

MOBILITY MODES AND CONTROL OF A PASSIVELY-ARTICULATED MULTI-SEGMENT WHEELED VEHICLE

Joshua Elliott^a, Austin Lines^b, and Laura Ray^a

^a Thayer School of Engineering at Dartmouth College, 14 Engineering Dr, Hanover, NH 03755

^b Cold Regions Research and Engineering Laboratory, 72 Lyme Road, Hanover, NH 03755

joshuajameselliott@gmail.com, austin.lines@gmail.com, laura.e.ray@dartmouth.edu

Abstract

This paper presents mobility modes and control methods for the *SnoWorm*, a passively-articulated multi-segment autonomous wheeled vehicle concept for use in Earth's polar regions. *SnoWorm* is based on *FrostyBoy*, a four-wheeled GPS guided rover built for autonomous surveys across ice sheets. Data collected from *FrostyBoy* were used to ground-truth a ROS/Gazebo model of vehicle-terrain interaction for simulations on snow surfaces. The first mobility mode, inchworm movement, uses active prismatic joints that link the *SnoWorm*'s segments, and allow them to push and pull one another. This pushing and pulling of individual segments can be coordinated to allow forward motion through terrain that would immobilize a single-segment vehicle. The second mobility mode utilizes fixed links between *SnoWorm*'s segments and uses the tension or compression measured in these links as a variable to control wheel speeds and achieve a targeted force distributions within the multi-segment vehicle. This ability to control force distribution can be used to distribute a towed load evenly across the entire *SnoWorm*. Alternatively, the proportion of the load carried by individual segments can be increased or decreased as needed based on each segment's available drawbar pull or wheel slip.

Keywords: Multi-Segment, Simulation, Polar Robot, Wheeled Vehicle, Inchworm

1. Introduction

Uncrewed ground vehicles (UGVs) have been used to collect a wide range of scientific data in the polar regions and beyond Earth. Typically wheeled or tracked, UGVs come in a wide range of sizes and masses, from the 24 kg *SnoBot* (Lever et al. 2006), to NASA's one ton *Perseverance* rover. The four wheeled, 70 kg *Yeti*, has used ground penetrating radar (GPR) to peer into ice sheets and survey crevasses (Arcone et al. 2016, Lever et al. 2013). *Yeti*'s suspension was designed to negotiate sastrugi, a pattern in wind-scoured snow that creates linear features similar to frozen ocean waves. The 750 kg *Lunokhod-1* was the first UGV used beyond Earth when it landed on the Moon as part of the Soviet space program (Pyle 2019), and it carried an array of scientific instruments (Kassel 1970). *Lunokhod-1* was able to explosively disconnect any of its eight wheels in case of damage or immobilization. A major limitation and driving element of the design of these vehicles is off-road mobility. The promise of UGVs lies in their ability to make measurements over a spatial-temporal domain. Mobility is therefore an overriding consideration that other decisions must work around.

The 2004 Mars Exploration Rovers (MERs) had drastically different mission durations as a result of immobilization. *Opportunity* operated until 2018 before losing communication during a dust storm. *Spirit* lost communication in 2010 after an immobilization in 2009 forced it to endure the Martian winter without its solar panels pointed in the optimal direction. If *Spirit* had not become immobilized, it could have provided close to another decade of science data. However, had *Spirit* been designed to make the probability of that immobilization near zero, it would have required compromises to other systems rendering the extra mission time less valuable. *Curiosity*'s ongoing operations must balance the risk of damage to its wheels from sharp rocks, with the risk of immobilization in deep sand. Mission planners select routes that straddle sandy and rocky areas, often resulting in routes between areas of interest that are far from direct (Arvidson et al. 2017). *Curiosity* devotes much of its limited mass to scientific instruments, and its mobility system exists in service to those instruments.

For all of these vehicles, the mass that can be devoted to their mobility systems must be weighed against other mission needs. A Mars rover could be designed to reduce the probability of immobilization relative to existing UGVs, but would have less payload for sensors - its *raison d'être*. Research has shown that increasing a wheel's width and radius results in lower ground pressure, lower sinkage, more traction, and a general decrease in the probability of immobilizations (Lines 2020, Lines et al. 2021). However, these changes require changes to other vehicle systems that further increase mass and negate some of their benefits. For UGVs to be used more widely than they are now, they require mobility improvements that push beyond the simple scaling of wheels, or the lower efficiency of using tracks.

This paper presents *SnoWorm*, a vehicle concept for vastly improved UGV mobility on snow. Instead of relying on larger wheels or tracks for improved mobility, it is a composite vehicle made of multiple linked segments. Each segment is similar in scale to the *FrostyBoy* UGV on which it is based, but with active prismatic joints embedded in a connector link between segments. These prismatic joints allow segments to push and pull one another, or to drive in unison. *SnoWorm* has been simulated with up to six segments, with no known limitations preventing a higher number of segments. Turning radius is based on the angular limits of the axles and connector links between segments rather than the total number of segments. However, large numbers of segments prevent turns of greater than 180° without the vehicle striking itself. Figure 1 shows a three segment *SnoWorm* rendered within the Gazebo simulation environment, which will be discussed in the next section.

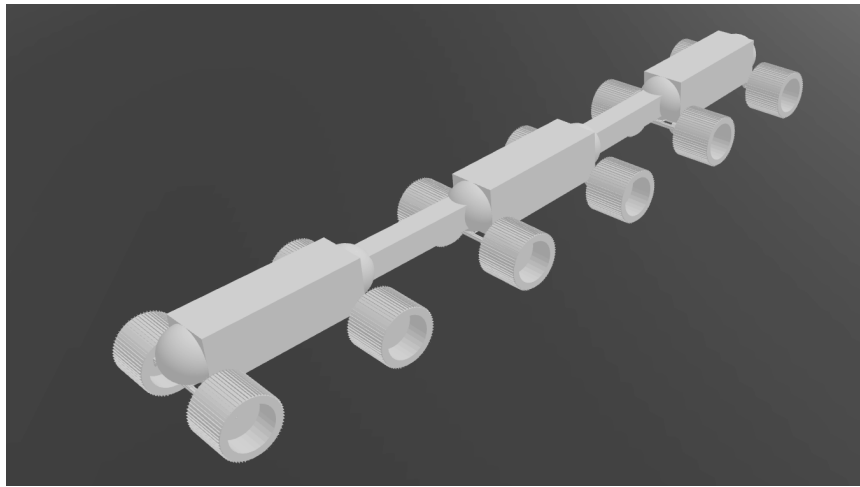


Figure 1: The vehicle geometry, dubbed *SnoWorm*, used in the remainder of this paper.

2. Modeling and Simulation

The combination of Gazebo and Robot Operating System (ROS) was chosen for the simulation environment and software hierarchy respectively for this work. Gazebo and ROS are widely used tools in the robotics research community that enable rapid software-in-the-loop testing, and make it easier to reuse software written for one robot on another. When considering this combined system, the work by Aucone (2019) was used as an example of how ROS and Gazebo support algorithm development for implementation on physical systems. Aucone developed motion planning algorithms for the *DuAxel* rover by working with its digital twin inside a Gazebo simulation. Those algorithms, written within ROS, were then loaded onto the physical *DuAxel*, where testing closely matched the Gazebo simulations.

Gazebo is a 3D robotics simulation environment that integrates a multibody physics simulation with rendering, sensors, actuators and more. Within Gazebo, motors and actuators can be placed on joints and simulated by manually or programmatically applying forces and torques, while various plugins can simulate sensors such as cameras and lidar. Gazebo plugins can also be used to execute custom code synchronously with each simulation timestep, and this work uses a plugin to calculate and apply forces and torques to the simulated vehicle.

Gazebo uses Open Dynamics Engine (ODE) as its default physics engine. ODE has two main parts - simulation of rigid body dynamics and collision detection. The rigid body dynamics simulator programmatically generates and integrates the equations of motion in three dimensions. The collision detection component determines when objects have collided and the resulting reaction forces. The custom terrain model used in this paper uses the normal force between the wheels and ground

plane to calculate the vehicle-terrain interaction forces and torques. ODE allows rigid bodies to deform relative to each other, but does not include sinkage in the typical terramechanical sense. This deformation is instead treated as material compression governed by a Young's modulus and damping coefficient.

Robots within Gazebo are defined by linking rigid bodies via passive or active joints to create complex multibody systems. Gazebo defines multibody systems via Unified Robotic Description Format (URDF) files where each link (rigid body) and joint is defined. Masses and moments of inertia are calculated externally and defined within the URDF, along with joint limits, damping coefficients, coefficients of friction, and many other variables. While typically constant during the simulation, link and joint properties can be modified while the simulation is running.

The links within Gazebo can be defined as either simple shapes or via STL file. The latter option allows for CAD based vehicle geometries to be imported directly into Gazebo. The visual and collision models of a link can be defined separately. This could mean using an STL for a wheel's visual model while its collision geometry is a simple cylinder of approximately the same dimension. Within this work simple geometric shapes are adopted for the sake of simulation speed.

Robot Operating System (ROS) is a middle layer between algorithms and robotic hardware. It is primarily a system for passing information between software nodes (individual programs) and an overall architecture for robotic software development. A ROS powered robot will contain a computer running ROS that will load each of the nodes that make up the software for the robot. As an example, one node might handle image processing from an on board camera and transmit the location of obstacles to other nodes, a second node might take the obstacle locations, then calculate and transmit the desired vehicle path, while a third node might calculate and publish commands to follow the desired path in a manner specific to the robot's locomotive system. If any of these nodes encounters an error and crashes, ROS will relaunch that node and allow the system to keep functioning without disruption.

ROS nodes communicate via user defined topics and services. Multiple nodes can publish and subscribe to the same topic by using a uniform topic name and message type. Gazebo also publishes a number of topics about the state of the simulation, so that ROS nodes can subscribe to those topics and understand the Gazebo system state. Within this paper, these Gazebo published topics act as sensor readings for the angles, speeds, and positions of joints and links within the simulation.

Gazebo's built-in method for defining friction between surfaces (known as the "friction cone" method (Smith 2006)) performs well when simulating rigid wheeled robots on flat artificial surfaces, but lacks the complexity required to simulate interactions with a deformable, variable surface such as snow. The inability to vary terrain properties based on location within the simulation is another major limitation. However, custom code in the form of plug-ins can be added to ROS/Gazebo to implement the needed changes.

3. Terrain Model

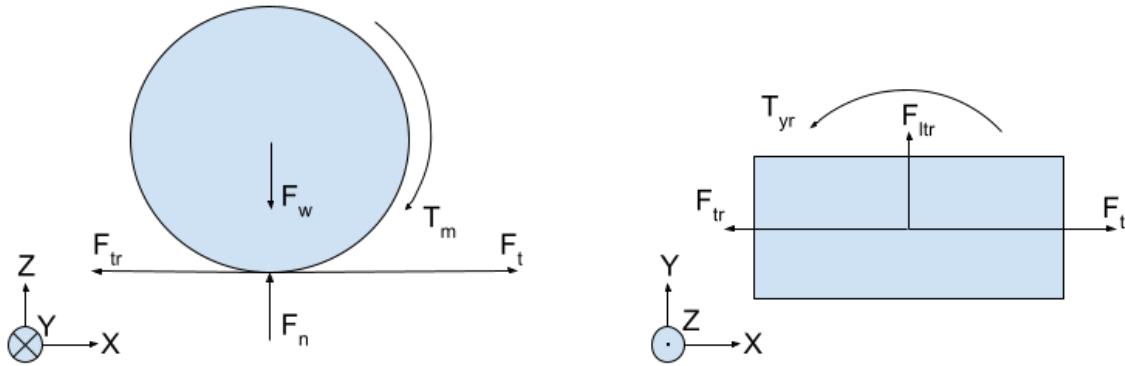
Because of the limitation with Gazebo's default friction model, this paper uses a custom terrain model to calculate and spatially vary the forces and torques resulting from vehicle-terrain interactions. The terrain model is a modified Coulomb friction model, where coefficients of friction change with wheel slip, and with a location dependent terrain coefficient in order to empirically match the variable nature of snow.

The Bekker-Wong model uses iterative integration to find the drawbar pull and resistive torque resulting from the vehicle terrain interaction based on an empirical normal pressure distribution at the contact patch and the resulting shear stress distribution. The Bekker-Wong model considers the effects of sinkage on the normal pressure distribution. The net result is a relationship between longitudinal force and slip between wheel and terrain. Rather than using Bekker-Wong to calculate this relationship, this paper uses a set of empirically derived curves based on the 'Magic Formula' (Pacejka and Bakker 1992), and ground-truthed against data collected from the physical *FrostyBoy*. While the Bekker-Wong model varies both tractive force and terrain resistance force with sinkage and slip, this custom terrain model varies only the tractive force with slip, and sinkage is not considered explicitly.

The terrain model used in this paper outputs the forces and torques described in Table 1 and shown in Fig. 2.

Table 1: Forces and Torques used in the custom terrain model

Variable	Name	Description
F_t	Tractive Force	Slip dependent tractive force
F_{tr}	Terrain Resistance Force	Resists longitudinal wheel movement
F_{ltr}	Lateral Terrain Resistance Force	Resists lateral wheel movement
F_w	Weight	Force pushing down from the wheel
F_n	Normal Force	Force pushing up from the ground
T_m	Motor Torque	Applied by the motor/gearbox
T_{yr}	Yaw Resistance Torque	Resists wheel yaw

*Figure 2: Forces and torques acting on a wheel in the x-z plane*

The custom terrain model takes as inputs six variables for each wheel. These variables include the longitudinal and lateral speed over ground, V_w and $V_{\perp}$, wheel rotational speed, ω_w , wheel/axle yaw rate, ω_z , normal force, F_n , and location dependent terrain coefficient, X_t . The weight of the wheel pushing down on the ground, F_w , is used internally by Gazebo. From these values, other variables are generated that empirically define the forces and torques resulting from wheel-terrain interactions.

$X_t = 1.0$ is meant to approximate névé or high-cohesion snow - terrain that provides the best mobility with a combination of high tractive force at low slip, low terrain resistance force, and low terrain resistance torque. As the terrain coefficient is decreased towards zero, the shape of the force-slip curve changes and the magnitude of the maximum force decreases, terrain resistance force increases, and terrain resistance torque increases. This lower terrain coefficient is meant to simulate loose, poorly-bonded snow, where a vehicle is prone to becoming immobilized from excessive wheel slip.

Wheel slip is defined here as

$$i = \frac{R_w \omega_w - V_w}{R_w \omega_w} \quad (1)$$

where V_w is the longitudinal ground speed, R_w the wheel radius, ω_w the wheel speed in radians per second, and i is the resulting slip.

A function directly mapping wheel slip to tractive or braking force is one way of calculating these forces. The empirical ‘Magic Formula’ (Pacejka and Bakker 1992), is used for this mapping between wheel slip and the resulting force. The Magic Formula is normally used for modeling the tractive (and braking) force between a rubber tire and asphalt road, but it can be tuned to represent this relationship for a wide variety of vehicle-terrain interaction:

$$F_t = X_t F_n b_{tF} \sin(2.8 X_t \operatorname{atan}(4 X_t^2 i - (4 X_t^2 i - \operatorname{atan}(4 X_t^2 i)))) \quad (2)$$

F_n is the normal force between wheel and terrain, b_{tF} is a unitless coefficient, and F_t is the resulting tractive force from the interaction.

The terrain model used in this paper does not directly consider terrain properties such as cohesion and friction angle, but the model can be tuned such that the force-slip curves match those generated using the Bekker-Wong model, which considers

those and other terrain properties. Equation 2 was also based on the work by Lee (2013), who analysed the relationship between slip and tractive force for a rigid wheeled vehicle on snow with terrain modeling and an instrumented test vehicle. Lee produced a number of these relationships between force and slip through empirical testing and with simulation.

Resisting longitudinal wheel motion is the terrain resistance force. While the comparable force in the Bekker-Wong model varies with slip and sinkage, here it is given as

$$F_{tr} = \frac{b_{tr} \text{sign}(V_w) F_n}{X_t^A} \quad (3)$$

where b_{tr} is a unitless coefficient of terrain resistance force, and A is a constant used to tune the relative changes in different forces and torques with changes to the terrain coefficient. These two forces, F_{tr} and F_t , are summed and applied to the wheel longitudinally through a custom Gazebo plugin. These two forces, along with the reaction forces between the wheel and its axle, are used internally by Gazebo to calculate the accelerations for each link at each timestep.

Figure 3 shows the tractive, resistive, and net longitudinal force acting on a wheel for two different values of the terrain coefficient. The plots assume static forward motion with one quarter of the vehicle's weight distributed over each wheel. The two plots show that for $X_t = 1$, a positive non-zero slip is required to overcome the terrain resistance force and enable forward movement. For $X_t = 0.5$, no value of slip is enough to overcome the terrain resistance force and enable forward movement through traditional wheeled mobility.

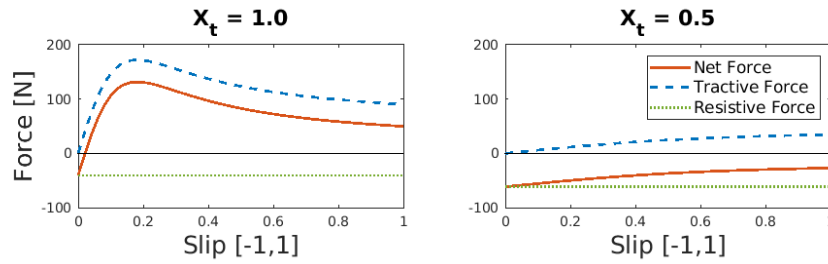


Figure 3: Estimated FrostyBoy per wheel longitudinal forces for two different terrain coefficients.

These two longitudinal force F_t and F_{tr} are then combined into a single net force that is equivalent to the single net longitudinal force acting on a wheel in the Bekker-Wong model. This net longitudinal force acting at the contact patch between the wheel and terrain causes a torque that resists the motor torque, T_m , which is applied to each wheel from a simulated motor.

The motor controller node sets a motor torque, which is then constrained by the torque-speed curve of the motor. *FrostyBoy* uses a 40:1 gear reduction between its motors and wheels, and *SnoWorm* therefore uses the same gear reduction. The motor and gearbox models are combined into a single equation that considers the mechanical and electrical properties of both. The drive train losses, including those from the actual gearboxes, are included as a 20% reduction in output torque. The drivetrain efficiency is therefore not directly present in the model.

A yaw resistance torque provides self alignment, and a lateral terrain resistance force resists sideways motion. These two variables are set qualitatively based on experience, rather than through data based ground-truthing. This is because the data available to ground-truth the terrain model was limited to longitudinal motion. Therefore, while other forces and torques included functions of the terrain coefficient, neither the yaw resistance torque nor lateral terrain resistance force include these functions.

4. Ground-Truth

The terrain model used to simulate vehicles within Gazebo was ground-truthed via data collected with the *FrostyBoy* UGV during the winter of 2020 in Lebel-sur-Quévillon, Quebec, Canada. While the resulting terrain model is specific to the particular wheels and approximate ground pressure of *FrostyBoy*, those same wheels and ground pressure are used for the simulated *SnoWorm*. In this way, data collected with an existing vehicle is leveraged to help inform the design of a future one.

In 2020 *FrostyBoy* was instrumented to measure the output speed and torque for each motor, drawbar force (net force to tow a sled), as well as ground speed, GPS position, and attitude. A Campbell Scientific CR1000X was used to log all data to a CFM100 CompactFlash memory module which was then transferred to a computer for analysis. A VectorNav VN-200 GNSS aided Inertial Navigation System (INS) was rigidly mounted to the vehicle chassis and measured accelerations with a resolution finer than 0.005 m/s^2 . The INS fuses data from a 9 DOF IMU and GNSS system to estimate ground velocity with an accuracy finer than 0.05 m/s . The INS provides many other data points, some of which are used internally for sensor fusion, but otherwise do not directly pertain to this paper.

The motor speeds and currents were directly measured by the RoboteQ FBL2360 motor controllers that also powered the vehicle's brushless DC motors. Torque was then calculated from current using the motors' torque constant and assuming 80% system efficiency for the combination of motor controller, motor, and gearbox. Instantaneous drawbar force was measured with an INTERFACE STA-3 S-beam load cell that was integrally mounted as part of *FrostyBoy*'s tow hitch. The load cell has a full range of 5 kN with a resolution of 1 N.

FrostyBoy was driven across a variety of surface types at different speeds and while towing different loads. The data collected is presented as five cases in Table 2, each of which shows steady state motion averaged over ten seconds. The first case was from an icy gravel road, the next three cases were all from moderately bonded snow on the surface of the frozen Lac (lake) Quévillon, while case 5 was from very deep and poorly-bonded snow near the limits of trafficability. The icy road is close to the 'ideal' surface, except that the near zero sinkage meant the grousers were not able to engage the terrain, and wheel slip was therefore higher than for ideal conditions. The table includes steady-state values for longitudinal ground speed, wheel speed converted to m/s based on wheel radius, wheel slip, output torque, and drawbar force as measured by a load cell between the vehicle and a towed weighted sled. X_t is the assigned terrain coefficient based on the mapping between simulation and empirical data. Case 4 is the only non-zero drawbar force, since it is the only case where *FrostyBoy* was towing a sled.

Table 2: Five cases of steady-state *FrostyBoy* driving

Case	Speed [m/s]	$R_w\omega_w$ [m/s]	Steady State Slip [%]	T_m [Nm]	Drawbar Force [N]	X_t
1	2.5	2.66	6 % (2%)	10	N/A	1
2	1.23	1.28	3.9%	27	N/A	0.83
3	1.98	2.05	3.4%	30	N/A	0.8
4	1.84	1.93	4.7%	32.8	28.8	0.8
5	1.11	1.58	30%	45	N/A	0.65

From the data collected in Lebel-sur-Quévillon, the terrain model coefficients can be defined so that the simulation matches the real world data. Setting the model to match a single terrain type is as simple as changing values via trial and error until simulation and data converge. The coefficients that define steady state longitudinal motion must be defined so that the simulated *FrostyBoy* quantitatively matches the physical *FrostyBoy*.

Table 3 gives the different coefficients and their values as used in this paper. While only the terrain coefficient, X_t , is varied during simulations, the other coefficients must be set to allow the simulated terrain to morph between different realistic snow surfaces as the terrain coefficient is varied.

Table 3: Coefficients internal to the terrain model.

Coefficient	Description	Value	units
b_{trF}	Terrain Resistance Force	0.165	<i>unitless</i>
b_{tF}	Slip to Tractive Force	0.7	<i>unitless</i>
b_{ltrF}	Lateral Resistance Force	0.1	<i>unitless</i>
b_{yT}	Yaw Resistance Torque	0.001	$1/m$
A	Terrain Coefficient Exponent	0.6	<i>unitless</i>

The first step in setting coefficients is to define them for the idealized terrain represented by $X_t = 1$, i.e., compacted snow or névé with minimal sinkage and where slip as low as 2% can develop sufficient net traction for forward motion. Case 1 in Table 2 was partly used as a reference for this terrain, but with the understanding that case 1 was actually on very

slick snow and therefore slip was much higher than it would have been on a hard snow surface represented by $X_t = 1$. The unitless coefficient of terrain resistance force (b_{trF}) gives a terrain resistance force that in turn requires a realistic motor torque to overcome. Since this idealized terrain should also require only small values of slip to produce large drawbar pulls, the coefficient of tractive/braking force (b_{tb}) is set to 0.7. This results in a simulation that maintains the steady state speed and torque from Case 1, but with only 2% slip versus 6% slip required on the icy terrain.

The terrain coefficient in the equations that govern the terrain forces and torques are modified so that they match the fifth entry in Table 2 at the expected value of $X_t = 0.65$. Case 5 was assigned a value of $X_t = 0.65$, then different coefficients were modified within the terrain model to match this case without affecting model behavior when $X_t = 1$. The terrain coefficient that represents the best match for cases 2, 3, and 4 were then found and are shown in Table 2. These terrain coefficients result in the expected torque, but in each case the simulated slip is slightly higher than it was in the real world. This is not unexpected, since the different cases represent different real world snow surfaces, rather than a perfect continuum of snow.

5. Mobility Modes

This section examines two different mobility modes for the *SnoWorm* that take advantage of its multi-segment nature. The first is inchworm movement, where an active prismatic joint in the link between segments allows segments to push and pull on another using a force other than their wheels. This mode is useful when traditional wheeled mobility fails and an alternative is required to prevent immobilization. The second mobility mode is force control, where during forward driving the vehicle uses the wheel speeds of individual segments to control the forces between them. By setting inter-segment forces, the *SnoWorm* can distribute a towed load evenly across all its segments, or have some take up more of the load. This could be done to relieve a segment if its drivetrain was damaged, or if a sensor requires additional electrical power.

5.1. Inchworm Movement

Work by Spanksi (1967) and Czako et al. (1963) created and tested the idea of inchworm movement in the context of high mobility military vehicles. More recently, the concept was extensively tested on physical hardware by Creager et al. (2015), who worked with a four wheeled, two axle vehicle (Bartlett et al. 2008). Similar to the earlier work on the Thrust-Stride-System (Gross-Scharmann 1960), Creager's testing revealed the relationship between rolling and inching, and showed that in many cases inching allows movement through otherwise immobilizing terrain. This testing supports the conclusions of the earlier work on inchworm movement, which showed that for all but the softest soils, inchworm movement can allow for extrication from immobilization. Creager et al. (2015) also compared the energy consumption between rolling and inching, and showed that inchworm movement was more efficient than traditional wheeled mobility in immobilizing and marginal terrain.

This section adds an active prismatic joint to each connector link of the *SnoWorm*. This allows individual segments to push and pull one another for improved mobility in marginal and immobilizing terrain. Figure 4 shows a sequence of images from Gazebo of the *SnoWorm* performing inchworm movement. Between the first and second image, segment two is brought forward by contacting the joint ahead of it, and expanding the joint behind it. Between the second and third image, segment three is brought forward by the same series of expansion and contraction.

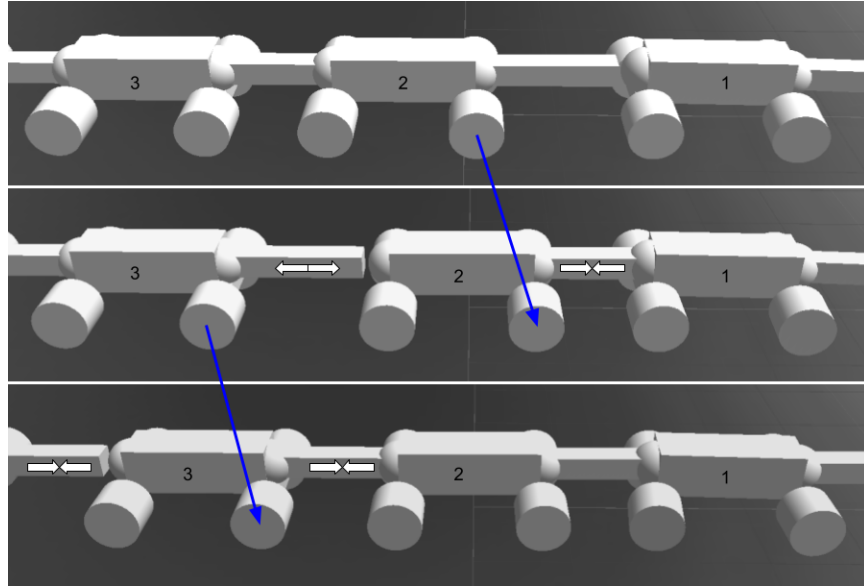


Figure 4: Sequence of screen captures taken from within Gazebo showing inchworm movement.

For a two segment vehicle using an inter-segment prismatic joint to implement inchworm movement, the sequence of motions is as follows: starting from an extended state, the vehicle locks the front segment's wheels and drives its rear segment's wheels forward, while contracting the prismatic joint. The front segment stays in place, while the rear segment moves forward. Once the vehicle reaches the fully contracted state, the rear segment's wheels are locked while the front segment's wheels are driven forward, and the prismatic joint is expanded. Locking wheels can be implemented through closed loop speed control to maintain zero speed. This process is repeated until the vehicle reaches its destination, or enters terrain that allows for traditional wheeled mobility.

Based on simulation work, inchworm behavior allows a vehicle to make forward progress through terrain that would otherwise cause immobilization. Because one aspect of immobilizing terrain is often an increased terrain resistance force, the wheels being held stationary are able to act as anchors to resist being pushed or pulled backwards by the prismatic joint.

The viability of inchworm movement holds true with Gazebo's default friction model and with the custom terrain model implemented in this paper. The underlying mathematics of inchworm movement draws on the earlier work by Czako et al. (1963) and Creager et al. (2015). When the front segment of a two segment vehicle is driving forward, and the prismatic joint is expanding

$$F_c + F_{t_{front}} - F_{tr_{front}} > 0 \quad (4)$$

$$F_{tr_{rear}} - F_c \geq 0 \quad (5)$$

where F_c is the force from the prismatic joint pushing the two segments apart, $F_{t_{front}}$ and $F_{tr_{front}}$ are the tractive and terrain resistance force of the front segment respectively, and $F_{tr_{rear}}$ is the terrain resistance force of the rear segment. Combining Equations 4 and 5 into a single inequality, while assuming the rear segment's wheels are not actively driven yields the following

$$F_{t_{front}} + F_{tr_{rear}} - F_{tr_{front}} > 0 \quad (6)$$

As long as the difference between the terrain resistance force and the traction force of the front segment is less than the terrain resistance force of the rear segment, the vehicle will move forward. This inequality could break down if the terrain resistance at the front segment is much higher than at the rear, or if the force from the prismatic joint is too high and Equation 5 no longer holds. While the example here has the wheels on the rear segment held stationary, the earlier work by Czako et al. (1963) instead has them driven forward at a low speed. By attempting to drive forward, these wheels can provide not only the terrain resistance force they are pushing against, but also traction force from their positive slip. While this worked on various soils, it may instead cause excessive excavation and further immobilization on snow surfaces.

This simple conclusion, that non-zero traction force is the sole requirement for successful inchworm movement, ignores the case where the "stationary" segment slides backwards. This conclusion also assumes that the relationship between the force of the prismatic joint and the terrain forces always holds, which may not always be the case.

Simulating this behavior results in the outputs shown in Fig. 5, where a six segment *SnoWorm* engages in inchworm movement where two segments are held stationary for each segment that moves. A Simple force controller is used on the prismatic joint to limit backwards sliding of the stationary segments. This simulation took place of terrain with $X_t = 0.25$, a much lower value than any of the real world cases experienced by *FrostyBoy*. Simulations with higher terrain coefficients had similar results, but with less backwards sliding of stationary segments, higher forward speeds, and lower forces required for the prismatic joints. Note how negative speeds peak shortly after motion starts, then drop off as the prismatic force is decreased. Segments 0 & 3, 1 & 4, and 2 & 5 each have the same location within this movement pattern, their speeds therefore overlap.

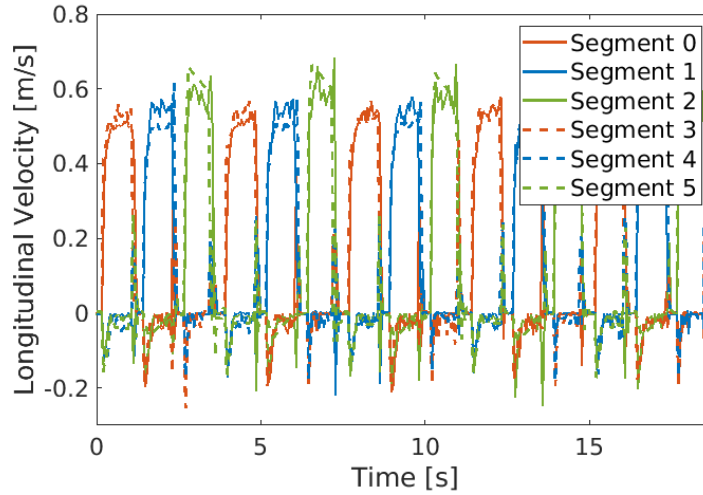


Figure 5: Inchworm movement for a six segment *SnoWorm*.

Because of the low value of X_t , individual segments cannot generate positive net longitudinal force. This is overcome in inchworm movement since as equation 6 showed earlier, the limitation is on the difference in force acting on multiple segments at once. The prismatic joint provides the necessary additional force to push a segment forward when its wheels cannot generate their own net positive longitudinal force.

5.2. Force Control

A multi-segment vehicle can use control schemes to distribute the forces from a towed load differently across different segments. The simplest control scheme for distributing the forces is to drive *SnoWorm* without using the inter-segment forces as a control input. Figure 6 shows this case for a six segment *SnoWorm*, where a 200 N towing force is applied to the rear most segment (follower 5). The simulation begins at $Time = 0s$, and the vehicle controller activates just before $Time = 1s$. The first subplot shows the forces at the connector links between each segment, with positive forces indicating that the connector link is under tension. The second subplot shows the net forces acting on each segment, where a positive net force means a force pushing the vehicle forward. Once the vehicle starts actively driving, the net forces acting on each segment become uniform, and the 200 N load is evenly distributed across the six segment *SnoWorm*. In this example, the terrain coefficient is set to 0.9, and the closed loop linear wheel speed is 1.8 m/s.

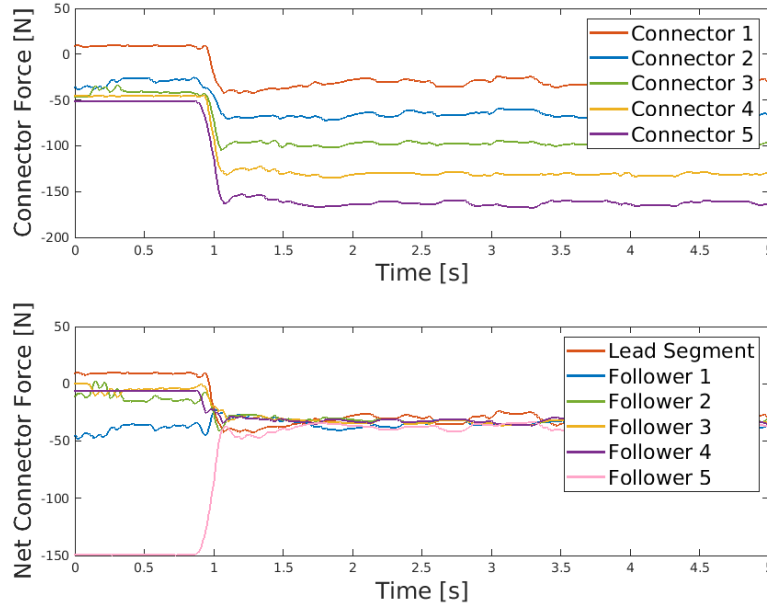


Figure 6: Forces between segments with a 200 N force acting on the final segment. Segments begin driving simultaneously.

In Fig. 6 each start moving simultaneously, and this is not realistic. A real world vehicle would have some slight difference in when each segment starts powering their wheels, be it from signal delay or different amounts of static friction within gearboxes. To ensure the results are not artifacts of this simultaneous start, the same simulation is performed with an exaggerated one second delay between each segment's start time. Figure 7 shows the simulation outputs in the same format as Fig. 6, and reaches the same steady state condition after the rear-most segment begins driving at $Time \approx 6s$.

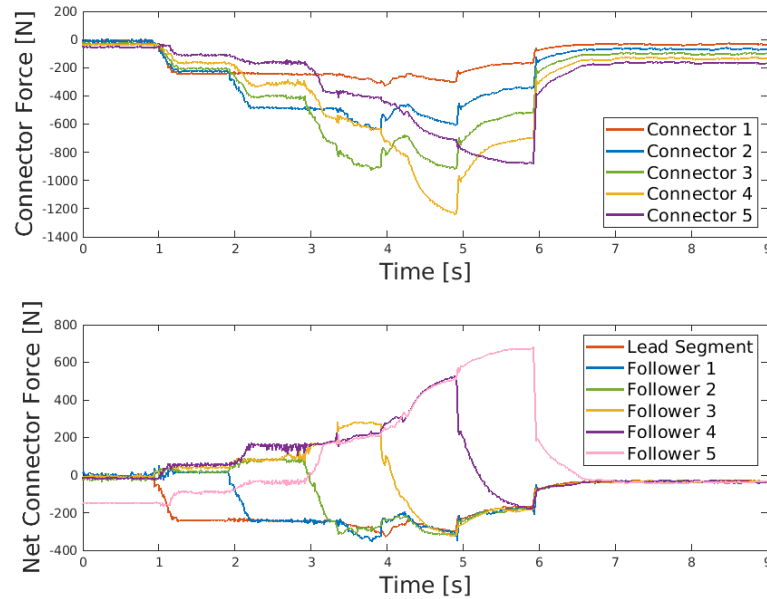


Figure 7: Forces between segments with a 200 N force acting on the final segment. Segments begin driving staggered by one second.

These figures are not meant to imply that the specific forces shown during startup are accurate, but that the simulation consistently reaches the same steady state forces with different initialization procedures.

Equalizing forces between segments is not the only control possibility. *SnoWorm* segments can also be controlled to

unevenly distribute towed loads. A proportional-integral controller uses a segment's wheel speed as the control variable, and the force in an adjacent connector as the process value. This controller was simulated for a two segment vehicle where a 200 N force is pulling the rear segment backwards, and the force between segments is being controlled to zero. The output from the PI controller, V_{sp} the linear velocity setpoint, is constrained to positive values so that the lead segment does not drive backwards when the simulation initializes and there is tension between the segments from the rearward force on the rear segment. Figure 8 shows the time domain outputs from this simulation, where the connector force starts at a negative value (tension between segments), then goes positive as the vehicles start moving, and settles to zero as the controller operates.

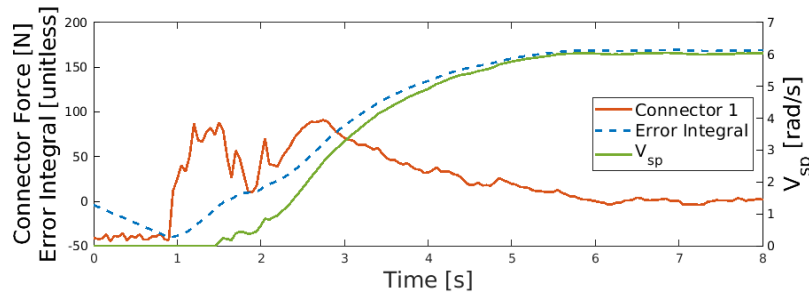


Figure 8: Time domain variables from a PI controller used to control the connector force for a two segment vehicle.

The applications are limited for controlling the force between segments in a two segment vehicle, but are more useful with an increasing number of segments. As an example, Fig. 9 shows the case where a three segment vehicle works to zero out the net connector force on the middle segment, while spreading the towed load evenly across the leader and rear-most segments. The towed load is again 200 N, so the lead segment works to maintain a constant 100 N tension between it and the middle segment, the middle segment works to maintain a constant 100 N tension between it and the rear-most segment. Figure 10 shows the force distribution.

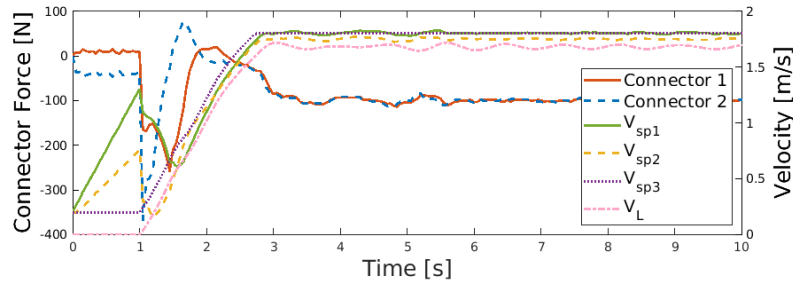


Figure 9: Time domain variables from a three segment SnoWorm working to maintain zero net connector force on the middle segment.



Figure 10: Force diagram of a three segment SnoWorm working to maintain zero net connector force on the middle segment.

The velocity setpoint for the wheels on the front segment and rear segment are nearly the same, while the middle segment's wheels are driven slower in order to provide less tractive force, and therefore take on less of the drawbar force. Both connector links have the same 100 N tension, so the net force acting on the middle segment from the connector links is zero.

The next possibility for force control is where the net connector force acting on a segment is positive, and where there is a change over time on which segment has this positive net connector force, which could be coordinated with the location of

the *SnoWorm*. This could be useful if passing over a localized area of poor terrain, or an especially sensitive area where there is a desire to keep wheel ruts and sinkage resulting from slip to a minimum.

Figure 11 shows the case where rather than the middle segment providing an equal proportion of the required drawbar force, or zero net force through the connectors, it instead has a net positive connector force. This means that the segments behind and ahead of it are pushing and pulling it respectively. The other two segments therefore not only take on the entire 200 N towed load, but also provide drawbar force to move the middle segment. Figure 11 is similar to Fig. 9, but the connector forces are not kept equal, and the wheel speed of the middle segment is lower relative to the other segments.

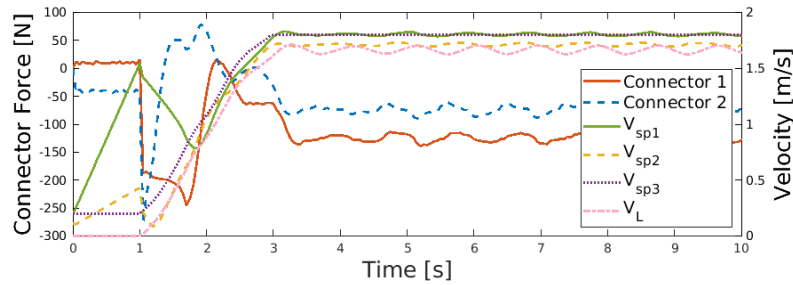


Figure 11: Time domain variables from a three segment *SnoWorm* working to maintain positive net connector force on the middle segment.

6. Conclusion

While wheels are a dominant form of UGV mobility, there are certain off-road environments where traditional wheeled mobility can cause immobilization. Rather than replacing wheels with more energetically intensive tracks, or more complex legged mechanisms, The *SnoWorm* provides the benefits of wheels while offering alternative mobility modes that take advantage of its multi-segment nature. Both inchworm movement and force control use *SnoWorm*'s inter-segment links to transfer force between segments and provide improved mobility relative to the equivalent number of single-segment vehicles. These behaviors are simulated with a combination of ROS and Gazebo while using a custom terrain model that has been empirically ground truthed against data collected from *FrostyBoy*, a vehicle with equivalent size and weight, operating on snow.

Acknowledgements

This work was primarily supported by the Engineering PhD Program at the Thayer School of Engineering at Dartmouth College. Support was also provided by the National Science Foundation's (NSF) Division of Civil, Mechanical and Manufacturing Innovation's (CMMI) program for Dynamics, Control and Systems Diagnostics' (DCSD) award #1824687.

References

- Arcone, S.A., Lever, J.H., Ray, L.E., Walker, B.S., Hamilton, G., Kaluzienski, L., 2016. Ground-penetrating radar profiles of the McMurdo Shear Zone, Antarctica, acquired with an unmanned rover: Interpretation of crevasses, fractures, and folds within firn and marine ice. *Geophysics* , 81(1), WA21–WA34.
- Arvidson, R.E., Iagnemma, K.D., Maimone, M., Fraeman, A.A., Zhou, F., Heverly, M.C., Bellutta, P., Rubin, D., Stein, N.T., Grotzinger, J.P., Vasavada, A.R., 2017. Mars Science Laboratory Curiosity Rover Megaripple Crossings up to Sol 710 in Gale Crater. *Journal of Field Robotics* , 34(3), 495–518.
- Aucone, E., 2019. Enhancing Mobility and Control of the DuAxel Rover System. Master's thesis, University of Pisa, Italy.
- Bartlett, P.W., Wettergreen, D., Whittaker, W., 2008. Design of the Scarab Rover for Mobility & Drilling in the Lunar Cold Traps. Technical Report, Carnegie Mellon University, Pittsburgh, PA.
- Creager, C., Johnson, K., Plant, M., Moreland, S., Skonieczny, K., 2015. Push-pull locomotion for vehicle extrication. *Journal of Terramechanics* , 57, 71–80.

- Czako, T., Janosi, Z., Liston, R.A., 1963. Analysis of Multi-element Inching Vehicles. Technical Report, US Army Land Locomotion Laboratory.
- Gross-Scharmann, F., 1960. Die Triebkraftsteigerung bei Geländefahrzeugen durch das Schub-Schritt-Verfahren (Increased Driving Force for Off Road Vehicles Through the Thrust-Stride-System). Landtechnische Forschung (agricultural research) , 11(4), 89–96.
- Kassel, S., 1970. Lunokhod-1 Soviet Lunar Surface Vehicle A Report prepared for. Technical Report, RAND Corporation, Santa Monica, CA.
- Lee, J.H., 2013. Calibration and validation of a tire-snow interaction model. Journal of Terramechanics , 50(5-6), 289–302.
- Lever, J.H., Delaney, A.J., Ray, L.E., Trautmann, E., Barna, L.A., Burzynski, A.M., 2013. Autonomous GPR surveys using the polar rover yeti. Journal of Field Robotics , 30(2), 194–215.
- Lever, J.H., 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), 527–551.
- Lines, A., 2020. Advances in Autonomous Ground Traverses of the Earth’s Polar Regions. Ph.D. thesis, Dartmouth College, Hanover, NH.
- Lines, A., Elliott, J., Ray, L., 2021. Rigid Wheel Design and Evaluation for Lightweight Snow Rover. International Society for Terrain-Vehicle Systems. Forthcoming.
- Pacejka, H.B., Bakker, E., 1992. The Magic Formula Tyre Model. Vehicle System Dynamics , 21(S1), 1–18.
- Pyle, R., 2019. Interplanetary Robots: True Stories of Space Exploration. 1st ed. Amherst, New York.
- Smith, R., 2006. Open Dynamics Engine v0.5 User Guide. URL: <http://ode.org/ode-latest-userguide.html>.
- Spanksi, P.L., 1967. Design and Fabrication Mobility Exercise ”A” Test Rigs. Technical Report, Mobility Systems Laboratory US Army Tank Automotive Command, Warren, MI.