PROPRIOCEPTIVE SENSING OF TERRAIN FORCES BY COMPLIANT, FOUR-WHEELED ROVING MODULES

Adam Gronewold[a], Joshua Elliott[b], Christopher Lyke[c], Austin Lines[d], Marguerite Généreuxe, Grace Player[f] Andrew Skow[g], Philip Mulford[h], & Dr. Laura Ray Ph.D.[i]

Thayer School of Engineering at Dartmouth College, 14 Engineering Drive, Hanover, NH 03755
[a]Adam.M.Gronewold.TH@Dartmouth.edu, [b]joshuajameselliott@gmail.com, [c]Christopher.M.Lyke.21@Dartmouth.edu,
[d] austin.lines@gmail.com, [e]Marguerite.A.Genereux.21@Dartmouth.edu, [f]Grace.M.Player.21@Dartmouth.edu,
[g]Andrew.E.Skow.21@Dartmouth.edu, [h]Philip.M.Mulford.Jr.TH@Dartmouth.edu, [i]Laura.E.Ray@Dartmouth.edu

Abstract

This paper presents a class of four-wheel drive autonomous robots designed to collaboratively traverse terrains with a deformable upper layer, where soil properties result in limited traction and have the potential to cause immobilization. The robots are designed to have front and rear axle yaw degrees of freedom, and front and rear axle roll degrees of freedom providing ground compliance and maneuverability on friable terrain. These degrees of freedom, along with four individually driven wheels and an actuated translational degree of freedom inside a mid-frame joint, enable poses and modes of mobility that differ significantly from a rigid vehicle. A primary goal of this work is to assess the capacity to use this vehicular form as a testbed that leverages these vehicle dynamics to assess mobility. Using a custom ROS-Gazebo simulation environment, a heterogenous driving surface is created and used to evaluate this capability. We show that the vehicle can sense imbalanced terrain resistances proprioceptively. Additionally, we show that rigidity of the vehicle can be controlled through a simple feedback control loop governing the robot's unconstrained axles to maintain a proper heading angle and still can provide an avenue to monitor the dynamics related to full-vehicle immobilization.

Keywords: Modular robotics, Simulation, Dynamic modeling, Terrain compliant rovers
Presenter: Adam Gronewold

1. Introduction

Mobile platforms for use over long distances must provide reliable mobility and minimize energy use. A complete vehicle-terrain model captures the vehicle dynamic response when moving over a terrain as well as the terrain response to a vehicle driving over the terrain; retaining mobility on low cohesion terrain requires vehicle-terrain models that capture important coupling or interaction dynamics that are absent in existing models. Investing in vehicle-terrain modeling for lightweight robots supports vehicle and tractive element design, detection of incipient immobilization in time to avoid becoming mired, and subsequent control strategies to make headway while avoiding immobilization. An extensive literature exists for wheel and track interactions with soils under steady longitudinal motion. These models are derived from Bekker terramechanics theory (Bekker, 1962; Bekker, 1969; Wong & Reece, 1967a,b), which was developed and validated for *heavy* vehicles. It has been shown independently by a number of authors that the behavior of *light* vehicles differs from what this theory predicts (Lever et al., 2006; Ray, 2009; Jayakumar et al., 2014; Meirion-Griffith and Spenko, 2011). Specifically, for lightweight vehicles in soft terrain, the normal pressure distribution that gives rise to tractive forces deviates from the Bekker model. Additionally, effects of slip-induced sinkage (i.e., a robot "digging itself in") through excavation of friable terrain is less often considered.

Vehicle-terrain models for lightweight ground robots must be extended to include *all* important dynamics, verified through *in situ* experiments, and integrated with vehicle design and control: existing models do not adequately predict dynamical system behavior in low mobility conditions and are not amenable to real-time diagnostics and control. In an effort to extend these models, several authors have used single-wheel testbeds comprised of a loaded wheel instrumented to measure slip, wheel speed, normal load, drawbar pull, resistance, and sinkage (Meirion-Griffith and Spenko, 2011; Irani et al., 2010; Bauer et al., 2005). While single-wheel testbeds enable the development of longitudinal terrain force

models, they fail to capture the interactions that exist between terrain forces at each wheel and the resulting vehicle motion or propensity for immobilization. Lines et al. (2021) observed the vehicle dynamics coupled with the terrain interaction cause effects that govern the mobility of the entire robot. In particular, they document an immobilization mode where two diagonal wheels of a 90-kg robot lose net traction and stall, while the other two wheels experience full slip conditions prior to immobilization with corresponding terrain excavation. Such a failure mode indicates the need to address the relationship between spatially varying terrain parameters and the dynamics of the vehicle as a whole; if a vehicle loses traction at one wheel, the resulting yaw moment on the vehicle extends beyond the domain modeled by Bekker-like models (steady, longitudinal motion). *In situ* evaluation of terrain characteristics requires a testbed that can be used directly on a terrain of interest.

This paper presents a class of four-wheel rovers that serves this purpose. The modules, termed SHREWs, are designed to explore environments where inaccessibility and extreme conditions necessitate autonomous functionality (e.g., sand or snow-covered terrain). The rover, shown in Fig. 1, was designed as a solution to the 2020 NASA BIG Idea Challenge, which sought out novel concepts to investigate lunar permanently-shadowed regions. Inspired by the common shrew, which, when newly born and still mostly blind, collaboratively navigates the forest floor by forming a linked chain of pups behind their mother, SHREWs can join to form caravans to travel as a collective. In the face of low-cohesion soils like lunar regolith, modules subject to immobilizing terrain are extricated by mobile members of the same chain. This allows mechanical design to aid in resolving domain-specific challenges autonomy alone is ill-equipped to resolve.

Fig. 1*. A SHREW prototype.*

While collective autonomy of these modules is of interest, the dynamics of an individual SHREW robot enable novel mobility modes and poses that differ significantly from rigid, four-wheel drive robots. Each SHREW (22-35kg depending on instrumentation) has front and rear axle yaw degrees of freedom, and front and rear axle roll degrees of freedom provide ground compliance. These, along with four individually driven wheels and an actuated translational degree of freedom inside a mid-frame joint, comprise nine total degrees of freedom – five of which are driven by motors. The actuated midframe enables the SHREW to change its wheelbase, advancing the front and rear ends in an inchworm-type maneuver. The functionality allows the robot to push and pull itself through immobilizing terrain, providing net force to one side of the unit that supplements that provided by drive motors. The yaw joints allow a SHREW to take on a variety of poses (Fig. 2). A total of five pose types, and intermediaries between them, are possible – two for straight travel (standard and crab) and three for turning (symmetric, asymmetric-to-the-front, and asymmetric-to-the-rear). Straight-line motion in the crab configuration allows a SHREW to avoid dislodged material and to proceed with four distinct wheel tracks to limit sinkage. Asymmetric configurations are well suited to simplify the dynamics of an entire caravan in that a single reference heading angle must be determined for each vehicle instead of two. Lastly, symmetric steer, which is similar to Ackerman steering, limits the scrubbing encountered in differential turning of a rigid vehicle and decreases the minimum turning radius of the SHREW.

During operation**, s**imple feedback control laws maintain the velocity of each wheel, as shown in Fig. 3 (top). Each feedback loop is nested, in parallel with its same-axle counterpart, inside either a front or rear proportional feedback controller to maintain the angle of the respective axle by incrementing the reference speed of one wheel and decrementing the reference speed of the other (Fig. 3 bottom). The cascading architecture is used to maintain and switch between poses while in motion. The capacity to change modes enhances the mobility of both individual modules and a caravan as a whole. The simple proportional feedback controller with gain K in Fig. 3 (bottom) can be adjusted to enforce rigidity of the vehicle or allow some compliance.

The dynamics of an individual SHREW provide a novel testbed for assessing terrain characteristics. On a heterogenous surface, resistance forces are imbalanced from right to left. This naturally imparts a yaw moment on the leading axle, which causes one wheel to advance more rapidly than the other. Because the individual axles of a SHREW are unconstrained in a manner that rigid, four-wheel vehicles are not, an avenue is therefore established to directly observe the impact of vehicle-terrain interactions on mobility. A chief aim of this work is to study the rover's capacity to react to terrain variations and to provide a preliminary assessment of the sensitivity to which these variations may be estimated proprioceptively. Monitoring the yaw dynamics of a SHREW enables a path for estimation of the net traction under each wheel individually. This paper presents a digital model of the SHREW developed in a ROS-oriented Gazebo simulation to assess the potential of using the robot as a testbed for observing vehicle-terrain interactions.

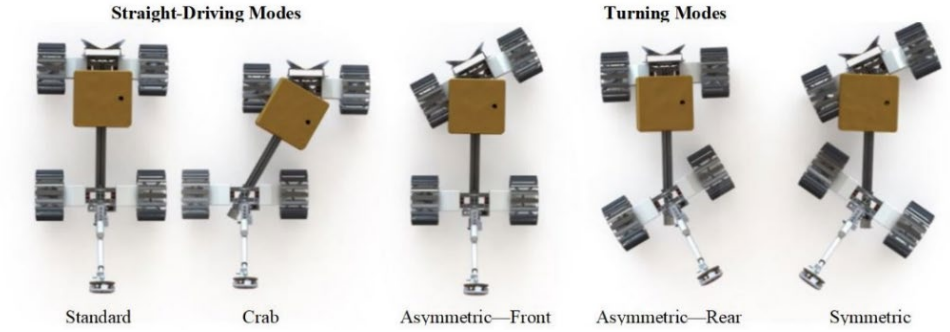

Fig. 2. *Named pose types of an individual SHREW*

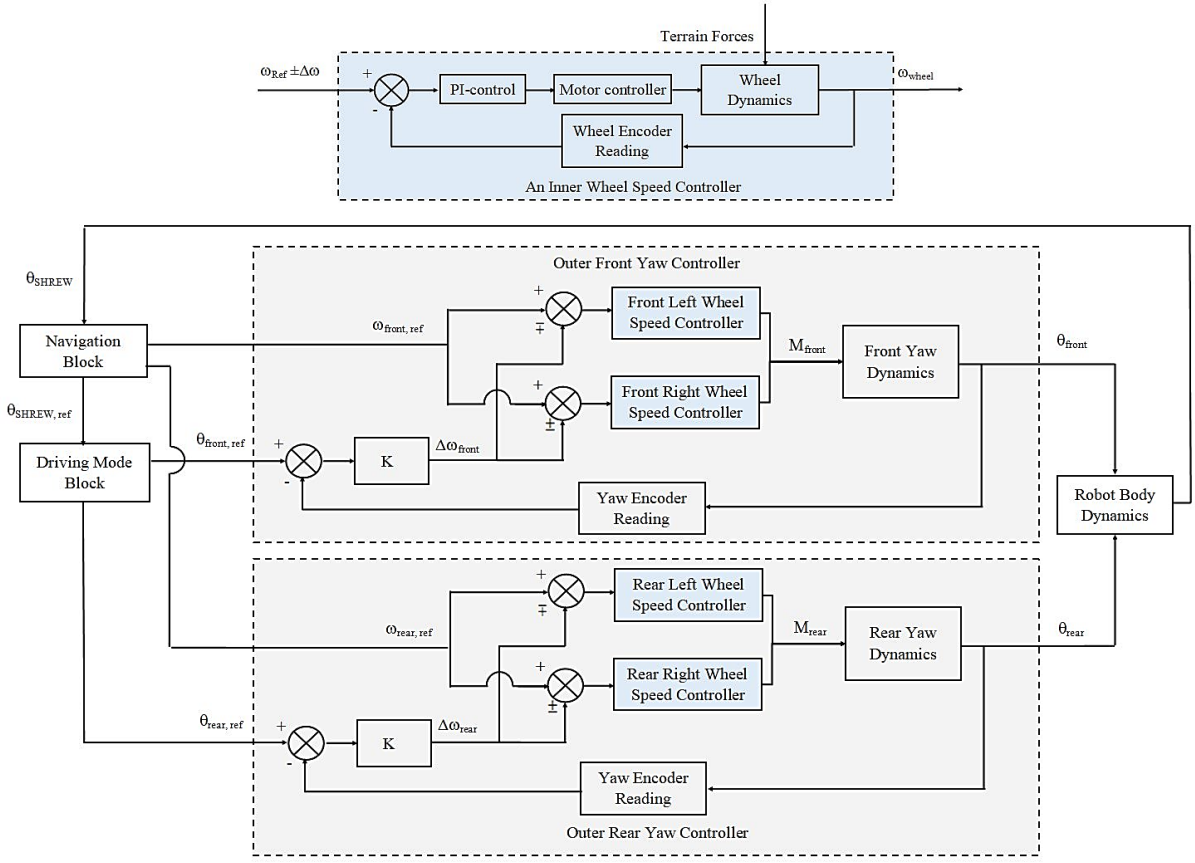

Fig. 3. *Cascading control architecture for speed and body heading control.* ***(top)*** *Inner proportional integral wheel speed control loop.* ***(bottom)*** *Outer proportional control to maintain yaw angle of each axis and heading angle of the overall vehicle.*

2. Background

Models of wheel interaction with terrain that are rooted in Bekker terramechanics theory, which applies to a vehicle moving longitudinally over terrain at constant speed compacting terrain vertically as it moves, are arduous to implement in real time. However, a simplified model comprised of a *net* force or drawbar pull resulting from such a model can be implemented in Gazebo. Figure 4 shows the applied torque T and normal load W on a driven, rigid wheel moving with longitudinal velocity V. The loads induce sinkage z, normal stress distribution σ, and shear stress distribution τ along the contact patch with the terrain. Net forces develop at the vehicle-terrain interface from stress distributions, named the *drawbar pull* or net traction force F_x and resistance torque T_r. These loads are the resultant of integrals of components of $\sigma(\theta)$ and $\tau(\theta)$ over the contact patch. Bekker theory relates normal stress to sinkage z through terrain stiffness constants k_c, k_ϕ, power n, and wheel width b using the empirical Bernstein-Goriatchkin power-law $\sigma(z) = (k_c/b + k_\phi)z^n$. The Mohr-Coulomb shear stress-shear displacement relationship governs shear stress, adding two more parameters (Wong, 2001). Wong and Reece (1967a) parameterize the location θ_{max} of $\sigma_{\max}$ along the contact patch as $\theta_{max} = (c_1+c_2 s)\ \theta_1$ (where s is wheel slip) for a total of eight parameters in this terrain model. The normal stress $\sigma(z)$ is transformed to a function of θ to integrate over the contact patch, and at constant velocity static equilibrium conditions hold enabling derivation of terrain forces from pressure distributions (Wong & Reece, 1967a).

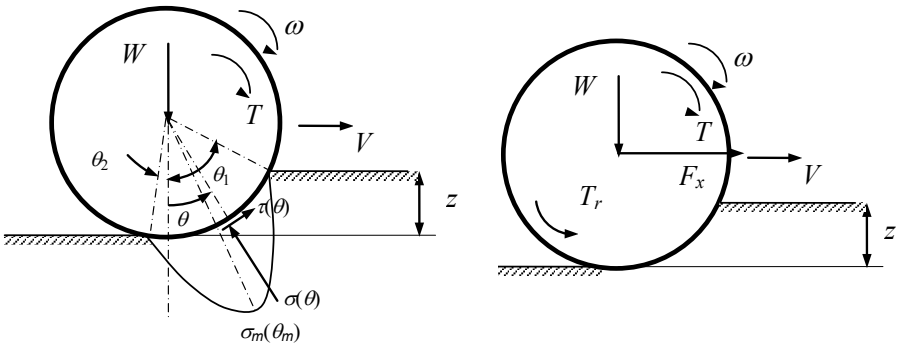

***Fig. 4**. Applied forces and moments on a driven, rigid wheel in deformable terrain and (left) resulting stresses, (right) net result of vehicle-terrain interaction represented as net traction force or drawbar pull F_x and resistive torque T_r.*

Bekker terramechanics theory assumes that all terrain resistance of a rigid track or wheel moving at a constant speed is due to vertical compaction of deformable terrain, and the resistive torque and resultant tractive effort required to overcome terrain resistance are calculated at the static equilibrium condition. The models of stress distribution are asymmetric, and the limits of integration are found iteratively to balance the supported load (Garber & Wong 1981, van Wyk et al. 1996). The contact patch is difficult to define, and often the lower extent θ_2 is taken as 0 degrees from vertical in this iterative solution, an assumption that breaks down as a vehicle's tractive elements excavate friable terrain under high wheel slip. The required iteration of a set of integral equations makes the model intractable for real-time control or for implementation in simulation software such as Gazebo. Most important, the model does not replicate the behavior of lightweight vehicles on low cohesion terrain observed *in situ*, which is attributed to normal statistical variation of terrain response, abrupt changes in terrain properties, and dynamic effects, such as increased terrain resistance when wheels slip, excavating friable terrain and sinking further than the static model predicts.

Despite these shortcomings, Bekker terramechanics forms a foundation from which others have developed alternate pressure-sinkage relationships and associated drawbar pull vs. slip models that can be used for simulation studies. However, without *in situ* capabilities for evaluating such models, or extension to coupled terrain-vehicle dynamics, they are limited in their ability to predict incipient immobilization or for use in controlling a vehicle to avoid immobilization. To address the need for designing vehicles capable of acquiring *in situ* data, one may turn to dynamic modelling programs in order to evaluate the ability of a testbed to elucidate coupled vehicle-terrain interactions with terrain models that incorporate simplified traction and resistance force vs. slip relationships. This approach is taken in the current work.

An assortment of programs are available for multi-body simulation (MBS). In general, software can be classified into one of two categories. The first type represents physical joints as mathematical constraints and treats contact forces as either unilateral or bilateral constraints on the bodies involved. These contact constraints are then added to the same solver used to find global system forces. The second type models joints as part of the robot itself and then parameterizes their effects in the rigid-body dynamics. When the simulation is advanced forward, joint constraints are solved separately from

contact constraints, with the intention of reducing computing time. Resolving constraints into reaction forces is largely based on solving a Linear Complementarity Problem (LCP). LCPs are a key class of problems in optimization and multi-body simulation that can be equivalently represented as quadratic programs (L**ötstedt, 1982;** Baraff, 1994) and often leads to greatly simplifying contacts to increase the speed of simulation (Drumwright and Shell, 2012). However, for modelling of wheel-terrain interactions this simplification results in a homogenous driving surface that is not informed by terramechanics theory, as deformation is not included. Many programs make use of the same physics engines. For example, the Open Dynamics Engine (ODE) is used by Gazebo, MORSE, iCubSim, and many others. This suggests that programs are largely distinguished by their features and computation times instead of the modelling techniques employed. MSC.Adams provides an optional plugin to include the basic Bekker-model formulation, although it can be prohibitively costly for university research. Trease et al. (2011) designed the Adams-based Rover Terramechanics and Mobility Interaction Simulator (ARTEMIS), an extension to Adams that includes Bekker-Wong terrain formulation created to study the Mars Exploration Rover. In it, they implement a subroutine to model slip, sinkage, and deformation; however the model of the terrain is still homogenous in its parameters.

Given the preceding discussion, Gazebo, the simulator selected for use in this work, was chosen for accessibility as an open-source platform, the available user support, and the ability to model heterogenous terrain resistance, even if simplified.

3. Modelling Overview

This study uses ROS Noetic and Gazebo-11 for developing and simulating the dynamics of a SHREW. The Robot Operating System (ROS) is a meta-operating system that provides an interface to develop and manage the functionality of a robot. It is comprised of a peer-to-peer network over which incoming and outgoing data may be passed asynchronously between processes linked by-way-of TCP/IP socket connections. This allows ROS to be used as middleware between the software and a physical robot, a virtual abstraction of the same robot, or a collection of robots (Romero, 2014).

A variety of libraries are available for control, visualization, simulation, message passing, logging, and other functions. In particular, Gazebo is high-fidelity multi-body simulator that interfaces with ROS via socket connection. It provides robust physics engines, graphical rendering, hardware abstraction, and software plugins which enable real-time modification of simulations. Its advantages over other simulators are realized in the development timeline, as these utilities reduce the infrastructure needed for a custom simulation. Correspondingly, it is often sufficient to construct a basic model of the robotic platform of interest and the scripts that dictate its movement in the environment, reserving low-level control, sensor models, localization, and entities like obstacles to the plumbing of Gazebo and ROS. This abstracts away much of the complexity of the simulation to be managed entirely by open-source tools. What follows is a technical description of this simulation route.

Model files, typically defined in the Unified Robot Description Format (URDF) or the Simulation Description Format (SDF), are comprised of rigid bodies known as links and their constraints known as joints. Links are defined by an inertia matrix, mass, and visual and collision geometries that may or may not be the same. Joints are defined by a type, e.g., prismatic, revolute, universal, screw, or ball; a parent/child relationship between two links; and the limits of the joint. A variety of configuration parameters are set to signal to the physics engine how to manage the behavior of a link and its contacts with other models or to manage the constraint between two links within a single model.

The accuracy of a simulation is highly dependent the physics engine selected and on these parameters, which are largely agnostic of the engine in use – but not entirely. Gazebo currently supports four different physics engines. The current work makes used of ODE. In this, integration of rigid bodies involves solving a modified Linear Complementarity Problem to determine how joint constraints are realized as reaction forces in simulation. The constraint equation is

$$J * v = c + CFM * \lambda \tag{1}$$

where v is the vector of body velocities, J is a contact Jacobian matrix reflecting element masses and contact geometries of the bodies, c is the joint velocity or "right-hand side vector", CFM is a constraint-force mixing term, and λ is a restoring force. λ is determined by

$$\lambda = \frac{c}{h\, JM^{-1}J^{T} + CFM} \tag{2}$$

where h is the step size and M is a joint-space inertia matrix. The CFM tag may be set to limit the extent to which joint constraints are enforced, with the intention of modelling spongy or springy interactions (Smith, 2019). It adds a small positive value to the system matrix in the LCP, while another tag, the error reduction parameter (ERP), specifies the error which will be maintained in the constraint during the next simulation step. These tags are related by

$$ERP = \frac{hk_p}{hk_p + k_d} \tag{3}$$

$$CFM = \frac{1}{hk_p + k_d} \tag{4}$$

where k_p and k_d are the stiffness and damping parameters of ODE (they are not standard coefficients). Contacts between separate bodies are determined by creating a virtual joint which includes friction between elements; three tags, *mu1*, *mu2,* and *fdir1* define this. The first two of these name "Coulomb friction" coefficients in perpendicular directions, and *fdir1* names the direction of *mu1* (Fig. 5, left). When two surfaces are in contact, the one with the lower friction coefficient is used in computation.

The Coulomb friction terms are described in the ODE manual as a "tangential anti-slipping force at the contact point if the contact normal force is nonzero," which is an important distinction. The Open Dynamics Engine user manual describes unrealistic behavior that may occur during rolling contacts such as a wheel passing over rough surfaces (Fig. 5, right). In the face of two contact joints appearing beneath a moving, curved link, if no slip is allowed at either contact, the center of the curve is required to simultaneously move in arcs 1 and 2, as shown in the figure. Consequently, if a constant-force approximation is used, the moving link will stop instantaneously, because the normal force is independent of tangential friction forces. ODE resolves this by approximating a "friction cone" as a pyramid that is used to determine when a contact should stick or slide. In this example, contact 1 will have zero normal force in the pyramid model. Thus, no force will be applied at this location. The approach substitutes undesirable stopping behavior for a modified Coulomb friction model that stabilizes the simulation, but still leads to non-physical behavior (Smith, 2019). A similar cone exists for tags related to slip. Resolving the LCP under contact and the algorithmic approach taken is described in greater detail in Baraff (1994) on which ODE's model was based.

In Gazebo, setting appropriate dynamic parameters is notoriously difficult, because each physics engine requires different parameters; even if a parameter is accepted by multiple physics engines, it may be used in computation differently. Likewise, because they are used to approximate physical interaction in real-time, many of the parameters use nonstandard units, or they are named based on what they impact in simulation instead of what they actually represent in computation, as is the case with friction coefficients. For the CFM and ERP terms specifically, a full matrix is determined by specifying a single parameter leading to homogeneity for a single link. These challenges lead developers to prioritize the performance of a robot against *some* terrain instead of prioritizing the accuracy of the terrain model on its own and then modifying the robot model until it performs as desired. Fortunately, the default parameters of the ground plane inherent to Gazebo are well tuned to mimic a uniform driving surface like concrete. This makes Gazebo well suited to simulate simple mobile robots used primarily in-doors. However, representing physical, off-road terrain and the non-rigid dynamics of wheel-terrain interfaces to high-fidelity is challenging in Gazebo and ODE.

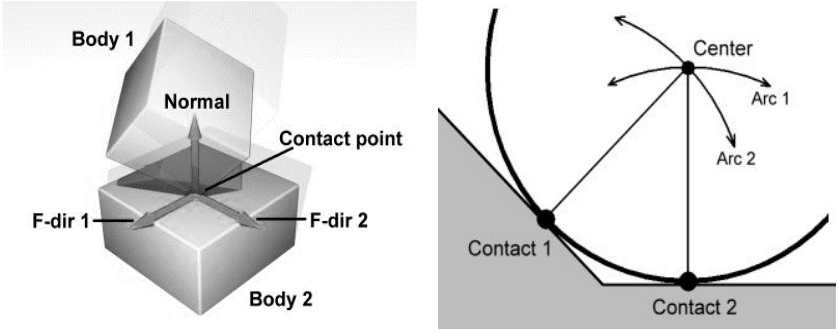

Fig. 5. (left) *A contact joint as modelled in the Open Dynamics Engine.* ***(right)*** *Rolling contact that gives rise to the friction pyramid approximation that is used by Gazebo default dynamics engine, ODE (Smith, 2019).*

A common approach in dynamic modelling is to use a digital elevation model (DEM) to form a terrain that is, at the very least, heterogenous in surface elevation. In Gazebo, this approach still requires the user to tune the constraint parameters described, and because the DEM is treated as a single link, variation in resistance forces on flat surfaces are

not captured. A second approach one may consider is to represent a granular soil as a collection of spherical bodies loosely attached by joints, which forms a joint group. Utilizing ODE in this way requires computation to be distributed over several machines because the number of joints in each group (m_1), the number of bodies (n), and the number of DOF constrained by the joints (m_2) increase computing time by $k_1\, O(m_1) + k_2\, O(n) + \ k_2 O({m_2}^3)$. Each constrained DOF adds a row and column to the system matrix in the LCP problem, which leads to the right-most term (Smith, 2019). Clearly ODE, like many other physics engines, prioritizes robust real-time rate and computing time and gives only passing attention to terrain models. A reduced order approach is taken in this work, described in the remainder of this section.

A custom simulation environment is created to assess vehicle performance over a variety of flat surfaces. Figure 6 illustrates how the simulation environment is conceptualized in comparison to real world operation. Three elements define the interaction between a SHREW and the terrain: a vehicle model, terrain or a terrain model, and a wheel-terrain interface. The SHREW model is realized in a URDF where parts like wheels and axles are defined as links and make use of Solidworks part files to form their visual and/or contact meshes. Link parameters, namely the wheel friction coefficients, are set high enough such that contact forces are only a function of the terrain parameters. Parts like motors and bearings are modelled as joints. In the case of the joints between the wheels and axles, a damping tag is set to provide a dynamic response which mirrors the motors of the physical SHREW.

In contrast to the physical terrain, the properties of the virtual terrain are known (to the extent possible under the limitations of Gazebo) as they are defined directly in a second URDF describing the terrain. A grid is created by attaching several planar links in a fixed joint chain (Fig. 7). The center link is additionally attached to the ground plane of the simulation. This comprises a terrain made of nine bodies (n=9) and 10 joints (m_1=10) constraining 60 degrees of freedom (m_2=60). Excluding randomly materializing contact joints between the SHREW and terrain, the total computation time needed to processes the terrain each simulation iteration is only $k_1O(10)+k_2O(216,009)$. Blocks are made distinct from each other in their friction and slip coefficients.

Here, a simplified test is used to evaluate a SHREW traversing a heterogenous surface. It entails a SHREW beginning entirely in a flat terrain cell, such that all wheels are subject to the same resistive force (Fig. 7). This cell is defined by a friction coefficient of 0.6 and a slip coefficient of 0.00012, meant to mimic a clayey loam surface. A goal location directly in front of the vehicle is published and the vehicle begins to advance to this point using symmetric steering to maintain a straight trajectory. Between the starting location and the goal is a pair of terrain cells, one identical to the starting cell and the other with a friction coefficient of 0.1 and a slip coefficient of 0.1. This is designed to lead to imbalanced forces from left to right after the robot has reached a steady state velocity. Note that the value of these parameters should not be interpreted physically. The key point is that the starting cell mimics a high traction surface and the cell inducing an imbalance mimics a low traction surface.

The virtual terrain and SHREW models are realized in the simulation environment when loaded onto the core Gazebo server used to simulate the physics of the simulation and are made visual by Gazebo graphical client. The interaction between wheels and the terrain exists entirely on the Gazebo server and state measurements are made via Gazebo's API. Some of these measurements are passed back to a custom Gazebo plugin and some to ROS-oriented scripts that control the robot. The Gazebo plugin and a custom API are used to modify the simulation model in real time. Implementation processes utilize ROS services and topic management to pass information, like wheel speed commands, to the model plugin and back to Gazebo. A simplified diagram characterizing the flow of messages on the ROS side is included in Fig. 8. ROS topics are represented by blocks and grouped processes are represented by nodes. Arrows exiting topics represent a subscription maintained by the process it is connected to, whereas incoming arrows represent a publisher sending information on the named topic. The cascading control architecture shown in Fig. 3 is implemented as follows: the Gazebo API passes information to the /gazebo/model_states topic, which is then utilized by the custom Gazebo plugin (the /gazebo node) to publish the state of all joints to the /joint_states topic. Navigation is managed on the /shrew_mover node, which receives the joint states through a subscription. The joint data is processed to determine an appropriate reference angular rate for each wheel and new wheel speed commands are published to the /shrew_robot/wheels/cmd_vel topic. The Gazebo plugin then receives the command through its own subscription to the named topic and finally passes it along to Gazebo's API to modify the wheel speeds with a built-in PI-controller.

Transforms between a base coordinate frame of the terrain and the SHREW are published by the /full_sim_world_tf2_broadcaster process. Similarly, the /robot_state_publisher broadcasts a set of coordinate frame transformations between all links in the SHREW model. Lastly, the /shrew_static_tf2_broadcaster publishes a static transformation to connect the transformation tree of the simulated environment to the transformation tree of the robot, such that all transforms can be aggregated by the ROS-based /tf service, which keeps track of the changes in coordinate frames. This allows other ROS services used by the simulation environment to easily transform data into any coordinate frame in the connected tree.

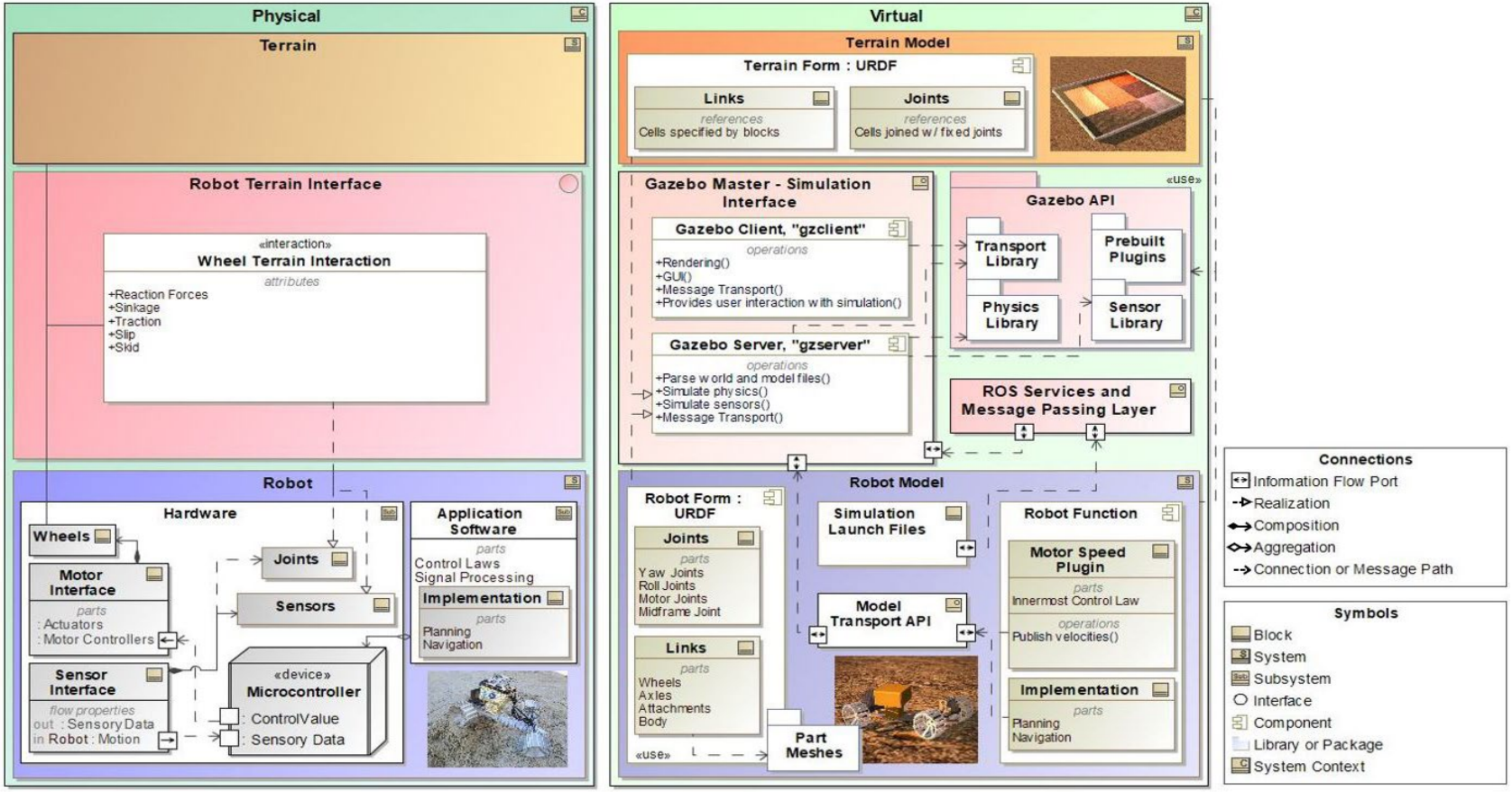

Fig. 6. *Model structure of the simulation environment in comparison to the operation of a SHREW in the real world.*

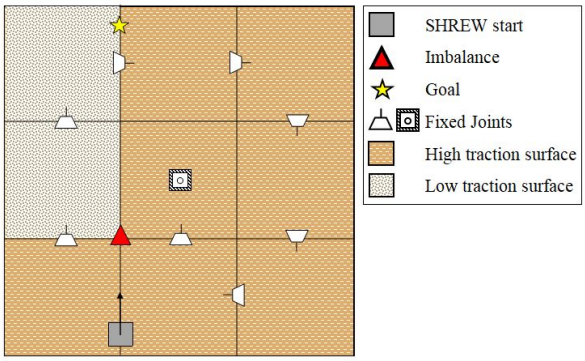

Fig. 7. *Terrain grid comprised of nine bodies, 10 joints, and 60 constrained DOF. The test case path over mixed surfaces is shown.*

Fig. 8. *Simplified topic subscription and publisher diagram*

A second robot model was created to act as a baseline for comparison. This is defined in a third URDF. It is identical to the flexible SHREW robot in that it makes use of the same Solidworks part files; has the same mass properties; and uses the same friction, damping, and slip parameters, however its roll and yaw joints are fixed and thus the model is a rigid robot with differential steering through left and right wheel torques. Its operation is managed by its own set of plugins and implementation files which are modified to control the robot by differential steering. The intention of the simulation is to show the difference in response of the flexible SHREW and the rigid robot when the terrain transitions from high to low traction surfaces.

4. Results

A single run is made for both the flexible and the rigid version of the robot, and wheel speed, torque, contact locations, and slip are logged during simulation. Front and rear yaw joint angles and front and rear yaw joint angular rates are logged during the simulation of the flexible SHREW. The heading angle and yaw rate of the rigid robot is logged during its run, as determined via the ROS tf service. An additional set of runs are made, varying the proportional gain of the outer yaw controller of the flexible SHREW to assess the effects of a more rigid or compliant vehicle. The results of simulation are shown in Figs. 9 through 11. Figure 9 shows the velocities, applied torque, and slip of each wheel, heading, and yaw rate for the traverse of the rigid SHREW, sampled at 1kHz. Figure 10 shows the wheel state data and yaw joint angles and rates for the non-rigid SHREW, also sampled at a rate of 1kHz. In both, the vertical lines represent the moment the front end and rear end transitioned from the high traction surface to mixed, respectively. The right wheels transition to the lower friction surface, while the left wheels remain on a high friction surface.

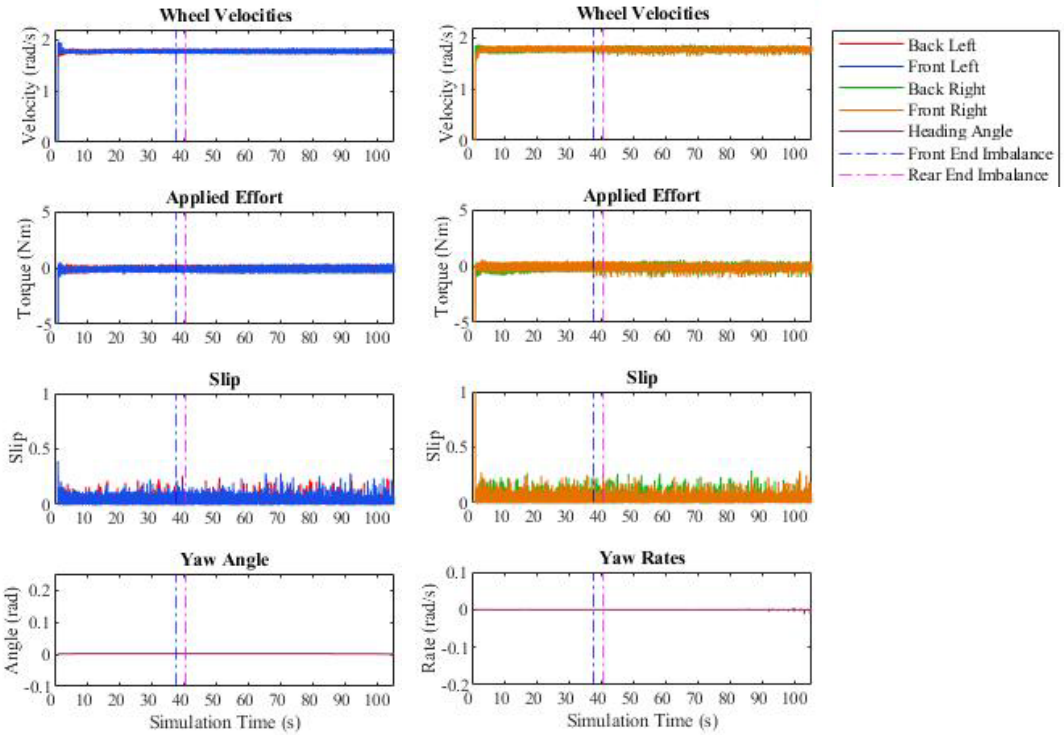

Fig. 9. *Effects of imbalanced terrain forces in rigid SHREW robot.*

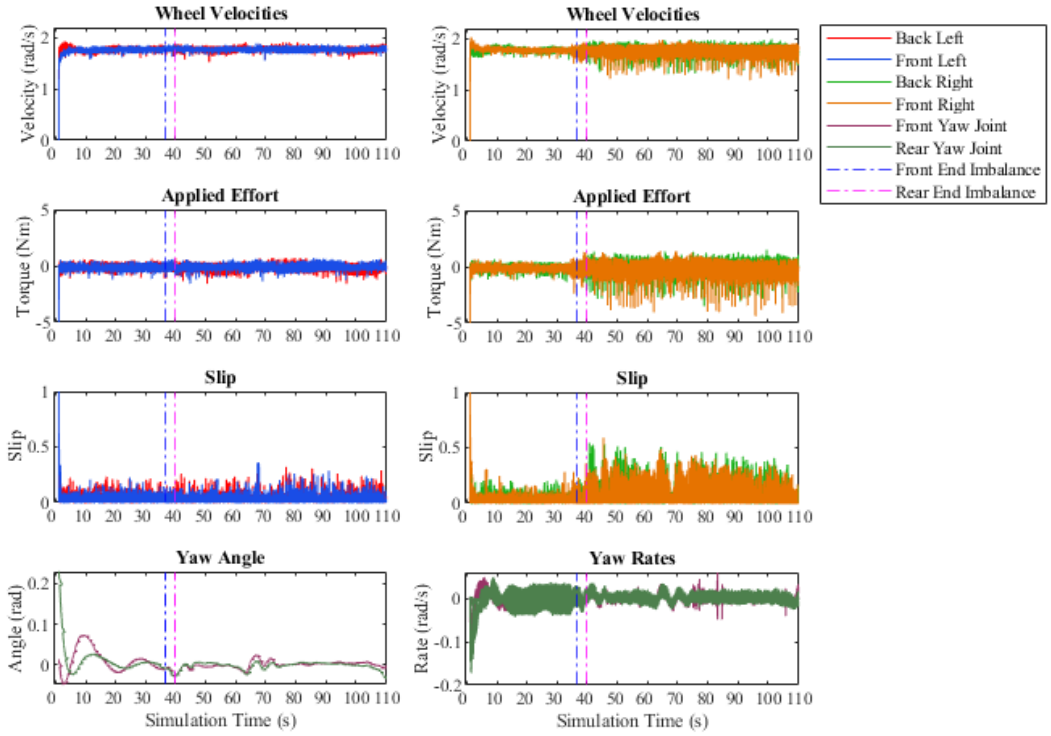

Fig. 10. *Effects of imbalanced terrain forces in non-rigid SHREW robot.*

As can be seen in Fig. 10, imbalanced forces are more apparent in a SHREW with non-rigid yaw joints than with a rigid vehicle. As would be expected, when the flexible SHREW encounters an imbalance in traction forces generated from left to right sides, the right-side wheel torque increases to rebalance the forces, and the right-side velocities also increase as does wheel slip. The terrain transition is apparent in these state variables. In comparison, any change in applied torque, wheel speeds, wheel slip, and overall yaw rate of the robot is negligible for the rigid robot.

Excluding startup and shutdown noise, the velocity and torque data is approximated by a Gaussian distribution. For both the rigid and non-rigid runs, the front right and back right wheel data is aggregated into a right-side set. These are then broken into samples prior to meeting imbalanced forces, and samples after imbalanced forces. The means and one standard deviation are reported in Table 1 indicating the changes observed in Fig. 10 for the non-rigid SHREW and minimal impact of the transition from high to low friction surface on the rigid vehicle. Figure 10 additionally shows how deviation in the axle yaw angle of the non-rigid SHREW is apparent following imbalanced forces. For comparison, Fig. 9 includes the heading angle and yaw rate of the rigid SHREW during its traverse, which provide no discernable change as induced by the lower friction surface. These results clearly communicate the ability for a SHREW to sense changes in the terrain proprioceptively, as was a primary intent of this work.

Table 1. Simulation Statistics

Vehicle	Sample	Mean
Rigid SHREW	Right-side, Pre-Imbalance Vel.	1.765 ± 0.032 rad/s
	Right-side, Post-Imbalance Vel.	1.765 ± 0.030 rad/s
	Right-side, Pre-Imbalance Torque	-0.184 ± 0.224 Nm
	Right-side, Post-Imbalance Torque	-0.189 ± 0.216 Nm
Non-rigid SHREW	Right-side, Pre-Imbalance Vel.	1.766 ± 0.035 rad/s
	Right-side, Post-Imbalance Vel.	1.761 ± 0.091 rad/s
	Right-side, Pre-Imbalance Torque	-0.181 ± 0.211 Nm
	Right-side, Post-Imbalance Torque	-0.324 ± 0.542 Nm

Figure 11 demonstrates how the proportional gain used in the outer yaw controller (Fig. 3) is not only important for maintaining the heading angle of each axle of the flexible SHREW; it also bears substantial impact on the sensitivity to which imbalanced forces may be sensed. The front axle yaw joint state is shown for three different gains, and the data for each run are aligned in time such that the transition from high traction to low traction occurs at the same location in the plot. As the time response of the feedback loop decreases, the change in angle becomes less apparent. For our virtual model, a proportional gain of 1 is on the lower end of values which is strong enough to maintain the reference angle but with the largest settling time among the three gains. In contrast, a gain of 2.4 is strong for the model; a value much higher than this will overcompensate for yaw angle errors leading to overshoot. This causes undesirable oscillations as the maximum reference speed of one wheel will be reached while the other wheel will momentarily stop, thus applying a yaw moment that drives the axle past the reference yaw angle. While a value of 1 results in a more perceptible change in angle, the large settling time makes a tight turn more difficult. On the other end of the spectrum, a value of 2.4 results in a smaller setting time as is shown prior to meeting mixed resistance forces or a more rigid vehicle obtained through feedback control. This comes at the expense of less perceptible changes in yaw angle. This demonstrates that proprioceptive sensing of imbalanced forces can be enhanced through controlling the vehicle yaw stiffness with a proportional control loop. Empirically, a gain of 2 provides a sufficient medium that does not lead to noticeably worse performance than a higher gain (except as start-up) and preserves the sensing capacity found in lower gains.

5. Conclusion

SHREWs can follow a trajectory over soft, heterogenous terrain by using the low-level feedback architecture described to monitor variations in terrain resistance from left-to right and react by adjusting wheel speeds to maintain a desired pose. The Ackerman-type steering of SHREWs, enabled by unconstrained yaw joints, is controllable and asymptotically stable about reference trajectories. While the simulation environment of Gazebo poorly reflects the complexities of soil mechanics, and thus wheel-terrain interactions, it provides evidence that monitoring individual axle responses to imbalanced forces caused by terrain heterogeneity elicits this heterogeneity in the response of proprioceptive sensors,

such as rate gyros, wheel speed sensors, and motor torques. Additional investigation must be made to understand how imbalanced resistances materialize in the data of physical SHREWs and future work should entail the construction of algorithms or heuristics that formalize a relationship between proprioceptive data observed and terramechanics of lightweight vehicles.

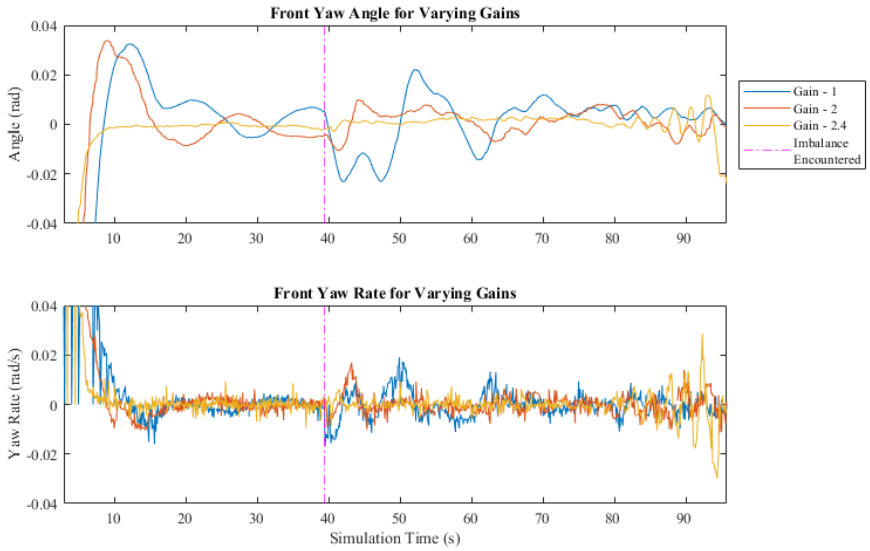

Fig. 11. *Impact of varying the proportional gain in the yaw angle controller shown in Fig. 3.*

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1824687 and NASA Grant No. 0NSSC20M005.

References

Baraff, D., 1994. Fast Contact Force Computation for Nonpenetrating Rigid Bodies. Computer Graphics Proc. Annual Conference Series, Orlando. pp. 23-34.

Bauer, R., Leung, W., Barfoot, T., 2005. Experimental and simulation results of wheel-soil interaction for planetary rovers. IEEE/RSJ International Conference on Intelligent Robots and Systems., pp. 586-591.

Bekker, M. G., 1962. Theory of Land Locomotion: The Mechanics of Vehicle Mobility. University of Michigan Press, Ann Arbor, Michigan.

Bekker, M. G., 1969. Introduction to Terrain-vehicle Systems. University of Michigan Press, Ann Arbor, Michigan.

Drumwright, E., Shell, D. A., 2012. Extensive Analysis of Linear Complementarity Problem (LCP) Solver Performance on Randomly Generated Rigid Body Contact Problems. IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5034-5039.

Garber, M., & Wong, J. Y. (1981). Prediction of ground pressure distribution under tracked vehicles—I. An analytical method for predicting ground pressure distribution. Journal of Terramechanics, 18(1), 1-23.

Irani, R. A., Bauer, R. J., Warkentin, A., 2010. Modelling a Single-Wheel Testbed for Planetary Rover Applications. ASME Dynamic Systems and Control Conference. 1. pp. 181-188.

Jayakumar, P., Melanz, D., Maclennan, J., Gorsich, D., Senatore, C., Iagnemma, K., 2014. Scalability of classical terramechanics models for lightweight vehicle applications incorporating stochastic modeling and uncertainty propagation. Journal of Terramechanics. 54. pp. 37-57.

Lever, J., Denton, D., Phetteplace, G. E., Wood, S. D., Shoop, S. A., 2006. Mobility of a lightweight tracked robot over deep snow. Journal of Terramechanics. 43 (4). pp. 527-551.

Lines. A., Elliot, J., Ray, L., 2021. Rigid Wheel Design and Evaluation for Lightweight Snow Rover. Proc. of the ISTVS 20th International Conference, forthcoming.

Löstedt, P., 1982. SIAM Journal on Applied Mathematics. 42 (2). pp. 281-296.

Meirion-Griffith, G., Spenko, M., 2011. A modified pressure-sinkage model for small, rigid wheels on deformable terrains. Journal of Terramechanics. 48. pp. 149-155.

Ray. L. E., 2009. Estimation of Terrain Forces and Parameters for Rigid-Wheeled Vehicles. IEEE Transactions on Robotics. Vol. 25 (3). pp. 717-726.

Romero, A. M., 2014. ROS Concepts. https://wiki.ros.org/ROS/Concepts.

Smith, R. L. Open Dynamics Engine Manual, 2019. http://ode.org/wiki/index.php?title=Manual

Trease, B., Arvidson, R., Linderman, R., Bennett, K., Zhou, F., Iagnemma, K. Senatore, C., Van Dyke, L., 2011. Dynamic Modeling and Soil Mechanics for Path Planning of the Mars Exploration Rovers. Proc. of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. pp. 1-11.

Van Wyk, D. J., Spoelstra, J., & De Klerk, J. H. (1996). Mathematical modelling of the interaction between a tracked vehicle and the terrain. Applied mathematical modelling, 20(11), 838-846.

Wong, J. Y., 2001. Theory of Ground Vehicles third ed. John Wiley and Sons Inc., New York, New York.

Wong, J. Y., Reece, A. R., 1967a. Prediction of Rigid Wheel Performance Based on the Analysis of Soil-wheel Stresses: Part I. Performance of Driven Rigid Wheels. Journal of Terramechanics. 4 (1). pp. 81-98.

Wong, J. Y., Reece, A. R., 1967b. Prediction of Rigid Wheel Performance Based on the Analysis of Soil-wheel Stresses: Part II. Performance of Towed Rigid Wheels. Journal of Terramechanics. 4 (2). pp. 7-25.