (9-10) change with vehicle speed. For this reason, it is important to compare the simulation results at different speeds.

Reference simulations were run in CarSim at 5 m/s, 10 m/s, and 20 m/s using the built-in double lane change procedure. This procedure uses the path defined by the ISO 3888-1 standard [24] and closed-loop steering and throttle controllers to execute the maneuver. Along with the true lateral

velocity and yaw-rate, the true wheel angles for each run were collected. The wheel angles for the 20 m/s simulation are shown below in Figure 8.

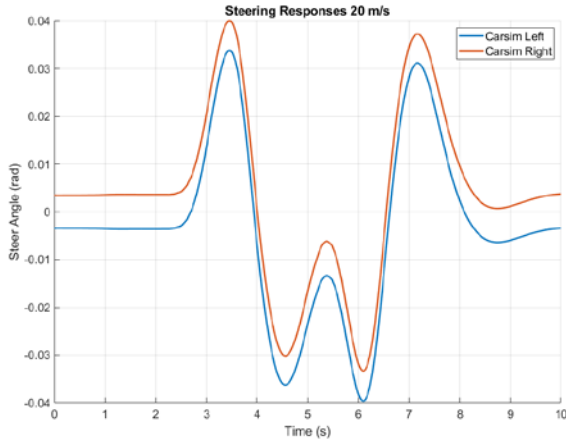


Figure 8: CarSim Double Lane Change Wheel Angles at 20 m/s

The same double lane change simulations were run in ANVEL and Gazebo. To reduce discrepancies in the actual input steering, the collected CarSim wheel angle data was used to directly control the steering of the ANVEL and Gazebo models. In Gazebo, this was achieved by publishing the desired steer values over a ROS topic to the low-level steering controller in the model plugin. The ANVEL model was controlled via the external API. Both models used closed-loop throttle control to maintain the desired test speed.

5. SIMULATION RESULTS

The lateral velocity and yaw-rate states were collected from each ANVEL and Gazebo test run for comparison. In ANVEL, each test was conducted with the default ODE friction VTI model and the Pacejka VTI model. All tests in Gazebo relied on the ODE friction model for tire force generation. The results for the 5 m/s, 10 m/s, and 20 m/s tests are shown below in Figure 9, Figure 10, and Figure 11 respectively.

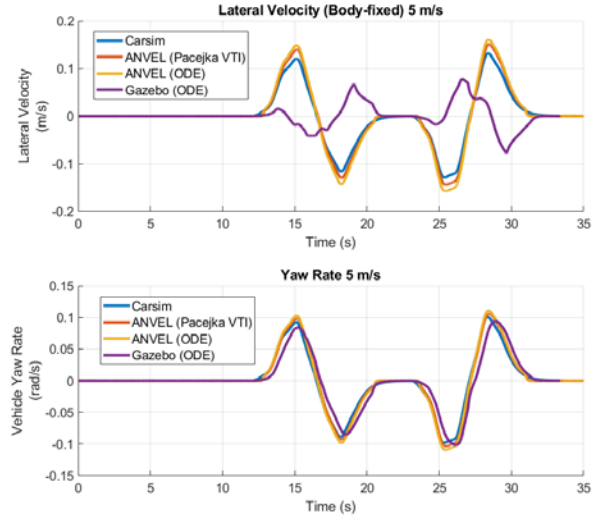


Figure 9: Lateral Velocity (Top) and Yaw-rate (Bottom) Comparison for DLC at 5m/s

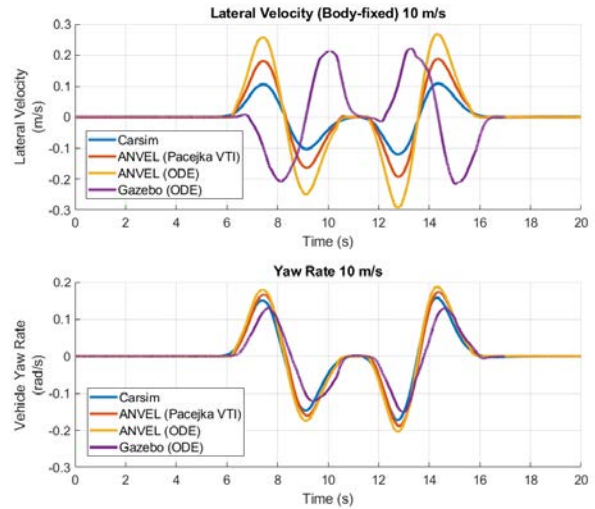


Figure 10: Lateral Velocity (Top) and Yaw-rate Comparison for DLC at 10 m/s

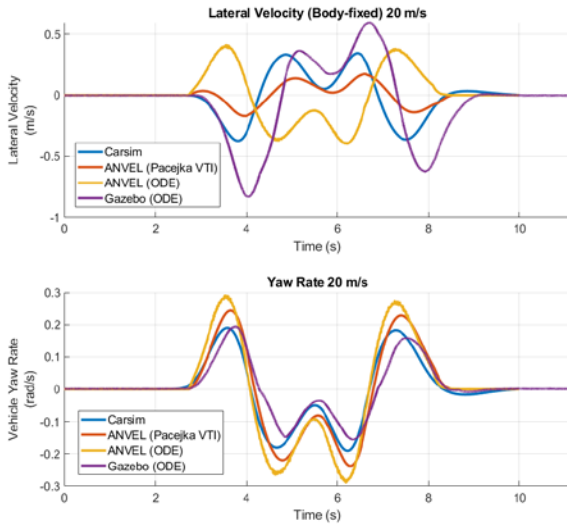


Figure 11: Lateral Velocity (Top) and Yaw-rate Comparison for DLC at 20 m/s

In the low speed case of 5 m/s, the yaw-rate of all four simulations match well. On the other hand, the Gazebo model’s lateral velocity does not appear to follow the trend of the other three simulations. In fact, it looks as if the direction of the lateral velocity is opposite that of CarSim and ANVEL. This same phenomena can be observed with a realization of the bicycle model that is non-minimum phase. Figure 10 shows good correlation in yaw-rate, but the lateral velocity of all four simulators seems to diverge. For this test, the Gazebo model again exhibits non-minimum phase behavior and the CarSim model appears to be getting closer to the crossover speed defined in Equation (15). At 20 m/s Gazebo, CarSim and ANVEL (Pacejka VTI) all show non-minimum phase behavior. As previously noted, the only term in Equation (15) that was not explicitly set in the Gazebo and ANVEL models is the rear tire stiffness. Controlling the effective tire stiffnesses of each axle, particularly the rear axle, would have the greatest impact on bringing the lateral velocity response of each of these simulators closer together. At all test speeds, the ANVEL model using the Pacejka VTI matches closest with CarSim in both the lateral velocity and yaw-rate states, further suggesting that detailed modeling of

the tires has the largest effect on the lateral dynamic output.

Position data, relative to the simulator’s origin, was also collected for each double lane change test. The difference between the CarSim path and both the ANVEL and Gazebo paths could be corrected with feedback control, like many path-tracking controllers employ. Because path-tracking was not an objective of this study, the ANVEL and Gazebo models’ steering were controlled open loop. The difference in the fixed-frame path can be studied to evaluate how errors in the lateral dynamics can accumulate in the vehicle position. Figure 12, Figure 13, and Figure 14 show the fixed-frame positions of each model at 5 m/s, 10 m/s, and 20 m/s, respectively.

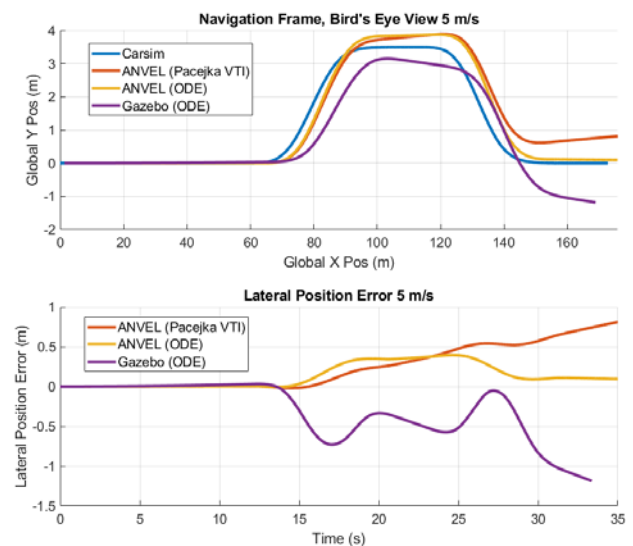


Figure 12: Fixed-frame positions of DLC at 5 m/s

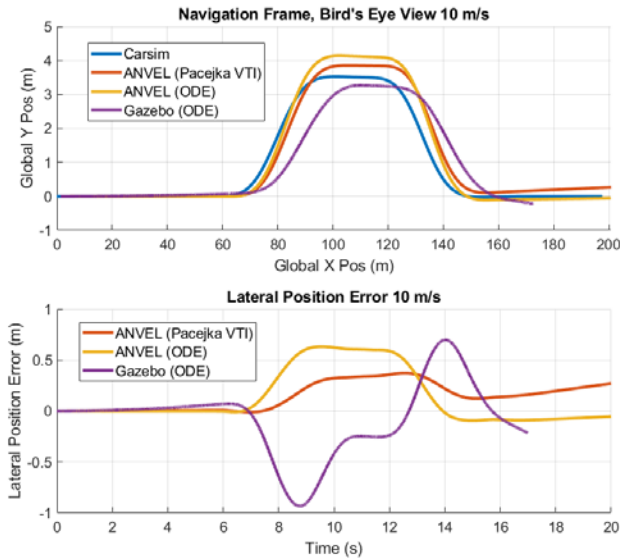


Figure 13: Fixed-frame positions of DLC at 10 m/s

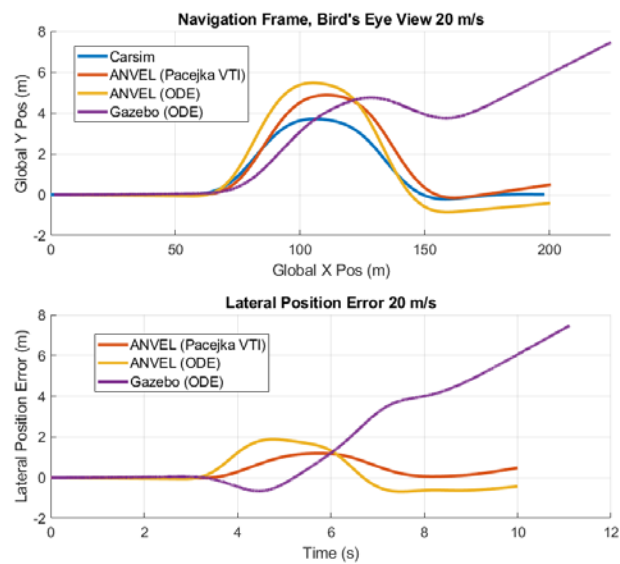


Figure 14: Fixed-frame positions of DLC at 20 m/s

Figure 12 and Figure 13 show the both ANVEL and Gazebo have similar errors (relative to the CarSim path), within ± 1.5 m, at low speeds. At the higher speed shown in Figure 14 however, a significant yaw difference has accumulated in the Gazebo model, causing the position difference to grow unbounded at a high rate. This is likely because contact forces generated at the tire/surface

collision for higher values of wheel slip are not being accurately modeled with the ODE friction model in Gazebo.

6. CONCLUSIONS AND FUTURE WORK

For decades, industry and academia have relied on simulators, such as CarSim, with specialized vehicle dynamic models to rapidly prototype control and estimation algorithms for UGVs. New robotics simulation platforms, such as ANVEL and Gazebo, provide better HIL/SIL capabilities and deep integration for the tools used to develop UGV software, such as ROS, making them attractive alternatives. The high dynamic double lane change test was used in this work to compare vehicle handling capabilities of ANVEL and Gazebo, using CarSim as a reference.

The simulations in this study suggest advanced tire and terrain modeling produce handling dynamics closer to the reference model, CarSim. The ANVEL vehicle using the Pacejka VTI model has the most similar performance. At low speeds, Gazebo has comparable handling to ANVEL; however, for higher speed applications, ANVEL may be the more suitable simulator. This study has shown that differences in the ANVEL and Gazebo handling is most visible in the lateral velocity dynamic state because of the non-minimum phase effect at higher speed. In the future, the crossover velocity, or the velocity at which the system becomes non-minimum phase, may be used as a metric to bring the handling characteristics of these two simulators closer together.

Future work with both simulators is expected, as each has their relative merits. Gazebo is an incredibly flexible robotics simulator with deeply integrated support for ROS. ANVEL provides easier out-of-the-box setup for vehicle specific simulations and supports a multi-language API that provides a quick and simple way to interact with many aspects of the simulation. In future work, a custom plugin will be created to implement a more advanced tire model for the Gazebo vehicle. If the same Pacejka model used in ANVEL can be

developed for Gazebo, then a more direct comparison can be made between the two. Because vehicle weight transfer plays an important role in the maximum capable traction, the effects of suspension modeling for both simulators should also be evaluated. In conclusion, improving the vehicle dynamics fidelity of both HIL/SIL tools will help designers of UGV systems rapidly develop algorithms with safety and confidence.

1. REFERENCES

- [1] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, an Open-source Multi-Robot Simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154 vol.3, Sep. 2004.
- [2] P. J. Durst, C. Goodin, C. Cummins, B. Gates, B. Mckinley, T. George, M. M. Rohde, M. A. Toschlog, and J. Crawford, "A Real-time, Interactive Simulation Environment for Unmanned Ground Vehicles: The Autonomous Navigation Virtual Environment Laboratory (ANVEL)," in *2012 Fifth International Conference on Information and Computing Science*, pp. 7–10, July 2012.
- [3] J. Craighead, R. Murphy, J. Burke, and B. Goldiez, "A Survey of Commercial & Open Source Unmanned Vehicle Simulators," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 852–857, April 2007.
- [4] J. Jackson, "Microsoft Robotics Studio: A Technical Introduction," *IEEE Robotics Automation Magazine*, vol. 14, pp. 82–87, Dec 2007.
- [5] D. S. Michal and L. Etzkorn, "A Comparison of Player/Stage/Gazebo and Microsoft Robotics Developer Studio," in *Proceedings of the 49th Annual Southeast Regional Conference, ACM-SE '11*, (New York, NY, USA), pp. 60–66, ACM, 2011.
- [6] A. Harris and J. M. Conrad, "Survey of Popular Robotics Simulators, Frameworks, and Toolkits," in *2011 Proceedings of IEEE Southeastcon*, pp. 243–249, March 2011.
- [7] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage Project: Tools for Multi-robot and Distributed Sensor Systems," in *Proc. of the Intl. Conf. on Advanced Robotics (ICAR)*, (Coimbra, Portugal), p. 317323, 2003.
- [8] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "USARsim: a Robot Simulator for Research and Education," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1400–1405, April 2007.
- [9] L. Pitonakova, M. Guiliani, A. Pipe, and A. Winfield, "Feature and Performance Comparison of the V-REP, Gazebo, and ARGoS Robot Simulators," in *Proceedings of the 19th Towards Autonomous Robotic Systems Conference*, vol. 10965 of *Lecture Notes in Computer Science*, pp. 357–368, Springer, 2018.
- [10] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A Versatile and Scalable Robot Simulation Framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, Nov 2013.
- [11] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a Modular, Parallel, Multi-engine Simulator for Multi-robot Systems," *Swarm Intelligence*, vol. 6, pp. 271–295, Dec 2012.
- [12] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an Open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

- [13]C. E. Agüero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo, E. Krotkov, and G. Pratt, “Inside the Virtual Robotics Challenge: Simulating Real-time Robotic Disaster Response,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, pp. 494–506, April 2015.
- [14]R. Smith, “Open Dynamics Engine ODE Multibody Dynamics Simulation Software.” Internet: www.ode.org, April 2019.
- [15]D. Meger, J. C. G. Higuera, A. Xu, P. Giguère, and G. Dudek, “Learning Legged Swimming Gaits From Experience,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2332–2338, May 2015.
- [16]J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Von Stryk, “Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo,” in *Simulation, Modeling, and Programming for Autonomous Robots* (I. Noda, N. Ando, D. Brugali, and J. J. Kuffner, eds.), (Berlin, Heidelberg), pp. 400–411, Springer Berlin Heidelberg, 2012.
- [17]J. Towler and M. Bries, “ROS-Military: Progress and Promise,” in *Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2018.
- [18]C. R. Hudson, A. Lalejini, B. Odom, C. L. Bethel, D. W. Carruth, P. J. Durst, and C. Goodin, “ANVEL-ROS: The Integration of the Robot Operating System with a High-fidelity Simulator,” in *Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2015.
- [19]B. Nelson, D. Bevly, J. Ratowski, and B. Theisen, “ANVEL HIL/SIL additions for Navigation Algorithm Development,” in *Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2017.
- [20]M. S. Corperation, “Customer Testimonials.” Internet: <https://www.carsim.com/publications/testimonials.php>, 2019.
- [21]J.-H. Lo, S. Eye, S. M. Rohde, and M. M. Rohde, “Simulation-based Prototyping of Stability Control Concepts for High-speed Teleoperation of Heavy Vehicles,” in *Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2016.
- [22]H.B. Pacejka, *Tyre and Vehicle Dynamics*. Oxford, England: Elsevier Butterworth-Heinemann, 2006.
- [23]T.D. Gillespie, *Fundamentals of Vehicle Dynamics*. Warrendale, PA: SAE International, 1992.
- [24]*Passenger Cars – Test Track for a Severe Lane-change manoeuver – Part 1: Double-lane change*, ISO 3888-1:2018, International Organization for Standardization, Geneva, Switzerland, Dec. 2018.