Data-Driven Slip Model for Improved Localization and Path Following applied to Lunar Micro-Rovers

Samuel Ong

CMU-RI-TR-22-74
16th December, 2022

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
David Wettergreen (*Chair*)
Dimitrios (Dimi) Apostolopoulos
Brian Jackson

Submitted in partial fulfillment of the requirements for the degree of Masters of Science in Robotics.

To my mum and dad, who had always supported me in all my endeavors.

Abstract

Micro-lunar rovers need to solve a slew of challenges on the Moon, with no human intervention. One such challenge is the need to know their location in order to navigate and build maps. However, localization is challenging on the moon due to lack of supporting infrastructure, such as global positioning satellites. It is made more difficult due to constraints on available sensing and computing on such rovers. Low gravity and regolith properties cause varying wheel slippage, further complicating position estimation.

This thesis presents a data-driven slip model comprising of 3 layers that enables lunar micro rovers to perform more accurate position estimation through wheel odometry and follow planner commanded drive-arc closely. This is crucial in micro rovers with sensors limitations, arising from the inability to carry better sensors due to size and weight limitations. We start by describing a kinematics model formulation leveraged from previous literature which allows easy integration of our slip model. Next, we describe in detail the layers in the slip model and present how the parameters of the model are derived. We conclude by demonstrating the effectiveness of our slip model in achieving our localization and controls goals through testing using a skid-steer surrogate rover.

Acknowledgments

This thesis would not have been possible if not for David S. Wettergreen and William (Red) L. Whittaker, who brought me into the realm of space robotics, along with the folks who poured their heart and soul into achieving the MoonRanger mission.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Space Exploration

There are increasing efforts in planetary exploration over the past decades, with increasing effort in colonizing the moon. With the Artemis program, NASA (National Aeronautics and Space Administration) introduced the Commercial Lunar Payload Services (CLPS) initiative, bringing together various companies for a combined effort to explore the Moon and prepare for human missions.

The number of robotic wheeled platforms landing on Moon is steadily increasing. Earlier this year, the Indian Space Research Organisation (ISRO) attempted to launch a Rover - Pragyann (Figure 1.1) to space, but unfortunately crashed on landing. In Q1 2023, the Iris Rover (Figure 1.2) designed by students at Carnegie Mellon University is expected to launch on Astrobotic first lunar lander.

NASA's Volatiles Investigating Polar Exploration Rover, or VIPER (Figure 1.3, is also slated to launch in late 2024 to the polar regions of the moon in search for lunar ice. Additionally, another rover - Lunar Vertex (Figure 1.4 will launch on the 3rd CLPS mission to visit Reiner Gamma, a high magnetic region on Moon.

For interest and information, a summary of other lunar robotic platforms being slated to launch in the 2020s can be found in [3].

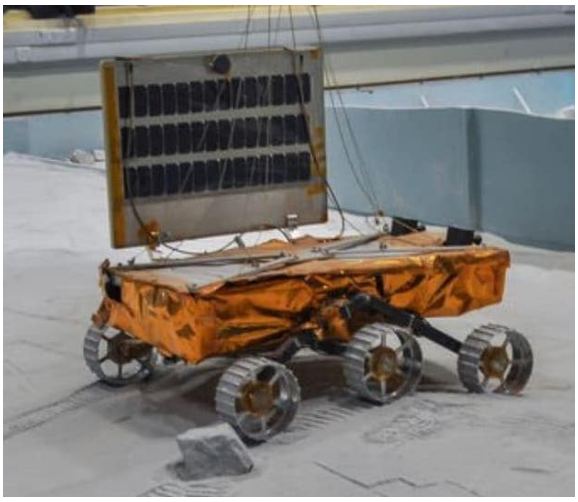

Figure 1.1: Pragyann rover - Image courtesy: ISRO

Figure 1.2: Iris Rover - Image courtesy: Iris Team

1.2 Problem Description

Lunar rovers needs to solve a slew of challenges on Moon, with no human intervention. One such challenge is the need to know their location in order to navigate and build maps. They would also need to relay accurate position of science objects back to Earth for scientist to tag points of interest for future missions. However, localization is challenging on the moon due to lack of supporting infrastructure, such as global positioning satellites. It is made more difficult due constraints on available sensing and computing on rovers, where flight hardware are usually based on flight heritage and

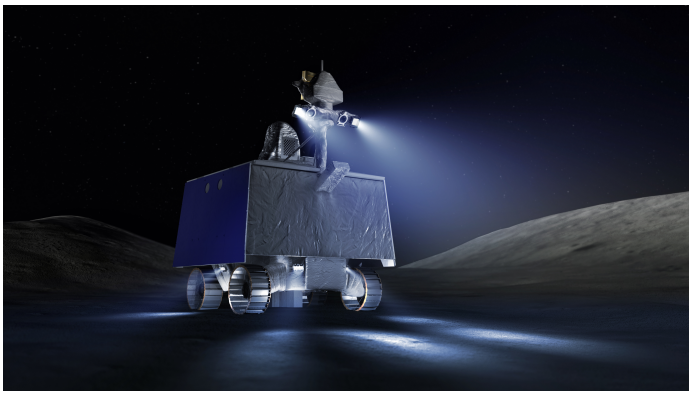

Figure 1.3: Viper - Image courtesy: NASA Ames

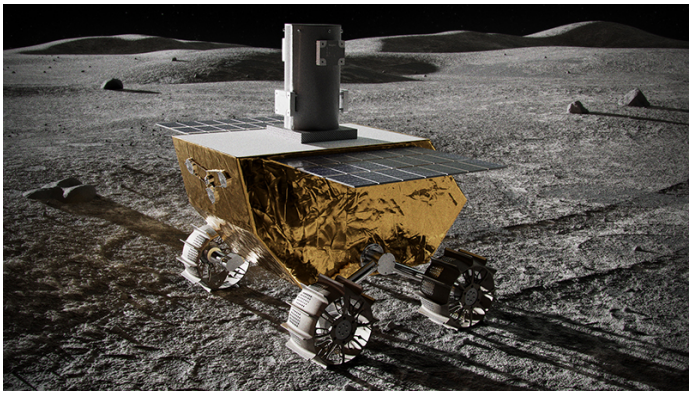

Figure 1.4: Lunar Vertex - Image courtesy: Johns Hopkins APL

radiation hardened requirement. Low gravity and regolith[1] properties causes varying wheel slippage, further complicating position estimation. Future conceptualized lunar rovers must increasingly travel longer distances at higher speed, often due to mission requirements and the need to move through areas of light and shadow quickly, which results in a higher magnitude of wheel slippage. NASA conceptualized a mission to sent a robotic rover (Intrepid) to the moon that has a planned traverse of 800km over 4 years [4].

In this thesis, we chose to focus on localization for space exploration rovers, specifically for lunar micro-rovers[2], tying in with the development of a lunar micro-rover here - MoonRanger. Additionally, this localization problem becomes harder for lunar micro-

[1]A region of loose unconsolidated rock and dust that sits atop a layer of bedrock
[2]We loosely define micro-rovers as rovers with masses of one to a few kilograms

rovers due to sensor limitations. Thus, it is increasingly challenging to create accurate position estimation for these micro rovers, and becomes a exciting problem to work on. We developed a slip model that is incorporated in our on-board position estimation stack that improves position estimates with a reduction of absolute trajectory errors up to 4x, compared to when a slip model was not used. We also attempt to use the slip model in heading compensation that provides better path following capability by following tighter drive arcs.

1.3 Contributions

The main contributions of this thesis can be summarized as:

1. Development of a data-driven slip model (novelty)
2. Application of a Kinematics Model (application)
3. Improving Dead Reckoning accuracy through slip compensation (result)
4. Better path following through more accurate drive-arcs (result)
5. Experimental testing and validation of methods (methods)
6. Datasets created in development

Chapter 2

Background

We begin this chapter by introducing our micro lunar rover - MoonRanger. Next, we discuss previous work in using visual odometry for slip estimation and explain why it does not worth for MoonRanger. Lastly, we explore various methods used by other researchers in deriving a slip model and its limitations, along with other benefits of a good slip model.

2.1 MoonRanger

MoonRanger, a micro lunar-rover, is the size of a suitcase and weighs about seven pounds when it is on the moon. It is too small to carry a radio to reach Earth, so exploration must be completely autonomous once it is out of range of its lander. While exploring the Moon, it senses terrain, navigates, estimates its location, builds maps, and measures the concentration of hydrogen proportional to the water that underlies its route.

MoonRanger's science objective is to search for lunar ice. If ice is sufficiently concentrated, and if the melted water can be processed, it would become a key resource for living and working on the moon. Humans can drink the water, grow food with it, and breathe the oxygen derived from it and colonizing the Moon becomes a possibility. Hydrogen and oxygen from the water can power fuel cells to produce electricity. The hydrogen and oxygen can also be used to make rocket propellants for exploring

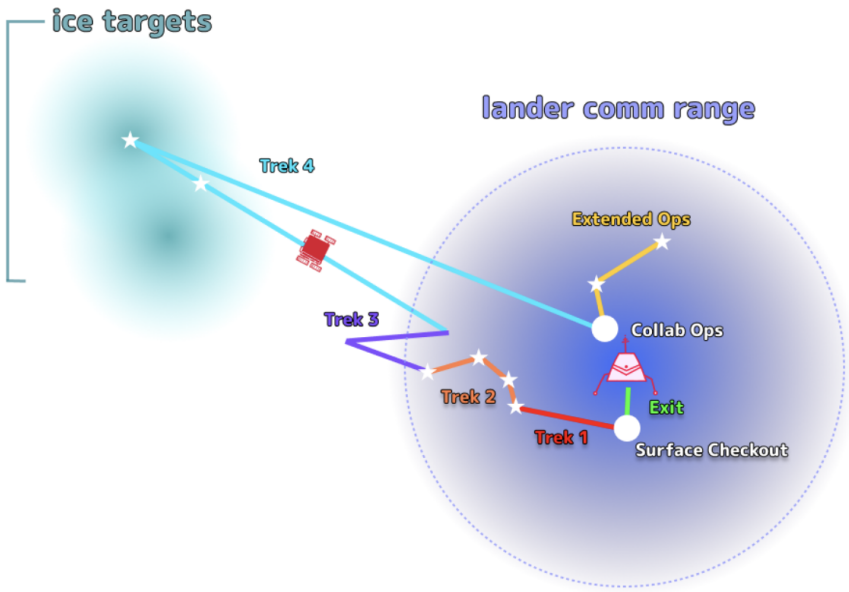

Figure 2.1: Mission Plan for MoonRanger

beyond the Moon and shuttling back to Earth.

Figure 2.1 visualizes the mission plan that MoonRanger would execute. Trek 4 is the most challenging trek for MoonRanger, which is needed to search for the presence of lunar ice. It requires MoonRanger to exit the communication range of the lander, while autonomously traverse a round-trip distance of **1km**. To transmit the location and presence of lunar ice requires MoonRanger to return back to the lander, which necessitates accurate localization.

2.1.1 Sensor Constraints

MoonRanger sensing suite is made up of:

- Front and Back Stereo Cameras
- STIM-300 Inertial Measurement Unit (IMU)
- nanoSSOC-D60 Sunsensor
- 4 Hall Sensors at each wheels

MoonRanger front camera subsystem comprises of a stereo pair of Leopard Imaging

IMX274 cameras, fitted with Kowa LM3QS56 3mm lenses. These lenses provide a 70 x 87° on a 1/2.5" sensor. Cameras are mounted 24.4cm above ground and pointed downwards at an angle of 26° from the horizon. Stereo image pairs are processed at a rate of 1Hz.

Figure 2.2 shows sample images of the MoonYard captured by MoonRanger Surrogate Rover's cameras which has the same camera configuration. These images emphasize the drawback of low mounted cameras with limited field of view, that are unable to capture much features in an image.

Figure 2.2: Sample Images captured from Morphin Cameras

2.2 Visual Odometry for Slip Estimation

Visual Odometry (VO) can be used to estimate wheel slip ratios, where augmenting it with RANSAC (random sampling sequence) also helps to improve accuracy of motion estimation shown in [5]. Similarly in [6], the authors demonstrated how VO can be used to estimate and update a slip prediction model. Whereas state-of-the-art rovers such as the Mars Exploration Rovers (MER) use VO to directly provide accurate localization, detecting and compensating for unforeseen wheel slippage during a drive. VO has been proven highly successful for the MERs (spirit and opportunity), converging more than 95%, detecting slip ratio more than 125% and measuring translation as small as 2mm [7].

The latest Mars Rover - Perseverance, continues to use VO for position and slip estimation, but have shifted the processing off to a Xilinx Virtex 5 FGPA (field-programmable gate array) to improve processing time. Running visual odometry algorithms on 20-MHz RAD6000 CPU onboard the MER took 160 seconds to estimate

a relative pose change. On Perseverance, the computation has been split between a faster 200-MHz RAD750 CPU and Virtex 5 FPGA, and the algorithm takes 8.8 seconds to compute [8]. 8.8 seconds is a long duration, showcasing the huge amount of computation power required to process the stereo image pair and run VO algorithms in space.

2.2.1 Effectiveness for MoonRanger

From our testing, medium size rocks that are approximately 10cm in height get registered by the mapper when they are approximately 2m from Morphin (MoonRanger's surrogate rover). The limited range and field of view makes it challenging to perform visual odometry. Some effort was placed on implementing visual odometry but was unsuccessful. Figure 2.3 shows state-of-the-art ORB (Oriented FAST and Rotated BRIEF) feature detection [9] on collected sample images. Some corners on the rocks were detected as features but were not consistent over multiple frames to yield useful translation estimates. Additionally, multiple similar feature across the tracks were picked up by the feature detector, yielding erroneous estimates.

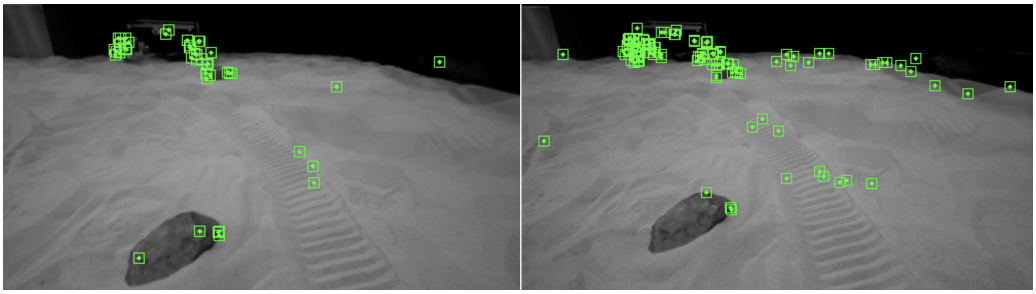

Figure 2.3: Sample Images captured from Morphin Cameras

To verify that feature detection does work on featured environment, and performs poorly on featureless environment, we applied the same VO algorithm on images captured in a feature-rich environment. A sample picture of the feature rich environment is showed in Figure 2.4 and sample images with the detected features captured by MoonRanger's cameras can be seen in Figure 2.5. From the captured images, we can see a much higher number of features being detected, which improves the convergence of VO.

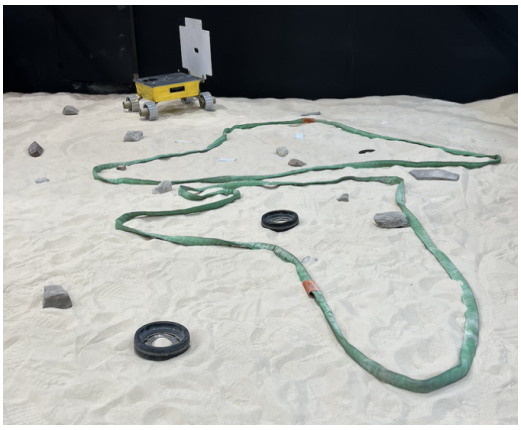

Figure 2.4: Feature Rich Environment

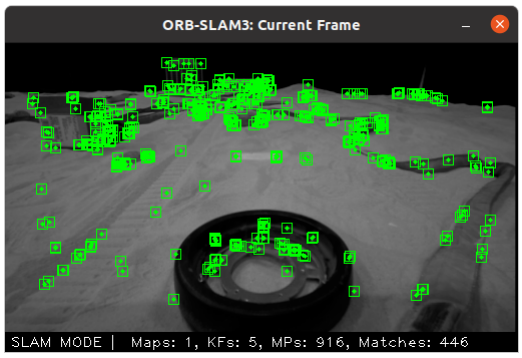

Figure 2.5: Sample Processed Image

Results of applying VO on images captured by MoonRanger's stereo camera pairs in a feature-rich environment and featureless environment can be found in Figure 2.6. VO performed relatively well in the feature-rich environment (blue and pink dotted lines), being able to replicate the ground truth trajectory (green line). However, in the featureless environment, we see that VO consistently underestimates the translation of the rover at every time step, where the green line is the ground and blue dotted line is the estimated translation from VO.

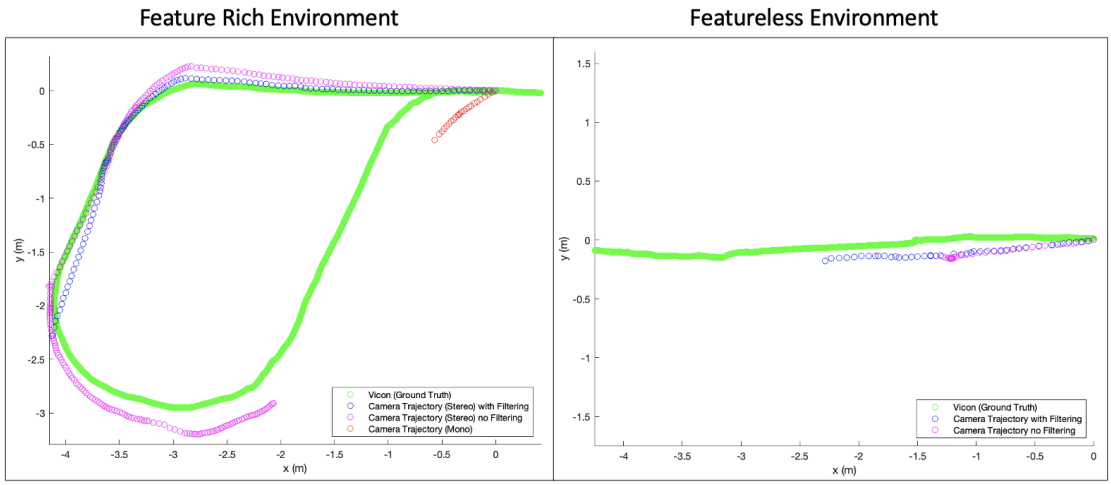

Figure 2.6: Sample Images captured from Morphin Cameras

Note: This section is a combined work conducted at CMU between the author, Rohan

Deshpande and Swapnil Pande.

2.2.2 Featureless Visual Odometry

It is possible to perform visual odometry based on correlation between images instead of relying on detected features in the stereo images. This can be achieved through phase correlation and a series of Fourier transform functions, generating the position changes across frames in 6 degree of freedom, comprising of delta changes to translation and rotation [10]. However, due to limited computation budgets, this option was not explored.

2.3 Other Methods to Estimate Slip

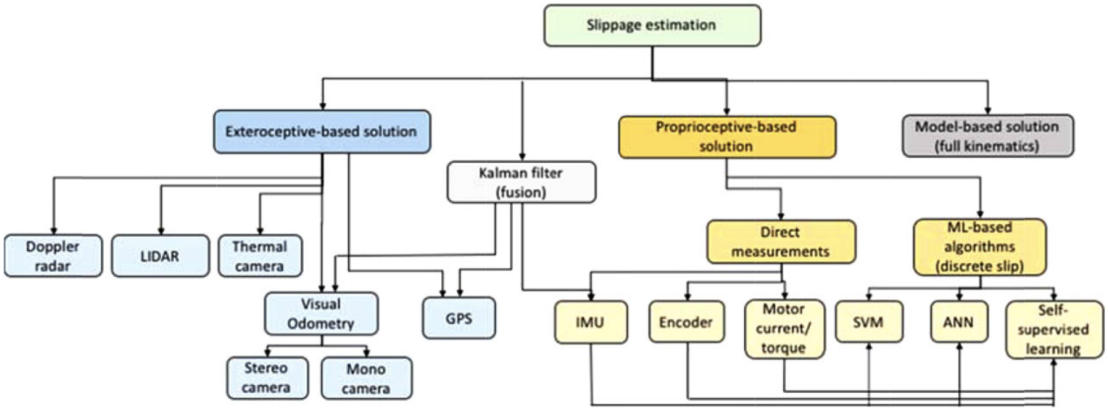

Figure 2.7: Type of Slip Estimation Methods - Image Courtesy: [1]

Figure 2.7 summarized by [1] provides a good overview of the remaining slip estimation methods. Since we have shown in Section 2.2 that extereoceptive-based solutions do not work for Lunar Micro Rovers, we will focused on proprioceptive-based solutions. Before we discuss propioceptive-based solutions, note that it is also possible to estimate slip without a slip model by treating it as a disturbance. In [11], the authors used an EKF (Extended Kalman Filter) to estimate wheel slippage by modeling it as a process noise, improving vehicle position localization. However, this approach was limited to straight drives and point turns, where localization accuracy heavily depends on the measurement matrix and initial covariance matrix.

Modeling slip with direct measurements can also be an effective way to estimate slip during rover drives. In [12], Sutoh et al. in Shinshu University applied slip compensation on a 4-wheel skid steered rover to improve position estimates. The authors employed experimental results to generate a slip model, relating input wheel rates to slip compensated wheel rates. Though not wheeled vehicles, the authors in [13] presented a slip model which estimates individual track slippage for different maneuvers (straight driving and turning). The authors also verified the slip model proposed by [12] on weak slopes, and found that the model explains 75.3% of the observed data.

In [14], Bussmann et al. explored using IPEM (Integrated Prediction Error Minimization) described in [15] to calibrate a slip model described in [16]. The authors conducted experiment on Mt. Etna using a independently steerable 4-wheel rover with articulated suspension to demonstrate the benefits of position estimation with a calibrated slip model. In [17], the authors used slope angle and soil parameters to derive a slip model that estimates slip ratio based on the current terrain angle, but it is limited to straight drives. The authors in [18] used a different approach to derive a slip model, where slip ratio is estimated from wheel odometry and input torque, generating a slip model. However, accuracy of slip estimation is heavily dependent on accuracy of driving resistance, and testing were limited to straight drives with induced slip through large acceleration and braking.

2.3.1 Model Learning Methods

The advancement of model learning methods makes learning of a slip model possible. Knowing the type of terrain the vehicle is traversing on can be useful for slip estimation, where the authors in [19] used a multi-layer, deep neural network on vehicle's vibration data to classify 5 different terrains that the vehicle is currently traversing on. Whereas in [20], the authors used support vector machine to classify the terrain based on images and proprioceptive sensors to classify wheel slippage into 3 groups (low, medium, high slip).

A supervised learning method - Gaussian Process Regression (GPR) can also be used to learn and predict wheel slippage. The GPR model is capable of providing

uncertainty measures over prediction [21] after being trained from prior knowledge of wheel slippage scenarios. Wheel slippage predictions and the uncertainty measures are useful in bounding the localization error of the rover at every time step.

The authors in [22] used GPR to learn a slip model and have shown good prediction of longitudinal and lateral wheel slippage through input features such as roll, pitch, wheel current and angular velocities of the wheels. Similarly in [23], the authors uses a time-series GPR model to predict wheel slippage at the current time step, where the inputs to the model is the data of past wheel slippage. To get a short summary of how GPR works, one can refer to [24].

2.4 Slip-based Controls

Other than improving position estimates with slip models, knowledge of slip ratio also enables effective controls of planetary rovers which is achieved in [25] through slip-based traction controls, allowing rovers to traverse steeper slopes. However, the authors did not purpose a viable method to estimate slip ratios during drive. Similarly in [26], the authors also demonstrated how compensating for slip ratios can lead to better path following in every slope angle. Referenced earlier in Section 2.2, the authors who used VO to estimate slip in [6], also presented a method in [27] that enabled better path following of a rover using a slip model. This shows that knowledge of wheel slippage is crucial to enable trajectory following.

Chapter 3

Vehicle Modeling

A vehicle model is needed to transform wheel rates to body velocity for position estimation. In this chapter, we first describe the differential-drive model which is commonly used to compute the forward and inverse kinematics of a 4-wheeled vehicle without explicit steering, while explaining the limitations of such a model. We then present the kinematics model[1] derived by Seegmiller in [2], which provides an elegant solution for incorporating wheel slip in a generic kinematics model.

3.1 Differential-Drive Model

The differential-drive model (Figure 3.1) is commonly used to control independently driven 4-wheeled vehicles without explicit steering. A drive command is usually specified in terms of a drive arc, which is parameterized by a linear velocity v and a turn radius R. Using Equation 3.1, we can convert the drive arc to the left and right wheel velocities. Yaw rate $\omega = 0$ for straight driving and $\omega = \frac{v}{R}$ for turning.

$$\begin{bmatrix} v_l \\ v_r \end{bmatrix} = \begin{bmatrix} v - w \cdot \omega \\ v + w \cdot \omega \end{bmatrix} \tag{3.1}$$

[1]Work in this chapter is a combined effort with Ben Kolligs at CMU [28].

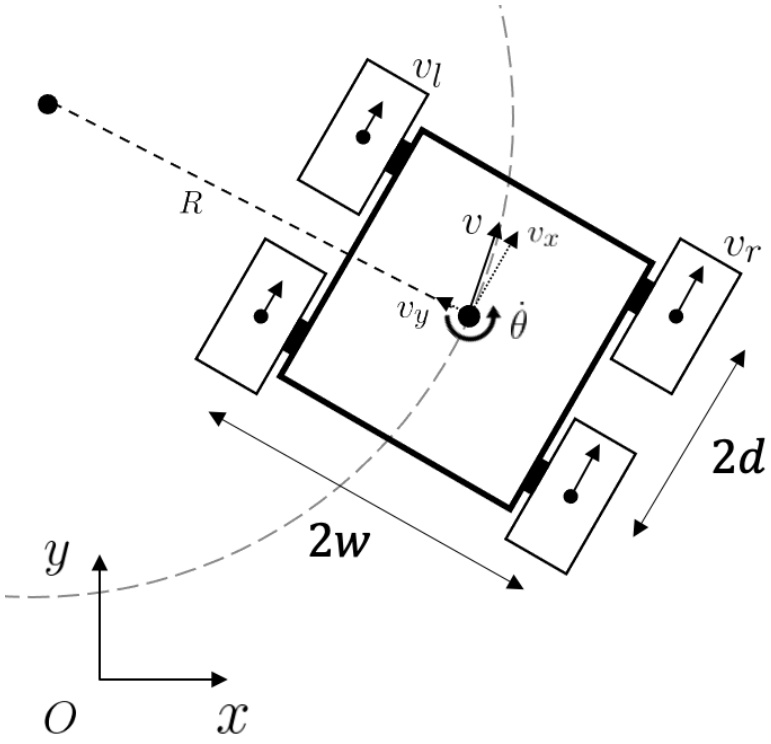

Figure 3.1: Differential-Drive Model

Equation 3.2 allows us to obtain the state of the system in the body frame.

$$\begin{bmatrix} \dot{x_b} \\ \dot{y_b} \\ \dot{\theta_b} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ \frac{-1}{2w} & \frac{1}{2w} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \tag{3.2}$$

To obtain the state of the system in the global frame from the body frame, we can use the following transformation matrix in Equation 3.3.

$$\begin{bmatrix} \dot{x_w} \\ \dot{y_w} \\ \dot{\theta_w} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x_b} \\ \dot{y_b} \\ \dot{\theta_b} \end{bmatrix} \tag{3.3}$$

There is no easy way to incorporate individual wheel slippage into Equation 3.1 to obtain slip compensated wheel velocities for position estimation. Additionally, in the forward kinematics wheel rates computation for v_l and v_r, we are unable to enforce constraints on wheel slippage to compensate for slip during driving, which would lead to inaccurate drive arcs.

To overcome this limitation of the differential drive model, Sutoh et al. proposed a workaround which involves artificially compensating wheel slip for position estimation by scaling the input wheel velocities v_r and v_l shown in Equation 3.4 [12]. The value of n is experimentally obtained, and has a value < 1. The scaled wheel velocities v'_r and v'_l is then used to calculate the slip compensated position of the rover with Equation 3.2.

$$\frac{v'_r}{v'_l} = \left|\frac{v_r}{v_l}\right|^n \tag{3.4}$$

3.2 Kinematic Model

Here, we present an elegant solution for incorporating wheel slip in a kinematics model described by Seegmiller in [2].

3.2.1 Frame Information

We refer to [2] where we use a similar method to specify the frame information for MoonRanger rover. The columns in of Table 3.1 represents:

- Column 1: index
- Column 2: frame name
- Column 3: parent frame name
- Column 4: R for revolute about the X/Y/Z axis
- Column 5: Y/N for yes/no
- Column 6: Pose of the frame with respect to its parent frame when joint displacement is zero. Pose comprises solely of translation $\begin{bmatrix} x & y & z \end{bmatrix}^\top$ since orientation is 0, thus excluded.

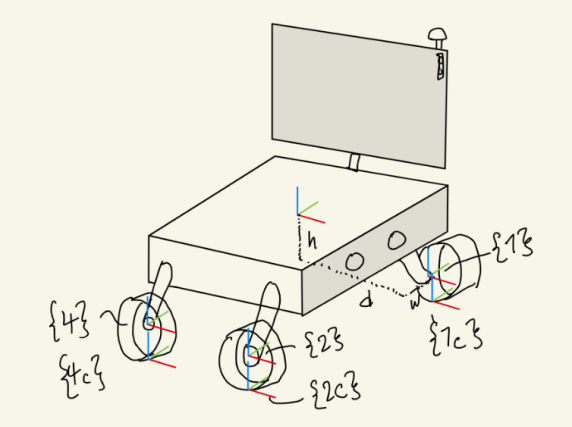

Figure 3.2: MoonRanger Frame Diagram - Image courtesy: Ben Kolligs

Table 3.1: MoonRanger Frame Information

i	Frame	Parent	Type	Actuated	x	y	z
0	body(b)	world(w)					
1	FL	body	RY	Y	d	w	$-h$
2	FR	body	RY	Y	d	$-w$	$-h$
3	BL	body	RY	Y	$-d$	w	$-h$
4	BR	body	RY	Y	$-d$	$-w$	$-h$

in meters, $d = 0.2222$, $w = 0.32195$, $h = 0.0745$

3.2.2 State Vector

We define the rover state vector as

$$q = \begin{bmatrix} \psi_x & \psi_y & \psi_z & x & y & z & \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}^\top$$

where the first 6 elements describe the rover orientation and position, while the last 4 elements describe the joint values (wheel rates).

3.2.3 Wheel Contact Constraints

We account for wheel slippage by enforcing constraints on the relative motion between each wheel and the ground. Here, we provide an example of a single wheel, using a

3-row constraint v_{c1} to describe the wheel contact constraint.

$$A\dot{q} = v_{c1} \tag{3.5}$$

Using this constraint, we can account for wheel slippage directly in the kinematics equation by factoring in a $3n$-row vector - v_c during the calculation of wheel rates or body velocity, where n is the number of wheels on a vehicle. The wheel Jacobian A will be described in the section below.

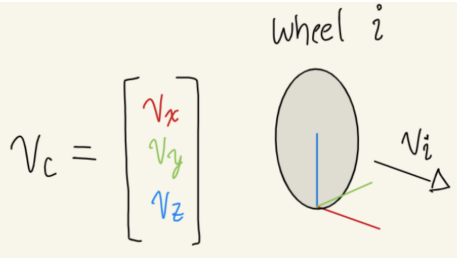

Figure 3.3: Wheel Contact Constraint - Image courtesy: Ben Kolligs

3.2.4 Forming the Jacobians

We refer to Algorithm 2 from [2] to form the wheel Jacobian A. In the case of MoonRanger, there exists only revolute joints at the wheels. This simplifies the formation of the wheel Jacobian, and we would only need to employ Equation 3.6, which describes how the revolute joint for each particular wheel i is populated. Equation 3.7 shows how the joint Jacobian of wheel i is calculated.

$$A = (R_c^w)^T \begin{bmatrix} [\vec{r}_c^{\,w} - \vec{r}_i^{\,w}]_\times^T R_i^w & R_i^w & \Theta_i \end{bmatrix} \tag{3.6}$$

$$\Theta_i = R_i^w\big(:, a(i)\big) \times \big(\vec{r}_c^{\,w} - \vec{r}_i^{\,w}\big) \tag{3.7}$$

Where $a(i)$ is the axis of rotation of the joint i. $[\vec{a}]_\times$ refers to the skew-symmetric matrix of $\vec{a}$. The other notations are standard notations, where R_b^a refers to the rotation matrix from frame b to frame a, and r_b^a is the vector coordinates that describe the translation in x, y, z from frame b to frame a.

We provide an example of how we form the wheel Jacobian for the wheel 1, which values are referenced from Table 3.1:

$$\vec{r}_c^{\,w} = \begin{bmatrix} d \\ w \\ -h \end{bmatrix}, \; \vec{r}_1^{\,w} = \begin{bmatrix} d \\ w \\ -h+r \end{bmatrix}, \; R_1^w\big(:, a(1)\big) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \; R_1^w = R_c^w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & -h & -w & 1 & 0 & 0 & -r & 0 & 0 & 0 \\ h & 0 & d & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ w & -d & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

To form the full wheel Jacobian, where $A \in \mathbb{R}^{10\times 12}$, we stack the the wheel Jacobians vertically from wheel 1 to wheel 4. We assume that the contact frame is static with respect to the rover ($R_c^w = \mathbb{I}^{3\times 3}$), therefore the wheel Jacobian remains constant even when the rover is traversing. The full wheel Jacobian and the corresponding psuedoinverse for MoonRanger can be found in Appendix A.

3.2.5 Navigation Kinematics

The navigation kinematic equation allows us to obtain the body state vector from the wheel rates vector. This equation is used specifically for pose estimation, where we calculate the body state vector from the wheel rates feedback from the wheel encoders.

$$J_{body}^{\dagger} \times (v_c - (J_{joint} \times \Omega)) = v_b = \begin{bmatrix} \psi_x \\ \psi_y \\ \psi_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \tag{3.9}$$

where $J_{joint} \in \mathbb{R}^{12\times 4}$ and $J_{body} \in \mathbb{R}^{12\times 6}$. J_{joint} is formed by taking the first 6 columns of A, and J_{body} the last 4 columns of A. As J_{body} has linearly independent columns,

we can compute its pseudoinverse as $J^{\dagger}_{body} = (J^{T}_{body} \times J_{body})^{-1} \times J^{T}_{body}$.

3.2.6 Actuation Kinematics

The actuation kinematic equation allows to obtain the wheel rates vector from the body state vector. Specifically, we use the desired body velocity v_x and yaw rate ψ_y calculated from a drive-arc to calculate the wheel rates to drive the vehicle.

$$J^{\dagger}_{joint}\left(v_c - (J_{body} \times v_b)\right) = \Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{3.10}$$

where $J_{joint} \in \mathbb{R}^{12\times4}$ and $J_{body} \in \mathbb{R}^{12\times6}$. J_{joint} is formed by taking the first 6 columns of A, and J_{body} the last 4 columns of A. As J_{joint} has linearly independent columns, we can compute its pseudoinverse as $J^{\dagger}_{joint} = (J^{T}_{joint} \times J_{joint})^{-1} \times J^{T}_{joint}$.

3.2.7 Extension to other vehicles

The beauty of the kinematic model is that it is extendable to other vehicles with articulated suspension and explicit steering capabilities. In Figure 3.4, the back right (BR) wheel has a different wheel contact frame from the other wheels, since it is off the ground. The kinematic model shown in this chapter is capable of transforming commanded body velocity to wheel velocities in such scenarios, where a differential model could not. Since the formulation of such a model is generic, we are confident that the slip model presented in the next chapter can be applied to other vehicles with different configuration. Model information of other vehicles with articulated suspensions can be found in Appendix E of [2].

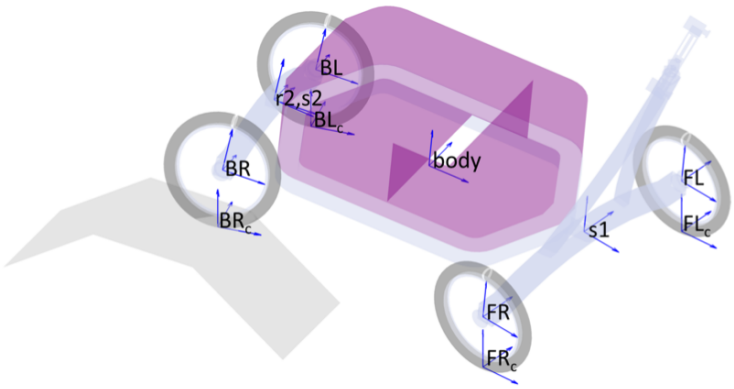

Figure 3.4: Zoë Frame Design - Image Courtesy: Neal Seegmiller [2]

Chapter 4

Slip Estimation

4.1 Slip Model

We split the slip model sequentially to 3 unique layers, where each layer is dependent on the previous layer. This allows us to reduce the complexity of each layer and estimate wheel slippage for different scenarios based on what the rover is executing. An overview of how the model looks like can be seen in Figure 4.1.

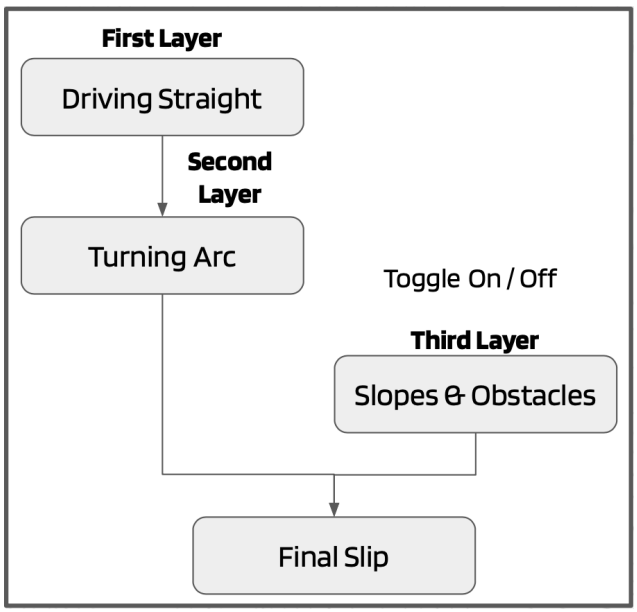

Figure 4.1: Slip Model Overview

4.2 Straight Driving

In the first layer of the model, we perform estimation of wheel slippage at the wheel level, regardless of what type of maneuvers (straight / turning) that the rover is executing. Wheel slippage occur due to the the movement of regolith underneath the wheel, which is a result of traversing soft, non-compact terrain. Theoretically, as the speed of the wheel increases, the amount of slip also increases.

We define the slip ratio of the wheel as:

$$\text{slip ratio} = \begin{cases} 1 - \frac{v}{r\omega} & \text{if } v \geq r\omega; \\ \frac{r\omega}{v} - 1 & \text{if } v \leq r\omega. \end{cases} \tag{4.1}$$

where v is the linear velocity of the wheel, r is the radius of the wheel and ω is the angular velocity of the wheel. Here, we assume that the linear velocity of the wheel is equal to the linear velocity of the rover. A positive slip ratio means that the vehicle is slipping and a negative slip ratio means that the vehicle is skidding.

Since we expect the slip ratio at different commanded speed values to differ, there is a need to determine slip ratios for a range of commanded drive speed values that we expect the rover to drive at. For a particular commanded speed value, the slip ratio for that drive speed can be calculated with encoder feedback that provides the rotational speed of the wheel Ω and with a ground truth linear velocity of a rover. Hereafter, we used Ω to denote the rotational speed of a wheel in rad/s.

After experimentally obtaining slip ratios for selected speeds across the range of possible commanded drive speeds, we could fit a best fit line across the discrete data points to obtain a continuous function that would allow us to estimate slip ratio across different driving speeds. This step is depicted in Equation 4.2 where function f can be the continuous function fitted from the data points or an interpolation function, which is built from 2 vectors of equal length, comprising of the slip ratio at each drive speed. We then convert the slip ratio to slip velocity by multiplying it with the linear velocity of the wheel seen in Equation 4.3, since the kinematics model

described in Section 3.2 takes in slip constraints in m/s.

$$s = f(\Omega r) \tag{4.2}$$

$$v_{slip} = s * \Omega r \tag{4.3}$$

4.3 Turning Arcs

In the second layer of the slip model, we attempt to estimate the additional wheel slippage caused by driving the left and right wheels at different wheel rates when trying to execute a drive arc maneuver. We first define a term - drive ratio, as the ratio between the left and right wheel rate, described in Equation 4.4, which has a value ≥ 1. For straight driving, $\Omega_l = \Omega_r$, therefore drive ratio is $= 1$.

$$\text{drive ratio} = \frac{\max(\Omega_l, \Omega_R)}{\min(\Omega_l, \Omega_R)}. \tag{4.4}$$

We then define a linear scalar factor S, that scales the slip velocity values obtained from the first layer, described in Equation 4.5, which accounts for the additional wheel slippage when performing a turning arc maneuver.

$$\begin{bmatrix} v_l' \\ v_r' \end{bmatrix} = S \begin{bmatrix} v_l \\ v_r \end{bmatrix} \tag{4.5}$$

where v_l and v_r are the slip velocities estimated from the 1st layer.

To determine the values of slip factor that estimate the additional wheel slippage for different drive ratios, we first need to collect encoder feedback wheel rates and the ground truth body velocity of a rover performing turning arc maneuver of various radius over uniform loose terrain. Next, we can determine the slip factor S by finding a value that minimizes the error between the estimated turning arc trajectory computed with the kinematics model described in Section 3.2 and the ground truth trajectory. This can be done through trial and error or by using least square regression to minimize error.

With the experimentally determined slip factors S obtained for a range of selected

drive ratios, we can formulate the calculation of S during driving with a simple interpolation function shown in Equation 4.6.

$$S = f(\text{drive ratio}) \tag{4.6}$$

where f could be a 1 dimensional interpolation function formulated with a list of slip factor values and a list of corresponding drive ratios of equal length or it could also be a best fit curve fitted through the data points.

4.4 Slopes and Obstacles

We decided to treat slopes and obstacles as disturbances to the entire rover, resulting in an error to the estimated body velocity from the Kinematics Model described in Section 3.2. This allows us to use regression methods to predict the error using available features on the rover, such as the current pitch angle, instead of directly trying to estimate the individual wheel slippages caused by overcoming obstacles.

To obtain the body velocity error, we first need to convert the slip ratios for the left and right side wheels estimated from the first two layers into the rover estimated body velocity vector. This can be done by first converting the slip ratios to slip velocities (in m/s) by multiplying the slip ratio with the wheel linear velocity. We then populate the slip velocities for each wheel into the slip constraint vector - v_c, before feeding it into the navigation kinematics equation, described in Section 3.2.5, to obtain the body velocity vector - v_b.

We now define the scalar error in x, which we are trying to predict as:

$$e_x = v_b^x - v_g^x \tag{4.7}$$

where v_g^x is the ground truth x body velocity vector. For a visual example of this error, one can refer to Figure 5.10, which is described in detail in Section 5.3.3.

4.4.1 Gaussian Process Regression

Gaussian process regression (GPR) is a non-parametric, bayesian approach to regression. It has been used to directly predict wheel slippage shown in [22] and [23]. We propose to use GPR to predict the error in body velocity e_x described in the section above. Feature selection and model fitting are described in Section 5.3.3.

We chose GPR over neural networks and other learning methods for body velocity error prediction due to

- Easy tuning owing to little hyper-parameters needed to define a GPR.
- Non-parametric model, requiring no input parameters.
- Provides uncertainty measures over prediction which could be useful to improve robustness of generated maps and paths.

For completion of this section, an equation to describe GP and how e_x is obtained is presented in Equation 4.8. With e_x, we are able to compute the slip compensated body velocity of the rover through Equation 3.3, where $\dot{x}_b = v_b^x - e_x$.

$$e_x = GP(\text{n-features}) \tag{4.8}$$

4.5 Final Model

To summarize the slip model, we present a detailed view of how the layers of the model comes together in Figure 4.2. Layers 1 and 2 take in wheel encoder readings and estimates slip velocity at each wheel through interpolation. These values are populated into the x-component of the wheel contact constraint v_c described in Section 3.2.3 and passed into the navigation kinematics equation to calculate the *initial body velocity estimate*. Layer 3 predicts the *error in body velocity estimate* e_x from layer 1 and 2 using a Gaussian process model. Subtracting the error from the body velocity estimate produces the *final slip compensated body velocity* $\dot{x}_b$ that will be used for position estimation.

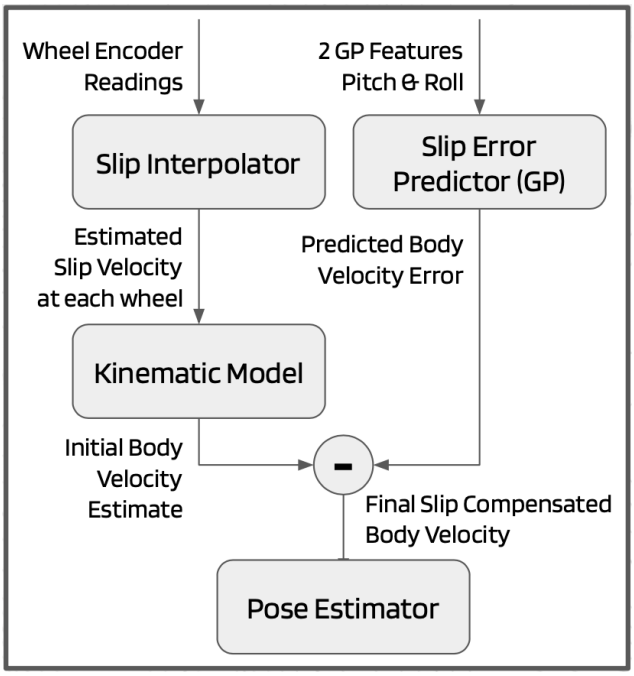

Figure 4.2: Final Slip Model Details

Chapter 5

Slip Compensation Results

In this chapter, we seek to verify the effectiveness of our proposed slip model in achieving accurate localization for lunar micro-rovers by employing MoonRanger's surrogate rover - Morphin.

5.1 Test Setup

We conducted all of our testing with our test rover - Morphin, seen in Figure 5.1. It's main specifications are outlined in Table 5.1. Morphin sensing suite is very similar to MoonRanger, where it is equipped with a STIM 300 IMU, the exact same stereo camera configurations. The only difference is the actuators, where MoonRanger is equipped with Maxon Motors, while we employed GoBilda hobby-grade motors on Morphin to fulfill higher torque requirements when driving on Earth. The encoder feedback from the wheel actuators are coming in at 10Hz. The software stack runs on a NVIDIA Jetson TX2i coupled with a Spacely Carrier Board which is also the setup for MoonRanger. The general autonomy stack runs on NASA cFS (core Flight System) with an interface to a ROS node for controlling the wheel actuators.

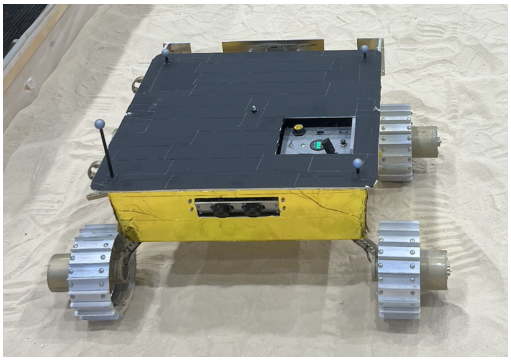

Figure 5.1: Morphin

Item	Value
Mass	26.3 kg
Size (L, W, H)	63, 80, 28 cm
Wheel Diameter	19.1 cm
Wheel Width	8 cm
Grouser Height	1 cm
Wheel Base	44 cm
Wheel Track	60 cm

Table 5.1: Morphin's Specifications

Testing is done at CMU's Planetary Robotics Highbay within a confined sand pit, measuring 5m by 5m. The sand pit is filled with Quikcrete playsand. An image of the test environment can be seen in Figure 5.2.

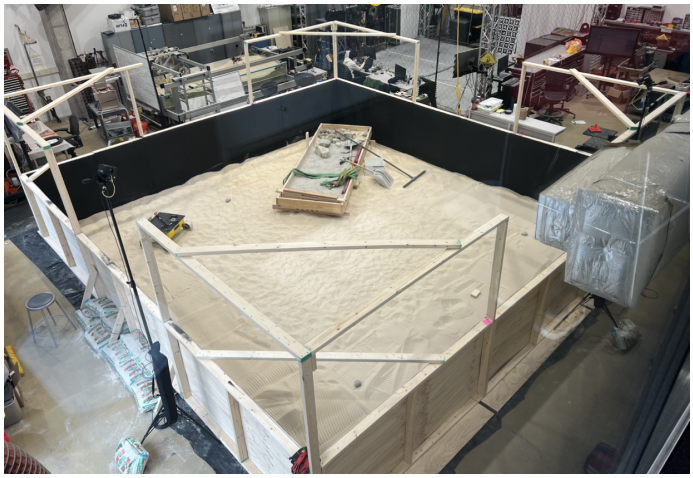

Figure 5.2: MoonYard Testing Area at CMU

5.1.1 Wheels Synchronization

We initially ran into wheel synchronization issues on Morphin due to the wheel actuators running on an open loop configuration. There is availability of encoder feedback for each wheel actuator, but there were not used to synchronize wheel rates.

This led to severe de-synchronization of the wheels and noticeable deviation from the commanded trajectory. The effect is pronounced at lower wheel rates ($1 - 3$ cm/s) as the motor drivers are unable to drive effectively at such low rates.

To overcome the wheel synchronization issue, we routed the encoders from the wheel actuators directly into the motor controllers (Roboteq's SDC 2160) and selected closed loop velocity control on the motor controllers to obtain accurate wheel synchronization.

5.1.2 Ground Truth

We employed a motion capture system to obtain the ground truth position and orientation of Morphin so that we could evaluate the effectiveness of our proposed model. Our setup consist of 4x mobile Bonita Vicon 10 cameras, which we deployed around the 4 edges of the sand pit. The ground truth position and orientation of Morphin is then retrieved from the Vicon cameras through the `vicon_bridge` ROS (Robot Operating System) package.

5.2 Soil Preparation

Soil preparation is essential to ensure consistency in the terrain across different runs. Uneven or no soil preparation can potentially lead to large variances in the results obtained. Thus, we conducted soil preparation in MoonYard before every run.

We first marked the area that the rover will be traversing on, which allowed us to focus our soil preparation in a confined area. Using a shovel, we dug about $8 - 10$ cm into the ground and flipped the sand on the spot, in an attempt to loosen the sand (Figure 5.3). Lastly, we used a rake to gently flatten the terrain. Sample images of the soil before and after raking can be seen in Figures 5.4 and 5.5 respectively.

Figure 5.3: Soil Preparation Action

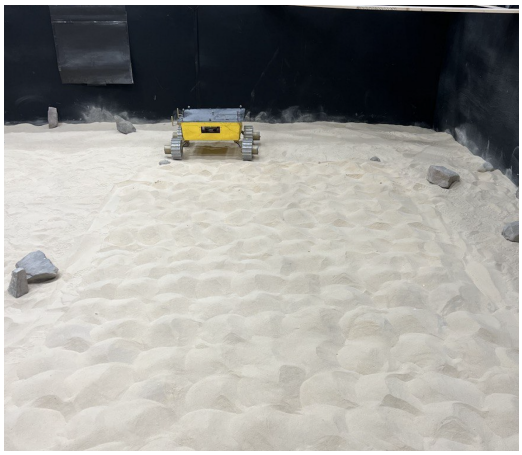

Figure 5.4: Terrain before raking

Figure 5.5: Terrain after soil preparation

5.3 Slip Model

5.3.1 Straight Driving

To obtain the slip ratio for Morphin at different speed, we drove Morphin at its expected driving speed of $[2, 7]$ cm/s, with an interval of 1cm/s to collect 7 data points.

We logged the ground truth speed and encoder feedback of Morphin for every run. Our post-process of the data includes averaging of the ground truth data (Vicon) for a better velocity estimate, since the vicon data is logged at 100Hz. A sample of the post-processed body velocity data looks can be found in Figure 5.6.

We collected 3 runs for every drive speed, averaged the slip ratio and plotted a best

fit line to determine the slip ratio at each speed. A linear relationship was sufficient to deduce the slip estimate for each rover speed of intervals 1 cm/s. The best fit line can be found in Figure 5.7 and slip ratios can be found in Table 5.2. We believe that this linear relationship is caused by driving at low speeds coupled with grousers that penetrate deep into the terrain.

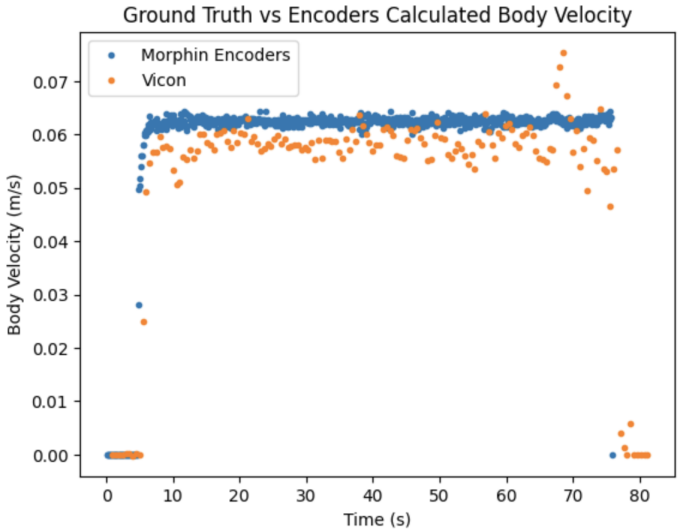

Figure 5.6: 6cm/s Slip Data from Morphin and Vicon

Drive Speed	2	3	4	5	6	7	cm/s
Slip Ratio	5.12	5.42	5.72	6.03	6.33	6.63	%
Velocity Compensated	0.10	0.16	0.23	0.30	0.38	0.46	cm/s

Table 5.2: Slip - Straight Driving

5.3.2 Turn Arcs

To obtain the slip factor S that accounts for additional wheel slippage due to difference in left and right wheel velocities, we commanded Morphin to perform a series of drive arcs with radius varying from $[1.0, 3.0]$ m at an interval of 0.5m. After collection of the ground truth trajectory and encoder values, we tune the slip factor S by hand, until the curvature of the estimated trajectory lines up with the ground truth. We

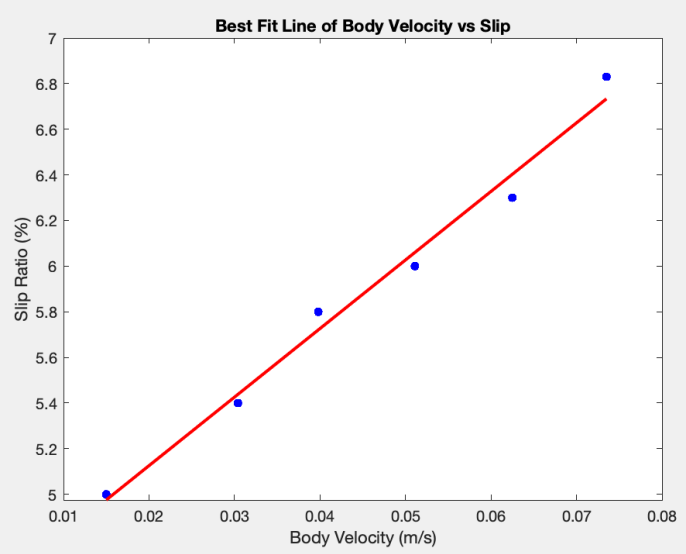

Figure 5.7: Body Velocity vs Slip best fit line

also ensured that the total distance estimated by the slip model matches up with the ground truth distance traveled by Morphin. This effect is reflected when the arc length of the 2 trajectories matches up.

Figure 5.8 shows a few turning arcs that we performed and how applying a slip factor allows the estimated trajectory to align with the ground truth trajectory. The baseline is represented in green which showcases what would be the estimated trajectory if slip factor was not applied. The 1m drive arc did not match out properly due to high slip at the outer wheels when performing a tight drive arc. We believe that lateral slip caused the slight mismatch in the slip-compensated estimated trajectory and the ground truth trajectory. One should also notice that the arc length of the 2 trajectories (orange and blue) matches up, which confirms that wheel slippage is being compensated at the right amount.

The final slip factor values S that described the results in Figure 5.8 can be found in Table 5.3. We have also extrapolated the drive ratio to 2.0 and limited the max slip factor to 2.0 to account for noise and limit the max value of the slip factor. The action space on MoonRanger is configured to command drive arcs with radius $[1.0, 3.0]$ m. We stopped our testing at drive radius > 3.0m since the value of drive ratio is tending towards 1, and we assumed that a linear interpolation can derive the

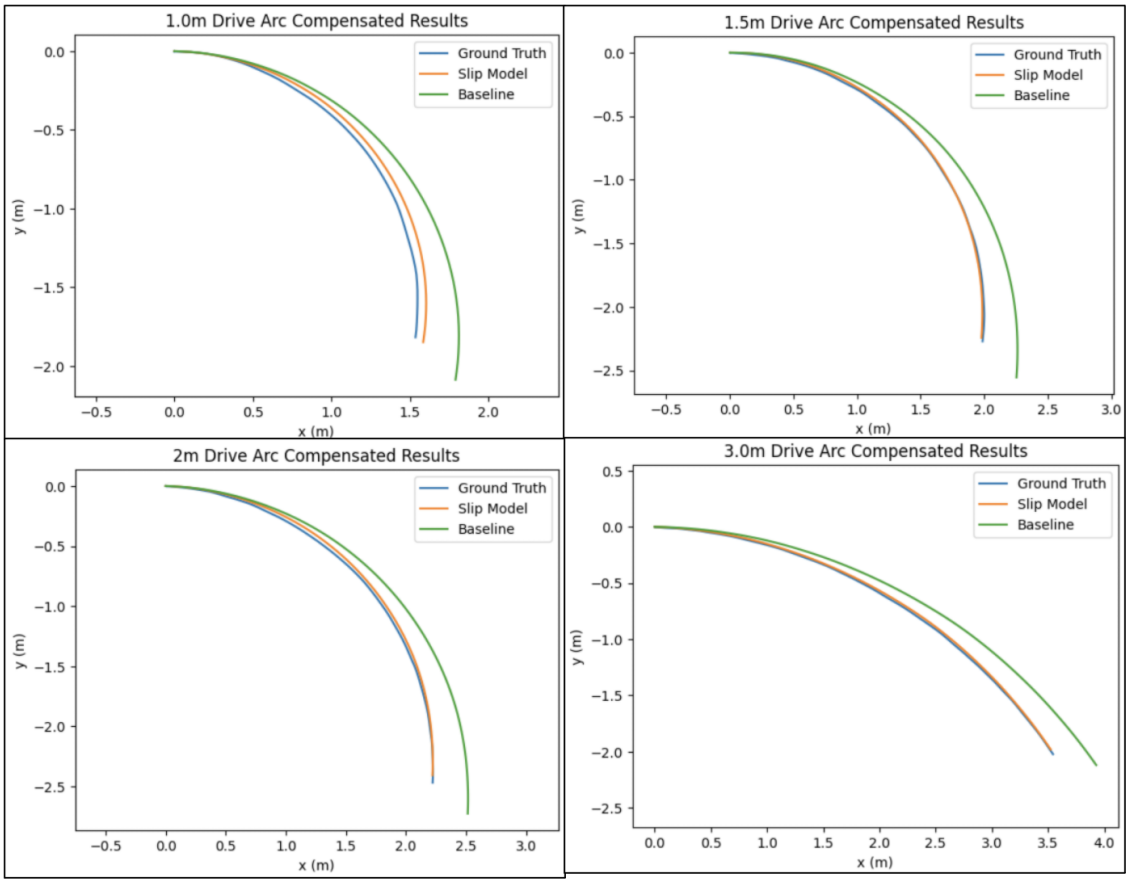

Figure 5.8: Comparison between slip factor compensated position estimation vs Baseline

correct slip factor thereafter.

Arc Radius (m)		1.0	1.5	2.0	2.5	3.0	straight
Drive Ratio	2.0000	1.8570	1.5000	1.3529	1.2727	1.2222	1.000
Slip Factor - S	2.00	1.90	1.65	1.55	1.50	1.40	1.00

Table 5.3: Slip Factor for various Turn Arcs

5.3.3 Slopes and Obstacles

We conducted 6 runs, driving Morphin over obstacles and slopes to obtain the training data for GPR. An example of such run can be seen in Figure 5.9 (left) which shows

the crest and trench that Morphin has driven over.

Figure 5.9: Morphin going over a crest and trench (left) and embedded rocks (right)

This sample run produces a body velocity plot seen in Figure 5.10. We see a near constant slip compensated body velocity estimated from the wheel rates. This is expected since we are estimating solely from the encoders values which are near constant due to a constant commanded speed. However, the ground truth body velocity measured by Vicon is varying due to Morphin traversing over obstacles, in this case going over slopes. When going over the slope initially, we see a positive difference between the ground truth and estimated body velocity, which signifies that Morphin is slowing down due to higher wheel slippage. While going downhill on the slope, there is a negative difference between the ground truth and estimated body velocity, signifying that Morphin slipped more. The difference between the 2 curves at every time-step is the prediction error (e_x) that we are using as the training input.

5.3.4 Feature Selection

The only available features that could predict Morphin is traversing over obstacles is data from the IMU. We pre-processed the 6 obstacles and slopes run, aligning sensor timestamps and trimming any outliers. We plotted the body velocity error with 9 different features, namely Morphin's current orientation, angular velocity and linear acceleration to determine if there exist any statistical significance in the features.

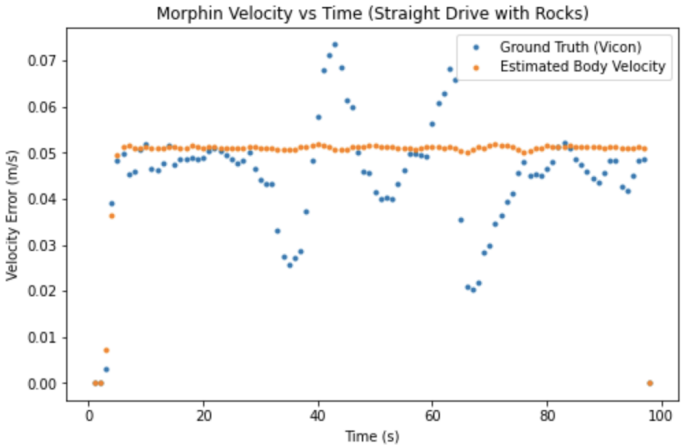

Figure 5.10: Difference in Estimated and Actual Body Velocity due to going over Slopes

We noticed a strong relationship in pitch in predicting the body error velocity, shown in Figure 5.11. The general trend showed that a negative pitch resulted in a negative error (Morphin speeding up due to additional slip) and a positive pitch resulted in positive error. Roll did not have that strong of a relationship but does seem to help in weighting the occurrences, where in general, they are more negative error data points than positive error, which make sense from an intuitive standpoint. As such, we decided to keep roll as a feature, which would theoretically help in making better prediction (more independent features produces better prediction). Additionally, orientation of Morphin is readily available and filtered at every timestep, simplifying the implementation. Note that we discarded angular velocities and linear accelerations as features due to noisy measurements.

5.3.5 Fitted Model

Model Fitting was done with *scikit-learn* machine learning toolbox. The input features were normalized using `StandardScaler` and fitted using `GaussianProcessRegressor`.

We chose to use RBF (Radial Basis Function) kernel since it can better represent a non-linear function through varying the length scale of the kernel and added a white noise kernel for better generalization. Kernel selection is a huge topic by itself, and

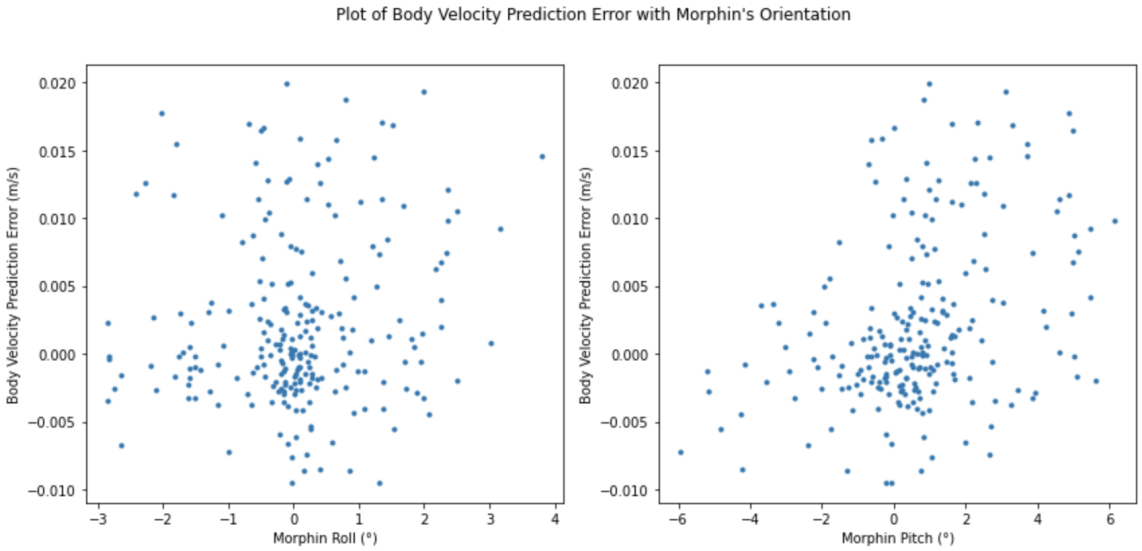

Figure 5.11: Body Velocity Prediction Error against Morphin's Roll and Pitch

we would not go into detail in this thesis.

To fit the model, we split the data points, 230 points (Figure 5.11), into a training and test set at a ratio of 0.8 : 0.2. The fitted model is shown in Figure 5.12. Views from other angles can be found in Appendix B. From the shape of the GP model, we can deduce that the prediction would be a positive error (higher wheel slippage) at positive pitch which was implied in the data points seen in Section 5.3.4. The model has a slight concave shape which make sense intuitively, where a combination of roll and pitch would lead to possibly larger wheel slippage.

The solid surface in the middle describes the mean prediction, while the top and bottom surface shows 1 standard deviation (std) of prediction above and below the mean. Some points do lie outside 1 std but are within 3 std. We evaluated the prediction accuracy with the testing set which is approximately 45 points. The prediction error evaluated is `0.001297` m/s which is **2.6**% of the average Morphin driving speed of 0.05m/s.

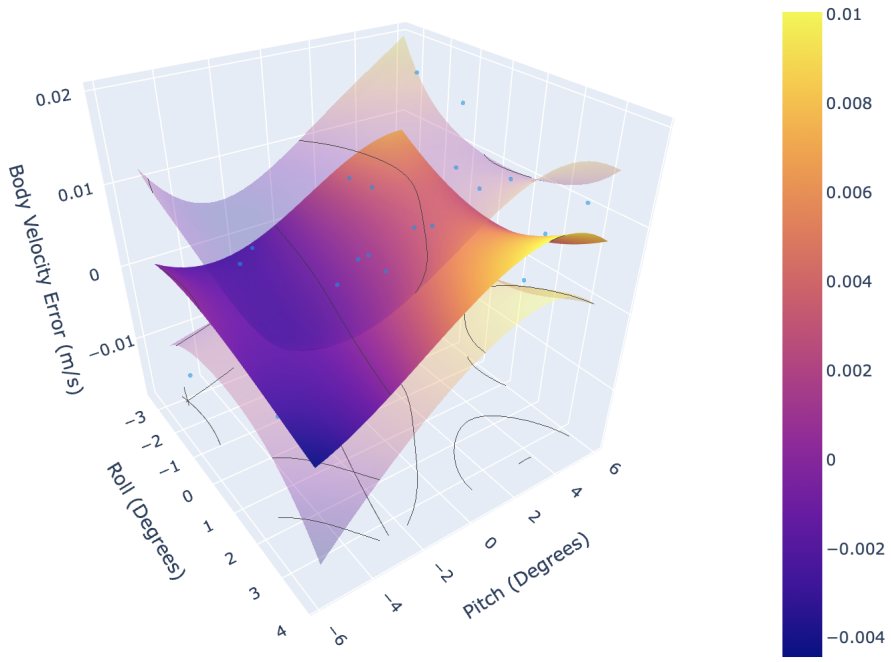

Figure 5.12: Final GP Model

5.4 Pose Estimation Results

To verify the estimation accuracy of the slip model, we performed 2 separate runs around MoonYard, each completing a loop. The estimated distance traveled by Morphin for each loop is approximately 14.0m.

We used the Absolute Trajectory Error (ATE) metric to benchmark the result from the ground truth [29]. The root mean squared error (RMSE) is usually used to evaluate the quality of the trajectory, shown in Equation 5.1 [30]. The original method proposes for us to associate the points from the model trajectory to the ground truth trajectory, based on the timestamp of the trajectory and the closeness of each point. We decided to skip this step since we are confident of the alignment as the data for the ground truth and model are bagged together. We then align the trajectory by finding the rotation matrix and translation vector that best aligns the model and ground truth trajectory. The translation error at each time-step is then calculated. We leveraged and made modification to interface ATE calculation code [31] to our trajectories.

$$e_{ATE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \|\mathbf{x}_i - \hat{\mathbf{x}}'_i\|_2^2} \tag{5.1}$$

5.4.1 Run 1 - Clockwise

The results for the first loop is shown in Figure 5.13, with ATE statistic reported in Table 5.4 In this run, we did not put any obstacles. The green line is the baseline, which shows the estimated trajectory without slip compensation. Overall, we see that the ground truth (blue) and slip compensated (orange) matches up pretty nicely. For this run, GPR was not active.

Trajectory Type	Slip Model	Baseline
RMSE	0.070606	0.318154
Mean	0.063075	0.317880
Median	0.059574	0.317597

Table 5.4: ATE Statistics for Run 1

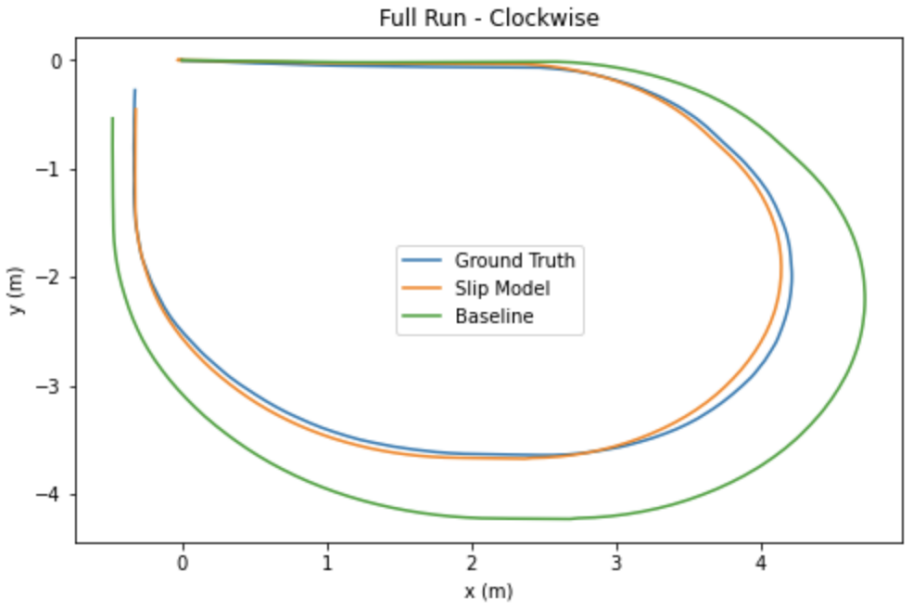

Figure 5.13: Position Estimation Results from Run 1 in Moonyard

5.4.2 Run 2 - Anticlockwise

The results for the second loop is shown in Figure 5.13, with ATE statistic reported in Table 5.4 In this run, we did an anti-clockwise drive of MoonYard, to evaluate slip compensation in the other direction. The green line is the baseline, which shows the estimated trajectory without slip compensation. Overall, we see that the ground truth (blue) and slip compensated (orange) matches up pretty nicely. For this run, GPR was not active.

Trajectory Type	Slip Model	Baseline
RMSE	0.063826	0.266379
Mean	0.052748	0.257564
Median	0.039723	0.267472

Table 5.5: ATE Statistics for Run 2

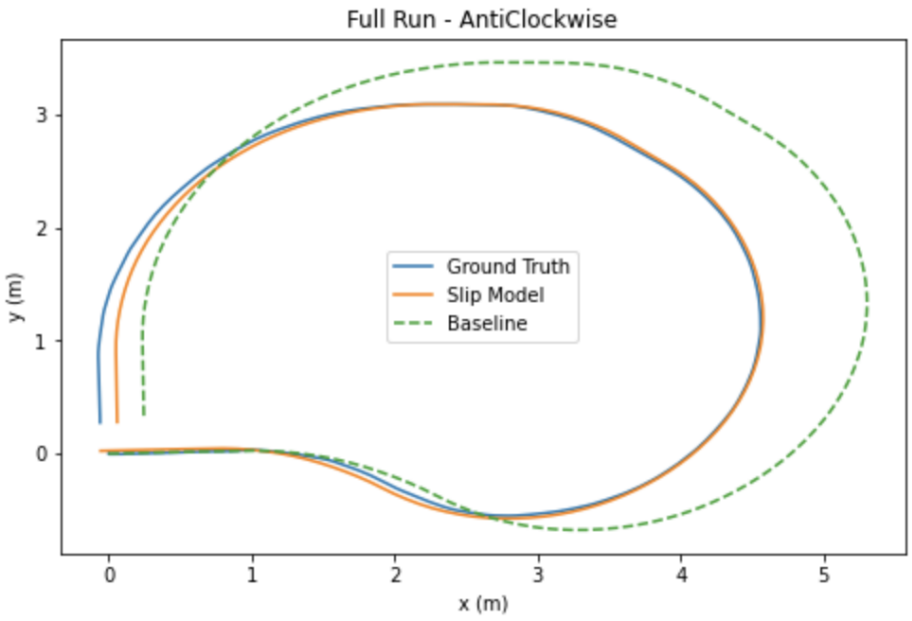

Figure 5.14: Position Estimation Results from Run 2 in Moonyard

5.4.3 Run 3 - Anticlockwise 2 loops

We attempted to do 2 full loops around the MoonYard with some overlap in the path taken, hoping to detect any drifts. The results for the third loop is shown in Figure 5.15, with ATE statistic reported in Table 5.6. In this run, we did not put any obstacles. The green dashed line is the baseline, which shows the estimated trajectory without slip compensation. Overall, we see that the ground truth (blue) and slip compensated (orange) still matches up pretty nicely after 2 full loops and a directional change at the end of the loop..

Trajectory Type	Slip Model	Baseline
RMSE	0.066989	0.260709
Mean	0.062455	0.251837
Median	0.059229	0.253398

Table 5.6: ATE Statistics for Run 3

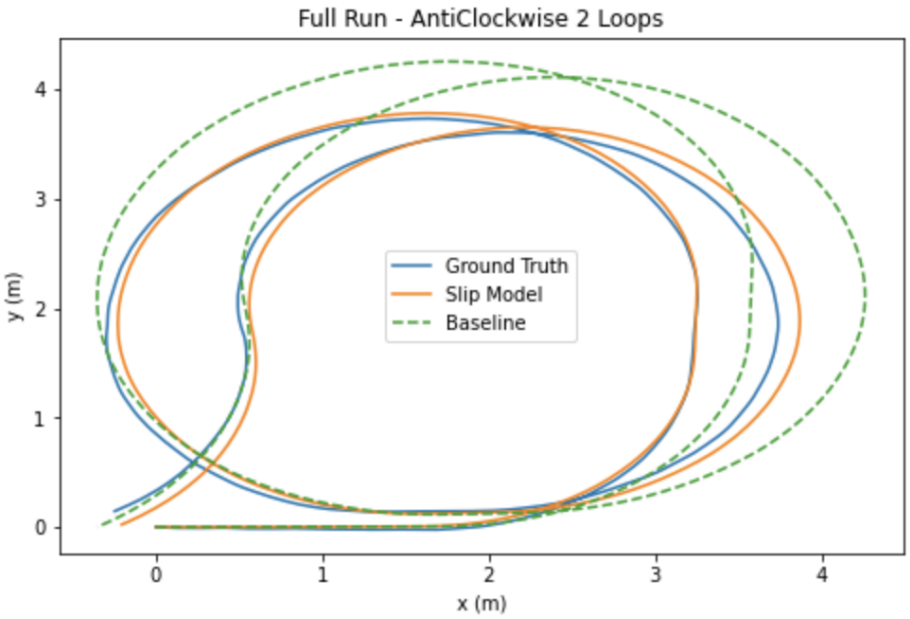

Figure 5.15: Position Estimation Results from Run 3 in Moonyard

5.4.4 Run 4 - Clockwise with Obstacles

For the fourth loop, we places embedded rocks at the start of the run, along with a crest and trench at the middle of the run. GPR was active for this run, which result is depicted in red in Figure 5.17. Here, we see that the position estimation with GPR performs slightly better than just the slip model without GPR (orange). The trajectory plots can be found in Figure 5.16 and ATE statistics in Table 5.7.

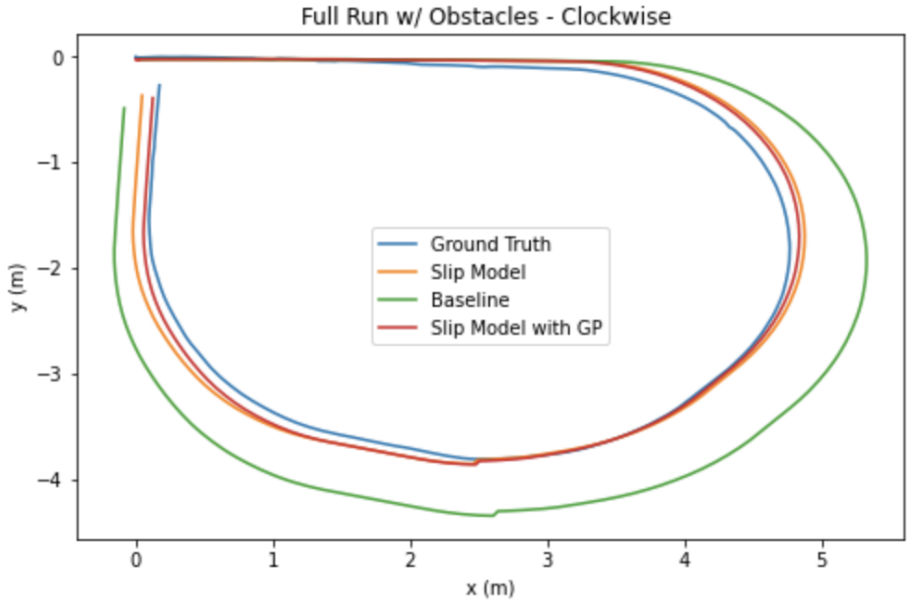

Figure 5.16: Position Estimation Results from Run 4 in Moonyard with Obstacles

The RMSE error with the slip model using GPR is lower than without, showing that the GP model is performing well with compensating for additional wheel slippage. Both Run 1 and Run 4 are similar in trajectory, but we see that the RMSE in Table 5.7 is larger than Table 5.4. This is expected since Run 4 comprises of obstacles, which causes additional wheel slippage not accounted for in the first 2 layers of the slip model.

Trajectory Type	Slip Model	Slip Model (GPR)	Baseline
RMSE	0.111354	0.083232	0.352198
Mean	0.104932	0.076326	0.343781
Median	0.115733	0.080302	0.363103

Table 5.7: ATE Statistics for Run 4

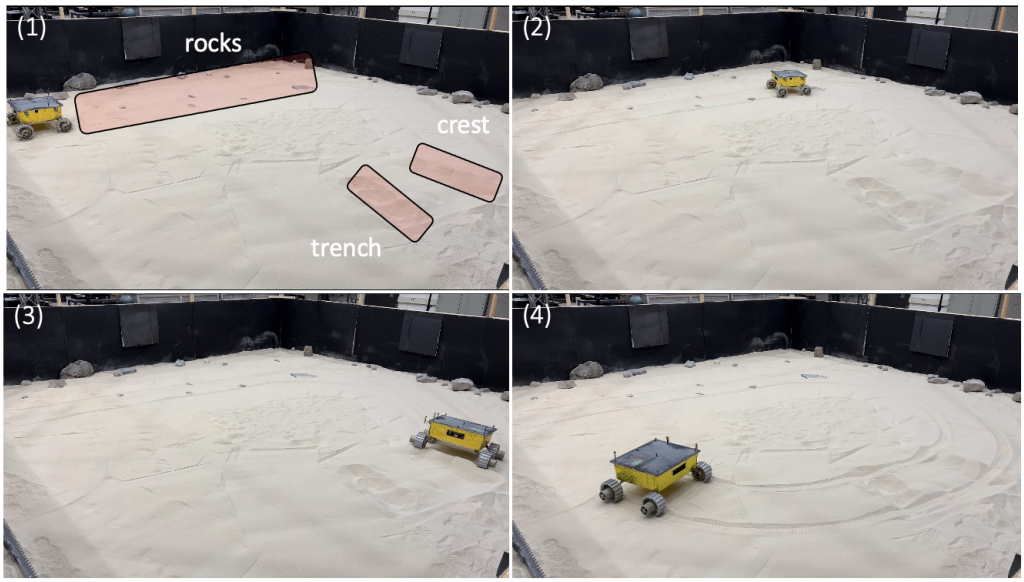

Figure 5.17: Images of Run 4 in Moonyard with Obstacles

Chapter 6

Path Following

6.1 Following Turning Arcs

It is crucial for MoonRanger to follow the turning arc accurately that the planner commands. Since MoonRanger has a short range and unable to detect obstacles > 2 meters away, it has to make a tight turning arc to avoid the obstacle. If MoonRanger is unable to follow the turning arc, it would inch closer to the obstacle and the planner would flag that MoonRanger has no possible path forward.

Figure 6.1: Morphin avoiding an obstacle

6.2 Scaling Yaw Rates

We originally proposed to forward calculate the wheel slippage ratio through our slip model described in Chapter 4, using Equation 4.2 and 4.5 with the commanded wheel rates. With the wheel slippage ratio, we obtain the wheel slip velocity at each wheel in m/s by multiplying s with the linear speed of the wheel $\Omega * r$. We can then construct the slip constraint vector v_c and perform actuation kinematics in Section 3.2.6 to generate the slip compensated wheel rates.

However, through our preliminary testing with a surrogate rover, we are unable to get it to track the commanded turning arc accurately at tight turning arcs up to 2.5m in radius.

Since the body velocity of MoonRanger commanded during turns is always a constant at 0.05m/s, we proposed to instead correlate the commanded yaw rate and actual radius taken by the rover in our test runs to effectively follow turning arcs accurately. We refer to Table 6.1 and the toy examples for a better understanding of the correlation.

Actual Radius	1.5	2.1	2.6	m
Commanded Radius	1.0	1.5	2.0	m
Commanded Yaw Rate	0.050	0.0333	0.025	rad/s

Table 6.1: Radius of Arc taken and Commanded Yaw Rates

Without any compensation, the commanded heading for driving a 1.5m turn arc would be computed as $\frac{v_b}{r} = \frac{0.05}{1.5} = 0.0333$. However, we know from our table that to drive an actual radius of 1.5m requires a commanded heading of 0.050 rad/s. As such the scaling factor Y can be computed as

$$Y_{\text{1.5m turn}} = \frac{\text{Required Yaw Rate}}{\text{Commanded Yaw Rate}} = \frac{0.050}{0.0333} = 1.5$$

By experimentally driving a rover at different yaw rates, we can populate a lookup table with scaling factor Y for different commanded turn radius. During run-time, we lookup the scaling factor based on the commanded radius and scale the commanded

yaw rate accordingly to achieve better following of turning arcs.

6.3 Path Following Results

We commanded Morphin to perform 4 turning arcs at 4 different headings, with the ground truth trajectory shown in Figure 6.2. We then attempt to fit a circle around the path taken by Morphin to determine the actual radius of turn taken. A sample of this result can be seen in Figure 6.4, where a commanded heading of 0.05 rad/s led to a turn radius of 1.55m. Results of other turning arcs are shown in Table 6.2.

Commanded Yaw Rate	0.075	0.050	0.0375	0.025	rad/s
Best Fit Radius	1.095	1.549	2.132	2.381	m

Table 6.2: Fitted Arc Radius

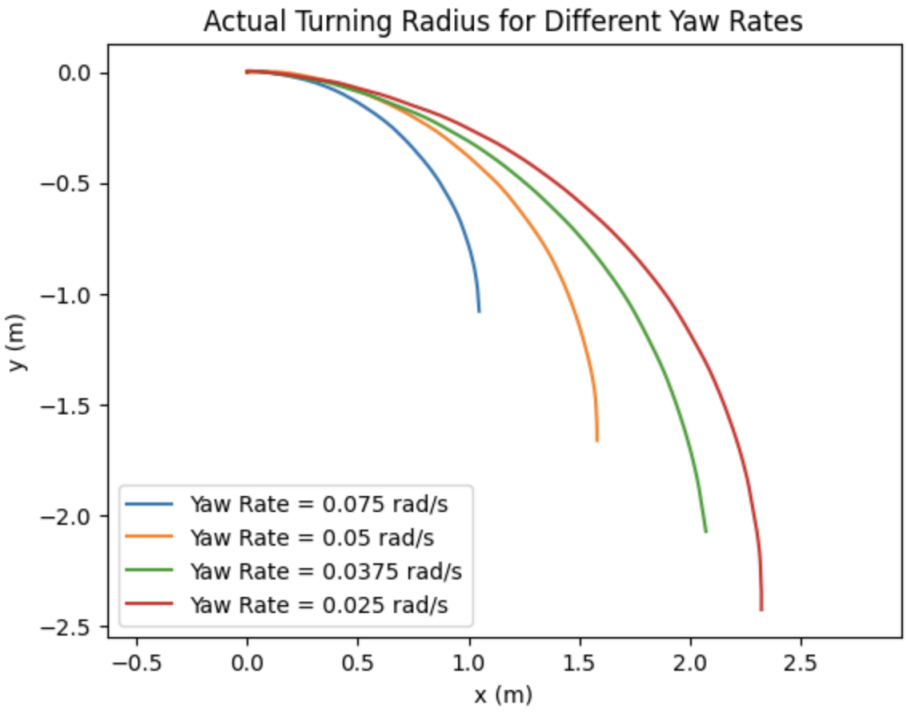

Figure 6.2: Actual Radius Executed vs Yaw Rate

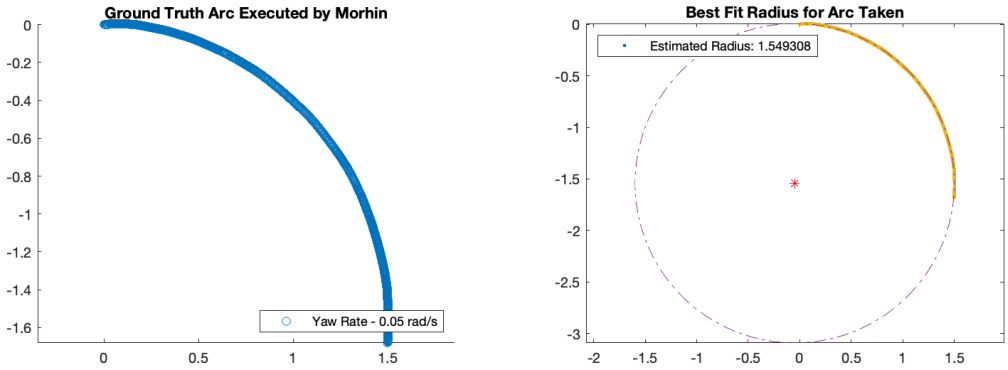

Figure 6.3: Fitting Drive Arc Radius

We plotted a best fit curve with the data obtained in Table 6.2 in an attempt to find the required yaw rate to produce a desired turn radius. Next, we compute the scaling factor required to scale the commanded yaw rate calculated by the kinematic model to the required yaw rate to achieve the desired turning radius. The yaw rate scaling factors are presented in Table 6.3 and stored as a look up table on Morphin to enable better path following when executing drive arcs.

Desired Turn Radius	1.0	1.5	2.0	2.5	m
Commanded Yaw Rate	0.050	0.033	0.025	0.020	rad/s
Yaw Rate Scale Factor	1.59	1.66	1.5	1.15	
Required Yaw Rate	0.0795	0.0554	0.0375	0.0230	rad/s

Table 6.3: Yaw Rate Scale Factor for Different Turn Radius

Lastly, we executed multiple successive turning arcs of radius 1.0m to show the improvement of scaling the yaw rate to achieve improved path following. In Figure 6.5, we see that Morphin takes an approximately 1.5m turning arc radius even though the desired turning arc radius is 1m. However, after scaling the yaw rate, Morphin completes turning arc radius of approximately 1.05m , seen in Figure 6.6.

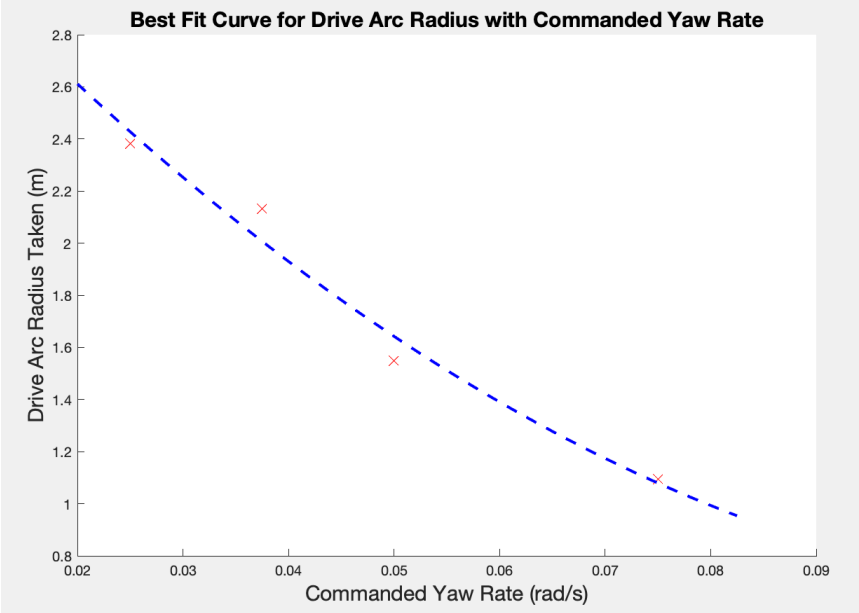

Figure 6.4: Fitting Drive Arc Radius

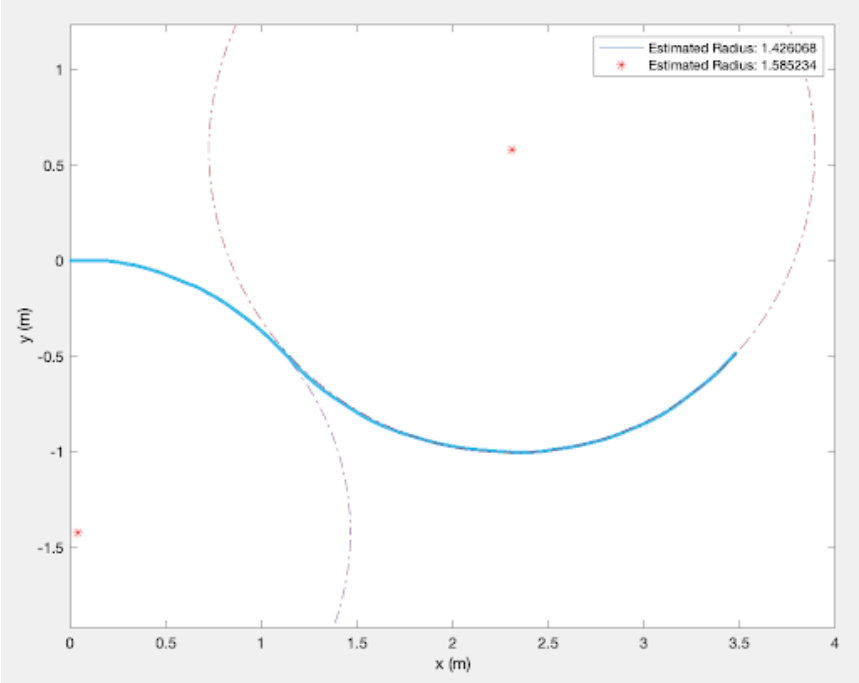

Figure 6.5: Ground Truth Position of Morphin driving at a Commanded Yaw Rate of 0.05 rad/s

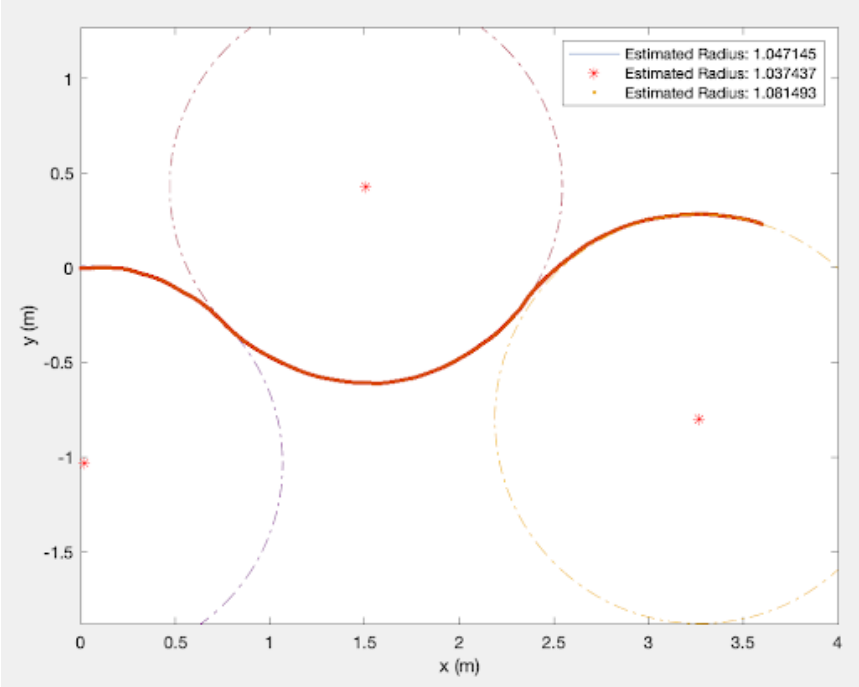

Figure 6.6: Ground Truth Position of Morphin driving at a Required Yaw Rate of 0.08 rad/s

Chapter 7

Conclusions

In this thesis we discussed the different methods that one can estimate slip on a wheel vehicle, and the limitations of such methods on a micro lunar rover. We outlined the kinematic model described in [2] and reason why it is a better kinematic model for slip compensation over differential drive model. We then proposed a new slip model comprising of 3 different layers and address how each layer accounts for wheel slippage at different drive scenarios. We then touch on the need of better path following for micro lunar rovers and proposed a simplistic method to scale commanded heading to achieve tighter turn arcs. Lastly, we presented how our slip model parameters were derived, along with its results.

7.1 Limitations

We observed during our testing that there were significant errors in the position estimates when Morphin was up against huge rocks that cannot be surmounted. Such a scenario is shown in Figure 7.1, where a steep vertical face rock is embedded into the ground causing Morphin to experience $90-100\%$ wheel slippage when it is unable to surmount the rock. The position estimates calculated from the slip model and wheel encoder values continues to increase but Morphin is not physically moving forward.

Another limitation can be found in the kinematics model described in Section 3.2. In

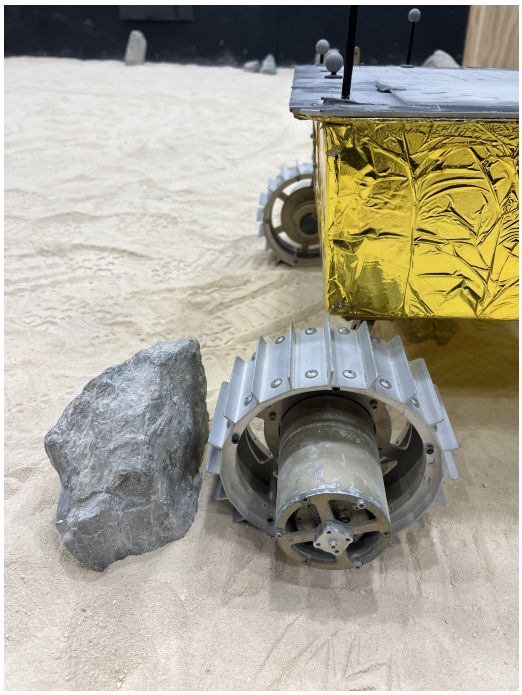

Figure 7.1: Morphin's wheel up against a rock that is too steep to surmount

the derivation of the body Jacobian, we assumed wheel contact frames do not change and its parallel to the ground. However, in reality, the wheel contact frames are constantly changing when MoonRanger is traversing over slopes and obstacles. This could lead to additional wheel slippage, larger forces on wheels and drift in vehicle heading angle [32].

7.2 Achieving Mission Objective

Described in Section 2.1, MoonRanger needs to accurately estimate its position in Trek 4 so that it can return back to the lander and downlink the data collected during the drive to Earth. It is difficult to quantify the localization error accumulated when MoonRanger performs a 1km drive on the Moon, but we noticed a localization error in the range of $2-3\%$ of the total distance traveled by Morphin during our testing. Coupling in additional uncertainty in obtaining accurate slip model parameters on the Moon and traversing over complex terrain, we believe that MoonRanger's localization error should be closer to 10% of the total distance traveled. This is potentially

acceptable as the max communication range of the lander is greater than 100m, but MoonRanger could face significant throttling of downlink speeds since it is so far from the lander.

However, we believe that utilizing the slip model we proposed, MoonRanger has a higher likelihood of mission success compared to when a slip model is not used. Additionally, we saw enhancement in stitching of meshes in the mapper due to better position estimates and improvement in following planner commanded trajectories leading to better obstacle avoidance.

7.3 Estimating Slip Parameters on the Moon

Reliable performance of the slip model depends heavily on obtaining accurate slip parameters for the 3 slip layers. In this section, we discuss potential methods that could provide best estimates of the slip parameters for MoonRanger, which is operating on lunar regolith with lunar gravity.

Slip parameters can potentially be estimated by conducting slip tests on a lunar simulant such as the GRC-1, designed by Oravec et al. [33]. This would also require a vehicle that has the same mobility to MoonRanger, and a mass on the Earth that is equal to the mass of MoonRanger on the Moon. This reduction in weight is often difficult, but can be achieved by using a gravity offloading system, which involves a helium balloon to reduce the weight of the vehicle, pictured in Figure 7.2. This might not be practical to test at CMU since it requires a huge amount of GRC-1, which is both hazardous to breathe and expensive to obtain. An alternative could be to conduct the gravity offloaded test at NASA Glenn Research Center which has a huge sand pit filled with GRC-1.

In the event where the above gravity offload tests are not possible, we intend to derive and update the slip parameters when MoonRanger is physically on the Moon. This is accomplished by performing a series of drive maneuvers, driving MoonRanger at key speeds and turning radius described in Section 5.3.1 and 5.3.2. We plan to perform offline VO on the stereo images pair which are logged at 1Hz during the drives to obtain ground truth information needed for tuning of the slip parameters, since VO

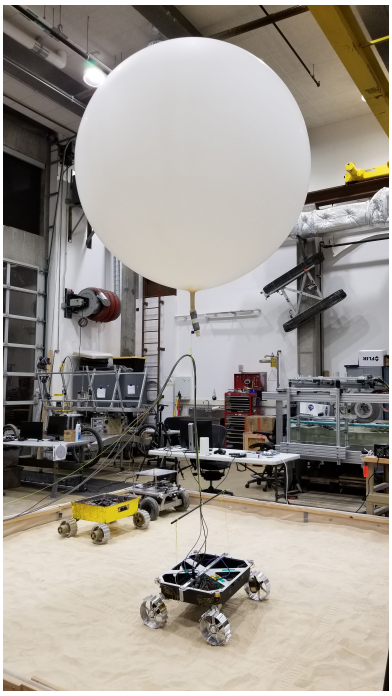

Figure 7.2: Gravity Offloading on MoonRanger's Surrogate Rover - Terramule

is unable run autonomously reliably. The stereo images pair would be retrieved from MoonRanger by downlinking the images to Earth using the lander. After tuning the slip parameters with the ground truth information calculated from the stereo images pair, we would send a patch to MoonRanger and update the slip parameters.

7.4 Future Work

There are 2 potential avenues to improve the slip model and localization accuracy proposed in this thesis, which address the limitations described earlier in this chapter. Toupet et al. shown how wheel contact angles could be derived through suspension angles and how incorporating them in the kinematic model improves rover mobility [32]. Though MoonRanger does not have any suspensions to derive wheel contact angles, we believe that some information of the terrain profile can be extracted from the meshes built by the mapper, which can be used to estimate the wheel contact angles.

Next, a simplified version of VO could be implemented on MoonRanger to detect high slip cases ($> 80\%$ slip). This version of VO need not be very accurate since it is not used to provide position estimates, but rather to detect high wheel slippage, which would prevent rover entrapment and reduce accumulation error of position estimates. Additionally, it can be configured to run at longer intervals between each iteration, reducing computation requirements.

Appendix A

MoonRanger Symbolic Jacobians

$$A_{moonranger} = \begin{bmatrix} 0 & -h-r & -w & 1 & 0 & 0 & -r & 0 & 0 & 0 \\ h+r & 0 & l & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ w & -l & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -h-r & w & 1 & 0 & 0 & 0 & -r & 0 & 0 \\ h+r & 0 & l & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -w & -l & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -h-r & -w & 1 & 0 & 0 & 0 & 0 & -r & 0 \\ h+r & 0 & -l & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ w & l & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -h-r & w & 1 & 0 & 0 & 0 & 0 & 0 & -r \\ h+r & 0 & -l & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -w & l & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.1}$$

$$J_{body} = \begin{bmatrix} 0 & -h-r & -w & 1 & 0 & 0 \\ h+r & 0 & l & 0 & 1 & 0 \\ w & -l & 0 & 0 & 0 & 1 \\ 0 & -h-r & w & 1 & 0 & 0 \\ h+r & 0 & l & 0 & 1 & 0 \\ -w & -l & 0 & 0 & 0 & 1 \\ 0 & -h-r & -w & 1 & 0 & 0 \\ h+r & 0 & -l & 0 & 1 & 0 \\ w & l & 0 & 0 & 0 & 1 \\ 0 & -h-r & w & 1 & 0 & 0 \\ h+r & 0 & -l & 0 & 1 & 0 \\ -w & l & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

$$J_{joint} = \begin{bmatrix} -r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -r & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.3}$$

$$J_{joint}^{\dagger} = \begin{bmatrix} -\frac{1}{r} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{r} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{r} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{r} & 0 & 0 \end{bmatrix} \tag{A.4}$$

$$J^{\dagger}_{body} = \frac{1}{4}\begin{bmatrix} 0 & 0 & a & 0 & 0 & -a & 0 & 0 & a & 0 & 0 & -a \\ 0 & 0 & -b & 0 & 0 & -b & 0 & 0 & b & 0 & 0 & b \\ -c & d & 0 & c & d & 0 & -c & -d & 0 & c & -d & 0 \\ 1 & 0 & -e & 1 & 0 & -e & 1 & 0 & e & 1 & 0 & e \\ 0 & 1 & -f & 0 & 1 & f & 0 & 1 & -f & 0 & 1 & f \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

where $a = \dfrac{1}{w}$

$b = \dfrac{1}{l}$

$c = \dfrac{w}{l^2 + w^2}$

$d = \dfrac{l}{l^2 + w^2}$

$e = \dfrac{rh^2 + hr^2}{hrl}$

$f = \dfrac{rh^2 + hr^2}{hrw}$

Appendix B

GP Model Alternate Views

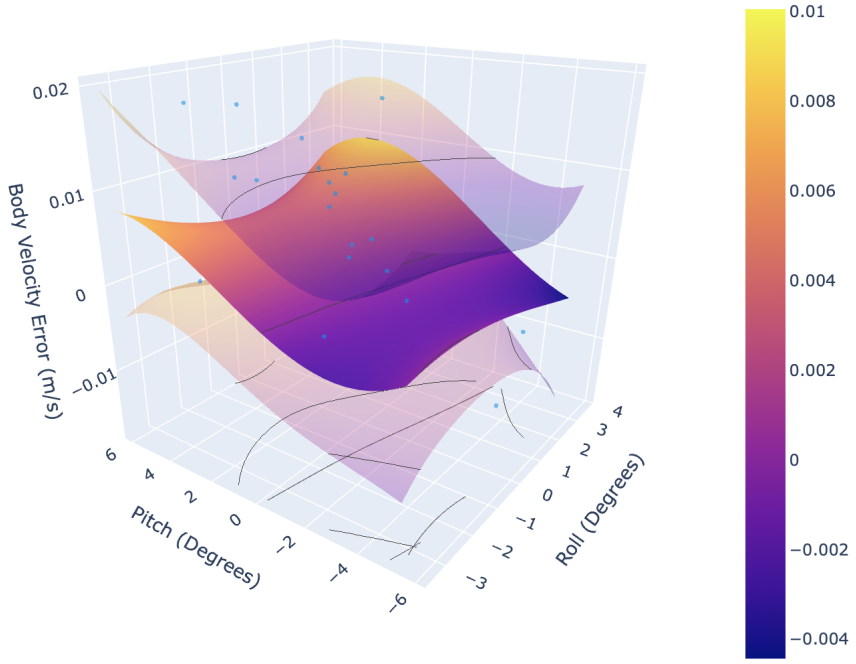

Figure B.1: Final GP Model - Side View 1

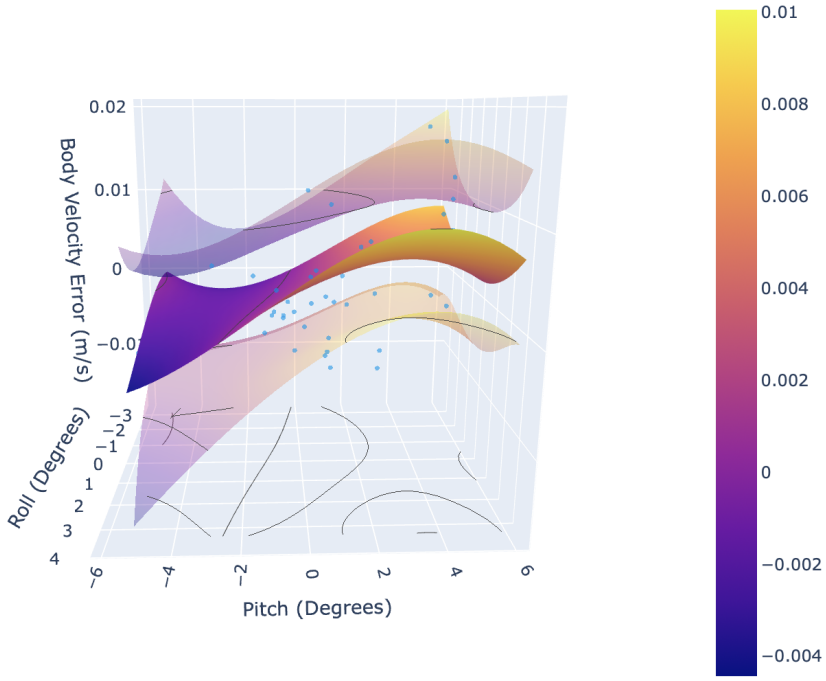

Figure B.2: Final GP Model - Side View 2

Bibliography

[1] Ramon Gonzalez and Karl Iagnemma. Slippage estimation and compensation for planetary exploration rovers. state of the art and future challenges. *Journal of Field Robotics*, 35(4):564–577, 2018. (document), 2.7, 2.3

[2] Neal Seegmiller. *Dynamic Model Formulation and Calibration for Wheeled Mobile Robots*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, October 2014. (document), 3, 3.2, 3.2.1, 3.2.4, 3.2.7, 3.4, 7

[3] Jatan Mehta. The bevy of rovers heading for the moon, Nov 2022. 1.1

[4] Mark Robinson and John Elliot. *Intrepid Planetary Mission Concept Study Report*. 2020. 1.2

[5] Lin Hui Li, Yi Bing Zhao, Jing Lian, and Shu Mei Wu. Driving control of planetary rover based on slip ratio estimation. *Applied Mechanics and Materials*, 121-126:3238–3242, 2011. 2.2

[6] D.M. Helmick, Yang Cheng, D.S. Clouse, M. Bajracharya, L.H. Matthies, and S.I. Roumeliotis. Slip compensation for a mars rover. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005. 2.2, 2.4

[7] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007. 2.2

[8] Sally Ward-Foxton. China, uae and usa head to mars – where is europe?, Jul 2020. 2.2

[9] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map SLAM. *CoRR*, abs/2007.11898, 2020. 2.2.1

[10] David Casagrande. Real-time featureless visual odometry for sea floor imaging with a lagrangian float. 2.2.2

[11] Eric T. Baumgartner, Hrand Aghazarian, Ashitey Trebi-Ollennu, Terrance L. Huntsberger, and Michael S. Garrett. state estimation and vehicle localization for the fido rover. *SPIE Proceedings*, 2000. 2.3

[12] Masataku Sutoh, Yutaro Iijima, Yuta Sakakieda, and Sachiko Wakabayashi. Motion modeling and localization of skid-steering wheeled rover on loose terrain. *IEEE Robotics and Automation Letters*, 3(4):4031–4037, 2018. 2.3, 3.1

[13] Genki Yamauchi, Keiji Nagatani, Takeshi Hashimoto, and Kenichi Fujino. Slip-compensated odometry for tracked vehicle on loose and weak slope. *ROBOMECH Journal*, 4(1), December 2017. Funding Information: (26This w·3767).ork was supported in part by a Grant-in-Aid for JSPS Research Fellow Publisher Copyright: © 2017, The Author(s). 2.3

[14] Kristin Bussmann, Lukas Meyer, Florian Steidle, and Armin Wedler. Slip modeling and estimation for a planetary exploration rover: Experimental results from mt. etna. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 2.3

[15] Neal Seegmiller, Forrest Rogers-Marcovitz, Greg Miller, and Alonzo Kelly. Vehicle model identification by integrated prediction error minimization. *The International Journal of Robotics Research*, 32(8):912–931, 2013. 2.3

[16] Neal Seegmiller and Alonzo Kelly. Enhanced 3d kinematic modeling of wheeled mobile robots. *Robotics: Science and Systems X*, 2014. 2.3

[17] Haibo Gao, Chao Chen, Liang Ding, Weihua Li, Haitao Yu, Kerui Xia, and Zhen Liu. Tracking control of wmrs on loose soil based on mixed h 2 / h control with longitudinal slip ratio estimation. *Acta Astronautica*, 140:49–58, 2017. 2.3

[18] Urara Kono, Hiroshi Fujimoto, and Yoichi Hori. Localization of wheeled mobile robots from slip ratio estimation with simple model. *2021 IEEE International Conference on Mechatronics (ICM)*, 2021. 2.3

[19] Chengchao Bai, Jifeng Guo, Linli Guo, and Junlin Song. Deep multi-layer perception based terrain classification for planetary exploration rovers. *Sensors (Basel, Switzerland)*, 19, 2019. 2.3.1

[20] Ramón González and Karl Iagnemma. Deepterramechanics: Terrain classification and slip estimation for ground robots via deep learning. *ArXiv*, abs/1806.07379, 2018. 2.3.1

[21] Carl Edward Rasmussen and Williams Christopher K I. *Gaussian processes for machine learning*. MIT Press, 2008. 2.3.1

[22] Tianyi Zhang, Song Peng, Yang Jia, Junkai Sun, He Tian, and Chuliang Yan. Slip estimation model for planetary rover using gaussian process regression. *Applied Sciences*, 12(9), 2022. 2.3.1, 4.4.1

[23] Cagri Kilic, Nicholas Ohi, Yu Gu, and Jason N. Gross. Slip-based autonomous zupt through gaussian process to improve planetary rover localization. *IEEE Robotics and Automation Letters*, 6(3):4782–4789, 2021. 2.3.1, 4.4.1

[24] Jie Wang. An intuitive tutorial to gaussian processes regression. Apr 2022. 2.3.1

[25] Kazuya Yoshida, Hiroshi Hamano, and Toshinobu Watanabe. Slip-based traction control of a planetary rover. *Springer Tracts in Advanced Robotics*, page 644–653. 2.4

[26] Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida. Slope traversal experiments with slip compensation control for lunar/planetary exploration rover. In *2008 IEEE International Conference on Robotics and Automation*, pages 2295–2300, 2008. 2.4

[27] Daniel M. Helmick, Stergios I. Roumeliotis, Yang Cheng, Daniel S. Clouse, Max Bajracharya, and Larry H. Matthies. Slip-compensated path following for planetary exploration rovers. *Advanced Robotics*, 20:1257 – 1280, 2006. 2.4

[28] Moonranger 4-wheel kinematics model, Nov 2021. 1

[29] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 5.4

[30] System evaluation - filter evaluation metrics. 5.4

[31] Informatik ix computer vision group, Mar 2016. 5.4

[32] Olivier Toupet, Jeffrey Biesiadecki, Arturo Rankin, Amanda Steffy, Gareth Meirion-Griffith, Dan Levine, Maximilian Schadegg, and Mark Maimone. Traction control design and integration onboard the mars science laboratory curiosity rover. In *2018 IEEE Aerospace Conference*, pages 1–20, 2018. 7.1, 7.4

[33] H.A. Oravec, X. Zeng, and V.M. Asnani. Design and characterization of grc-1: A soil for lunar terramechanics testing in earth-ambient conditions. *Journal of Terramechanics*, 47(6):361–377, 2010. 7.3